Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik

Bachelor Thesis Bioinformatics

The landscapes of CD8$^+$ T cell immunogenicity from a self-tolerance based perspective in sequence space

Manuel Glöckler

02.09.19

Reviewers

Dr. Leon Kuchenbecker
(Bioinformatics)
Applied Bioinformatics Group
Universität Tübingen

Prof. Dr. Oliver Kohlbacher
(Bioinformatics)
Applied Bioinformatics Group
Universität Tübingen

Glöckler, Manuel:
The landscapes of CD8$^+$ T cell immunogenicity from a self-tolerance based perspective in sequence space
Bachelor Thesis Bioinformatics
Eberhard Karls Universität Tübingen
Thesis period: 01.05.19-02.09.19

Abstract

Vaccination is one of the most successful medical treatments ever developed in human history. Traditionally vaccines are produced from parts of microorganisms or attenuated ones. However, this has several disadvantages in vaccine production and administration. For example, only a small fraction of the administrated material is relevant to induce an immune response; other redundant material can cause major complications. These disadvantages can be overcome with the use of synthetic peptide-based vaccines. Nevertheless, it remains challenging to find highly immunogenic antigens necessary for efficient vaccination. Thus, the development of methods that can accurately predict immunogenicity is of great interest; however, the prediction remains challenging. For the prediction of immunogenicity, the mechanism of self-tolerance is interesting. Immune cells reacting to self-antigens are negatively selected during their development, known as central tolerance, or are suppressed by regulatory mechanism, known as peripheral tolerance. Generally one would, therefore, assume that an immunogenic antigen should differ more from self-antigens than a non-immunogenic one.

This thesis investigates if this assumption can be observed through accurate similarity measurements in sequence space and thus be utilized to predict immunogenicity. We were able to show that a small set of experimentally validated MHC binding self-peptides is representative for the immunological relevant human proteome compared to comparable sets created through MHC binding prediction. Further, we could show that for the most part, a difference in the similarity to self-peptides is present between immunogenic and non-immunogenic peptides. For some HLA alleles, we detected a significantly lower similarity to self-peptides between immunogenic and non-immunogenic peptides. This scoring can be shaped by position-specific weights to obtain for the majority of investigating HLA alleles a significantly lower median similarity to self. Moreover, we investigated an alternative representation of peptides with residue feature maps from the AAIndex. We additionally mined for physicochemical properties of amino acid residues and could identify several indices in the AAIndex database that show significantly higher distances to self for immunogenic peptides. Furthermore, we determined peptide positional weight or feature maps that could boost the classification performance of a simple classifier.

Acknowledgements

Firstly I am deeply grateful to both Dr. Leon Kuchenbecker and Professor Dr. Oliver Kohlbacher to be able to investigate this topic.

Leon, as the advisor for all my thesis work throughout the past months, deserves special recognition for his always highly competent remarks and suggestions. He provided necessary data and literature, without this thesis would not have been possible. I would also like to thank Leon for constructive criticism of the manuscript during the thesis draft. His proofreading of several chapters was very helpful.

Additionally, I want to thank his colleague Leon Bichmann that involved himself several times in discussions and stand in during Leons holidays.

Finally, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study.

Contents

List of Figures

List of Tables

List of Abbreviations

MHC	**M**ajor **H**istocombapility **C**omplex
pMHC	**p**eptide-**M**ajor **H**istocombapility **C**omplex
HLA	**H**uman **L**eukocyte **A**ntingens
TCR	**T**-**C**ell **R**eceptor
BCR	**B**-**C**ell **R**eceptor
CD	**C**luster of **D**ifferentiation
HLA-A*01:01	HLA-[allele]*[supertype]:[subtype]
BLOSUM	**BLO**cks **SU**bstitution **M**atrix
k-mer	A k-mer is a subsequence of length k contained within a sequence.
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

In this chapter, the biological backgrounds of immunogenicity and its association with self-tolerance are introduced in Section 1.1. Basic knowledge of this subject is essential in order to understand the methods applied in this thesis to model self-tolerance on a sequence basis. In Section 1.2, the scope of this thesis and its relevance in bioinformatics are described.

1.1 Background

1.1.1 Immune system

The immune system protects the host from pathogenic entities. In vertebrates, it can be divided into two parts, the innate and the adaptive immune system. The *innate immune system* is the first line of defense and is made up of two main components. First, barriers to prevent the ingress of pathogens. Second, antimicrobial cells and molecules. The recognition of pathogens is mostly based on germline-encoded receptors that were evolutionary shaped to recognize molecular patterns present on pathogenic microbes. Its specificity is therefore constant over time and shaped to be self tolerant by evolution. Additionally, it is unable to develop a memory and is, therefore, less relevant in the context of vaccination. The most crucial part may be the presentation of antigens to induce an adaptive immune response through MHC class II molecules. However, this thesis focuses on MHC class I presentation, which is present on all nucleated cells and further explained in Subsection 1.1.2, thus the innate immune response is of little interest in this thesis.

On the other hand, the *adaptive immune system* consists of two types of lymphocytes, T and B cells. The antigen receptors for B cells are called immunoglobulins, which can be membrane bound as B-cell-receptor (BCR) or secreted as antibodies. T cells only express a membrane-bound T-cell-receptor (TCR) that is restricted to recognize peptide fragments presented by the major histocompatibility complex (MHC). We call such a peptide immunogenic or a T cell epitope if an immune response is induced. This MHC restriction reflects the functional difference. B cells secrete antibodies, and T cells are specialized for cell-cell-interactions, they either kill infected body cells ($CD8^+$ cytotoxic T cells) or interact with other immune cells in order to coordinate an immune reaction ($CD4^+$ helper T cells). In contrast to the innate immune system, the BCR and TCR are generated through a stochastic recombination process, called V(D)J-recombination. This process can generate a high number of different receptors, and therefore, each mature T/B lymphocyte expresses a different receptor. This diversity leads to the key principle for the high adaptability of the immune system, called clonal selection. In each individual, regardless of antigenic contacts, there is a large number of T and B lymphocytes. Because each cell expresses a different receptor, diversity at the level of the cell population is created. If one of these cells has contact with a suitable antigen, it proliferates and differentiates, creating a large number of short-lived effector cells and some long-lived memory cells that remain in the body and protect against a recurrent infection.

The different specificity of BCR and TCR implies that while BCRs can recognize a high number of highly diverse antigens, the TCR is restricted to peptides that are presented on MHC molecules. Therefore, this thesis focuses on T cell immunogenicity, as the antigenic identity is well defined and can be modeled more easily. It allows a sequence-based view of immunogenicity, where we can compare amino acid sequences rather than complex antigens. Additionally, the memorization through memory cells is the basis for all kind of vaccination. Introducing a pathogen-associated antigen that can trigger an adaptive immune response leads to the memorization of this antigen. If the antigen associated pathogen then infects the vaccinated individual, the memory cells can trigger a fast and effective response, preventing the formation of a disease. However, this adaptability comes at a cost. As receptors are created randomly, there is a chance that they recognize host-associated antigens, so-called self-antigens or in the context of T cell immunogenicity self-peptides. Such *autoreactivity* can lead to autoimmune diseases. To sustain the host's viability, a highly effective self-tolerance evolved, which will be described in the next section [MWSS18].

1.1.2 Self tolerance

Self-tolerance is defined as the ability of the immune system to recognize self-produced antigens as a non-threat while simultaneously maintaining sufficient immunocompetence. As T cells are MHC-restricted, the recognition of an antigen is dependent on peptide MHC presentation. For MHC class I, the peptides are usually derived from intracellular proteins in the cytoplasm. The proteasome processes these and TAP transports the peptide into the endoplasmatic reticulum. There they are loaded onto MHC class I molecules and transported to the cell surface. As such cytoplasmic proteins are either self-derived or from an intracellular infection, such pMHC class I complexes are recognized by cytotoxic killer cells that kill the corresponding cell if this peptide is recognized as a threat. As we restrict to MHC class I peptides in this thesis, we also only consider CD8$^+$ T cell immunogenicity; the corresponding TCR-pMHC interaction is discussed in Subsection 1.1.3. Since both self and foreign antigens are presented on MHC molecules, the immune system developed two main mechanisms to maintain self-tolerance — the central and peripheral tolerance [MWSS18].

Central tolerance

T cells develop from a lymphoid predecessor in the thymus. In this tissue, a unique T cell receptor is created randomly through V(D)J recombination. In early T cell development, the T cells with TCRs that interact weakly/intermediately with self-peptide-MHC complexes are positively selected through survival signals. This process ensures that the selected T-cells will have a sufficient MHC affinity, as a T cell have to interact with pMHCs for activation (MHC restriction). In a later stage of T cell development, T cells that strongly interact with self-peptides are negatively selected by apoptotic signals. This selection prevents the formation of self-reactive T cells that are capable of inducing an autoimmune disease. The mature T cell repertoire is, therefore, dependent on host proteome. However, a too strict selection would weaken immunocompetence, and therefore, some self-reactive T cells escape thymic selection, leading to the next mechanism of self-tolerance [MWSS18].

Peripheral tolerance

An additional mechanism of tolerance in the mature lymphocyte repertoire after the cells have left the central lymphatic organs is called *peripheral tolerance*. Fundamentally there are always at least two signals needed for T cell activation. Danger signals must accompany the TCR recognition of a specific antigen. The innate immune system transmits these only in case of tissue damage or infection. Since self-peptides are generally not accompanied by danger signals, an autoreactive T cell usually receives no costimulation. The missing costimulation prevents activation, and the cell goes into an inactive state called anergy. If it still comes to activation, other mechanisms may prevent an autoimmune reaction, for example, through inhibition by regulatory T cells [MWSS18].

By considering in this context an arbitrary peptide, we would expect a lower immunogenicity for peptides that are similar to self-peptides, as we would expect from a distant one. As self-peptides shape the T cell repertoire, it is less likely that a TCR recognizing a peptide similar to self exists. Additionally, even if such a peptide is recognized, the autoreactive T cell is likely inhibited by mechanisms of peripheral tolerance.

1.1.3 T cell receptor recognition

As discussed in the previous section, we would expect a lower immunogenicity for peptides that are similar to self-peptides. However, defining this similarity is not trivial because the TCR interaction with the peptide determines it. In other words, a T cell can only distinguish differences in peptides, if the TCR can perceive these differences. Hence the molecular mechanism of peptide recognition is essential to understand and discover features that affect recognition and, therefore, define how similar two peptides appear for a given TCR.

While the mechanisms of TCR recognition are still under research, from several crystal structures of TCR-pMHC complexes and other studies, some potential properties of TCR recognition were obtained. These showed that while TCRs bind in a conserved diagonal footprint to the pMHC complex, the atomic interaction varies widely [GTW99]. As MHC class I peptides are usually anchored at the peptide residue position two (P2) and position nine (P9), there is often a central bulge which dominates the interactions with the TCR. For nonamer peptides, these are represented mostly between P4-P8 [RW02, CdBK12, GTW99]. For MHC class II, the peptide has a conserved polyproline type II conformation, and the critical side chain interactions are more uniformly distributed [RW02, GTW99]. However, in both cases, the up-facing sidechains dominate TCR interaction, and only a weak contribution of the peptide's backbone was observed [GTW99]. This indicates that MHC bound peptide conformation is associated with immunogenicity. It has been observed that the same epitope can differ in immunogenicity for HLA subtypes, which only differ by a single amino acid. This behavior was not caused by differences in MHC presentation , rather by the conformation of the peptide [TEP^{+}05]. Additionally, local conformational changes through the binding to the pMHC class were observed, that may play a part in expanding TCR specificity [GTW99]. This flexibility may explain degenerated T cell recognition. T cells are cross-reactive and can recognize both similar and sometimes very different peptides presented on MHC molecules [CdBK12, FdBL^{+}08, WAC^{+}07]. It has been shown that a nonameric peptide contains enough information to differentiate between self and nonself peptides with a rather small identical overlap of 0.2%, calculated from the overlap of the human and several thousands of viral and bacterial proteomes. However, through the flexible recognition of the TCR, about one third of the nonself peptides is expected to be indistinguishable from self-peptides [CdBK12, MWSS18]. This degeneracy is especially interesting in terms of self-tolerance, as T cells should remain tolerant against this huge fraction of nonself peptides. It was shown that similar amino acid substitutions do not perturb T cell specificity, which further establishes that T cell epitopes should have low similarity to self-peptides [FdBL+08]. However, degenerated T cell recognition was also observed for peptides with low peptide similariy [WAC+07].

1.1.4 CD8$^+$ T cell epitopes

In summary, we can collect factors that distinguish MHC class I epitopes from non-epitopes. First, the abundance of pMHCs [KSW+08], the more pMHCs are expressed by the cell, the higher the chance that a T cell can recognize the corresponding antigen. Therefore a potential T cell epitope must be able to bind to an MHC sufficiently good. Additionally, the MHC binding has to be stable, as increasing half-life leads to an increased chance of T cell recognition [HRR+12, SVR+94, KSW+08]. Secondly, the peptide has to be recognized by a TCR, i.e.; it should be immunogenic. Thirdly even if a peptide is presented under the right conditions, if it is too similar to self, it may be blocked by mechanisms of peripheral tolerance.

We can conclude several critical factors that differentiate epitopes from non-epitopes on a sequence basis. Given an arbitrary peptide sequence, we first have to prove that it can bind to an MHC molecule, as peptides that do not bind on MHC molecules can not induce an immune response and can, therefore, be classified as non-epitopes. As sequence motifs determine MHC binding, some highly accurate binding predictors can satisfy this requirement [JPA+17]. Self-derived peptides that undergo MHC processing, transport and binding and are not only expressed in immunologically privileged tissues, shape the mechanisms of self-tolerance and are therefore most likely non-epitopes. A non-self peptide that is similar to such self-antigens, and thus indistinguishable for a T cell from self, should also be subject of self-tolerance and therefore should not represent an epitope [FdBL+08]. Accordingly, a CD8$^+$ epitope must fulfill the MHC presentation criteria and should be less similar to self-antigens as an MHC presented non-epitope, as otherwise, it should be subject of self-tolerance, too. However, the similarity is determined by the highly complex and degenerated T cell recognition mechanism. Modeling this accurately from sequence information is difficult; nonetheless, we can model several properties, e.g., residue position, residue similarity, and several physicochemical properties that have been associated with immunogenicity [CMG+13, FdBL+08, RW02].

1.2 Scope of this thesis

In order to develop a computer-aided system of peptide vaccine design, an accurate model of MHC peptide processing, presentation, and immunogenicity is needed. Many studies for MHC processing and presentation lead to powerful prediction tools that are nowadays well established. While binding affinity is a requirement for immunogenicity, its correlation with it is too weak to achieve a good predictive performance. It has been shown that binding stability correlates better with immunogenicity, but still not good enough for an accurate prediction [HRR+12, SVR+94]. However, these approaches only focus on good presentability of a peptide to T cells and therefore neglect the second requirement - TCR recognition. Early methods used sequence information of immunogenic and non-immunogenic peptides but could not obtain suitable predictive performance [TH07]. Newer approaches included systemic effects such as self-tolerance and could achieve better performance on a small but qualitative dataset [TFZ+11]. The latest approach quantifies immunogenicity by modeling TCR-pMHC contact potentials in sequence space [OY18]. However, a great breakthrough was not obtained by either of these approaches.

Additionally, Bresciani et al. associated immunogenicity of MHC class II peptides with a significantly lower median peptide similarity to the host's proteome [BPS+16]. If MHC class I peptide similarity to self can be assessed similarly on a sequence basis is an open question. As MHC class I peptide binding affinity is more predictive than that of MHC class II, the modeling of MHC restrictions is more reliably. Further, a recently obtained mass spectrometry-based HLA ligandome may allow for a more representative set of relevant self-peptides, as these peptides were observed to be presented on living self.

All in all, this raises several questions that will be answered by this thesis: Can MHC class I peptides be associated with a lower median peptide similarity to the relevant host proteome? Can the HLA-Ligandome represent this relevant proteome? Are there features that can more accurately describe the similarity of peptides to the TCR, based on the fact that the immune system has to distinguish epitopes from self? Can we reliably predict immunogenicity based on the expected differences between epitopes and non-epitopes to self-antigens?

Chapter 2

Methods and Material

As described in Chapter 1, self-tolerance describes the capability of the immune system to distinguish self from non-self and eliminate or inhibit self-reactive T cells. Therefore, MHC presented peptides that are very similar to self-peptides, are unlikely to induce a T-cell response and thus should have low immunogenicity. For a sequence-based analysis of self-tolerance, we, therefore, need a representative set of peptides from the human self-proteome, an adequate measurement of peptide similarity and a set of peptides with known immunogenicity. In this chapter, it is clarified in Section 2.1, which data sets for self proteome representation and immunogenicity were used. Secondly, the used similarity and distance metrics, as well as the implemented computational methods, will be described in Subsection 2.2.1 and Subsection 2.2.2. Thirdly an optimization technique based on a genetic algorithm will be introduced in Subsection 2.2.3, in order to optimize defined scoring metrics for a more potent measurement of the similarity to the self proteome.

2.1 Material

In the following section, we will describe three different ways of self proteome representation. Additionally, to associate immunogenicity with similarity to self, we need a set of peptides that are known to be immunogenic or not. The chosen dataset will be described in Subsection 2.1.2.

2.1.1 Self-proteome datasets

As described in Chapter 1, a requirement for a T-cell response is the presentation of a peptide on an MHC molecule. Therefore the sequence space of the human proteome can be reduced to the set of peptides that bind to MHC molecules. Three sets are used in this thesis: One experimentally validated set of peptides that are presented by MHC molecules on living cells, the

HLA-Ligandome. Second, a set of peptides from the human proteome that are predicted to bind to MHC molecules mentioned as the predicted proteome. Third, a collection of predicted MHC binding peptides that are expressed in the thymus mentioned as predicted thymus proteome.

HLA-Ligandome

The ligandome is an experimentally validated set of MHC binding peptides. Samples were obtained from different human tissues, except the thymus. From the extracted living cells, all pMHC complexes were collected, and MHC class I and II complexes were separated. The peptides were isolated from the MHC molecules, and the peptide sequence was determined through mass spectrometry. This allows determining peptides that are actually present on cell surfaces. These peptides successfully undergo the peptide processing machinery and bind sufficiently well to MHC molecules to be presented on the cell surface. However, this method does not allow to determine the HLA allele of the MHC molecule involved in the complex. To map the obtained peptides to HLA alleles, binding affinities were predicted using NetMHCpan 4.0 [JPA+17]. This process is described more detailed by Mayer et al. [BNB+19]. While using the same process, this inhouse dataset comprises measurements from healthy cells and is not yet publicly available at the time of writing.

The data set was filtered for MHC class I nonameric peptides. Duplicate entries were removed. This lead in total to 44,466 MHC class I peptides. As the HLA alleles are unevenly covered, only those who are well represented across all datasets were selected. This is mainly shaped by the number of peptides for that T cell assay results are available, as research is focused on special HLA-alleles, e.g., HLA-A*02:01. The selected alleles and their peptide frequencies are shown in Figure 2.1. In total, the filtered dataset includes 19,646 peptides.

Predicted ligands from the human proteome

The human proteome was obtained from Uniprot (ID: UP000005640, 16. May 2019) [Con18]. The state of the art MHC class I predictor NetMHCpan 4.0 was used for MHC binding prediction [JPA+17]. Because this thesis focuses on nonameric peptides, all 9-mers of the human proteome were predicted using the Python interface epitopepredict. This was done for all investigated HLA alleles.

Only peptides binding to MHC molecules can affect T-cell selection. As recommended by the NetMHCpan authors, we introduce a percentile rank cutoff at a percentile rank score of two. This lead, in mean, to 280,414 peptides per allele. As these peptides can be expressed in all tissues of the human body, this dataset should model peripheral and central tolerance explicitly.

The created set should represent a superset of the HLA-Ligandome as well. On average 95% of the peptides in the HLA-Ligandome are present in this set too. We will refer to this set as the predicted proteome in further sections.

Predicted thymus proteome

This set was used to evaluate a self-tolerance based T-cell epitope predictor and showed best performance. The dataset contains whole-genome microarray data from the NCBI Gene Expression Omnibus and the EBI ArrayExpress database. Because of conflicting measurements regarding the presence of proteins, the data was filtered to contain only proteins that are present and marginally expressed in the thymus [TFZ$^+$11].

