# Discovering sequence motifs in quantitative and qualitative pepetide data

**Andreatta, Massimo; Nielsen, Morten; Lund, Ole**

*Publication date:*
2012

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

- PhD Thesis -

# Discovering sequence motifs in quantitative and qualitative peptide data

Massimo Andreatta

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

September 30, 2012

# Preface

This thesis was prepared at the Center for Biological Sequence Analysis, Department of Systems Biology, Technical University of Denmark as a requirement for obtaining the Ph.D. degree. The Ph.D. received funding from the Graduate School of Immunology, Faculty of Health Sciences, University of Copenhagen, and the European Union Seventh Framework Programme FP7/2007 2013 under grant agreement n$^o$ 222773. The work was mainly carried out at the Center for Biological Sequence Analysis, Denmark and at the Instituto Fundación Leloir, Argentina, under the supervision of Professor Ole Lund and Associate Professor Morten Nielsen.

Kongens Lyngby
September 2012

Massimo Andreatta

# Contents

## Abstract

Proteins are central to virtually all processes within the cell. The vast amount of functions performed by proteins in biological processes is conferred by their ability to bind in a selective and specific manner to other molecules. The nature of these interactions is, in general terms, three-dimensional, as binding sites normally consist of a pocket or a groove on the protein surface. However, in many cases such interactions contain a linear component and can be more conveniently represented, or approximated, by a protein-peptide interaction. Whereas time-consuming structural studies are necessary in systems where the three-dimensional aspect of the interaction is prevalent, protein-peptide interactions can normally be represented simply by a linear binding motif. Phage display and peptide microarray technologies allow generating large libraries of peptide sequences and the parallel detection of thousands of interactions in a single experiment, with virtually unlimited choice of potential targets and variants of these targets.

However, the amount and complexity of data produced by high-throughput techniques poses serious challenges to researchers of limited bioinformatics expertise who need to analyze and interpret such data. The first paper in this thesis presents a new, publicly available method based on artificial neural networks that allows custom analysis of quantitative peptide data. The online *NNAlign* web-server provides a simple yet powerful tool for the discovery of sequence motifs in large-scale peptide data sets. It was successfully applied to characterize the binding motifs of MHC class I and class II molecules, and for the prediction of protease cleavage on data generated by a large-scale peptide microarray technology.

In the second paper, *NNAlign* was applied to binding data for HLA-DP and DQ molecules, two classes of HLA molecules with recognized importance in immune response but poorly characterized sequence motifs. The sequence logos of 5 HLA-DP and 6 HLA-DQ molecules provide a characterization of their binding motifs at an unprecedented level of detail.

The third paper in this thesis deals with the presence of multiple motifs, due to the experimental setup or the actual poly-specificity of the receptor, in peptide data. A new algorithm, based on Gibbs sampling, identifies multiple specificities by performing two tasks simultaneously: alignment and clustering of peptide data. The method, available online as a web-server, was applied to various data sets including mixtures of MHC binding data and distinct classes of ligands to SH3 domains.

Next, we investigated how string kernels could be used to identify pattern in peptide data, with particular focus on the MHC class I system. We suggest a strategy that, unlike most available methods, allows to learn from peptides of multiple lengths to achieve improved predictive performance. This appeared particularly important in alleles and peptide lengths where experimental data was limited.

The last chapter presents a method to rationally guide the discovery of T-cell epitopes from ELISPOT and ICS assays based on peptide pool matrices. By prediction of binding affinity, analysis of peptide pools intersections, and combination of information from different donors, we show that the method can effectively rank potential epitope candidates and reduce the number of experimental tests needed to identify new epitopes.

Taken as a whole, this thesis provides a valuable series of algorithms and tools for the analysis of peptide data, both from the point of view of characterization of sequence motifs and the prediction of protein-peptide interactions.

## Dansk resumé

Proteiner spiller en central rolle i næsten alle processer i cellen. Proteiner udfører talrige funktioner i biologiske processer takket være deres evne til at binde selektivt og specifikt til andre molekyler. Disse interaktioner er generelt af tredimensionel karakter, fordi binding sites normalt består af en "lomme" eller en sprække på proteinoverfladen. I mange tilfælde indeholder sådanne interaktioner en lineær komponent og kan nemmere repræsenteres, eller tilnærmes, med en protein-peptid interaktion. Tidskrævende strukturelle undersøgelser er nødvendige når det tredimensionelle aspekt af interaktionen ikke kan ignoreres, hvorimod protein-peptid interaktioner kan repræsenteres som et lineært bindingsmotiv. Phage display og peptid-microarrays gør det mulig at generere store biblioteker af peptidsekvenser og lave parallelle målinger af tusinder af interaktioner i et enkelt eksperiment.

Mængden og kompleksiteten af de data, disse high-throughput teknikker producerer, skaber store udfordringer for forskere med begrænset bioinformatisk ekspertise, som har brug for at analysere og fortolke den slags data. Den første artikel i denne afhandling beskriver en ny, offentligt tilgængelig metode baseret på artificial neural networks, der tillader brugerdefinerede analyser af kvantitative peptid data. Web-serveren *NNAlign* er en simpel men effektiv metode til opdagelsen af sekvensmotiver i store peptid datasæt. Den blev brugt med succes til at karakterisere bindingsmotiver for MHC klasse I og klasse II molekyler, og til forudsigelse af proteaseskløvning udfra et stor peptid-microarray datasæt.

I den anden artikel, blev *NNAlign* brugt på bindingsdata for HLA-DP og DQ molekyler. Disse molekyler spiller en vigtig rolle i immunresponset, men har dårligt karakteriserede sekvensmotiver. Sekvenslogoer for 5 HLA-DP og 6 HLA-DQ molekyler giver et billede af deres bindingsmotiver med en hidtil uset detaljegrad.

Den tredje artikel i denne afhandling omhandler detektion af multiple motiver i peptid-data, som kan fremkomme enten på grund af den eksperimentelle opsætning eller fordi receptoren faktisk er polyspecifik. En ny algoritme baseret på Gibbs sampling, identificerer multiple specificiteter ved at udføre to opgaver samtidig: alignment og clustering af peptidsekvens data. Metoden, som er tilgængelig online som web-server, blev anvendt til forskellige datasæt blandt andet blandinger af MHC bindingsdata og forskellige klasser af ligander til SH3 domæner.

Derefter, undersøgte vi hvordan string kernels kan bruges til at identificere motiver i peptid data, med særlig fokus på MHC klasse I molekyler. Vi foreslår en strategi, der i modsætning til de fleste tilgængelige metoder, gør det muligt at lære fra peptider med forskellige længder og derved opnå en bedre prediktiv ydeevne. Dette så ud til at være særlig vigtigt for alleler og peptid længder, hvor mængden af eksperimentelle data var begrænset.

