New Approaches to *in silico* Design of Epitope-Based Vaccines

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Dipl.-Inform. Nora C. Toussaint
aus Berlin

Tübingen
2011

Tag der mündlichen Qualifikation:	24.10.2011
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Oliver Kohlbacher
2. Berichterstatter:	Prof. Dr. Hans-Peter Lenhof (Universität des Saarlandes)
3. Berichterstatter:	Prof. Dr. Ole Lund (Technical University of Denmark)

Abstract

Traditional trial-and-error based approaches to vaccine design have been remarkably successful. One of the major successes was the eradication of smallpox in the 1970s. However, there are still many diseases for which no viable vaccine could be found, HIV infection and cancer being among the most prominent examples. Here, new, rationally designed types of vaccines, as, e.g., epitope-based vaccines (EVs) are a promising alternative. Due to their manifold advantages and their applicability in personalized medicine EVs have recently been attracting significant interest. EVs make use of target-specific immunogenic peptides, i.e., epitopes, to trigger an immune response.

In this thesis we propose new approaches to the *in silico* design of EVs. Given a set of target antigens, the first step in EV design is the discovery of candidate epitopes. Computational approaches to epitope discovery comprise major histocompatibility complex (MHC) binding prediction and T-cell reactivity prediction. The key problem in MHC binding prediction is the lack of experimental data for the vast majority of known allelic MHC variants. We present two support vector machine (SVM)-based approaches to overcome this problem. The first approach improves the predictive power of SVMs for alleles with little experimental binding data. The second approach – for the first time – allows predictions for all known MHC variants by exploiting structural similarities between different MHC molecules. The key problem in T-cell reactivity prediction are the complex dependencies of T-cell reactivity on the host proteome. We present the first approach that takes these dependencies into account. Our method markedly outperforms previously proposed approaches, indicating the validity of our approach.

Due to regulatory, economic, and practical concerns only a small number of candidate epitopes can be included in the EV. Hence, it is crucial to identify the optimal set of peptides for a vaccine. We formulate the epitope selection problem within a mathematical framework based on integer linear programming. The resulting optimization problem can be solved efficiently and yields a provably optimal peptide combination. We can show that the method performs considerably better than existing solutions. Furthermore, the framework is very flexible and can easily handle additional criteria.

For EV delivery, the selected epitopes are commonly concatenated into a single polypeptide. Since an unfavorable epitope order can result in the degradation of the intended epitopes, optimal epitope assembly is critical for the success of the EV. We present a graph-theoretical formulation of this problem that allows the efficient determination of optimal epitope orders. Application of the presented EV design approaches to realistic vaccine design studies yields promising results.

Zusammenfassung

Die traditionellen experimentellen Methoden zum Impfstoffentwurf blicken auf eine lange und erfolgreiche Geschichte zurück. Zu den bedeutendsten Erfolgen gehört die Ausrottung der Pocken in den 1970er Jahren. Dennoch gibt es immer noch Krankheiten, gegen die bisher kein geeigneter Impfstoff entwickelt werden konnte. Die wohl bekanntesten Beispiele sind HIV-Infektion und zahlreiche Krebserkrankungen. Hier sind neue, rational entworfene Impfstoffe, wie zum Beispiel epitopbasierte Impfstoffe (EVs), eine vielversprechende Alternative. EVs haben aufgrund ihrer Anwendbarkeit in der personalisierten Medizin sowie aufgrund vielfältiger weiterer Vorteile in letzter Zeit zunehmend an Aufmerksamkeit gewonnen. Sie basieren auf kurzen, aus Antigen abgeleiteten Peptiden, den so genannten Epitopen.

In dieser Arbeit stellen wir neue Ansätze zum computergestützten Entwurf von EVs vor. Ausgehend von einer Menge von Antigensequenzen werden im ersten Schritt des EV-Entwurfs Kandidatenepitope bestimmt. *In-silico*-Ansätze zur Bestimmung von Epitopen umfassen die Vorhersage von Peptiden, die an Moleküle des Haupthistokompatibilitätskomplexes (MHC) binden, sowie die Vorhersage von T-Zell-reaktiven Peptiden. Das Hauptproblem bei der Vorhersage von MHC-bindenden Peptiden ist der Mangel an experimentellen Daten für einen Großteil der bekannten allelischen MHC-Varianten. Wir stellen zwei Ansätze zur Überwindung dieses Problems vor. Beide Ansätze basieren auf Supportvektormaschinen (SVMs). Der erste Ansatz verbessert die Vorhersagekraft von SVMs in Bezug auf MHC-Allele mit wenigen experimentellen Daten. Der zweite Ansatz ermöglicht – zum ersten Mal – die Vorhersage für alle bekannten MHC-Varianten durch Ausnutzung struktureller Ähnlichkeiten zwischen unterschiedlichen MHC-Molekülen. Das Hauptproblem bei der Vorhersage von T-Zell-reaktiven Peptiden ist die komplexe Abhängigkeit der T-Zell-Reaktivität vom Immunsystem des Patienten. Wir stellen den ersten Vorhersageansatz vor, der diese Abhängigkeit berücksichtigt. Unsere Methode übertrifft bereits publizierte Methoden merklich, was auf die Gültigkeit unseres Ansatzes hindeutet.

Aufgrund von regulatorischen, ökonomischen und praktischen Überlegungen kann nur eine kleine Menge der Kandidatenepitope in den EV einbezogen werden. Daher ist es äußerst wichtig, die optimale Kombination von Peptiden für einen Impfstoff zu identifizieren. Wir formulieren das Epitopselektionsproblem als ganzzahliges lineares Programm. Das resultierende Optimierungsproblem lässt sich effizient lösen und liefert eine beweisbar optimale Peptidkombination. Wir können zeigen, dass unsere Methode deutlich bessere Ergebnisse liefert als existierende Lösungsansätze. Darüberhinaus ist der Ansatz sehr flexibel und kann ohne Weiteres zusätzliche Kriterien berücksichtigen.

Zur Verabreichung des EVs werden die Peptide üblicherweise zu einem einzelnen Polypeptid zusammengefügt. Da eine ungünstig gewählte Epitopanordnung zum Abbau der gewünschten Epitope durch das Proteasom führen kann, ist die optimale Anordnung von wesentlicher Bedeutung für den Erfolg des Impfstoffs. Wir stellen eine graphentheoretische Formulierung dieses Problems vor, die es ermöglicht, die optimale Anordnung der Epitope effizient zu bestimmen. Eine Anwendung der vorgestellten Methoden in realistischen Impfstoffentwurfsstudien liefert vielversprechende Ergebnisse.

Acknowledgments

First of all, I would like to thank my advisor Prof. Oliver Kohlbacher for introducing me to the fascinating subject of computational immunology and for giving me the opportunity to work in this area. Although I did not have any prior knowledge on bioinformatics, Oliver trusted in me and took me on as a PhD student. He gave me the freedom to explore and provided me with guidance when I asked for it. I very much appreciate his constant support and our inspiring discussions.

I am very much obliged to Prof. Hans-Peter Lenhof and Prof. Ole Lund for going to the time and effort of reviewing this thesis.

Furthermore, I am very grateful to all my collaborators, especially to Prof. Yoram Louzoun, Dr. Gunnar Rätsch, and Dr. Gunnar W. Klau: Yoram Louzoun gave me the opportunity to spend some time in his lab in Israel. It was a great pleasure to work with him and his group. In numerous scientific discussions with Gunnar Rätsch and in the projects with him and his PhD student Christian Widmer I learned quite a bit about machine learning and science in general. The close collaboration with Gunnar W. Klau and his group was both scientifically and socially very rewarding to me.

I would like to thank my colleagues in the Kohlbacher lab for fruitful discussions and kind support. Special thanks go to the current and former members of the computational immunology SIG: first of all Magdalena Feldhahn and Sebastian Briesemeister but also Andreas Kämper, Mathias Walzer, Nico Pfeifer, and Pierre Dönnes. Moreover, I owe thanks to my former student Matthias Ziehm for his dedicated work on epitope prediction. Claudia Walter deserves my gratitude for her kind help with all kinds of administrative and organizational issues. Furthermore, many thanks go to our system administrators Jan Schulze and Muriel Quenzer.

I am grateful to Oliver, Gunnar, Lena, and Juliane for their valuable comments on the manuscript.

The last years would not have been nearly as pleasant without all the great people at the Center for Bioinformatics Tübingen, in particular Juliane, Lena, Sandra, and Kerzi: Kerzi and Sandra walked me through the early downs of the PhD. Lena, who I have been lucky to share the office with, has not only provided me with water, coffee, candy and lunch, but has also been a great help when it came to teaching assistance and immunology questions. The lunch hours and coffee room chats with Juliane greatly contributed to keeping my balance at the office.

I am very grateful to my friends and family for support, patience and steady reminders of what really matters in life. My deepest gratitude belongs to Gunnar for his constant willingness to offer encouragement and advice. His endless, infectious enthusiasm for science as well as his virtually implacable belief in the researcher in me kept me going when there seemed to be no light at the end of the tunnel.

In accordance with the standard scientific protocol, I will use the personal pronoun *we* to indicate the reader and the writer, or my scientific collaborators and myself.

Contents

Chapter 1

Introduction

Motivation

The development of vaccines and their subsequent large-scale prophylactic use was undoubtedly one of the most important advancements in medicine. Vaccines make use of the adaptive part of the human immune system to protect from infections as well as to fight chronic diseases and cancer.

From the very beginning, the development of vaccines was a process largely driven by experiment. Attenuation of pathogens turned out to be a very reliable route to new vaccines. However, there are still many diseases for which no viable vaccine could be found, HIV infection and cancer being among the most prominent examples. In these cases, the rational design of vaccines based on a molecular understanding of immunity, shows great promise. Rational vaccine design uses experimental and computational methods to identify immunogenic substances, i.e., *antigens*, suited to form the basis of the vaccine. Given a set of potential antigens, several options for the construction of a vaccine exist: the antigens or parts thereof can be used, either as intact protein [1, 2] or as corresponding RNA or DNA [3, 4]. Using only the smallest immunogenic regions of a protein antigen, the so-called *epitopes*, is particularly appealing, as these *epitope-based vaccines (EVs)* have numerous advantages over other types of vaccines: apart from the comparatively simple production in a well-controlled process, EVs can be designed – at least in theory – to evoke an immune response that is very specifically directed at highly immunogenic regions of antigens. This design process can also take the immune system of the host into consideration, providing the basis for a personalized therapy. It is thus hardly surprising that EVs have recently gained a lot of attention. Several clinical studies with EVs were successful [5–7]. Various commercial products have now entered clinical phase III trials, indicating that the first EVs for humans can be expected to enter the market in the near future.

The process of designing an EV can be roughly divided into three steps: epitope discovery, epitope selection and epitope assembly (Figure 1.1). In each of these steps, computational methods can be employed to facilitate the work of immunologists and to assist them in their decisions. This *in silico*-guided approach to EV design promises improved vaccines at reduced development time and cost.

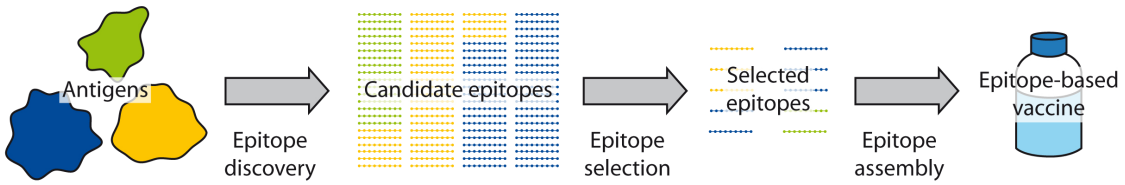


Figure 1.1: Epitope-based vaccine design. Given a set of antigens, candidate epitopes with respect to a target population or individual have to be determined (*epitope discovery*). Out of the set of candidate epitopes, the most suitable subset for use in the EV has to be selected (*epitope selection*). The EV is assembled from the selected epitopes (*epitope assembly*). Figure based on [8].

Epitope Discovery

Given one or more target antigens, a set of candidate epitopes needs to be determined and validated experimentally. Candidate epitopes are all antigen-derived peptides capable of inducing adaptive immunity in the target population.

Adaptive immunity is, at its core, triggered by the recognition of epitopes by cells of the immune system. Key players in cellular adaptive immunity are *MHC molecules* and *T cell receptors (TCRs)*. MHC molecules present peptides on the cell surface for surveillance by the immune system. The recognition of such a peptide:MHC (pMHC) complex by the TCR of a *T cell* induces an immune response (Figure 1.2). The genes encoding for MHC molecules are located in a large cluster of genes called *major histocompatibility complex (MHC)*. Within this complex different loci encode for MHC molecules, MHC class I (MHC-I) and MHC class II (MHC-II). MHC-I molecules typically bind and present peptides of length 8 to 10 derived from intracellular proteins. In contrast, MHC-II molecules present longer peptides, typically of length 13 to 18, derived from extracellular proteins. Because T cells recognize immunogenic peptides only when bound to MHC molecules, peptide binding to MHC is essential for the induction of a T cell-mediated immune response. It is, however, not sufficient: not every peptide that can bind to an MHC molecule will induce an immune response.

Due to the complex host dependencies that account for the T-cell recognition of a pMHC complex and the incomplete biological knowledge of the underlying processes, the

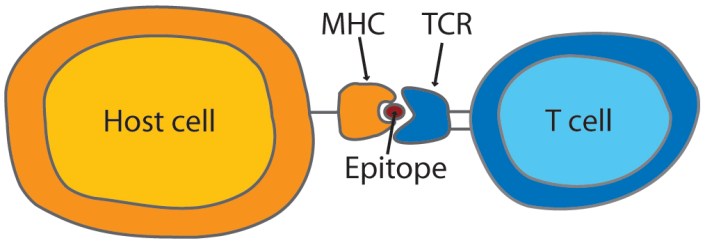


Figure 1.2: An MHC molecule presents an epitope (red) on the cell surface. The complex is recognized by the TCR of a T cell.

prediction of T-cell epitopes is a rather challenging problem. However, the capability of a peptide to induce a T cell-mediated immune response, i.e., its immunogenicity, has been shown to correlate well with MHC binding affinity [9]: most high-affinity peptides seem to be immunogenic. Since the processes governing pMHC binding are well-understood, the epitope discovery problem is typically reduced to the problem of MHC binding prediction.

MHC Binding Prediction

In 2004, Singh-Jasuja *et al.* presented the Tübingen approach [10] to acquire an experimentally validated initial list of epitopes from tumor-associated antigens. In their work, the incorporation of computational methods for prediction of pMHC binding in the process is proposed. Since such prediction methods help to reduce the number of experiments to be performed, they have become standard tools in immunology. Most popular approaches are either matrix-based [11, 12] or use machine learning techniques [13, 14]. Typical matrix-based approaches assume that each side chain of a peptide contributes independently to the peptide's binding affinity. In contrast, most machine learning methods are capable of capturing nonlinear correlations. Hence, the resulting models are generally more accurate. Their construction does, however, require considerable amounts of data.

A major challenge in MHC binding prediction is the highly polymorphic nature of the MHC. As of today, 4,946 MHC-I and 1,457 MHC-II alleles are known (IMGT/HLA database [15, 16], release 3.4.0). Different allelic variants of MHC molecules bind and present different sets of peptides. Hence, the MHC ligandome varies from individual to individual. This has important implications for vaccine design: because T cells recognize immunogenic peptides only when bound to MHC molecules, a peptide capable of inducing a T cell-mediated immune response in one individual might not be capable of doing so in another. Moreover, within a population, some MHC alleles are more common than others and the allele distribution differs between populations.

Classical prediction methods develop allele-specific models and require a minimum of experimental binding data for the respective allele. For the majority of all known MHC alleles the amount of available data is insufficient. However, to address the challenges of vaccine design, the binding specificities the gene products of all MHC alleles need to be known. It is thus crucial to find a way to overcome the problem caused by the lack of data.

We propose two approaches to overcome this problem and thus to improve MHC-I binding prediction. Both approaches employ *support vector machines (SVMs)*, a powerful and very popular machine learning technique. Given a sufficiently large set of examples, e.g., MHC *binders* and MHC *non-binders*, an SVM detects patterns within the examples and, based on these, learns to make intelligent decisions regarding unseen data. In order to do so, SVMs require so-called *kernel functions* or *kernels*, which measure the similarity between examples.

Our first approach to address the problem of scarce data focuses on improving the predictive power of SVMs for alleles with little experimental binding data by more skillfully exploiting the available information. MHC molecules bind peptides in an extended conformation. Within the complex the peptide's side chains interact with surrounding side chains of the MHC and also with each other. Each of the peptide's side chains contributes

to the binding affinity. The respective contributions are influenced by the side chain's physicochemical properties (e.g., hydrophobicity, charge, size), by its position within the peptide sequence as well as by the physicochemical properties of its neighboring side chains. While SVM-based predictors incorporating physicochemical properties of amino acids have been proposed [17, 18], the sequential information has not explicitly been taken into account. Particular kernel functions for sequence data, so-called *string kernels*, are perfectly suited to exploit the elongated conformation of putative MHC binding peptides. However, they do not allow an easy incorporation of prior knowledge on the properties of individual amino acids. We developed kernel functions that combine the benefits of string kernels with the advantages of physicochemical descriptors for amino acids. The proposed incorporation of amino acid properties into string kernels yields improved performances on MHC-I binding prediction compared to standard string kernels and to other previously proposed kernels without affecting the computational complexity. This improvement is particularly pronounced when data is less abundant.

Our second approach is based on the idea that structural similarities between different MHC molecules allow to employ binding information of one allele for the binding prediction of another. Groundbreaking in this regard is the work by Sturniolo *et al.* [19]. They consider the MHC-II binding groove to be composed of individual pockets. For each of the pockets a variety of amino acid compositions, the pocket variants, exists. These variants have been shown to be shared among different alleles. Using a matrix-based approach, Sturniolo *et al.* determined the binding specificity of an MHC-II allele by combining the binding affinities of the individual pocket variants constituting the binding groove. While this approach drastically increased the number of MHC-II alleles for which binding predictions can be performed, it is far from allowing predictions for all known MHC-II alleles. Inspired by their work, we propose a more general approach, which for the first time allows predictions for all known MHC-I alleles, independent of the availability of experimental binding data. A similar approach has been developed independently by Nielsen *et al.* [20].

T-Cell Epitope Prediction

Numerous groups worldwide have been working on the development of techniques for MHC binding prediction. As a result, state-of-the-art predictors can perform accurate predictions for all MHC alleles. As discussed above, MHC binding prediction is only a necessary prerequisite for T-cell epitope prediction. Hence, solving the binding prediction problem does not fully answer the questions arising in the context of vaccine design.

The ability of a peptide to trigger a T cell-mediated immune response depends on the host immune system, more specifically on the MHC alleles expressed and on the T-cell repertoire. Given a suitable MHC molecule that presents the peptide of interest on the cell surface, induction of an immune response requires recognition of the pMHC complex by a host T cell. However, the mechanisms governing T-cell recognition of pMHC complexes are not yet fully understood and therefore difficult to model. Furthermore, the availability of a suitable TCR is determined by a selection process that depends on the host's proteome. Existing T-cell epitope prediction methods [21, 22] do not include this

systemic property but employ peptide sequence information only. This limited view on T-cell reactivity contributes to the methods' insufficient prediction accuracies. In this thesis, we present a novel approach to T-cell epitope prediction combining sequence information with information on system-wide properties. Our results show that the incorporation of knowledge on the T-cell selection process yields considerable performance improvements over purely sequence-based approaches.

Epitope Selection

Having determined a set of candidate epitopes, the most suitable subset for inclusion in the EV has to be selected. Regulatory, economic and practical concerns impose strong limitations on the number of epitopes that can be included in an EV. Hence, a small set has to be selected from the list of epitope candidates: the set of epitopes, which yields the best immune response in the target population. This is a critical task in the design process, because the success of the vaccine is determined by the initial choice of epitopes. Moreover, the epitope selection problem can become very complex: properties to consider for the final choice vary from case to case. Key properties are the immune response to be expected in the target population, tolerance towards antigenic mutations, range of targeted antigens and cell-surface presentation. Given the set of candidate peptides, computational methods can be employed to determine the relevant attributes of each candidate. The final choice of the set of epitopes to be used in the vaccine, however, is typically performed manually by a group of experts. Only recently, the epitope selection problem has attracted the attention of bioinformatics groups. So far, only heuristic selection approaches to this optimization problem have been published [23, 24]. None of these can guarantee that there is not a better vaccine possible from the given set of epitopes. We formalize the epitope selection problem in a mathematical framework based on integer linear programming. Our approach allows an elegant and flexible formulation of numerous requirements on the epitopes to be selected. The method performs better than existing solutions and has runtimes of a few seconds for typical problem sizes.

Epitope Assembly

Regarding the delivery strategy of the epitopes selected for an EV, no consensus has been reached in the literature. Various delivery strategies are being explored in clinical studies [25]. A common approach is the concatenation of the vaccine epitopes into a single polypeptide, a so-called *string-of-beads construct*. Here, the chosen epitope order is crucial for the success of the vaccine: Once inside the host cell, the polypeptide will eventually be degraded into peptide fragments. An unfavorable epitope order can result in the degradation of the intended epitopes. Thus, in order to ensure full epitope recovery, knowledge on cleavage specificities has to be taken into account during construct assembly. Identification of the epitope order yielding the best epitope recovery is highly complex: 20 epitopes, for example, can be assembled into a string-of-beads construct in 6×10^{16} different ways. Manual determination of an optimal epitope order is obviously infeasible. However, this problem had not been addressed computationally before. We propose to

translate the corresponding optimization into a graph-theoretical problem. It then turns out to be the well-known Travelling Salesman Problem. Employing integer linear programming approaches, we show that the optimal ordering of realistically-sized epitope sets can be found efficiently.

Applications

In order to make the epitope selection framework available to immunologists, we developed OptiTope, a publicly available and easy-to-use web server for the selection of an optimal set of peptides for EVs. Furthermore, we applied our vaccine design approaches in two studies. In the first study we design a peptide cocktail vaccine against the hepatitis C virus. The resulting EV is superior to a previously proposed EV with respect to various quality criteria including overall immunogenicity and population coverage. In the second study we analyze the feasibility of designing string-of-beads vaccines against highly variable viruses providing broad population coverage as well as broad coverage of viral strains. Our analyses focusing on vaccines against hepatitis C virus, human immunodeficiency virus and influenza virus show promising results.

Structure of the Thesis

This thesis is structured into eight chapters. Following this introduction, the biological and the algorithmic background are introduced in Chapters 2 and 3, respectively. Chapter 4 describes and evaluates the methods and techniques we developed for *in silico* epitope discovery. Subsequently, our algorithms for *in silico* epitope selection and assembly are presented in Chapters 5 and 6. The seventh chapter focuses on applications: the epitope selection webserver OptiTope as well as the two *in silico* vaccine design studies are described. Chapter 8 provides a general conclusion of the presented work.

Chapter 2

Biological Background

In the following the biological background of this thesis will be introduced. We will start with a basic introduction to the immune system. Due to its importance for epitope-based vaccination, the focus will be on the cellular branch of the adaptive immune response and its key players. Subsequently, the fundamentals of vaccines and specifically of epitope-based vaccines will be introduced. For a more thorough introduction please refer to [26, 27]. Parts of this chapter have previously been published [8].

2.1 The Immune System

2.1.1 Introduction

The immune system protects the host from infectious agents, so-called *pathogens*, and cancer. Two fundamentally different but interdependent systems work in concert to provide this protection: the innate and the adaptive immunity. *Innate immunity* represents a first line of defense against many common pathogens. Anatomic barriers like skin and mucosa, physiological barriers like body temperature as well as phagocytic cells like macrophages belong to the innate immune system. When the barriers of innate immunity do not succeed in eliminating a pathogen, adaptive immunity comes into play.

2.1.2 Adaptive Immunity

Adaptive immunity owes its name to its capability to adapt to the pathogens it encounters. Unlike innate immunity, it is thus capable of learning: once a pathogen has been recognized, the adaptive immune system will remember it, allowing a stronger and more rapid response on reexposure. Based on this *immunological memory*, a successful immune response induces long-term protection, i.e., *immunity*, against the respective pathogen.

Lymphocytes, more precisely *B lymphocytes (B cells)* and *T lymphocytes (T cells)*, are key players in adaptive immunity. They carry receptors on their cell surface that enable them to recognize foreign substances. A substance that can be recognized and responded to by the adaptive immune system is called *antigen* [27]. Lymphocytes do not recognize the antigen in its entirety but only a small region, the so-called *epitope*. Each lympho-

cyte carries several thousand copies of the same receptor. Due to a genetic recombination mechanism, the genes encoding for the antigen receptor can generate 10^9 to 10^{10} different specificities. Despite this vast amount of different receptors the adaptive immune system generally does not attack self molecules but very selectively attacks foreign substances. Selection processes that eliminate lymphocytes carrying receptors recognizing self are responsible for this *self-tolerance*. Once a lymphocyte recognizes an epitope, it proliferates and differentiates into *effector cells* and *memory cells*. The short-lived effector cells are engaged in the elimination of the antigen, while the long-lived memory cells mediate the immunological memory and respond rapidly on reexposure to the same antigen.

The antigen receptors of B cells are membrane-bound antibodies. They recognize epitopes of antigen that is free in solution, ranging from bacteria to soluble proteins and polysaccharides. On recognition, B cells differentiate into antibody-secreting plasma cells. The secreted antibodies bind to antigen and thereby neutralize it or coat it facilitating elimination by cells of the innate immune system. The adaptive immune response induced by B cells is called *humoral immune response*.

The antigen receptors of T cells only exist in a membrane-bound form. These *T-cell receptors (TCRs)* are structurally similar to antibodies. However, they are only capable of recognizing antigenic peptides bound to specific cell-membrane proteins called *major histocompatibility complex (MHC)* molecules. The adaptive immune response induced by T cells is called *cellular* or *cell-mediated immune response*.

2.2 Cellular Immune Response

Vaccines make use of the the humoral as well as the cellular part of the adaptive immune response. The focus of this thesis is on the cellular immune response, which we will describe in more detail here.

2.2.1 Introduction

Key players in the cellular immune response are MHC molecules and TCRs. MHC molecules present peptides on the cell surface for surveillance by the immune system. These peptides are derived from intracellular and extracellular proteins via a process called *antigen processing*. Migrating T cells, via their TCRs, scan peptide:MHC (pMHC) complexes on the surfaces of other cells. On recognition of an epitope bound to an MHC molecule, a T-cell response is induced. The type of response depends on the respective T cell: *cytotoxic T cells (CTLs)* kill infected and abnormal self-cells, *helper T cells* activate other cells of the innate and adaptive immune system. A peptide that is capable of inducing a T cell-mediated immune response when bound to a specific allelic variant of MHC molecules is said to be *immunogenic* with respect to the corresponding MHC allele.

2.2.2 The Major Histocompatibility Complex

The *major histocompatibility complex (MHC)* is a large cluster of genes in jawed vertebrates. Within this region of the genome the *MHC class I (MHC-I)* and *MHC class II (MHC-II)*

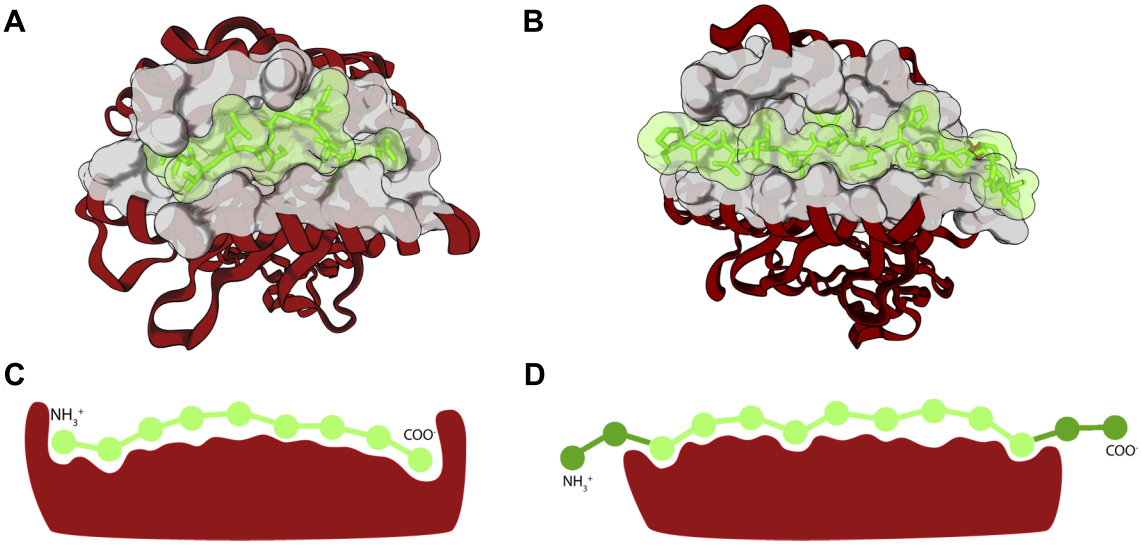


Figure 2.1: Peptides bound to MHC-I and -II binding grooves. A) 3D structure of a 10-mer bound to an MHC-I molecule (PDB-ID: 1JF1). B) 3D structure of a 15-mer bound to an MHC-II molecule (PDB-ID: 1BX2). The binding groove is colored gray. C) Sketch of a ninemer bound to an MHC-I binding groove. D) Sketch of a 13-mer bound to an MHC-II binding groove. AAs protruding out of the binding groove are colored dark green. (A and B were visualized with BALLView [29]. C and D are based on [8].)

genes are located. They encode for the α-chain of MHC-I and for the α- and β-chains of MHC-II molecules. In addition, the MHC comprises various other genes coding for proteins involved in antigen processing. In humans, this complex is called *human leukocyte antigen (HLA)* complex.

The MHC is polygenic: it contains several different MHC-I and MHC-II genes. In humans, there are three MHC-I α-chain genes (HLA-A, HLA-B, HLA-C) and three pairs of genes encoding for the MHC-II α- and β-chains (HLA-DR, HLA-DP, HLA-DQ). Furthermore, the MHC genes are highly polymorphic. As of today, 4,946 MHC-I and 1,457 MHC-II alleles are known. They encode for 3,647 different MHC-I and 1,073 different MHC-II molecules, respectively (IMGT/HLA database [15, 16], release 3.4.0).

The products of the MHC-I and MHC-II genes bind peptides in a peptide-binding groove (Figure 2.1A,B) and present them to T cells. The binding groove is formed by a large flat β-sheet enclosed by two α-helices. Its amino acid (AA) sequence varies between products of different MHC alleles. Thus, different allelic variants of MHC molecules bind different sets of peptides. Ligands of a particular MHC molecule display an allele-specific binding motif with a small number of highly conserved positions, the so-called *anchors*. Groups of MHC allelic variants with similar binding specificities are referred to as *MHC supertypes* [28].

Despite these similarities, MHC-I and MHC-II molecules have different functions.

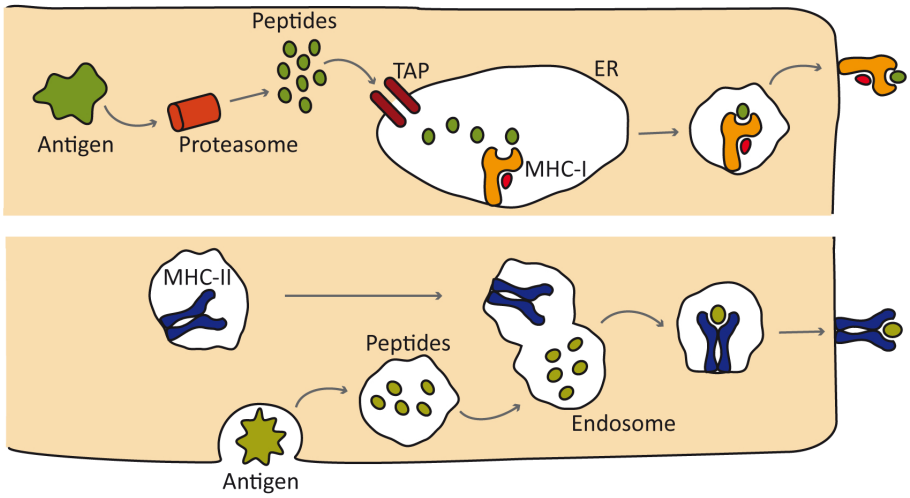


Figure 2.2: Antigen processing pathways. Intracellular pathway (top) and extracellular pathway (bottom) [8].

MHC-I molecules present peptides derived from intracellular proteins. They are expressed on all nucleated cells. MHC-I molecules consist of an α-chain, encoded in the MHC, and a smaller β_2-microglobulin chain. The binding groove of MHC-I molecules is closed at both ends (Figure 2.1A,C), restricting the length of MHC-I binding peptides to about eight to ten amino acids. Every individual possesses up to six different allelic variants of MHC-I molecules. MHC-II molecules, on the other hand, are only expressed on the surface of special *antigen-presenting cells (APCs)* where they present peptides derived from extracellular proteins. MHC-II molecules consist of an α- and a β- chain encoded in gene pairs in the MHC. Their binding groove is open at both ends allowing for a wide variety of ligand lengths because the ends of a bound peptide can protrude (Figure 2.1B,D). MHC-II molecules typically bind peptides of length 13 to 18. However, the binding groove only interacts with nine AAs. Every individual possesses up to 12 different allelic variants of MHC-II molecules.

Due to the high polymorphy of the MHC, the pool of MHC molecules varies from individual to individual. Moreover, within a population, some MHC alleles are more common than others and the allele distribution differs between populations. As we will discuss later, this has important implications for vaccine design.

2.2.3 Antigen Processing

Intracellular and extracellular antigens undergo different processing pathways before a small fraction of their peptides are presented to T cells (Figure 2.2). An intracellular antigen, like any other protein inside a host cell, is eventually degraded into peptide fragments by a protease complex, the *proteasome*. Some of these peptides are transported into the endoplasmic reticulum via the *transporter associated with antigen processing (TAP)*. Before and after TAP transport, peptide fragments may undergo N-terminal trimming (e.g.,

[30] and references therein). In the endoplasmic reticulum a pMHC-I complex is assembled and transported to the surface provided the peptide could bind to the respective MHC-I molecule. Each step of the intracellular processing pathway displays a certain degree of specificity and thus influences the peptides presented to T cells. By far the most specific step is MHC binding.

Extracellular antigen is ingested by APCs. Within endocytic vesicles, the antigen is degraded into peptide fragments. Some of these fragments bind to MHC-II molecules and are presented on the cell surface.

2.2.4 T Cells

Diversification

T cells originate in the bone marrow and mature in the thymus. During maturation the T cell's specific TCR variant is generated. The TCR is a heterodimer. Each of the two chains consists of a variable and a constant domain. The genes encoding the TCR chains are organized in groups of distinct segments. Rearrangement of these segments results in gene products comprising one of each segment type. On the order of 10^9 different TCR specificities can be generated. T cells that fail to successfully rearrange their TCR genes die by apoptosis. All others undergo thymic selection.

Thymic Selection

Thymic selection is responsible for shaping a T-cell repertoire that displays a wide variety of different TCRs to provide maximal protection against pathogens while being tolerant towards self-peptides. Two processes contribute to thymic selection: positive and negative selection. During *positive selection*, which takes place in the thymic cortex, the TCRs are tested for self-MHC recognition. Only T cells expressing TCRs capable of recognizing self-MHC are expedient for the host immune system. All others do not pass positive selection and die by neglect. Besides ensuring self-MHC recognition, positive selection is also responsible for determining the future function of a T cell. In the beginning of the thymic selection process T cells are double positive, i.e., they express *CD4* and *CD8* co-receptors on their cell surface. After positive selection they are either CD4 positive ($CD4^+$) or CD8 positive ($CD8^+$). The TCR of a $CD4^+$ T cell is restricted to MHC-II and the TCR of a $CD8^+$ T cell is restricted to MHC-I.

Positively selected cells relocate within the thymus to the medulla, where they reside for four to five days scanning self-pMHC complexes on the surface of APCs [31]. T cells displaying a high-affinity for a self-pMHC complex are potentially harmful for the organism. They are thus deleted in a process termed *negative selection*, which is responsible for ensuring self-tolerance of the T-cell repertoire. This so-called *central tolerance* applies to proteins expressed in the thymus and also to tissue-specific proteins not expected to be expressed in the thymus, e.g., pancreatic insulin [27]. A gene called *AIRE (autoimmune regulator)* has been found to control expression of such proteins by certain APCs in the medulla [32]. However, this does not apply to all self-proteins. Various mechanisms outside the thymus are responsible for preventing a T cell-mediated immune response to

peptides derived from self-proteins not expressed in the thymus. This *peripheral tolerance* complements the central tolerance provided by thymic negative selection.

Both positive and negative selection are based on the same process: the recognition of self-pMHC complexes by TCRs. It is still not understood how one process can result in further maturation of a T cell during positive selection as well as in cell death during negative selection [27]. A popular hypothesis to resolve this selection paradox is the affinity model. According to this model the quality of the TCR-pMHC interaction determines whether a T cell passes thymic selection. TCRs with no or low affinity for self-pMHCs are rejected during positive selection, TCRs with high affinity during negative selection. Only T cells expressing TCRs with intermediate affinity for self-pMHC pass thymic selection and migrate to the periphery as *naive* T cells [31].

Peripheral Tolerance

Thymic selection does not delete all self-reactive T cells. A second layer of tolerance-inducing mechanisms protects the host from those self-reactive T cells that have entered the peripheral T-cell repertoire. Analogous to thymic negative selection, *peripheral negative selection* eliminates or inactivates self-reactive T cells on recognition of pMHC. Elimination of all T cells displaying a high affinity for a pMHC complex, however, would render T cell-mediated immune responses impossible. Hence, peripheral negative selection must only take effect under certain conditions: In the absence of co-stimulatory signals indicating infection, naive T cells that recognize a pMHC complex die or become anergic.

Another mechanism contributing to peripheral tolerance is the control of self-reactive T cells by a functionally distinct T-cell subpopulation, the *regulatory T cells*. Regulatory T cells express self-pMHC-specific TCRs. Self-antigens that activate $CD4^+$ T or $CD8^+$ T cells can also activate regulatory T cells resulting in suppression of the former [33].

Activation

Naive T cells circulate continuously in the peripheral lymphoid tissues in search of their respective antigen. They can only be activated by APCs. In order to become activated, a naive T cell has to recognize an epitope bound to an MHC molecule while simultaneously receiving a co-stimulatory signal, which can only be expressed by APCs. On activation, T cells proliferate and differentiate into *armed effector T cells*. $CD8^+$ T cells differentiate into CTLs that kill infected target cells. $CD4^+$ T cells differentiate into helper T cells, either into T_H1 or in T_H2 cells. T_H1 cells activate cells of the innate immune system such that they destroy intracellular microorganisms. T_H2 cells in contrast activate B cells.

All armed effector T cells cause effects on their target cells via the production of effector molecules. *Cytokines*, small proteins that alter the behavior of cells, are mainly employed by helper T cells but also by CTLs. *Cytotoxins* mediate the destruction of cells and are released by CTLs.

After clearance of an infection, homeostatic mechanisms cause most effector cells to die by apoptosis. Those that are retained become long-lived memory cells and provide immunological memory. On subsequent exposure to the same antigen, these memory cells will respond rapidly.

Homeostasis

New T cells enter the peripheral T-cell repertoire constantly: naive T cells relocate from the thymus and T cells in the periphery proliferate. In order to maintain the size of the T-cell repertoire stable, this constant growth has to be balanced [27]. This is achieved via a homeostatic process that is analogous to thymic positive selection: T cells in the periphery receive survival signals by interacting with self-pMHC complexes. Those cells that are deprived of survival signals undergo apoptosis.