For all 9-mers in the obtained proteins, the MHC binding affinity was predicted as described above. We obtained, on average, 8460 MHC binding peptides. The allele HLA-A*11:01 was underrepresented with only 466 MHC ligands. This set of peptides now represent the peptides that are present in the thymus and therefore models explicitly central tolerance. Implicitly this also models peripheral tolerance as these proteins can also be present in the periphery. The HLA-Ligandome does not contain samples from the thymus; thus, there is only an overlap of 3%.

2.1.2 Immunogenicity dataset

A recently released paper by Ogishi et al. [OY18] analyzed the immunogenicity in sequence space. In this project, peptides with functional T cell assay were collected from public databases (e.g., IEDB, LANL, HIV sequence Database, LANL HCV, EPIMHC, TANTIGEN, and data from several papers). For IEDB, peptides with inconsistent assay results were classified as immunogenic if at least one positive functional T cell assay exists. Peptides presented on nonhuman MHC molecules were excluded, while peptides presented on HLA in nonhuman hosts were included (e.g., transgenic mouses).

In terms of our analysis, this dataset was again filtered for nonameric MHC class I peptides. Because this thesis focuses on human self-tolerance, peptides from HLA in nonhuman host should be excluded as tolerance is shaped by the host's proteome. Therefore peptides were evidence for immunogenicity was only obtained in nonhuman hosts were excluded. However, the dataset contained some peptides with unannotated host's. These peptides were not extracted from IEDB instead from other databases or research project that often do not include host origin in output formats. Only the best-characterized epitopes were included from these databases. As the authors did not mention the inclusion of non-human hosts for these databases, we included these peptides into our analysis [OY18]. The frequency distribution of HLA alleles in this data set is shown in Figure 2.1 for immunogenic and non-immunogenic peptides.

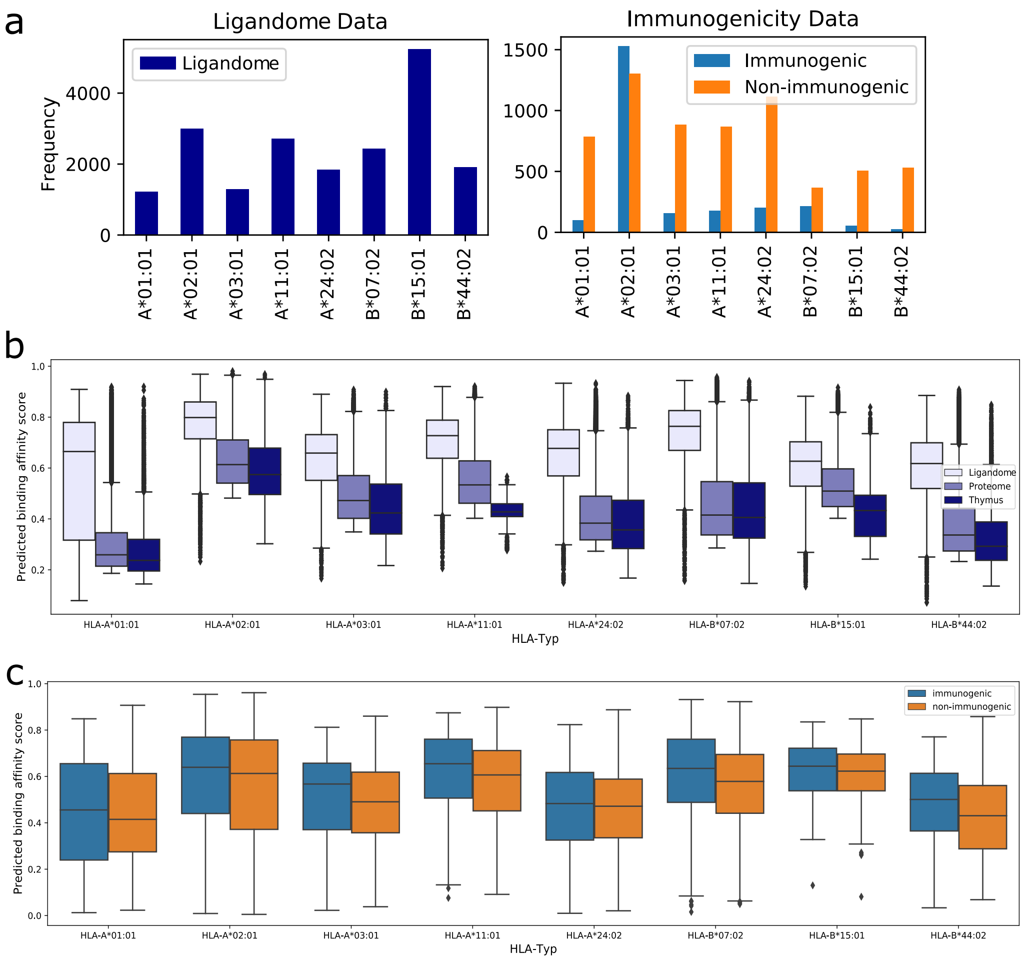


Figure 2.1: (a) The number of nonameric peptides for the selected HLA alleles in the HLA-Ligandome and for the immunogenic and non-immunogenic peptides. **(b)** The predicted binding affinity score obtained by NetMHCpan 4.0[JPA$^+$17] for the HLA-Ligandome, the predicted proteome and the predicted thymus peptides. **(c)** The predicted binding affinity score obtained by NetMHCpan 4.0[JPA$^+$17] for all immunogenic and non-immunogenic peptides.

Some peptides were labeled with inconclusive MHC notation (e.g., HLA-A, A1, B15 or NA). Only peptides which are mapped unambiguously to an HLA-subtype where included. As binding motifs of MHC molecules vary even for HLAs with a common supertype, including them would lead to noise in the data. Dissimilarities across peptides can then be explained by different HLA binding motifs instead of immunogenicity [JPA$^+$17].

Another requirement for our analysis is MHC binding. For some peptides, MHC binding evidence was given through MHC binding assays or MHC binding prediction. Around 33% of selected immunogenic and 5% of selected non-immunogenic peptides had annotated MHC binding assays. Unfortunately, for many peptides, there was no binding evidence annotated. However, for positive T cell assay result, we can assume MHC binding as otherwise, the assay should be negative. For non-immunogenic peptides, this is not the case; therefore, all immunogenic and non-immunogenic peptides binding affinity was scored using NetMHCpan 4.0 [JPA$^+$17] for their labeled allele. The obtained distributions are shown in Figure 2.1 c. For all HLA alleles, the immunogenic peptides had a slightly higher score than non-immunogenic ones. As binding affinity and stability correlate with immunogenicity, we can expect such a behaviour [HRR$^+$12, SVR$^+$94]. As the range of binding affinity scores for non-immunogenic peptides does not differ strongly from immunogenic peptides, these peptides were included even if no MHC evidence was annotated.

2.2 Methods

In the previous section, we described the collected data to represent the human proteome that is immunologically relevant for CD8$^+$ T-cell immunogenicity. If a peptide is similar to a peptide in these sets, we can expect that mechanisms of self-tolerance inhibit immunogenicity. This refers to the (k)-nearest-neighbor problem, given a suitable metric. We first defined a BLOSUM similarity score and introduced a trie-based branch and bound mechanism to solve the nearest neighbor problem in a suitable time. The used methods are described in Subsection 2.2.1. As an alternative approach, we also implemented the ability to describe peptides as numerical feature vectors based on residues feature maps form the AAIndex database. Thereby we can solve the (k)-nearest-neighbor search in numerical space with, e.g., the euclidean distance. Such residue feature maps can describe special physicochemical properties, and the nearest neighbor problem can be solved efficiently by KDTrees or Locality sensitive hashing (LSH). These algorithms and the scoring method are introduced in Subsection 2.2.2. To search for position-specific weights or relevant chemical properties, a genetic algorithm is introduced in Subsection 2.2.3.

2.2.1 Sequence similarity measurement

To compare the similarities of two peptides, we need to define a similarity score. A straightforward approach would be to count the different amino acids in the two sequences. However, this would neglect the physicochemical properties of different residues. We would expect a higher similarity comparing amino acids with similar properties as ones with different ones.

BLOSUM similarity score

A popular method to account for this is the use of scoring matrices, for example, the BLOSUM similarity matrix. These reflect log-odds scores of the substitution probabilities in conserved regions of protein families and therefore also reflect physicochemical properties of amino acid residues. As for substitutions of amino acids with similar properties, it is more likely to have a smaller impact on the structure and function of a protein than a replacement with an amino acid with different properties. Therefore we could define the similarity of two peptides as the sum of the corresponding BLOSUM scores. This will have the unfavorable effect that equal peptides have not necessarily the highest score. For this and the sake of comparability, this score is normalized as proposed by Bresciani et. al [BPS+16]. For two given peptides $A = a_1 \dots a_n$ and $B = b_1 \dots b_n$ of length n, the BLOSUM similarity score is given as

$$s(A,B) = \frac{\sum_{i=1}^{n} bl(a_i, b_i)}{\sqrt{(\sum_{i=1}^{n} bl(a_i, a_i)) \cdot (\sum_{i=1}^{n} bl(b_i, b_i))}} \tag{2.1}$$

in which $bl(x, y)$ is the BLOSUM62 score for the residue pairs x and y [HH92, BPS+16]. Now an identical match has a similarity score of 1.0, and the scores are normalized to a range between -1 and 1.

Similarity to self

Given the defined similarity measure, the similarity of a peptide to a set of peptides representing all self-peptides can be defined as the highest pairwise similarity to one of the peptides in this set. In other words, the nearest-neighbor problem has to be solved. Alternatively, we can also consider the k closest neighbors, leading to the k-nearest-neighbor problem. This allows considering peptides that are sufficiently similar but not the most similar, as self-tolerance may not only be dependent on the most similar, rather a set of highly similar peptides.

As shown in Section 2.1, the number of peptides representing the self-proteome can reach up to several hundred thousand. Therefore solving the (k)-nearest neighbor problem naively for several thousand query peptides is computationally expensive. In order to reduce time complexity, a trie based branch and bound algorithm was implemented. A trie is a special search tree representing a set of strings. Each string is represented as a path from the trie root to a leaf. Strings with a common prefix share the same path. Be k the length of n strings represented by a trie. Given a query string q of length k, finding an exact matching string requires a traversal from the root to a leaf. If for all characters in the query a path exists in the trie, the query string is contained in the trie. This leads to time complexity of $\mathcal{O}(k)$, compared to $\mathcal{O}(n)$ time for a linear search. As k is in our case substantially smaller than

n, finding an exact match is substantially faster. Given an adequate scoring matrix, like BLOSUM62, we can find the nearest neighbor by computing the score for all paths to a leaf in the trie. However, we can prune entire branches of the trie if the maximal obtainable score of a shared prefix is less then a score of the query with a complete peptide. This base idea is shown in Figure 2.2. For an additive similarity score and a given query peptide, we can define the maximal obtainable score between a query and a prefix from the search space, as the score between query and prefix plus the score of an identically matching suffix. The highest similarity of amino acid pairs is obtained from an equal match; therefore, an adequate similarity matrix should satisfy this condition.

The similarity score defined in Equation 2.1 is not additive in a straightway. Nevertheless, the sums of the BLOSUM62 scores are additive, and therefore, we calculate the bound as following:

- Because calculating the square root is computationally expensive we use for bounding the term

  $$s(A,B)^2 = \frac{\left(\sum_{i=1}^{n} bl(a_i,b_i)\right)^2}{\left(\sum_{i=1}^{n} bl(a_i,a_i)\right)\cdot\left(\sum_{i=1}^{n} bl(b_i,b_i)\right)}. \tag{2.2}$$

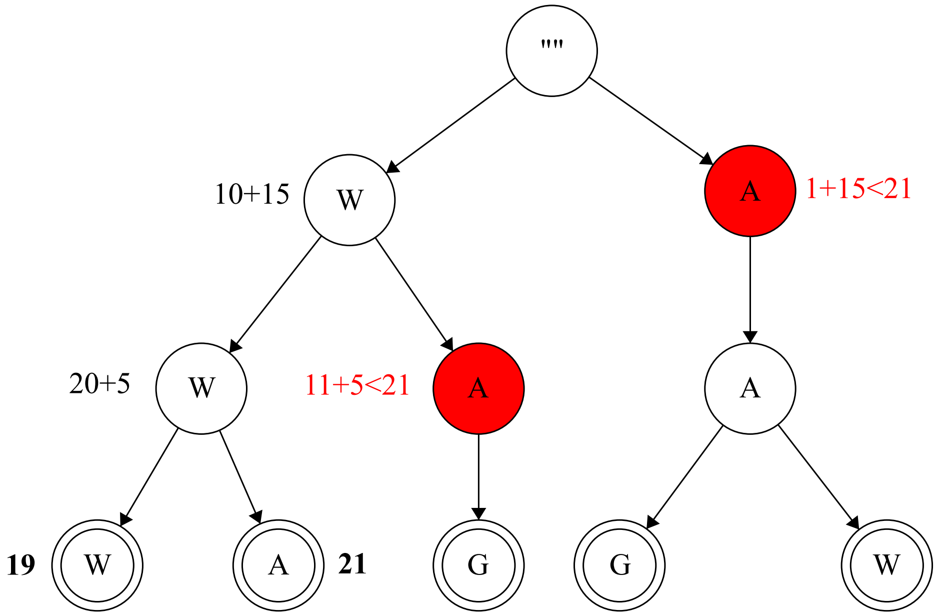
  The BLOSUM similarity score can then be obtained by rooting the value of Equation 2.2.

- Given is a query $A = a_1 \dots a_n$ and a prefix $B = b_1 \dots b_m$ of a sequence contained in the trie with $m < n$. Such a prefix is represented as a path from the root to an inner node. We define the maximum obtainable score as

  $$bound(A,B) = \frac{\Big(\sum_{i=1}^{m} bl(a_i,b_i) + \sum_{j=m+1}^{n} bl(a_j,a_j)\Big)^2}{\Big(\sum_{i=1}^{n} bl(a_i,a_i)\Big)\cdot\Big(\sum_{i=1}^{m} bl(b_i,b_i) + \sum_{j=m+1}^{n} bl(a_j,a_j)\Big)}. \tag{2.3}$$

The prefix of A and B is static, and the BLOSUM62 score can be calculated as usual. However, the suffix of B is unknown. Nevertheless, the highest obtainable similarity score for the given prefix is obtained, if the suffix match with that of the query A. If the thereby obtained maximal obtainable score is smaller than the score of a complete word, we can stop the search in this branch.

To find k-nearest-neighbors with this approach, firstly the best nearest neighbor is computed. To find the second-best nearest neighbor, the already found best neighbor is excluded from bounding. As the trie was already traversed hopefully using several bounds, the second-best bound can be reused in the search for the next nearest neighbor. Nevertheless, for increasing k,



Scoring Matrix:

	W	A	G	Y
W	10	1	-1	1
A		4	-2	-3
G			5	-2
Y				6

Trie contains, following strings:
WWW, WWA, WAG, AAG, AAW

Example query: WWG

Figure 2.2: An example trie for the strings WWW, WWA, WAG, AAG, and AAW with an example scoring matrix. To find the nearest neighbor for the query peptide WWG, we start at the left branch and compute the score $sc(WWW, WWG) = 19$ by traversing the trie to the first leaf. The current bound is now 19. Because in this branch are still leaves, we traverse back and consider the next leaf. As $sc(WW, WW) + sc(G, G) = 25$ there may be a better suffix in this branch, we detect that $sc(WWA, WWG) = 21$ is truly better and therefore update the bound. For the next branch the highest obtainable score is $sc(WA, WW) + sc(G, G) = 16$. As a higher score was already found, we can skip his branch. The next branch can be skipped again, leading to the nearest neighbor WWA with a score of 21.

the bounds get worse, and an increasing number of nodes has to be traversed. Thereby for big k's, it can be faster to compute all scores in a naive way, sort them and get the k highest scoring peptides.

Additionally, this method allows the use of position-specific weights. This is in our interest because as discussed in Subsection 1.1.3, the contacts with the TCR are not equally distributed over all peptide positions. Therefore it is maybe useful to weight functional hotspots higher than areas that have only minor influence in TCR recognition.

Problematic is the comparison of peptides with unequal length. First, the mechanism of extending peptide length in the MHC binding groove is mostly central bulging [SPHB00]. This leads to a different pMHC interface as residues that bulge out of the groove interact most with the TCR [RW02]. This questions if peptides without the same size are comparable. Second, a gap has to be introduced; however, as these peptides are bound to the MHC molecule in a fixed orientation and interact with specific positions that differ between peptides with unequal length, the gap placement is hard. A gap indicates that this position is not relevant for immunogenicity. However, which residues should be gaped is unknown and can not be obtained from a raw amino acid sequence, thereby structural information is necessary. If a simple alignment is done, then the gap is placed so that the global score is maximized. This can score peptides higher that are actually different from the perspective of a TCR. As this is impossible to model accurately only from sequence information, we reduced the problem to find the highest-scoring consecutive substring and hope that peptides that have highly common substring interact similarly with the TCR. Because MHC class I binding peptide length is usually between eight and eleven [SPHB00]. We, therefore, compute the BLOSUM similarity for MHC binding peptides as following. Given a query peptide and a search space of peptides that have a length between eight and eleven, we search for the highest-scoring 8 to 11-mer between the query and peptides in the search space. Is the query for example of length ten, then it consists of three 8-mers, two 9-mers, and one 10-mer. The nearest neighbor in the search space is, therefore, the highest-scoring 8, 9, or 10-mer that is present in one of the strings in the search space. However, solving this problem with a trie would neglect all time benefits, as for all non-prefix substrings the trie data structure will not boost the performance. We, therefore, just maintained all possible 8, 9, 10, and 11-mers of the search space in the trie. The nearest neighbor for a query peptide can then be obtained similar to the length invariant algorithm by finding the nearest neighbor of all 8, 9, 10 and 11-mers of the query peptide, returning the longest highest-scoring k-mer found in the trie. Nonetheless, this does not accurately model the consensus view that variations in peptide length are possible through a central bulging mechanism with anchor residues still at peptide position two and the C terminal end. However, in some cases, it has been shown that protrusion is the mechanism of extension, which would be modeled reasonably well [SPHB00].

All in all, this leads to good time improvement, especially for a high number of peptides. However, the similarity computation through scoring matrices severely limits the possibilities. Especially in comparison to numerical space, where highly efficient nearest-neighbor algorithms are available. This fact makes a mapping of amino acid sequences into numerical vectors attractive, leading to the next section.

2.2.2 Feature mapped sequence distance measurement

In contrast to the sequence comparison with scoring matrices, another widely used technique is to encode a given peptide with a numerical feature vector. This has the advantage that we can choose from arbitrary mappings of amino acids to numerical value's, representing specific physicochemical properties. The AAIndex is, for example, a database that contains over 500 such amino acid indices [KK00]. One possible representation is presented in Section 2.2.2. Because a sequence is now represented as a numerical vector, several efficient nearest neighbor algorithms are available and will be discussed here.

Residue feature distance score

There are several ways of encoding a peptide as a numerical vector. Generally, an amino acid sequence of length n is mapped into a numerical vector $v \in \mathbb{R}^m$, where m can be smaller or substantially greater than n. One method used in this thesis is with multiple amino acid indices from the AAIndex database. One AAIndex maps every amino acid to a real number and therefore one possible encoding is to map a given peptide of length n to a vector $v \in \mathbb{R}^n$ [KK00]. One index often represents only a single property, but sometimes it is necessary to consider multiple properties. We can represent a peptide of length n, encoded by k indices, as a vector $v \in \mathbb{R}^{n \cdot k}$. If feature maps have a different range of values, this leads to an unequal weighting of different features. If this is not desired, then mappings should be normalized.

As the corresponding peptides are now represented as numerical vectors, we can define the distance between peptides as the distance between the encoded feature vectors represented by a proper distance metric. We choose the euclidean distance. For two given vectors $v, w \in \mathbb{R}^n$ it is defined as:

$$d(v,w) = ||v - w||_2 = \sqrt{(v_1 - w_1)^2 + \ldots + (v_n - w_n)^2} \tag{2.4}$$

Feature encoded peptide distance to self

As for the similarity, the distance to self can be defined as the lowest pairwise distance to a self-peptide, which again represents the (k)-nearest-neighbor problem. Since the peptides are now encoded as numerical vectors, and a proper distance metric is used, there are several powerful ways to solve this problem.