Det sidste kapitel fremlægger en metode til at hjælpe opdagelsen af T-celle epitoper fra ELISPOT og ICS assays, på baggrund af peptid-pool matricer. Metoden kombinerer eksperimentelle målinger fra forskellige donorer med forudsigelser af peptid-MHC bindingsaffinitet, til at effektivt rangordne potentielle epitopkandidater og reducere antallet af eksperimentelle tests der er nødvendige for opdagelse af nye epitoper.

Som en helhed giver denne afhandling en værdifuld samling algoritmer og metoder til analyse af peptid data, både med hensyn til karakterisering af sekvensmotiver og forudsigelse af protein-peptid interaktioner.

## Papers included in this thesis

- **Paper I: Andreatta M**, Schafer-Nielsen C, Lund O, Buus S, Nielsen M (2011) NNAlign: A web-based prediction method allowing non-expert end-user discovery of sequence motifs in quantitative peptide data. *PLoS ONE* 6(11): e26781. doi:10.1371/journal.pone.0026781

- **Paper II: Andreatta M**, Nielsen M (2012) Characterizing the binding motifs of 11 common human HLA-DP and HLA-DQ molecules using NNAlign.
*Immunology* 136(3): 306-311. doi: 10.1111/j.1365-2567.2012.03579.x

- **Paper III: Andreatta M**, Lund O, Nielsen M (2012) Simultaneous alignment and clustering of peptide data using a Gibbs sampling approach. Accepted with minor revision in *Bioinformatics*

## Papers not included in this thesis

- **Paper IV: Andreatta M**, Nielsen M, Møller Aarestrup F, Lund O (2010) In Silico Prediction of Human Pathogenicity in the $\gamma$-Proteobacteria *PLoS ONE* 5(10):e13680. doi:10.1371/journal.pone.0013680

- **Paper V:** Roque FS, Jensen PB, Schmock H, Dalgaard M, **Andreatta M**, Hansen T, Søeby K, Bredkjær S, Juul A, Werge T, Jensen L J, Brunak S (2011) Using Electronic Patient Records to Discover Disease Correlations and Stratify Patient Cohorts
*PLoS Comput Biol* 7(8): e1002141. doi:10.1371/journal.pcbi.1002141

## Abbreviations

| | |
|---|---|
| AA | Amino acid |
| ANN | Artificial neural network |
| ARI | Adjusted Rand index |
| AUC | Area under the ROC curve |
| CD | Cluster of differentiation |
| CTL | Cytotoxic T lymphocyte |
| CV | Cross-validation |
| HLA | Human leukocyte antigen |
| HMM | Hidden Markov model |
| ICS | Intracellular cytokine staining |
| KLD | Kullback-Leibler distance |
| LO | Log-odds |
| MC | Monte Carlo |
| MCC | Matthews correlation coefficient |
| MHC | Major histocompatibility complex |
| MSA | Multiple sequence alignment |
| PFR | Peptide-flanking region |
| PSSM | Position-specific scoring matrix |
| RLS | Regularized least squares |
| RMSE | Root-mean-square error |
| ROC | Receiver operating characteristic |
| SVM | Support vector machine |
| TCR | T-cell receptor |
| YF | Yellow fever |

# Chapter 1

# Introduction

Peptides are short amino acid sequences occurring ubiquitously in biological processes, such as metabolism, signal transduction and immune response. They are also extensively used in research to mimic functional or (linear) structural aspects of proteins and protein interactions. These include families of peptide-recognition domains (SH3, PDZ, WW, etc.), membrane receptors (e.g. the Major Histocompatibility Complex) and enzymes (e.g. kinases and phosphatases). Given the linear component of the interaction, the specificities of these systems can be effectively characterized by means of linear peptides [1].

If a protein-protein interaction can be represented, or approximated, by a protein-peptide interaction, the first evident advantage is a far lesser complexity of the system [2]. Whereas time-consuming structural studies are necessary in systems where the three-dimensional aspect of the interaction is prevalent, protein-peptide interactions can normally be represented simply by a linear binding motif. Another important advantage of using peptides lies in the relative ease in generating large libraries of sequences, such as in phage display or peptide microarrays [3, 4]. These technologies allow the parallel detection of thousands of interactions in a single experiment, with virtually unlimited choice of potential targets and variants of these targets.

In this thesis, we address the challenges of extracting information from very large peptide data sets. Firstly, the capability of high-throughput technologies presents the experimentalist with massive amounts of data, which can effectively only be interpreted with the aid of computational methods. This is addressed in chapter 2, where an interpretation service able to handle large-scale peptide data sets is presented.

Secondly, motifs contained in linear peptides are often weak and short. In these conditions, identifying the binding register within peptides is a challenging task. We will refer to this as the "alignment" problem. In chapter

3 we discuss how neural networks are capable of capturing such weak motifs and accurately describe the promiscuous protein-peptide interactions of HLA-DP and HLA-DQ molecules.

Thirdly, receptor-ligand data sets often contain multiple motifs. We will call this the "poly-specificity" problem. The presence of multiple patterns can either be due to the experimental setup or to the actual poly-specificity of the receptor. Chapter 4 tackles these issues, and presents a method that solves simultaneously the alignment problem and the poly-specificity problem.

Chapter 5 takes inspiration from a recently published method based on string kernels to suggest how to improve MHC class I binding predictions by training the method on peptides of multiple lengths. Finally, in chapter 6 we present a strategy to aid the discovery of T-cell epitopes from ELISPOT and ICS assays based on peptide pool matrices.

Before venturing into these aspects, we use this introductory chapter for a brief review the applications of peptides in protein science, how peptide data are harvested and what methods may be applied to extract information from these data. For the reader who is not completely familiar with the bioinformatics jargon and methods, we introduce some basic concepts of machine learning and data representation that are crucial to understand the rest of this thesis.

## 1.1 Peptide-based data

A first, essential distinction to be made is between qualitative and quantitative peptide data. Qualitative data are binary, whereas quantitative data span a spectrum of values between a minimum and a maximum. For instance, consider the interaction between peptides and a given molecule. A qualitative assessment of such interaction returns a binary answer: peptides either bind or do not bind to the molecule, without any intermediate levels. Conversely, quantitative measurements provide a scale of real values which, in this example, would represent the strength of the peptide-protein interaction. A quantitative scale therefore ranges from absolute non-binders to very strong binders with all the intermediate levels of binding strength in between.