2.2.5 Experimental Data

Within this thesis experimental MHC binding affinity data as well as experimental immunogenicity data are used. In the following, we will briefly introduce a selection of experimental methods that are commonly used to generate these kinds of data.

MHC Binding

Experimental methods to determine MHC binding affinities are commonly based on the competition of the peptide of interest with a labeled reference peptide (e.g., radio-labeled or fluorescently labeled) [34]. The labeled reference peptides are bound to isolated MHC molecules and varying concentrations of target peptide are added. After a certain amount of time, unbound peptides are removed and the amount of labeled peptides bound to the MHC molecules is measured. The concentration at which the target peptide inhibits binding of the reference peptide to 50% of the MHC molecules is used as a measure for MHC-binding affinity. It is called the *50% inhibitory concentration (IC_{50})*. A low IC_{50} corresponds to a high binding affinity. Since the IC_{50} of a peptide depends on the respective reference peptide, IC_{50} values measured using different reference peptides are not necessarily comparable.

Immunogenicity

Peptide immunogenicity is typically determined by measuring cytokine secretion of activated T cells or by measuring the frequency of peptide-specific T cells [35].

Cytokine secretion can be measured with an enzyme-linked immunospot (ELISPOT) assay [36]: A microplate is coated with cytokine-specific antibodies. The cells to be analyzed for peptide-specific reactivity as well as APCs and the peptide of interest are added. Activated peptide-specific T cells will produce cytokines, which will be captured by the antibodies. After removal of the cells, captured cytokines can be detected by adding labeled cytokine-specific antibodies.

The frequency of peptide-specific T cells can be determined using MHC tetramer technology. Fluorescently labeled tetramers of the pMHC complex of interest are employed to stain the respective T cells [37]. The stained T cells can subsequently be counted using flow cytometry.

2.3 Vaccines

2.3.1 Introduction

Vaccines make use of the adaptive part of the human immune system to protect from infections as well as to fight chronic diseases and cancer. The word *vaccine* is derived from the Latin term for cowpox, *variolae vaccinae*, and honors the pioneering work on immunology of the English physician Edward Jenner. At the end of the 18th century, Edward Jenner observed that cowpox virus could be used to protect against the infection with smallpox virus. Another pioneer in this field was the French microbiologist Louis Pasteur, who, in the second half of the 19th century, discovered that administration of attenuated Cholera bacteria induces protective immunity while causing only mild symptoms. These discoveries paved the way for the development of vaccines and for their large-scale use in medicine.

The basic concept of vaccination is to induce immunity without actually causing disease. Immunity can be acquired by active or passive immunization. *Active immunization* aims at eliciting protective immunity and immunological memory. This can be achieved naturally by infection with a pathogen or artificially by administration of a vaccine. Just like the vaccines developed by Edward Jenner and Louis Pasteur, many vaccines used today are *whole-organism vaccines*, i.e., they employ an *attenuated* or *inactivated* form of the pathogen to generate immunity. One of the major successes of this approach was the eradication of small pox in the 1970s.

Passive immunization aims at inducing temporary immunity by transferal of preformed antibodies. Such an induction of passive immunity occurs naturally when maternal antibodies are transferred to the fetus. Artificial passive immunization via vaccination is employed, for example, as tetanus prophylaxis after a dog bite. In contrast to active immunization, passive immunization does not activate the host's immune system. It is therefore not capable of inducing immunological memory against the pathogen; the resulting protection is transient.

Vaccines can have detrimental side effects. Hence, before a vaccine can enter the market, it has to pass a thorough approval process comprising animal studies followed by three phases of clinical trials. Only vaccine candidates that prove to be safe and effective with respect to the proposed indications will be approved. In order for the time- and money-consuming development of a vaccine to be cost-efficient, the pharmaceutical industry additionally requires vaccines to be inexpensive, easy to produce as well as convenient to store, transport, distribute and administer.

2.3.2 Epitope-Based Vaccines

Introduction

The traditional whole-organism approach to designing vaccines for active immunization has proven to be very successful. However, using an entire organism as vaccine involves a risk of inducing unwanted host responses by material contained in the pathogen. Furthermore, there are still many diseases for which no viable vaccine could be found, e.g., HIV and cancer. Here, novel strategies are called for. The increasing knowledge on pathogens and

the immune system paves the way for a rational approach to vaccine design, limiting the vaccine to those parts that are relevant for an immune response. *Rational vaccine design* uses experimental and computational methods to identify antigens, suited to form the basis of the vaccine. There are numerous options for constructing a vaccine once a set of potential antigens is known. The antigens or antigenic peptides can be used, either as complete AA sequence or as its corresponding RNA or DNA.

Epitope-based vaccines (EVs), using only the smallest immunogenic regions of a protein antigen, are a fairly new approach. They can be designed - at least in theory - to evoke an immune response that is very specifically directed at highly immunogenic regions of antigens. This design process can also take the immune system of the host into consideration: the tailoring to specific MHC alleles and combinations thereof provides the basis for a personalized therapy. Moreover, EVs have numerous advantages that make them particularly interesting for the pharmaceutical industry, for example, safety as well as ease of production, analytical control, and distribution.

It is not surprising, that EVs have recently been getting more and more attention. They have proven successful in preclinical trials in mice [38], on which many of the preliminary studies have been conducted. A large number of clinical studies, both from academia and industry, have also been successful and have entered and/or completed clinical phase I and II trials [5–7, 39, 40]. Several commercial products have now entered clinical phase III trials. The indications for the vaccines in trial are mostly various cancers (e.g., leukemia, colorectal cancer, gastric cancer, lung cancer) and infectious diseases (predominantly HIV and hepatitis C virus). Since health agencies such as the FDA require proof of the effectiveness and safety of every individual component of a vaccine, the number of epitopes to be included in the EV is typically kept rather small. However, so far there is no EV approved for use in human on the market. This is mainly attributed to difficulties with peptide stability and delivery [41].

Delivery

In the search for a practical, safe and potent means to deliver EVs, various delivery strategies are being explored in clinical studies [25]. Strategies range from (A) peptide cocktails (Figure 2.3A) to (B) concatenation of the epitopes into a string-of-beads construct (Figure 2.3B) to (C) concatenated epitopes with intervening spacer sequences (Figure 2.3C) or (D) longer polypeptides assembled by skillfully overlapping the selected epitopes (Figure 2.3D). Taken out of their antigenic context, epitopes display a reduced and possibly insufficient capability of inducing an immune response, as they are likely to be ignored by the immune system. This is obviously disadvantageous for vaccine potency. A means to overcome this problem are *adjuvants*. Adjuvants enhance the immunogenicity by provoking inflammatory reactions, i.e., they attract the attention of the immune system. In addition, some adjuvants protect the epitopes from extracellular proteases [42].

EV delivery strategies can be roughly divided into two classes: strategies delivering the peptides themselves and strategies delivering the corresponding RNA or DNA. Examples for the former are *lipopeptide vaccines*, *immunostimulating complexes* and *cell-based delivery*. Examples for the latter are *recombinant-vector vaccines* and *DNA vaccines*.

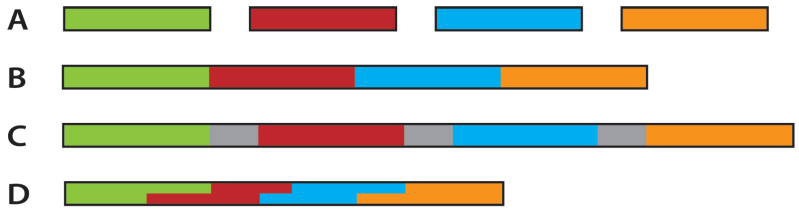


Figure 2.3: Peptide delivery strategies. A) Peptide cocktail, B) string-of-beads construct, C) string-of-beads construct with spacer sequences, D) polypeptide from overlapping epitopes.

Lipopeptide vaccines. In a lipopeptide vaccine a single peptide or a longer polypeptide is conjugated to a lipid. This vaccination strategy is inspired by bacterial lipoproteins, in which the lipid mediates attachment to the cell membrane as well as internalization into the host cell. Furthermore, it induces cytokine secretion of cells of the innate immune system [43]. Lipopeptides have been reported to induce potent immunity without adjuvant.

Immunostimulating complexes. An immunostimulating complex (ISCOM) is a small spherical particle with built-in adjuvant, mimicking a virus [44]. Acting as carrier structures, ISCOMs deliver the contained antigen or antigenic peptides into the cytosol.

Cell-based delivery. In cell-based peptide delivery the respective peptide is directly loaded onto cultured APCs *ex vivo*. Subsequently, the loaded APCs are administered. A major advantage of this strategy is the bypassing of the antigen processing [41].

Recombinant-vector vaccines. Recombinant-vector vaccines employ attenuated viruses to deliver the vaccine polypeptide into the host cell. A gene encoding a longer polypeptide containing the respective epitopes is introduced into the virus, which serves as a vector. Inside the host it replicates and the host cells express the polypeptide [26]. The choice of organism to be used as vector has a substantial influence on the safety and potency of the vaccine.

DNA vaccines. In DNA vaccines plasmid DNA encoding the vaccine polypeptide is injected directly into the muscle. Muscle cells and nearby APCs take up the DNA and express the polypeptide [26].

Chapter 3

Algorithmic Background

In the following, we will introduce the algorithmic background of this work. Section 3.1 deals with combinatorial optimization, which forms the basis for the vaccine design algorithms proposed in this thesis. In Section 3.2 the focus is on machine learning, a technique we employ for epitope discovery. For a more thorough introduction please refer to [45] for combinatorial optimization and to [46, 47] for the machine learning techniques employed in this thesis.

3.1 Combinatorial Optimization

3.1.1 Introduction

Optimization in general is concerned with making the best possible choice from a set of available alternatives. A constrained optimization problem has the form

$$\begin{aligned} &\text{maximize} && f_0(\mathbf{x}) \\ &\text{subject to} && f_i(\mathbf{x}) \leq b_i, \quad \text{for } i = 1 \dots m. \end{aligned} \tag{3.1}$$

The elements $x_1, \dots, x_n$ of the vector $\mathbf{x}$ are the *optimization variables* of the problem, the function $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the *objective* or *objective function*, the functions $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \dots, m$, are the *constraint functions*, and the constants $b_1, \dots, b_m$ are the *bounds* of the constraints. The set $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n | f_i(\mathbf{x}) \leq b_i, i = 1, \dots, m\}$ is called the *feasible region* or *search space*. Each $\mathbf{x} \in \mathcal{S}$ is a *feasible solution*. If $\mathcal{S}$ is non-empty, the corresponding problem is said to be *feasible*. A solution $\mathbf{x}^* \in \mathcal{S}$ is an *optimal solution* to the optimization problem in (3.1), if for all $\mathbf{x} \in \mathcal{S}$:

$$f_0(\mathbf{x}^*) \geq f_0(\mathbf{x}).$$

Depending on the form of the objective and of the constraint functions, an optimization problem is assigned to a particular class. Problems with linear objective and linear constraints, for example, belong to the *linear programs (LPs)*. LPs represent a special type of *convex optimization problems*, which comprise optimization problems with convex objective and a convex feasible set. The following optimization problem, for example, is an LP:

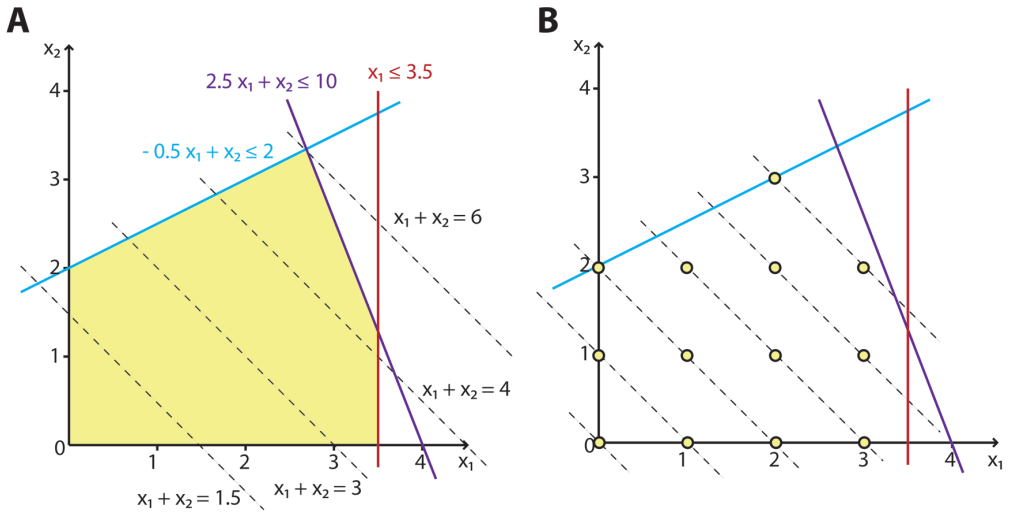


Figure 3.1: Linear optimization problems. The linear program given in (3.2) (A) and the corresponding integer linear program given in (3.4) (B) are shown. The constraint functions define a polytope (yellow area in A), which represents the feasible region or search space of the LP. The search space of the ILP comprises the integral solutions within the polytope (yellow dots in B). Different values of the objective function are displayed as dashed lines.

$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & 2.5x_1 & + & x_2 & \leq & 10, \\ & -0.5x_1 & + & x_2 & \leq & 2, \\ & x_1 & & & \leq & 3.5, \\ & x_1, x_2 & \in \mathbb{R}_0^+. & & & \end{array} \tag{3.2}$$

Figure 3.1A shows the geometrical interpretation of this problem.

Combinatorial optimization seeks an optimal object in a finite set of objects. Typically, this set has a concise representation (e.g., a graph) and grows exponentially in the size of the representation [48]. In contrast to general optimization problems, all optimization variables in a combinatorial optimization problem are integral [45]. Optimization problems with integral as well as continuous optimization variables are called *mixed integer programs*.

3.1.2 Integer Linear Programs

In this thesis we are concerned with problems with linear objective, linear constraints and only integral optimization variables. These problems are called *integer linear programs (ILPs)*. An ILP corresponds to an LP with integral unknowns. We write

$$\begin{array}{lll} \text{maximize} & \mathbf{c}^T\mathbf{x} & \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} & \\ & \mathbf{x} \in \mathbb{Z}^n. & \end{array} \tag{3.3}$$

where $\mathbf{c} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m,n}$ and $\mathbf{b} \in \mathbb{R}^m$. Typically, the optimization variables are binary, i.e., $x_i \in \{0, 1\}$. While the exact solution to convex optimization problems and thus LPs can be found efficiently, ILPs are NP-complete. An example for an ILP is given in the following:

$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & 2.5x_1 & + & x_2 & \leq & 10, \\ & -0.5x_1 & + & x_2 & \leq & 2, \\ & x_1 & & & \leq & 3.5, \\ & x_1, x_2 & \in \mathbb{Z}_0^+. & & & \end{array} \tag{3.4}$$

It corresponds to the LP in (3.2) with integrality constraints on all optimization variables. The geometric interpretation of this problem is shown in Figure 3.1B.

3.1.3 Methods

A well-known problem in combinatorial optimization is the *Travelling Salesman Problem (TSP)* [49]: Starting from and returning to his home town, a travelling salesman has to visit a given set of cities. The task is to find a shortest possible tour that visits each city exactly once. While there are only 12 feasible solutions for five cities, the number of solutions grows rapidly to 181,440 for 10 cities and to 6×10^{16} for 20 cities. Assuming that a computer requires $0.5\,\mu$s to evaluate one tour, an exhaustive search for 20 cities would take 964 years. This combinatorial explosion, which is typical for combinatorial optimization problems, calls for efficient optimization methods.

Optimization methods can be divided into two classes: heuristic methods and exact methods. The former find one or more good but not necessarily optimal solutions in reasonable time. If an optimal solution exists, the latter guarantee to find it at the cost of potentially longer run times.

In the following, we will briefly introduce a small selection of algorithms for solving ILPs: *branch-and-bound algorithms* [50], *cutting plane algorithms* [51], and *branch-and-cut algorithms* [52]. Approaches to optimally solving an ILP are generally based on solving the *LP relaxation*, i.e., the ILP without integrality constraints. This can be done efficiently. The corresponding objective value is called *dual bound* of the ILP. If the dual bound is integral, it corresponds to the solution of the ILP. Otherwise, further steps are required to solve the ILP.

Branch-and-bound algorithms. If the solution $\mathbf{x}'$ of the LP relaxation is not integral, branch-and-bound algorithms split the optimization problem into two subproblems. This step is called *branching*. Different branching strategies have been proposed. Often, an optimization variable that is not integral in the current solution is used: if $x_1' = 1.8$, then the constraint $x_1 \leq 1$ is added to one subproblem and the constraint $x_1 \geq 2$ to the other. In a *bounding* step upper and lower bounds of the current subproblem are determined. Every subproblem will continuously be solved and subdivided resulting in a search tree until either the global optimal solution is found or the bounding step reveals that the current branch can be pruned since it does not contain the optimal solution.

Cutting plane algorithms. If the solution $\mathbf{x}'$ of the LP relaxation is not integral, cutting plane algorithms will add a constraint to the relaxation which is satisfied by all feasible solutions to the ILP, but not by $\mathbf{x}'$. The constraint represents a *cutting plane* removing the current optimal solution from the feasible region. The modified LP relaxation yields a new dual bound. Solving of the LP relaxation and subsequently adding a cutting plane is performed until an integral solution is found.

Branch-and-cut algorithms. Branch-and-cut algorithms combine a cutting plane with a branch-and-bound approach.

State-of-the-art ILP solvers (e.g., CPLEX [53], MOSEK [54], GLPK [55]) typically employ *branch-and-cut algorithms*. Despite the NP-completeness, these tools find optimal solutions for ILPs quite efficiently up to a certain problem size. Factors influencing the effectiveness of these tools are, amongst others, size and sparsity of the problem, i.e., of the constraint matrix A [56].

3.2 Machine Learning

Machine learning is a popular and very powerful tool for prediction problems in computational biology. Given a sufficiently large set of examples (e.g., MHC binders and MHC non-binders) a machine learning algorithm detects patterns within the examples. Based on these patterns, the machine learning algorithm learns to make intelligent predictions regarding unseen data. A key advantage of machine learning is its capability to explain data for which no explicit model is available.

Prediction problems make up a large fraction of the problems dealt with in computational biology. A predictor maps objects or instances from the problem domain $\mathcal{X}$ to a label from the output domain $\mathcal{Y}$. Typical prediction problems to be solved using machine learning are classification and regression problems. In classification, an object is assigned to one of a given number of classes. Depending on the number of classes we are talking about binary classification or multi-class classification. Regression, on the other hand, deals with a continuous output domain. An MHC binding classifier, for example, is concerned with classifying a peptide as binder or non-binder while an MHC binding regressor predicts the binding affinity of a peptide to a given MHC molecule.

Within this thesis we employ a learning approach called *support vector machines*, which we will introduce in the following. Subsequently, we will describe techniques and measures for performance evaluation of prediction methods.

3.2.1 Support Vector Machines

Support vector machines (SVMs) are a fairly young approach to machine learning, which was introduced by Vapnik *et al.* [57] in 1995. They have been successfully applied to various problems from computational biology, for example gene identification [58], protein subcellular localization [59], and protein classification [60]. SVMs use the dot product to

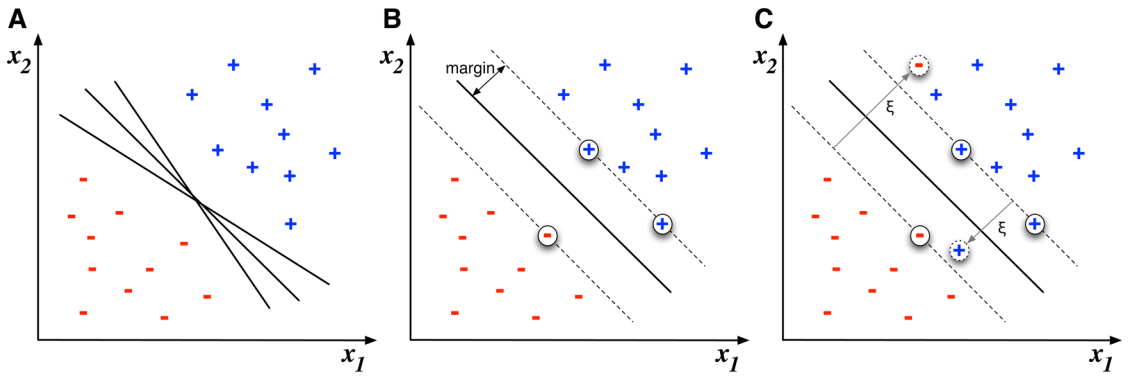


Figure 3.2: Linear classifiers. A) An infinite number of lines perfectly separate the two classes of points (plusses and minusses) in two dimensions. B) Maximum-margin hyperplane (solid line): The dashed lines mark the margin area. The data points highlighted with a circle are the support vectors. C) Maximum-margin hyperplane (solid line) with a soft margin: Misclassified data points are highlighted with a dotted circle. Misclassifications are penalized via the slack variables ξ_i. Based on [47].

measure the similarity between two instances. Hence, they require the instances from the problem domain to be represented as vectors in a dot product space $\mathcal{H}$. If the problem domain $\mathcal{X}$ is not a dot product space, a *mapping* or *encoding* $\Phi : \mathcal{X} \to \mathcal{H}$ has to be employed. Common encodings for AA sequences, for example, range from binary bit strings representing the AA sequence, that is, sparse encoding, to real-valued numbers corresponding to physicochemical properties of the respective AAs (e.g., size, charge, and hydrophobicity).

In the following, we will introduce SVM basics, beginning with binary support vector classification for linearly separable data, followed by linear support vector regression and kernel functions, which allow non-linear classification and regression.

Support Vector Classification

Given a set of data points $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where $\mathbf{x}_i \in \mathcal{H}$ and $y_i \in \{\pm 1\}$, *support vector classification (SVC)* aims at finding a linear function defining a hyperplane that perfectly separates the two classes $\{\pm 1\}$ (Figure 3.2A). This function is called *discriminant function*. It has the form

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{3.5}$$

with *weight vector* $\mathbf{w} \in \mathcal{H}$ and *bias* $b \in \mathbb{R}$. The hyperplane is represented by points satisfying the equation $f(\mathbf{x}) = 0$. It divides the input space $\mathcal{H}$ into two half spaces, one half space per class. This yields the following decision function:

$$\begin{aligned} g(\mathbf{x}) &= \operatorname{sgn}(f(\mathbf{x})) \\ &= \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b). \end{aligned} \tag{3.6}$$

It assigns an instance $\mathbf{x} \in \mathbb{R}^n$ to one of the classes in $\{\pm 1\}$. Please note that in the case of $f(\mathbf{x}) = 0$, no class can be assigned unambiguously since $\text{sgn}(0) = 0$.

If it is possible for a linear function to correctly classify each instance, i.e., if the data is linearly separable, there exists an infinite number of separating hyperplanes. According to statistical learning theory, the hyperplane separating the two classes with the largest margin, i.e., the hyperplane with the greatest distance to any of the data points in $\mathcal{D}$, is a good choice (Figure 3.2B). A hyperplane with $\mathbf{w}$ and b rescaled such that the points closest to the hyperplane satisfy $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1$ is called *canonical hyperplane*. Its margin is $\frac{1}{||\mathbf{w}||}$. Thus, the margin-maximizing canonical hyperplane can be found by minimizing $||\mathbf{w}||$, i.e., by solving the optimization problem:

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{H},\, b\in\mathbb{R}} \quad & \frac{1}{2}\,||\mathbf{w}||^2 \\ \text{subject to} \quad & y_i\big(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\big) \geq 1 \quad \text{for all } i = 1, \ldots, m. \end{aligned} \tag{3.7}$$

Such an optimization problem is typically solved via the *Lagrangian dual* [46]. The Lagrangian dual of the problem in (3.7) is

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{subject to} \quad & \alpha_i \geq 0 \quad \text{for all } i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y_i = 0. \end{aligned} \tag{3.8}$$

The solution of this dual problem also solves the original problem in (3.7), the so-called *primal problem*. It can be shown [47], that the weight vector $\mathbf{w}$ in equation 3.7 can be written as

$$\mathbf{w} = \sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i. \tag{3.9}$$

Incorporation into the decision function (3.6) yields

$$g(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b \right). \tag{3.10}$$

In this formulation, the classification of an instance $\mathbf{x}$ only depends on those training examples $\mathbf{x}_i$ with a Lagrange multiplier $\alpha_i > 0$. These are the examples that lie on the margin of the canonical hyperplane, i.e., $y_i\big(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\big) = 1$ (Figure 3.2B). They are called *support vectors*.

The optimization problem in (3.7) is feasible only if the data is linearly separable. However, in practice this is usually not the case. Deriving a classifier from non-separable data requires some misclassifications to be allowed. This can be achieved by introducing *slack variables* $\xi_i \geq 0$ for all $i = 1, \ldots, m$ to relax the constraints in (3.7) [61]:

$$y_i\big(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\big) \geq 1 - \xi_i \quad \text{for all } i = 1, \ldots, m. \tag{3.11}$$

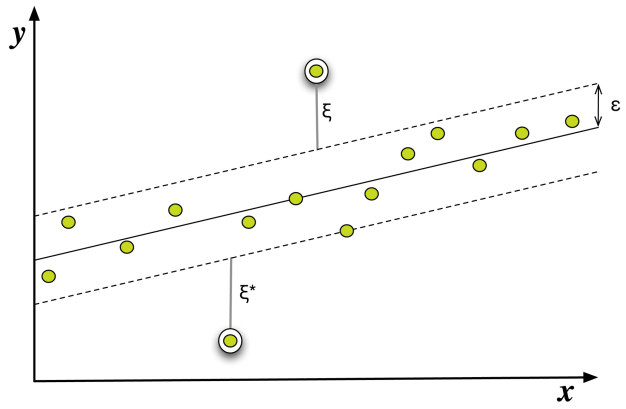


Figure 3.3: ε-SVR. The dashed lines mark the ε-tube around the regression line. Deviations within this area are not considered as errors. All others are penalized: data points above the tube via ξ, data points below the tube via ξ^*.

The *hard margin* is turned into a *soft margin* (Figure 3.2C). In order to prevent the margin from becoming too soft, i.e., ξ_i becoming too large, slack variables have to be penalized in the objective function. The objective function of the soft-margin SVC is as follows:

$$\min_{\mathbf{w}\in\mathcal{H},\ b\in\mathbb{R},\ \xi\in\mathbb{R}^m} \quad \frac{1}{2}\ ||\mathbf{w}||^2 + C\ \sum_{i=1}^{m} \xi_i\ , \tag{3.12}$$

where $C \geq 0$. The trade-off between margin maximization and error minimization is determined by the constant C. The corresponding dual problem is

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\ \sum_{i,j=1}^{m} \alpha_i\alpha_j y_i y_j\ \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \quad \text{for all } i = 1, \ldots, m \\ \text{and} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0. \end{aligned} \tag{3.13}$$

The decision function of the soft-margin SVC is identical to the decision function of the hard-margin SVC given in (3.10).

Support Vector Regression

Support vector regression (SVR) aims at learning a real-valued function $f(\mathbf{x})$ from the training examples $\mathcal{D} = \left\{(\mathbf{x}_i, y_i)\right\}_{i=1}^{m}$ with instances $\mathbf{x}_i$ as before and target values $y_i \in \mathbb{R}$ [62]. One formulation of this problem is ε-SVR. ε-SVR learns a function $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ yielding a deviation of at most ε from the target values (Figure 3.3). Slack variables allow for errors, i.e., deviation larger than ε. Two types of errors can occur: $f(\mathbf{x}_i) - y_i > \varepsilon$ and $f(\mathbf{x}_i) - y_i < \varepsilon$. Each type is covered by one type of slack variable: $\boldsymbol{\xi}$ and $\boldsymbol{\xi}^*$, respectively (Figure 3.3).

The optimization problem for an ε-SVR is:

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{H},\ b\in\mathbb{R},\ \xi,\xi^*\in\mathbb{R}^m} \quad & \frac{1}{2}\ ||\mathbf{w}||^2 + C\ \sum_{i=1}^{m}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \langle \mathbf{w}, \mathbf{x}_i\rangle - b \leq \varepsilon + \xi_i \\ & \langle \mathbf{w}, \mathbf{x}_i\rangle + b - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \text{ for all } i = 1, \ldots, m. \end{aligned} \tag{3.14}$$

This yields the following dual problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*\in\mathbb{R}^m} \quad & -\varepsilon \sum_{i=1}^{m}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{m}(\alpha_i^* - \alpha_i)y_i \\ & -\frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\,\langle \mathbf{x}_i, \mathbf{x}_j\rangle \\ \text{subject to} \quad & 0 \leq \alpha_i, \alpha_i^* \leq C \text{ for all } i = 1, \ldots, m \\ \text{and} \quad & \sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0\ . \end{aligned} \tag{3.15}$$

The corresponding prediction function is:

$$\begin{aligned} f(\mathbf{x}) &= \langle \mathbf{w}, \mathbf{x}\rangle + b \\ \text{with } \mathbf{w} &= \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)\ x_i\ . \end{aligned} \tag{3.16}$$

Non-linear SVMs

So far we have only considered linear SVMs. However, depending on the problem at hand, a non-linear function might perform better. The data in Figure 3.4A, for example, cannot be classified properly by a linear function. In SVMs non-linearity is achieved by mapping the instances into another, commonly higher dimensional, dot product space, typically referred to as *feature space*, in which they can be separated linearly. For instance, applying the *feature map* $\phi : \mathbb{R}^2 \to \mathbb{R}^3$ with

$$\phi\big((x_1, x_2)\big) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \tag{3.17}$$

to the data points in Figure 3.4A yields a distribution of the two classes in 3D that can be separated linearly (Figure 3.4B). Thus, mapping the input data into a suitable feature space prior to the application of the SVC algorithm permits non-linear separation. However, there is major drawback to this approach: For realistically sized problems the feature space can be very high-dimensional. Working in such a high-dimensional space is expensive with respect to computation time and memory usage. A solution to this problem was proposed by Boser *et al.* [63]. Consider the feature map in (3.17). The dot product can be computed as

$$\big\langle \phi(\mathbf{x}), \phi(\mathbf{x}')\big\rangle = \Big\langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (x'^2_1, x'^2_2, \sqrt{2}x'_1x'_2)\Big\rangle = \big\langle \mathbf{x}, \mathbf{x}'\big\rangle^2\ . \tag{3.18}$$

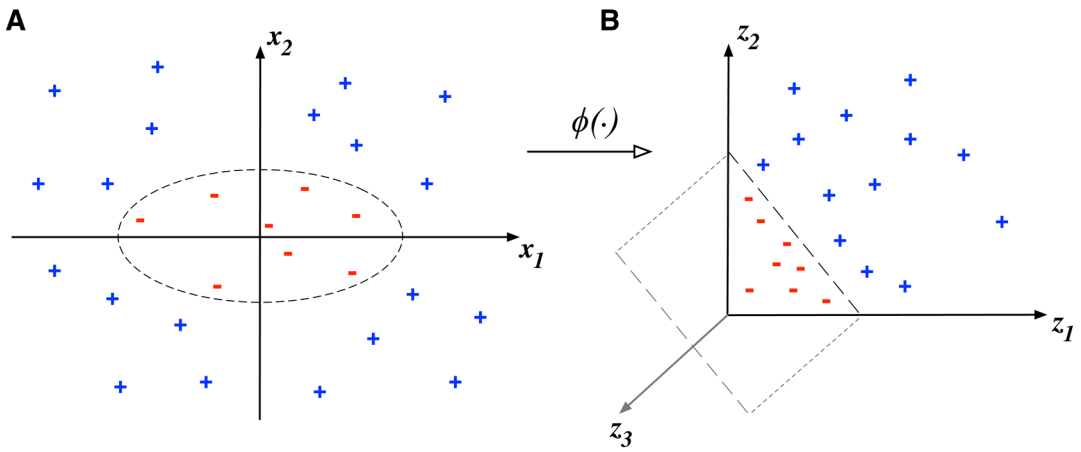


Figure 3.4: Input space vs. feature space. A) Input space: The data points cannot be separated properly by a linear function. B) Feature space: Mapping the two-dimensional input data given in A to a three-dimensional feature space using the feature map $\phi\big((x_1, x_2)\big) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ enables linear separation. Based on [46].

Note that the dot product in the feature space corresponding to ϕ can be determined without explicitly mapping into it (cf. right hand side of equation 3.18). This observation is useful since SV algorithms via their duals only depend on dot products between instances $\mathbf{x}_i$. Hence, it is sufficient to know the function $K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ to perform non-linear classifications. Knowledge of the corresponding feature map ϕ is not essential. The soft margin SVC (3.13) can be restated as follows:

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^m} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \quad \text{for all } i = 1, \ldots, m \\ \text{and} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{3.19}$$

and accordingly

$$f(\mathbf{x}) = \sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{3.20}$$

with $K(\mathbf{x}, \mathbf{x}')$ corresponding to the dot product in some feature space. If the underlying feature map is non-linear, then so is the SVC. Non-linearity in SVR is achieved analogously.

Kernel Functions

The question remains how to determine whether a function corresponds to the dot product in some feature space. According to *Mercer's Theorem* [64] this is the case if the function is symmetric and if its Gram matrix is positive semi-definite. Such a function is called *kernel function* or *kernel*. Kernels represent similarity measures.

Numerous kernel functions defining meaningful similarity measures have been proposed. Two popular kernels for real-valued data are the *polynomial kernel* and the *Gaussian RBF kernel*. The polynomial kernel of degree d with $d \in \mathbb{N}$ is defined as

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^d. \tag{3.21}$$

The higher the degree the more flexible is the resulting classifier [47]. A degree of one yields the dot product in input space, i.e., a linear kernel.

The Gaussian RBF kernel is defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right) \tag{3.22}$$

where $\sigma > 0$ determines the width of the Gaussian. Like the degree of the polynomial kernel, σ controls the flexibility of the resulting classifier [47].

A kernel type commonly used in computational biology are *string kernels*, i.e., kernels for sequence data. The basic idea of string kernels is to compare two input sequences based on the substrings they are composed of. String kernels can be divided into two classes: (i) kernels describing the k-mer content of sequences of varying lengths [65] and (ii) kernels for identifying localized signals within sequences of fixed length [66, 67]. The first class is typically used for classifying whole protein or mRNA sequences, while the second class is typically used to recognize a specific site in a window of fixed length sliding over a sequence. Of particular interest to this thesis is the second class. A representative of this class of string kernels is the *weighted degree (WD) kernel* [67]. The WD kernel of degree k is defined as

$$K_k^{\mathrm{wd}}(\mathbf{x}, \mathbf{x}') = \sum_{d=1}^{k} \beta_d \sum_{i=1}^{L-d+1} \mathbf{I}\left(\mathbf{x}_{[i:i+d]} = \mathbf{x}'_{[i:i+d]}\right) \tag{3.23}$$

where $\mathbf{x}_{[i:i+d]}$ is the substring of length d of $\mathbf{x}$ at position i. The weighting of the substring lengths β_d with $\beta_d = 2\frac{k-d+1}{k^2+k}$ assigns a lower weight to higher order matches.

3.2.2 Model Performance

Training an SVM on a given set of examples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ involves several parameters. An SVC with Gaussian RBF kernel, for example, has two parameters: σ and C. The configuration of these parameters strongly affects the performance of the resulting prediction model. The process of choosing a set of parameters is called *model selection*. Model selection is performed on the *training data set*. After a model has been selected, it is evaluated on the *test data set* with respect to a suitable performance measure. Training data and test data are disjoint subsets of $\mathcal{D}$.

Model Selection and Evaluation

Model selection is often performed as a *grid search*: Given a fixed number of possible values per parameter, all parameter combinations are evaluated. The combination yielding the best prediction model is selected.

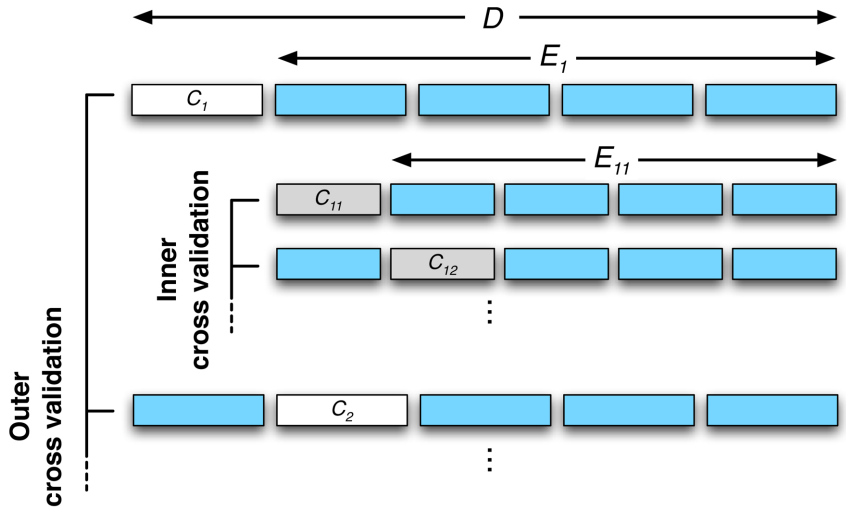


Figure 3.5: Two-fold nested five-fold cross validation. The training data D is partitioned into five disjoint subsets. In each of the outer cross validation rounds $i = 1, \ldots, 5$, model selection and training are performed on E_i and testing on C_i. Model selection comprises an inner cross validation. In each of the inner cross validation rounds $j = 1, \ldots, 5$ models for different parameter combinations are trained on E_{ij} and evaluated on C_{ij}. The parameter combination yielding the best performance in the inner cross validation is used for training on E_i in the respective outer cross validation round. Based on [68].

The *de facto* standard to thoroughly assess how a predictor will perform in practice, is k-*fold cross validation*. k-fold cross validation comprises k rounds of model selection and subsequent model evaluation. The data set $\mathcal{D}$ is adequately partitioned into k partitions of equal size: $\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_k$ with $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ for all $i \neq j$. In each round one of the subsets serves as test set C_i, while the remaining $k-1$ subsets are combined to form the training set E_i. Model selection and training are performed on the training set, model evaluation on the test set. Each of the k subsets is used as test set once. The overall performance of the predictor with respect to k-fold cross validation corresponds to the average over the performances from the k model evaluations.

In *two-times nested* k-*fold cross validation* the model selection phase of the k-fold cross validation comprises a cross validation-based grid search (Figure 3.5): The training data, $E_i = \mathcal{D} \setminus C_i$, is partitioned into n parts. For each parameter combination a model is trained on all combinations of $n-1$ inner subsets and evaluated on the remaining inner subset C_{ij}. The parameter combination with the best average performance over all n inner cross validation rounds is used to train a model on E_i. The resulting model is evaluated on the outer test set C_i.

Since on small data sets the initial partitioning may affect the resulting performance, the performance assessment is more precise if it is based on several rounds of k-fold cross validation, each employing a different partitioning.

Performance Measures

For model evaluation, predictions are performed on instances with known labels. Given the known labels $\mathbf{Y}$ and the model's predicted labels $\mathbf{Z}$, several measures can be employed to assess the performance of the respective model [69]. When choosing a performance measure one has to distinguish between measures for classification and for regression.