If n peptides can be described by n real values, we can sort the set in $\mathcal{O}(n \log n)$ time and search a query in $\mathcal{O}(\log n)$ by binary search. If n peptides are described by n k-dimensional vectors, we can similarly construct a tree, where we start at the top node and decide at each node if the query object is in the left or right branch. This solves the query problem in $\mathcal{O}(\log n)$ and leads to the k-d tree algorithm. Thereby the nearest neighbor problem can be solved in $\mathcal{O}(\log n)$. However, in high dimensional space, we mostly end up testing nearly all nodes, and the complexity grows to $\mathcal{O}(n)$ [SC08]. Therefore a hashing technique for a fast approximate nearest neighbor search was implemented, which will be described in the next section.

Locality sensitive hashing

As optimal nearest neighbor methods struggle with the curse of dimensionality, a common technique is to create a hashing function that separates search space into bins. The nearest neighbor for a query can then be obtained by determining the nearest neighboring element in its corresponding bin. In the sight of nearest neighbor search, we want therefore a hashing function that hashes a query point into a bin that contains points that are close to the query, as these are potential nearest neighbors. One technique is locality sensitive hashing (LSH), and the chosen implementation is introduced now.

The fundamental principle of LSH is that if two points are close together and a projection operation is applied, it is likely that the projected points are still close to each other. This can easily be visualized in two-dimensional space, as shown in Figure 2.3. If we linearly project points to a random line that is divided into quantification buckets of width w, we observe that two distant points are only likely to fall into the same bucket, if the projection line is approximately orthogonal to these points. For the most other orientations, the points will fall in different buckets. This allows hashing high dimensional points into bins where points that are less distant to each other are likely to fall into the same bin.

Formally this can be implemented as follows. The core of the hash functions is a scalar projection or dot product with a projection vector drawn from a standard normal distribution $\mathcal{N}(0,1)$. To quantify this into buckets that are represented by an integer, we choose the following hash functions:

$$h^{\vec{v},b}(\vec{x}) = \left\lfloor \frac{\vec{x} \cdot \vec{v}^T + b}{w} \right\rfloor \tag{2.5}$$

where w is the quantization bin width, b a random variable uniformly drawn from $[0, w]$, $\vec{x}$ the vector to hash and in our case representing the peptide. The

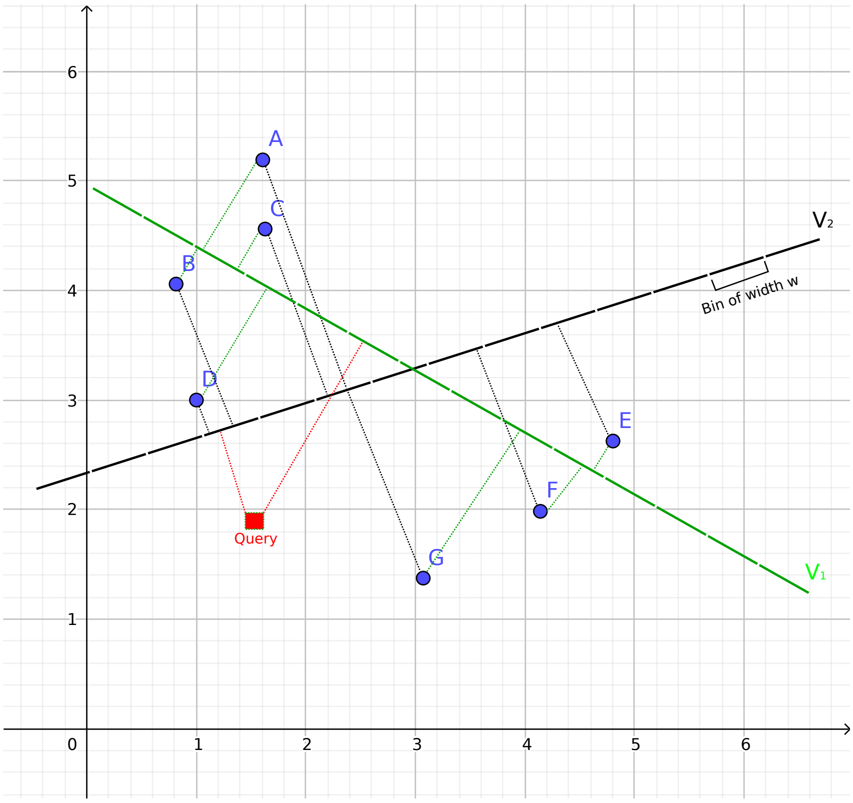


Figure 2.3: Two different scalar projections on V_1 and V_2, are shown. These are separated into quantification bins of width w. Data points (blue, A-G) that are close to each other fall into the same or in neighboring bins for both projections. For a given query point (red), we hash it with the same projections. As observable, the query is mapped to the same bucket as the real nearest neighbor D for the projection V_2. For V_1, it falls into an empty bucket; nonetheless, the true nearest neighbor is in the next closest occupied bin.

elements of the projection vector $\vec{v}$ are independently chosen from a Gaussian normal distribution.

That such a random projection hash points that are close to each other with a high probability into the same bucket is visualized in Figure 2.3. However, formally, this is caused by a property of Gaussian normal distributions called p-stability. A distribution D is p-stable if for any independent identically distributed random variables $V_1, \ldots, V_n$ distributed according to D and any real numbers $x_1, \ldots, x_n$, the random variable $\sum_i x_i V_i$ is distributed as

$$\left(\sum_i |x_i|^p\right)^{\frac{1}{p}} V.$$

The random variable V is distributed according to D. A Gaussian probability distribution is 2-stable. Given two points p, q and the hash function h defined in

Equation 2.5, this property leads to the following: The hash function projects the points p, q onto a real line as illustrated in Figure 2.3. From p-stability follows that for two points with a corresponding distance $||p-q||_2$ to each other, the distance between their projection $(p\vec{v} - q\vec{v})$ is distributed as $||p - q||_2 V$, where V is a Gaussian distributed random variable [DIIM04]. If we chop this projection line into bins of width w the projected points are likely to fall in the same bin if they have a low distance to each other. We can conclude that if p, q are close to each other, there is a high probability P_1 that they fall into the same bucket. For points that are far from each other, there is a lower probability P_2 that they fall into the same bucket. We can increase the separation of points that are more distant to each other by performing k independent dot products. As $P_1 > P_2$ this increase the ratio that points at different distances fall into the same bucket, as $\left(\frac{P_1}{P_2}\right)^k > \frac{P_1}{P_2}$. This will decrease the bin size and increase the number of bins, as only less distant points will likely fall into the same bin. However, this also decreases the probability that close points fall into the same buckets, as $P_1^k < P_1$. To neglect this effect, L independent projections are made, as it is unlikely that true near neighbors are not in the same bucket in all L projections. Increasing the bucket width w will lead to an increasing number of points in the bucket [SC08, DIIM04].

To solve the nearest neighbor problem given a set of self-peptides S and a query q represented as real vectors, we now can compute the nearest neighbor as following:

1. Hash each self peptide $s \in S$ with k independent hash functions, as defined in Equation 2.5, into buckets characterized by a key $k \in \mathbb{N}^k$. Save this hashing in a hash table H.

2. Create L hash tables H as described above.

3. Given a query q we compute for each hash table H the corresponding hash keys $k_1, \ldots, k_L$. For all self-peptides $s \in h(k_1), \ldots, h(k_L)$ we can compute the nearest neighbor with an optimal method of choice. If all bins are empty, we search for the next closest bin and perform a nearest neighbor search there.

All in all, the search space is thereby reduced to the number of elements contained in these bins. We also ensure that the actual nearest neighbor is likely to be in this set, which can be further increased, by increasing L or the bin width w to an almost optimal nearest neighbor algorithm. Even if the method fails to find the true nearest neighbor, most likely a point that is close to the query is returned. Increasing k leads to a decrease in the number of elements per bin and therefore to benefits in computation time. With a good combination of these parameters, we can obtain a highly accurate and time-efficient nearest neighbor search for an arbitrarily large search space.

2.2.3 Evolutionary algorithm

In the sight of self-tolerance, we would expect immunogenic peptides to be less similar to self-peptides. However, as explained in Subsection 1.1.3 this similarity is determined by the TCR. As TCR interaction varies between residue position, the similarity between peptides is dependent on similarities between specific peptide positions. Additionally, immunogenicity has been associated with chemical properties, and therefore, special residue features may play a role in TCR recognition [CMG+13, CdBK12, FdBL+08].

On an MHC binding nonameric peptide, each position has a different influence on immunogenicity. For example, P2 and P9 are often anchor residues and necessary for MHC binding. On the other hand, peptide conformation can vary strongly in MHC class I with TCR interacting residues mostly between P4-P8 [CdBK12, RW02]. We are interested in residues positions that are associated with immunogenicity, not MHC binding, as all peptides are already binders. Some residues have less interaction with TCR and are therefore assumed to have less impact on immunogenicity [CdBK12]. Such position-specific relevances can be modeled with position-specific weights. Residue positions with high weight resolve high match rewards, but also high mismatch penalties, according to the defined BLOSUM-based similarity score. The nearest neighboring self-peptide would, therefore, contain most likely a similar amino acid residue at a high weighted position. If we weight the right positions high, we can more accurately describe residue positions that are more relevant for TCR recognition. On the other hand, we can model relevant physicochemical properties of amino acid by describing the corresponding peptides with different feature mappings, e.g., as defined in Subsection 2.2.2. Immunogenicity has not been associated with a single AAIndex [CMG+13]; however, a combination is may feasible.

Finding such combinations of physicochemical properties or position-specific weight refers to an optimization problem. As we expect lower similarity to self for immunogenic peptides, compared to non-immunogenic ones, we can try to optimize this expectation for the underlying data. However, such optimization is hard as the search spaces are huge. The AAIndex consist of approximately 500 different indices and therefore we obtain $500 \cdot 499 \cdot 498 \cdot 497 \cdot 496 \approx 10^{13}$ different distinct combination of five indices. If we restrict the position-specific weights to five integer values, we obtain $5^9 \approx 10^6$ different combinations. In order to evaluate scoring, we need to compute the whole BLOSUM similarity score distributions to self-peptides represented through, e.g., the HLA-Ligandome. Even with the optimized computational methods, this is time-intensive, and therefore, an optimal optimization method is not calculatable in feasible time.

An evolutionary algorithm is a metaheuristic optimization algorithm, which can handle such big search spaces and still obtain good results in a feasible time. The base algorithm can be described as following:

1. Create a random starting population of N individuals that are represented through chromosomes that contain m genes. Each gene represents a parameter that is coined into a chromosome that fully characterizes the desired model.
2. Repeat the following steps for several generations until convergence:
    - Evaluate fitness of all individuals by a fitness function f.
    - Select the fittest individuals. Individuals with higher fitness have a higher chance to be selected.
    - Breed the selected individuals and apply a cross-over-technique to produce the next generation.
    - Mutate the next generation by a given a mutation probability.
3. Select the fittest individual in the last generation.

Restricting weights to integer weights in a given range creates combinations in a countable search space. For our optimization problem, an individual can then be represented as a list of position-specific weights or a list of AAIndex indices that describe different amino acid properties. Each distinct weight or index, therefore, represents a gen of a chromosome. We can then generate a starting population by selecting random weights or indices for each chromosome [Whi94].

The fitness of an individual can then be determined by minimizing the median similarity of immunogenic peptides compared to non-immunogenic ones. However, this will overfit the solutions to our assumptions, as we force the expected behaviors to be present.

We also created an alternative fitness function using a support vector machine (SVM) to classify peptides according to their BLOSUM similarity score or feature mapped distance score, given a list of weights or AAIndices. This approach will not force the expected lower immunogenic similarity to be present; it will increase the separability of immunogenic and non-immunogenic peptides. Furthermore, it gives an insight of how much these optimizations can increase the predictability. An SVM is a machine learning approach for supervised learning. The SVM algorithm separates a set of points into classes so that around the class boundaries, a range as wide as possible remains free from points. Starting from a set of training points with labeled classes, each point is represented by a vector in a vector space. The SVM algorithm now finds a hyperplane that separates these two classes, according to an error term, best. The penalty parameter C can determine the tradeoff between minimizing training error and maximizing the margin. As a hyperplane can not be bend, a clear separation is only possible for linearly separable points. Nevertheless, we can use the kernel-trick to obtain a non-linear boundary by projecting data in higher dimensional space in which we hope to obtain a better separability [BC00]. In our case, a point can be represented as, e.g., the BLOSUM score of the k-nearest neighbors. The immunogenic and non-immunogenic datasets have an unbalanced number of peptides. As non-immunogenic tend to have more, this would lead to an imbalanced training. As the SVM algorithm try to minimize the training error, it will most likely classify most peptides as non-immunogenic as most classifiers seeking an accurate performance over the full range of training data. Therefore we apply a class weighting in training, for, e.g., n immunogenic peptides and m non-immunogenic peptides, we give the immunogenic class a weight of $\frac{m}{n}$ and the non-immunogenic class a weight of 1. This simulates equally sized datasets and allows the training of an SVM with imbalanced data sizes. We use stratified k-fold-cross-validation to avoid overfitting. Thereby we fold our data into k smaller sets and train the SVM k times with $k-1$ sets, while testing classification only with the test set excluded from training. This will prevent overfitting and therefore ensure generalizability. To find the best fitting individual, we can now try to maximize the cross-validation accuracy. However, for imbalanced data, accuracy is not a good choice as it is strongly biased by the most abundant class. For example, if we have three times more non-immunogenic peptides than immunogenic ones, the accuracy for classifying all peptides as non-immunogenic is 75%, while that of classifying all peptides as immunogenic is only 25%. Therefore, we choose the F1-score, which is a function of precision and recall. Precision is defined as the number of true positives divided by all positive predictions and recall as the number of true positives divided by the actual number of positives. The F1-score is then defined as

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}.$$

Maximizing this with the genetic algorithm will seek a balance between Precision and Recall, and leads to a measurement of accuracy that is not affected by the large numbers of non-immunogenic peptides [GYD$^+$08, AC10, PVG$^+$11].

We added elitism into the genetic algorithm, which includes a small amount of the fittest individuals unchanged into the next generation. A tournament selection method was implemented. For a given tournament size, this number of individuals is randomly sampled from the population. The winner of the tournament, the fittest individual, is selected. For the thereby selected parents, a uniform cross-over technique is applied. The genes for a child are uniformly chosen from its parents. Mutations of a gen follow a given probability and replace a gen with a random one.

These basic principles allow us to start with a starting population that is substantially smaller than the search space but still obtain good results in optimization. This reduces computational time enough to allow the analysis of position-specific weights and chemical properties based on our sequence-based view of self-tolerance [Whi94].

2.2.4 Further methods in computational analysis

The previously described methods were implemented in Python. They are published as the Python module "pepdist" on Github. These methods were used to compute the BLOSUM similarity score and feature mapped distance score to the self proteome representations and optimize these through the genetic algorithm. We used the Python module scipy for statistical tests and for a KDTree implementation [JOP+]. For plotting, we used seaborn and matplotlib [Hun07]. The SVM classifiers and cross-validations were performed using scikit-learn [PVG+11].

Chapter 3

Results and Analysis

We first computed the BLOSUM-based similarity between peptides of known immunogenicity against the nearest neighbor of a reference peptide population using the k-nearest-neighbor approach for the three different self proteome representations defined in the previous chapter. To investigate the influence of residue positions on the BLOSUM score, we applied a genetic algorithm to obtain position-specific weights. Subsequently, we evaluated the scoring distribution for feature encoded peptides based on the distance to self. In a similar fashion, we used a genetic algorithm to obtain residue features that can be associated with immunogenicity.

3.1 BLOSUM-based similarity to self

We computed the BLOSUM-based similarity score using the methods discussed in Subsection 2.2.1. The score was calculated for the HLA-Ligandome, predicted proteome, and predicted thymus proteome. Position-specific weights were determined that minimize the immunogenic median BLOSUM similarity to the nearest neighboring self-peptide, compared to that of a non-immunogenic one. Similarly, we determined weights that maximize the cross-validation F1-score of a simple SVM classifier.

3.1.1 BLOSUM similarity to HLA-Ligandome

We computed the scores with the standard BLOSUM62 matrix [HH92]. As all peptides have an equal length, the score was calculated globally, and no weights were applied per position. We calculated the score for all selected HLA alleles HLA-A*01:01, HLA-A*02:01, HLA-A*03:01, HLA-A*11:01, HLA-A*24:02, HLA-B*07:02, HLA-B*15:01 and HLA-B*44:02. For all peptides in, e.g., HLA-A*01:01, the BLOSUM similarity score is calculated as the highest pairwise similarity to self-peptides in the HLA-Ligandome also presented on

HLA-A*01:01. In other words, the nearest neighbor problem was solved for all peptides presented on the same HLA allele using the trie based approach explained in Subsection 2.2.1. We found 2, 38, 0, 1, 0, 1, 2 and 0 exact matches for immunogenic peptides and 109, 84, 30, 21, 3, 20, 349 and 47 matching non-immunogenic peptides for their corresponding allele. We plotted the obtained BLOSUM similarity score to the nearest neighboring self-peptide as an empirical cumulative distribution function, which are shown in Figure 3.1. The exact matching peptides are excluded. Similarly, cumulative empirical distributions were created for the mean BLOSUM similarity score of the 10 and 100 nearest neighbors. These are shown in Figure 1 and Figure 2. A two-sample Kolmogorov-Smirnov test was performed to determine if there is a significant difference in these distributions. The null hypothesis that two independent samples are drawn from the same continuous distribution is rejected if the p-value is below 0.05. The corresponding p-values are listed in Table 3.1.

Table 3.1: P-values of a two-sample Kolmogorov-Smirnov test for the obtained distributions against the HLA-Ligandome using the k-nearest-neighbor mean BLOSUM similarity score for $k \in \{1, 10, 100\}$. Values below the significance threshold are highlighted.

HLA	KS-Test (k=1)	KS-Test (k=10)	KS-Test (k=100)
A*01:01	**0.0371**	0.0568	0.0603
A*02:01	$\mathbf{8.146 \cdot 10^{-7}}$	$\mathbf{1.3512 \cdot 10^{-9}}$	$\mathbf{2.8029 \cdot 10^{-11}}$
A*03:01	**0.0074**	**0.0004**	**0.0003**
A*11:01	**0.0002**	$\mathbf{2.8178 \cdot 10^{-7}}$	$\mathbf{2.3471 \cdot 10^{-7}}$
A*24:02	0.0955	**0.0198**	**0.0092**
B*07:02	0.2825	**0.0248**	0.3285
B*15:01	0.1088	**0.0088**	**0.0144**
B*44:02	0.4722	0.2803	0.4803

We recognized a trend of decreasing p-values for increasing values of k. However, in some cases they decreased, especially for $k = 100$. For HLA-A*01:01, HLA-A*24:02 and HLA-B*15:01, we detected a trend of immunogenic peptides to be less similar to peptides in the HLA-Ligandome than for non-immunogenic ones. According to a Mann-Whitney rank test, the median BLOSUM similarity is significantly lower than that of non-immunogenic ones for HLA-B*15:01 (p-value < 0.004) and HLA-A*24:02 for k=10 (p-value < 0.013). The BLOSUM score distributions were significantly different for most alleles considering all $k \in \{1, 10, 100\}$. However, for HLA-A*02:01, HLA-A*03:01 and HLA-A*11:01, we observed a trend of immunogenic peptides to be more similar to the HLA-Ligandome than for non-immunogenic ones. We saw no significant difference for HLA-B*44:02 and, except for k=10, for HLA-B*07:02.