It would appear that quantitative data is always more informative than its qualitative counterpart. However, due to the nature of the problem at hand or the characteristics of the assay used to study it, only qualitative observations may be available. For example, proteolytic cleavage of peptides by a protease may be regarded as a binary event. The amino acid chain is either cleaved or not. In other cases, where there actually is a measurable quantity, the experimental method may not be designed to detect it. The bioinformatics tools used to interpret the data are also subject to the qualitative/quantitative distinction. For instance, one of the strengths of the *NNAlign* method discussed in chapter 2 compared to other approaches, lies in its ability to exploit quantitative data and produce quantitative predictions.

Another major distinction into two groups can be made based on the biological properties of the receptor being investigated: the "aligned" case and the "unaligned" case. In the aligned case, the receptor binds peptides that are in the same register and have known length (the data is pre-aligned). The unaligned case is more complex, as the location of the binding core within the peptides is unknown, and bioinformatics methods aiming at analyzing such data must perform a sequence alignment to extract a sequence motif. Prominent examples of these two kinds of peptide data are the peptide interactions with respectively the Major Histocompatibility Complex (MHC) class I and MHC class II. We discuss these two systems next.

### 1.1.1 Aligned data - MHC class I

MHC class I molecules are transmembrane receptors that play an essential role in the immune system. Their function is to present peptide fragments derived from cytosolic proteins to cytotoxic T-cells (CTLs). If the peptide bound to the MHC class I molecule is recognized as foreign, CTLs will activate an immune response to destroy the infected cells.

Antigen presentation through the MHC class I pathway is a highly specific process, with only 1-5% of a set of random natural peptides binding to any given MHC molecule [5]. Moreover, in the vast majority of cases only peptides with length of 8-11 amino acids (and most commonly 9 AAs) can be recognized by MHC class I molecules. This is due to the particular conformation of the binding cleft of class I proteins, which is closed at both ends and can accommodate only peptides of limited length. In Figure 1.1a is shown the structure of a MHC class I molecule with a peptide in its binding cleft.

In humans, the MHC is also referred to as the Human Leukocyte Antigen (HLA) system. HLA class I molecules are extremely polymorphic, with up to 4,000 HLA-A, -B and -C alleles identified to date in the human population [6]. Polymorphism, which affects mostly the residues placed within or close to the binding cleft, implies that different allelic variants recognize peptides with different properties. However, despite such large polymorphism, class I molecules can be clustered into groups of molecules that bind largely overlapping peptide repertoires. Such groups of alleles, denominated as supertypes, have been quantified around 9-12 depending on the method employed [7, 8].

### 1.1.2 Unaligned data - MHC class II

As opposed to MHC class I, which samples antigens from cytosolic proteins, MHC class II molecules present peptides derived from extracellular proteins. Peptides bound to class II molecules are displayed to helper T-cells, which are capable of affecting and activating other cells of the immune system in response to a foreign fragment, for example derived from a bacterium infecting the blood.

From a structural point of view, MHC class II molecules differ functionally from class I proteins primarily in terms of the conformation of their

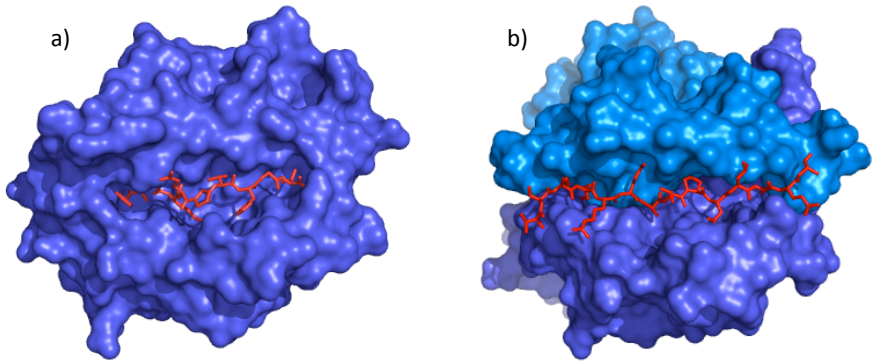



**Figure 1.1. The MHC class I and class II molecules with bound peptide ligand. a)** MHC class I molecule HLA-A2.1 (A2) with bound 9-mer peptide FLKEPVHGV in red sticks (PDB entry 1I1F [9]). Note that the binding groove is closed at both ends and can accommodate only peptides of limited length. **b)** MHC class II molecule HLA-DR1 with bound 14-mer peptide VSKMRMATPLLMQA (PDB entry 3QXA [10]). The $\alpha$ chain is in light blue, the $\beta$ chain in dark blue. The HLA binding groove is open at both ends and the ligand can extend outside the extremities of the pocket. Figures made using the PyMOL software [11].

binding cleft. As shown in Figure 1.1b, the binding groove of MHC class II molecules is open at both extremities. This does not pose constraints on the length of the peptide ligand, which can stick out freely at both ends of the pocket. Although normally only about 9 amino acids of the ligand are directly interacting with residues of the binding cleft (the binding core), peptides of up to 30 amino acids (even whole proteins) can be loaded onto MHC class II molecules [12]. This is why we refer to this system as "unaligned": the position of the binding core within the entire peptide is not known *a priori*, unlike in the class I system where the ends of the binding groove constrain peptides to a fixed length.

## 1.2 Peptide microarrays

Peptide microarrays consist of a library of peptides immobilized on a planar solid surface (typically a glass or coated slide). Such slides can be incubated with a biological sample, allowing the parallel detection of many peptide-target interactions. A wide range of biological systems can be investigated by means of peptide arrays, including antibody-antigen interactions, protease cleavage sites, kinase and phosphatase specificities, peptide-MHC binding, and in general all interactions where peptides can act as short linear analogues of proteins [4, 13, 14, 15, 16, 17].