Using binary SVC, the known labels $\mathbf{Y}$ are binary. The values in $\mathbf{Z}$ can either be binary, i.e., the output of the decision function (3.6), or real-valued, i.e., the output of the discriminant function (3.5). When dealing with binary predicted labels all performance information is contained in four numbers: the number of correctly predicted positive examples (*true positives, TP*), the number of wrongly predicted positive examples (*false negatives, FN*), the number of correctly predicted negative examples (*true negatives, TN*), and the number of wrongly predicted negative examples (*false positives, FP*) [69]. Based on these numbers several performance measures are defined. Commonly used measures are *accuracy (ACC)* with

$$ACC(\mathbf{Y}, \mathbf{Z}) = \frac{TP + TN}{TP + FN + TN + FP} \ , \tag{3.24}$$

sensitivity (SE) with

$$SE(\mathbf{Y}, \mathbf{Z}) = \frac{TP}{TP + FN} \ , \tag{3.25}$$

and *specificity (SP)* with

$$SP(\mathbf{Y}, \mathbf{Z}) = \frac{TN}{TN + FP} \ . \tag{3.26}$$

In order to use these measures on real-valued predicted labels, a threshold separating positive from negative predictions has to be chosen. Since the selected threshold determines whether a prediction is considered positive or negative, different thresholds will yield different performance values. Setting the threshold to 0 corresponds to employing the decision function. The *area under the receiver operating characteristics curve (auROC)* measures the performance without requiring the selection of a specific threshold. The ROC curve plots $SE(\mathbf{Y}, \mathbf{Z})$ on the y-axis vs. $1 - SP(\mathbf{Y}, \mathbf{Z})$ on the x-axis as a function of the threshold. The *auROC* takes values between 0 and 1. A perfect predictor has an *auROC* of 1, a random predictor has an *auROC* of 0.5. Generally, when comparing different classifiers, the threshold-independent auROC is to be preferred over the threshold-dependent measures [70].

In regression, a commonly used performance measure is the *Pearson correlation coefficient*. It is defined as

$$PCC(\mathbf{Y}, \mathbf{Z}) = \sum_i \frac{(y_i - \bar{y})(z_i - \bar{z})}{\sigma_{\mathbf{Y}} \sigma_{\mathbf{Z}}} \tag{3.27}$$

where $\bar{y}$ and $\bar{z}$ are the mean values of $\mathbf{Y}$ and $\mathbf{Z}$, respectively, $\sigma_{\mathbf{Y}}$ and $\sigma_{\mathbf{Z}}$ are the corresponding standard deviations. The Pearson correlation coefficient assumes values between -1 and $+1$, with $+1$ indicating perfect correlation, 0 indicating random predictions, and -1 indicating total disagreement.

Chapter 4

Epitope Discovery

After having introduced the basic biological and algorithmic background of this thesis, we will now turn to the *in silico* design of EVs. The first step of designing an EV with respect to a given set of target antigens is epitope discovery (Figure 4.1). In the following, the methods and techniques we developed for this step of the EV design pipeline will be described and evaluated. Parts of this chapter have previously been published [8, 71, 72].

4.1 Introduction

Given one or more target antigens, a set of candidate epitopes needs to be determined and validated experimentally (Figure 4.1). The incorporation of *in silico* methods for epitope discovery can help to drastically reduce the number of experiments that have to be performed.

Due to the complex host dependencies that account for the T-cell recognition of a pMHC complex and the incomplete biological knowledge of the underlying processes, the prediction of T-cell epitopes is a rather challenging problem. However, Sette *et al.* [9] demonstrated a correlation between immunogenicity and MHC binding affinity of a peptide. Since the processes governing pMHC binding are well understood, the epitope discovery problem is typically reduced to the problem of MHC binding prediction. Only few approaches to predict T-cell epitopes directly have been published [21, 22].

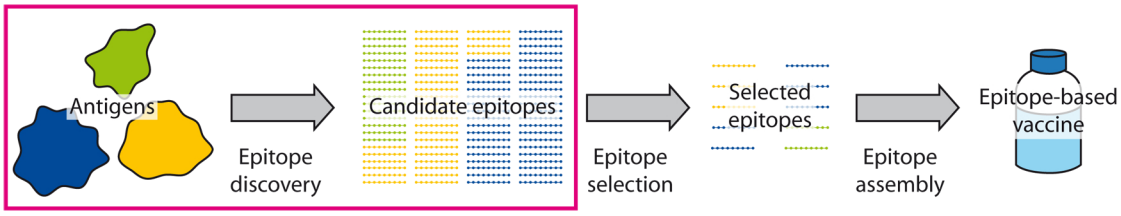


Figure 4.1: EV design: epitope discovery step. Given one or more target antigens, a suitable set of candidate epitopes needs to be determined. Figure based on [8].

Classical MHC binding prediction methods are allele-specific models and rely on the availability of a certain amount of experimental binding data for the respective allele. The most popular approaches are position-specific scoring matrices (e.g., SYFPEITHI [12], HLA BIND/Bimas [11]) and machine learning (e.g., SVMHC [73, 74], NetMHC [14, 75]). Sufficient training data is available only for the gene products of a small percentage of the 4,946 different MHC-I alleles (IMGT/HLA database [76], release 3.4.0). However, to address the challenges of vaccine design, the binding specificities of all allelic variants need to be known. In order to develop methods capable of binding prediction for alleles with few or no known binding peptides, a way to overcome the lack of data has to be found.

Structure-based approaches offer independence from experimental binding data by using known structures of pMHC-I complexes. Approaches range from docking [77] to binding matrices based on pairwise potentials [78]. Given an adequate template structure, these approaches allow for binding prediction for alleles with little or no available binding data. Structure-based approaches generalize well. However, most approaches require large computation times and prediction accuracies are not yet on par with sequence-based methods.

A popular sequence-based approach to address the problem of scarce data for a specific MHC allele is the incorporation of additional biological knowledge. MHC molecules bind peptides in an extended conformation. Within the complex the peptide's side chains interact with surrounding side chains of the MHC and also with each other. Each of the peptide's side chains contributes to the binding affinity. The respective contributions are influenced by the side chain's physicochemical properties (e.g., hydrophobicity, charge, size), by its position within the peptide sequence as well as by the physicochemical properties of its neighboring side chains. Several groups have successfully used knowledge on AA properties to improve MHC binding prediction [17, 18, 75]. However, the sequential information has not explicitly been taken into account. String kernels are perfectly suited to exploit the elongated formation of putative MHC binding peptides but they do not allow an easy incorporation of AA properties. In order to improve the predictive power of SVMs for alleles with little available data, we developed kernel functions that combine the benefits of string kernels with the ones of physicochemical descriptors for AAs. The proposed incorporation of AA properties in string kernels yields improved performances on MHC-I binding prediction compared to standard string kernels and to other previously proposed kernels without affecting the computational complexity. This improvement is particularly pronounced when data is less abundant. The improved kernels for MHC binding prediction are presented in Section 4.2.

Another approach to overcome the lack of experimental data employs sequence and structural information. It is based on the idea that structural similarities between different MHC molecules allow to use binding information of one allele for the binding prediction of another. Among the first to pursue such a *pan-specific* approach for MHC-II were Sturniolo *et al.* [19]. They consider the MHC-binding groove to be composed of individual pockets. For each of the pockets a variety of sequence compositions, the pocket variants, exists (Figure 4.2). These variants have been shown to be shared among different MHC molecules. Using a modular matrix approach, Sturniolo *et al.* determined the binding specificity of an MHC molecule by combining the binding affinities of the individual pocket

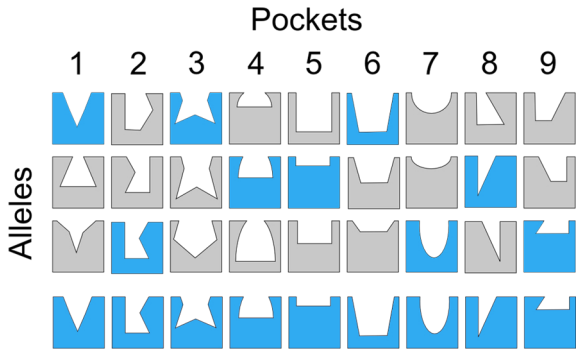


Figure 4.2: Schematic representation of the modular structure of an MHC binding groove. A number of sequence compositions exist for each pocket. These pocket variants, represented by different shapes, are shared across different allelic variants of MHC molecules.

variants constituting the binding groove. In 2007, DeLuca *et al.* [79] generalized this approach and applied it to MHC-I. Using the pocket definition of Chelvanayagam [80] in combination with a modular matrix approach, DeLuca *et al.* were able to increase the number of predictable MHC-I alleles by a factor of five. Nevertheless, for the vast majority of known MHC-I alleles the modular matrices were incomplete. These alleles could not be covered by this approach. Also inspired by the work of Sturniolo *et al.*, in Section 4.3 we propose a more general approach, which for the first time allows predictions for all known MHC-I alleles, independent of the availability of experimental binding data. Our approach performs better than existing allele-specific methods on alleles included in the training set. On alleles not included in the training set it achieves remarkable results, which are comparable to those of allele-specific methods. However, NetMHCpan, a related approach developed independently by Nielsen *et al.* [20], yields even better results.

As discussed above, MHC binding is only a necessary prerequisite for T-cell reactivity. The ability of a peptide to trigger a T-cell response, i.e., the peptide's immunogenicity, depends on the host immune system, more specifically on the MHC alleles expressed and the T-cell repertoire. Given a suitable MHC molecule that presents the peptide of interest on the cell surface, induction of an immune response requires recognition of the pMHC complex by a host T cell. The mechanisms governing T-cell recognition of pMHC complexes are not yet fully understood and therefore difficult to model. Furthermore, the availability of a suitable TCR is determined by positive and negative selection of T cells and thus by the host's proteome. These complex dependencies and the incomplete biological knowledge make the prediction of T-cell epitopes a very challenging problem. Only few classification approaches have been published. They range from early methods based on the amphipathic helix model of T-cell epitopes [81] to sequence-based machine learning methods [21, 22]. Bhasin and Raghava [21] proposed a consensus approach using artificial neural networks and SVMs on sparse-encoded peptides. Tung *et al.* [22] skillfully selected 23 physicochemical AA properties from the Amino Acid Index database (AAIndex) [82]. Based on this 23-dimensional AA encoding, the authors trained an SVC with Gaussian RBF kernel. Recently, Tung *et al.* have proposed a string-kernel-based SVM approach to

predict T-cell epitopes with respect to HLA-A2 (unpublished work). All of the approaches above yield only limited prediction accuracies. They suffer from an improper choice of data basis as well as from a limited view on this systemic property. None of the methods is based on a data set that has been analyzed with respect to a bias in MHC binding affinity. If the examples of one of the classes, immunogenic or non-immunogenic, display higher MHC binding affinities than those of the other class, it cannot be ruled out that the respective classifiers are trained to predict MHC binding instead of T-cell reactivity. Furthermore, the methods proposed in [21, 22] disregard the relevance of the MHC context for T-cell reactivity. The authors employ T-cell epitope data independent of the respective MHC allele for training. Since a particular peptide can be immunogenic with respect to one allelic variant of MHC molecules and non-immunogenic with respect to another, non-allele-specific T-cell epitope predictors cannot be expected to yield useful results. Apart from the choice of the data basis, all of the approaches above are purely sequence-based and do not consider the complex dependencies involved in peptide immunogenicity. In Section 4.4 we present a novel approach to T-cell epitope prediction incorporating knowledge on the T-cell selection process. Demonstrating a proof-of-concept, we can show that our model performs considerably better than purely sequence-based approaches.

4.2 Improved Kernels for MHC Binding Prediction

4.2.1 Introduction

MHC molecules bind peptides in an extended conformation. Each of the peptide's side chains contributes to the binding affinity. The respective contribution is influenced by the position of a side chain within the peptide sequence as well as by its neighboring side chains. While standard kernels like the polynomial or the RBF kernel are capable of capturing the interactions between different peptide side chains, they do not fully consider the order of the side chains within the sequence. In contrast, localized string kernels are capable of taking the side chain order into account. A localized string kernel very well suited to handle MHC binding data is the WD kernel (3.23).

When using string kernels on protein sequences, one key disadvantage is that prior knowledge about the properties of individual AAs, for example their size, hydrophobicity, secondary structure preference, cannot be easily incorporated. Given a sufficient amount of training data, these properties can be learned implicitly by machine learning methods. However, especially when dealing with small training data sets as common in MHC binding prediction, inclusion of this information in the sequence representation would be beneficial.

Previous approaches to utilizing prior knowledge on AA properties and, hence, on relations between AAs in SVMs have either taken advantage of Blast or PSI-Blast profiles for improving kernels describing k-mer content [60, 83, 84] or focused on using properties for single AAs with standard kernels, i.e., ignoring the AA order within the sequences [17, 18, 75].

We propose a complementary approach of employing physicochemical or other information on AA properties. In the following, we present new kernel functions that combine

the benefits of physicochemical descriptors for AAs with the ones of string kernels. The main idea is to replace the comparison of substrings, which is computed during kernel computation, with a term that takes the AA properties into account. While this seems quite simple at first sight, it is less so, when considering k-mers instead of single AAs. The key insight is how to compute the kernels such that the beneficial properties of sequence kernels do not get lost. In particular, we would like that either the use of uninformative descriptors (e.g., sparse encoding) or the choice of distinct kernel parameters reduces the new kernel to the original string kernel.

Since string kernels describing k-mer content are not suited for epitope discovery, within this thesis we will focus on localized string kernels, in particular on the WD kernel. The proposed modifications can also be applied similarly to string kernels describing k-mer content as presented in [71]. Since we are interested in AA sequences, for the remainder of this section we generally assume the alphabet Σ to be the AA alphabet.

4.2.2 Methods

Idea

The WD kernel (3.23), as most other string kernels, compares substrings of length k between the two input sequences $\mathbf{x}$ and $\mathbf{x}'$ of length L. The involved term $\mathbf{I}(\overline{\mathbf{x}} = \overline{\mathbf{x}}')$ with $\overline{\mathbf{x}}, \overline{\mathbf{x}}' \in \Sigma^k$ can equivalently be written as:

$$\mathbf{I}(\overline{\mathbf{x}} = \overline{\mathbf{x}}') = \left\langle \Phi_k(\overline{\mathbf{x}}), \Phi_k(\overline{\mathbf{x}}') \right\rangle ,$$

where $\Phi_k(\overline{\mathbf{x}}) \in \mathbb{R}^{|\Sigma^k|}$. $\Phi_k(\overline{\mathbf{x}})$ can be indexed by a substring $s \in \Sigma^k$ and it is $\Phi_k(\overline{\mathbf{x}})_s = 1$, if $s = \overline{\mathbf{x}}$, and 0 otherwise. For the sake of the derivation, let us consider $\Phi_1 : \Sigma \mapsto \{0, 1\}$, generating a simple encoding of the letters into $|\Sigma|$-dimensional unit vectors, i.e., a sparse encoding. It can easily be seen that the substring comparison can be rewritten as

$$\mathbf{I}(\overline{\mathbf{x}} = \overline{\mathbf{x}}') = \prod_{i=1}^{k} \left\langle \Phi_1(\overline{\mathbf{x}}_i), \Phi_1(\overline{\mathbf{x}}'_i) \right\rangle .$$

When using Φ_1 as the basis of substring comparisons, relations between AAs are ignored: all mixed pairs of AAs are equally dissimilar. Only homogeneous pairs of AAs are considered. Replacing Φ_1 with a feature map Ψ that considers AA properties yields the following kernel on AA substrings (*AA substring kernel, AASK*):

$$K_k^{\Psi}(\overline{\mathbf{x}}, \overline{\mathbf{x}}') = \prod_{i=1}^{k} \left\langle \Psi(\overline{\mathbf{x}}_i), \Psi(\overline{\mathbf{x}}'_i) \right\rangle . \tag{4.1}$$

The corresponding feature space is then not spanned by $|\Sigma|^k$ different combinations of letters, but by D_{Ψ}^k, where D_{Ψ} is the number of properties used to describe a single AA. We can now recognize sequences of AAs that have certain properties (e.g., first AA: hydrophobic, second AA: large, third AA: positively charged, etc.): For every combination of products of features involving exactly one AA property per substring position, there is one feature induced in the kernel. A richer feature space including combinations of several

properties from every position can be obtained using the following two formulations. The first is based on the polynomial kernel (3.21):

$$K^{\Psi}_{k,d}(\overline{\mathbf{x}}, \overline{\mathbf{x}}') = \left(\sum_{i=1}^{k} \left\langle \Psi(\overline{\mathbf{x}}_i), \Psi(\overline{\mathbf{x}}'_i) \right\rangle \right)^{d}, \tag{4.2}$$

and the second on the Gaussian RBF kernel (3.22):

$$K^{\Psi}_{k,\sigma}(\overline{\mathbf{x}}, \overline{\mathbf{x}}') = \exp\left(-\frac{\sum_{i=1}^{k} \|\Psi(\overline{\mathbf{x}}_i) - \Psi(\overline{\mathbf{x}}'_i)\|^2}{2\sigma^2} \right). \tag{4.3}$$

Depending on the problem at hand, such a considerably richer feature space can be beneficial.

Modified WD Kernel

Replacing the substring comparison $\mathbf{I}(\overline{\mathbf{x}} = \overline{\mathbf{x}}')$ with one of the more general formulations in (4.1), (4.2), or (4.3) together with an AA encoding Ψ, directly implies a generalized form of the WD and other string kernels. For the WD kernel we can write:

$$K^{\mathrm{wd},\Psi}_{k}(\mathbf{x}, \mathbf{x}') \;=\; \sum_{d=1}^{k} \beta_d \sum_{i=1}^{L-d+1} K^{\Psi}_{d}(\mathbf{x}_{[i:i+d]}, \mathbf{x}'_{[i:i+d]}). \tag{4.4}$$

$K^{\mathrm{wd},\Psi}_{k}$ is a linear combination of kernels and therefore a valid kernel [85]. Independent of the choice of AASK, the modified WD kernel can be computed efficiently, with a complexity comparable to that of the original.

Of particular interest is the *WD-RBF kernel*, i.e., the combination of the WD kernel and the RBF-AASK (4.3):

$$K^{wd,\Psi}_{k,\sigma}(\mathbf{x}, \mathbf{x}') \;=\; \sum_{d=1}^{k} \beta_d \sum_{i=1}^{L-d+1} \exp\left(-\frac{\sum_{j=1}^{d} \|\Psi(\mathbf{x}_j) - \Psi(\mathbf{x}'_j)\|^2}{2\sigma^2} \right). \tag{4.5}$$

For $\sigma \to 0$ and an encoding Ψ with $\Psi(a) = \Psi(b)$ if and only if $a = b$, the WD-RBF kernel corresponds to the WD kernel: the RBF-AASK will be one only if the substrings are identical, otherwise it will be zero. Hence, as in the WD kernel only identical substrings will be considered.

4.2.3 Experimental Results

We evaluate the classification and the regression performance of the proposed kernels on a benchmark data set for MHC binding prediction: The IEDB benchmark data set from Peters *et al.* [86], which is part of the *Immune Epitope Database (IEDB)* [87], contains quantitative binding data (IC_{50} values) for various MHC alleles. Splits for a five-fold cross validation are given. We employ a subset of this data (hereafter $\mathsf{IEDB}^{\mathsf{h9}}$): binding data of nonameric peptides with respect to human MHC, yielding 35 allele-specific data sets. (The number of examples per allele in $\mathsf{IEDB}^{\mathsf{h9}}$ can be found in Table B.1 in the appendix.)

For the classification, peptides with IC_{50} values greater than 500 nM were considered non-binders, all others binders.

We use three sets of physicochemical descriptors for AAs: (1) five descriptors derived from a principal component analysis of 237 physicochemical properties (pca), (2) three descriptors representing hydrophobicity, size, and electronic properties taken from the AAIndex (zscale), and (3) 20 descriptors corresponding to the respective entries of the BLOSUM50 substitution matrix [88] (blosum50).

The main goal of the work presented in this section is the methodological improvement of existing string kernels by incorporation of prior knowledge on AA properties. In order to analyze the benefits of the proposed modifications we conducted performance comparisons between the original and the modified string kernels as well as standard kernels.

Preliminary Performance Analysis

Preliminary classification experiments on three human MHC alleles (HLA-A*23:01, HLA-B*58:01, HLA-A*02:01) were carried out to analyze the performance of the different kernels: WD (3.23), RBF (3.22), poly (3.21), WD-RBF (4.5), WD-poly (as WD-RBF, but with polynomial-AASK) combined with different encodings (pca, zscale, blosum50). The alleles were chosen to comprise a small data set (HLA-A*23:01, 104 examples) as well as a medium (HLA-B*58:01, 988 examples) and a large (HLA-A*02:01, 3089 examples) data set. The respective cross validation results are given in Table 4.1. For each of the alleles a different kernel type performs best: poly (pca) for HLA-A*23:01, RBF (blosum50) for HLA-B*58:01 and WD-RBF (blosum50) for HLA-A*02:01. The latter performs second-best on HLA-A*23:01 and HLA-B*58:01. As for the benefits of the modification of the WD kernel, the WD-poly and WD-RBF kernels outperform the WD kernel in 17 out of 18 cases.

Learning Curve Analysis

From Table 4.1 the trend can be observed that the kernels that use AA properties benefit more for smaller datasets. In order to validate this hypothesis, we performed learning curve analyses for WD and WD-RBF (blosum50) in a classification and a regression setting on the largest data set, i.e., HLA-A*02:01. Performance is measured by averaging the auROC and the PCC, respectively. To average over different data splits in order to reduce random fluctuations of the performance, we performed 100 runs of two-times nested five-fold cross validation. In each run, thirty percent of the available data was used for testing. From the remaining data training sets of different sizes (20, 31, 50, 80, 128, 204, 324, 516, 822, 1308) were selected randomly. Figure 4.3 shows the mean performances with standard errors. Both for classification and regression, it can clearly be seen that the fewer examples are available for learning, the stronger is the improvement of the WD-RBF kernel over the WD kernel. Intuitively this makes sense, as the more data is available, the easier it will be to infer the relation of the AAs from the sequences in the training data alone.

Table 4.1: Performances of kernels employing sequential structure and/or AA properties on three MHC alleles. auROCs and standard deviation were determined in two-times nested five-fold cross validation. Best (bold) and second-best (underlined) performances per MHC allele are highlighted. An asterisk marks performance improvement due to the proposed modifications.

Kernel	**HLA-A*23:01**		**HLA-B*58:01**		**HLA-A*02:01**	
	auROC	(std)	auROC	(std)	auROC	(std)
WD	0.7307	(0.0900)	0.9314	(0.0279)	0.9485	(0.0076)
Poly (pca)	**0.8363**	(0.0808)	0.9428	(0.0336)	0.9354	(0.0111)
Poly (zscale)	0.7964	(0.0727)	0.8778	(0.0637)	0.9052	(0.0070)
Poly (blosum50)	0.8220	(0.0442)	0.4948	(0.0560)	0.4729	(0.0246)
RBF (pca)	0.8277	(0.0904)	0.9396	(0.0303)	0.9345	(0.0114)
RBF (zscale)	0.7847	(0.0787)	0.9235	(0.0347)	0.9157	(0.0072)
RBF (blosum50)	0.8204	(0.0864)	0.9509	(0.0317)	**0.9520**	(0.0072)
WD-Poly (pca)	0.7879*	(0.0858)	0.9406*	(0.0319)	0.9495*	(0.0084)
WD-Poly (zscale)	0.7983*	(0.0902)	0.9499*	(0.0348)	0.9483	(0.0073)
WD-Poly (blosum50)	0.8307*	(0.1077)	0.9491*	(0.0224)	0.9490*	(0.0070)
WD-RBF (pca)	0.8133*	(0.0806)	<u>0.9510</u>*	(0.0265)	0.9486*	(0.0051)
WD-RBF (zscale)	0.7782*	(0.1222)	0.9487*	(0.0434)	0.9500*	(0.0074)
WD-RBF (blosum50)	<u>0.8312</u>*	(0.0993)	**0.9571***	(0.0265)	<u>0.9503</u>*	(0.0067)

Performance on the IEDB^{h9} Data Set

The preliminary analysis showed the WD-RBF kernel with blosum50 encoding to perform best. For a more comprehensive comparison, classification and regression performance of WD and WD-RBF (blosum50) kernels were assessed on the IEDB^{h9} data set. We used two-times nested five-fold cross validation to (1) perform model selection over the kernel and regularization parameters, (2) estimate the prediction performance. Performance is measured by averaging the auROC and the PCC, respectively. In classification, WD-RBF outperforms WD on 23 alleles (Figure 4.4A). (Allele-specific performances are listed in Table B.2 in the appendix.) This is significant with respect to the binomial distribution: Assuming equal performance of WD and WD-RBF, the probability of WD-RBF outperforming WD 23 out of 35 times is approximately 0.02.

In regression, WD-RBF outperforms WD on 27 alleles (Figure 4.4B). (Allele-specific performances are listed in Table B.3 in the appendix.) Again assuming equal performance of both kernels, the probability of WD-RBF outperforming WD 27 out of 35 times is approximately 0.002.

SVM Computations

All SVM computations were performed using the Matlab interface of the freely available large-scale machine learning toolbox Shogun [89]. All used kernels are implemented as part of the toolbox.

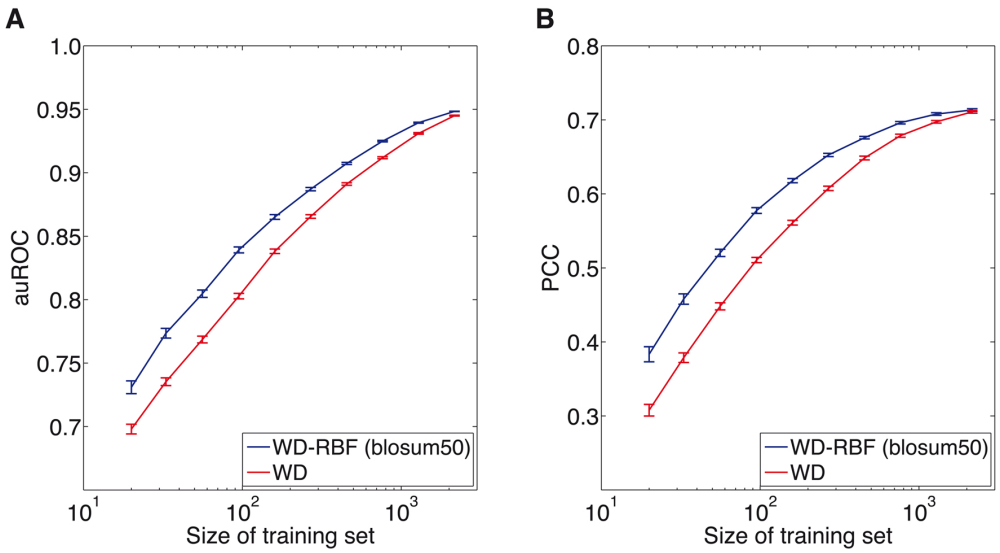


Figure 4.3: Learning curve analyses on HLA-A*02:01. Shown are performances of WD-RBF (blosum50) and WD kernel averaged over 100 different test splits (30%) and for increasing numbers of training examples (up to 70%). The training examples were used for training and model selection using five-fold cross validation. A) Classification performance (auROC). B) Regression performance (PCC).

4.2.4 Discussion

We have proposed new kernels that combine the benefits of physicochemical descriptors for AAs with the ones of string kernels. String kernels are powerful and expressive, yet one needs sufficiently many examples during training to learn relationships between AAs in the very high-dimensional and sparsely populated space induced by the string kernel. Standard kernels based on physicochemical descriptors of AAs, on the other hand, cannot exploit the sequential structure of the input sequences. They implicitly generate many more features, numerous of which will be biologically implausible. Here, one also needs more examples to learn which subset of features is needed for accurate discrimination, especially for longer protein sequences.

We could show that the proposed modifications of the WD kernel yield significant improvements in the prediction of MHC binding peptides. As expected, the improvement is particularly strong when data is less abundant. The experiments demonstrate that the proposed kernels indeed perform better than string kernels and non-substring kernels. These improvements are not major, but consistent. It has to be noted that a big difference between the previously proposed kernels and the proposed kernels cannot be expected: The proposed kernels essentially work on subsets of the features of previously proposed kernels and the improvements that we observe mainly come from the SVM's degraded performance when including uninformative features (which typically is not very pronounced).

In summary, the proposed modifications, in particular the combination with the RBF-AASK, consistently yield improvements without seriously affecting the computing time.

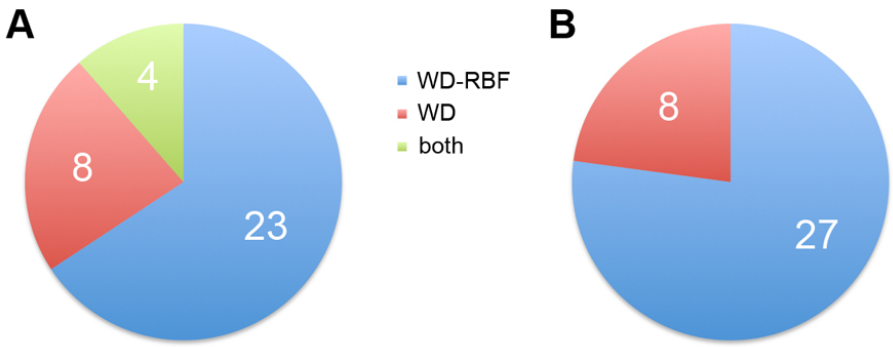


Figure 4.4: Performances of WD and WD-RBF (blosum50) kernels on the IEDB^{h9} data set. The pie charts display the number of alleles for which the WD (red) and the WD-RBF (blue) performed best, respectively, and the number of alleles for which they performed equally (green). A) Classification performances. B) Regression performances.

Utilization of the RBF-AASK allows for recovery of the original string kernel formulation by appropriately choosing σ. Hence, when σ is included in model selection, the proposed kernels should perform at least as good as the original string kernels.

4.3 MHC Binding Prediction for All MHC-I Alleles

4.3.1 Introduction

The approach discussed above aims at overcoming the lack of data in MHC binding prediction by skillfully exploiting the available allele-specific data. In contrast, our second approach employs similarities between the different allelic variants of MHC molecules. Inspired by the work of Sturniolo *et al.* [19], we propose a general method to the prediction of MHC-I binding peptides. This method, UniTope, for the first time allows predictions for all known MHC-I alleles, independent of the availability of experimental binding data. Based on an analysis of crystal structures of pMHC-I complexes, we determined, which MHC residues contribute to the individual pockets of the MHC binding groove. This information was used to determine pocket variants of the gene products of all known MHC-I alleles. Since the majority of known MHC-I binders are of length nine, we focused on nonameric peptides only and, accordingly, on a decomposition of the binding groove into nine individual pockets. A single SVM model was trained for all alleles. The input vectors for the SVM contain a physicochemical encoding of a peptide and of the nine pocket variants of the respective MHC molecule. A related approach, NetMHCpan, was developed independently by Nielsen *et al.* [20]. NetMHCpan is an artificial-neural-network-based consensus approach that also employs input vectors comprising peptide and MHC allele encoding. However, instead of representing the MHC-I binding groove by individual pockets, Nielsen *et al.* represent each allele by a *pseudo sequence*. The pseudo sequence comprises all non-conserved MHC residues found within 4Å of the ligand. The authors train a set of neural networks based on three different peptide-MHC encodings and the predicted affinity is determined by averaging over the individual predictions.

Performance tests with our approach yield promising figures: UniTope performs better than existing allele-specific methods on alleles included in the training set. On alleles not included in the training set, which correspond to alleles without experimental binding data, it achieves remarkable results, comparable to those of allele-specific methods.

4.3.2 Methods

Pocket Definition

A pocket of the MHC-I binding groove is composed of all residues in contact with the corresponding residue of a bound ninemer: MHC residues in contact with the first residue of the peptide belong to the first pocket, those in contact with the second residue belong to the second and so forth. The MHC-I sequence indices of residues found to contribute to a specific pocket are recorded in the *pocket profile*. In order to determine the pocket profiles, 3D structures of nonameric peptides bound to MHC-I molecules had to be analyzed. 75 crystal structures of such pMHC-I complexes were retrieved from the Protein Data Bank (PDB) [90] and analyzed using the BALL framework [91]. A list of these 75 structures is given in Table B.4 in the appendix. We applied the SS contact criterion [78] to determine contacts between MHC and peptide residues: An MHC residue and a peptide residue are defined to be in contact if they are at most 4Å apart. Interactions with the MHC backbone as well as with the peptide backbone are omitted. To ensure consistent indexing, MHC-I position indices were determined as follows: We retrieved AA sequences derived from all known human MHC-I alleles from the IMGT/HLA database [92] (release 2.16). Sequences derived from alleles which have been shown not to be expressed were discarded. Furthermore, all sequences with an incomplete binding groove were removed. A multiple sequence alignment (MSA) of the remaining sequences using ClustalW [93] showed a conserved sequence (`GSHSMRYF`) at the beginning of the α-chain. All MHC sequences were truncated to begin with this conserved sequence. The resulting pocket profiles are displayed in Figure 4.5.

Prediction Model

Our aim is to develop a single prediction model for all known allelic variants of MHC-I molecules. Hence, our instances are MHC-peptide pairs and the feature vectors comprise an encoding of the peptide and an encoding of the MHC allele, more precisely, of the corresponding gene product. We use the five-dimensional pca encoding [94], which was described in Section 4.2, to encode individual AAs. The nonameric peptides are encoded AA-wise, yielding 45 features. The MHC alleles are encoded pocket-wise. In order to model the physicochemical environment within the pockets, a pocket $P = \{p_1, p_2, \ldots, p_n\}$ is encoded by averaging over the pca encodings of the pocket's residues, i.e.,

$$\Phi^{\mathrm{P}}_{\mathsf{pca}}(P) = \frac{1}{n} \sum_{i=1}^{n} \Phi^{\mathrm{AA}}_{\mathsf{pca}}(p_i) \tag{4.6}$$

where $\Phi^{\mathrm{AA}}_{\mathsf{pca}}(p)$ is the pca encoding of AA p. This yields feature vectors of length 90: five features per peptide AA and five features per pocket. Based on these feature vectors and

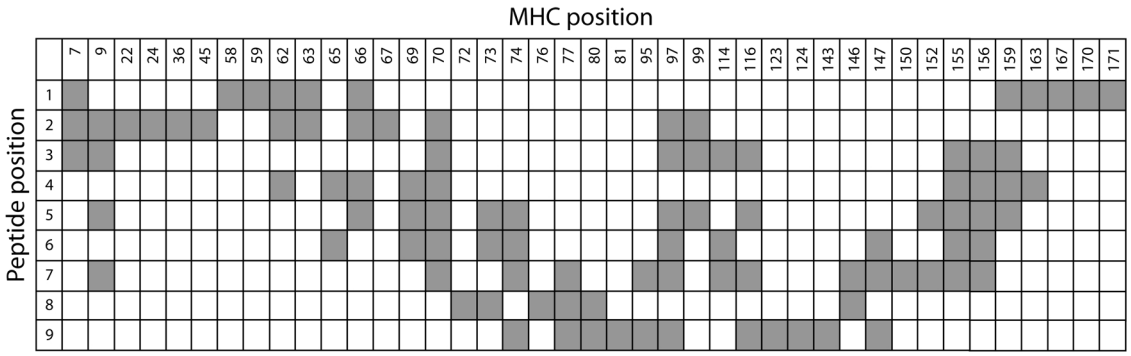


Figure 4.5: Pocket profiles. The columns give the MHC residue numbering. The rows correspond to the peptide positions. Gray squares mark interactions of the respective MHC residue with the respective peptide residue. MHC residues interacting with peptide position 1 are assigned to pocket 1 and so on.

the corresponding pMHC binding affinities an SVR model with a Gaussian RBF kernel is trained.

4.3.3 Experimental Results

UniTope is designed to allow predictions for all known MHC-I alleles whether they were included in the training set or not. MHC alleles which were included in the training set are termed *seen*, those not included are termed *unseen alleles*.

In order to assess the performance of UniTope a comparison to other prediction methods is expedient. Here we are faced with a common problem in comparing prediction methods: a fair comparison between two methods is only given if both methods employed the same data for training and, if considering cross validation results, the same data splits. To facilitate the evaluation of MHC-I binding prediction methods, Peters *et al.* published the IEDB benchmark data set comprising quantitative pMHC-I affinity measurements (IC_{50} values) relating to various human, mouse, macaque, and chimpanzee MHC-I alleles [86]. Along with this data set, the authors established a set of benchmark predictions with three prediction methods: a neural network and two matrix-based approaches. All methods yield quantitative predictions based on allele-specific binding data only. The neural network approach (hereafter ANN) [75] outperforms the matrix-based approaches. Thus, in the following, we will compare the performance of UniTope on seen and unseen alleles to the performance of ANN. Subsequently, other pan-specific prediction methods will be considered.

We employ the IEDB benchmark derived IEDB^{h9} data set, which was introduced in Section 4.2. The IC_{50} values are *log*-transformed into the range $[0, 1]$ according to [75]: $1 - \frac{log(IC_{50})}{log(50{,}000)}$, yielding high scores for low IC_{50} values, i.e., high binding affinities, and low scores for high IC_{50} values, i.e., low binding affinities. As performance measure the Pearson correlation coefficient (PCC, see equation 3.27) is employed.

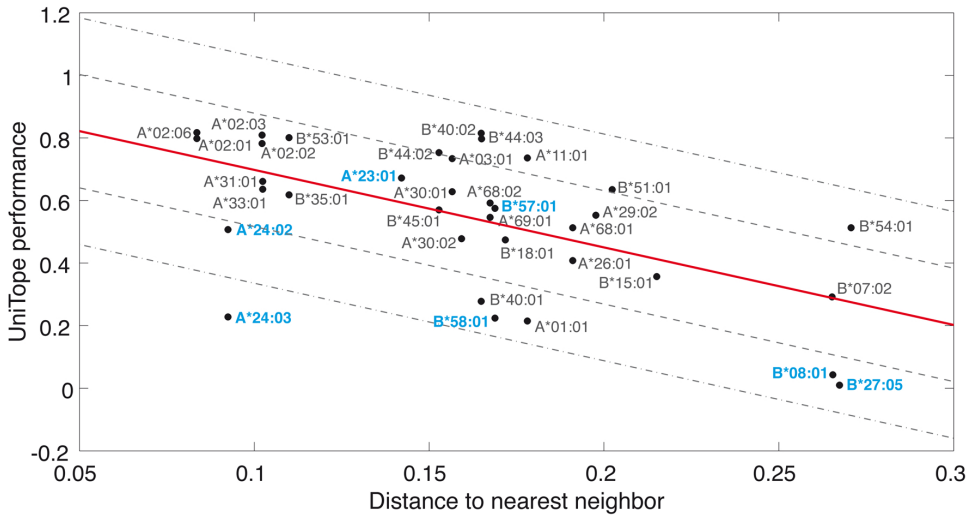


Figure 4.6: Correlation between UniTope performance and distance to nearest neighbor. The PCC achieved by UniTope on unseen alleles on a specific MHC allele is plotted against the allele's distance to its nearest neighbor. The PCC between UniTope performance and distance to nearest neighbor is -0.59. The regression line is plotted as solid red line. One and two standard deviations are plotted as gray dashed and dashed-dotted line, respectively. MHC alleles discussed in the text are shown in blue.

The average PCC yielded by ANN on the IEDB^{h9} data set is 0.54 with a minimum of 0.19 (HLA-B*40:01) and a maximum of 0.83 (HLA-B*18:01). The individual performances are listed in Table B.5 in the appendix.

Performance on Unseen Alleles

Performance on unseen alleles is measured via a *leave-one-out validation*. In leave-one-out validation the data for one allele, the unseen allele, is omitted from training. A model selection is performed on the remaining data and the final model is used for predictions on the unseen allele. In this setting, the performance of UniTope on unknown MHC specificities is simulated.

Leave-one-out validation is performed for each allele represented in the IEDB^{h9} data set. We determine the best parameter combination via five-fold cross validation on the training set. On average, UniTope yields a PCC of 0.55 with a minimum of 0.01 (HLA-B*27:05) and a maximum of 0.82 (HLA-A*02:06). While this definitely leaves room for improvement, it is on par with the performance of the allele-specific ANN. The individual performances of UniTope are listed in Table B.6 in the appendix.