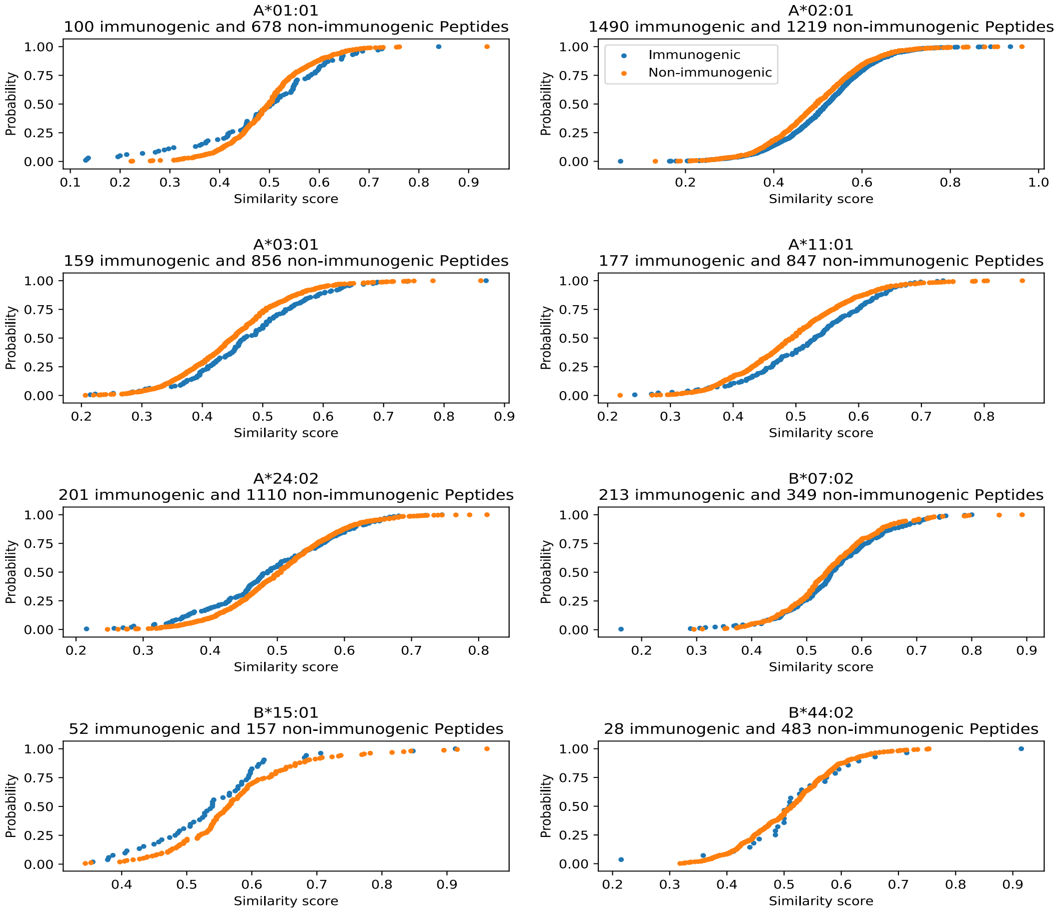


Figure 3.1: Empirical cumulative distributions of the BLOSUM similarity score from the nearest neighbor in the HLA-Ligandome. The titles contain the corresponding HLA allele and number of included peptides, excluding exact matches.

Length invariant BLOSUM similarity to HLA-Ligandome

We extracted all peptides from the corresponding datasets, as described in Section 2.1. However, we did not restrict only to nonameric peptides this time. Thereby we obtained peptides with lengths between 8 and 12. As described in Subsection 2.2.1, we computed the BLOSUM similarity to self as the score of the highest scoring consecutive substring. The resulting empirical cumulative score distributions against the length invariant HLA-Ligandome are shown in Figure 3. These are similar to the distributions considering only nonameric peptides. However, they differed less, according to a two-sample Kolmogorov-Smirnov test their distributions were different only for HLA-A*02:01, HLA-A*03:01, HLA-A*11:01 and HLA-B*15:01 (p-values < 0.01). Immunogenic peptides had according to a Mann Whitney rank test a significantly lower median BLOSUM score for HLA-B*15:01 (p-values < 0.008). A BLOSUM score of one now additionally represents an exact matching substring. For the corresponding alleles we obtained 2, 54, 1, 1, 2, 2, 4, 0 and 0 identical immunogenic substrings and 198, 122, 47, 31, 3, 32, 571, 109 and 0 identical non-immunogenic substrings. These numbers are slightly higher than for nonameric peptides; however, we simultaneously consider more peptides.

Dependency on MHC binding affinity

Immunogenic as well as non-immunogenic peptides are validated or predicted to bind to an MHC molecule. This fact should neglect differences that are caused by MHC binding motifs and not by immunogenicity. Nevertheless, as shown in Figure 2.1, the distributions of binding affinity scores differ slightly, i.e., immunogenic peptides tend to have a higher affinity. Because we can expect a higher similarity between peptides that both have high binding affinities as for peptides that differ in affinity, a higher BLOSUM similarity score to self can be partially explained by the fact that the HLA-Ligandome and immunogenic peptides have a higher binding affinity.

To investigate this, we equalized the binding affinity distributions in immunogenic and non-immunogenic datasets. If different binding affinities cause the observed differences in score distributions, the differences should vanish for peptides with equal binding affinity distributions. Therefore, we binned all immunogenic peptides in ten quantification bins according to their predicted binding affinity score. We receive a histogram which approximates the immunogenic binding affinity distribution. For these quantification bins, we randomly subsample without replacements from the non-immunogenic peptides the same frequencies as for the immunogenic set. According to a two-sample Kolmogorov-Smirnov test, both sets are then drawn from the same distribution (p-values > 0.8). The random subsampling was repeated 100 times, and for the obtained sets the BLOSUM similarity to the HLA-Ligandome is computed as previously discussed. The obtained distributions are shown in Figure 4. We detected some variance, but on average, they were similar to the distributions for the whole data sets.

3.1.2 BLOSUM similarity to the predicted proteome and thymus proteome

Similarly to the HLA-Ligandome, the BLOSUM-based similarity scores against the predicted human proteome and thymus proteome were computed using the same k-nearest-neighbor approach with $k \in \{1, 10\}$ for all HLA alleles. The obtained empirical cumulative BLOSUM score distributions are shown in Figure 3.2 and Figure 3.3. Keep in mind that equal matches are excluded from these distributions, for example in HLA-B*15:01 of the predicted proteome we found 461 non-immunogenic peptides in the self proteome, which causes a remarkably different distribution compared to the HLA-Ligandome or predicted thymus proteome. The number of equal matches are listed in Table 3.2. We observed a higher number of equally matching peptides in the predicted proteome as in the HLA-Ligandome or thymus proteome. This behavior is expected as it contains substantially more self-peptides. Yet, the thymus proteome also contains more peptides than the HLA-Ligandom but shows a smaller number of identically matching peptides. A two-sample Kolmogorov-Smirnov test was again used to evaluate the significance of the differences between both distributions. The obtained p-values are shown in Table 3.3. Generally, the p-values were consistently higher in the predicted proteome compared to the HLA-Ligandome or thymus proteome, except for HLA-A*01:01.

Table 3.2: The number of exactly matching peptides for immunogenic and non-immunogenic peptides in the predicted proteome and thymus proteome.

	Proteome		Thymus	
HLA	immunogenic	non-immunogenic	immunogenic	non-immunogenic
A*01:01	6	127	1	9
A*02:01	271	175	11	11
A*03:01	14	68	0	0
A*11:01	8	28	0	0
A*24:02	37	14	0	2
B*07:02	8	53	0	2
B*15:01	3	461	0	4
B*44:02	2	61	0	1

Table 3.3: P-values of a two-sample Kolmogorov-Smirnov test for the obtained similarity distributions using the k-nearest-neighbor BLOSUM similarity for all allele types with $k \in \{1, 10\}$ against the predicted proteome and thymus set. Significantly different distributions are highlighted.

	Proteome		Thymus	
HLA	KS-Test(k=1)	KS-Test(k=10)	KS-Test(k=1)	KS-Test(k=10)
A*01:01	**0.036**	**0.022**	**0.054**	**0.019**
A*02:01	0.582	0.315	0.054	**0.019**
A*03:01	0.950	0.482	**0.031**	0.190
A*11:01	0.754	0.414	0.164	0.063
A*24:02	0.065	**0.028**	**0.039**	**0.002**
B*07:02	0.489	0.156	0.264	0.156
B*15:01	0.338	0.277	**0.003**	$\mathbf{2.3 \cdot 10^{-5}}$
B*44:02	0.459	0.895	0.909	0.905

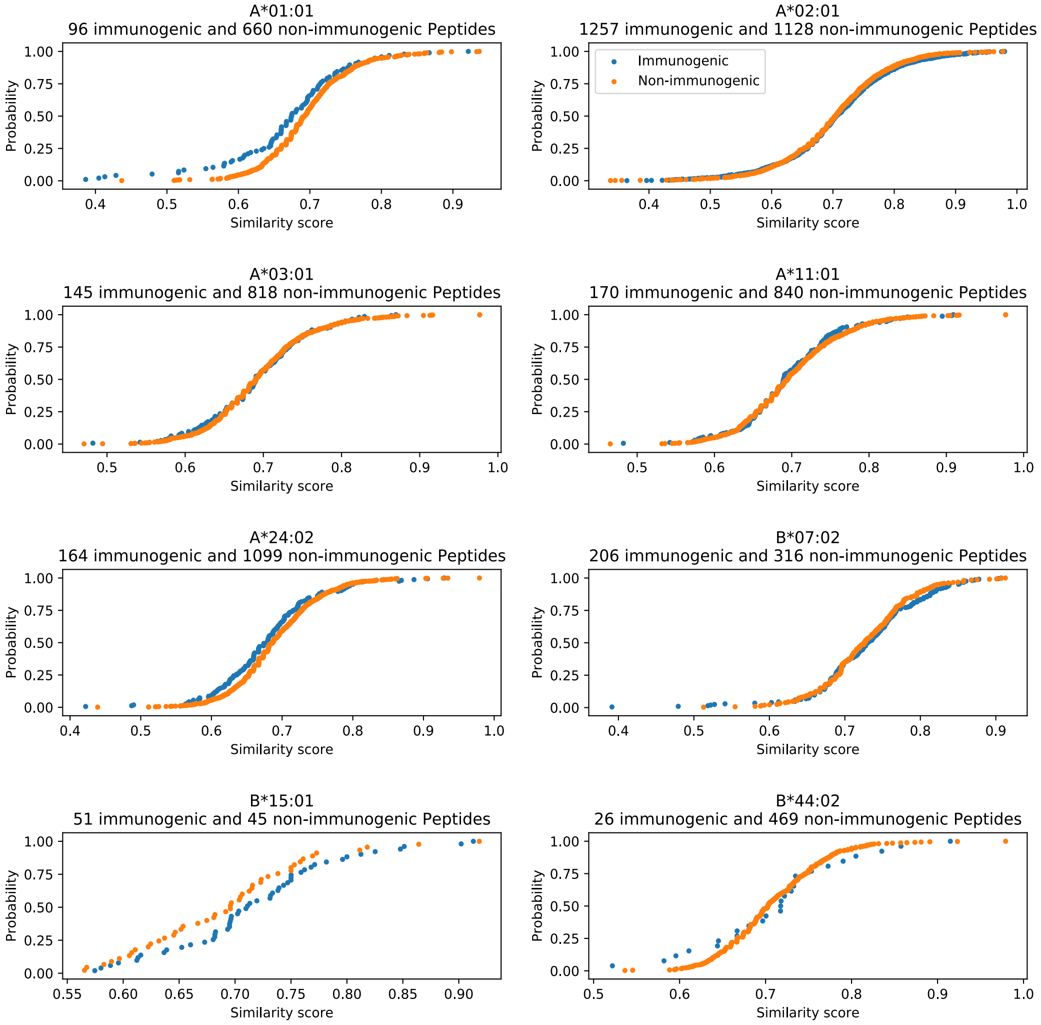


Figure 3.2: Empirical cumulative distributions of BLOSUM similarity score for the nearest neighbors in the **predicted proteome** is plotted. The corresponding allele and numbers of all relevant peptides, excluding equal peptides, is annotated above each plot.

Comparing this to the results obtained for the HLA-Ligandome as self-representation, we observe similar behaviors. However, the distributions differ less for the predicted proteome. Especially the thymus set shows similar BLOSUM similarity score distributions compared to the HLA-Ligandome, even though the overlap between both sets is only 3%. Similar trends for decreasing p-values by increasing k in the k-nearest neighbor search were observed. Thymus had mostly lower p-values than proteome and in some cases even lower than for the HLA-Ligandome. A significantly lower median BLOSUM similarity to self was achieved for HLA-A*01:01 and HLA-A*24:02 for thymus and the predicted human proteome (p-values < 0.03); furthermore, by HLA-B*15:01 for the thymus proteome.

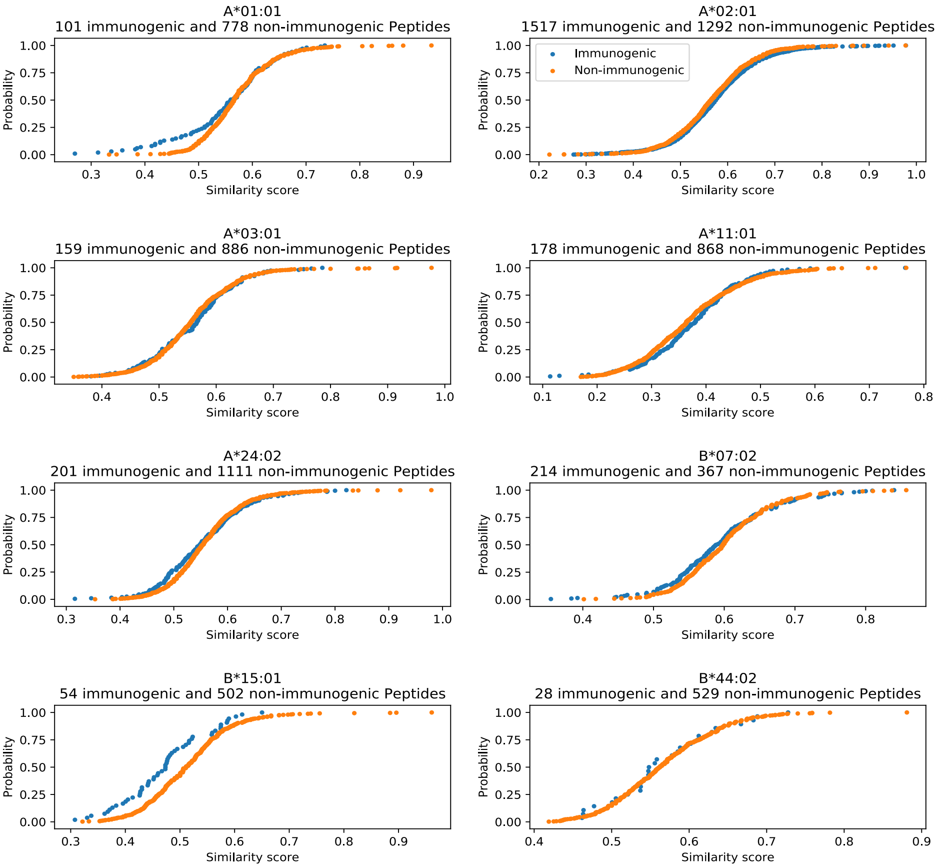


Figure 3.3: Empirical cumulative distributions of the BLOSUM similarity score for the nearest neighbor in the **predicted thymus proteome**. The corresponding allele and numbers of included peptides, excluding equal peptides, is annotated above each plot.

3.1.3 Position specific weights

As described in Subsection 1.1.3, the TCR interaction with the peptides are not uniformly distributed [CdBK12, RW02]. Therefore we tried to find position-specific weights that optimize the BLOSUM similarity score distributions against the HLA-Ligandome. As we expect immunogenic peptides to be less similar to self compared to non-immunogenic peptides, we first tried to achieve a lower median similarity for immunogenic than for non-immunogenic peptides. To solve this optimization problem, we applied a genetic algorithm [BPS$^+$16, FdBL$^+$08, Whi94].

We used five different weights $w \in \{1, 2, 3, 4, 5\}$. The algorithm was started with a population of 50 distinct individuals. Tournament selection was applied with a tournament size of three. Elitism was set to keep the five best individuals, and the mutation rate was set to five percent. Two different fitness functions were applied. First, we minimized the p-value of a Mann-Whitney rank test, testing for a lower median similarity score of immunogenic peptides.

Exactly matching peptides were excluded for fitness evaluation, as for the distributions. The algorithm converged after approximately 10 to 15 generations. We repeated the runs two times for different starting generations and accepted the best individual if all runs got similar or the same weights, otherwise, we reran the algorithm one more time and selected the best.

The weights to minimize the similarity to self for immunogenic peptides are listed in Table 3.4, from now on referred to as MIN weights. The genetic algorithm succeeded to find position-specific weights that cause distributions with a significantly lower median BLOSUM similarity to self of immunogenic peptides for most alleles, except for HLA-A*02:01, HLA-A*03:01 and HLA-A*11:01. These weights reveal that immunogenic peptides are significantly less similar to self-peptides mostly at positions 4, 5, and 8 for HLA-A and positions 1,2,4,7 and 8 for HLA-B. However, these weights represent weights that fit the underling data best for the corresponding fitness function. Therefore, this does not necessarily imply that these weights fit for all MHC binding peptides of the same allele. To assess if the obtained weights are generalizable, we performed a stratified 2-fold-cross-validation. Immunogenic and non-immunogenic datasets were randomly shuffled and split in two. We optimized weights similarly with the genetic algorithm on one half and determined the median BLOSUM similarity score to the HLA-Ligandome on the other. We made this for both sets, and the obtained averaged medians are listed in Table 1. As we optimized weights by minimizing the median BLOSUM score of immunogenic peptides, this behavior should be observed in the cross-validation results if the obtained weights are generalizable. A lower median BLOSUM score for immunogenic peptides was observed for HLA-A*01:01, HLA-A*24:02, HLA-B*07:02 and HLA-B*15:01. On the other hand, HLA-A*02:01, HLA-A*03:01, HLA-A*11:01 and HLA-B*44:02 showed in the testing sets opposite behaviors and therefore showed less generalizability. For HLA-B*44:02, this is most likely caused by the small number of immunogenic peptides [AC10].

To determine fitness unbiased from our main assumption and to evaluate the predictability, we optimized the F1-score of a simple SVM classifier through the same algorithm. All peptides were thereby represented as two-dimensional points that represent the BLOSUM similarity score for the first and second nearest neighbor. The SVM was set to use a radial basis function kernel, and the penalty parameter $C = 1$. We maximized during the genetic algorithm the mean F1-score of a 5-fold-cross-validation as described in Subsection 2.2.3 [BC00, AC10, PVG$^+$11]. The obtained weights are listed in Table 3.5, from now on referred to as SVM weights. The highest weights obtained P1, P3, P5, and P8 for HLA-A and P1, P3, P5, and P7 for HLA-B. The obtained classification boundary, together with the 5-fold-cross-validation F1-scores, are shown in Figure 3.5. Only for HLA-A*01:01 and HLA-B*15:01 the classifier clearly associated immunogenic peptides with a lower similarity to self-peptides.

Table 3.4: Position-specific weights obtained by minimizing the median similarity of immunogenic peptides compared to non-immunogenic ones. The minimized p-values for Mann-Whitney rank test (MU) and additionally for two-sample Kolmogorov-Smirnov test (KS) are attached. The last rows represent the mean weights per position for HLA-A and HLA-B.

HLA	P1	P2	P3	P4	P5	P6	P7	P8	P9	KS	MU
A*01:01	3	1	1	5	3	2	1	1	5	10^{-5}	10^{-5}
A*02:01	5	1	4	1	1	1	5	5	1	0.9	0.25
A*03:01	1	1	1	4	5	2	1	3	5	0.86	0.33
A*11:01	1	1	2	5	1	3	5	5	1	0.6	0.47
A*24:02	1	1	5	3	5	3	1	4	1	10^{-4}	10^{-4}
B*07:02	4	5	2	4	1	1	5	1	1	0.08	10^{-3}
B*15:01	4	2	5	1	2	2	1	5	1	10^{-7}	10^{-7}
B*44:02	1	5	1	4	1	5	3	5	1	0.01	0.01
HLA-A	2.2	1	2.6	3.6	3	2.2	2.6	3.6	2.6		
HLA-B	3	4	2.6	3	1.3	2.6	3	3.6	1		

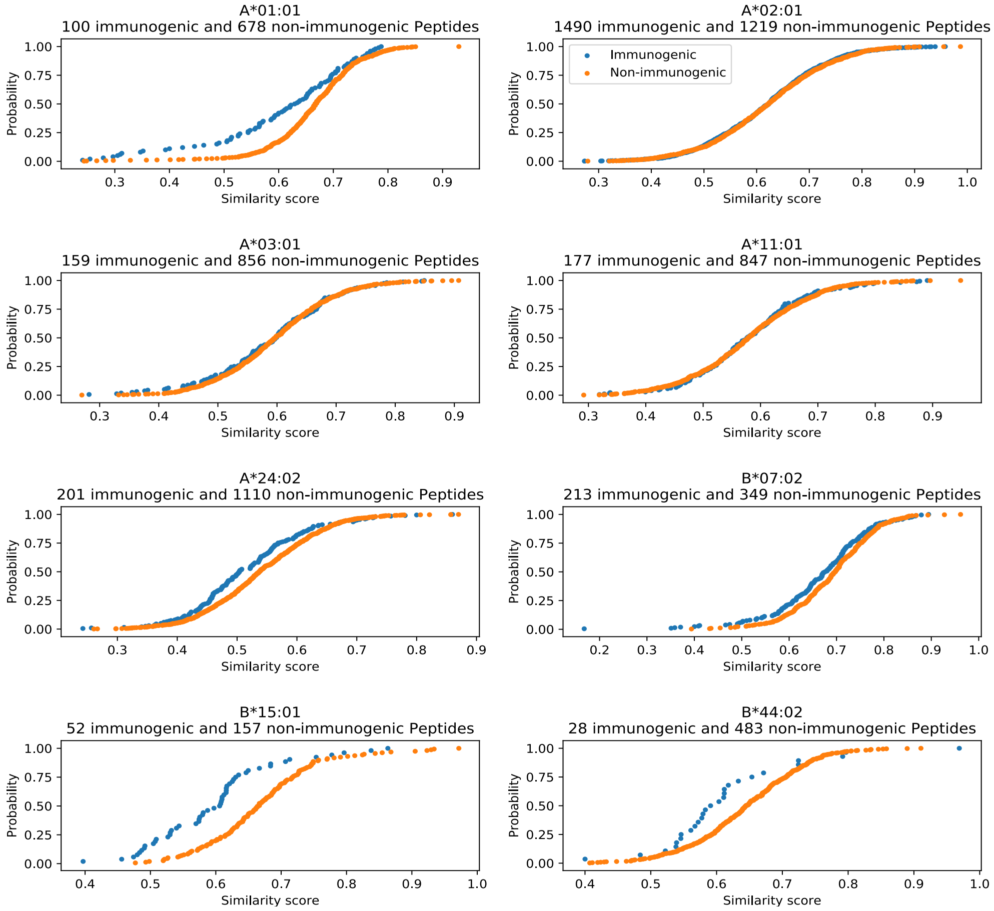


Figure 3.4: The empirical cumulative distributions of the BLOSUM similarity score, weighted by positional weights described in Table 3.4, against the HLA-Ligandome. The corresponding HLA allele and size of the datasets, excluding equal peptides, is annotated above each plot.