In general terms, there are two kinds of peptide array technologies: parallel *in situ* synthesis arrays, and spotting arrays. With the first method, the peptide sequences are built directly on the slide starting from a matrix of linker groups coupled to the surface. Peptides are then elongated one amino

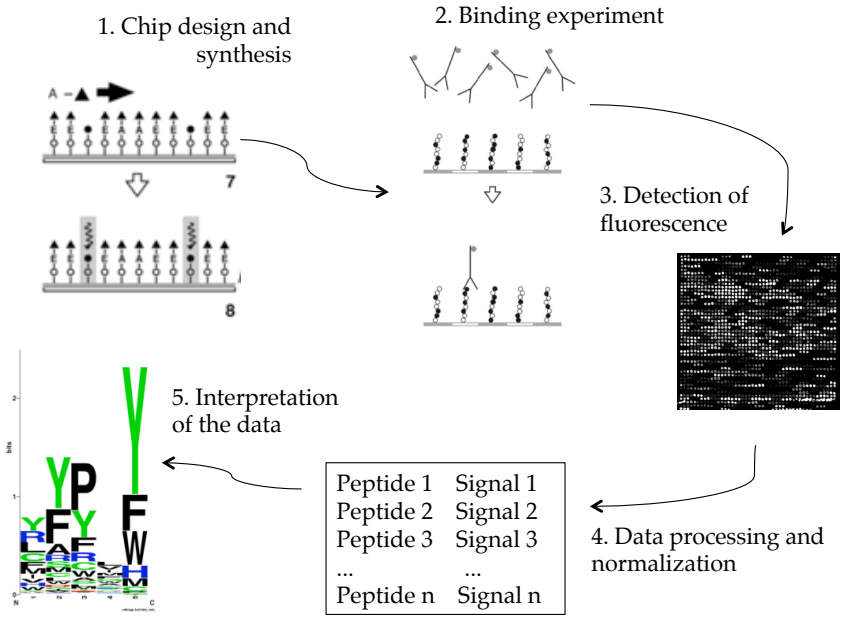



**Figure 1.2. Diagram of a peptide microarray experiment.** After synthesis of the peptide chains on the chip surface, the slide can be assayed with a target molecule (for example, antibodies). Interactions are detected by imaging devices and translated to an image of spots with varying intensity. As each spot can be mapped to a certain peptide, the matrix can be described as a list of peptides with a relative intensity value. Finally, bioinformatics analysis extrapolate from the quantitative measurements the pattern(s) governing the biological event under analysis.

acid at a time by means of photomasks that selectively de-protect, at each photolithographic iteration, the N-terminal of selected peptide chains. As a consequence, an exposed chain is coupled to a new amino acid, and the peptide grows by one unit. The process is repeated until all amino acid chains reach the desired length. In spotting arrays, on the other hand, peptide libraries are pre-synthesized and then applied to the slide in a determined order. Both approaches have advantages and disadvantages, reviewed for example in [4] and [18].

By means of fluorescent labels, interactions of the target molecules with peptides on a slide can be detected as luminous spots. The interaction may often also be regarded as quantitative, where different levels of signal intensity can be related to different strength or equilibrium of interaction. Such quantitative measurements allow representing the outcome of a peptide chip experiment as a list of peptides each with a paired value of intensity, a convenient format for pattern recognition by bioinformatics methods. The pipeline of a typical peptide chip experiment is depicted in Figure 1.2.

Over the past few years high-density peptide chips technologies have evolved rapidly, increasing by orders of magnitude the number of peptides

that can be screened in a single experiment. For example, in *Paper I* is shown the application of microarrays containing more than 100,000 peptides on a single slide for the analysis of protease cleavage specificities. Considering the growing capabilities of peptide chips to generate high-throughput data, including virtually all kinds of mutations and modifications of the original peptide sequence, they are an extremely powerful tool for studying various aspects of proteins and protein interactions. However, one must keep in mind that peptides can only explore the linear components of proteins. For instance, peptide arrays enable the detection of antibodies against linear epitopes, but not directly the identification of conformational epitopes formed by amino acids that are only in close proximity in the tertiary protein structure [14].

## 1.3 Sequence motifs and PSSMs

A sequence motif is an amino acid or nucleotide pattern shared by several sequences and presumably related to some biological function. Very simple sequence motifs can be represented by a regular expression listing the allowed and disallowed amino acids at each position in the motif. For instance, the general WW domain binding motif can be expressed as [AP]-P-P-[AP]-Y, where [AP] means either A or P. However there are exceptions, and they would have to be spelled out individually. But most importantly, the regular expression description does not say anything about the importance of different positions in the motif, and the relative importance of different amino acids at each position (in the WW domain example, are A and P equally probable? and are all prolines equally important?).

A quantitative representation of a binding motif may be derived from the data. Given a set of aligned sequences known to contain the motif, we can construct a matrix of probabilities $p_{i,a}$ expressing the frequency of all amino acids $a$ at each position $i$ of the alignment. These probabilities are commonly compared to the frequency $q_a$ of each residue $a$ in random proteins, resulting in a matrix of log-odds ratios:

$$LO_{i,a} = \log_2 \frac{p_{i,a}}{q_a} \tag{1.1}$$

The background amino acid frequencies $q_a$ can be regarded as uniform for all symbols ($q_a = 0.05$ for a 20 amino acids alphabet), but as some amino acid are more abundant than others in natural proteins, the background frequencies are commonly estimated from a large data set of proteins. In Table 1.1 are listed the $q_a$ values calculated from the UniProt database [19].

The log-odds matrix can be used as a Position-Specific Scoring Matrix (PSSM), allowing the detection of the motif in other sequences that were not used to construct the PSSM. By simply summing the (position-specific) scores of each amino acid in the query sequence, one obtains a quantitative measure of how closely the query matches the motif. However, there

**Table 1.1.** Amino acid background frequencies calculated from UniProt entries

| Amino acid | Abb | Frequency |
|---|---|---|
| Alanine | A | 0.074 |
| Arginine | R | 0.052 |
| Asparagine | N | 0.045 |
| Aspartic acid | D | 0.054 |
| Cysteine | C | 0.025 |
| Glutamine | Q | 0.034 |
| Glutamic acid | E | 0.054 |
| Glycine | G | 0.074 |
| Histidine | H | 0.026 |
| Isoleucine | I | 0.068 |
| Leucine | L | 0.099 |
| Lysine | K | 0.058 |
| Methionine | M | 0.025 |
| Phenylalanine | F | 0.047 |
| Proline | P | 0.039 |
| Serine | S | 0.057 |
| Threonine | T | 0.051 |
| Tryptophan | W | 0.013 |
| Tyrosine | Y | 0.032 |
| Valine | V | 0.073 |

are a few caveats about PSSM construction, and in particularly two deserve some discussion: data redundancy and small data sets. Data redundancy can be controlled by sequence weighting methods, and pseudocount correction deals with data sets composed of few points.

### 1.3.1 Sequence weighting

Data redundancy occurs when there is a sampling bias in the set of sequences containing the motif. The presence of several identical or nearly identical sequences may excessively skew the motif towards the presence of certain features that are only abundant because of over-sampling of a certain region of the sequence space. Ideally, one would want to downweigh such nearly identical sequences, and thereby enhance sequence diversity in the alignment. There are several strategies to implement sequence weighting, but we will mention two: heuristic weighting and sequence clustering.