UniTope performs particularly well on HLA-A*02:06 and produces merely random predictions for HLA-B*27:05. What determines whether UniTope will perform well on an allele? In order to analyze this, we consider each allele's distance to its nearest neighbor in the IEDB^{h9} data set. The distance between two alleles a and b is defined as the Euclidean

distance between their pocket encodings:

$$d(a,b) = \left\| \left[\Phi^{\mathrm{P}}_{\mathsf{pca}}(P^1_a), \ldots, \Phi^{\mathrm{P}}_{\mathsf{pca}}(P^9_a)\right] - \left[\Phi^{\mathrm{P}}_{\mathsf{pca}}(P^1_b), \ldots, \Phi^{\mathrm{P}}_{\mathsf{pca}}(P^9_b)\right] \right\|$$

where P^i_x is the i-th pocket of allele x. Figure 4.6 plots these distances against the respective UniTope performance. The PCC between the distance to the nearest neighbor and the performance of UniTope is -0.59. In the following we will examine some of the alleles that deviate strongly from the regression line in Figure 4.6. HLA-B*27:05 (PCC = 0.01) belongs to the B27 supertype [28]. Its binding motif differs from those of the other alleles in the N-terminal anchor position, where it prefers basic residues [28]. Hence, in the leave-one-out validation none of the alleles in the training set is suited as a representative. The same is true for HLA-B*08:01 (PCC = 0.04), which is the only B08 allele in the data set. According to [28], B08 alleles display a unique mode of peptide binding. The remaining alleles' binding information is thus of no avail to predicting HLA-B*08:01 binding affinities. A typical phenomenon of pan-specific predictions is displayed by HLA-B*58:01 (PCC = 0.22) and its nearest neighbor HLA-B*57:01 (PCC = 0.58), both from the same supertype: while the sparsely populated HLA-B*57:01 (59 data points) benefits from the highly populated HLA-B*58:01 (988 data points), the few HLA-B*57:01 data points do not suffice to adequately represent HLA-B*58:01 in the leave-one-out validation. HLA-A*24:03 (PCC = 0.23) is located very close to the related HLA-A*24:02 (PCC = 0.51) in feature space. Both alleles have only few data points in the $\mathsf{IEDB}^{\mathsf{h9}}$ data set, 254 and 197, respectively. This suggests a similar UniTope performance for both alleles in the leave-one-out validation. However, this is not the case: we observe a deviation of more than two standard deviations from the regression line for HLA-A*24:03 and of about one standard deviation for HLA-A*24:02. This discrepancy can be explained by the presence of another allele from the same supertype: HLA-A*23:01 (PCC = 0.67), which is also located nearby. Analysis of the three allele-specific data sets reveals a strong overlap in the data sets of HLA-A*23:01 and HLA-A*24:02: 78 peptides are contained in both data sets. The binding affinities of these peptides with respect to HLA-A*23:01 and HLA-A*24:02 correlate very well (PCC = 0.71). In contrast, the HLA-A*24:03 data set does not overlap with the other data sets.

Performance on Seen Alleles

The performance on seen alleles is measured via a two-times nested five-fold cross validation using the splits specified in the IEDB benchmark data set. UniTope clearly outperforms the allele-specific ANN. It yields an average PCC of 0.67 with a minimum of 0.39 (HLA-B*08:01) and a maximum of 0.84 (HLA-A*02:06, HLA-A*11:01). The individual performances are listed in Table B.5 in the appendix. In this setting, the performance on the special cases HLA-B*27:05 and HLA-B*08:01 is significantly better than in the leave-one-out validation: 0.47 vs 0.01 and 0.39 vs. 0.04, respectively.

Comparison to Pan-Specific Methods

A thorough leave-one-out comparison to the pan-specific approach proposed by DeLuca *et al.* [79] is unfortunately not possible. This approach allows leave-one-out validation only for two of the 35 alleles contained in the IEDB^{h9} data set. All others cannot be fully represented by pocket variants of the remaining alleles. Since a comparison on two alleles would not be significant, we refrained from reimplementing DeLuca's approach to obtain performances on IEDB^{h9}.

In order to perform a comparison with NetMHCpan [20], we obtained predictions for the *seen-allele* and the *unseen-allele* setting from the authors (personal communication with M. Nielsen). On unseen alleles, NetMHCpan yields an average PCC of 0.64 with a minimum of -0.06 (HLA-B*27:05) and a maximum of 0.86 (HLA-A*02:01). In this setting, it outperforms UniTope in 30 out of 35 alleles. On seen alleles, NetMHCpan yields an average PCC of 0.79 with a minimum of 0.60 (HLA-B*08:01) and a maximum of 0.89 (HLA-A*02:01). It outperforms UniTope on all 35 alleles. The individual NetMHCpan performances are listed in Tables B.6 and B.5, respectively.

SVM Computations

All SVM computations were performed using the Python interface of the machine learning toolbox Shogun [89].

4.3.4 Discussion

A key problem in MHC-I binding prediction is the highly polymorphic nature of the MHC-I loci accompanied by a lack of data for the majority of known allelic variants. To address the challenges of vaccine design, predictors for MHC alleles with little or no experimental binding data are needed. We have proposed a pan-specific approach to the prediction of pMHC-I binding affinities. Our approach, UniTope, for the first time allows predictions for all known MHC-I alleles, independent of the availability of experimental binding data. This is in contrast to allele-specific approaches which can only perform predictions for alleles with a sufficient amount of binding data to build a model on. While UniTope outperforms existing allele-specific predictors on alleles included in the training set, it performs on par with these predictors on alleles not included the training set, demonstrating the validity of our approach. Nevertheless, NetMHCpan achieves better performances with a very similar approach.

Pan-specific approaches can also be applied across species. This allows the binding prediction for non-human MHC alleles, where binding data is even scarcer. However, in agreement with [20], our results clearly show that in order for the pan-specific approach to yield satisfying results for a particular allele binding data of a related allele has to be included in the training. It comes as no surprise that the amount of available related data also influences the performance of the pan-specific predictor. The correlation between the allele-specific performance of UniTope and the respective MHC allele's distance to its nearest neighbor renders this distance an adequate measure of confidence for UniTope

predictions. Incorporation of related data set sizes promises to improve the confidence measure.

NetMHCpan yields impressive performances for the majority of the IEDB^{h9} alleles, seen and unseen. It is the best pan-specific MHC-I binding predictor available. What is the secret to the success of NetMHCpan? According to the authors it is the way the data is represented. However, an attempt to measure up to its performance by reimplementing the NetMHCpan approach using SVMs was not successful. Assuming that Nielsen *et al.* thoroughly described their approach and that we correctly reimplemented it, two possible explanations remain: (a) we either chose unfavorable parameter ranges for the grid search and the secret lies in the data representation after all or (b) artificial neural networks are simply better suited for the pan-specific modelling of pMHC-I binding. A more detailed analysis would be required to come to a well-founded conclusion regarding the secret to the success of NetMHCpan. From a problem-oriented point of view, however, it is far more expedient to focus on optimally exploiting the superior performances of NetMHCpan and its allele-specific counterpart NetMHC [75] for EV design.

4.4 T-Cell Epitope Prediction

4.4.1 Introduction

So far, we have only considered pMHC binding. However, induction of an immune response requires recognition of a pMHC complex by a host T cell. Complex dependencies and an incomplete biological knowledge make the prediction of T-cell epitopes a very challenging problem. The more recent machine-learning based approaches employ a sparse encoding [21], a physicochemical encoding (POPI by Tung *et al.* [22]) or a WD kernel (POPISK by Tung *et al.*, unpublished work) to train SVMs on peptide sequences. Common to all of the approaches is their limited prediction accuracy. Reasons for this can be found in the choice of training data and in the methods' limited view on the problem.

Taking a wider view on T-cell reactivity, we propose to incorporate knowledge on the immune system's self-tolerance into the prediction: Selection processes in the thymus and in the periphery eliminate T cells capable of reacting against self-peptides. Thus, antigenic peptides that are very similar to self-peptides are highly unlikely to induce a T-cell response. We present an approach to T-cell epitope prediction that combines sequence information and information on self-tolerance. Based on a carefully designed data set comprising T-cell reactivity data in the context of HLA-B*35:01, we can show that our predictor outperforms purely sequence-based predictors, demonstrating the validity of our approach.

4.4.2 Modelling Self-Tolerance

Self-tolerance describes the capability of the immune system to distinguish self from non-self. A key mechanism for ensuring self-tolerance is negative T-cell selection. Negative T-cell selection eliminates or inactivates self-reactive T cells, rendering antigenic peptides that are very similar to self-peptides highly unlikely to induce a T-cell response. Thus, in order to model self-tolerance, we require a representative set of self-peptides presented to

T cells during negative selection. Since only MHC binding peptides are presented, MHC binding affinity should be taken into account. Furthermore, an adequate similarity measure for peptides is needed. For the sake of convenience, we will use a distance measure instead of a similarity measure.

Reference Proteome

Since negative T-cell selection takes place in the thymus, the thymus proteome represents a reasonable reference set for central tolerance. However, this set is too small to also model peripheral tolerance: in this case the complete host proteome has to be considered.

We generate peptide reference sets based on the thymus proteome, modelling central tolerance, as well as on the human proteome, modelling both central and peripheral tolerance. Only peptides binding to the MHC molecule under consideration can be employed for T-cell selection. Hence, only peptides predicted to be MHC binders ($IC_{50} \leq 500$) by the state-of-the-art MHC-I binding predictor NetMHC [14, 75] (version 3.0) are included in the peptide reference sets. Since the majority of all known MHC-I ligands are of length nine, we will only consider ninemers in the following.

Human proteome. The general human proteome was retrieved from the International Protein Index (hereafter IPI) [95] (version 3.47).

Thymus proteome. We used gene expression data to define the thymus proteome. Whole genome microarray data was downloaded from the NCBI Gene Expression Omnibus [96] and from the EBI ArrayExpress database [97]. The employed microarrays only cover about 43% of all proteins present in the IPI human proteome. Thus, we cannot make a statement regarding the expression in the thymus of the remaining 57%. Due to conflicting measurements regarding the presence of proteins in the thymus, we employ a majority voting to unambiguously assign each protein to one of the three classes: present, marginally expressed, and absent. For details on the employed data sets as well as on the majority voting, please refer to Tables B.7 and B.8 and to Algorithm B.1 in the appendix. According to the majority voting, approximately 45% of the covered proteins are present in the thymus, 26% are marginally expressed, and 28% are absent from the thymus. Given these three groups of proteins, we define a *minimum thymus proteome* (hereafter thymus-min) consisting of all proteins in the present group and a *maximum thymus proteome* (hereafter thymus-max) comprising present and marginally expressed proteins.

Binding Affinities

Not every self-peptide contributes to self-tolerance. Crucial features are pMHC binding as well as, according to the affinity model, pMHC:TCR affinity. We account for the aspect of pMHC binding by including only self-peptides predicted to bind to the respective MHC molecule in the reference peptide set. Regarding the pMHC:TCR affinity, there is no accurate prediction method available. However, the stability of the ternary complex pMHC:TCR is affected by pMHC affinity, which can be predicted accurately. Thus, we

regard pMHC binding as an estimate of pMHC:TCR affinity and incorporate it explicitly into our model of self-tolerance. The reported correlation between pMHC affinity and peptide immunogenicity [9] additionally justifies this approach.

Measuring Distance to Self

We define the distance of a peptide to a set of peptides to correspond to the smallest pairwise distance to one of the peptides in the set. The distance between two peptides of equal length can be calculated in various ways, ranging from a simple Hamming distance-based approach to substitution matrix-based distance measures. Within this work, we employ a distance measure that has been derived from the BLOSUM45 substitution matrix [88]. It corresponds to a sum of AA distances and can thus be represented by a symmetric 20×20-matrix. The matrix is generated as follows: The substitution matrix A is turned into a symmetric matrix A' by replacing each entry a_{ij} by $\frac{a_{ij}+a_{ji}}{2}$. The entries are shifted by adding the absolute value of the smallest entry, such that the resulting matrix A'' contains no negative entries: $a''_{ij} = a'_{ij} + |\min(A')|$. Subsequently, the matrix entries are normalized via division by the maximum entry in A'' yielding the matrix A''': $a'''_{ij} = \frac{a''_{ij}}{\max(A'')}$. In a last step, the distance matrix M is obtained by subtracting the entries of A''' from 1: $m_{ij} = 1 - a'''_{ij}$.

Given this distance matrix, the distance of a target peptide to the reference set can be determined. However, linearly comparing a peptide to a set of several hundreds of thousands of peptides is computationally expensive. We therefore use a memory-efficient trie-based approach [98]. Each self-peptide is represented by a leaf of the trie. Peptides descending from the same node within a trie have a common prefix. This representation of the reference peptide set allows us to prune branches containing peptides that are too distant to be of interest. Determining the self-peptide in the IPI-based peptide reference set (861,352 HLA-B*35:01-binding ninemers) that is closest to a given target peptide takes less than a second.

Feature Encoding

Each peptide is encoded based on a set of the $k >> 0$ most similar self-peptides from the respective reference proteome. The rationale behind considering several similar peptides instead of the single most similar peptide is that it allows to consider (a) high-affinity self-peptides that are sufficiently similar but not the most similar, and (b) similarity distributions among the closest self-peptides.

For each target peptide p^*, the distances to the k nearest self-peptides $p_1, \ldots, p_k$ are determined. Distances are in the range $[0, 1]$ with 0 meaning that the target peptide is identical to a self-peptide and 1 being the maximum possible distance with respect to the distance measure. Additionally, we determine the binding affinities of the target and the k self-peptides using NetMHC. Binding affinities are also in the range $[0, 1]$ with 0 meaning low affinity, i.e., non-binder, and 1 meaning very high affinity, i.e., strong binder. Let $d(p)$ be the distance of peptide p to p^* and $b(p)$ the binding affinity of peptide p to the MHC molecule corresponding to the allele under consideration. The self-tolerance feature vector

is generated by concatenating the binding affinity of the target peptide, the distances to the k nearest self-peptides as well as the respective binding affinities, yielding the following feature vector:

$$\Phi(p^*) = \big[b(p^*), d(p_1), \ldots, d(p_k), b(p_1), \ldots, b(p_k)\big].$$

We choose k to be 100 resulting in a 201-dimensional feature vector.

4.4.3 Data

Due to the complexity of the problem at hand we impose strong restrictions on the data set used in this study: we require all data points to be generated in one laboratory and with the same experimental approach. Furthermore, the data set has to contain a sufficient amount of positive and negative examples. Such data sets are very difficult to obtain from publicly available sources like the IEDB [87]. Thus, we base this study on a single small, non-public data set.

The group of Stefan Stevanović (Department of Immunology, University of Tübingen) kindly provided us with experimental T-cell reactivity data. A set of peptides derived from Epstein-Barr virus (EBV) antigens that were either predicted to bind to HLA-B*35:01 by SYFPEITHI [12] or selected by an expert were analyzed for T-cell reactivity using ELISPOT assays. The data set (hereafter $\mathcal{D}$) comprises 151 nonameric peptides: 49 immunogenic (positive examples, $\mathcal{D}_+$), 102 non-immunogenic (negative examples, $\mathcal{D}_-$).

In order to preclude the learning of MHC binding instead of T-cell reactivity, a subset of $\mathcal{D}$ has to be selected for training, such that the distribution of HLA-B*35:01 binding affinities in the positive examples is equivalent to that in the negative examples. Since experimental MHC binding data is not available, we employ NetMHC scores as binding affinities. The scores range from 0.07 to 0.85 with an average of 0.56 and 0.39 for peptides in $\mathcal{D}_+$ and $\mathcal{D}_-$, respectively. On the basis of these scores, we divide the peptides into eight bins of equal width (Figure 4.7A). Our aim is to adapt the binding affinity distributions, i.e., adjust the numbers of immunogenic and non-immunogenic peptides per bin, while keeping as many of the scarce positive examples as possible. From all bins containing more negative than positive examples, we select all positives and the same number of negative examples, preferably high-affinity. From all bins containing more positive than negative examples, we select all negatives and the same number of positive examples, preferably low-affinity. If there are left-over negatives from the previous bin, we keep one more low-affinity positive example and add an additional high-affinity negative example to the previous bin. The resulting set $\tilde{\mathcal{D}}$ contains 45 immunogenic ($\tilde{\mathcal{D}}_+$) and 45 non-immunogenic ($\tilde{\mathcal{D}}_-$) examples (Figure 4.7B). The average NetMHC score of peptides in $\tilde{\mathcal{D}}_+$ and $\tilde{\mathcal{D}}_-$ is 0.54 and 0.55, respectively. A two-sample t-test could not identify a significant difference between the NetMHC scores of these two sets ($p = 0.862$).

4.4.4 Experimental Results

Our aim is to assess the benefit of incorporating self-tolerance information into T-cell epitope prediction. In order to do so, we need to train a sequence-based predictor on $\tilde{\mathcal{D}}$ and subsequently extend this predictor to include our model of self-tolerance. All auROCs

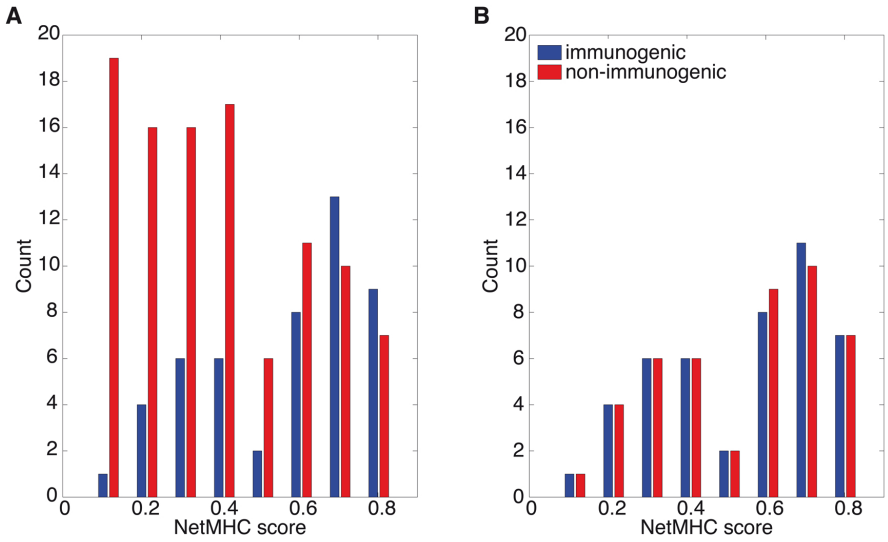


Figure 4.7: Data set composition. The distribution of NetMHC scores in the immunogenic and non-immunogenic examples of (A) the original set of EBV ninemers, $\mathcal{D}$, and (B) the subset selected for training, $\tilde{\mathcal{D}}$, is displayed.

reported within this subsection are averaged over 100 runs of two-times nested five-fold cross validation to reduce random fluctuations of the performance.

Sequence-Based Predictions

For the prediction of T-cell epitopes based on sequence information only, we use SVC with a Gaussian RBF kernel on blosum50-encoded peptide sequences. The mean auROC is 0.72. As a comparison, an approach based on POPI, i.e., SVC with a Gaussian RBF kernel on physicochemically encoded peptide sequences, achieves a mean auROC of 0.69. Employment of a WD kernel as in POPISK yields a mean auROC of 0.70.

Incorporation of Self-Tolerance

Incorporation of self-tolerance is achieved by adding sequence-based and self-tolerance-based kernels. The self-tolerance-based kernel is a Gaussian RBF kernel on the 201-dimensional feature vectors encoding self-tolerance information with respect to one of the three reference proteomes: thymus-min, thymus-max or IPI. While incorporation of the thymus-max-based self-tolerance information resulted in a considerable improvement (auROC = 0.78), this was not the case for the IPI- and thymus-min-based self-tolerance information (auROCs of 0.70 and 0.71, respectively). This is consistent with the performance of purely self-tolerance-based predictors: the thymus-max-based predictor (auROC = 0.58) clearly outperforms the IPI- (auROC = 0.48) and thymus-min-based (auROC = 0.44) predictors.

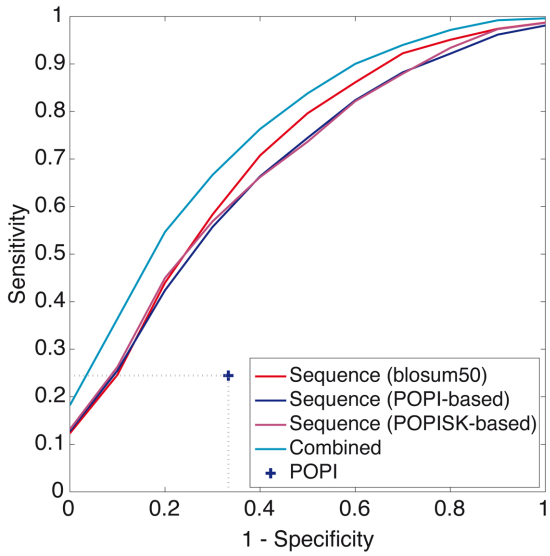


Figure 4.8: Performances of sequence-based and combined T-cell epitope prediction models. The mean ROC curve over 100 runs of two-times nested five-fold cross validation is displayed for three sequence-based predictors and for a predictor combining sequence and self-tolerance information. The sequence-based predictors are (a) blosum50: a Gaussian RBF kernel with blosum50 encoding, (b) POPI-based: a Gaussian RBF kernel with the physicochemical encoding proposed in [22], and (c) POPISK-based: a WD kernel. The combined predictor is based on a blosum50-encoding of the peptide sequences and a thymus-max-based self-tolerance model. Additionally, the average performance of POPI [22, 99] is given (+).

Comparison to Previously Proposed Approaches

We compare our combined model to the non-allele-specific predictor POPI as well as to HLA-B*35:01-specific reimplementations of the POPI and POPISK approaches. A comparison to the prediction method proposed in [21] is omitted due to the user-unfriendly web server. Since this method also disregards MHC allele information, it can safely be assumed not to perform significantly better than POPI. The performances of the individual methods are displayed in Figure 4.8.

We retrieved POPI predictions for the peptides in our data set from the POPI web server [99]. POPI assigns a peptide to one of four immunogenicity classes: none, little, moderate and high. In order to compare the performance of POPI to our prediction models, the four classes need to be mapped to the two classes immunogenic and non-immunogenic. We consider all classes except for none as immunogenic. Using this mapping and the same splits as above, we determine the mean sensitivity and specificity of POPI. From Figure 4.8 it can clearly be seen that our allele-specific T-cell epitope predictors significantly outperform the non-specific POPI. The negative effect of disregarding allele information when training immunogenicity predictors becomes obvious, especially when considering the performance of our allele-specific reimplementation of POPI.

Since POPISK is a predictor for T-cell epitopes in the context of the HLA-A2 supertype,

a direct comparison is not possible. However, the performance of the WD-kernel-based predictor should provide a reasonable estimate. It yields a mean auROC of 0.70, which is well below the performance of our combined model.

4.4.5 Discussion

The potency of an EV strongly depends on the capability of the included peptides to induce an immune response. The development of prediction methods capable of accurately predicting T-cell epitopes is thus of major interest to immunologists and the pharmaceutical industry. Whether a particular peptide is a T-cell epitope or not depends on the availability of an MHC molecule that presents the peptide on the cell surface and on the availability of a suitable TCR for the specific pMHC complex. The latter is determined by the host's proteome. These complex dependencies and the incomplete biological knowledge, render the prediction of T-cell epitopes particularly challenging. The accuracy of current prediction methods is too low to permit their use in most biomedical applications. Reasons for this can be found in the choice of training data and in the methods' limited view on T-cell reactivity.

Previously proposed methods employ peptide sequence information only. We propose to move from simple sequence-based predictors to predictors that take relevant system-wide properties like the self-tolerance of the immune system into account. Using a carefully designed set of EBV peptides provided by the group of Stefan Stevanović (Department of Immunology, University of Tübingen) we could show that the incorporation of a model of self-tolerance considerably improves T-cell epitope prediction.

Our model of self-tolerance is based on the similarity of the target peptides to a set of self-peptides derived from a representative reference proteome. Three different reference proteomes were considered: thymus-min and thymus-max for modelling central tolerance and IPI for modelling central as well as peripheral tolerance. Employment of the IPI-based model for T-cell epitope prediction does not yield an improvement over sequence-based predictions. However, employment of a model of central tolerance via the non-conservatively defined thymus proteome, thymus-max, does. This indicates that our model of self-tolerance represents central tolerance more accurately than peripheral tolerance, despite the fact that we use a probably incomplete thymus proteome. Mechanisms of peripheral tolerance are responsible for preventing self-reactive T cells in the periphery from inducing an immune response. Such T cells have escaped negative selection in the thymus either because (a) they display a low avidity for self-antigens presented in the thymus [100] or because (b) their cognate self-antigen is not expressed in the thymus [32]. The first case applies to proteins from the thymus proteome. This aspect of peripheral tolerance could thus be covered by our model of central tolerance: peptides similar to self-peptides expressed in the thymus are unlikely to be immunogenic. The second case applies to tissue-specific proteins not presented in the thymus. Naive T cells circulate between blood and secondary lymphoid organs. Thus, the encounter of a naive T cell with an APC presenting peptides derived from tissue-specific proteins is highly unlikely [101]. Consequently, naive T cells reactive to tissue-specific proteins are not necessarily deactivated by peripheral

tolerance mechanisms. The assumption that peptides similar to self-peptides in general are not likely to be immunogenic does not apply, rendering our model of self-tolerance inappropriate to represent peripheral tolerance.

It has to be noted that this study is based on a single small data set comprising immunogenicity data with respect to one virus and one MHC allele. Performing multiple runs of two-times nested five-fold cross validation ensures the generation of statistically significant results. However, more – and preferably larger – data sets for other alleles are needed to evaluate the general validity of our approach, including the model of self-tolerance, the feature encoding and the choice of distance measure. The amount of immunogenicity data available from the IEDB [87] is continuously increasing. However, due to the variance caused by the use of various experimental approaches in various laboratories, the vast majority of this data does not fulfill the strict requirements we impose. Thus, as of now, a thorough, statistically correct study on the small but consistent EBV data set is the best we can do.

To the best of our knowledge this is the first study predicting T-cell epitopes incorporating system-wide properties. Despite the small sample size, the work presented here demonstrates a proof-of-concept for the incorporation of a model of self-tolerance for T-cell epitope prediction.

Chapter 5

Epitope Selection

The previous chapter was concerned with the discovery of candidate epitopes from a given set of target antigens. In the following, we will concentrate on the subsequent step in the EV design pipeline: epitope selection (Figure 5.1). Given a set of candidate epitopes, the task is to select the set of epitopes which yields the best immune response in the target population. Parts of this chapter have previously been published [8, 102, 103].

5.1 Introduction

Epitope selection is a critical step in the design of an EV. Regulatory and economic issues impose a strong limitation on the number of epitopes that can be included in an EV. From a set of target antigens, candidate epitopes can be determined either experimentally or, as proposed in the previous chapter, computationally. The crucial task is to select a set of epitopes from these candidate epitopes that yields the best immune response in the target population while at the same time keeping the number of epitopes low. The selection is usually made on a case-by-case basis considering properties such as overall immunogenicity, mutation tolerance, population coverage, coverage of antigens, antigen processing as well as properties of the source antigen. The selection methods used by the pharmaceutical industry are manifold.

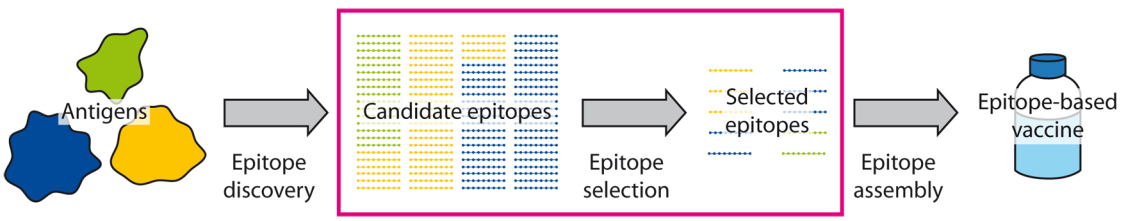


Figure 5.1: EV design: epitope selection step. Out of the set of candidate epitopes determined via epitope discovery, the most suitable subset for use in the EV has to be selected. Figure based on [8].

Given the set of candidate peptides, the relevant attributes of each candidate can be determined *in silico*. However, the final choice of the set of epitopes to be used in the vaccine is typically performed manually. Several groups have addressed this problem computationally. In 2005, De Groot *et al.* [23] published an approach to designing highly immunogenic and conserved epitopes to be used in EVs. The authors use EpiMatrix [104] to estimate the MHC-II binding affinity of highly conserved ninemers from HIV-1 proteins. Peptides with high binding affinities are then used for the construction of extended peptides containing multiple overlapping ninemers. *In vitro* evaluation of the immunogenicity of a selected set of these extended peptides yielded promising results.

In 2007, Vider-Shalit *et al.* [24] proposed using a genetic algorithm to design an ordered sequence of epitopes to be used in an EV. Information on peptide conservation and identity to self-peptides is used to pre-filter the set of candidates, while information on MHC-I allele probabilities in the target population is used to select alleles of interest. The scoring function employed for the heuristic takes into account the number of covered MHC-I alleles, the number of covered antigens, the number of covered MHC/antigen combinations, and the probability of each epitope to be properly cleaved in the sequence.

Common to manual selection and the computational approaches above is the fact that the solutions are not necessarily optimal. None of the approaches can guarantee that there is not a better vaccine possible from the given set of epitopes.

We propose an ILP-based mathematical framework to efficiently determine an optimal set of vaccine epitopes. Our algorithm outperforms previously proposed strategies and has runtimes on the order of seconds for typical problem sizes. Moreover, it is highly flexible and can be easily adapted to various requirements.

5.2 Approach

In order to find an optimal set of epitopes, we first have to define what characterizes a *good* vaccine or, correspondingly, a *good* set of epitopes. This issue is highly controversial in the literature and only large-scale data from vaccination trials will provide sufficient data to validate the different approaches retrospectively. With this in mind, we do not suggest one optimal epitope selection strategy, but instead suggest a mathematical framework that allows working with various definitions of the term 'good vaccine'. For a chosen definition, however, the algorithm will yield a combination of epitopes that is provably optimal.

In the following, a 'reasonable' definition of a good vaccine will be introduced. This will allow us to present the mathematical formulation and to illustrate how immunological requirements can be translated into mathematical constraints. For specific applications, the requirements and constraints may of course deviate from those given. For example, sequence variation in an antigen would be much more important for an HIV vaccine than for a cancer vaccine. The framework is flexible enough to allow for such different requirements, as will be illustrated in a vaccine design study in Chapter 7.

A good vaccine displays a high overall immunogenicity, which means it is capable of inducing potent immunity in a large fraction of the target population. This simple definition forms the basis of our mathematical formulation, which aims at maximizing overall

immunogenicity of the selected epitopes. We extend this definition by additionally requiring a high mutation tolerance as well as a certain degree of allele and antigen coverage. Furthermore, the selected epitopes should display a high probability of being produced by the antigen processing pathway. We thus obtain a brief list of basic requirements:

Mutation tolerance. It has been shown that the change of a single residue can turn an MHC binding peptide into a non-binding and an immunogenic into a non-immunogenic peptide [105]. Hence, mutations within the targeted antigen regions can lead to an escape from immune response and thereby impair the effectiveness of the vaccine. High genetic variability as observed in, e.g., the HIV, the hepatitis C virus (HCV), and the influenza virus (IV) can thus affect protection by a vaccine. Selection of highly conserved epitopes reduces the chance of viral immune escape. Additionally, the effect of a single mutation on an EV can be limited by preferentially selecting non-overlapping epitopes.

Allele coverage. An MHC allele is said to be covered by a set of epitopes if at least one of the epitopes is capable of inducing a T-cell response when bound to the corresponding MHC molecule. Within a population MHC alleles occur with different frequencies. Hence, requiring a certain number of alleles to be covered is equivalent to requiring a certain degree of population coverage.

Antigen coverage. Depending on the developmental stage, the expression frequencies of viral proteins differ. Selecting epitopes from a wide variety of antigens, i.e., providing high antigen coverage, increases the chance of detecting a virus at any developmental stage.

Antigen processing. Before a peptide is presented by an MHC-I molecule on the cell surface, it passes through an antigen processing pathway, which includes proteasomal cleavage and TAP transport. Knowledge of these steps' specificities allows exclusion of peptides which are unlikely to ever be presented to a T cell.

From all possible epitope combinations of a given size satisfying these requirements, the ones with a maximum overall immunogenicity will be called 'optimal'. The search for an optimal epitope set for an EV can be interpreted as an optimization problem: out of a given set of epitopes, choose a subset which, out of all subsets meeting the above-named requirements, displays maximum overall immunogenicity. Due to regulatory, economic, and practical concerns the size of such a subset is usually kept rather small.

5.3 Mathematical Abstraction

The overall immunogenicity of an EV depends on the immunogenicity of the vaccine epitopes in the target population, i.e., the immunogenicity of the epitopes with respect to the corresponding MHC alleles. Given a set of epitopes and a set of MHC alleles we make the following assumption: the immunogenicity of all epitopes with respect to all alleles corresponds to the sum of the immunogenicities of every single epitope with respect to

each of the alleles. Furthermore, the contribution of an allele directly depends on its probability of occurring within the target population. More common alleles are weighted more than uncommon ones. Thus, overall immunogenicity $\mathcal{I}$ can be derived mathematically as a weighted sum over immunogenicities of epitopes E with respect to the given MHC alleles A:

$$\mathcal{I}(E, A) = \sum_{e \in E} \sum_{a \in A} p_{mhc}(a) \cdot i(e, a) \tag{5.1}$$

where $p_{mhc}(a)$ corresponds to the probability of allele a in the target population and $i(e, a)$ to a measure of the immunogenicity of epitope e when bound to the gene product of allele a.

5.3.1 ILP Formulation

Our goal is to maximize overall immunogenicity while constraining the possible solutions to sets of peptides which satisfy the above-mentioned requirements for a good vaccine. This problem can be conveniently formulated as an ILP. Solving the ILP will yield an optimal solution according to our definition of an optimal epitope set. For the sake of clarity, we start out with the very basic definition of an optimal epitope set. In the next step the resulting ILP will be extended to represent the more refined definition.

The set of candidate epitopes is E. Each epitope $e \in E$ is associated with a binary decision variable $\mathsf{x_e}$, where $\mathsf{x_e} = 1$, if the respective epitope belongs to the optimal set, and $\mathsf{x_e} = 0$ otherwise. This ILP maximizes overall immunogenicity (see ILP 5.1). The only constraint is the number of epitopes to select.

ILP 5.1: ILP corresponding to the basic definition of an optimal epitope set.

$$\begin{aligned} &\text{maximize} && \textstyle\sum_{e \in E} \mathsf{x_e} \sum_{a \in A} p_{mhc}(a)\ i(e, a) \\ &\text{subject to} && \textstyle\sum_{e \in E} \mathsf{x_e} = k \qquad (5.1a) \end{aligned}$$

DEFINITIONS

A	Set of observed MHC alleles
E	Set of candidate epitopes

PARAMETERS

$i(e, a)$	Immunogenicity of epitope e with respect to allele a
k	Number of epitopes to select
$p_{mhc}(a)$	Probability of MHC allele a occurring in the target population

VARIABLES

$\mathsf{x_e} = 1$	if epitope e belongs to the optimal set, otherwise $\mathsf{x_e} = 0$

We will now extend this basic ILP to represent a more refined definition of a good epitope set. In order to implement the additional requirements we introduce another set of binary decision variables: each MHC allele a is associated with a variable $\mathsf{y_a}$. If allele a is covered, meaning that an epitope which is sufficiently immunogenic with respect to the gene product of allele a belongs to the optimal set, $\mathsf{y_a} = 1$, otherwise $\mathsf{y_a} = 0$. The extended ILP accounts

for mutation tolerance, as well as for allele and antigen coverage (see ILP 5.2). Mutation tolerance is obtained by constraints 5.2b and 5.2c: (5.2b) guarantees that only epitopes with a certain degree of conservation are selected, (5.2c) prevents selection of overlapping epitopes. (5.2d) and (5.2e) ensure a minimum allele coverage: (5.2d) guarantees that only covered alleles will be considered as covered, i.e., $\mathsf{y_a} = 1$, while (5.2e) enforces the allele coverage threshold τ_{mhc}. Coverage of every antigen by at least τ_a epitopes is ensured by (5.2f). Additionally, (5.2g) prevents the selection of peptides which are unlikely to result from antigen processing.

ILP 5.2: ILP corresponding to the extended definition of an optimal epitope set.

$$
\begin{array}{lll}
\text{maximize} & \sum_{e\in E} \mathsf{x_e} \sum_{a\in A} p(a)\ i(e,a) & \\
\text{subject to} & \sum_{e\in E} \mathsf{x_e} = k & \text{(5.2a)} \\
\forall e \in E: & \mathsf{x_e}\ \tau_C \le c(e) & \text{(5.2b)} \\
\forall (p,r) \in O: & \mathsf{x_p} + \mathsf{x_r} \le 1 & \text{(5.2c)} \\
\forall a \in A: & \sum_{e\in I_a} \mathsf{x_e} \ge \mathsf{y_a} & \text{(5.2d)} \\
 & \sum_{a\in A} \mathsf{y_a} \ge \tau_{mhc} & \text{(5.2e)} \\
\forall i \in \{1,\ldots,n\}: & \sum_{e\in E_i \cap I} \mathsf{x_e} \ge \tau_a & \text{(5.2f)} \\
\forall e \in E: & \mathsf{x_e}\ \tau_{ap} \le p_{ap}(e) & \text{(5.2g)}
\end{array}
$$

Definitions

A	Set of observed MHC alleles
E_i	Set of epitopes from the i-th antigen
E	Set of all candidate epitopes ($E = E_1 \cup \ldots \cup E_n$)
I_a	Set of epitopes which, when bound to the gene product of an MHC allele a, display an immunogenicity greater than or equal to a given threshold
I	Set of all sufficiently immunogenic epitopes ($I = \bigcup_{a\in A} I_a$)
O	Set of overlapping pairs of epitopes

Parameters

$c(e)$	Conservation of epitope e
$i(e,a)$	Immunogenicity of epitope e with respect to allele a
k	Number of epitopes to select
$p(a)$	Probability of MHC allele a occurring in the target population
$p_{ap}(e)$	Probability that epitope e will be produced during antigen processing
τ_a	Minimum number of epitopes from each antigen to be included
τ_{ap}	Antigen processing threshold
τ_c	Conservation threshold
τ_{mhc}	Minimum number of MHC alleles to be covered

Variables

$\mathsf{x_e} = 1$	if epitope e belongs to the optimal set, otherwise $\mathsf{x_e} = 0$
$\mathsf{y_a} = 1$	if allele a is covered by the optimal set, otherwise $\mathsf{y_a} = 0$

It might be desirable to obtain several optimal or nearly optimal epitope sets. As proposed in [106], suboptimal solutions can be obtained by adding the constraints given in (5.2), where S_j represents the optimal set of epitopes found in iteration j and s represents the desired number of epitope sets.

$$\sum_{u \in S_j} \mathsf{x_u} \leq k - q \quad \text{for } j = 1 \ldots s-1 \tag{5.2}$$

The ILP has to be solved iteratively s times. After each iteration, the ILP for the next iteration, $j+1$, is created by adding the corresponding constraint to the ILP of iteration j. Every resulting epitope set differs from all other solutions in at least q peptides, $1 \leq q \leq k$.