Table 3.5: Position specific weights obtained by maximizing cross-validation F1-score of a SVM classifier. The last rows represent the mean weights per position for HLA-A and HLA-B subtypes.

HLA	P1	P2	P3	P4	P5	P6	P7	P8	P9
A*01:01	2	1	1	1	5	4	1	1	5
A*02:01	5	1	2	1	3	1	5	4	4
A*03:01	5	3	5	3	2	3	2	3	1
A*11:01	4	4	2	1	3	4	1	3	1
A*24:02	3	2	5	4	5	2	1	5	2
B*07:02	5	1	3	2	5	5	1	5	5
B*15:01	4	2	4	3	1	1	4	1	1
B*44:02	4	2	4	2	3	2	5	2	3
HLA-A	3.8	2.2	3	2	3.6	2.8	2	3.2	2.6
HLA-B	4.3	1.6	3.6	2.3	3	2.6	3.3	2.6	3

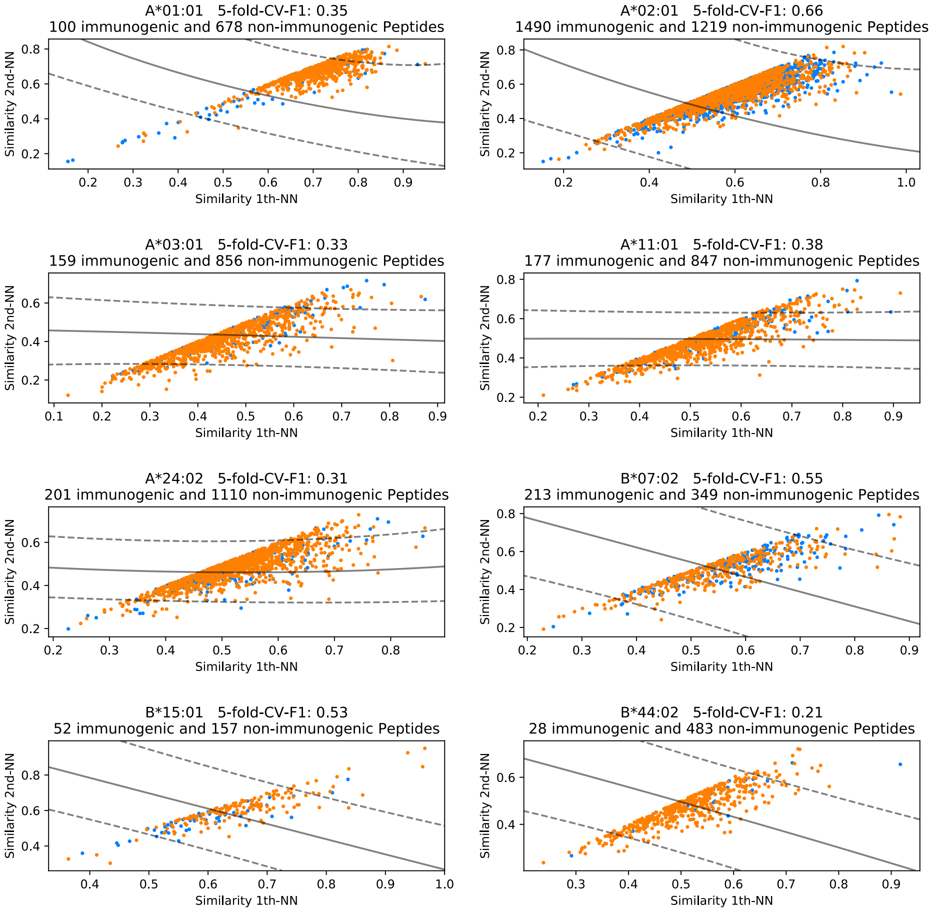


Figure 3.5: A scatter plot showing the BLOSUM similarity score of the first and second nearest neighboring self-peptide in the HLA-Ligandome, weighted as shown in Table 3.5. The SVM decision boundary is shown as a black line. Above each plot, the corresponding HLA allele, the mean 5-fold-cross-validation F1-score, and the relevant peptides, excluding equal matches, is annotated.

Additionally, we trained SVMs with the same settings for the BLOSUM similarity score to the HLA-Ligandome using a uniform weighting, the MIN weights, and the SVM weights. The corresponding F1-scores are listed in Table 3. For comparison, we also included the F1-scores of a dummy classifier that randomly classifies peptides with respect to the number of data points included. We classified the data 100 times with this classifier and annotated the mean F1-score. The SVM weights obtained, on average a 22% higher F1-score than the dummy classifier and an 8% higher F1-score than the MIN weights. The MIN weights increased the F1-scores if the genetic algorithm could find weights that result in a significantly lower median BLOSUM scores.

3.2 Feature map encoded distance to self

In the previous section, we measured the similarity using a score derived from a BLOSUM62 matrix. Thereby we consider the overall physicochemical properties of amino acids [HH92]; however, some chemical properties enhance immunogenicity more than others [CMG$^+$13]. Therefore it may be beneficial to consider only a subset of properties. We encoded peptides as described in Subsection 2.2.2. First, we revied feature maps associated with immunogenicity in previous studies and then mined for physicochemical properties in the AAIndex database ourselves. After encoding the peptides with feature maps, the distance to self was determined by solving the nearest neighbor problem using the euclidean distance for peptides with known immunogenicity to self-peptides contained in the HLA-Ligandome.

3.2.1 AAIndex encoded distance to self

In previous studies, there was no single amino acid property described in the AAIndex database associated with immunogenicity. Nevertheless, a combination may be feasible [CMG$^+$13]. Such a combination was searched to create the immunogenicity predictor POPI. They could associate 23 indices from the AAIndex with immunogenicity [TH07]. We therefore first described all peptides with these 23 indices and then computed the k-nearest neighbors for $k \in \{1, 10, 100\}$ as described in Subsection 2.2.2. The obtained distributions of the feature encoded distance score to peptides in the HLA-Ligandome are shown in Figure 3.6. Keep in mind that now a distance metric rather than a similarity is shown. Therefore we would expect for immunogenic peptides a higher distance to self-peptides than for non-immunogenic ones. Comparing to the BLOSUM similarity approach, we would expect an inverted behavior of the similarity/distance to self distributions. Exactly this behavior is observed in the plots.

To evaluate the difference in these distributions, a two-sample Kolmogorov-Smirnov test was used, and the resulting p-values are shown in Table 3.6. While these showed similar behaviors as for the BLOSUM similarity score, consistently higher p-values were obtained. In most cases, except HLA-A*01:01, HLA-B*07:02 and B*44:02, we observed a significant difference in these distributions. According to a Mann Whitney rank test only HLA-B*15:01 showed a significant trend for immunogenic peptides to be more distant to self than for non-immunogenic peptides (p-value < 0.0004).

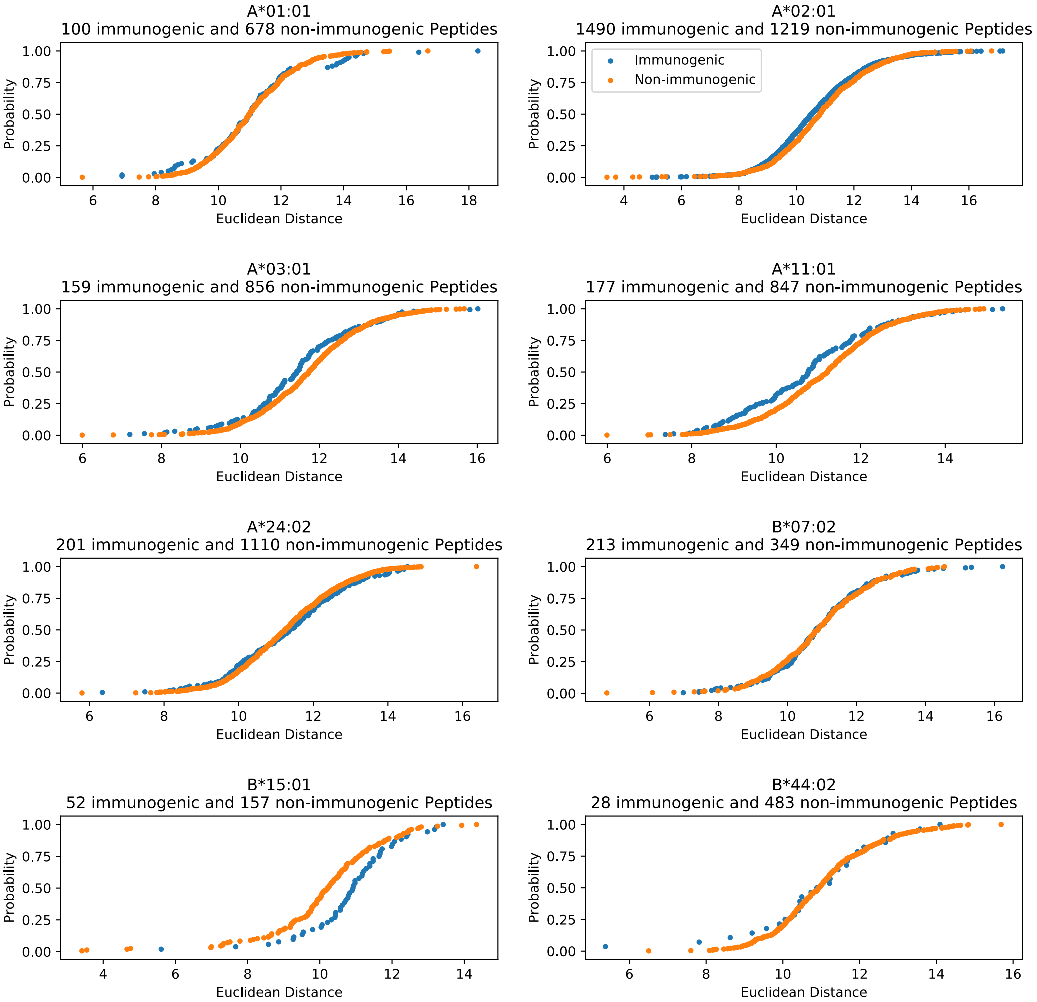


Figure 3.6: Empirical cumulative distributions of the euclidean distance from the nearest neighbor in the HLA-Ligandome. Peptide are described with the 23 AAIndices associated with immunogenicity [TH07]. The corresponding allele and number of included peptides, excluding equal peptides, is annotated above each plot.

Table 3.6: P-values of a two-sample Kolmogorov-Smirnov test for the obtained distributions using the k-nearest-neighbor mean feature map encoded distance score for all HLA alleles with $k \in \{1, 10, 100\}$. Significantly different distributions are highlighted.

HLA	KS-Test (k=1)	KS-Test (k=10)	KS-Test (k=100)
A*01:01	0.3566	0.3678	0.5946
A*02:01	**0.0002**	$\mathbf{2.62 \cdot 10^{-5}}$	$\mathbf{2.34 \cdot 10^{-5}}$
A*03:01	**0.0099**	**0.0007**	**0.0008**
A*11:01	**0.0003**	**0.0002**	**0.0007**
A*24:02	0.4626	**0.0359**	**0.0146**
B*07:02	0.6906	0.9955	0.2097
B*15:01	**0.0005**	**0.0004**	**0.0103**
B*44:02	0.9467	0.6739	0.5096

3.2.2 Physicochemical properties relevant to self-tolerance/immunogenicity

As the indices obtained in the creation of POPI [TH07] showed low differentiability, we tried to identify a small set of indices from the AAindex relevant for immunogenicity. We solved this problem similar to the positional weighting approach with a genetic algorithm. First, all AAIndex entries were z normalized and all indices that contain NA values are excluded. This lead to 553 feature mappings. We performed a redundancy reduction and only considered AAIndex entries that have a Pearson's correlation coefficient below 0.9 to each other. This lead to 331 feature mappings that were considered by the genetic algorithm.

We defined a chromosome as a set of five amino acid indices. We are starting from a population of 1000 distinct random individuals. First, we evaluated the fitness to minimize the immunogenic distance to self. This was done by minimizing the p-values of a Mann-Whitney rank test. The tournament size was set to three, elitism to ten and the mutation rate to one percent. The identified combinations are listed in Table 3.7, from now on referred to as MAX indices. The corresponding feature encoded distance score distributions to the HLA-Ligandome are shown in Figure 3.7. All combinations were able to obtain highly significant p-values for their corresponding test. All HLA alleles share no single feature map; however, there are feature maps that are shared by multiple HLA alleles, e.g., KUMS000103. In some cases, the algorithm obtained the best results by reusing a single AAIndex twice. To test their generalizability, we again made a 2-fold-cross-validation, similarly to the position weighted approach. For immunogenic peptides, this showed a higher median distance for all HLA alleles except HLA-A*02:01 and HLA-B*44:02. Therefore, the obtained indices should be generalizable [AC10]. The corresponding medians are listed in Table 2.

On a comparable basis, we again extracted a combination of five AAIndices that achieved best 5-fold-cross-validation F1-scores. All peptides were thereby represented as two-dimensional points that represent the Euclidean distance of the first and 10th nearest neighboring self-peptide in the HLA-Ligandome. The SVM was set to use a radial basis function kernel and the default penalty parameter $C = 1$ [BC00, AC10, PVG^{+}11]. This lead to the indices listed in Table 3.8, from now on referred to as SVM indices. The corresponding decision boundaries of the trained SVMs are shown in Figure 3.8. Comparing the position weighted approach, we obtained higher F1-scores and more complex decision boundaries. Immunogenic peptides were associated with a higher distance to self-peptides for HLA-A*24:02 and HLA-B*15:01.

Table 3.7: Obtained feature maps from the AAIndex, obtained by minimizing the median similarity of immunogenic peptides compared to non-immunogenic ones.

HLA	AAIndex ID's				
A*01:01	KUMS000103	RACS820113	RACS820113	RICJ880101	RICJ880107
A*02:01	CHOP780204	FAUJ880104	FAUJ880111	FUKS010101	WOLS870103
A*03:01	EISD860102	FAUJ880107	KARS160112	KARS160119	KARS160119
A*11:01	AURR980101	AURR980120	KARS160110	KARS160122	KUMS000103
A*24:02	KUMS000103	OOBM770104	QIAN880105	QIAN880123	QIAN880123
B*07:02	BUNA790103	FASG760105	KARS160119	QIAN880123	QIAN880123
B*15:01	BEGF750103	GEOR030107	NAKH900104	PALJ810114	RICJ880117
B*44:02	BROC820101	JOND920102	LIFS790102	NOZY710101	SUEM840102

Table 3.8: Feature maps obtained from the AAIndex by maximizing cross-validation F1-score of a SVM classifier.

HLA	AAIndex ID's				
A*01:01	FAUJ880111	KLEP840101	LEVM780102	QIAN880138	ROBB790101
A*02:01	BULH740102	OOBM850103	SWER830101	TANS770108	VASM830103
A*03:01	ANDN920101	FINA910101	LAWE840101	RICJ880117	SNEP660103
A*11:01	FINA910101	MEEJ810101	PONP800105	SNEP660103	VINM940102
A*24:02	CIDH920101	GEOR030108	KUMS000103	QIAN880123	ROBB760111
B*07:02	CHAM820102	OOBM850103	OOBM850105	WILM950104	WILM950104
B*15:01	BROC820101	GEIM800102	GEIM800102	QIAN880116	WIMW960101
B*44:02	FUKS010111	GEOR030108	ISOY800102	RICJ880106	SUEM840101

Additionally, we trained SVMs with all peptides encoded by the MAX indices and SVM indices. The F1-scores for both cases are listed in Table 4. The SVM optimized indices performed 8% better than the MAX indices. The same improvement was observed for the SVM weights in the position-specific optimization approach. It performed 30% better than the dummy classifier.

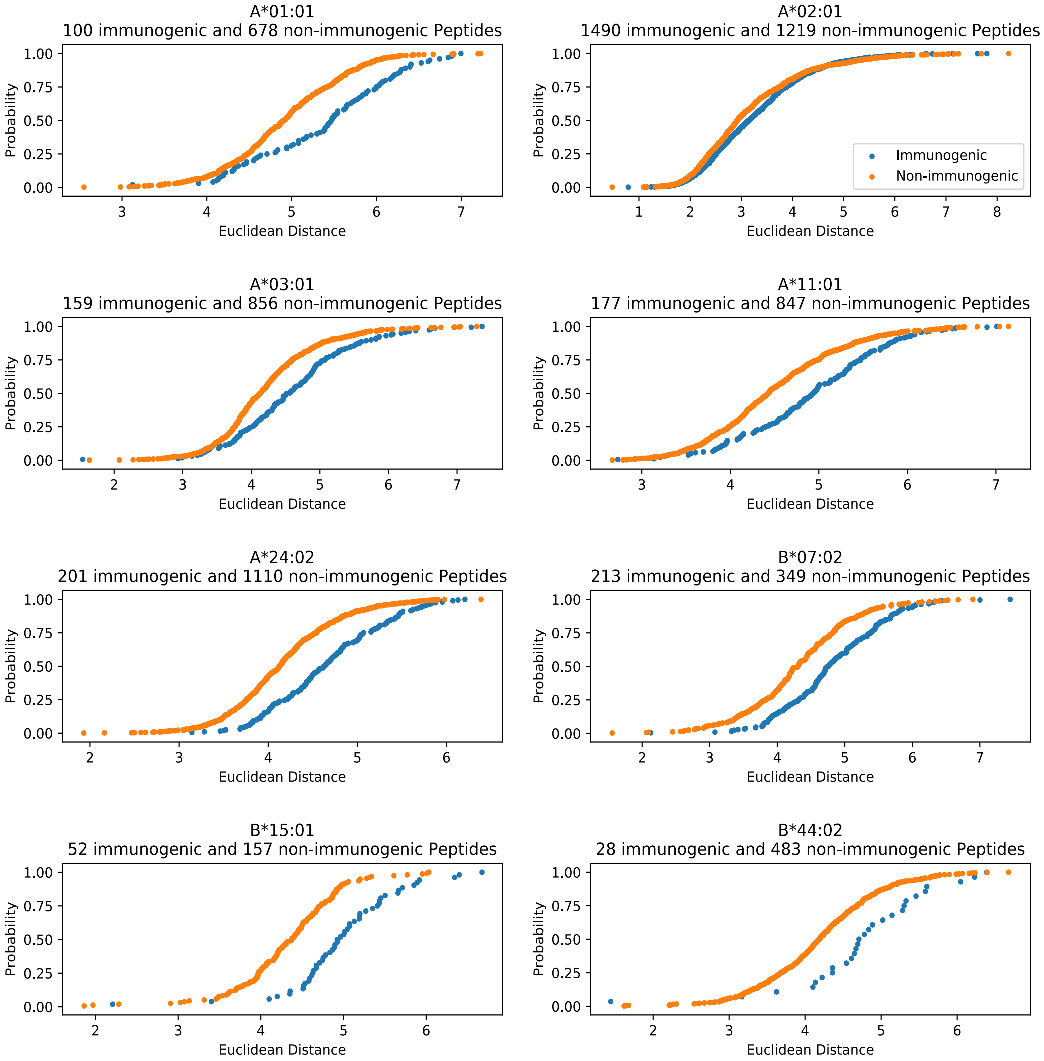


Figure 3.7: Empirical cumulative distributions of the according to Table 3.7 encoded peptides nearest neighboring distance to a self-peptide in the HLA-Ligandome. The corresponding allele and numbers of relevant peptides, excluding equal peptides, is annotated above each plot.