Heuristic weighting, in its implementation by Henikoff and Henikoff [20], calculates a weight $w_x$ for each sequence $x$ according to the relationship:

$$w_x = \sum_{i=1}^{L} \frac{1}{r_i s_i} \tag{1.2}$$

where $r_i$ is the number of different amino acids at position $i$ in the alignment, $s_i$ is the number of occurrences in the alignment of the amino acid found at position $i$ in sequence $x$, and $L$ is the length of the alignment. Heuristic sequence weighting is very fast, and its computation time increases linearly with the number of sequences.

Sequence clustering is based on the Hobohm 1 algorithm [21]. In Hobohm 1, sequences are considered in order, one at a time. The first sequence will always form a cluster. The second sequence is compared to the first, and if they have % identity larger than a certain threshold (commonly 62%) they are placed together in the same cluster, otherwise the second sequence opens a new cluster. And so on, each sequence is compared to the previously accepted sequences and clustered accordingly. After clustering, a sequence $x$ in cluster $c$ receives a weight $w_x = 1/N_c$, where $N_c$ is the number of sequences in cluster $c$. Since each data point must be compared to any other data point in the set, the complexity of the algorithm is proportional to the square of the number of sequences.

### 1.3.2 Pseudocount correction

When a data set is small, the amino acid frequencies used to calculate the log-odds scores (equation 1.1) are only based on very few observations. While the data redundancy problem had to do with over-sampled regions of sequence space, here we are facing the opposite problem: the availability of only few data points leaves large uncovered holes in sequence space. Pseudocounts aim to partially cover these holes by exploiting similarities between different amino acids. For example, in many cases hydrophobic amino acids are to some extent interchangeable. If a position in the alignment appears to be preferentially composed by hydrophobic amino acids, it is likely that other hydrophobic amino acids are allowed at that position even though they were not directly observed due to lack of data. Similarities between amino acids are commonly measured in terms of BLOSUM scores [22]. Following the implementation by Altschul et. al [23], the pseudocount frequencies $g_a$ for a given column in the alignment are calculated using:

$$g_a = \sum_{b=1}^{20} f_b q(a|b) \tag{1.3}$$

where $f_b$ is the observed frequency for amino acid $b$ in the column, and $q(a|b)$ is the conditional probability of having amino acid $b$ substituted with amino acid $a$, as given by BLOSUM scores. It follows that, although a given amino acid $a$ may be rarely observed at a certain position (low $f_a$), if other amino acids $b$ similar to $a$ (high $q(a|b)$) are very frequent (high $f_b$) then the pseudocount frequency $g_a$ can be boosted to higher values.

The effective amino acid frequency $p'_a$ for amino acid $a$ can then be calculated with:

$$p'_a = \frac{\alpha f_a + \beta g_a}{\alpha + \beta} \tag{1.4}$$

where $\alpha$ and $\beta$ are the relative weights given to the observed ($f_a$) and pseudocount ($g_a$) frequencies. $\alpha$ is the number of sequences in the alignment minus 1. However, if sequence weighting is employed to reduce redundancy as described in the previous section, $\alpha$ may be more properly considered as the *effective* number of sequences. In the case of sequence weighting based on clustering, $\alpha$ would be the number of clusters estimated by the Hobohm 1 algorithm, which reflects better the number of actual data points in the alignment. With heuristic sequence weighting, $\alpha$ can be estimated as the average number of different amino acids (average $r_i$ from equation 1.2) across the alignment columns.

## 1.4 Artificial neural networks

Scoring matrices are based on the assumption that positions in the motif are independent from each other. That is, presence of a given residue at some position does not influence the amino acid preference at another position in the motif. However, such an assumption is often not true. For example, significant correlations between different positions have been found in peptide recognition domains [24]. In MHC-binding, it has been shown that there are signals of higher order correlation in amino acids located between the anchor positions [25].

Such correlations can be captured by artificial neural networks (ANN) with hidden layers. As in biological networks of neurons in the brain, the basic units of the system (the neurons) do not have predefined tasks but rather act collectively by influencing each other's behavior. Such interconnected structure allows simple units of basic function to exhibit complex (or "intelligent") global behavior. The simplest kind of ANN, widely used in bioinformatics, is the feedforward neural network, where signal travels from input to output without loops or cycles. A schematic depiction of a feedforward network is shown in Figure 1.3.

A given input example $x$ is presented, through some kind of encoding, to the input layer of the network. Then, the values stored in the input neurons are passed to the hidden layer. Each hidden neuron receives a number of signals from the input neurons each with an associated weight. These weights, which are the parameters of the network, quantify the influence of a particular input on stimulating the neuron, and can also have negative values if that particular input tends to deactivate the neuron. The weighted inputs for a given neuron are commonly summed, and if the combined inputs are above a certain threshold the neuron is "active" and will transmit a positive signal to the next layer, otherwise the neuron is off. Rather than a sharp step function, a sigmoid function is commonly used for its mathematical properties in differentiation, taking the form $f(x) = \frac{1}{1+e^{-x}}$ (see inset box in Figure 1.3). The same applies for any subsequent hidden layers and for the output layer: the weighted inputs for each neuron are summed and filtered through a sigmoid function, to generate output for the next layer (or global output, in the case of the final layer).

The above procedure describes how a given input is streamed through the network to produce output. In other words, how a prediction can be obtained from an already trained ANN. But first, a network must be trained for the problem at hand, i.e. it must identify the optimal set of parameters (the weights connecting neurons) that produce output values as close as possible to some desired target values. This can be done using a back-propagation algorithm, which iteratively updates the weights to minimize the error between predicted and target outputs. In online learning, the error is back-propagated from output to input layer every time a training example is presented to the network. Each training point may have to be presented up to hundreds of times to the network to ensure that the error converged to a minimum.

In order to train ANNs on protein/peptide data, amino acids must be translated to numbers using some kind of encoding scheme. Two common encoding schemes, both used in the *NNAlign* method described in chapter 2, are sparse encoding and BLOSUM encoding. In Sparse encoding, a vector of length 20 represents each amino acid, where all values are set to zero apart from the one indexing the observed amino acid. For example, the first amino acid in the alphabet Alanine would be represented as 10000000000000000000, the second amino acid Arginine as 01000000000000000000 and so on. An additional digit may be added to identify the unknown amino acid X. BLOSUM encoding is different as it takes into account amino acid similarity, and produces a vector with the 20 BLOSUM matrix values for the observed amino acid. Such vectors are presented to the input layer of the neural networks to represent each position of a sequence alignment. More details about implementation and ensembling of ANNs on peptide data are given in chapter 2.