5.3.2 Non-Linear Requirements

In order to incorporate a requirement into the ILP framework it must be formulated as a linear constraint. There are, however, reasonable requirements for good vaccines which are non-linear. Two examples of such requirements will be discussed below.

Example 1: Population coverage. A major interest in vaccine design is population coverage: For what fraction of a target population will the resulting EV be effective? In theory this corresponds to the probability of an individual in the population carrying at least one MHC allele covered by the epitopes in the EV. Given a set of MHC alleles A as before and their distribution within a population, the population coverage of a particular set of epitopes can be computed. For this computation the polygeny of the MHC has to be taken into account. It is $A = A_1 \cup \ldots \cup A_m$ with A_i being the set of alleles at locus i. Let $p_\ell(a)$ be the probability of an allele a occurring at the corresponding MHC locus. Then, disregarding linkage disequilibrium, the probability of an individual in the population carrying an allele from the set A_i at locus i corresponds to

$$\tilde{p}_\ell(A_i) = 1 - (1 - \sum_{a \in A_i} p_\ell(a))^2. \tag{5.3}$$

Let $\mathsf{y_a}$ be as described above. It follows that the probability of an individual carrying at least one MHC allele covered by the epitopes in E, and thus the population coverage of E given A, is

$$\kappa(E, A) = 1 - \prod_{i=1}^{m} (1 - \sum_{a \in A_i} \mathsf{y_a} p_\ell(a))^2. \tag{5.4}$$

Example 2: Average number of active epitopes per individual. Population coverage of an epitope set states what fraction of a population carries an MHC allele associated with one of the epitopes. It does not give any information on the number of active epitopes per individual. The number of epitopes within a set which are active for a specific individual depends on the individual's MHC genotype. Given the haploidic probabilities of MHC alleles within a population the probability of an MHC genotype can be calculated. Alleles not included in the set A are accounted for by adding a representative allele x to each locus. The frequency of the representative at locus i results from $p_\ell(x_i) = 1 - \sum_{a \in A_i} p_\ell(a)$.

Let G be the set of genotypes within the population of interest and $p(g)$ the probability of genotype g. Furthermore, let $\eta(E, g)$ be the number of epitopes in an epitope set E which are immunogenic with respect to an MHC allele in g. The average number of active epitopes per individual in the population results from

$$\psi(E) = \sum_{g \in G} p(g)\ \eta(E, g). \tag{5.5}$$

These non-linear requirements cannot be incorporated into the ILP directly. It is, however, possible to search a sufficiently large set of optimal and suboptimal solutions for the best set of epitopes that displays the required properties. Furthermore, for some non-linear requirements, linear substitute requirements may be found: e.g., requiring a certain degree of population coverage is equivalent to requiring a certain degree of allele coverage.

It has to be noted that, while ILPs require constraints to be linear, the non-linearity of a constraint does not necessarily imply that the corresponding optimization problem cannot be solved efficiently. The solver CPLEX [53], for example, can handle integer programs with quadratic constraints. Furthermore, heuristics can be employed to determine near-optimal solutions to such problems. For more information on optimally solving integer non-linear programs please refer to [107, 108].

5.4 Experimental Results

In order to show the effectiveness of the proposed approach, we compare our optimal strategy (best overall immunogenicity, BOI) with two simple approaches:

- randomly select k peptides out of a pool of good epitopes (random set of epitopes, RSE) and
- a simple greedy approach: pick the k best epitopes from the set (best epitope-wise immunogenicity, BEI).

As performance measures the theoretical gain in overall immunogenicity as well as the number of epitopes required to achieve a similar overall immunogenicity are employed.

The three epitope selection strategies were used to select different-sized sets of peptides from a set of 4,461 HCV ninemers. (The data set is described in more detail in Section 7.2.) For BOI, the basic ILP 5.1 was used to maximize overall immunogenicity. BEI selects the epitopes with the highest sum of immunogenicities irrespective of the probabilities of the corresponding MHC alleles.

The overall immunogenicity of each epitope set was determined and is displayed in Figure 5.2. For RSE, mean and standard deviation of 100 random selections of different-sized epitope sets from the 100 most immunogenic peptides are shown. The BEI curve shows sudden increases in overall immunogenicity from 0 to 1, 10 to 13, and from 20 to 21 epitopes. This is caused by the selection of epitopes that are highly immunogenic with respect to HLA-A*02:01, which is the most common among the considered alleles ($p_\ell = 0.145$). All other selected epitopes are highly immunogenic with respect to less

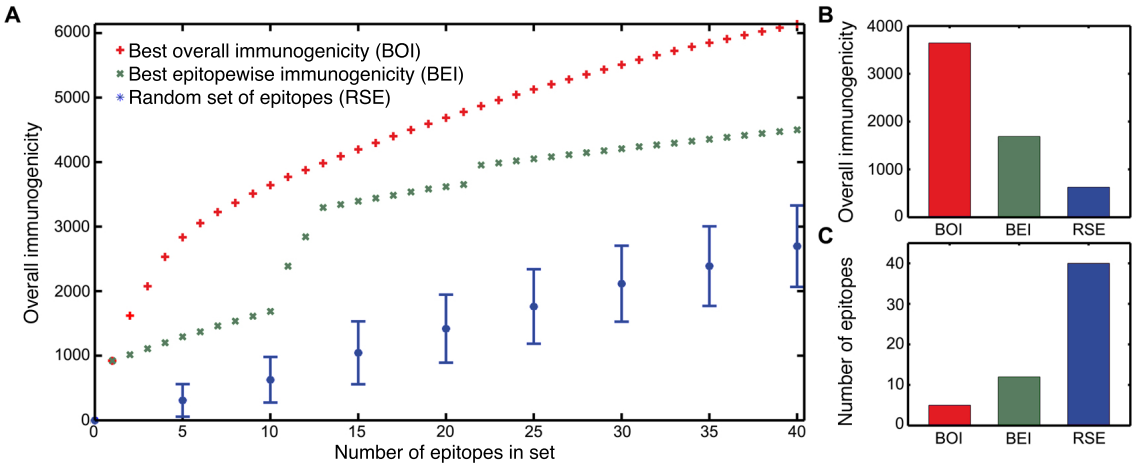


Figure 5.2: Comparison of different epitope selection strategies. Best overall immunogenicity (BOI), best epitope-wise immunogenicity (BEI) and random selection (RSE) are compared with respect to overall immunogenicity. A) Overall immunogenicity of different-sized epitope sets. B) Overall immunogenicity of a set of 10 epitopes. C) Number of epitopes required to achieve an overall immunogenicity of at least 2,699.

common alleles like HLA-B*27:05 ($p_\ell = 0.015$) or HLA-B*51:02 ($p_\ell = 0.003$). Thus, the former contribute more extensively to the overall immunogenicity than the latter.

The average overall immunogenicity of the randomly chosen epitope sets is rather low: scores range from about 308 for five epitopes, to 1,763 for 25 epitopes, to 2,699 for 40 epitopes. The other two approaches start from a minimum overall immunogenicity of more than 900 and reach immunogenicities of 4,502 (BEI) and 6,142 (BOI), respectively. To achieve an immunogenicity of at least 2,699, BOI requires five and BEI 12 epitopes (Figure 5.2). For sets with more than one epitope, the scores yielded by the BOI strategy are between about 20% (13 epitopes) and 120% (6 epitopes) higher than those of the BEI strategy.

A thorough comparison to the heuristic approach proposed by Vider-Shalit *et al.* [24] follows in Section 7.2.

5.5 Problem Size

Reasonable indicators for an ILP's level of difficulty are the size of the constraint matrix as well as the fraction of non-zero elements in the matrix. The basic ILP 5.1 has one variable per candidate epitope and exactly one constraint. Hence, the problem size grows linearly with the number of candidate epitopes. All elements in the constraint matrix are non-zero.

The extended ILP 5.2 has one variable per candidate epitope and one per allele. Since there are several constraints per epitope, one constraint per allele and one constraint per antigen, the corresponding constraint matrix grows quadratically with the number of candidate epitopes and target alleles and linearly with the number of antigens. However, the

number of non-zero elements in the matrix grows only linearly, yielding an extremely sparse and thus a presumably simple optimization problem. Furthermore, the problem size can be reduced considerably by employing the conservation threshold τ_C as well as the antigen processing threshold τ_{ap} as filter instead of using them as constraints. In addition to the negligible elimination of two constraints per epitope, removal of all insufficiently conserved epitopes as well as of those unlikely to result from antigen processing reduces the number of candidate epitopes and, hence, the number of variables and constraints.

To assess the computational costs of our approach, we performed an analysis comprising three realistic EV design problems. Epitope selection was performed on three sets of conserved epitopes derived from target antigens of different viruses: 3,062 candidate epitopes from nine HIV antigens, 4,370 candidate epitopes from ten IV antigens, and 9,384 candidate epitopes from 40 HCV antigens, four strains with ten antigens each. (For more details on the data sets refer to Chapter 7.) The ILP takes antigen processing, overlapping epitopes, allele coverage and antigen coverage into account. The task was to select ten epitopes covering 20 of 27 target MHC alleles. Required antigen coverage and antigen processing threshold were adapted for each problem to ensure feasibility. The commercial solver CPLEX [109] solved each of these problems in a split second. Table 5.1 lists runtime, number of variables and constraints as well as sparsity of each of the problems. It becomes clear, that the structure of the epitope selection problem allows to efficiently solve realistically-sized problems, including problems bigger than those examined here. The level of difficulty of a particular selection problem depends largely on the composition of the set of candidate epitopes under consideration.

Table 5.1: Properties of the epitope selection ILP for different EV design problems. Number of candidate epitopes, number of antigens, number of variables and number of constraints of each ILP are listed. The column *sparsity* contains the percentage of constraint matrix elements that are zero. Runtime corresponds to the time CPLEX needed to solve the ILP.

Problem	Epitopes	Antigens	Variables	Constraints	Sparsity	Runtime
HIV	3,062	9	3,089	24,210	99.8%	0.15 s
IV	4,370	10	4,397	34,662	99.9%	0.26 s
HCV	9,384	40	9,411	80,463	99.9%	0.71 s

5.6 Implementation

We used the commercial ILP solver IBM ILOG CPLEX 9.1 [109] with its C++ interface ILOG Concert Technology 2.1 to implement and solve the epitope selection ILP. (For more details on the implementation refer to Section 7.2.)

5.7 Discussion

The selection of an epitope set with very high overall immunogenicity is crucial for the efficacy of an EV. Depending on the number of candidate epitopes to choose from, the

number of alleles to be considered, as well as on the additional requirements, this problem can become very complex. We propose a mathematical framework that can be used to solve this problem quickly for practical problem sizes. For several characteristic examples, we show that immunological requirements can be conveniently formulated as an ILP. The solution of this ILP yields an optimal set of epitopes: the set of epitopes that displays the highest overall immunogenicity of all sets which meet the pre-defined requirements. To our knowledge, this is the only approach that yields provably optimal solutions to the vaccine design problem for EVs. In contrast to previous heuristics, the optimal solution yields either significantly better overall immunogenicity for the same number of epitopes or a smaller number of required epitopes to reach the same level of immunogenicity. The flexibility of the framework allows selecting other objective functions, too, for example, maximizing antigen or allele coverage. This will be demonstrated in the vaccine design studies in Chapter 7.

The optimal selection of epitopes yields – in theory – significantly higher overall immunogenicities than other strategies (e.g., selection of the best epitopes or evolutionary algorithms). However, one should keep in mind that the selection of the epitopes is still a difficult and controversial issue since the underlying processes are not yet fully understood. In particular, the interplay of different epitopes poses a difficult problem. Competition between epitopes will probably result in reduced immunogenicity of peptide cocktails, an effect that has been observed in various studies. On the one hand, this represents a problem, because the assumption of independence between epitopes is one of the key assumptions made in this work (and in all related approaches). Lacking an accurate model of these competition effects, however, it seems like the best assumption one can make. On the other hand, the effects of competition are a compelling reason to employ this type of selection approach. Competition effects will be less severe for fewer peptides, therefore a selection procedure that yields the same overall immunogenicity with fewer peptides can in fact mitigate this problem. Assuming that competition primarily arises between epitopes binding to the same allelic variant of MHC molecules, one can also introduce additional constraints to reduce competition (e.g., find the best combination that contains at most two epitopes per allele). In the long run, a thorough quantitative analysis of larger vaccination studies might shed some light on these effects and their importance.

Also, the notion of immunogenicity alone, or the ability to evoke an immune response in a certain fraction of patients, is not necessarily a true measure of quality for a vaccine. In their recent review on the quality of the T-cell response [35], Seder *et al.* argue that protective T-cell responses are too complex to be sufficiently described by a measure of magnitude alone. An adequate metric would thus not only account for the magnitude but also for the multifunctional quality of the response. The flexibility of our framework allows for the incorporation of a different quality measure for immunogenicity and a careful comparison of the peptide cocktails suggested by different objective functions would be very interesting.

Chapter 6

Epitope Assembly

The mathematical framework presented in the previous chapter yields an optimal set of epitopes with respect to a given definition of a good vaccine. String-of-beads type vaccines require these epitopes to be combined into one long polypeptide. In the following, we will focus on this epitope assembly step, the last step of the EV design pipeline (Figure 6.1).

6.1 Introduction

In their review of EVs, Purcell *et al.* [41] point out that, to date, there are no EVs for humans on the market. This is mainly attributed to the difficulties associated with peptide stability and delivery. Various delivery strategies are being explored in clinical studies [25]. A popular approach is the assembly of the vaccine epitopes into a single polypeptide (Figure 2.3). Here, presentation of the vaccine epitopes to T cells relies on the accurate processing of the polypeptide. An unfavorable epitope order can result in the degradation of the vaccine epitopes. Furthermore, the generation of unintended junctional epitopes may elicit undesired immune responses [110]. The epitope order is thus crucial for the success of the vaccine. It should be optimized with respect to the specificities of the antigen processing pathway. Combinatorial explosion aggravates the identification of the epitope order yielding the best epitope recovery: While five epitopes can be assembled into

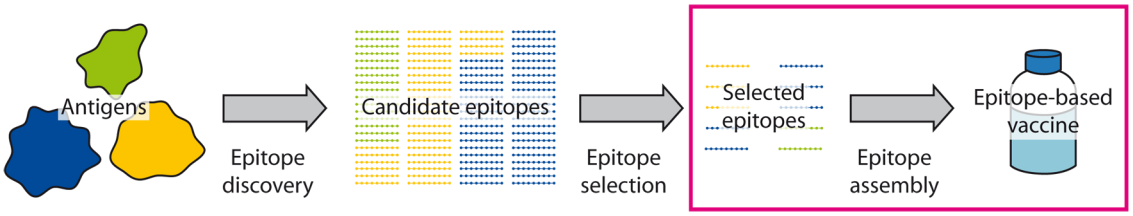


Figure 6.1: EV design: epitope assembly step. For string-of-beads type vaccines the epitopes provided by the epitope selection step have to be skillfully assembled into a longer polypeptide. Figure based on [8].

a string-of-beads construct in 120 ways, there are 3.6×10^6 different ways for 10 epitopes and 2.6×10^{32} for 30 epitopes. Since manual determination of an optimal epitope order is obviously infeasible, computational approaches are called for.

All previously proposed computational approaches to epitope assembly have been published in combination with an epitope selection strategy. They have, thus, already been mentioned in the previous chapter. DeGroot *et al.* [23] created fixed-length polypeptides by overlapping as many epitopes as possible. From a ranked list of MHC-II binding ninemers, a top-ranking ninemer is selected. This core epitope is extended by seven AAs in both directions with as many high-ranking overlapping epitopes as possible. The approach proposed by Vider-Shalit *et al.* [24] employs a genetic algorithm to design a string-of-beads construct. Epitope selection and assembly are performed simultaneously. In addition to requirements on the contained epitopes, the scoring function considers the probability of each epitope to be properly cleaved in the sequence. Although not actually EVs, the *in silico* methods proposed by Fischer *et al.* [111] and Nickle *et al.* [112] to compress the variation found in naturally occurring antigens into a small number of composite antigens can also be considered as related to epitope assembly.

None of the algorithms above yields the optimal order of a given set of epitopes with respect to epitope recovery. In this chapter, we show that this problem is related to the well-studied TSP. Employing an ILP formulation of the TSP and a sophisticated ILP solver, an optimal epitope order can be found within seconds for typical problem sizes. Furthermore, we demonstrate that an efficient implementation of a TSP heuristic finds optimal solutions to the problem even faster.

6.2 Approach

We formulate the epitope-ordering problem in graph-theoretic terms. Let $G = (V, E, w)$ be a complete directed and weighted graph with nodes V, edges E, and weights w. Each node represents an epitope and each edge a possible concatenation of the respective pair of epitopes: edge (a, b) represents the concatenation $a - b$, i.e., epitope a is the N-terminal neighbor of b. Each edge (a, b) is assigned a weight w_{ab} corresponding to the logarithm of the probability that epitopes a and b will be cleaved properly if a is the N-terminal neighbor of b. Given this graph, the optimal epitope ordering corresponds to the maximum Hamiltonian path, i.e., the path that visits each node exactly once and has maximum weight (Figure 6.2A). The problem of finding a Hamiltonian path in G is equivalent to finding a Hamiltonian cycle in a slightly modified graph G'. G' is obtained by adding a node ε to G as well as equally weighted edges from ε to all other nodes and from all other nodes to ε (Figure 6.2B). The maximum Hamiltonian path in G can be obtained by removing ε from the maximum Hamiltonian cycle in G'. Equivalent to finding the maximum Hamiltonian path is the problem of finding the minimum Hamiltonian path. The latter is more commonly known as TSP and has been studied extensively. In order to benefit from this, we interpret the epitope-ordering problem as TSP. Each epitope corresponds to a city. Since the TSP aims at finding the shortest tour, we use the negative of the logarithms of the cleavage probabilities as road lengths, i.e., the higher the probability

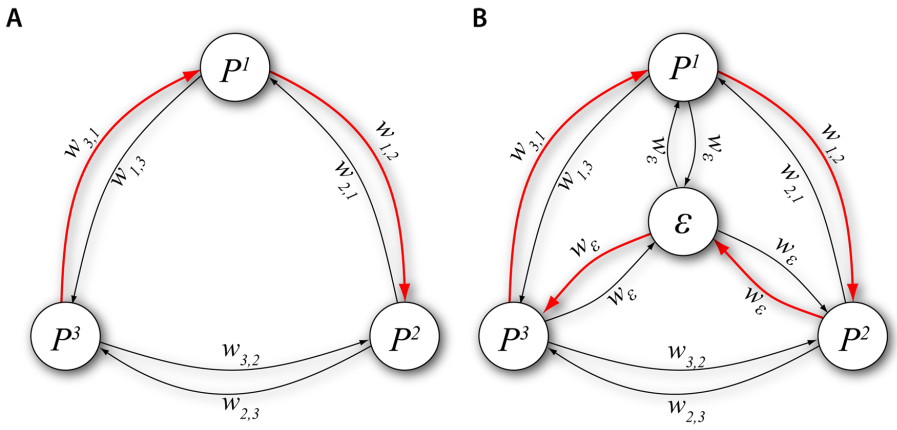


Figure 6.2: Graph representation of the epitope-ordering problem. Nodes represent epitopes, edges represent concatenations and edge weights correspond to logarithms of cleavage probabilities. A) Graph G: a Hamiltonian path corresponds to a particular epitope ordering. B) Extended graph G': a Hamiltonian cycle corresponds to a particular epitope ordering. Example: The Hamiltonian path $P^3 - P^1 - P^2$ in G corresponds to the Hamiltonian cycle $\varepsilon - P^3 - P^1 - P^2 - \varepsilon$ in G'.

the shorter the road. A dummy epitope ε corresponding to the salesman's home town or rather to both the N- and C-terminal end of the polypeptide is introduced. Removal of this dummy epitope from the shortest path through all epitopes yields the epitope order with the highest overall epitope cleavage probability (Figure 6.2).

The TSP has been shown to be NP-complete. Nevertheless, several algorithms exist that can efficiently solve the problem for small numbers of cities. We employ a previously proposed ILP formulation of the TSP [113] in combination with a sophisticated ILP solver to solve the epitope ordering problem. Furthermore, we employ an efficient implementation of a TSP heuristic [114].

6.2.1 ILP Formulation

The ILP formulation that we employ (ILP 6.1) is based on the Miller-Tucker-Zemlin formulation of the TSP [115]. The first two sets of constraints guarantee that the resulting path contains all epitopes and that every epitope has exactly one N- and one C-terminal neighbor. However, these constraints do not ensure the optimal solution to be a Hamiltonian cycle. Given these constraints and the graph G' in Figure 6.2B, for example, the solution $P^3 - P^1 - P^3$, $P^2 - \varepsilon - P^2$, which corresponds to two subcycles, would be feasible. However, in order to optimally assemble all epitopes into one polypeptide we require the solution to correspond to a single cycle. This is ensured by the remaining three constraints. These constraints exclude all solutions that contain more than one cycle from the search space. The variable u_a represents the ordinal number of epitope a in the cycle, starting from the dummy epitope, which is assigned the ordinal number 1 by (6.1c). All other nodes are assigned an ordinal number between 2 and the number of epitopes including the

dummy epitope by (6.1d). (6.1e) requires all epitopes except for the dummy epitope to have a higher ordinal number than their N-terminal neighbor. If the edge between two nodes $a, b \in E$ is included in the solution, i.e., $\mathsf{x_{ab}} = 1$, (6.1e) becomes $\mathsf{u_b} \geq \mathsf{u_a} + 1$: the ordinal number of b is required to be greater than the ordinal number of a. Otherwise, (6.1e) becomes $\mathsf{u_a} - \mathsf{u_b} \leq |E'| - 2$, which is always true due to the (6.1d). This way the only possibility to form a cycle and thus to fulfill the first two constraints is by inclusion of the dummy epitope. Given these constraints and the graph G' in Figure 6.2B the solution $P^3 - P^1 - P^3$, $P^2 - \varepsilon - P^2$ is not feasible: In the cycle $P^3 - P^1 - P^3$, P^3 is the N-terminal neighbor of P^1 and P^1 is the N-terminal neighbor of P^3. According to the fifth set of constraints, the ordinal number of P^1 has to be greater than the ordinal number of P^3 and *vice versa*, which is a contradiction. However, the solution $\varepsilon - P^3 - P^1 - P^2 - \varepsilon$, which corresponds to a Hamiltonian cycle, is feasible: assigning the ordinal numbers 2, 3, and 4 to P^3, P^1, and P^2, respectively, every node except for the dummy node has a higher ordinal number than its N-terminal neighbor. Since (6.1e) does not apply to the dummy node, the solution is a feasible solution for the ILP.

ILP 6.1: ILP formulation of the epitope ordering problem.

minimize	$\sum_{a,b \in E'} w_{ab}\, \mathsf{x_{ab}}$		
subject to	$\forall a \in E'$:	$\sum_{b \in E'} \mathsf{x_{ab}} = 1$	(6.1a)
	$\forall a \in E'$:	$\sum_{b \in E'} \mathsf{x_{ba}} = 1$	(6.1b)
		$\mathsf{u}_\varepsilon = 1$	(6.1c)
	$\forall a \in E$:	$2 \leq \mathsf{u_a} \leq \lvert E' \rvert$	(6.1d)
	$\forall a, b \in E, a \neq b$:	$\mathsf{u_a} - \mathsf{u_b} + 1 \leq (\lvert E' \rvert - 1)(1 - \mathsf{x_{ab}})$	(6.1e)

DEFINITIONS

E	Set of epitopes
ε	Dummy epitope
E'	Set of epitopes including dummy epitope ($E' = E \cup \{\varepsilon\}$)

PARAMETERS

w_{ab}	Negative of the logarithm of the probability that epitopes a and b will be processed properly when a is the N-terminal neighbor of b in the polypeptide

VARIABLES

$\mathsf{x_{ab}} = 1$	if epitope a is the N-terminal neighbor of epitope b; otherwise $\mathsf{x_{ab}} = 0$
$\mathsf{u_a}$	Position of epitope a in the optimal polypeptide ($1 \leq \mathsf{u_a} \leq \lvert E' \rvert$)

6.2.2 Heuristic

We use a TSP heuristic proposed by Lin & Kernighan [116]. The Lin-Kernighan heuristic (LKH) is based on the *λ-opt algorithm*, which again is based on the *λ-opt concept*: A tour is *λ-optimal* if no shorter tour can be obtained by replacing λ links. The larger the value of λ the more likely it is for a λ-optimal tour to be optimal. Starting from a randomly

generated tour, the λ-opt algorithm tries to find two sets of λ links A and B, such that replacing A with B in the current tour results in a better tour. Since the number of operations required to find such a pair of sets grows exponentially with λ, typically the values $\lambda = 2$ and $\lambda = 3$ are chosen. The LKH is a *variable* λ*-opt* algorithm. Instead of specifying λ in advance, in each iteration the LKH examines for increasing values of λ whether the current tour can be converted into a better tour by exchanging λ links.

We employ Helsgaun's implementation of this heuristic [114]. According to the author, the implementation found optimal solutions for all problem instances with known optimum he was able to obtain, including a 13,509-city problem. Runtime of the algorithm is approximately $\mathcal{O}(n^{2.2})$.

6.3 Incorporation of Proteasomal Cleavage Predictions

Several algorithms for the prediction of the probability of a peptide to result from antigen processing exist. They either focus on proteasomal cleavage, TAP transport or both. As noted in several places, the influence of TAP transport is often rather limited [24, 73]. Thus, in this study, TAP transport is not considered. Existing proteasomal cleavage predictors either predict cleavage sites [73, 117] or cleaved fragments [118, 119].

6.3.1 Cleavage Site Predictions

The incorporation of cleavage site predictions into our approach is straightforward as long as the predictor does not require more flanking residues than available when concatenating two epitopes. Given two peptides a and b, the negative of the scores predicted for the peptide bond between the C-terminus of a and the N-terminus of b can be used as weight for the corresponding edge (a, b). If cleavage of this peptide bond is highly unlikely, the respective edge can be excluded from the solution by setting edge costs to infinity. Obviously, exclusion of edges has to be employed moderately to keep the problem feasible.

Depending on the number of flanking residues required by the cleavage site predictor it might be possible and beneficial to penalize or, to a certain degree, explicitly exclude cleavage within vaccine epitopes. Penalization can be achieved by adding a fraction of the respective scores to the corresponding edges.

6.3.2 Cleaved Fragment Predictions

Cleaved fragment predictors require as input the respective peptide along with a certain number of its N- and C-terminally flanking residues, i.e., in our setting, the N- and C-terminally neighboring peptides. This consideration of epitope triples would interfere with the TSP interpretation, which only allows for the consideration of epitope tuples. Whether cleaved fragment predictions can be incorporated into our approach thus depends on the predictor's scoring function. The cleaved fragment predictor proposed by

Ginodi *et al.* [119], for example, is based on a linear scoring function S with

$$S(P) = S_1(f_N) + S_2(p_1) + \sum_{i=2}^{n-1} S_3(p_i) + S_4(p_n) + S_5(f_C) \tag{6.1}$$

where $P = p_1 \dots p_n$ is a peptide fragment with N-terminally flanking residue f_N and C-terminally flanking residue f_C. The function's linearity allows for a simple rearrangement, S', such that considering epitope tuples is sufficient to determine the summed cleavage scores for a set of peptides in a string-of-beads construct. For nonameric peptides $P^1, P^2, \dots, P^k$ arranged in a polypeptide $P^1 - P^2 - \dots - P^k$ it is

$$\sum_{i=1}^{k} S(P^i) = \sum_{i=1}^{k+1} S'(P^{i-1} - P^i) \tag{6.2}$$

with

$$S'(P^{i-1} - P^i) = S_{\text{suf}}(P^{i-1}) + S_{\text{pre}}(P^i) \tag{6.3}$$

and

$$S_{\text{pre}}(P) = S_1(f_N) + S_2(p_1) + \sum_{i=2}^{5} S_3(p_i) \tag{6.4}$$

$$S_{\text{suf}}(P) = \sum_{i=6}^{8} S_3(p_i) + S_4(p_9) + S_5(f_C) \tag{6.5}$$

where P^0 and P^{k+1} correspond to the dummy epitope $\varepsilon = \epsilon_1\epsilon_2 \dots \epsilon_9$ and $S_i(\epsilon_j) = 0$ for $i = 1, \dots, 5$ and $j = 1, \dots, 9$. Incorporation of the cleaved fragment predictor proposed by Ginodi *et al.* can thus be achieved by assigning the negative of the score S' to the respective edges. A drawback of employing S' instead of S is that it does not provide us with cleavage probabilities for the individual vaccine epitopes: Epitope orders that would render a vaccine epitope unlikely to be cleaved cannot be detected during construction of the graph. Hence, specific exclusion of such orders is not possible. Penalization of epitope orders yielding good cleavage scores for unintended junctional peptides, however, is possible to a certain extent. This can be achieved by incorporating (a fraction of) the cleavage scores of junctional peptides into the corresponding edge weight.

6.4 Experimental Results

6.4.1 Efficiency

The complexity of the TSP and thus of the epitope ordering problem raises the question of whether an optimal epitope order can be found in reasonable time for real world problems. We employ the ILP solver CPLEX [53] as well as a highly effective implementation of the LKH (LKH, version 2.0.5) [114] to solve the epitope ordering problem for different-sized epitope sets.

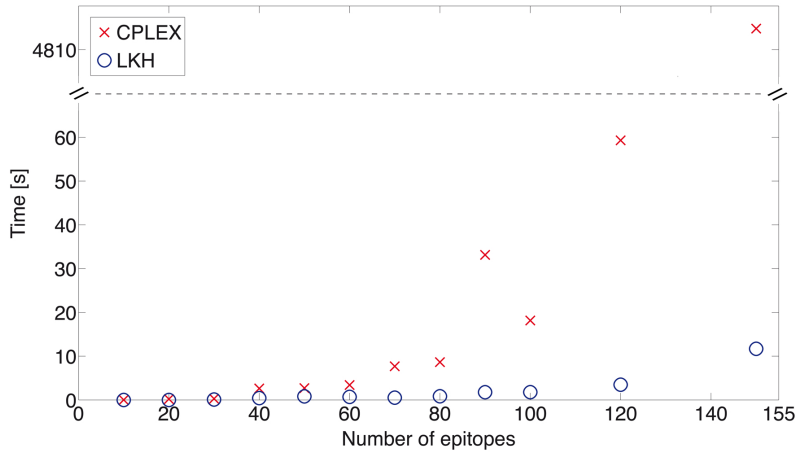


Figure 6.3: Runtimes of the epitope ordering algorithms. Runtimes of a heuristic (LKH) and an exact algorithm (CPLEX) to solve the epitope ordering ILP for different-sized epitope sets are displayed. In all cases, the heuristic found the optimal solution.

Figure 6.3 shows the runtimes of CPLEX and LKH on epitope sets ranging from size 10 to 150. LKH finds the optimal solution for each problem within seconds. On 150 epitopes CPLEX is more than 400 times slower than the heuristic: 4,800 s vs. 12 s. However, although still considerably slower, for solving problems in the practically relevant single- to double-digit range CPLEX also requires only seconds.

6.4.2 Effectiveness

Favorable epitope orders are characterized by a high recovery of the vaccine epitopes by antigen processing and by a higher probability of vaccine epitopes to result from antigen processing compared to unwanted junctional epitopes. In order to assess the effectiveness of our approach in generating favorable epitope orders, we apply the approach to 36 different-sized HCV epitope sets (5 to 40 epitopes) selected by our epitope selection framework in Section 5.4. From each set we designed a string-of-beads construct optimized for C-terminal cleavage of the vaccine epitopes using the cleavage site predictor NetChop [30] (version 3.1). Additionally, we designed 100 randomly ordered polypeptides from each set. Epitope recovery was evaluated based on an *in silico* cleavage of all polypeptides (Figure 6.4). While the optimized orderings yield perfect recovery for all epitope sets, the random orderings achieve an average recovery of 85% ± 4% (Figure 6.4A,C). The mean NetChop scores of the C-termini of the vaccine epitopes in the optimized orderings are between 12.1% and 17.4% (average: 15.2%) higher than those in the random orderings (Figure 6.4B,C).

Both random and optimized string-of-beads constructs yield above-average NetChop scores for the vaccine epitopes: 0.77 vs. 0.42 (factor 1.8) and 0.89 vs. 0.43 (factor 2.1), respectively. This preference for cleaving C-termini of epitopes may be attributed to the fact that NetChop was trained on naturally processed MHC-I binding peptides and is thus

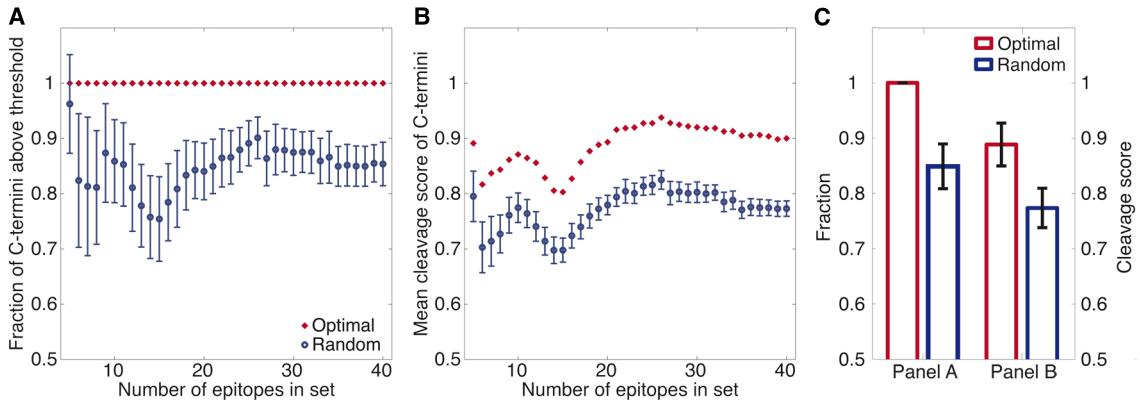


Figure 6.4: Cleavage patterns of the designed string-of-beads constructs. The fraction of C-termini of the vaccine epitopes predicted to be cleaved (A) and the mean cleavage score of the C-termini of the vaccine epitopes (B) are shown as red diamonds for the optimized and as blue circles with error bars for the randomly generated string-of-beads constructs. C) Mean fractions (Panel A) and mean cleavage scores (Panel B) of the C-termini of vaccine epitopes predicted to be cleaved over all different-sized string-of-beads constructs.

biased towards cleavage after AAs common in C-terminal positions of MHC-I binders. However, the difference between cleavage scores for vaccine and junctional epitopes is more pronounced in the optimized string-of-beads constructs compared to the random constructs.

A combination of the IBM ILOG CPLEX [53], the GNU Linear Programming Kit GLPK [55] and the GNU MathProg modeling language GMPL was used to formulate and solve the epitope ordering ILPs. (For more details on the implementation refer to Section 7.3.)

6.5 Discussion

A crucial step in the construction of a string-of-beads vaccine is the epitope assembly. An unfavorable epitope order can result in the degradation of the vaccine epitopes and in the generation of unintended junctional epitopes. Since manual determination of an optimal epitope order is infeasible, we propose a graph-based approach that optimally assembles a set of epitopes into a string-of-beads construct. We could show that our algorithm finds favorable epitope orderings efficiently. This problem had not been addressed computationally before.

Formulating the epitope ordering problem in graph-theoretic terms and relating it to the TSP allowed us to benefit from the wealth of research that has been done on solving the TSP. We employed two previously published approaches to solving the TSP: an exact ILP approach and the highly effective LKH. Although the heuristic is considerably faster than the ILP solver CPLEX, both find the optimal solution for realistically-sized

epitope sets within seconds. The freely available ILP solver GNU Linear Programming Kit GLPK [55], which easily solved the epitope selection problem described in the previous chapter, is significantly slower (130 s for 30 epitopes, more than a day for 40 epitopes) but still reasonably for typical problem sizes.

Our experiments show that the optimal epitope ordering displays more favorable *in silico* cleavage patterns than random orderings. However, the actual extent of the advantage of optimized string-of-beads constructs over random constructs stands or falls with the accuracy of the employed proteasomal cleavage predictor. State-of-the-art proteasomal cleavage predictors yield good accuracies but leave room for improvement. Nevertheless, even based on a poor predictor the resulting cleavage pattern can be expected not to be worse than one obtained without an epitope ordering algorithm.

Several experimental groups [110, 120, 121] have suggested to introduce spacer sequences between the epitopes to improve epitope recovery and prevent the cleavage of junctional epitopes. While we focused on the construction of string-of-beads constructs without spacers, our assembly approach can easily be adapted to include spacers: Given a spacer s, the edge (a, b) is weighted by the sum of the log-probabilities that a and b will be cleaved properly from the sequence $a - s - b$, respectively. However, depending on the length of the chosen spacer and the number of flanking residues required by the chosen proteasomal cleavage predictor, the predicted cleavage probabilities of the vaccine epitopes might depend solely on the spacer sequence. The cleaved fragment predictor proposed by Ginodi *et al.* [119], for example, requires only one flanking residue to determine the cleavage score of an epitope. Hence, a spacer length of one AA or more will result in the same cleavage scores for the vaccine epitopes independent of their order within the polypeptide. In this setting, it would be expedient to employ the number of unwanted epitopes expected to be generated from the sequence $a - s - b$ or the probabilities of these unwanted epitopes to weigh the edge (a, b).

Chapter 7

Applications

In the previous chapters we have proposed new approaches to the *in silico* design of EVs. This chapter focuses on applications of some of these methods. We begin with introducing OptiTope, a publicly available and easy-to-use web server that we developed to make our mathematical framework for epitope selection (Chapter 5) available to immunologists. Subsequently, we will present two vaccine design studies. The first study focuses on the design of a peptide cocktail vaccine against HCV. In the second study we employed our algorithms to analyze the feasibility of designing string-of-beads vaccines against highly variable viruses like HIV, HCV and IV yielding broad coverage of the targeted population as well as of the various viral strains. Parts of this chapter have previously been published [102, 103].

7.1 OptiTope – A Web Server for Epitope Selection

Based on a specific application of our epitope selection framework, OptiTope aims at assisting immunologists in the critical task of epitope selection. It is an easy-to-use web-based tool to efficiently determine an optimal set of epitopes in a specific individual or a target population.

7.1.1 Web Interface

OptiTope requires the following input data: (1) sequences of known antigens, (2) a target population, i.e., MHC alleles and corresponding probabilities and (3) the user's requirements on the epitope set to be selected. The information given by the user is transformed into an optimization problem. If this problem is feasible, OptiTope will return an optimal set of epitopes along with additional information on their respective contribution to the overall immunogenicity. Otherwise, OptiTope will propose changes to the user's requirements that might yield a feasible optimization problem. The structure of the web interface is depicted in Figure 7.1.

An introductory tutorial is provided on the OptiTope home page to assist new users in learning how to use the web server.