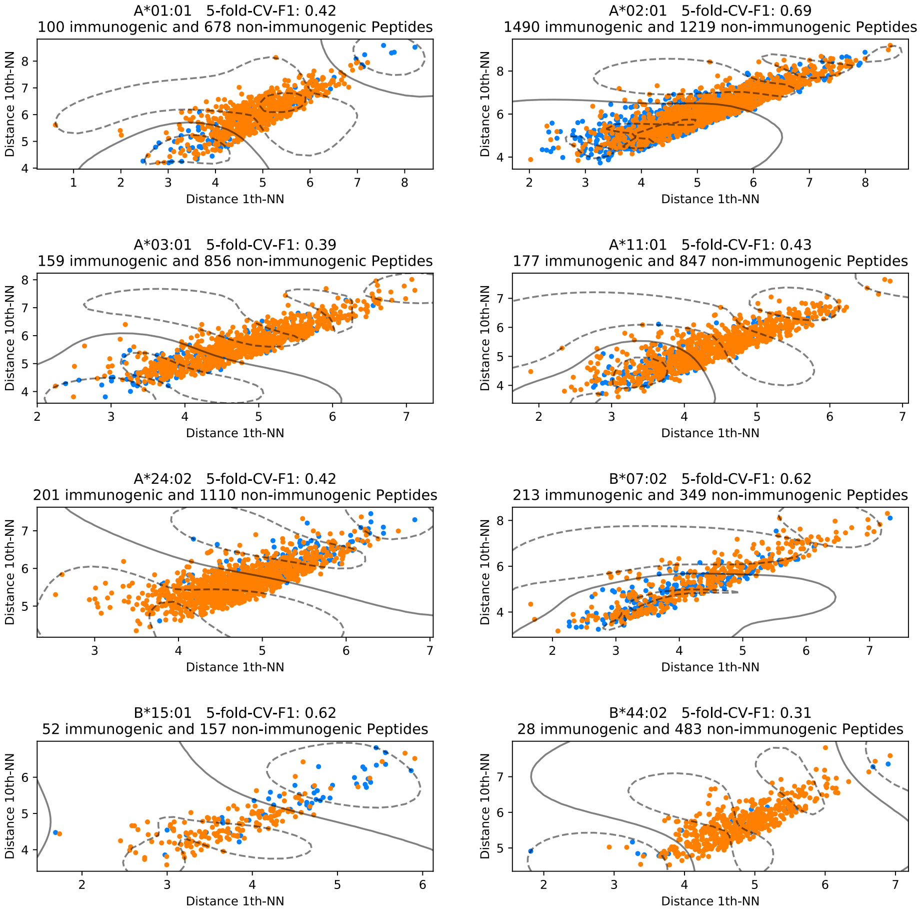


Figure 3.8: A scatter plot showing the Euclidean distances, for peptides encoded by AAIndices shown in Table 3.8, of the first and 10th nearest neighboring self-peptide in the HLA-Ligandome. The corresponding decision boundary of the SVM is shown as black line. Above each plot, the corresponding HLA allele, the mean 5-fold-cross-validation F1-score and the included number of peptides, excluding equal matches, is annotated.

3.3 Methods Benchmark

We benchmarked our methods empirically. All processing was done on a personal computer with the CPU AMD Ryzen 5 1600 (6-core, 12-threads). All methods were implemented to support multiprocessing; however, for this benchmark, we only used one thread.

We compared the trie based method, a naive approach, and the LSH as implemented in "pepdist". Additionally, a KDTree implementation from the Python module scipy was used [JOP+]. We randomized the search space to contain 10^4, 10^5, and 10^6 peptides. All methods were performed on 100 random query peptides. For LSH and KDTree, these were encoded by five random AAIndex indices: SUEM840101, GEOR030104, RADA880102, CHAM810101, and LEVM780106. For LSH the parameters were set to $L = 3$, $k = 5$ and $w = 5$. The obtained computation times are listed in Table 3.9. We can observe that as expected the naive implementation increase in $\mathcal{O}(n)$. The trie-based branch and bound algorithm is theoretically also in $\mathcal{O}(n)$; however, the computation times show that bounding can decrease this strongly, especially for a big search space, as then it is likely to find a good bound. The KDTree showed a better performance and the LSH an even better. The increase of computation time for LSH is caused by the static parameterization and therefore leading to a increase in the bin sizes. A variable parameterization can keep computation times constant.

To demonstrate the flexibility of LSH through parameterization, we measured time and mean squared error for different L and k. The bin width w was set to five. This computation was done for 100 random query peptides and 10,000 random search peptides. They were encoded with the same AAIndices as previously. The performance is visualized in Figure 3.9. Thereby mean squared errors close to zero ($< 10^{-31}$) can be observed. The mean squared error and time are in a negative exponential relationship with the parameter L. Therefore the parameter should not be chosen arbitrary high or small. For k a strong increase of the mean squared error without major time improvement is detected after $k = 5$, suggesting it as optimal value for the corresponding data.

Table 3.9: Computation times of several nearest neighbor algorithm. A naive linear algorithm, the implemented branch and bound algorithm (Trie), a k-d-tree algorithm and the implemented locality sensitive hashing (LSH).

	Naive	Trie	KDTree	LSH	
Sample size				Time	Error
$N = 10^4$	0.09 s	0.04 s	0.014 s	0.0003 s	0.04
$N = 10^5$	0.9 s	0.2 s	0.034 s	0.0008 s	0.01
$N = 10^6$	8.9 s	0.5 s	0.1 s	0.008 s	0.004

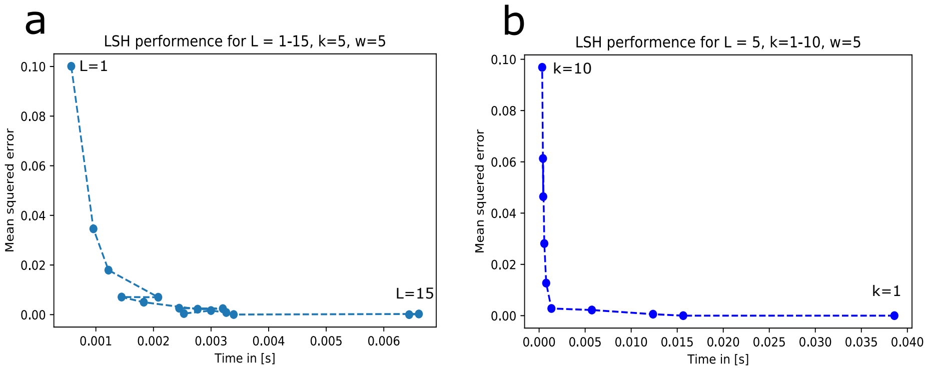


Figure 3.9: Mean query time and mean squared error for 100 random query peptides for different LSH paremeterization. First for different L **(a)**, then for different k **(b)**.

Chapter 4

Discussion and Outlook

The potency of peptide-based vaccination depends on peptide immunogenicity. An accurate prediction of peptide immunogenicity can therefore drastically reduce experimental efforts in vaccine design. Due to the complexity of the immune system, assessing the immunogenicity is a hard task. Therefore, self-tolerance is especially interesting as it models in vivo systemic effects as well as TCR interaction indirectly. The TCR has to differentiate self from non-self to maintain self-tolerance. A peptide that is recognized as non-self and induces an immune response must, therefore, differ from self-peptides. [BPS$^+$16, FdBL$^+$08]. This thesis investigates if this can be observed in a sequence-based model of self-tolerance and if it can be used to predict immunogenicity.

In this thesis, we evaluated the sequence-based model of self-tolerance using a BLOSUM62-based similarity score [HH92]. We compute the similarity to self as the highest pairwise similarity between a query peptide with known immunogenicity and a population of self-peptides in a reference set as defined in Chapter 2. First, we used an experimentally validated set of self-peptides, the HLA-Ligandome as reference. For the investigated HLA alleles the immunogenic peptides were only marginally present in the HLA-Ligandome. These peptides can be classified as false-positives, caused, for example by the soft immunogenicity classification, as peptides with contradictory T cell assay results were classified as immunogenic if there was at least one positive assay [OY18]. A much higher number of equal matching peptides was detected for non-immunogenic peptides. However, for most HLA alleles except HLA-B*15:01, there are more non-equal peptides, which is consistent with the fact that only a small overlap of nonameric peptides between the human and bacterial/viral proteomes exist [CdBK12]. As peptides from human origin were not excluded; these may cause the larger overlaps. However, peptides with human origin are only marginally present in the non-immunogenic set, which origin predominantly from the Vaccinia virus. We come to the apparent conclusion that peptides equal to a self-peptide, which are genuinely presented on an MHC molecule, can always be classified as non-immunogenic. We decided, therefore, to exclude such peptides from the analysis. For non-equal peptides, we could show that for most HLA alleles, there is a significant difference in the BLOSUM similarity score distribution between immunogenic and non-immunogenic peptides. We detected a trend for a more significant difference when the BLOSUM similarity score is evaluated for k-nearest-neighbors instead of a single one. However, in some cases, particularly for $k = 100$, it increases the p-value, which indicates that there is an allele-specific optimal value for k. Nevertheless, this revealed unexpected results. As described in Chapter 1, we expect a lower similarity to self-peptides for immunogenic peptides; however, we only detected a significantly lower median BLOSUM similarity score of immunogenic peptides for HLA-B*15:01 and HLA-A*24:02. For HLA-A*01:01, an insignificant trend to be less similar to self was observed, and for the other alleles, there is a trend to be more or equally similar to self-peptides.

We also determined the BLOSUM similarity to the HLA-Ligandome for length invariant peptides, by computing the BLOSUM similarity score for the longest highest-scoring consecutive substring as defined in Subsection 2.2.1. While this method may not be optimal for the comparison, as the primary mechanism of extending size is central bulging [SPHB00], we detected interesting trends. Computing similarities between length invariant peptides with these methods will score an exactly matching subsequence with a BLOSUM score of one. The slight increase in the frequency of peptides scored with one can be explained by the overall increase in included peptides and therefore by more equal matching peptides rather than by equally matching subsequences. As the main mechanism of extending size is central bulging with anchor residues at position two and the C terminal end, we would expect such a behavior, because comparing subsequences will displace anchor position. If protrusion would be the main mechanism of extending size, we should observe more matching subsequences [SPHB00]. The BLOSUM similarity score distributions to self-peptides in the HLA-Ligandome are similar to them with restriction to nonameric peptides. In the sight of self-tolerance a restriction to nonameric peptides is only problematic if a nonameric peptide is most similar to a self-peptide with a different length, as then, its nearest neighbor is excluded by this restriction. If this is the case, we should observe score distributions that differ; however, there are only slight differences that can be explained with the unmodeled central bulging. Longer peptides bulge out more than smaller ones, which creates different pMHC interfaces for the TCR and questions comparability of such peptides. Additionally, the TCR interacting residues change, making a comparison even harder [RW02, GTW99, SPHB00]. For the sake of simplicity, we continued the analysis with restriction to nonameric peptides.

In order to investigate the observed behavior, we reviewed several possible explanations. Firstly we validated the influence of MHC binding properties. All peptides are predicted or experimentally shown to bind to their labeled MHC molecule. This should neglect all differences that are caused by MHC binding motifs, as all compared peptides share the same motifs. However, the immunogenic set showed a higher predicted binding affinity score than non-immunogenic peptides. This trend is consistent with the fact that immunogenicity correlates with binding affinity and stability [HRR$^+$12, SVR$^+$94]. Nevertheless, this could partially explain higher similarities to the HLA-Ligandome, as the contained peptides have a high binding affinity as well. We examined this by equalizing predicted binding affinity score distributions between immunogenic and non-immunogenic peptides and detected only a small influence on the BLOSUM-based similarity score to self-peptides in the HLA-Ligandome. Therefore, we conclude that differences in MHC binding affinity do not cause a big difference in the BLOSUM similarity to self distributions.

The unexpectedly higher similarity of immunogenic peptides can possibly be explained by the fact that the HLA-Ligandome does not represent the complete immunologically relevant self proteome. It contains only several thousand peptides; thereby, the true most similar self-peptide can be missing. Additionally, the HLA-Ligandome contains no peptides that were extracted in thymic samples and therefore, only models peripheral tolerance explicitly. Hence we investigated alternative representations of the immunological relevant human proteome, the predicted proteome, and the predicted thymus proteome as defined in Section 2.2.2. The predicted proteome represents the whole human proteome that is potentially visible to a TCR and therefore models all peptides that can induce mechanisms of central or peripheral tolerance. In contrast to this, the thymus set contains peptides from proteins that are at least marginally expressed in the thymus and therefore explicitly models central tolerance. However, both sets do not account proteasomal processing and TAP transport, account for this through suitable predictors may result in better representations. All mechanisms involved in MHC presentation are modeled in the HLA-Ligandome, as these peptides were observed to be presented by living cells. Further, the predicted proteome does not account for proteins that are only marginally expressed or only expressed in immunologically privileged regions and, therefore, are not accessible for the immune system and not involved in self-tolerance [MWSS18].

We computed the BLOSUM similarity score for all three self-representations. More exactly matching immunogenic and non-immunogenic peptides are present in the proteome. Especially for HLA-A*02:01, more immunogenic than non-immunogenic peptides are contained. This is probably caused by false-positive MHC binders, as the creation method ignores processing, transport as well as precursor protein abundance and accessibility. Thereby the set is expected to contain many self-peptides that can bind to MHC but usually are not presented by a cell in vivo and are not accessible for the immune system. On the other hand, we observed nearly no investigated peptides that are equal to a self-peptide in the predicted thymus set. This is interesting as several matching peptides were observed to be present in the HLA-Ligandome or predicted proteome, which indicates that the matching self-peptides are only expressed in the peripheral tissues. For non-immunogenic peptides, this suggests that the non-immunogenic nature of these peptides is not induced by central tolerance through equal matching self-antigens in the thymus, rather that it is subject to either peripheral tolerance or that tolerance is induced through degenerated T cell recognition [FdBL$^+$08, CdBK12]. All three BLOSUM score distributions are similar. The predicted proteome performed worst, for most HLA alleles the BLOSUM similarity score distribution did not differ significantly. The thymus dataset, as well as the predicted proteome, do not model peptide processing; however, the HLA-Ligandome does. Further, the expression in relevant tissues is only considered by the thymus proteome and HLA-Ligandome. This implies that less different BLOSUM scores between immunogenic and non-immunogenic peptides in the predicted proteome are prematurely caused by not considering the peptide abundance in immunological relevant tissues, rather than not considering MHC ligand processing. This suggests that a big chunk of MHC presentable peptides in the human proteome is less relevant for self-tolerance, as they are presented in immunologically privileged tissues. We concluded from the similar distributions that HLA-Ligandome is representative for the self proteome. For HLA-A*01:01 the predicted (thymus) proteome showed a significantly lower median BLOSUM similarity to self, which was only insignificant for the HLA-Ligandome. This may be caused by the fact that it is underrepresented in the HLA-Ligandome and contained the fewest amount of self-peptides. Especially the similar behavior of ligandome and predicted thymus proteome is interesting, as the sets overlap only with 3% but still result in similar BLOSUM similarity to self distributions. This indicates that while the expressed thymus proteins can differ from those in peripheral tissues, the MHC presented peptides share common properties to distinguish self from non-self. It is well known that thymus epithelial cells can express peripheral as well as tissue-specific antigens [DSKK01]. The small overlap between thymus proteome and HLA-Ligandome does reflect this behavior only partially and suggests a more indirect induction of self-tolerance by presenting peptides with similar properties. This is especially present for HLA-B*15:01, as there is a big chunk of non-immunogenic peptides present in the HLA-Ligandome or predicted human proteome; however, not in the predicted thymus. This suggests that corresponding equally matching peptides are only presented in peripheral tissues and not in the thymus. Nevertheless, these show a significantly higher similarity to self, compared to that of immunogenic ones and implies that self-tolerance is induced thought degenerated T cell recognition of peptides with a high similarity to each other [FdBL$^+$08]. The obtained results are also consistent with the results of a self-tolerance predictor, which obtained the best performance on a similar thymus set, but no performance increase on similarly predicted ligands of the human proteome [TFZ$^+$11]. Comparing the obtained BLOSUM similarity score distributions to that of MHC class II epitopes for a similar scoring function obtained by Berescani et al. [BPS$^+$16], we do not observe a straight lower similarity for immunogenic peptides. This can be caused by the fact that TCR interactions are less uniformly distributed and the peptide conformation is less conserved for MHC class I peptides than for MHC class II ones [GTW99, RW02].

The BLOSUM-based similarity score itself can bias the unexpected trend for higher similarities to self of immunogenic peptides. Applying this score finds globally similar peptides, but the TCR interaction is dominated only by some residues in the central region between position 4 and 8 [RW02, CdBK12]. As the presentability of the peptide is predicted or experimentally validated for immunogenic and non-immunogenic peptides, the BLOSUM-based similarity score should account less for positions that are associated with MHC binding, instead of more for positions that are associated with TCR interaction. We tried to include this in the BLOSUM-based score by including position-specific weights. We chose to weight each position with a number between one and five. Excluding entire positions can lead to a higher overlap of immunogenic and self-peptides, which is not desirable. Further, all peptide positions can influence immunogenicity even if they have less TCR contact but determine conformational similarities or factors that influence immunogenicity like MHC binding and stability [SVR$^+$94, HRR$^+$12, TEP$^+$05]. A range from one to five for the weights allows high weights for positions that have a presumably strong influence on immunogenicity, but also consider global similarity. Allowing a bigger range of weights may represent position specificity better. However, we identify these weights with a genetic algorithm; thereby, the increased range would strongly increase the search space size, leading to more computational demand for optimization. For all alleles, we computed the position-specific weights that minimize the median immunogenic BLOSUM similarity to self, comparing to the non-immunogenic similarity. The obtained position-specific weights (MIN weights) could influence BLOSUM-based similarity in an expected way and also reveal positional trends of the data. We obtained mostly different weighting for the alleles, which suggests different position specificity for different MHC alleles. This is consistent with the fact that peptide MHC conforamtion can vary even for MHC subtypes that only differ in one amino acid [TEP$^+$05]. Nevertheless, we observed averaged trends for HLA-A and HLA-B. For HLA-A the residue positions P4, P5, and P8 obtained on average the highest weights, which is consistent with the fact that central positions P4-P8 have most TCR interaction [CdBK12, RW02]. The P2 residue yielded the smallest weight, presumably because it is an anchor residue and shared between MHC binding peptides. Nevertheless, P9 received a higher weight and is an anchor residue, too. The P1 position received an intermediate weight which is consistent with the fact that the P1 residue is often not associated with immunogenicity or TCR contact [CMG$^+$13, CdBK12]. For HLA-B, we observed different trends, and P1, P2, P4, P7, and P8 obtained high weights. The high weightings of P4, P7, and P8 are consistent with HLA-A and TCR contact profiles [CdBK12]. However, interestingly the P5 residue is weighted much lower than in HLA-A, which suggests that it is less critical for HLA-B. The high weight of P2 for HLA-B suggest a strong influence of MHC associated properties to immunogenicity or at least a trend in the corresponding data. We tested if thereby obtained position-specific weights are generalizable with a stratified 2-fold-cross validation. The results indicate that weights for HLA-A*01:01, HLA-A*24:02, HLA-B*07:02, HLA-B*15:01 are generalizable. More folds for the validation would be more meaningful [AC10]; however, this was computational too expensive.