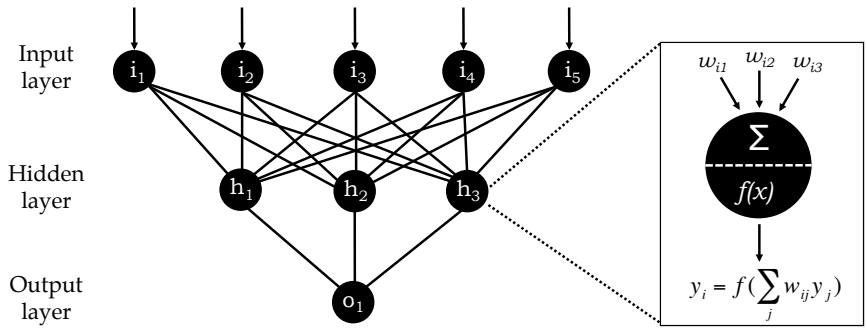



**Figure 1.3. Schematic representation of a typical feedforward neural network.** In this example the ANN consists of an input layer composed of 5 neurons ($i_1$-$i_5$), one hidden layer with 3 neurons ($h_1$-$h_3$) and a single output neuron ($o_1$). Note that in the general case there may be multiple hidden layers and/or multiple output neurons. In the inset box is shown the behavior of a single neuron $i$: the inputs $y_j$ are multiplied by their weights $w_{ij}$ and summed, and the output $y_i$ of the neuron depends on the activation function $f(x)$ where $x$ is the weighted sum of the inputs.

## 1.5 Sequence logos

Sequence logos were introduced by Schneider et al. [26] as a visualization tool for DNA and amino acid patterns. Given a multiple sequence alignment (MSA), the pattern is represented as a stack of letters for each position in the alignment, where the height of each letter represents the relative frequency of the element at that position. Sequence logos are very popular because they condense several pieces of information in a single plot: *i)* the general consensus of the alignment; *ii)* the favored residues at each position; *iii)* the relative frequencies of every residue at every position; *iv)* the information content at each position in the MSA; *v)* significant locations in the alignment, such as anchor positions for binding.

In a Shannon sequence logo, the height $R_i$ of a column $i$ in the logo is given as the information content in bits at that particular position, calculated using

$$R_i = \log_2 A + \sum_{a=1}^{A} f_{a,i} \log_2 f_{a,i} \tag{1.5}$$

where $f_{a,i}$ is the frequency of amino acid $a$ at position $i$, and $A$ is the size of the alphabet (normally $A = 20$ for proteins and $A = 4$ for nucleic acid sequences).

The height $h_{a,i}$ of symbol $a$ at position $i$ is given by

$$h_{a,i} = R_i f_{a,i} \tag{1.6}$$

The value of $R_i$ ranges from 0, for a position where all symbols have the same frequency (maximum entropy), to $\log_2 A$ for a completely conserved position in the alignment.

If the number of sequences in the MSA is limited, equation 1.5 tends to underestimate entropy [27]. The small sample bias is normally accounted for using a correction factor [27, 28] or by means of pseudocounts [23]. Another aspect to consider when making a sequence logo is data redundancy: when closely related sequences are over-represented in a MSA, this bias would reflect in the motif visualized by the logo, emphasizing excessively the contribution of the redundant sequences. Data redundancy in MSAs can be controlled by means of position-based sequence weights [20].

The most popular webserver for the generation of sequence logos from a sequence alignment is WebLogo [28] but several others exist [29, 30, 31, 32]. Among these, Seq2Logo [32] is particularly attractive as it allows for sequence redundancy reduction, pseudocount correction, and provides different types of logo besides the Shannon sequence logo. Starting from the consideration that different amino acids occur in nature with different background frequencies $q_a$, equation 1.5 can be re-written as

$$R_i = \sum_{a=1}^{A} f_{a,i} \log_2 \frac{f_{a,i}}{q_a} \tag{1.7}$$

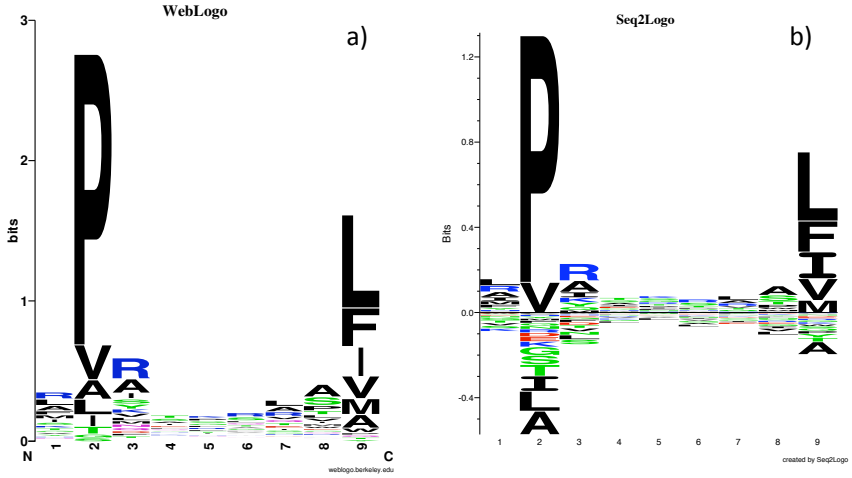



**Figure 1.4. Sequence logos made using WebLogo [28] and Seq2Logo [32]**. Both sequence logos were derived from the same data set of 100 binders to the MHC molecule HLA-B*07:02.

The logo representation derived from equation 1.7 is termed Kullback-Leibler (KL) logotype. In this context, a residue observed with higher frequency than expected by chance ($f_{a,i} > q_a$) is called an 'enriched' amino acid, whereas a residue observed more rarely than the background frequency ($f_{a,i} < q_a$) is called a 'depleted' amino acid. In Seq2Logo enriched amino acids are shown on the positive side of the $y$-axis and depleted amino acids as upside-down letters on the negative side of the $y$-axis.

Throughout this thesis we use both WebLogo and Seq2Logo to display motifs in MSAs. As a reference for comparison, Figure 1.4 shows the sequence logos produced by the two methods on the same data set of 100 binders to the HLA-B*07:02 molecule. The WebLogo representation is a Shannon logo, plotting the information content at each position and the relative frequency of each amino acid in the alignment. The Kullback-Leibler representation made with Seq2Logo displays also under-represented amino acids on the negative $y$-axis. Notice how certain amino acids such as Alanine (A) and Leucine (L) appear favored in the Shannon logo but depleted in the KL logo. This is mainly because A and L are rather frequent in naturally occurring sequences (respectively 7.4% and 9.9% as opposed to the 5% expected on average for a 20 letter alphabet), and this bias is only taken into account by the KL logotype. The latter should arguably be the favored logo representation on non-equiprobable alphabets.