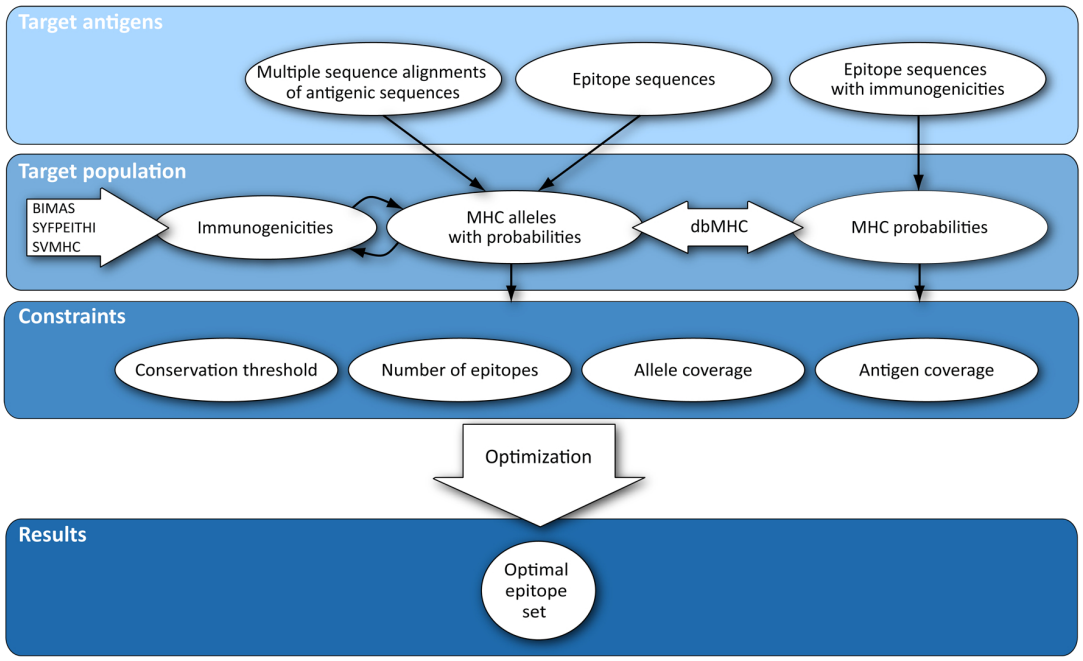


Figure 7.1: Structure of the OptiTope web interface. OptiTope is divided into four parts: target antigens, target population, constraints and results. (1) Target antigens: three different formats of antigenic sequences can be entered: (i) a list of multiple sequence alignments of target-specific antigens, (ii) a list of epitopes and (iii) a table of epitopes with (experimentally determined) immunogenicities with respect to specific alleles. (2) Target population: the information required to specify the target population (MHC alleles and corresponding probabilities) depends on the chosen input format. (3) Constraints: OptiTope offers a set of constraints, which can be modified or excluded by the user. (4) Results: if feasible, OptiTope presents an optimal set of epitopes.

Conceptual design

For ease of use, the web interface is divided into four steps: three input steps and one output step. Step-by-step, the user is asked to enter the required data. Navigation through the individual steps is guided by a navigation bar at the top of each site. The navigation bar indicates the current step and contains corresponding instructions. Furthermore, it provides access to a more detailed help page. In order to keep the page layout clear, settings and options are hidden from the user by default. They can be accessed via the advanced options button underneath the navigation bar.

Step 1: Target sequences

In the first step, the sequences of known target-specific antigens are entered. They can either be pasted directly or uploaded as a file. Three different formats are accepted: (1) a list of MSAs in FASTA format, (2) a list of epitopes of equal length, one epitope per line, and (3) a table of epitopes and their predicted or experimentally determined immunogenicities with respect to specific MHC alleles. Higher immunogenicity values ought to indicate stronger immunogenicities. Antigenic sequences entered as MSAs will be

converted into consensus sequences. From these sequences, all peptides of a given length will be derived and will be considered as candidate epitopes. The user can adjust the peptide length to be applied via the advanced options.

Step 2: Target population

In the second step, information on the target population has to be entered. This step is subdivided into two queries. The user is queried (1) for the MHC alleles to consider (if they have not already been entered in the previous step) and (2) for their probabilities in the target population.

1. MHC alleles can be selected by population or geographic area based on data [122] retrieved from the NCBI dbMHC database [123]. The corresponding probabilities will be employed for the next query. Alternatively, the MHC alleles can be selected manually from an expandable allele tree [124] or by pasting a list of alleles.

2. In this step, a list of the selected MHC alleles along with probabilities (default values or values retrieved from the NCBI dbMHC database, respectively) is given. These probabilities can either be modified manually or they can be replaced by population- or geographic-area-specific probabilities from the NCBI dbMHC database via the advanced options. Individual MHC alleles can be excluded from further processing. Furthermore, low probability MHC alleles can be excluded from the epitope selection process via a filter in the advanced options.

If the user has not entered the immunogenicities of the candidate epitopes together with the target sequences, OptiTope will employ a prediction method to determine the respective immunogenicities. The prediction method to be employed can be selected via the advanced options.

Step 3: Constraints

In the third step, the user is queried for the requirements on the epitope set to be selected. Depending on the data that have been entered in the previous steps – a summary of these data is given – a list of suitable constraints is displayed. The user can (de)select and modify these constraints. Potential constraints are:

- *Maximum number of epitopes to select.* This constraint defines the maximum number of epitopes OptiTope should select. It is the only mandatory constraint.

- *Minimum epitope conservation.* This constraint ensures that only epitopes that fulfill a user-defined conservation requirement will be considered.

- *Minimum number of alleles to cover.* If this constraint is selected, the optimal set of epitopes will be immunogenic with respect to the specified number of MHC alleles or more.

- *Minimum number of antigens to cover.* This constraint guarantees that the optimal epitope set will include epitopes from a specified number of antigens or more.

The advanced options offer the possibility to set an immunogenicity threshold, i.e., a minimum immunogenicity score required for a peptide to be considered immunogenic with respect to a specific allele. Only peptides which score above this threshold for at least one MHC allele will be considered during epitope selection.

Step 4: Results

The results page gives a summary of the input data and the selected constraints as well as the results of the optimization. If the optimization problem is feasible, a table containing the optimal set of epitopes will be displayed (Figure 7.2). For every epitope in the set the following information is given: its fraction of the overall immunogenicity, a list of the MHC alleles it covers and, if antigen information was given, the corresponding antigens. The user can switch to a more detailed results table, which contains additional information on epitope conservation and immunogenicities. Information on the size of the selected set, the number of covered alleles and on the number of covered antigens, if applicable, is displayed above the table. Furthermore, the coverage of each of the given MHC loci and the corresponding population coverage are given. (If locus A has a coverage of 75%, the probability of an individual from the target population carrying a covered allele at locus A is 75%. A population coverage of 80% corresponds to a probability of 80% for an individual from the target population to carry at least one of the covered alleles.) The results can be downloaded. A choice of two file formats is given: XLS (MS Excel) and CSV (comma separated values). For typical problem sizes, OptiTope finds an optimal set of peptides within seconds. Nevertheless, the user can choose to be notified of the completion of the request via e-mail. If the optimization problem is infeasible, meaning that no set of epitopes from the given antigenic sequences fulfills all requirements, a basic analysis of the problem is performed. Based on this analysis, OptiTope suggests constraint modifications that might result in a feasible problem. If the basic analysis does not yield a possible explanation for the infeasibility, OptiTope will suggest to deselect individual constraints or to increase the number of epitopes to be selected.

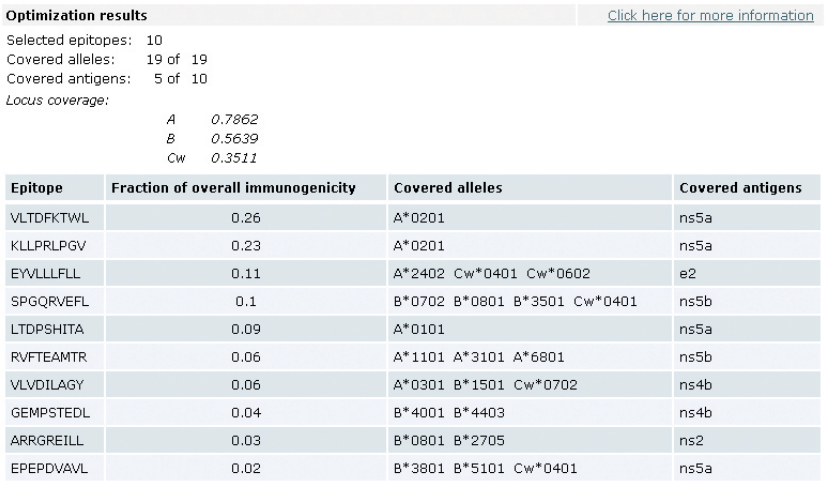
7.1.2 Implementation

OptiTope is incorporated into the website EpiToolKit [124], which is based on the Zope application server [125], and the content management system Plone [126]. For the user interface, we employ dynamic HTML with CSS and JavaScript. Python scripts are used for data validation and processing. OptiTope was thoroughly tested for compatibility with the popular web browsers Mozilla Firefox (version 3.0.5) and Microsoft Internet Explorer (version 7).

OptiTope uses the GNU Linear Programming Kit GLPK [55] and the GNU MathProg modeling language GMPL to formulate and solve the optimization problems.

7.1.3 Discussion

With OptiTope, we provide an easy-to-use tool that assists immunologists in designing EVs. Given a set of antigenic sequences of interest, a target population and special requirements



Optimization results Click here for more information

Selected epitopes: 10
Covered alleles: 19 of 19
Covered antigens: 5 of 10
Locus coverage:

A	*0.7862*
B	*0.5639*
Cw	*0.3511*

Epitope	Fraction of overall immunogenicity	Covered alleles	Covered antigens
VLTDFKTWL	0.26	A*0201	ns5a
KLLPRLPGV	0.23	A*0201	ns5a
EYVLLLFLL	0.11	A*2402 Cw*0401 Cw*0602	e2
SPGQRVEFL	0.1	B*0702 B*0801 B*3501 Cw*0401	ns5b
LTDPSHITA	0.09	A*0101	ns5a
RVFTEAMTR	0.06	A*1101 A*3101 A*6801	ns5b
VLVDILAGY	0.06	A*0301 B*1501 Cw*0702	ns4b
GEMPSTEDL	0.04	B*4001 B*4403	ns4b
ARRGREILL	0.03	B*0801 B*2705	ns2
EPEPDVAVL	0.02	B*3801 B*5101 Cw*0401	ns5a

Figure 7.2: A screenshot from the results page of OptiTope.

of the user, OptiTope efficiently determines an optimal set of epitopes. To our knowledge, OptiTope is the first web-based approach for optimal vaccine design.

Currently, OptiTope only offers immunogenicity predictions for MHC-I, i.e., the only way to include MHC-II in the selection process is via a table of epitopes and their immunogenicities with respect to specific MHC alleles. An inclusion of MHC-II prediction methods along with further MHC-I prediction methods would be desirable. Furthermore, the results page could be enhanced by linking selected epitopes that can be found in the IEDB [87] to the corresponding IEDB site.

7.2 Design of a Peptide Cocktail Vaccine

The focus of the first vaccine design study is on evaluating the performance of the epitope selection framework presented in Chapter 5 in comparison to the heuristic approach proposed by Vider-Shalit *et al.* [24]. Vider-Shalit *et al.* use a genetic algorithm to design a string-of-beads EV against HCV, comprising 25 epitopes. Employing our epitope selection framework and two different definitions of good vaccine, we design HCV peptide cocktail vaccines. The resulting vaccines outperform the peptide cocktail corresponding to the EV proposed in [24] with respect to various quality criteria including overall immunogenicity and population coverage.

7.2.1 Materials & Methods

Protein sequences. HCV protein sequences (AA frame 1) for ten different proteins (C, E1, E2, p7, NS2, NS3, NS4A, NS4B, NS5A, NS5B) and four different subtypes (1a, 1b, 2a, 3a) were retrieved from the Los Alamos hepatitis C sequence database [127]. (Unfortunately, Vider-Shalit *et al.* could not provide us with the protein sequences they used.) For each protein of each subtype an MSA was created using MUSCLE [128], resulting in 40 MSAs. From each MSA a consensus sequence was created. All ninemers from these consensus sequences were regarded as potential epitopes.

MHC alleles & binding. MHC alleles, their probability of occurring in the human population, and binding affinity score thresholds were directly taken from Vider-Shalit *et al. In silico* predicted MHC binding affinities using BIMAS matrices [11] are employed as a measure of immunogenicity.

Peptide conservation. To allow a comparison of our results with those of Vider-Shalit *et al.*, we adopt their definition of peptide conservation: A peptide is considered to be at least x% conserved if all of its AAs display a conservation of at least x%. All insufficiently conserved peptides are disregarded.

Antigen processing. To score the probability of a peptide being a result of antigen processing, we used the proteasomal cleavage matrix from the supplementary material of [24]. As noted in several places, the influence of TAP transport is often rather limited [24, 73]. Consideration of TAP transport is thus omitted in this study.

7.2.2 Experimental Results

For the comparison of our epitope selection framework with the work of Vider-Shalit *et al.* [24] we applied ILP 5.2 to the HCV data and 27 of the 29 alleles from [24]. The alleles HLA-B*07:02 and HLA-B*35:01 were omitted, since none of the candidate peptides binds to them. Probably due to an error in sequence processing (personal communication with Y. Louzoun, corresponding author of [24]), a peptide (AALENLVTL) which does not belong to any of the proteins under consideration was included in the 25 epitopes selected by Vider-Shalit *et al.* We exclude this peptide and base our comparison on sets of 24 epitopes.

For the 24 epitopes to be selected, we require a minimum conservation of 90%, an allele coverage of 27, and an antigen coverage of at least one epitope per antigen. Furthermore, only epitopes with antigen processing scores within the top 30% of all sufficiently conserved candidate peptides were allowed to be selected. The selected epitopes (hereafter $\mathsf{E}_{\mathsf{ILP}}$) are listed in Table 7.1. Four epitopes are known HCV epitopes and can be found in the IEDB (release 2008_4_1_3_28) [87]. Another 11 epitopes are contained in known longer epitopes. The overall immunogenicity of the selected set is 2,549. It includes binders for all 27 allelic variants with all 40 antigens being represented and covers 22.7% of all MHC/antigen combinations: A combination of MHC and antigen is covered by an

Table 7.1: Epitopes selected using the extended epitope selection ILP 5.2. Known HCV epitopes are marked with an asterisk. Epitopes contained in known longer epitopes are marked with a plus.

SFSIFLLAL*	GHRMAWDMM+	VYEADDVIL	CFTPSPVVV+	FLLLADARV*	GPADGMVSK+
YLYDHLAPM	GLRDLAVAV+	GPTPLLYRL+	TWVLVGGVL+	IELGGKPAL+	LAGGVLAAV
QYLAGLSTL+	NFVSGIQYL	VLSDFKTWL*	ARPDYNPPL+	KLLPRLPGV	RHTPVNSWL+
GLYLFNWAV	ALYDVVSTL*	RRCRASGVL+	WPLLLLLLA	VTYSLTGLW	YFVIFFVAA

epitope set if the latter includes an epitope from the respective antigen that binds the gene product of the respective allele. The average number of active epitopes per individual of the population is 13.3. The corresponding values of the epitope set selected by Vider-Shalit *et al.* (hereafter E_{VS}) are listed in Table 7.2.

Table 7.2: Overview over properties of HCV epitope sets selected using different strategies. Number of epitopes per set: 24. $\mathsf{E}_{\mathsf{ILP}}$: set selected by our ILP, E_{VS}: set selected by Vider-Shalit *et al.* without peptide AALENLVTL, $\mathsf{E}_{\mathsf{Comb}}$: set selected by our ILP extended by aspects of the scoring function of Vider-Shalit *et al.*

	$\mathsf{E}_{\mathsf{ILP}}$	E_{VS}	$\mathsf{E}_{\mathsf{Comb}}$
Overall immunogenicity	**2,549**	125	2,177
Allele coverage	**100%**	96.3%	**100%**
Antigen coverage	**100%**	87.5%	**100%**
MHC/antigen coverage	22.7%	19.2%	**30.5%**
Population coverage	**96.0%**	95.6%	**96.0%**
Avg. number of epitopes per individual	13.3	11.4	**17.3**
Number of epitopes in IEDB	**4**	1	1

In order to prove the flexibility of our epitope selection approach we incorporate aspects of the scoring function of Vider-Shalit *et al.* in the objective function of ILP 5.2, yielding ILP 7.1. Their scoring function scores a polypeptide x taking into account the number of covered MHC alleles, $M(x)$, the number of covered antigens, $A(x)$, the number of covered MHC/antigen combinations, $C(x)$, and a score for the probability of each epitope in the ordered sequence being properly cleaved, $p(x)$:

$$\mathcal{S}(x) \quad = \quad p(x) \cdot \Big(2.6 \cdot A(x) + 0.77 \cdot M(x) + C(x)\Big).$$

Since we aim at designing peptide cocktail vaccines, we omit the factor $p(x)$. Two sets of binary variables have to be introduced in order to count the number of covered antigens and the number of covered MHC/antigen combinations: $\mathsf{z}_\mathsf{i} = 1$ if an epitope from the i-th antigen belongs to the optimal set and $\mathsf{z}_\mathsf{i} = 0$ otherwise. $\mathsf{w}_{\mathsf{ai}} = 1$ if an epitope from the i-th antigen, which is sufficiently immunogenic with respect to MHC allele a, belongs to the optimal set and $\mathsf{w}_{\mathsf{ai}} = 0$ otherwise. Since immunogenicity scores tend to be higher than the weighted sums of the coverage scores and would therefore outweigh them, we scale the immunogenicity by a (purely empirical) factor of 0.1. We require two sets of additional constraints: (7.1a) and (7.1b). (7.1a) ensures that only covered antigens contribute to the objective function: z_i can only be one if at least one sufficiently immunogenic epitope from antigen i is included in the selected epitope set, otherwise it is zero. (7.1b) is responsible for the correct setting of the MHC/antigen coverage variables w_{ai}. This combined ILP still aims at high overall immunogenicity while at the same time extending the coverage of antigens, MHC alleles, and MHC/antigen combinations. The optimal epitope set selected using this combined ILP (hereafter $\mathsf{E}_{\mathsf{Comb}}$) is only 15% less immunogenic than the original epitope set

$\mathsf{E_{ILP}}$ and more than 17 times more immunogenic than $\mathsf{E_{VS}}$. As for MHC/antigen coverage, it outperforms both (Table 7.2). $\mathsf{E_{Comb}}$ includes one epitope which is already known and 14 epitopes which are contained in longer epitopes listed in the IEDB. Figure 7.3 shows that when using the combined objective function, 18 epitopes suffice to cover all alleles and antigens and furthermore to outperform $\mathsf{E_{VS}}$ in terms of immunogenicity (371 vs. 125) and MHC/antigen coverage (22.8% vs. 19.2%).

ILP 7.1: ILP corresponding to the combined epitope selection problem.

$$\begin{aligned}
\text{maximize} \quad & 0.1 \cdot \textstyle\sum_{e \in E} \mathsf{x_e} \ \sum_{a \in A} p(a)\ i(e,a) + \\
& 2.6 \cdot \textstyle\sum_{j=1}^{n} \mathsf{z_j} + 0.77 \cdot \sum_{a \in A} \mathsf{y_a} + \sum_{a \in A} \sum_{j=1}^{n} \mathsf{w_{aj}} \\
\text{subject to} \quad & \text{all constraints from the extended ILP 5.2} \\
& \forall i \in \{1, \ldots, n\}: \quad \textstyle\sum_{e \in E_i \cap I} \mathsf{x_e} \geq \mathsf{z_i} \qquad (7.1a) \\
& \forall i \in \{1, \ldots, n\}, a \in A: \quad \textstyle\sum_{e \in E_i \cap I_a} \mathsf{x_e} \geq \mathsf{w_{ai}} \qquad (7.1b)
\end{aligned}$$

Definitions

A	Set of observed MHC alleles
E_i	Set of epitopes from the i-th antigen
E	Set of all candidate epitopes ($E = E_1 \cup \ldots \cup E_n$)
I_a	Set of epitopes which, when bound to the gene product of an MHC allele a, display an immunogenicity greater than or equal to a given threshold τ_I
I	Set of all sufficiently immunogenic epitopes ($I = \bigcup_{a \in A} I_a$)

Parameters

$i(e,a)$	Immunogenicity of epitope e with respect to allele a
$p(a)$	Probability of MHC allele a occurring in the target population

Variables

$\mathsf{w_{ai}} = 1$	if allele a is covered by an epitope from the i-th antigen, otherwise $\mathsf{w_{ai}} = 0$
$\mathsf{x_e} = 1$	if epitope e belongs to the optimal set, otherwise $\mathsf{x_e} = 0$
$\mathsf{y_a} = 1$	if allele a is covered by the optimal set, otherwise $\mathsf{y_a} = 0$
$\mathsf{z_i} = 1$	if an epitope from the i-th antigen belongs to the optimal set, otherwise $\mathsf{z_i} = 0$

7.2.3 Implementation

We used Python for data preparation and the commercial ILP solver IBM ILOG CPLEX 9.1 [109] with its C++ interface ILOG Concert Technology 2.1 to implement and solve the epitope selection ILPs. Runtimes are on the order of a few seconds on standard PC hardware for each ILP employed in this study.

Data preparation. Given an MSA of the respective antigen sequences, the corresponding consensus sequence as well as information on the AA conservation within this sequence has to be determined. We implemented a Python script to create consensus sequences with conservation information from MSAs. If an AA is less than 100% conserved, the conserva-

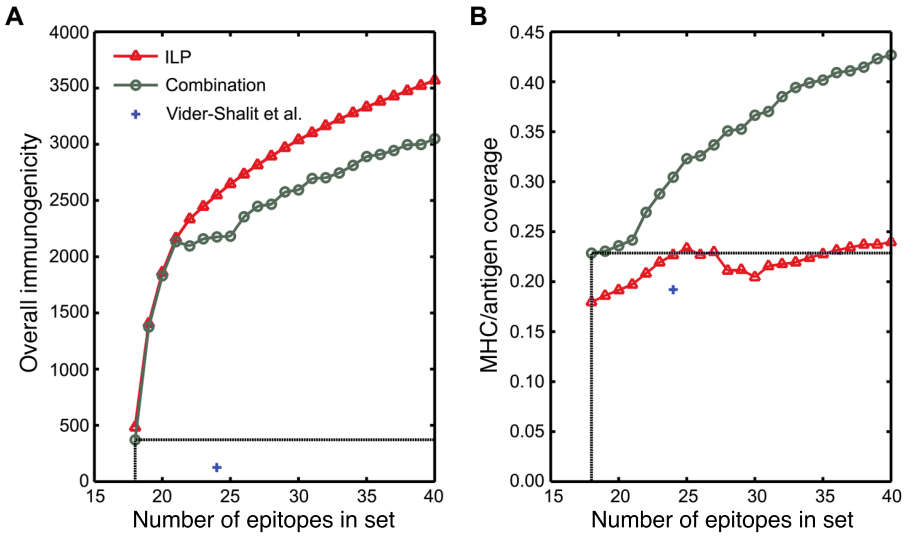


Figure 7.3: Comparison of properties of HCV epitope sets selected using different strategies. A) Overall immunogenicity. B) Coverage of MHC/antigen pairs.

tion is included in the resulting sequence. An example for such a consensus sequence with conservation information is given in the following:

```
MST[N:0.8846]PKPQR[K:0.9615]T-[K:0.9615]RNT[N:0.9231]RRP.
```

Optimization. In order to solve the epitope selection ILPs we implemented a C++ program that accesses CPLEX. The program requires as input the target sequences with conservation information as well as the list of alleles of interest, along with their frequencies in the target population and binding affinity score thresholds, the number of epitopes to select, the conservation and antigen processing thresholds, as well as the required antigen and allele coverages. Given this information, the C++ program extracts all sufficiently conserved ninemers and determines their BIMAS scores with respect to the MHC alleles of interest. Furthermore, it determines proteasomal cleavage scores [119] for the candidate epitopes. From this information, the ILP is generated. The CPLEX C++ interface provides classes for (binary) variables, objective functions and constraints. A CPLEX class representing optimization problems allows the combination of variable, objective function and constraint objects and the solution of the corresponding problem. After defining the required sets of binary variables, a CPLEX maximization objective function object as well as CPLEX objects for the constraints are created employing the binary variables. Subsequently, variables, objective function and constraints are added to a CPLEX optimization problem object. Subsequently, CPLEX solves the problem, yielding the optimal epitope set and the overall immunogenicity.

7.2.4 Discussion

In 2007, Vider-Shalit *et al.* proposed to use a genetic algorithm to simultaneously select and order epitopes for a string-of-beads EV focusing primarily on allele, antigen and MHC/antigen coverage [24]. The authors applied their algorithm to a set of HCV antigens. We employed the peptide cocktail corresponding to the resulting EV to evaluate the epitope selection framework presented in Chapter 5. In addition, we demonstrated the flexibility of our approach.

Two peptide cocktail vaccines were designed using the framework: one based on our refined definition of good vaccine, and one based on a combination of our refined definition and the one proposed in [24]. Both peptide cocktail vaccines outperform the vaccine proposed in [24] with respect to various quality criteria including MHC/antigen coverage, a property that was not incorporated in the design of our first vaccine. Furthermore, we could show that 18 epitopes suffice to provide the same coverage Vider-Shalit *et al.* provide with 24 epitopes.

The results of this study are consistent with the results presented in Chapter 5, demonstrating the utility of our epitope selection framework also for realistic EV design problems.

7.3 Design of String-of-Beads Vaccines

The focus of the second vaccine design study is on highly variable viruses. Traditional anti-viral vaccines are based on killed or attenuated viruses. These whole-organism vaccines rely primarily on the induction of a B-cell response and of neutralizing antibodies. Many widespread viruses, such as HCV, HIV, and IV have highly variable and rapidly mutating surface proteins, making the development of a potent whole-organism vaccine challenging. Vaccine trials against such viruses have either demonstrated the induction of time-limited protection (e.g., IV), failed or induced protection spanning a small part of the population [129–131]. In this study we aim to design universal string-of-beads vaccines, i.e., string-of-beads vaccines yielding broad population coverage as well as broad coverage of viral strains. In addition to taking population coverage, antigen coverage and epitope conservation into account, we also consider the choice of target antigens.

An important mechanism used by viruses in order to evade immune pressure is to reduce the number of epitopes in their proteins via mutations [132]. Such mutations typically have a fitness cost [133–135]. The presence of a high number of escape mutations in a protein indicates that the evolutionary advantage for the virus of removing epitopes in this protein outweighs the corresponding fitness cost. This is the case if a T cell-mediated immune response against the protein reduces the viral survival probability. Hence, proteins with a low epitope density may be promising, albeit difficult, targets for an EV. A caveat of this rationale is that the mutation costs may vary among proteins, and a low number of epitopes may indicate a low fitness cost for mutations instead of a particular importance of the protein. However, the group of Yoram Louzoun has previously shown for multiple viruses that a high epitope density correlates well with late expression of a protein or low protein copy numbers, while low epitope densities are typically observed in early expressed proteins

or in proteins with a high copy number [136–138]. Thus, at least in the viruses analyzed up to now, the relation between low epitope density and the importance of the immune response against the respective protein has been established. It is therefore reasonable, to employ low epitope density as an indicator for promising EV targets.

In the first part of this study, we address the question whether the design of a potent EV against highly variable viruses is feasible at all: Do these viruses exhibit a sufficient number of conserved epitopes to allow the design of an EV inducing broad and potent immunity in the target population? To what extend is it possible to focus the EV on proteins that are critical for the virus? The second part focuses on the optimal assembly of a string-of-beads construct from the selected epitopes.

Our results show that, with the exception of some highly variable and often short proteins, it is indeed possible to find optimal conserved epitopes from proteins with low epitope densities. The efficient assembly of these epitopes into string-of-beads constructs with favorable proteasomal cleavage patterns is demonstrated.

7.3.1 Materials & Methods

Protein sequences. Protein sequences of HCV, HIV and IV were retrieved as follows: HCV sequences for ten different proteins (Core, E1, E2, p7, NS2, NS3, NS4A, NS4B, NS5A, NS5B) and four different subtypes (1a, 1b, 2a, 3a) were taken from the first vaccine design study (Section 7.2). HIV-1 subtype B sequences for nine genomic regions (ENV, GAG, NEF, POL, REV, TAT, VIF, VPR, VPU) were obtained from the Los Alamos HIV sequence database [139]. Non-AA letters occurring in the HIV sequences were replaced by gap symbols. IV sequences of ten different proteins (HA, M1, M2, NA, NP, NS1, NS2, PA, PB1, PB2) were retrieved from the NCBI website [140]. For each viral protein an MSA was created using MUSCLE [128] (Version 3.7 with default parameters). The MSAs' consensus sequences were used for all following computations: specifically, ten sequences for HCV, nine sequences for HIV and ten sequences for IV. Gaps were removed from the consensus sequences. Each nonamer from the gap-free consensus sequences was considered as a potential epitope.

MHC alleles. In the current study, we used 28 MHC alleles: eight HLA-A alleles, 17 HLA-B alleles and three HLA-C alleles. The MHC alleles and their probabilities in the world population [122] are listed in Table C.1 in the appendix. The corresponding population coverage is approximately 91.7%.

MHC binding. The binding affinity of every potential pMHC complex was predicted using the BIMAS matrices [11]. Allele-specific thresholds maximizing the accuracy of the BIMAS predictions were taken from [137]. BIMAS scores were normalized by division by the corresponding threshold. We considered peptides predicted to bind to an MHC molecule with a normalized BIMAS score greater or equal to one as binders. All others were considered non-binders.

Proteasomal cleavage. The probability of a nonamer to result from proteasomal cleavage was scored using the matrix-based approach proposed by Ginodi *et al.* [119]. Nonamers with a negative cleavage score were not considered suitable candidates for inclusion in the EV.

Peptide immunogenicity. The impact of antigen processing on the presentation and, thus, on the immunogenicity of a peptide shall be accounted for in this study. Hence, instead of using solely MHC binding affinity as a measure of immunogenicity, the product of normalized BIMAS score and proteasomal cleavage score is employed:

$$i(e, a) = p_{cl}(e) \cdot b(e, a) \tag{7.1}$$

with $p_{cl}(e)$ the probability that epitope e will result from antigen processing and $b(e, a)$ the binding affinity of epitope e to the gene product of MHC allele a.

Peptide conservation. The MSAs were used to determine the conservation of peptides derived from the respective consensus sequence. Since the N- and C-terminally flanking residues contribute to the probability of an epitope to result from proteasomal cleavage [119], the conservation of these residues is incorporated in the conservation of a nonameric peptide: The conservation of a ninemer corresponds to the minimum conservation displayed by any of its nine residues and its up to two flanking residues. The conservation of a single residue depends on the composition of the MSA. It is defined as the fraction of sequences in the MSA that carry the consensus residue at the respective position.

Size of immune repertoire score. Viruses attempt to evade an immune response through the evolutionary accumulation of escape mutations [132]. Such escape mutations are only positively selected in a given protein, if the gain of the increase in survival probability outweighs the fitness costs of the mutations. For multiple viruses it has been shown that only proteins expressed at critical time periods or at a high copy number have low epitope densities. Such proteins promise to be good targets for an EV. In order to identify low-epitope-density proteins, the size of immune repertoire (SIR) score was defined [137]. The SIR score of a specific protein corresponds to the ratio of predicted epitopes to the number of expected epitopes, i.e., the normalized epitope density. A low SIR score indicates that the virus tries to hide the respective protein. We used the Peptibase web server [141] to determine the SIR scores of the viral proteins' consensus sequences with respect to the MHC alleles under consideration.

Epitope selection. In [137], it has been proposed that the detection of proteins a virus tries to hide, i.e., proteins with a low SIR score (low-SIR proteins), would maximize the impact of a vaccine. Therefore, we would like our epitope selection to be biased towards peptides from low-SIR antigens. Furthermore, we would like a stronger emphasis on epitope conservation. For the epitope selection ILP of this study (ILP 7.2) we adapted the objective

function of the extended epitope selection ILP 5.2 as follows:

$$\text{Maximize} \sum_{q \in Q} \frac{1}{\mathsf{SIR}(q)^2} \sum_{e \in E_q \cap B_a} \mathsf{x}_e \; c(e) \sum_{a \in A} p_{mhc}(a) \; i(e,a) \tag{7.2}$$

where Q is the set of antigens, E_q is the set of epitopes from antigen q, B_a is the set of epitopes binding to the gene product of MHC allele a, $c(e)$ is the conservation of epitope e, A is the set of MHC alleles under consideration, $p_{mhc}(a)$ is the probability of MHC allele a occurring in the target population, and $i(e,a)$ is the immunogenicity of epitope e with respect to allele a as defined in equation 7.1. The factor $c(e)$ provides for a bias towards highly conserved epitopes: highly conserved epitopes are weighted stronger than variable epitopes. The factor $\frac{1}{\mathsf{SIR}(q)^2}$ provides for the desired bias towards low-SIR antigens: it grows with decreasing SIR score resulting in a higher weight for epitopes from low-SIR antigens. Requirements on the epitope set are maximal possible allele and antigen coverage, a certain degree of epitope conservation and the obligatory fixed number of epitopes to select. The constraints are a subset of the constraints of ILP 5.2.

Epitope ordering and spacers. Given the selected epitopes, we employed the scoring function S' (6.3) and the epitope ordering ILP 6.1 to design a string-of-beads construct for each of the three viruses. Subsequently, the constructs were cleaved *in silico*. If a vaccine epitope was not likely to be cleaved properly from the optimized polypeptide we introduced spacers between the epitope and its neighbors to increase the respective epitope's cleavage probability. As spacer we used the sequence `AAY`, which has been shown to be beneficial in EV design [121].

7.3.2 Implementation

A combination of the commercial ILP solver IBM ILOG CPLEX [53], the GNU Linear Programming Kit GLPK [55] and the GNU MathProg modeling language GMPL was used to formulate and solve the ILPs. Runtimes are on the order of a few seconds on standard PC hardware for each ILP employed in this study.

Epitope selection. We used GMPL, a high-level modeling language for mathematical programs, to formulate ILP 7.2. Given a set of target antigen MSAs as well as the set of MHC alleles of interest and their probabilities in the target population, we used a modified version of the Python script described in Section 7.2 to generate consensus sequences with conservation information from a given set of MSAs. The MHC binding affinities, proteasomal cleavage scores, and SIR scores of the consensus sequences were determined using the Peptibase web server [141]. A Python script was implemented to assemble this data along with the required user defined thresholds and the number of epitopes to select into a GMPL data file. The ILP solver GLPK, which takes as input a GMPL model of an ILP and a corresponding GMPL data file, was used to solve the epitope selection ILP.

Epitope ordering. The epitope ordering ILP 6.1 was also formulated as a GMPL model. We used Python to extract the optimal set of epitopes from the GLPK output of the

ILP 7.2: Epitope selection ILP employed for the design of string-of-beads vaccines.

$$\text{maximize} \quad \sum_{q \in Q} \frac{1}{\mathsf{SIR}(q)^2} \sum_{e \in E_q \cap B_a} \mathsf{x_e}\ c(e)\ \sum_{a \in A} p_{mhc}(a)\ i(e,a)$$

$$\text{subject to} \quad \sum_{e \in E} \mathsf{x_e} = k \tag{7.2a}$$

$$\forall e \in E: \quad \mathsf{x_e}\ \tau_c \leq c(e) \tag{7.2b}$$

$$\forall a \in A: \quad \sum_{e \in B_a} \mathsf{x_e} \geq \mathsf{y_a} \tag{7.2c}$$

$$\sum_{a \in A} \mathsf{y_a} \geq \tau_{\mathsf{mhc}} \tag{7.2d}$$

$$\forall q \in Q: \quad \sum_{e \in E_q} \mathsf{x_e} \geq \tau_Q \tag{7.2e}$$

DEFINITIONS

A	Set of observed MHC alleles
Q	Set of target antigens
E_q	Set of epitopes from antigen q
E	Set of all candidate epitopes ($\bigcup_{q \in Q} E_q$)
B_a	Set of epitopes binding to the gene product of MHC allele a ($B_a \subseteq E$)

PARAMETERS

$c(e)$	Conservation of epitope e
$i(e,a)$	Immunogenicity of epitope e with respect to allele a
k	Number of epitopes to select
$p_{mhc}(a)$	Probability of MHC allele a occurring in the target population
$\mathsf{SIR}(q)$	SIR score of antigen q
τ_{mhc}	Minimum number of MHC alleles to be covered
τ_c	Conservation threshold
τ_Q	Minimum number of epitopes from each antigen to be included

VARIABLES

$\mathsf{x_e} = 1$	if epitope e belongs to the optimal set; otherwise $\mathsf{x_e} = 0$
$\mathsf{y_a} = 1$	if allele a is covered by the optimal set; otherwise $\mathsf{y_a} = 0$

previous step, to determine the edge weights of the epitope ordering graph, and to create a GMPL data file. Since the commercial solver CPLEX is significantly faster in solving the epitope ordering problem than the publicly available GLPK, we employed the former to solve the ILP. However, GLPK was required to convert the high-level ILP model and its input data into a CPLEX readable file format. Subsequently, we used the CPLEX command-line interface to solve the optimization problem. Analyses of the results were performed using Python.

7.3.3 Analyses

We first address the question of whether it is feasible to design potent universal string-of-beads vaccines for highly variable viruses. Such vaccines should be applicable to a large fraction of the target population and aim at highly conserved epitopes from various antigens. Focusing on HCV, HIV and IV, our first analysis examines the availability of

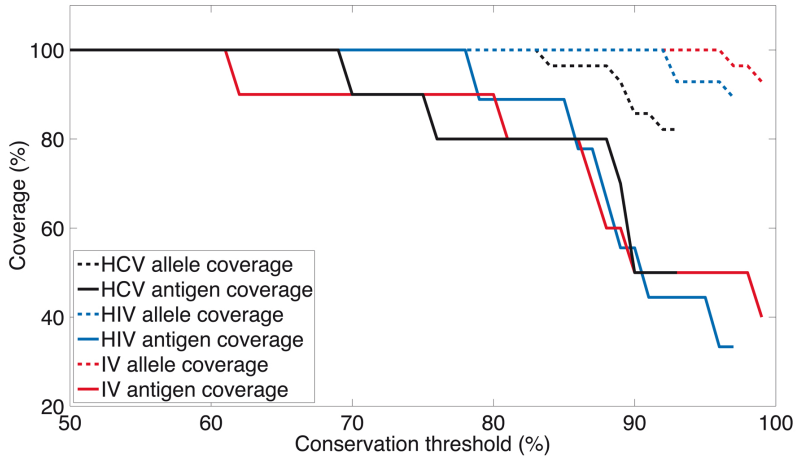


Figure 7.4: Allele and antigen coverage for HCV, HIV and IV considered as a function of the required epitope conservation. Dashed lines show allele coverage, solid lines show antigen coverage for HCV (black), HIV (blue) and IV (red), respectively.

conserved epitopes in the viral proteins. Given the presence of such epitopes, we investigate the possibility of including primarily epitopes from low-epitope-density antigens. Subsequently, having selected an optimal set of epitopes, we address the question whether a string-of-beads construct with favorable cleavage pattern can be produced from these epitopes.

The high variability of the viruses under consideration raises the question to what extent a strong conservation requirement as well as high antigen and allele coverage can be maintained simultaneously. Even in highly variable viruses, some proteins display conserved regions. By limiting the vaccine to epitopes from these regions, its antigen coverage as well as its allele coverage – and therefore its population coverage – may be very limited.

Given all candidate epitopes of each virus, we analyzed the influence of the conservation threshold on the allele and antigen coverage of the selected epitope set. We used the epitope selection ILP 7.2 to select 100 epitope sets of size 30 with increasing conservation threshold from 1% to 100% (Figure 7.4). Coverage of all alleles under consideration, i.e., approximately 91.7% population coverage, is feasible up to a conservation threshold of 83% (HCV), 92% (HIV), and 96% (IV). Full antigen coverage is feasible up to a conservation threshold of 69% (HCV), 78% (HIV), and 61% (IV). Epitope sets covering all alleles under consideration and all but one (two) antigen(s) are feasible up to a conservation threshold of 75% (83%) for HCV, of 85% (87%) for HIV, and of 80% (86%) for IV.