Unbiased from the assumption that BLOSUM similarity to self should be smaller for epitopes, we determined position-specific weights that maximized the 5-fold-cross-validation F1-score in a simple SVM classifier as described in Subsection 2.2.3. A query peptide is represented as a two-dimensional point, represented thought the position weighted BLOSUM similarity score of the first and second nearest neighbor in the HLA-Ligandome. This approach will not force the immunogenic peptides to have a lower similarity than the non-immunogenic ones; it will increase the discriminability of immunogenic and non-immunogenic peptides. Therefore the obtained position-specific weights should reveal positions that differ most in the BLOSUM similarity to the two nearest neighboring self-peptides between immunogenic and non-immunogenic peptides. We obtained partially different weights (SVM weights) compared to the MIN weights. For HLA-A the residue positions P1, P3, P5, and P7 obtained on average the highest weighting, for HLA-B the positions P1, P3, P5, P7, and P9. The obtained decision boundaries of the SVM classified peptides with low BLOSUM similarity score to self as immunogenic for HLA-A*01:01 and HLA-B*15:01. However, the F1-scores were increased on average by 10% using the SVM weights compared to a uniform weighting and increased on average by 2% using MIN weights. This at least proofs that the usage of position-specific weights can increase predictive performance. The MIN weights mostly increased F1-scores only if the algorithm was able to achieve a significantly lower median BLOSUM similarity to self. On average, the obtained classifiers using SVM weights for BLOSUM similarity score computation reached for the corresponding data an F1-score of 41%. A dummy classifier, which randomly classifies peptides with respect to the sample sizes obtained only an F1-score of 19%.

All in all, these results suggest that positional weights can model positional dependencies in an expected way and can increase the predictability. However, TCR recognition is highly complex. Determine the similarity of peptides with position-specific weights suggest that both peptides interact with similar positions to the TCR. The TCR-pMHC crystal structures show for a limited amount of peptides that contacts are position-dependent on average [CdBK12, RW02]. However, they do not imply that all peptides have to interact with these positions, which is modeled with static positional weights, as the peptide similarity is more dependent on high weighted positions. Therefore these weights should work for HLA alleles were the position specificity of peptide residues is more conserved better than for them were it is highly variable. Additionally, the degeneracy of TCRs is problematic. Some studies revealed that amino acid similarity accounts for T cell cross-reactivity, and similar amino acid substitutions do not perturb TCR specificity [FdBL$^+$08]. This principle is fundamental to associate immunogenicity with sequence similarity to self. However, in other cases peptides with minimal sequence similarity induced a T cell response for the same T cell clones [WAC$^+$07], which makes immunogenicity in this case, not quantifiable thought peptide similarity. These and several other factors explain why, in some cases, the assessment of immunogenicity thought similarity to self does not work accurately. Some peptides that are highly similar to self-peptides can be recognized thought degenerated T cell activation, leading to a highly similar peptide that is immunogenic. The results suggest that for most peptides of an HLA allele, the positional weights are valid, but for some, they may not reflect conformational reality leading to over or underscored similarities. Some alleles, e.g. HLA-A*01:01, work better, which indicates a more conserved position specificity. However, with today's limited knowledge of TCR specificity, it is impossible to account for this accurately, especially only from sequence information. Further research is needed to determine the most relevant features of TCR recognition to create more accurate similarity scorings.

We reviewed AAIndex indices that were previously associated with immunogenicity [TH07]. As described in Subsection 2.2.2, we encoded all peptides with these feature maps and computed the feature map distance score to the HLA-Ligandome. The obtained score distributions showed similar properties as the BLOSUM-based similarity to self. However, they only showed a significant trend of immunogenic peptides to be more distant to the self proteome for HLA-B*15:01 and HLA-A*24:02. Because the score distributions to the HLA-Ligandome were similar to that using the BLOSUM62-based scoring, the 23 AAIndices may describe a peptide similarly to the BLOSUM62 matrix [HH92]. As POPI did not separate immunogenic and non-immunogenic peptides by alleles, this may also be biased by MHC binding motifs [TH07]. Therefore, we mined for feature maps in the AAIndex database and found out that a combination of five feature maps can shape the feature mapped distance to the HLA-Ligandome. We first searched for AAIndices that would cause a higher distance to self for immunogenic peptides (MAX indices). Therefore we again used a genetic algorithm as described in Subsection 2.2.3. For all alleles, we could find a combination that causes a significantly higher distance to self-peptides in the HLA-Ligandome. In some cases the genetic algorithm reused some indices twice to obtain the best solution, suggesting that may a smaller combination is sufficient. We could show through a 2-fold-cross-validation that thereby obtained feature maps are generalizable, except for HLA-B*44:02. Using more folds would be more meaningful but also computationally more demanding [AC10]. However, the obtained indices do mostly not describe simple chemical properties, instead, e.g., describing frequencies in protein secondary structures (KUMS000103, QIAN880123) [KK00]. All alleles shared no features; nevertheless, some are shared by multiple alleles. We again obtained indices by maximizing the mean 5-fold-cross-validation F1-scores of an SVM classifier (SVM indices). Each peptide is represented as a two-dimensional point describing the distance of the first and 10th nearest neighbor, for a given encoding with five AAIndices. Some chemical properties were identified, e.g., positive charge (FAUJ880111) for HLA-A*01:01 [KK00]. However, most indices again described more complex properties. The SVM indices encoding lead for HLA-A*24:01 and HLA-B*15:01 to more distant immunogenic peptides. We again trained SVMs with the MAX and SVM indices and computed the F1-scores for all peptides. The classifier using SVM indices could achieve an F1-score of 49%, which is 8% better than the classifier using MAX indices and 30% better than the dummy classifier. This shows that describing peptides by amino acid feature maps can increase the predictive performance of a classifier. Higher F1-scores as for the position weighted BLOSUM similarity score was obtained. This suggests that TCR recognition can be modeled better with feature maps. A combination of both could presumably obtain an even better performance.

With “pepdist” we created a framework for the fast computation of k-nearest-neighbors using a BLOSUM-based similarity or a feature encoded distance metric. This allows for the optimization of position-specific weights or the extraction of chemical properties from the AAIndex. The benchmarks showed that the translation into numerical space could greatly improve performance. With locality sensitive hashing, we created a tool to find approximate nearest neighbors for arbitrary big datasets fastly. This showed that with proper parameterization, we could obtain an approximate nearest neighbor search with mean squared errors close to zero. For a static database, the proper parameterization can be trained through sample queries, leading to an accurate approximate nearest neighbor search that can determine the distance to self for several thousand query peptides in less than a second.

4.1 Conclusion

Coming back to the main questions stated in Subsection 1.2. We could show that HLA-Ligandome is representative for the immunological relevant proteome, even though it only explicitly models peripheral tolerance. However, the thymus proteome showed for some alleles a more differentiable BLOSUM similarity score to self. As the HLA-Ligandome is still work in progress, thymus samples may be included at a later time. Additionally, the peptides are mapped to HLA-alleles based on the highest predicted binding score. An inclusion of promiscuous binders would lead to a more complete allele association of peptides.

Next, we could show that the similarity/distance to self-peptides differs for most alleles. However, when using the simple BLOSUM similarity unweighted, only a minority of the investigated alleles showed a significantly lower median immunogenic similarity to self, compared to the non-immunogenic one. Nevertheless, this can be changed by using position-specific weights or by mining physicochemical properties from the AAIndex database. Thereby we could show that a majority of the HLA alleles show a lower immunogenic similarity using position-specific weights or a combination of five feature maps from the AAIndex. We could show that this can increase the predictive performance of an SVM classifier in two dimensions. We expect that using a more advanced classifier will increase the performance, but designing the best classification model would exceed the time frame of this thesis. A combination of more than five features may also improve the performance but also increase the number of possible combinations. This increases computation time, as the genetic should then increase its starting population size. Equally, the range of possible weights can be increased, which would also increase the number of combinations and leads to increased computational costs. Further algorithmic optimizations or more computational resources would be needed, to allow a more in-depth analysis.

Nevertheless, the discriminability of immunogenic and non-immunogenic peptides, based on the applied methods, is far from perfect. While for some HLA alleles it is good, there is hardly any difference in others. Even with further optimization on scoring, a purely sequence-based self-tolerance predictor would most likely not achieve the accuracy needed for therapeutic application. However, a consensus predictor combining multiple factors may can.

With “pepdist” we created a Python framework which enables to extract positional weights or feature maps in adequate time. We could implement several methods for fast nearest neighbor computation. Nonetheless, there is still room for improvements in the implementation. However, the obtained computational costs were sufficient for the purpose of this thesis, implementing the most efficient way of nearest neighbor search in Python would go beyond the scope of this thesis.

4.2 Outlook

We showed that position-specific weights can enhance the predictive performance of a classifier and that immunogenic peptides can mostly be associated with a lower similarity to self-peptides when relevant residue positions or physicochemical properties are considered. Additionally, maybe scoring matrices other than BLOSUM62 describe relevant properties better. These can be determined similarly with the implemented genetic algorithm. Further algorithmic optimizations or more computational resources can allow a more in-depth analysis of position-specific weights or relevant feature maps. On the other hand, a positional weighting can be introduced in the approach using feature maps to combine the position-specific TCR interactions and relevant physicochemical properties.

All in all, this can lead to a more powerful self-tolerance based immunogenicity predictor. With the recently released tool for assessing immunogenicity based on TCR contact potentials [OY18], and the use of other properties as sequence features and MHC binding affinity/stability, a next-generation consensus predictor can be created that may achieve the necessary accuracy for therapeutic usage.

Chapter 5

Further Tables and Figures

.1 HLA-Ligandome BLOSUM score for k=10

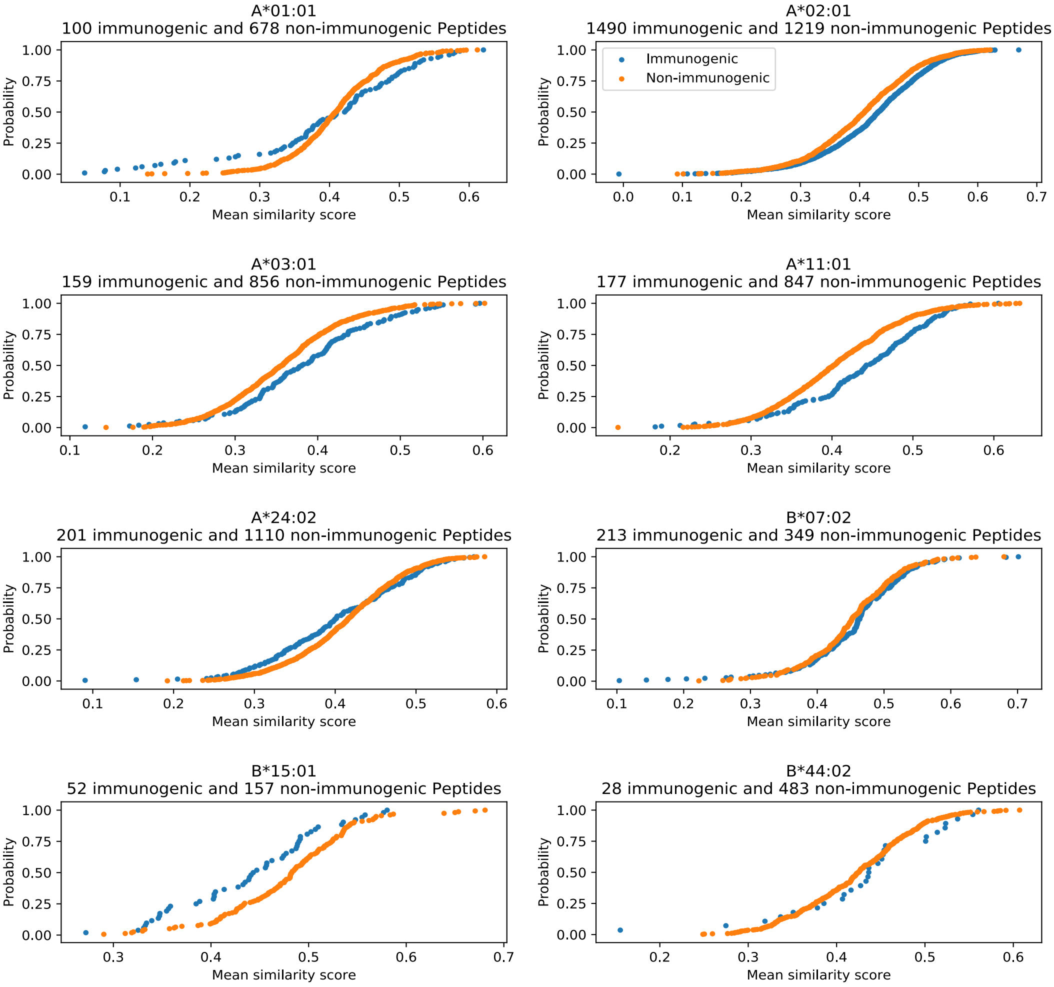


Figure 1: Empirical cumulative distributions of the mean BLOSUM similarity score for the 10 nearest neighbors in the HLA-Ligandome. The corresponding size and HLA-allele of the datasets is annotaed above each plot. Exact matches were excluded from the distribution.

.2 HLA-Ligandome BLOSUM score for k=100

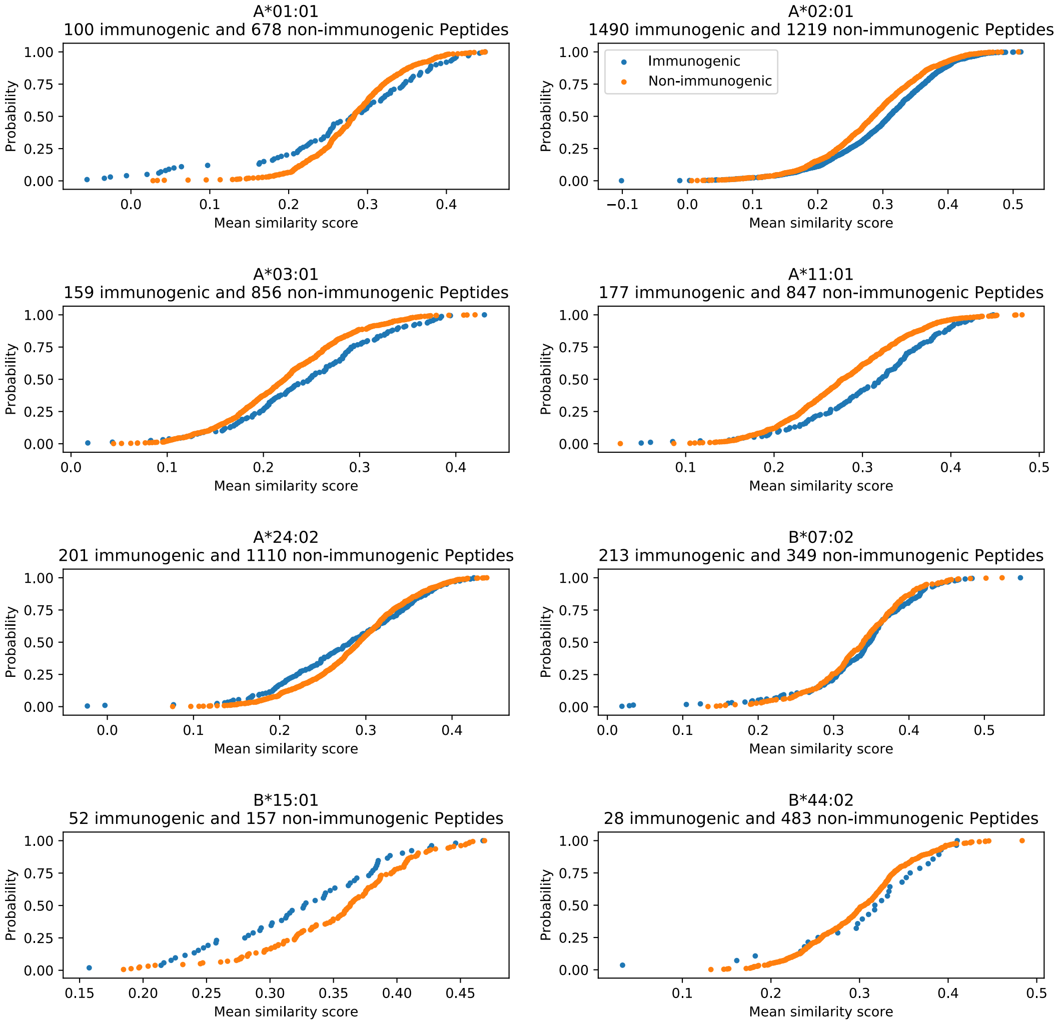


Figure 2: Empirical cumulative distributions of the mean BLOSUM similarity score for the 100 nearest neighbors in the HLA-Ligandome. The corresponding size and HLA-allele of the datasets is annotaed above each plot. Exact matches were excluded from the distribution.

.3 HLA-Ligandome BLOSUM score of length-invariant peptides

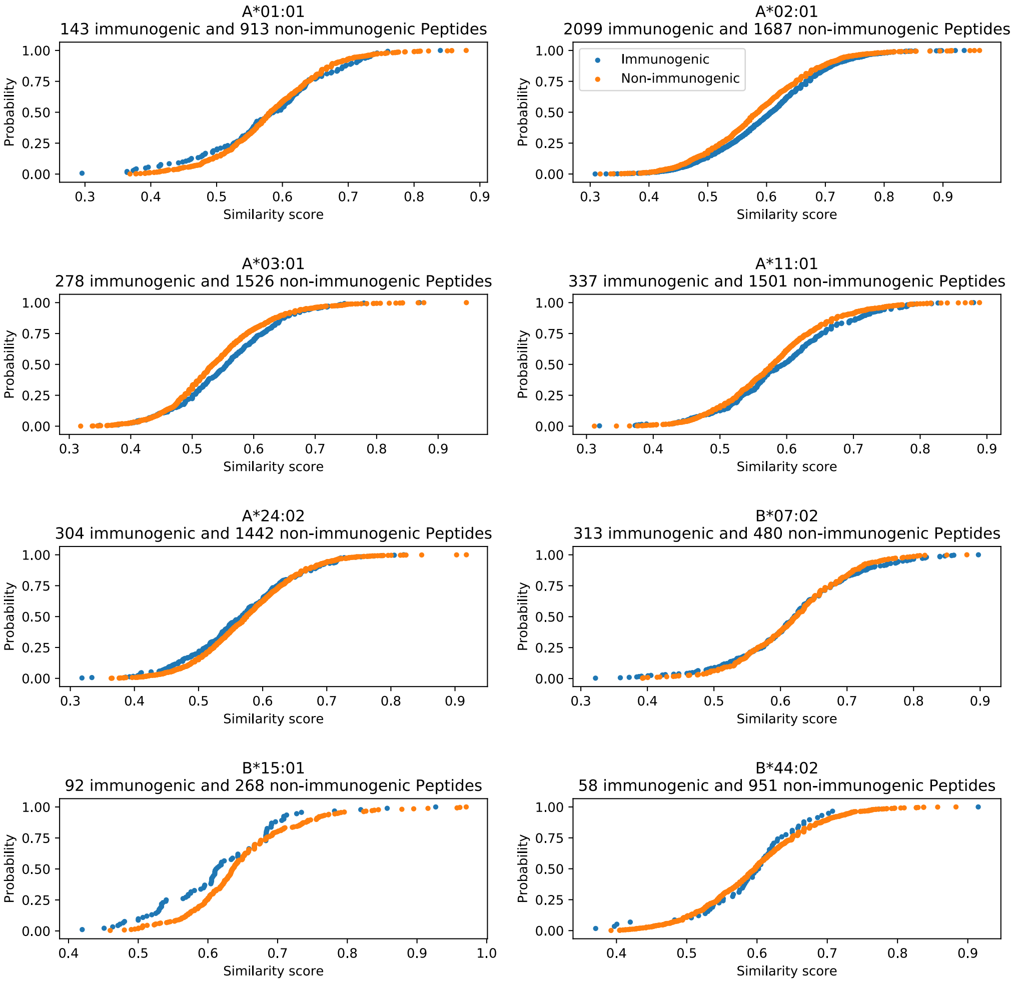


Figure 3: Empirical cumulative distributions of the BLOSUM similarity score of the nearest neighbor in the HLA-Ligandome considering peptides with invariant length. The nearest neighbor of a query is a self-peptide with the longest highest scoring substring. The corresponding size and HLA-allele of the dataset is annotated above each plot. Exact matching substrings were excluded from these distributions.