# Chapter 2

# Peptide sequence alignment using ANNs

An essential step in extracting information from peptide data is sequence alignment. Multiple local sequence alignment aims at identifying recurrent sub-sequences in a set of peptides, with the assumption that sequence similarity is related to some biological property shared by the peptides. However, when the shared sequence motif is weak and short, as is often the case with peptide data, conventional sequence alignment approaches tend to fail at this task [33]. Several bioinformatics methods have been developed to identify such subtle sequence signals including Gibbs sampling [34], Hidden Markov Models [35] and artificial neural networks [36]. In particular, artificial neural networks (ANNs) have shown high performance on this kind of data [37]. Part of their success resides in the ability of capturing higher order correlations, that is, the presence of a given residue at some position may affect the amino acid preference at another position in the alignment core.

Recently our group described a neural network-based method, *NN-align*, which has been shown to perform significantly better than any other published method for MHC class II binding prediction [36]. In the following section (2.1), we describe how the method was extended and adapted to handle quantitative peptide data in general, including applications to high-throughput peptide array data. The method was also implemented as a user-friendly web server allowing the non-expert end user to perform advanced bioinformatics analysis of large peptide data sets.

One important feature introduced in *NNAlign* is a new offset correction algorithm to improve the motif visualization of neural network ensembles. In section 2.2 we discuss in more detail the algorithm and its implementation.

## 2.1 Paper I

# NNAlign: a web-based prediction method allowing non-expert end-user discovery of sequence motifs in quantitative peptide data

*PLoS ONE*, November 2011 vol. 6(11): e26781. doi:10.1371/journal.pone.002678

Massimo Andreatta*[1], Claus Schafer-Nielsen[2], Ole Lund[1], Søren Buus[3] and Morten Nielsen[1]

[1]Center for Biological Sequence Analysis, Department of Systems Biology, Technical University of Denmark, DK-2800 Lyngby, Denmark [2]Schafer-N, DK-2100 Copenhagen, Denmark [3]Laboratory of Experimental Immunology, Faculty of Health Sciences, University of Copenhagen, DK-2200 Copenhagen, Denmark
* Corresponding author - massimo@cbs.dtu.dk

**Abstract**

Recent advances in high-throughput technologies have made it possible to generate both gene and protein sequence data at an unprecedented rate and scale thereby enabling entirely new "omics"-based approaches towards the analysis of complex biological processes. However, the amount and complexity of data that even a single experiment can produce seriously challenges researchers with limited bioinformatics expertise, who need to handle, analyze and interpret the data before it can be understood in a biological context. Thus, there is an unmet need for tools allowing non-bioinformatics users to interpret large data sets. We have recently developed a method, *NNAlign*, which is generally applicable to any biological problem where quantitative peptide data is available. This method efficiently identifies underlying sequence patterns by simultaneously aligning peptide sequences and identifying motifs associated with quantitative readouts. Here, we provide a web-based implementation of *NNAlign* allowing non-expert end-users to submit their data (optionally adjusting method parameters), and in return receive a trained method (including a visual representation of the identified motif) that subsequently can be used as prediction method and applied to unknown proteins/peptides. We have successfully applied this method to several different data sets including peptide microarray-derived sets containing more than 100,000 data points. *NNAlign* is available online at `http://www.cbs.dtu.dk/services/NNAlign`.

### 2.1.1 Introduction

Proteins are extremely variable, flexible and pliable building blocks of life that are crucially involved in almost all biological processes. Many diseases are caused by protein aberrations, and proteins are frequent targets of intervention. A plethora of high-throughput methods are currently being used to study genetic associations and protein interactions, and intense ongoing international efforts aim at understanding the structures, functions and molecular interactions of all proteins of organisms of interest (e.g. the Human Proteome Project, HPP). In some cases, linear peptides can emulate functional and/or structural aspects of a target structure. Such peptides are currently identified using simple peptide libraries of a few hundreds to thousands peptides whose sequences have been systematically derived from the target structure at hand − that is, if this is known. Even when the native target structure is unknown, or too complex (e.g. discontinuous) to be represented by homologous peptides, the enormous diversity and plasticity of peptides may allow one or more peptides to mimic relevant aspects of a given target structure [38, 39].

Peptides are therefore of considerable biological interest and so are methods aimed at identifying and understanding peptide sequence motifs associated with biological processes in health and disease. Indeed, recent developments in large-scale, high-density peptide microarray technologies allow

the parallel detection of thousands of sequences in a single experiment, and have been used in a wide range of applications, including antibody-antigen interactions, peptide-MHC interactions, substrate profiling, identification of modification sites (e.g. phosphorylation sites), and other peptide-ligand interactions [40, 13, 41, 42, 43]. One of the major advances of peptide microarrays is the ease of generating large numbers of potential target structures and systematic variants hereof [4].

Given the capability for large-scale data generation already realized in current "omics" and peptide microarray-based approaches, experimentalists will increasingly be confronted with extraordinary large data sets and the consequent problem of identifying and characterizing features common to subsets of the data. These are by no means trivial problems. Up to a certain level of size and complexity, data can be presented in simple tabular forms or in charts, however, larger and/or more complex bodies of data (e.g. in proteome databases) will need to be fed into bioinformatics data mining systems that can be used for automated interpretation and validation of the results, and eventually for *in silico* mapping of peptide targets. Moreover, such systems can conveniently be used to design next-generation experiments aimed at extending the description of target structures identified in previous analyses [44].

A wealth of methods has been developed to interpret quantitative peptide sequence data representing specific biological problems. By way of examples, SignalP, which identifies the presence of signal peptidase I cleavage sites, is a popular method for the prediction of signal peptides [45]; LipoP, which identifies peptidase II cleavage sites, predicts lipoprotein signal peptides in Gram-negative bacteria [46]; various prediction methods predict phosphorylation sites by identifying short amino acid sequence motifs surrounding a suitable acceptor residue [47, 48, 49, 50] etc. In general terms, these methods can be divided in two major groups depending on the structural properties of the biological receptor investigated, and of the nature of the peptides recognized. The simplest situation deals with interactions where a receptor binds peptides that are in register and of a known length. In this case, the peptide data is pre-aligned, and conventional fixed length, alignment-free pattern recognition methods like position specific weight matrices (PSSM), artificial neural networks (ANN), and support vector machines (SVM) can be used. Peptide-MHC class I binding is a prominent example of the successful use of such methods to characterize receptor-ligand interaction represented by pre-aligned data (reviewed in [51]). Another more complex type of problems deals with interactions where either the motif lengths, and/or the binding registers, are unknown. In these cases, the peptide data must a priori be assumed to be unaligned and any bioinformatics method dealing with such data is faced with the challenge of simultaneously recognizing the binding register (i.e. performing an alignment) and identifying the binding motif (i.e. performing a specificity analysis). Peptide-MHC class II binding is a preeminent example of a receptor-ligand interaction represented by unaligned data. Several bioinformatics methods have

been developed to identify binding motifs in such peptide data including Gibbs sampling [33], hidden Markov models (HMM) [52], stabilization matrix method (SMM) alignment [53], and alignment using artificial neural networks [36] (for more references see [6]). Another example of unaligned peptide data is that of antibodies interacting with linear peptide epitopes. Although B-cell epitopes frequently are conformational and three-dimensional in structure, some do contain linear components that can be represented by peptide interaction with the corresponding antibodies [54, 55, 56].