The first MHC alleles to lose coverage are HLA-B*38:01 ($>83\%$), HLA-B*44:03 ($>88\%$), HLA-B*51:02 and HLA-B*58:01 ($>89\%$) as well as HLA-B*40:01 ($>91\%$) for HCV. For HIV, HLA-B*51:02 and HLA-B*27:02 lose coverage at 93% and B*38:01 at 97%. In IV, coverage for HLA-B*38:01 is also lost at 97%, for B*40:01 it is lost at 99%. Noticeably, this list includes only HLA-B alleles. HLA-B alleles have been reported to have less diverse peptide repertoires than HLA-A alleles [142] and to impose greater selection pressure on viral evolution than HLA-A alleles [143, 144].

The first antigens to escape coverage are IV HA ($>62\%$), HCV E1 ($>69\%$), HCV NS2 ($>75\%$), HIV TAT ($>78\%$), IV NA ($>80\%$) and HIV VPU ($>85\%$). HCV E1 as well as IV HA and NA are surface proteins. Surface proteins are exposed to humoral immune pressure and are often reported to be highly variable [133, 145, 146]. HIV TAT and VPU, on the other hand, are early expressed proteins: their exposure to T-cell response is more critical for the virus. It has previously been shown that such proteins have very few epitopes and are subject to a stringent selection [138]. HCV NS2 derives from the NS2/NS3 protease. This protease is the first non-structural (NS) protein translated [147] and, therefore, its exposure is critical. In summary, two groups of proteins lose coverage early: those under B cell-mediated immune selection and those under T cell-mediated immune selection. Losing coverage of the latter group is more problematic since those are probably good targets for a T cell-based vaccine.

As expected, a high conservation threshold interferes with the optimal coverage of target MHCs and antigens. This emphasizes that skillful EV design requires a precise balancing of maintaining a high epitope conservation level and smart targeting at the protein and MHC level.

In addition to high conservation as well as high allele and antigen coverage, it is beneficial for a vaccine to particularly target proteins whose detection would maximize the impact on the virus. Such proteins are probably the ones that the virus has evolved to minimize exposure. We designed our objective function to favor epitopes from proteins with low epitope density by penalizing the selection of epitopes from proteins with high epitope density, i.e., high SIR scores. Requiring full allele coverage, best possible antigen coverage as well as a minimum conservation of 85% (HIV, IV) and 83% (HCV), respectively, 30 epitopes were selected from the candidate epitopes of each virus. For every antigen of the three viruses, Figure 7.5 shows the SIR score, the fraction of epitopes in the selected epitope set derived from the respective antigen, and the fraction of overall immunogenicity (based on equation 7.1) attributed to these epitopes. In the case of HCV (Figure 7.5A) and IV (Figure 7.5C), 50% of the antigens (5/10) have a normalized epitope density below one, i.e., display an unexpectedly low number of epitopes. The majority of the selected epitopes along with the bigger part of immunogenicity is derived from these low-SIR antigens. In the case of HIV (Figure 7.5B), all antigens have a SIR score below one, ranging from 0.23 for TAT to 0.94 for NEF. Therefore, all selected epitopes are derived from proteins with low epitope density. Among these, however, the vast majority (23/30) are derived from POL, which displays a comparatively high epitope density. Each of the remaining seven epitopes comes from one of the other antigens except for TAT. TAT, the HIV antigen with the lowest epitope density, does not display sufficient conservation to contribute epitopes to the selected set. Further analysis revealed that POL contributes 75% (122/162) of the candidate epitopes for HIV. Among these are several highly immunogenic ones, resulting in the predominance of POL epitopes in the set of selected HIV epitopes. This high density of highly conserved epitopes may be explained by the late expression and low copy number of POL [148, 149]. It has been argued that T-cell exposure of proteins translated at a later stage of the infection may not critically impair HIV, as some virions may have the time to bud before the cell is destroyed [136]. Furthermore, the low copy number of POL will result

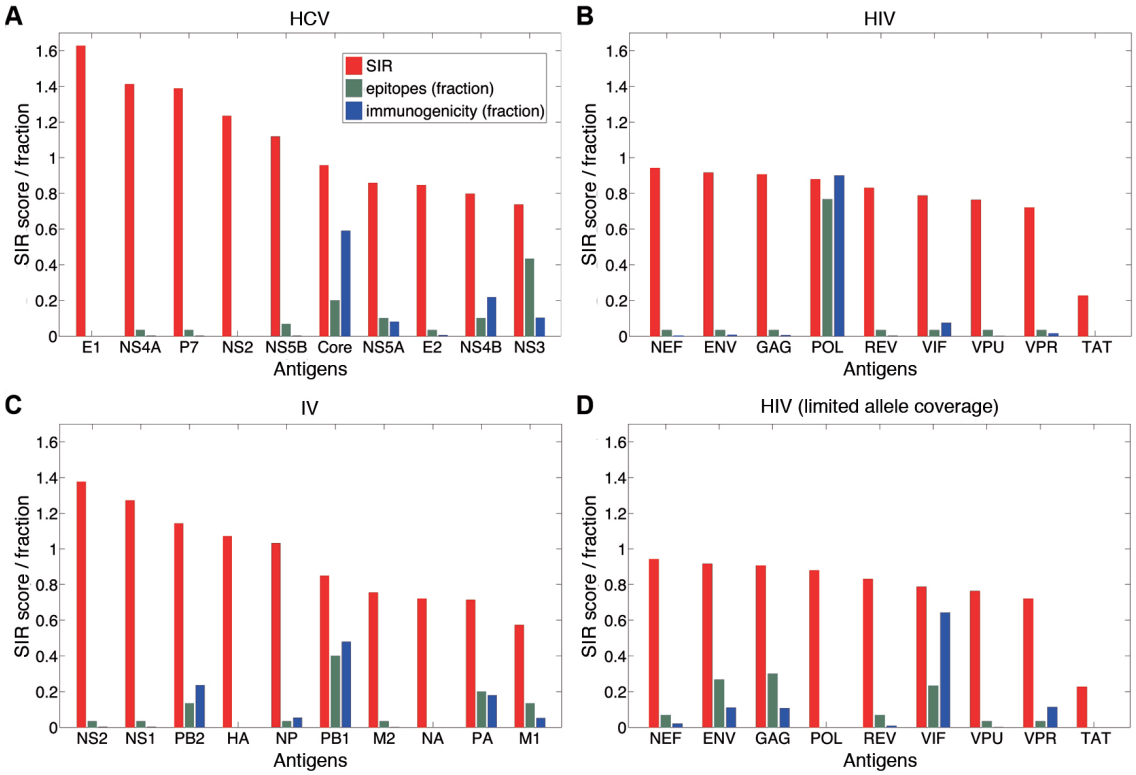


Figure 7.5: Antigen coverage of selected epitope sets. For each virus, a set of 30 epitopes, given a minimum conservation, was selected. For HIV, an additional set excluding POL was selected. SIR score, the fraction of epitopes from antigens in the set and the fraction of contribution to overall immunogenicity are displayed for each viral protein individually. Viral proteins are ordered by decreasing SIR score. A) HCV, conservation threshold: 83%, full allele coverage, no coverage of E1 and NS2. B) HIV, conservation threshold: 85%, full allele coverage, no coverage of TAT. C) IV, conservation threshold: 85%, full allele coverage, no coverage of HA and NA. D) HIV, conservation threshold: 85%, limited allele coverage (25 of 28), no coverage of TAT and POL.

in a comparatively low total number of epitopes expressed. Therefore, the selection against high-affinity epitopes in POL is less stringent than in other HIV proteins. Obviously, late expression and low copy number also make POL a poor EV target. It may thus be beneficial to limit the fraction of POL epitopes in the vaccine. We repeated the epitope selection for HIV excluding epitopes from POL. In order to do so, the requirement of full allele coverage had to be omitted. Without POL, only 25 out of the 28 HLA alleles can be covered, yielding a population coverage of approximately 90.6%. Figure 7.5D shows that the exclusion of POL epitopes results in a considerably more balanced antigen coverage. The fraction of epitopes from low-SIR antigens (REV, VIF, VPU, VPR) has increased from about 13% to 37% (11/30).

Given an optimal epitope set for each virus, we designed string-of-beads constructs optimized for proteasomal cleavage of the respective epitopes. Epitope recovery was evaluated

based on an *in silico* cleavage of every polypeptide. In the optimal configuration, all HCV epitopes included in the vaccine were contained in the resulting list of cleaved nonameric peptides while two epitopes from HIV and one from IV were missing. We introduced spacers next to these epitopes in the HIV and IV polypeptides resulting in full epitope recovery. Besides full recovery, vaccine epitopes should be more likely to result from proteasomal cleavage than other unwanted MHC binding peptides. In order to evaluate the polypeptides in this regard, we considered the ratio between the average cleavage scores of vaccine epitopes and of unwanted epitopes. If vaccine epitopes are more likely to be cleaved than other epitopes the ratio is greater than 1.0. The higher the ratio, the more favorable the cleavage pattern. The cleavage score ratios range from 1.4 for HIV to 2.0 for IV and 2.2 for HCV. Although the ratio of the HIV polypeptide is comparably low, all string-of-beads constructs yield favorable cleavage patterns.

7.3.4 Discussion

Traditional approaches to vaccine design had limited success in protecting against chronic infection with viruses such as HIV and HCV. While the resulting vaccines have been shown to be effective against IV, their efficacy is usually limited to a certain strain as defined by the hemagglutinin (HA) and neuraminidase (NA) groups. Furthermore, the time needed to manufacture such an IV vaccine is an issue: in a best case scenario, it takes five to six months for a vaccine against a new IV strain to become available [150]. EVs represent a promising alternative.

The main goal of this work is to analyze whether it is feasible to simultaneously optimize the protein targeting and the epitope quality in EVs against highly variable viruses. Such vaccines should be applicable to a large fraction of the target population and aim at highly conserved epitopes from various antigens. As has been shown for several viruses, proteins with low epitope density are probably good targets. Another important factor when designing an EV targeted at a particular geographic area is strain prevalence. Since our focus is on viruses in the world population, we do not consider it here. However, strain prevalence can easily be incorporated into the proposed epitope selection approach.

Our first analysis examined the availability of conserved epitopes in the viral proteins. In the second analysis, we investigated the possibility of including primarily epitopes from low-SIR antigens. Subsequently, having selected an optimal set of epitopes, we addressed the question whether a polypeptide with a favorable cleavage pattern can be produced from these epitopes. For three highly variable viruses, we have shown that it is indeed possible to select epitope sets that will yield potent vaccines. However, in general, such sets cannot cover all viral antigens. Specifically, this concerns highly mutable antigens, such as the envelope proteins of HCV and IV, or proteins subject to high evolutionary pressure, such as the HIV regulatory proteins. In the case of HIV, the majority of the optimal epitopes are derived from POL. However, since POL is expressed late and at low copy numbers, it may be a very poor target. In order to obtain better targets, it may therefore be advantageous to limit the number of POL-derived epitopes at the cost of a decrease in predicted immunogenicity.

Once an optimal set of epitopes is chosen, they have to be properly ordered for the string-of-beads construct. The optimal epitope order meets two important complementary criteria: (1) it ensures the cleavage of the selected epitopes by the proteasome and (2) it minimizes the probability of a cleavage of junctional epitopes, which could induce an immune response that would be useless against the targeted virus. We employed the ILP proposed in Chapter 6 (ILP 6.1) for epitope assembly. For the vast majority of vaccine epitopes, the optimal ordering does not require the use of spacers to ensure proper cleavage. However, using the epitope ordering ILP to directly design an optimal string-of-beads construct with spacers between all vaccine epitopes promises to yield even better results.

This study combines a large number of elements, which have not been considered in conjunction before: proper treatment of the epitope preprocessing by the proteasome [24, 102], optimal choice of antigenic targets and the optimal positioning of the selected epitopes [24]. The combination of these elements significantly raises the probability of producing clinically relevant vaccines.

Chapter 8

Discussion & Conclusion

Parts of this chapter have previously been published [8].

Traditional trial-and-error based approaches to vaccine design have been remarkably successful. One of the major successes was the eradication of smallpox in the 1970s. However, there are still many diseases for which no viable vaccine could be found, HIV infection and cancer being among the most prominent examples. Here, new, rationally designed types of vaccines, as, e.g., EVs, are promising alternatives.

The process of designing an EV can roughly be divided into three steps: epitope discovery, epitope selection and epitope assembly. *In silico* epitope discovery has the potential to drastically reduce the number of biological experiments that have to be performed. A key step in *in silico* approaches to epitope discovery is the prediction of MHC-binding peptides. This problem has been tackled with the full range of possible mathematical methods. Mostly machine learning methods have proven useful. However, the prediction of binders for MHC alleles with little or no experimental data is problematic. In Sections 4.2 and 4.3 we present two approaches to overcome the problem of scarcity of data. Our first approach improves the predictive power of SVMs for alleles with little experimental binding data by combining the benefits of string kernels with the ones of physicochemical descriptors for AAs. Our second approach, UniTope, for the first time allowed predictions for all MHC-I alleles. This was achieved by employing all available MHC-I binding data for the prediction of an individual allele's binding specificity. The kernels developed in our first approach are particularly useful when data is scarce. Next to allele-specific MHC binding prediction they promise to be beneficial in, e.g., the prediction of substrate specificities within enzyme families [151]. Use of these kernels in UniTope (Section 4.3), however, will not yield further improvements: Due to the pooling of all available binding data, training data is far from being scarce in this setting.

Pan-specific approaches like UniTope have contributed significantly to the advancement of *in silico* epitope discovery. Nevertheless, it has to be noted that there is a drawback to current approaches: for a considerable fraction of alleles pan-specific models perform worse than allele-specific models, i.e., inclusion of knowledge on other alleles' binding specificities is disadvantageous. Here, models trained for a specific allele incorporating a limited number of examples from highly related other alleles [152] show promise. However, this approach brings about another drawback: when confronted with a new MHC allele a

new model has to be trained, which can be very time consuming. Making such a method publicly available, e.g., via a web server, to offer predictions for all known MHC alleles is hence inapplicable. Alternative approaches are required and can be expected to come from the field of *transfer learning*. Transfer learning deals with transferring knowledge from one task to another task. It is a hot topic in the machine learning community [153–155]. Increasingly sophisticated transfer learning methods can be expected to improve pan-specific MHC binding prediction while at the same time being able to handle the increasing amounts of MHC binding data.

Today, despite minor shortcomings, the problem of predicting whether or how strongly a peptide binds to the product of a given MHC allele can be considered as solved. However, this is not the case for epitope discovery in general. The actual prediction of T-cell epitopes, i.e., to distinguish immunogenic from non-immunogenic MHC binders, is still in its infancy. It will be one of the key issues for immunoinformatics to address in the near future. Due to the complex host dependencies accounting for the T-cell recognition of a pMHC complex, the problem is particularly challenging. Previous methods do not consider these dependencies but utilize peptide sequence information only yielding limited prediction accuracies. In Section 4.4 we present a novel approach to T-cell epitope prediction that takes a systemic view on the problem. Evaluation on a small unbiased and allele-specific data set showed promising results. While the small sample allows demonstration of a proof-of-concept, a more thorough analysis of the individual aspects of our approach requires more data.

Reliable immunogenicity prediction is a prerequisite for fully rational EV design. The move from simple sequence-based predictors to predictors including system-wide properties promises to be a step in the right direction. However, in order for T-cell epitope prediction to eventually come of age, sufficiently large and consistent data sets for various alleles need to become publicly available. Here, the development of high-throughput methods to experimentally identify T-cell epitopes [156] will play an important role.

In this thesis, we only consider naturally occurring peptides. There have also been studies successfully using epitope analogs, namely fixed anchor and heteroclitic epitopes [25]. While fixed anchor approaches aim at increasing pMHC binding affinity by modification of MHC anchor residues, the heteroclitic epitopes approach aims at increasing pMHC:TCR affinity by modification of TCR interaction sites. The capability of such epitope analogs to induce more potent immune responses than the corresponding wild-type epitopes as well as to break T-cell tolerance has been shown in several studies [157, 158]. For the design of epitope analogs, it would be desirable to have accurate prediction methods that complement their prediction with information, for example, on what AA properties are important for a specific position. Such information cannot be easily retrieved from state-of-the-art prediction methods. It has to be noted that the use of modified epitopes is controversial. Induction of T-cell populations with low recognition efficiencies [159] and the appearance of cryptic epitopes with dominant immunogenicity [160] have been reported.

In EV design, epitope discovery is followed by epitope selection. Out of a given set of candidate epitopes, the vaccine epitopes have to be selected. Depending on the requirements on the desired epitope set, this task can become highly complex, rendering manual

selection of an optimal epitope set virtually impossible. Hence, efficient computational approaches to optimal epitope selection are called for. In Chapter 5, we present the first framework for epitope selection that is capable of finding the optimal epitope set with respect to a wide variety of different requirements. The method performs better than existing solutions and has runtimes of a few seconds. With OptiTope (Section 7.1) we provide an easy-to-use web-interface to this framework making it available to immunologists. While the flexibility of our framework allows for the incorporation of different requirements on the desired vaccine, more data is needed to actually reveal a suitable quality measure and the properties of a good vaccine. To allow for advances in this area, it is essential to make the data of previous, current and future clinical trials of EVs publicly available.

Depending on the chosen delivery strategy, the epitope selection step is followed by epitope assembly. While there is no commonly agreed on delivery strategy, many delivery strategies employed in (pre-)clinical trials are based on a concatenation of the individual epitopes, either with or without spacer, as DNA, RNA or as AA sequence. Here, the chosen epitope order strongly influences the success of the vaccine. Computational methods for epitope assembly can help to determine a favorable order efficiently. In Chapter 6, we propose the first *in silico* approach to optimally assemble epitopes into a string-of-beads vaccine. Based on a predictor for proteasomal cleavage, the resulting vaccine constructs are guaranteed to be optimal with respect to epitope recovery. The more accurate the proteasomal cleavage predictor, the better the resulting construct. In this thesis, we focus on $CD8^+$ T cells and their epitopes. The epitope ordering algorithm can in principal also be used to assemble $CD4^+$ T-cell epitopes. However, to the best of our knowledge there is no predictor for the extracellular antigen processing pathway available. Given the lack of a suitable predictor, the assembly approach taken by De Groot *et al.* [23], i.e., constructing extended peptides by overlapping epitope is expedient when dealing with $CD4^+$ T-cell epitopes.

The EV design heuristic proposed by Vider-Shalit *et al.* [24] combines epitope selection and assembly. While we have only considered an iterative approach – epitope selection followed by epitope assembly – it is also possible to simultaneously optimize these two EV design aspects via ILP. In contrast to the iterative procedure, which clearly focuses on the optimal epitope selection, the combined optimization would allow for a stronger emphasis on antigen processing. The relative importance of each aspect could be controlled via the objective function.

In Sections 7.2 and 7.3, we applied our epitope selection and assembly approaches to realistic vaccine design problems. The resulting vaccines are optimal with respect to the definition of good vaccine and yield perfect epitope recovery. In our study on string-of-beads vaccines against highly variable viruses (Section 7.3) we briefly touch on the topic of target antigen detection. It is generally agreed upon that some antigens are more suitable targets than others. Hence, a ranking of antigens should be incorporated when selecting epitopes for anti-virus EVs. So far, the only published approach to *in silico* ranking of viral antigens is the SIR score [137]. The authors have shown for multiple viruses that – at least in theory – only proteins expressed at critical time periods or at a high copy number, i.e., suitable vaccine targets, have a low SIR score. However, the actual explanatory power

of the score is controversial. For *in silico* EV design it would be desirable to have a widely accepted score. The constantly increasing number of sequenced viruses provides a sound basis for further statistical analyses in this regard. While based on real data, the results of the vaccine design studies are theoretical in nature. To complement them, it would be interesting to experimentally evaluate the vaccines and thus the performance of the proposed algorithms. Here, the development of a mouse model to efficiently test vaccine constructs and delivery strategies in a high-throughput manner would be of great help.

Ever decreasing sequencing costs and improving analysis methods will contribute greatly to advances in vaccine design. HLA typing will become considerably cheaper allowing for high-resolution clinical data. Furthermore, inexpensive and rapid sequencing of, e.g., cancer genomes is an important step towards an era of personalized medicine.

In the long term the use of *in silico* epitope discovery, selection and assembly, e.g., as part of a vaccine design pipeline from sequencing data to the EV components, will significantly change the way vaccines are developed. The use of standardized approaches for EVs will solve some of the issues of classical vaccines based on attenuated pathogens. In particular, it promises shorter development times, which can be essential for emerging infectious diseases. At the same time, it is indispensable for fully personalized vaccines, particularly for cancer immunotherapy.

Appendix A

Abbreviations

AA	Amino acid
AASK	Amino acid substring kernel
APC	Antigen-presenting cell
auROC	Area under the receiver operating characteristics curve
CTL	Cytotoxic T lymphocyte
EBV	Epstein-Barr virus
EV	Epitope-based vaccine
HCV	Hepatitis C virus
HIV	Human immunodeficiency virus
HLA	Human leukocyte antigen
IC_{50}	50% inhibitory concentration
ILP	Integer linear program
IV	Influenza virus
LKH	Lin-Kernighan heuristic
LP	Linear program
MHC	Major histocompatibility complex
MHC-I	Major histocompatibility complex class I
MHC-II	Major histocompatibility complex class II
MSA	Multiple sequence alignment
PCC	Pearson correlation coefficient
pMHC	Peptide:MHC complex
SIR	Size of immune repertoire
SVC	Support vector classification
SVM	Support vector machine
SVR	Support vector regression
TAP	Transporter associated with antigen processing
TCR	T-cell receptor
TSP	Travelling salesman problem
WD	Weighted degree

Appendix B

Epitope Discovery

Table B.1: IEDB^{h9} data set. The IEDB^{h9} data set is a subset of the IEDB benchmark data set [86]. It comprises the binding data of nonameric peptides with respect to human MHC-I alleles. The table lists the number of examples per allele.

Allele	Examples	Allele	Examples
HLA-A*01:01	1157	HLA-B*07:02	1262
HLA-A*02:01	3089	HLA-B*08:01	708
HLA-A*02:02	1447	HLA-B*15:01	978
HLA-A*02:03	1443	HLA-B*18:01	118
HLA-A*02:06	1437	HLA-B*27:05	969
HLA-A*03:01	2094	HLA-B*35:01	736
HLA-A*11:01	1985	HLA-B*40:01	1078
HLA-A*23:01	104	HLA-B*40:02	118
HLA-A*24:02	197	HLA-B*44:02	119
HLA-A*24:03	254	HLA-B*44:03	119
HLA-A*26:01	672	HLA-B*45:01	114
HLA-A*29:02	160	HLA-B*51:01	244
HLA-A*30:01	669	HLA-B*53:01	254
HLA-A*30:02	92	HLA-B*54:01	255
HLA-A*31:01	1869	HLA-B*57:01	59
HLA-A*33:01	1140	HLA-B*58:01	988
HLA-A*68:01	1141		
HLA-A*68:02	1434		
HLA-A*69:01	833		

Table B.2: Classification performances of WD-RBF and WD kernel. The table lists the mean auROCs obtained by a two-times nested five-fold cross validation of a WD-RBF (blosum50) and a WD kernel on the IEDB^{h9} allele subsets. The best performance per allele is highlighted (bold).

Allele	WD-RBF	WD
HLA-A*01:01	**0.967**	0.966
HLA-A*02:01	0.949	0.949
HLA-A*02:02	**0.881**	0.879
HLA-A*02:03	**0.903**	0.896
HLA-A*02:06	**0.913**	0.908
HLA-A*03:01	0.923	**0.924**
HLA-A*11:01	0.938	**0.940**
HLA-A*23:01	**0.813**	0.744
HLA-A*24:02	0.733	**0.748**
HLA-A*24:03	0.840	**0.844**
HLA-A*26:01	**0.926**	0.902
HLA-A*29:02	0.906	0.906
HLA-A*30:01	**0.931**	0.908
HLA-A*30:02	**0.802**	0.780
HLA-A*31:01	**0.921**	0.916
HLA-A*33:01	**0.905**	0.898
HLA-A*68:01	**0.856**	0.846
HLA-A*68:02	0.877	**0.879**
HLA-A*69:01	**0.869**	0.816
HLA-B*07:02	**0.947**	0.946
HLA-B*08:01	**0.917**	0.867
HLA-B*15:01	**0.913**	0.909
HLA-B*18:01	0.895	**0.905**
HLA-B*27:05	**0.931**	0.927
HLA-B*35:01	**0.863**	0.850
HLA-B*40:01	**0.937**	0.931
HLA-B*40:02	**0.781**	0.721
HLA-B*44:02	0.710	**0.749**
HLA-B*44:03	0.783	**0.806**
HLA-B*45:01	**0.871**	0.836
HLA-B*51:01	**0.869**	0.846
HLA-B*53:01	0.870	0.870
HLA-B*54:01	**0.884**	0.842
HLA-B*57:01	0.934	0.934
HLA-B*58:01	**0.949**	0.936

Table B.3: Regression performances of WD-RBF and WD kernel. The table lists the mean PCCs obtained by a two-times nested five-fold cross validation of a WD-RBF (blosum50) and a WD kernel on the IEDB^{h9} allele subsets. The best performance per allele is highlighted (bold).

Allele	WD-RBF	WD
HLA-A*01:01	**0.706**	0.705
HLA-A*02:01	**0.815**	0.810
HLA-A*02:02	**0.770**	0.765
HLA-A*02:03	**0.793**	0.770
HLA-A*02:06	**0.770**	0.753
HLA-A*03:01	**0.742**	0.733
HLA-A*11:01	**0.802**	0.793
HLA-A*23:01	**0.686**	0.609
HLA-A*24:02	**0.620**	0.617
HLA-A*24:03	0.587	**0.599**
HLA-A*26:01	**0.578**	0.561
HLA-A*29:02	**0.840**	0.799
HLA-A*30:01	**0.672**	0.668
HLA-A*30:02	0.574	**0.581**
HLA-A*31:01	**0.757**	0.753
HLA-A*33:01	**0.672**	0.671
HLA-A*68:01	**0.739**	0.736
HLA-A*68:02	**0.756**	0.749
HLA-A*69:01	**0.533**	0.518
HLA-B*07:02	**0.796**	0.786
HLA-B*08:01	**0.533**	0.512
HLA-B*15:01	0.709	**0.713**
HLA-B*18:01	**0.761**	0.752
HLA-B*27:05	**0.714**	0.684
HLA-B*35:01	**0.668**	0.658
HLA-B*40:01	0.557	**0.561**
HLA-B*40:02	**0.748**	0.734
HLA-B*44:02	0.614	**0.653**
HLA-B*44:03	0.792	**0.799**
HLA-B*45:01	**0.742**	0.706
HLA-B*51:01	0.708	**0.725**
HLA-B*53:01	0.737	**0.743**
HLA-B*54:01	**0.744**	0.727
HLA-B*57:01	**0.742**	0.698
HLA-B*58:01	**0.742**	0.709

Table B.4: Crystal structures employed to generate the pocket profiles for UniTope. PDB-IDs of the nonameric pMHC-I structures employed to generate the pocket profiles are listed per allele.

Allele	PDB-IDs
HLA-A*02:01	1AKJ, 1AO7, 1B0G, 1BD2, 1DUZ, 1EEY, 1EEZ, 1HHG, 1HHI, 1HHJ, 1HHK, 1I1F, 1I1Y, 1I7R, 1I7T, 1I7U, 1IM3, 1JHT, 1LP9, 1OGA, 1P7Q, 1QEW, 1QR1, 1QRN, 1QSE, 1S8D, 1S9W, 1S9X, 1S9Y, 1T1W, 1T1X, 1T1Y, 1T1Z, 1T20, 1T21, 1T22, 1TVB, 1TVH, 2BNQ, 2BNR, 2BSU, 2BSV, 2C7U, 2F53, 2F54, 2GIT
HLA-A*11:01	1Q94, 1X7Q
HLA-A*24:02	2BCK
HLA-B*08:01	1M05
HLA-B*15:01	1XR8, 1XR9
HLA-B*27:05	1HSA, 1JGE, 1W0V, 2A83, 2BSR, 2BST
HLA-B*27:09	1K5N, 1W0W
HLA-B*35:01	1A9B, 1A9E, 1CG9, 2CIK, 2H6P
HLA-B*44:02	1M6O
HLA-B*44:03	1N2R, 1SYS
HLA-B*44:05	1SYV
HLA-B*51:01	1E27
HLA-B*53:01	1A1M, 1A1O
HLA-B*57:03	2BVP
HLA-Cw*04:01	1IM9, 1QQD

Table B.5: Overall performance of MHC-I binding predictors on seen alleles. The table lists the allele-wise PCCs of the pan-specific predictors NetMHCpan and UniTope as well as of the allele-specific predictor ANN. Performance is measured in a five-fold cross validation on the IEDB^{h9} data and the given folds.

Allele	NetMHCpan	UniTope	ANN
HLA-A*01:01	0.840	0.602	0.450
HLA-A*02:01	0.893	0.842	0.536
HLA-A*02:02	0.856	0.842	0.557
HLA-A*02:03	0.862	0.838	0.616
HLA-A*02:06	0.859	0.843	0.702
HLA-A*03:01	0.845	0.788	0.426
HLA-A*11:01	0.878	0.843	0.537
HLA-A*23:01	0.855	0.780	0.494
HLA-A*24:02	0.735	0.566	0.304
HLA-A*24:03	0.732	0.433	0.420
HLA-A*26:01	0.716	0.506	0.358
HLA-A*29:02	0.871	0.811	0.636
HLA-A*30:01	0.784	0.675	0.330
HLA-A*30:02	0.725	0.615	0.393
HLA-A*31:01	0.835	0.773	0.415
HLA-A*33:01	0.769	0.655	0.553
HLA-A*68:01	0.822	0.725	0.634
HLA-A*68:02	0.811	0.703	0.656
HLA-A*69:01	0.726	0.565	0.491
HLA-B*07:02	0.850	0.707	0.410
HLA-B*08:01	0.602	0.386	0.346
HLA-B*15:01	0.800	0.543	0.613
HLA-B*18:01	0.813	0.666	0.832
HLA-B*27:05	0.700	0.467	0.546
HLA-B*35:01	0.763	0.649	0.398
HLA-B*40:01	0.667	0.389	0.187
HLA-B*40:02	0.840	0.818	0.805
HLA-B*44:02	0.745	0.647	0.735
HLA-B*44:03	0.797	0.720	0.718
HLA-B*45:01	0.683	0.659	0.767
HLA-B*51:01	0.785	0.630	0.636
HLA-B*53:01	0.825	0.809	0.746
HLA-B*54:01	0.817	0.740	0.642
HLA-B*57:01	0.783	0.645	0.524
HLA-B*58:01	0.842	0.663	0.590
Average	**0.792**	**0.673**	**0.543**

Table B.6: Leave-one-out validation performance of pan-specific MHC-I binding predictors. The table lists the allele-wise PCCs of the pan-specific predictors NetMHCpan and UniTope in a leave-one-out validation. Performance is measured in a five-fold cross validation on the IEDB^{h9} data and the given folds.

Allele	NetMHCpan	UniTope
HLA-A*01:01	0.378	0.215
HLA-A*02:01	0.857	0.798
HLA-A*02:02	0.805	0.782
HLA-A*02:03	0.795	0.809
HLA-A*02:06	0.803	0.817
HLA-A*03:01	0.760	0.734
HLA-A*11:01	0.835	0.736
HLA-A*23:01	0.811	0.672
HLA-A*24:02	0.647	0.507
HLA-A*24:03	0.729	0.228
HLA-A*26:01	0.535	0.408
HLA-A*29:02	0.692	0.553
HLA-A*30:01	0.633	0.628
HLA-A*30:02	0.559	0.478
HLA-A*31:01	0.695	0.661
HLA-A*33:01	0.638	0.636
HLA-A*68:01	0.688	0.513
HLA-A*68:02	0.721	0.592
HLA-A*69:01	0.711	0.546
HLA-B*07:02	0.516	0.292
HLA-B*08:01	0.385	0.043
HLA-B*15:01	0.443	0.357
HLA-B*18:01	0.624	0.474
HLA-B*27:05	-0.057	0.010
HLA-B*35:01	0.656	0.618
HLA-B*40:01	0.426	0.278
HLA-B*40:02	0.769	0.815
HLA-B*44:02	0.802	0.753
HLA-B*44:03	0.731	0.797
HLA-B*45:01	0.556	0.570
HLA-B*51:01	0.668	0.635
HLA-B*53:01	0.753	0.801
HLA-B*54:01	0.614	0.513
HLA-B*57:01	0.740	0.575
HLA-B*58:01	0.453	0.224
Average	**0.639**	**0.545**

Table B.7: Gene expression omnibus data sets used to define the thymus proteome.

Series	Sample	Microarray
GSE96	GSM2825	Affymetrix HG-U95A
	GSM2826	Affymetrix HG-U95A
GSE803	GSM12683	Affymetrix HG-U95A
	GSM12684	Affymetrix HG-U95B
	GSM12685	Affymetrix HG-U95C
	GSM12686	Affymetrix HG-U95D
	GSM12687	Affymetrix HG-U95E
GSE803	GSM12713	Affymetrix HG-U95A
	GSM12714	Affymetrix HG-U95B
	GSM12715	Affymetrix HG-U95C
	GSM12716	Affymetrix HG-U95D
	GSM12717	Affymetrix HG-U95E
GSE2361	GSM44672	Affymetrix HG-U133A

Table B.8: ArrayExpress data sets used to define the thymus proteome.

Series	Sample	Microarray
E-AFMX-5	3AJZ02022758 thymus	Affymetrix HG-U133A
	3AJZ02031411 thymus	Affymetrix HG-U133A
	1BJZ02022642 thymus	Affymetrix gnGNF1Ba
	1BJZ02022757 thymus	Affymetrix gnGNF1Ba
E-CBIL-1	thymus 1 HG-U95B-AG23-biotin-G2500A	Affymetrix HG-U95B
	thymus 2 HG-U95B-AG42-biotin-G2500A	Affymetrix HG-U95B
	thymus 1 HG-U95C-AG23-biotin-G2500A	Affymetrix HG-U95C
	thymus 2 HG-U95C-AG42-biotin-G2500A	Affymetrix HG-U95C
	thymus 1 HG-U95D-AG23-biotin-G2500A	Affymetrix HG-U95D
	thymus 2 HG-U95D-AG42-biotin-G2500A	Affymetrix HG-U95D
	thymus 1 HG-U95E-AG23-biotin-G2500A	Affymetrix HG-U95E
	thymus 2 HG-U95E-AG42-biotin-G2500A	Affymetrix HG-U95E
E-MTAB-24	3AJZ02022758 thymus.CHP	Affymetrix HG-U133A
	3AJZ02031411 thymus.CHP	Affymetrix HG-U133A
	1BJZ02022642 thymus.CHP	Affymetrix gnGNF1Ba
	1BJZ02022757 thymus.CHP	Affymetrix gnGNF1Ba
E-MTAB-25	1BJZ02022757 thymus.CHP	Affymetrix gnGNF1Ba
	1BJZ02022757 thymus.CHP	Affymetrix gnGNF1Ba
	1BJZ02022757 thymus.CHP	Affymetrix gnGNF1Ba
	1BJZ02022757 thymus.CHP	Affymetrix gnGNF1Ba
E-TABM-145	1BJZ02022757 thymus.CHP	Affymetrix HG-U133A
	1BJZ02022757 thymus.CHPAffymetrix HG-U133A	

Algorithm B.1: Weighted majority voting. This voting is applied to resolve conflicts in results of microarray experiments. Based on the number of absent, marginal and present assignments a protein is unambiguously assigned to exactly one class. The marginal classifications were weighted only 2/3 to achieve a better separation in absent and present, reducing the number of marginal assignments.

$a \Leftarrow$ number of absent calls
$m \Leftarrow$ number of marginal calls
$p \Leftarrow$ number of present calls
$sum \Leftarrow a + \frac{2}{3}m + p$
$label \Leftarrow$ unassigned
if $a/sum > \frac{2}{3}$ **then**
  $label \Leftarrow$ absent
else
  if $p/sum > \frac{2}{3}$ **then**
    $label \Leftarrow$ present
  else
    $label \Leftarrow$ marginal
  endif
endif

Appendix C

Applications

Table C.1: List of all MHC alleles used in the study on designing string-of-beads vaccines. The employed probabilities and BIMAS score thresholds are given.

MHC allele	Threshold	Probability (in %)
HLA-A*01:01	0.087	4.50
HLA-A*02:01	1.257	10.69
HLA-A*02:05	0.418	0.88
HLA-A*03:01	0.055	3.69
HLA-A*11:01	0.044	7.52
HLA-A*24:02	1.158	12.91
HLA-A*31:01	0.0964	2.43
HLA-A*68:01	2.399	1.77
HLA-B*04:01	5.224	1.55
HLA-B*07:02	1.623	3.61
HLA-B*08:01	0.072	2.95
HLA-B*15:01	0.217	2.06
HLA-B*27:02	50.043	0.15
HLA-B*27:05	163.868	1.11
HLA-B*35:01	3.202	3.24
HLA-B*37:01	1.424	0.44
HLA-B*38:01	7.929	0.66
HLA-B*39:01	4.734	1.77
HLA-B*40:01	21.178	5.31
HLA-B*40:06	1.044	0.52
HLA-B*44:03	6.209	2.21
HLA-B*51:01	39.207	3.24
HLA-B*51:02	155.050	0.22
HLA-B*52:01	8.589	0.88
HLA-B*58:01	13.356	2.65
HLA-Cw*04:01	4.061	8.26
HLA-Cw*06:02	2.169	5.09
HLA-Cw*07:02	62.183	9.66

Appendix D

Contributions

Chapter 4 — Epitope Discovery

Section 4.2 — Improved Kernels for MHC Binding Prediction

This work is part of a manuscript that has previously been published in *BMC Bioinformatics* [71]. It is joint work with Gunnar Rätsch (GR) and Oliver Kohlbacher (OK). NCT and GR conceived and designed the project. NCT prepared the data, implemented the kernels and performed experiments. GR supervised the project. OK contributed to the discussion. The contributions of Christian Widmer to the publication as well as GR's implementations and experiments have not been incorporated in this thesis.

Section 4.3 — MHC Binding Prediction for All MHC-I Alleles

In addition to myself, Magdalena Feldhahn (MF), Oliver Kohlbacher (OK) and Matthias Ziehm (MZ) contributed to this work. OK conceived the project. NCT designed and performed the experiments. MZ retrieved the 3D structures and determined the pocket profiles. MF and OK contributed to the discussion.

Section 4.4 — T-Cell Epitope Prediction

In addition to myself, Magdalena Feldhahn (MF), Sebastian Briesemeister (SB), Matthias Ziehm (MZ), Gunnar Rätsch (GR), Stefan Stevanovic (SS) and Oliver Kohlbacher (OK) contributed to this project. NCT and OK designed the experiments. NCT performed the experiments and retrieved and preprocessed the data. SS provided experimental data. MZ retrieved the microarray data and determined the thymus proteomes. OK and MF implemented the distance tries. MF performed the distance-to-self calculations and the MHC binding predictions. MF, SB, GR and OK contributed to the discussion. Parts of this section were presented at the *Second Immunoinformatics and Computational Immunology Workshop (ICIW 2011)* and are included in the workshop proceedings [161].

Chapter 5 — Epitope Selection

This work is part of a manuscript that has previously been published in *PLoS Computational Biology* [102]. It is joint work with Pierre Dönnes (PD) and Oliver Kohlbacher (OK). NCT, PD and OK conceived and designed the experiments. NCT performed the experiments. NCT and OK analyzed the data.

Chapter 6 — Epitope Assembly

This work is part of a manuscript that has previously been published in *Vaccine* [162]. It is joint work with Yaakov Maman (YM), Oliver Kohlbacher (OK) and Yoram Louzoun (YL). The contributions of YM have not been incorporated in this chapter. NCT and YL conceived and designed the experiments. NCT conceived the algorithm and performed the experiments. OK contributed to the discussion.