.4 Equalized binding affinities

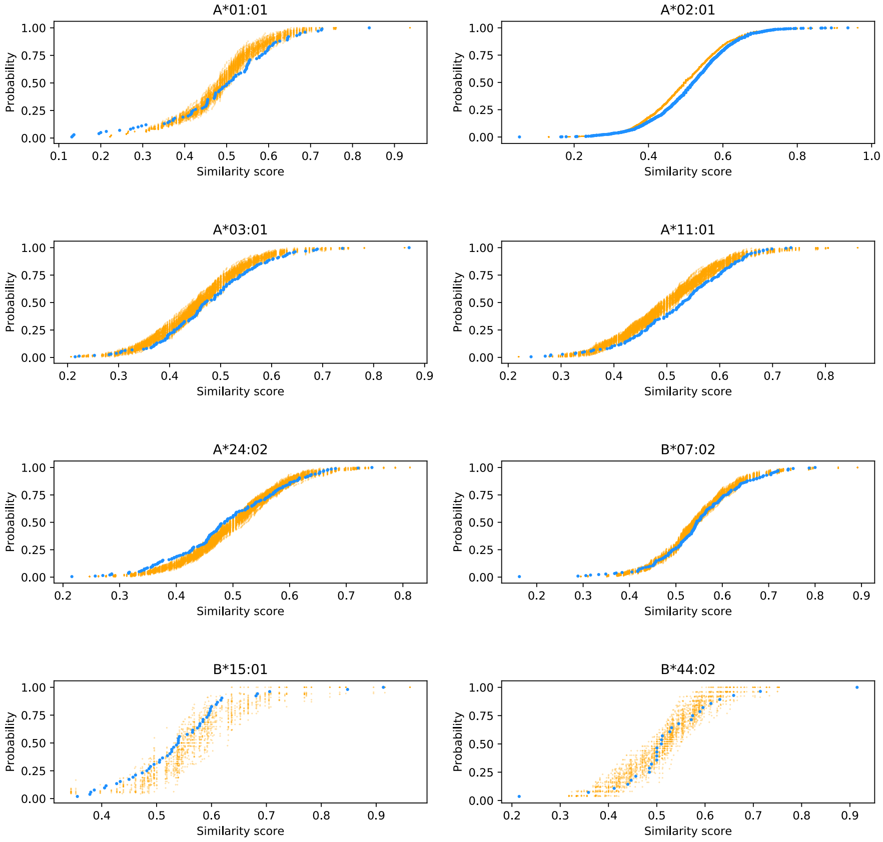


Figure 4: Empirical cumulative distributions of the BLOSUM-based similarity score for 100 subsampled non-immunogenic peptides. The immunogenic peptides were binned in quantification bins to approximate their MHC binding affinity score distribution. According to the obtained frequencies we subsampled non-immunogenic peptides into the same quantifications bins 100 times. This equalizes the binding affinity distributions between both data sets. With these sets, we computed similarity to self as usual, and the corresponding results are shown here. We see the BLOSUM scores for the immunogenic peptides and the BLOSUM scores for all 100 non-immunogenic subsets.

.5 Predicted proteome BLOSUM score for k=10

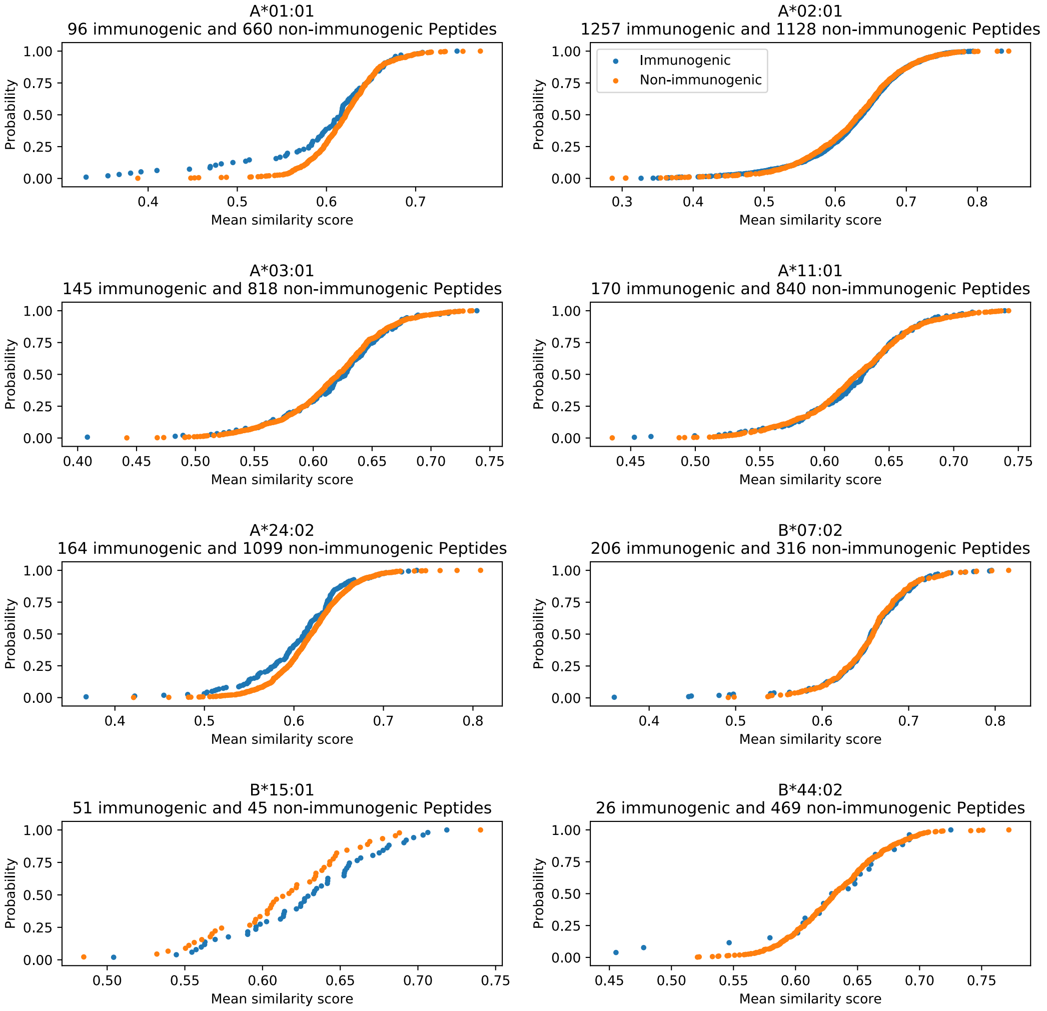


Figure 5: For each HLA allele the empirical cumulative distribution of the mean BLOSUM similarity score for the 10 nearest neighbors in the predicted proteome is plotted. The corresponding size of the datasets is annotaed above each plot. Exact matches were excluded from the distribution.

.6 Predicted thymus proteome BLOSUM score for k=10

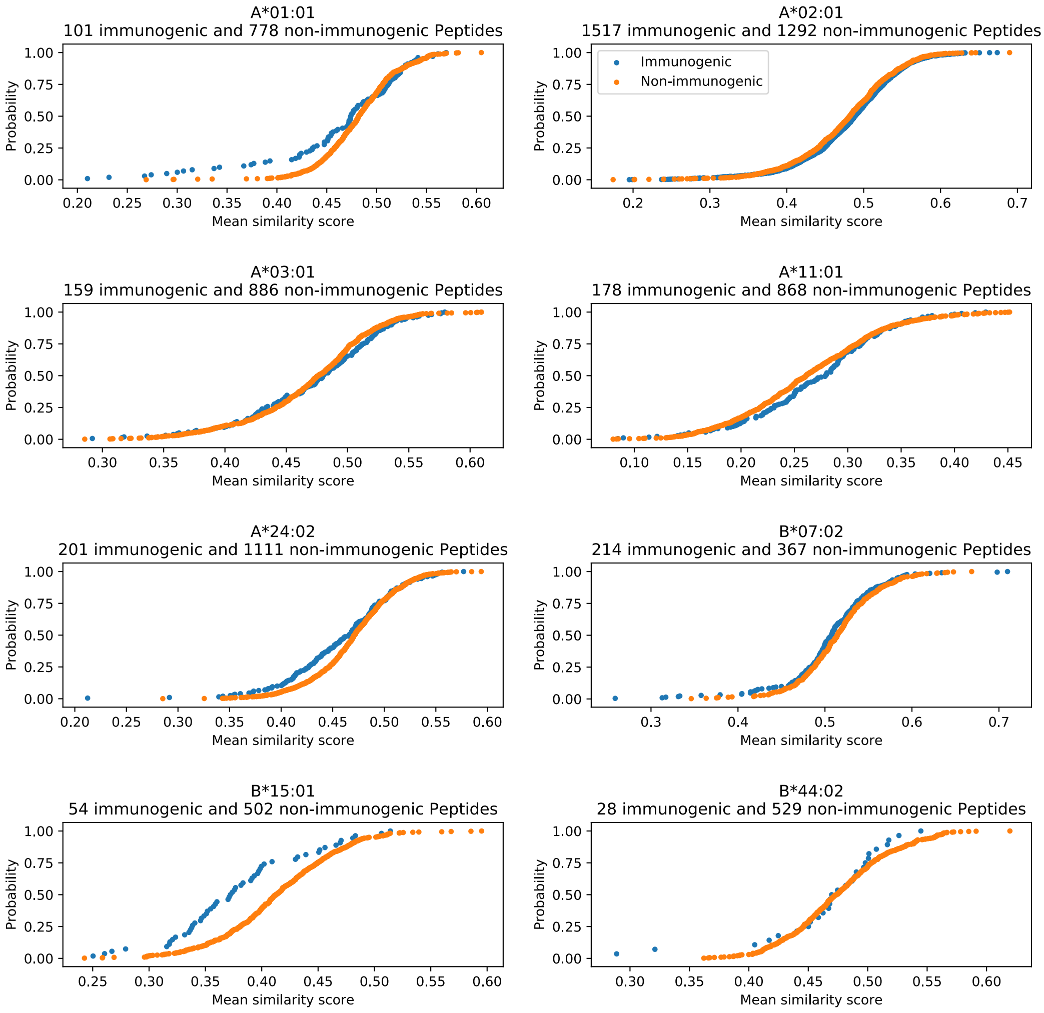


Figure 6: For each HLA allele the empirical cumulative distribution of the mean BLOSUM similarity score for the 10 nearest neighbors in the predicted thymus proteome is plotted. The corresponding size of the datasets is annotaed above each plot. Exact matches were excluded from the distribution.

.7 2-fold-cross-validation of position specific weights

Table 1: The median BLOSUM similarity scores obtained in a 2-fold cross-validation. The genetic algorithm should minimize the immunogenic median BLOSUM similarity.

HLA	Immunogenic	Non-immunogenic
A*01:01	0.614	0.638
A*02:01	0.642	0.634
A*03:01	0.563	0.547
A*11:01	0.644	0.631
A*24:02	0.551	0.572
B*07:02	0.696	0.706
B*15:01	0.610	0.654
B*44:02	0.633	0.606

.8 2-fold-cross-validation of AAIndex peptide encoding

Table 2: The median Euclidean distance obtained in a 2-fold cross-validation. The genetic algorithm should maximize the immunogenic median distance.

HLA	Immunogenic	Non-immunogenic
A*01:01	4.998	4.742
A*02:01	3.210	3.227
A*03:01	4.213	3.964
A*11:01	4.225	3.953
A*24:02	4.921	4.679
B*07:02	4.356	4.070
B*15:01	4.190	3.685
B*44:02	4.653	4.792

.9 F1-scores for the position specific weighting

Table 3: The F1-scores obtained for final models with uniform weighting (No weights), the determined optimized weights in Table 3.5 (SVM weights) and the determined weights in Table 3.4 that minimize immunogenic BLOSUM similarity to self (MIN weights). In the last column a stratified random dummy classifier is performance is annotated.

		Total F1-scores		
HLA	No weights	MIN weights	SVM weights	Dummy
A*01:01	0.13	0.32	0.37	0.13
A*02:01	0.59	0.51	0.66	0.54
A*03:01	0.3	0.25	0.32	0.15
A*11:01	0.34	0.30	0.38	0.17
A*24:02	0.22	0.28	0.32	0.15
B*07:02	0.35	0.32	0.54	0.17
B*15:01	0.43	0.51	0.51	0.15
B*44:02	0.14	0.14	0.14	0.05
Mean	0.31	0.33	0.41	0.19

.10 F1-scores for SVMs using different feature maps from the AAIndex

Table 4: The F1-scores of the feature mapped distance score for peptides encoded by AAIndices listed in Table 3.5 (SVM) and the determined weights in Table 3.4 (MAX) that maximize immunogenic distance to self. In the last column a stratified random dummy classifier is performance is annotated.

		Total F1-scores	
HLA	MAX	SVM	Dummy
A*01:01	0.34	0.42	0.13
A*02:01	0.56	0.69	0.54
A*03:01	0.34	0.39	0.15
A*11:01	0.38	0.43	0.17
A*24:02	0.38	0.41	0.14
B*07:02	0.58	0.61	0.17
B*15:01	0.54	0.64	0.15
B*44:02	0.19	0.32	0.05
Mean	0.41	0.49	0.19

Bibliography

[AC10] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statist. Surv.*, 4:40–79, 2010.

[BC00] Kristin P. Bennett and Colin Campbell. Support vector machines: Hype or hallelujah? *SIGKDD Explor. Newsl.*, 2(2):1–13, December 2000.

[BNB$^+$19] Tatjana Bilich, Annika Nelde, Leon Bichmann, Malte Roerden, Helmut R Salih, Daniel J Kowalewski, Heiko Schuster, Chih-Chiang Tsou, Ana Marcu, Marian C Neidert, Maren Lübke, Jonas Rieth, Mirle Schemionek, Tim H Brümmendorf, Vladan Vucinic, Dietger Niederwieser, Jens Bauer, Melanie Märklin, Janet K Peper, Reinhild Klein, Lothar Kanz, Hans-Georg Rammensee, Stefan Stevanovic, and Juliane S Walz. The hla ligandome landscape of chronic myeloid leukemia delineates novel t-cell epitopes for immunotherapy. *Blood*, 133(6):550–565, 2019.

[BPS$^+$16] Anne Bresciani, Sinu Paul, Nina Schommer, Myles B. Dillon, Tara Bancroft, Jason Greenbaum, Alessandro Sette, Morten Nielsen, and Bjoern Peters. T-cell recognition is shaped by epitope sequence conservation in the host proteome and microbiome. *Immunology*, 148(1):34–39, 2016.

[CdBK12] Jorg J. A. Calis, Rob J. de Boer, and Can Keşmir. Degenerate t-cell recognition of peptides on mhc molecules creates large holes in the t-cell repertoire. *PLOS Computational Biology*, 8(3):1–11, 03 2012.

[CMG$^+$13] Jorg J. A. Calis, Matt Maybeno, Jason A. Greenbaum, Daniela Weiskopf, Aruna D. De Silva, Alessandro Sette, Can Keşmir, and Bjoern Peters. Properties of mhc class i presented peptides that enhance immunogenicity. *PLOS Computational Biology*, 9(10):1–13, 10 2013.

[Con18] The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 11 2018.

[DIIM04] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, SCG '04, pages 253–262, New York, NY, USA, 2004. ACM.

[DSKK01] Jens Derbinski, Antje Schulte, Bruno Kyewski, and Ludger Klein. Promiscuous gene expression in medullary thymic epithelial cells mirrors the peripheral self. *Nature Immunology*, 2(11):1032–1039, 2001.

[FdBL$^+$08] Sune Frankild, Rob J. de Boer, Ole Lund, Morten Nielsen, and Can Kesmir. Amino acid similarity accounts for t cell cross-reactivity and for "holes" in the t cell repertoire. *PloS one*, 3(3):e1831–e1831, Mar 2008. 18350167[pmid].

[GTW99] K. Christopher Garcia, Luc Teyton, and Ian A. Wilson. Structural basis of t cell recognition. *Annual Review of Immunology*, 17(1):369–397, 1999. PMID: 10358763.

[GYD$^+$08] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou. On the class imbalance problem. *Fourth International Conference on Natural Computation, ICNC '08*, Vol. 4, 10 2008.

[HH92] S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[HRR$^+$12] Mikkel Harndahl, Michael Rasmussen, Gustav Roder, Ida Dalgaard Pedersen, Mikael Sørensen, Morten Nielsen, and Søren Buus. Peptide-mhc class i stability is a better predictor than peptide affinity of ctl immunogenicity. *European Journal of Immunology*, 42(6):1405–1416, 2012.

[Hun07] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[JOP$^+$] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed ¡today¿].

[JPA$^+$17] Vanessa Jurtz, Sinu Paul, Massimo Andreatta, Paolo Marcatili, Bjoern Peters, and Morten Nielsen. Netmhcpan-4.0: Improved peptide-mhc class i interaction predictions integrating eluted ligand and peptide binding affinity data. *Journal of immunology (Baltimore, Md. : 1950)*, 199(9):3360–3368, Nov 2017. 28978689[pmid].

[KK00] S. Kawashima and M. Kanehisa. Aaindex: amino acid index database. *Nucleic acids research*, 28(1):374–374, Jan 2000. 10592278[pmid].

[KSW$^+$08] Maya F. Kotturi, Iain Scott, Tom Wolfe, Bjoern Peters, John Sidney, Hilde Cheroutre, Matthias G. von Herrath, Michael J. Buchmeier, Howard Grey, and Alessandro Sette. Naive precursor frequencies and mhc binding rather than the degree of epitope diversity shape cd8+ t cell immunodominance. *Journal of immunology (Baltimore, Md. : 1950)*, 181(3):2124–2133, Aug 2008. 18641351[pmid].

[MWSS18] Kenneth Murphy, Casey Weaver, Lothar Seidler, and Lothar Seidler. *Janeway Immunologie -*. Springer-Verlag, Berlin Heidelberg New York, 9. aufl. edition, 2018.

[OY18] Masato Ogishi and Hiroshi Yotsuyanagi. The landscape of t cell epitope immunogenicity in sequence space. *bioRxiv*, 2018.

[PVG$^+$11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RW02] Markus G Rudolph and Ian A Wilson. The specificity of tcr/pmhc interaction. *Current Opinion in Immunology*, 14(1):52 – 65, 2002.

[SC08] M Slaney and Michael Casey. Locality-sensitive hashing for finding nearest neighbors [lecture notes]. *Signal Processing Magazine, IEEE*, 25:128 – 131, 04 2008.

[SPHB00] Anette Stryhn, LarsØstergaard Pedersen, Arne Holm, and Søren Buus. Longer peptide can be accommodated in the mhc class i binding site by a protrusion mechanism. *European Journal of Immunology*, 30(11):3089–3099, 2000.

[SVR$^+$94] A Sette, A Vitiello, B Reherman, P Fowler, R Nayersina, W M Kast, C J Melief, C Oseroff, L Yuan, J Ruppert, J Sidney, M F del Guercio, S Southwood, R T Kubo, R W Chesnut, H M Grey, and F V Chisari. The relationship between class i binding affinity and immunogenicity of potential cytotoxic t cell epitopes. *The Journal of Immunology*, 153(12):5586–5592, 1994.

[TEP+05] Fleur E. Tynan, Diah Elhassen, Anthony W. Purcell, Jacqueline M. Burrows, Natalie A. Borg, John J. Miles, Nicholas A. Williamson, Kate J. Green, Judy Tellam, Lars Kjer-Nielsen, James McCluskey, Jamie Rossjohn, and Scott R. Burrows. The immunogenicity of a viral cytotoxic t cell epitope is controlled by its mhc-bound conformation. *Journal of Experimental Medicine*, 202(9):1249–1260, 2005.

[TFZ+11] Nora C. Toussaint, Magdalena Feldhahn, Matthias Ziehm, Stefan Stevanović, and Oliver Kohlbacher. T-cell epitope prediction based on self-tolerance. In *Proceedings of the 2Nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB '11, pages 584–588, New York, NY, USA, 2011. ACM.

[TH07] Chun-Wei Tung and Shinn-Ying Ho. POPI: predicting immunogenicity of MHC class I binding peptides by mining informative physicochemical properties. *Bioinformatics*, 23(8):942–949, 03 2007.

[WAC+07] Kai W. Wucherpfennig, Paul M. Allen, Franco Celada, Irun R. Cohen, Rob De Boer, K. Christopher Garcia, Byron Goldstein, Ralph Greenspan, David Hafler, Philip Hodgkin, Erik S. Huseby, David C. Krakauer, David Nemazee, Alan S. Perelson, Clemencia Pinilla, Roland K. Strong, and Eli E. Sercarz. Polyspecificity of t cell and b cell receptor recognition. *Seminars in immunology*, 19(4):216–224, Aug 2007. 17398114[pmid].

[Whi94] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Bachelorarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum

Unterschrift