Chapter 7 — Applications

Section 7.1 — OptiTope – A Web Server for Epitope Selection

This work has previously been published in the web server issue of *Nucleic Acids Research* [103]. It is joint work with Oliver Kohlbacher (OK). NCT and OK designed the structure of the web server. NCT implemented the web server.

Section 7.2 — Design of a Peptide Cocktail Vaccine

This work is part of a manuscript that has previously been published in *PLoS Computational Biology* [102]. It is joint work with Pierre Dönnes (PD) and Oliver Kohlbacher (OK). NCT, PD and OK conceived and designed the experiments. NCT performed the experiments. NCT and OK analyzed the data.

Section 7.3 — Design of String-of-Beads Vaccines

This work is part of a manuscript that has previously been published in *Vaccine* [162]. It is joint work with Yaakov Maman (YM), Oliver Kohlbacher (OK) and Yoram Louzoun (YL). NCT and YL conceived and designed the experiments. NCT performed the experiments. OK contributed to the discussion. YM retrieved the influenza sequences. YL and YM performed the biological analysis.

Appendix E

Publications

Published Manuscripts

- **N. C. Toussaint**, P. Dönnes, and O. Kohlbacher. A mathematical framework for the selection of an optimal set of peptides for epitope-based vaccines. *PLoS Computational Biology*, 4(12):e1000246, 2008.
  - Text and figures from this manuscript appear in Chapters 1 & 5 and in Section 7.2 of this thesis.

- **N. C. Toussaint** and O. Kohlbacher. Towards in silico design of epitope-based vaccines. *Expert Opinion on Drug Discovery*, 4(10):1047-1060, 2009.
  - Text and figures from this manuscript appear throughout this thesis.

- **N. C. Toussaint** and O. Kohlbacher. OptiTope – a web server for the selection of an optimal set of peptides for epitope-based vaccines. *Nucleic Acids Research*, 37:W617-22, 2009.
  - Text and figures from this manuscript appear in Section 7.1 of this thesis.

- **N. C. Toussaint**, C. Widmer, O. Kohlbacher, and G. Rätsch. Exploiting physicochemical properties in string kernels. *BMC Bioinformatics*, 11(Suppl 8):S7, 2010.
  - Text and figures from this manuscript appear in Chapters 1 & 3 and in Section 4.2 of this thesis.

- C. Widmer*, **N. C. Toussaint***, Y. Altun, O. Kohlbacher, and G. Rätsch. Novel machine learning methods for MHC class I binding prediction. In: *Proceedings of PRIB 2010, Lecture Notes in Computer Sciences*, 2010.
  *) Joint first author.

- C. Widmer, Y. Altun, **N. C. Toussaint**, and G. Rätsch. Inferring latent task structure for multi-task learning via multiple kernel learning. *BMC Bioinformatics*, 11(Suppl 8):S5, 2010.

- A. Hildebrandt, A. K. Dehof, A. Rurainski, A. Bertsch, M. Schumann, **N. C. Toussaint**, A. Moll, D. Stöckel, S. Nickels, S. C. Mueller, H.-P. Lenhof, and O. Kohlbacher. BALL - Biochemical algorithms library 1.3. *BMC Bioinformatics*, 11:531, 2010.

- S. Canzar, **N. C. Toussaint**, and G. W. Klau. An exact algorithm for side-chain placement in protein design. *Optimization Letters*, 1-14, 2011.

- **N. C. Toussaint**, M. Feldhahn, M. Ziehm, S. Stevanović, and O. Kohlbacher. T-cell epitope prediction based on self-tolerance. In: *Proceedings of the Second Immunoinformatics and Computational Immunology Workshop*, 2011.
  - Text and figures from this manuscript appear in Chapter 1 and in Section 4.4 of this thesis.

- **N. C. Toussaint**, Y. Maman, O. Kohlbacher, and Y. Louzoun. Universal peptide vaccines – optimal peptide vaccine design based on viral sequence conservation. *Vaccine*, 2011 Aug 27 [Epub ahead of print].
  - Text and figures from this manuscript appear in Chapters 1 & 6 and in Section 7.3 of this thesis.

Bibliography

[1] M. Moutschen, P. Lonard, E.M. Sokal, F. Smets, M. Haumont, P. Mazzu, A. Bollen, F. Denamur, P. Peeters, G. Dubin, and M. Denis. Phase I/II studies to evaluate safety and immunogenicity of a recombinant gp350 Epstein-Barr virus vaccine in healthy adults. *Vaccine*, 25(24):4697–4705, 2007.

[2] P. A. Goepfert, G. D. Tomaras, H. Horton, D. Montefiori, G. Ferrari, M. Deers, G. Voss, M. Koutsoukos, L. Pedneault, P. Vandepapeliere, M. J. McElrath, P. Spearman, J. D Fuchs, B. A. Koblin, W. A. Blattner, S. Frey, L. R. Baden, Clayton Harro, Thomas Evans, and NIAID HIV Vaccine Trials Network. Durable HIV-1 antibody and T-cell responses elicited by an adjuvanted multi-protein recombinant vaccine in uninfected human volunteers. *Vaccine*, 25(3):510–518, 2007.

[3] M. Mancini-Bourgine, H. Fontaine, C. Bréchot, S. Pol, and M.-L. Michel. Immunogenicity of a hepatitis B DNA vaccine administered to chronic HBV carriers. *Vaccine*, 24(21):4482–4489, 2006.

[4] J. Nemunaitis, T. Meyers, N. Senzer, C. Cunningham, Howard West, Eric Vallieres, S. Anthony, S. Vukelja, B. Berman, H. Tully, B. Pappen, S. Sarmiento, R. Arzaga, S. Duniho, S. Engardt, M. Meagher, and M. A. Cheever. Phase I trial of sequential administration of recombinant DNA and adenovirus expressing L523S protein in early stage non-small-cell lung cancer. *Mol Ther*, 13(6):1185–1191, 2006.

[5] K. A. Chianese-Bullock, W. P. Irvin, G. R. Petroni, C. Murphy, M. Smolkin, W. C. Olson, E. Coleman, S. A. Boerner, C. J. Nail, P. Y. Neese, A. Yuan, K. T. Hogan, and C. L. Slingluff. A multipeptide vaccine is safe and elicits T-cell responses in participants with advanced stage ovarian cancer. *J Immunother*, 31(4):420–430, 2008.

[6] G. G. Kenter, M. J. P. Welters, A. R. P. M. Valentijn, M. J. G. Lowik, D. M. A. Berends van der Meer, A. P. G. Vloon, J. W. Drijfhout, A. R. Wafelman, J. Oostendorp, G. J. Fleuren, R. Offringa, S. H. van der Burg, and C. J. M. Melief. Phase I immunotherapeutic trial with long peptides spanning the E6 and E7 sequences of high-risk human papillomavirus 16 in end-stage cervical cancer patients shows low toxicity and robust immunogenicity. *Clin Cancer Res*, 14(1):169–177, 2008.

[7] C. L. Slingluff, G. R. Petroni, K. A. Chianese-Bullock, M. E. Smolkin, S. Hibbitts, C. Murphy, N. Johansen, W. W. Grosh, G. V. Yamshchikov, P. Y. Neese, J. W.

Patterson, R. Fink, and P. K. Rehm. Immunologic and clinical outcomes of a randomized phase II trial of two multipeptide vaccines for melanoma in the adjuvant setting. *Clin Cancer Res*, 13(21):6386–6395, 2007.

[8] N. C. Toussaint and O. Kohlbacher. Towards in silico design of epitope-based vaccines. *Expert Opin Drug Discovery*, 4(10):1047–1060, 2009.

[9] 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. *J Immunol*, 153(12):5586–5592, 1994.

[10] H. Singh-Jasuja, N. P. N. Emmerich, and H.-G. Rammensee. The Tübingen approach: identification, selection, and validation of tumor-associated HLA peptides for cancer therapy. *Cancer Immunol Immun*, 53(3):187–195, 2004.

[11] K. C. Parker, M. A. Bednarek, and J. E. Coligan. Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. *J Immunol*, 152(1):163–175, 1994.

[12] H. Rammensee, J. Bachmann, N. P. N. Emmerich, O. A. Bachor, and S. Stevanović. SYFPEITHI: database for MHC ligands and peptide motifs. *Immunogenetics*, 50(3-4):213–219, 1999.

[13] P. Dönnes and A. Elofsson. Prediction of MHC class I binding peptides, using SVMHC. *BMC Bioinf*, 3:25, 2002.

[14] S. Buus, S. L. Lauemøller, P. Worning, C. Kesmir, T. Frimurer, S. Corbet, A. Fomsgaard, J. Hilden, A. Holm, and S. Brunak. Sensitive quantitative predictions of peptide-MHC binding by a 'Query by Committee' artificial neural network approach. *Tissue Antigens*, 62(5):378–384, 2003.

[15] J. Robinson, A. Malik, P. Parham, J. G. Bodmer, and S. G. Marsh. IMGT/HLA database–a sequence database for the human major histocompatibility complex. *Tissue Antigens*, 55(3):280–287, 2000.

[16] J. Robinson, K. Mistry, H. McWilliam, R. Lopez, P. Parham, and S. G. Marsh. The IMGT/HLA database. *Nucleic Acids Res*, 39(Database issue):D1171–6, 2011.

[17] J. Cui, L. Y. Han, H. H. Lin, Z. Q. Tang, L. Jiang, Z. W. Cao, and Y. Z. Chen. MHC-BPS: MHC-binder prediction server for identifying peptides of flexible lengths from sequence-derived physicochemical properties. *Immunogenetics*, 58(8):607–613, 2006.

[18] W. Liu, X. Meng, Q. Xu, D. R. Flower, and T. Li. Quantitative prediction of mouse class I MHC peptide binding affinity using support vector machine regression (SVR) models. *BMC Bioinf*, 7:182, 2006.

[19] T. Sturniolo, E. Bono, J. Ding, L. Raddrizzani, O. Tuereci, U. Sahin, M. Braxenthaler, F. Gallazzi, M. P. Protti, F. Sinigaglia, and J. Hammer. Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. *Nat Biotechnol*, 17(6):555–561, 1999.

[20] M. Nielsen, C. Lundegaard, T. Blicher, K. Lamberth, M. Harndahl, S. Justesen, G. Røder, B. Peters, A. Sette, O. Lund, and S. Buus. NetMHCpan, a method for quantitative predictions of peptide binding to any HLA-A and -B locus protein of known sequence. *PLoS ONE*, 2(8):e796, 2007.

[21] M. Bhasin and G. P. S. Raghava. Prediction of CTL epitopes using QM, SVM and ANN techniques. *Vaccine*, 22(23-24):3195–3204, 2004.

[22] C.-W. Tung and S.-Y. Ho. POPI: predicting immunogenicity of MHC class I binding peptides by mining informative physicochemical properties. *Bioinformatics*, 23(8):942–949, 2007.

[23] A. S. De Groot, L. Marcon, E. A. Bishop, D. Rivera, M. Kutzler, D. B. Weiner, and W. Martin. HIV vaccine development by computer assisted design: the GAIA vaccine. *Vaccine*, 23(17-18):2136–2148, 2005.

[24] T. Vider-Shalit, S. Raffaeli, and Y. Louzoun. Virus-epitope vaccine design: informatic matching the HLA-I polymorphism to the virus genome. *Mol Immunol*, 44(6):1253–1261, 2007.

[25] A. Sette and J. Fikes. Epitope-based vaccines: an update on epitope identification, vaccine design and delivery. *Curr Opin Immunol*, 15(4):461–470, 2003.

[26] R. A. Goldsby, T. J. Kindt, B. A. Osborne, and J. Kuby. *Immunology*. W. H. Freeman and Company, New York, 5th edition, 2002.

[27] C. A. Janeway Jr., P. Travers, M. Walport, and M. J. Shlomchik. *Immunobiology: the immune system in health and disease*. Garland Science Publishing, New York, 6th edition, 2005.

[28] J. Sidney, B. Peters, N. Frahm, C. Brander, and A. Sette. HLA class I supertypes: a revised and updated classification. *BMC Immunol*, 9:1, 2008.

[29] A. Moll, A. Hildebrandt, H. P. Lenhof, and O. Kohlbacher. BALLView: a tool for research and education in molecular modeling. *Bioinformatics*, 22(3):365–6, 2006.

[30] M. Nielsen, C. Lundegaard, O. Lund, and C. Keşmir. The role of the proteasome in generating cytotoxic T-cell epitopes: insights obtained from improved predictions of proteasomal cleavage. *Immunogenetics*, 57(1-2):33–41, 2005.

[31] L. Klein, M. Hinterberger, G. Wirnsberger, and B. Kyewski. Antigen presentation in the thymus for positive selection and central tolerance induction. *Nat Rev Immunol*, 9(12):833–44, 2009.

[32] K. A. Hogquist, T. A. Baldwin, and S. C. Jameson. Central tolerance: learning self-control in the thymus. *Nat Rev Immunol*, 5(10):772–82, 2005.

[33] K. Wing and S. Sakaguchi. Regulatory T cells exert checks and balances on self tolerance and autoimmunity. *Nat Immunol*, 11(1):7–13, 2010.

[34] H. G. Rammensee, T. Friede, and S. Stevanović. MHC ligands and peptide motifs: first listing. *Immunogenetics*, 41(4):178–228, 1995.

[35] R. A. Seder, P. A. Darrah, and M. Roederer. T-cell quality in memory and protection: implications for vaccine design. *Nat Rev Immunol*, 8(4):247–258, 2008.

[36] C. Scheibenbogen, K. H. Lee, S. Mayer, S. Stevanović, U. Moebius, W. Herr, H.-G. Rammensee, and U. Keilholz. A sensitive ELISPOT assay for detection of CD8+ T lymphocytes specific for HLA class I-binding peptide epitopes derived from influenza proteins in the blood of healthy donors and melanoma patients. *Clin Cancer Res*, 3(2):221–6, 1997.

[37] J. D. Altman, P. A. Moss, P. J. Goulder, D. H. Barouch, M. G. McHeyzer-Williams, J. I. Bell, A. J. McMichael, and M. M. Davis. Phenotypic analysis of antigen-specific T lymphocytes. *Science*, 274(5284):94–6, 1996.

[38] S. Depil, O. Moralès, F. A. Castelli, N. Delhem, V. François, B. Georges, F. Dufossé, F. Morschhauser, J. Hammer, B. Maillère, C. Auriault, and V. Pancré. Determination of a HLA II promiscuous peptide cocktail as potential vaccine against EBV latency II malignancies. *J Immunother*, 30(2):215–226, 2007.

[39] N. Yajima, R. Yamanaka, T. Mine, N. Tsuchiya, J. Homma, M. Sano, T. Kuramoto, Y. Obata, N. Komatsu, Y. Arima, A. Yamada, M. Shigemori, K. Itoh, and R. Tanaka. Immunologic evaluation of personalized peptide vaccination for patients with advanced malignant glioma. *Clin Cancer Res*, 11(16):5900–5911, 2005.

[40] C. L. Slingluff, G. R. Petroni, G. V. Yamshchikov, D. L. Barnd, S. Eastham, H. Galavotti, J. W. Patterson, D. H. Deacon, S. Hibbitts, D. Teates, P. Y. Neese, W. W. Grosh, K. A. Chianese-Bullock, E. M. H. Woodson, C. J. Wiernasz, P. Merrill, J. Gibson, M. Ross, and V. H Engelhard. Clinical and immunologic results of a randomized phase II trial of vaccination using four melanoma peptides either administered in granulocyte-macrophage colony-stimulating factor in adjuvant or pulsed on dendritic cells. *J Clin Oncol*, 21(21):4016–4026, 2003.

[41] A. W. Purcell, J. McCluskey, and J. Rossjohn. More than one reason to rethink the use of peptides in vaccine design. *Nat Rev Drug Discov*, 6(5):404–414, 2007.

[42] E. Celis. Getting peptide vaccines to work: just a matter of quality control? *J Clin Invest*, 110(12):1765–8, 2002.

[43] K. Deres, H. Schild, K. H. Wiesmüller, G. Jung, and H.-G. Rammensee. In vivo priming of virus-specific cytotoxic T lymphocytes with synthetic lipopeptide vaccine. *Nature*, 342(6249):561–4, 1989.

[44] F. Morein and K. L. Bengtsson. Functional aspects of ISCOMs. *Immunol Cell Biol*, 76(4):295–9, 1998.

[45] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Discrete Mathematics and Optimization. Wiley-VCH, 1st edition, 1999.

[46] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2001.

[47] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Schölkopf, and G. Rätsch. Support vector machines and kernels for computational biology. *PLoS Comput Biol*, 4(10):e1000173, 2008.

[48] A. Schrijver. *Combinatorial Optimization*. Algorithms and Combinatorics. Springer, 2003.

[49] K. Menger. Eine neue Definition der Bogenlänge. In K. Menger, editor, *Ergebnisse eines Mathematischen Kolloquiums*, volume 2, pages 11–12. Teubner, Leipzig, 1932.

[50] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[51] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull Amer Math Soc*, 64(5):275–278, 1958.

[52] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

[53] International Business Machines Corp. IBM ILOG CPLEX, version 11.1, http://www.ilog.com.

[54] MOSEK ApS. Mosek, http://www.mosek.com.

[55] GNU Linear Programming Kit GLPK, version 4.32, http://www.gnu.org/software/glpk.

[56] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[57] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[58] G. Schweikert, A. Zien, G. Zeller, J. Behr, C. Dieterich, C. S. Ong, P. Philips, F. De Bona, L. Hartmann, A. Bohlen, N. Krüger, S. Sonnenburg, and G. Rätsch. mGene: accurate SVM-based gene finding with an application to nematode genomes. *Genome Res*, 19(11):2133–43, 2009.

[59] A. Höglund, P. Dönnes, T. Blum, H. W. Adolph, and O. Kohlbacher. MultiLoc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs and amino acid composition. *Bioinformatics*, 22(10):1158–65, 2006.

[60] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.

[61] C. Cortes and V. Vapnik. Support-vector networks. *Mach Learn*, 20(3):273–297, 1995.

[62] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.

[63] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA, 1992.

[64] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446, 1909.

[65] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 564–575, 2002.

[66] P. Meinicke, M. Tech, B. Morgenstern, and R. Merkl. Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. *BMC Bioinf*, 5(169), 2004.

[67] G. Rätsch and S. Sonnenburg. Accurate splice site detection for Caenorhabditis elegans. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*, pages 277–298. MIT Press, 2004.

[68] R. M. Clark, G. Schweikert, C. Toomajian, S. Ossowski, G. Zeller, P. Shinn, N. Warthmann, T. T. Hu, G. Fu, D. A. Hinds, H. Chen, K. A. Frazer, D. H. Huson, B. Schölkopf, M. Nordborg, G. Rätsch, J. R. Ecker, and D. Weigel. Common sequence polymorphisms shaping genetic diversity in Arabidopsis thaliana. *Science*, 317(5836):338–42, 2007.

[69] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[70] C. Lundegaard, O. Lund, C. Kesmir, S. Brunak, and M. Nielsen. Modeling the adaptive immune system: predictions and simulations. *Bioinformatics*, 23(24):3265–75, 2007.

[71] N. C. Toussaint, C. Widmer, O. Kohlbacher, and G. Rätsch. Exploiting physicochemical properties in string kernels. *BMC Bioinf*, 11 Suppl 8:S7, 2010.

[72] C. Widmer, N. C. Toussaint, Y. Altun, O. Kohlbacher, and G. Rätsch. Novel machine learning methods for MHC class I binding prediction. In T. Dijkstra, E. Tsivtsivadze, E. Marchiori, and T. Heskes, editors, *Pattern Recognition in Bioinformatics*, volume 6282 of *Lecture Notes in Computer Science*, pages 98–109. Springer Berlin / Heidelberg, 2010.

[73] P. Dönnes and O. Kohlbacher. Integrated modeling of the major events in the MHC class I antigen processing pathway. *Protein Sci*, 14(8):2132–2140, 2005.

[74] P. Dönnes and O. Kohlbacher. SVMHC: a server for prediction of MHC-binding peptides. *Nucleic Acids Res*, 34(Web Server issue):W194–W197, 2006.

[75] M. Nielsen, C. Lundegaard, P. Worning, S. L. Lauemøller, K. Lamberth, S. Buus, S. Brunak, and O. Lund. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Sci*, 12(5):1007–1017, 2003.

[76] J. Robinson, M. J. Waller, S. C. Fail, H. McWilliam, R. Lopez, P. Parham, and S. G. Marsh. The IMGT/HLA database. *Nucleic Acids Res*, 37(Database issue):D1013–7, 2009.

[77] D. Rognan, S. L. Lauemøller, A. Holm, S. Buus, and V. Tschinke. Predicting binding affinities of protein ligands from three-dimensional models: application to peptide binding to class I major histocompatibility proteins. *J Med Chem*, 42(22):4650–4658, 1999.

[78] O. Schueler-Furman, Y. Altuvia, A. Sette, and H. Margalit. Structure-based prediction of binding peptides to MHC class I molecules: application to a broad range of MHC alleles. *Protein Sci*, 9(9):1838–1846, 2000.

[79] D. S. DeLuca, B. Khattab, and R. Blasczyk. A modular concept of HLA for comprehensive peptide binding prediction. *Immunogenetics*, 59(1):25–35, 2007.

[80] G. Chelvanayagam. A roadmap for HLA-A, HLA-B, and HLA-C peptide binding specificities. *Immunogenetics*, 45(1):15–26, 1996.

[81] G. E. Meister, C. G. Roberts, J. A. Berzofsky, and A. S. De Groot. Two novel T cell epitope prediction algorithms based on MHC-binding motifs; comparison of predicted and published epitopes from Mycobacterium tuberculosis and HIV protein sequences. *Vaccine*, 13(6):581–591, 1995.

[82] S. Kawashima, H. Ogata, and M. Kanehisa. AAindex: Amino acid index database. *Nucleic Acids Res*, 27(1):368–369, 1999.

[83] H. Rangwala and G. Karypis. Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21(23):4239–4247, 2005.

[84] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Proceedings IEEE Computational Systems Bioinformatics Conference*, 2004.

[85] B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, USA, 1999.

[86] B. Peters, H.-H. Bui, S. Frankild, M. Nielsen, C. Lundegaard, E. Kostem, D. Basch, K. Lamberth, M. Harndahl, W. Fleri, S. S. Wilson, J. Sidney, O. Lund, S. Buus, and A. Sette. A community resource benchmarking predictions of peptide binding to MHC-I molecules. *PLoS Comput Biol*, 2(6):e65, 2006.

[87] B. Peters, J. Sidney, P. Bourne, H.-H. Bui, S. Buus, G. Doh, W. Fleri, M. Kronenberg, R. Kubo, O. Lund, D. Nemazee, J.V. Ponomarenko, M. Sathiamurthy, S. Schoenberger, S. Stewart, P. Surko, S. Way, S. Wilson, and A. Sette. The immune epitope database and analysis resource: from vision to blueprint. *PLoS Biol*, 3(3):e91, 2005.

[88] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *P Natl Acad Sci USA*, 89(22):10915–10919, 1992.

[89] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, and V. Franc. The SHOGUN machine learning toolbox. *J Mach Learn Res*, 11:1799–1802, 2010.

[90] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res*, 28(1):235–42, 2000.

[91] O. Kohlbacher and H. P. Lenhof. BALL–rapid software prototyping in computational molecular biology. *Bioinformatics*, 16(9):815–24, 2000.

[92] J. Robinson, M. J. Waller, P. Parham, N. de Groot, R. Bontrop, L. J. Kennedy, P. Stoehr, and S. G. E. Marsh. IMGT/HLA and IMGT/MHC: sequence databases for the study of the major histocompatibility complex. *Nucleic Acids Res*, 31(1):311–314, 2003.

[93] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–80, 1994.

[94] M.S. Venkatarajan and W. Braun. New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical–chemical properties. *J Mol Model*, 7:445–453, 2001.

[95] P. J. Kersey, J. Duarte, A. Williams, Y. Karavidopoulou, E. Birney, and R. Apweiler. The international protein index: an integrated database for proteomics experiments. *Proteomics*, 4(7):1985–8, 2004.

[96] T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, D. Rudnev, C. Evangelista, I. F. Kim, A. Soboleva, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, R. N. Muertter, and R. Edgar. NCBI GEO: archive for high-throughput functional genomic data. *Nucleic Acids Res*, 37(Database issue):D885–90, 2009.

[97] H. Parkinson, M. Kapushesky, N. Kolesnikov, G. Rustici, M. Shojatalab, N. Abeygunawardena, H. Berube, M. Dylag, I. Emam, A. Farne, E. Holloway, M. Lukk, J. Malone, R. Mani, E. Pilicheva, T. F. Rayner, F. Rezwan, A. Sharma, E. Williams, X. Z. Bradley, T. Adamusiak, M. Brandizi, T. Burdett, R. Coulson, M. Krestyaninova, P. Kurnosov, E. Maguire, S. G. Neogi, P. Rocca-Serra, S. A. Sansone, N. Sklyar, M. Zhao, U. Sarkans, and A. Brazma. ArrayExpress update – from an archive of functional genomics experiments to the atlas of gene expression. *Nucleic Acids Res*, 37(Database issue):D868–72, 2009.

[98] E. Fredkin. Trie memory. *Commun. ACM*, 3:490–499, 1960.

[99] POPI web server, http://iclab.life.nctu.edu.tw/POPI.

[100] W. L. Redmond and L. A. Sherman. Peripheral tolerance of CD8 T lymphocytes. *Immunity*, 22(3):275–84, 2005.

[101] D. L. Mueller. Mechanisms maintaining peripheral tolerance. *Nat Immunol*, 11(1):21–7, 2010.

[102] N. C. Toussaint, P. Dönnes, and O. Kohlbacher. A mathematical framework for the selection of an optimal set of peptides for epitope-based vaccines. *PLoS Comput Biol*, 4(12):e1000246, 2008.

[103] N. C. Toussaint and O. Kohlbacher. OptiTope – a web server for the selection of an optimal set of peptides for epitope-based vaccines. *Nucleic Acids Res*, 37(Web Server issue):W617–22, 2009.

[104] J. R. Schafer, B. M. Jesdale, J. A. George, N. M. Kouttab, and A. S. De Groot. Prediction of well-conserved HIV-1 ligands using a matrix-based algorithm, EpiMatrix. *Vaccine*, 16(19):1880–1884, 1998.

[105] J. I. Krieger, R. W. Karr, H. M. Grey, W. Y. Yu, D. O'Sullivan, L. Batovsky, Z. L. Zheng, S. M. Colon, F. C. Gaeta, J. Sidney, M. Albertson, M.-F. Del Guercio, R. W. Chesnut, and A. Sette. Single amino acid changes in DR and antigen define residues critical for peptide-MHC binding and T cell recognition. *J Immunol*, 146(7):2331–40, 1991.

[106] C. L. Kingsford, B. Chazelle, and M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics*, 21(7):1028–1036, 2005.

[107] I. E. Grossmann. Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization and Engineering*, 3:227–252, 2002.

[108] M. R. Bussieck and S. Vigerske. *MINLP Solver Software*. John Wiley & Sons, Inc., 2010.

[109] ILOG, Inc. ILOG CPLEX, version 9.1, http://www.ilog.com.

[110] F. X. Ding, F. Wang, Y. M. Lu, K. Li, K. H. Wang, X. W. He, and S. H. Sun. Multiepitope peptide-loaded virus-like particles as a vaccine against hepatitis B virus-related hepatocellular carcinoma. *Hepatology*, 49(5):1492–502, 2009.

[111] W. Fischer, S. Perkins, J. Theiler, T. Bhattacharya, K. Yusim, R. Funkhouser, C. Kuiken, B. Haynes, N. L. Letvin, B. D. Walker, B. H. Hahn, and B. T. Korber. Polyvalent vaccines for optimal coverage of potential T-cell epitopes in global HIV-1 variants. *Nat Med*, 13(1):100–106, 2007.

[112] D. C. Nickle, M. Rolland, M. A. Jensen, S. L. K. Pond, W. Deng, M. Seligman, D. Heckerman, J. I. Mullins, and N. Jojic. Coping with viral diversity in HIV vaccine design. *PLoS Comput Biol*, 3(4):e75, 2007.

[113] G. Pataki. Teaching integer programming formulations using the traveling salesman problem. *SIAM review*, 45(1):116–123, 2003.

[114] K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *Eur J Oper Res*, 126(1):106–130, 2000.

[115] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7:326–329, 1960.

[116] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Oper Res*, 21(March 1, 1973):498–516, 1973.

[117] C. Keşmir, A. K. Nussbaum, H. Schild, V. Detours, and S. Brunak. Prediction of proteasome cleavage motifs by neural networks. *Protein Eng*, 15(4):287–296, 2002.

[118] H. G. Holzhütter and P. M. Kloetzel. A kinetic model of vertebrate 20S proteasome accounting for the generation of major proteolytic fragments from oligomeric peptide substrates. *Biophys J*, 79(3):1196–1205, 2000.

[119] I. Ginodi, T. Vider-Shalit, L. Tsaban, and Y. Louzoun. Precise score for the prediction of peptides cleaved by the proteasome. *Bioinformatics*, 24(4):477–483, 2008.

[120] B. D. Livingston, M. Newman, C. Crimi, D. McKinney, R. Chesnut, and A. Sette. Optimization of epitope processing enhances immunogenicity of multiepitope DNA vaccines. *Vaccine*, 19(32):4652–60, 2001.

[121] M. P. Velders, S. Weijzen, G. L. Eiben, A. G. Elmishad, P. M. Kloetzel, T. Higgins, R. B. Ciccarelli, M. Evans, S. Man, L. Smith, and W. M. Kast. Defined flanking spacers and enhanced proteolysis is essential for eradication of established tumors by an epitope string DNA vaccine. *J Immunol*, 166(9):5366–73, 2001.

[122] D. Meyer, R. M. Singe, S. J. Mack, A. Lancaster, M. P. Nelson, H. Erlich, M. Fernandez-Vina, and G. Thomson. Single locus polymorphism of classical HLA genes. In *Immunobiology of the human MHC: proceedings of the 13th International Histocompatibility Workshop and Conference*, volume 1, pages 653–704, 2007.

[123] NCBI dbMHC database, http://www.ncbi.nlm.nih.gov/gv/mhc/.

[124] M. Feldhahn, P. Thiel, M. M. Schuler, N. Hillen, S. Stevanović, H.-G. Rammensee, and O. Kohlbacher. EpiToolKit – a web server for computational immunomics. *Nucleic Acids Res*, 36(Web Server issue):W519–22, 2008.

[125] Zope application server, http://www.zope.org.

[126] Plone, http://plone.org.

[127] C. Kuiken, K. Yusim, L. Boykin, and R. Richardson. The Los Alamos hepatitis C sequence database. *Bioinformatics*, 21(3):379–384, 2005.

[128] R. C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, 32(5):1792–1797, 2004.

[129] J. Grebely, D. L. Thomas, and G. J. Dore. HCV reinfection studies and the door to vaccine development. *J Hepatol*, 51(4):628–31, 2009.

[130] D. Y. Jin. Molecular pathogenesis of hepatitis C virus-associated hepatocellular carcinoma. *Front Biosci*, 12:222–33, 2007.

[131] N. D. Russell, B. S. Graham, M. C. Keefer, M. J. McElrath, S. G. Self, K. J. Weinhold, D. C. Montefiori, G. Ferrari, H. Horton, G. D. Tomaras, S. Gurunathan, L. Baglyos, S. E. Frey, M. J. Mulligan, C. D. Harro, S. P. Buchbinder, L. R. Baden, W. A. Blattner, B. A. Koblin, and L. Corey. Phase 2 study of an HIV-1 canarypox vaccine (vCP1452) alone and in combination with rgp120: negative results fail to trigger a phase 3 correlates trial. *J Acquir Immune Defic Syndr*, 44(2):203–12, 2007.

[132] D. G. Bowen and C. M. Walker. Mutational escape from CD8+ T cell immunity: HCV evolution, from chimpanzees to man. *J Exp Med*, 201(11):1709–14, 2005.

[133] M. Liu, H. Chen, F. Luo, P. Li, Q. Pan, B. Xia, Z. Qi, W. Z. Ho, and X. L. Zhang. Deletion of N-glycosylation sites of hepatitis C virus envelope protein E1 enhances specific cellular and humoral immune responses. *Vaccine*, 25(36):6572–80, 2007.

[134] V. Peut and S. J. Kent. Fitness constraints on immune escape from HIV: Implications of envelope as a target for both HIV-specific T cells and antibody. *Curr HIV Res*, 4(2):191–7, 2006.

[135] L. Uebelhoer, J. H. Han, B. Callendret, G. Mateu, N. H. Shoukry, H. L. Hanson, C. M. Rice, C. M. Walker, and A. Grakoui. Stable cytotoxic T cell escape mutation in hepatitis C virus is linked to maintenance of viral fitness. *PLoS Pathog*, 4(9):e1000143, 2008.

[136] T. Vider-Shalit, M. Almani, R. Sarid, and Y. Louzoun. The HIV hide and seek game: an immunogenomic analysis of the HIV epitope repertoire. *AIDS*, 23(11):1311–8, 2009.

[137] T. Vider-Shalit, V. Fishbain, S. Raffaeli, and Y. Louzoun. Phase-dependent immune evasion of herpesviruses. *J Virol*, 81(17):9536–45, 2007.

[138] T. Vider-Shalit, R. Sarid, K. Maman, L. Tsaban, R. Levi, and Y. Louzoun. Viruses selectively mutate their CD8+ T-cell epitopes–a large-scale immunomic analysis. *Bioinformatics*, 25(12):i39–i44, 2009.

[139] Los Alamos HIV sequence database, http://www.hiv.lanl.gov/.

[140] NCBI website, http://www.ncbi.nlm.nih.gov/.

[141] Peptibase web server, http://peptibase.cs.biu.ac.il/peptibase/.

[142] X. Rao, A. I. Costa, D. van Baarle, and C. Kesmir. A comparative study of HLA binding affinity and ligand diversity: implications for generating immunodominant CD8+ T cell responses. *J Immunol*, 182(3):1526–32, 2009.

[143] F. Bihl, N. Frahm, L. Di Giammarino, J. Sidney, M. John, K. Yusim, T. Woodberry, K. Sango, H. S. Hewitt, L. Henry, C. H. Linde, J. V. Chisholm 3rd, T. M. Zaman, E. Pae, S. Mallal, B. D. Walker, A. Sette, B. T. Korber, D. Heckerman, and C. Brander. Impact of HLA-B alleles, epitope binding affinity, functional avidity, and viral coinfection on the immunodominance of virus-specific CTL responses. *J Immunol*, 176(7):4094–101, 2006.

[144] P. Kiepiela, A. J. Leslie, I. Honeyborne, D. Ramduth, C. Thobakgale, S. Chetty, P. Rathnavalu, C. Moore, K. J. Pfafferott, L. Hilton, P. Zimbwa, S. Moore, T. Allen, C. Brander, M. M. Addo, M. Altfeld, I. James, S. Mallal, M. Bunce, L. D. Barber, J. Szinger, C. Day, P. Klenerman, J. Mullins, B. Korber, H. M. Coovadia, B. D. Walker, and P. J. Goulder. Dominant influence of HLA-B in mediating the potential co-evolution of HIV and HLA. *Nature*, 432(7018):769–75, 2004.

[145] G. M. Air and W. G. Laver. The neuraminidase of influenza virus. *Proteins*, 6(4):341–56, 1989.

[146] U. Gulati, C. C. Hwang, L. Venkatramani, S. Gulati, S. J. Stray, J. T. Lee, W. G. Laver, A. Bochkarev, A. Zlotnick, and G. M. Air. Antibody epitopes on the neuraminidase of a recent H3N2 influenza virus (A/Memphis/31/98). *J Virol*, 76(23):12274–80, 2002.

[147] S. Welbourn and A. Pause. The hepatitis C virus NS2/3 protease. *Curr Issues Mol Biol*, 9(1):63–9, 2007.

[148] T. Jacks, M. D. Power, F. R. Masiarz, P. A. Luciw, P. J. Barr, and H. E. Varmus. Characterization of ribosomal frameshifting in HIV-1 gag-pol expression. *Nature*, 331(6153):280–3, 1988.

[149] W. Wilson, M. Braddock, S. E. Adams, P. D. Rathjen, S. M. Kingsman, and A. J. Kingsman. HIV expression strategies: ribosomal frameshifting is directed by a short sequence in both mammalian and yeast systems. *Cell*, 55(6):1159–69, 1988.

[150] World Health Organization. Pandemic influenza vaccine manufacturing process and timeline, 2009. Retrieved from http://www.who.int on June 2, 2010.

[151] M. Röttig, C. Rausch, and O. Kohlbacher. Combining structure and sequence information allows automated prediction of substrate specificities within enzyme families. *PLoS Comput Biol*, 6(1):e1000636, 2010.

[152] N. Pfeifer and O. Kohlbacher. Multiple instance learning allows MHC class II epitope predictions across alleles. In *Algorithms in Bioinformatics*, volume 5251 of *Lecture Notes in Computer Science*, pages 210–221. Springer, 2008.

[153] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng. Boosted multi-task learning. *Mach Learn*, pages 1–25, 2010. 10.1007/s10994-010-5231-6.

[154] C. Widmer, N. C. Toussaint, Y. Altun, and G. Rätsch. Inferring latent task structure for multitask learning by multiple kernel learning. *BMC Bioinf*, 11(Suppl 8):S5, 2010.

[155] T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *J Mach Learn Res*, 6(1):615–637, 2005.

[156] S. R. Hadrup, M. Toebes, B. Rodenko, A. H. Bakker, D. A. Egan, H. Ovaa, and T. N. Schumacher. High-throughput T-cell epitope discovery through MHC peptide exchange. *Methods Mol Biol*, 524:383–405, 2009.

[157] R. Dyall, W. B. Bowne, L. W. Weber, J. LeMaoult, P. Szabo, Y. Moroi, G. Piskun, J. J. Lewis, A. N. Houghton, and J. Nikolić-Zugić. Heteroclitic immunization induces tumor immunity. *J Exp Med*, 188(9):1553–1561, 1998.

[158] S. Vertuani, A. Sette, J. Sidney, S. Southwood, J. Fikes, E. Keogh, J. A. Lindencrona, G. Ishioka, J. Levitskaya, and R. Kiessling. Improved immunogenicity of an immunodominant epitope of the HER-2/neu protooncogene by alterations of MHC contact residues. *J Immunol*, 172(6):3501–3508, 2004.

[159] T. B. Stuge, S. P. Holmes, S. Saharan, A. Tuettenberg, M. Roederer, J. S. Weber, and P. P. Lee. Diversity and recognition efficiency of T cell responses to cancer. *PLoS Med*, 1(2):e28, 2004.

[160] S. Gnjatic, E. Jäger, W. Chen, N. K. Altorki, M. Matsuo, S.-Y. Lee, Q. Chen, Y. Nagata, D. Atanackovic, Y.-T. Chen, G. Ritter, J. Cebon, A. Knuth, and L. J. Old. CD8+ T cell responses against a dominant cryptic HLA-A2 epitope after NY-ESO-1 peptide immunization of cancer patients. *Proc Natl Acad Sci U S A*, 99(18):11813–11818, 2002.

[161] N. C. Toussaint, M. Feldhahn, M. Ziehm, S. Stevanović, and O. Kohlbacher. T-cell epitope prediction based on self-tolerance. In *Proceedings of the Second Immunoinformatics and Computational Immunology Workshop*, 2011.

[162] N. C. Toussaint, Y. Maman, O. Kohlbacher, and Y. Louzoun. Universal peptide vaccines – optimal peptide vaccine design based on viral sequence conservation. *Vaccine*, 2011 Aug 27 [Epub ahead of print].