Even though most of the methods described above are standard methods for data-driven pattern recognition, the development of a prediction method for any given biological problem is far from straightforward, and the non-expert user will rarely be able to develop their own state-of-the-art prediction methods. We have recently described a neural network-based data driven method, *NN-align*, which has been specifically designed to automatically capture motifs hidden in unaligned peptide data [36]. *NN-align* is implemented as a conventional feed-forward neural network and consists of a two-step procedure that simultaneously identifies the optimal peptide-binding core, and the optimal configuration of the network weights (i.e. the motif). This method is therefore inherently designed to deal with unaligned peptide data, and it identifies a core of consecutive amino acids within the peptide sequences that constitute an informative motif. Note that the method does not allow for gaps in the alignment. Although *NN-align* was originally developed with the unaligned nature of peptide-MHC class II interaction in mind − and independent validations have shown that *NN-align* indeed performs significantly better than any previously published methods for MHC class II motif recognition [37] − the unique ability of this method to capture subtle linear sequence motifs in quantitative peptide-based data and its adaptability makes it extremely attractive for other applications as well. Here, we have adapted and extended the *NN-align* method so that it can handle quantitative peptide-based data in general. Making this method generally available for the scientific community, we have embedded it into a public online web-interface that facilitates both handling of input data, optimization of essential training parameters, visual interpretation of the results, and the option of using the resulting method to predict on user-specified proteins/peptides. Through the server the user can easily set up a cross-validation experiment to estimate the predictive performance of the trained method, and automatically reduce redundancy in the data. The logo visualization is also improved with an algorithm that aligns individual neural networks to maximize the information content of the combined alignment. This web-based extension of the *NN-align* method empowers experimentalists of limited bioinformatics background with the ability to perform advanced bioinformatics-driven analysis of his/her own sets of large-scale data.

### 2.1.2 Results

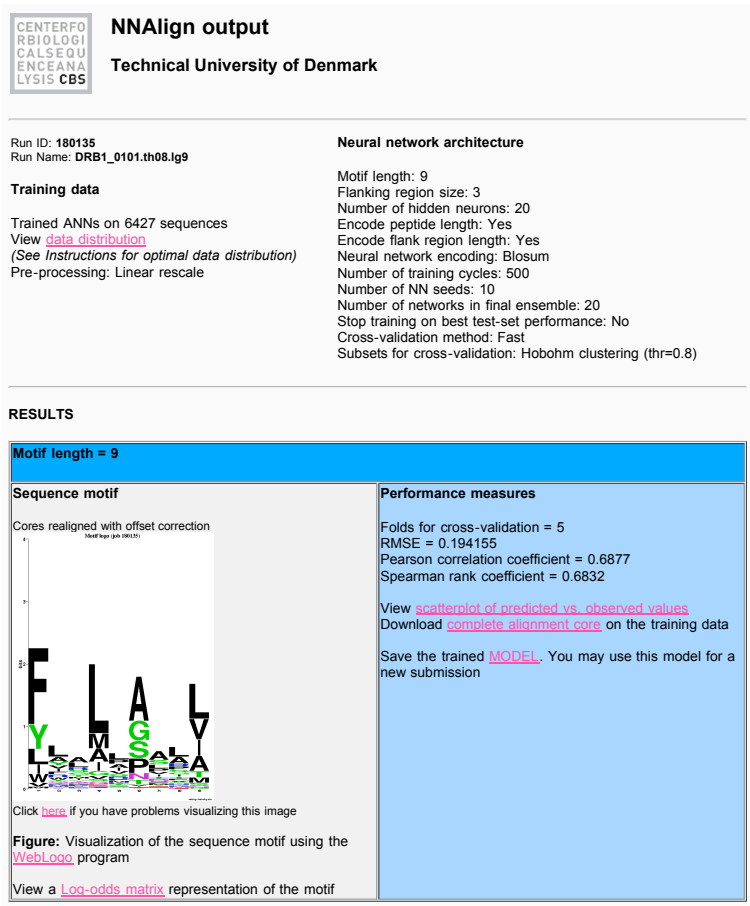
Enabling any non-expert end-user to extract specific information from quantitative peptide data using an advanced bioinformatics approach, we have

used our recently published *NN-align* method to generate a web-based extension with a reasonably simple, yet adaptable, web-interface and made this server publicly available at `http://www.cbs.dtu.dk/services/NNAlign`. Using this web server any user can submit quantitative peptide data (optimally based on actual discrete measurements, but even assigned classification, e.g. 0 and 1, can be used) and in return receive a trained method including training details and estimated predictive performance, a visual interpretation of the identified peptide pattern, and the trained model itself. The latter can be re-submitted to the web server at any later time and used to predict the occurrence of the learned motif in one or more concurrently submitted peptide sequences or FASTA format sequences.

The truly non-expert user has the option of using a set of default settings. Using these settings, the data is preprocessed using a linear transformation to make the data fall in the range from 0 to 1, and the *NN-align* method is trained using five-fold cross-validation. For each cross validation partition five networks, each initiated from different initial configurations, are trained with 3 hidden neurons. The only critical parameter that the user is required to specify is the motif length. The value used for this parameter is specific to each problem and the user is recommended to define a motif length (or an interval of motif length) that is relevant to the biological problem investigated by the peptide data. The default settings will in most cases allow the user to obtain a first impression of the motif contained in the data, and achieve a prediction method that allows the user to make prospective studies on uncharacterized proteins/peptides. The more experienced user has several advanced options to customize the training. For details on these options refer to Materials and Methods section, or the help section of the web-server.

An example output from the *NNAlign* Server is shown in Figure 2.1. Information about the training data is accompanied by a plot of the data distribution before and after the data processing needed to train the neural networks. An important feature is the possibility to download and save the trained model, and use it subsequently for predictions on new data. The results page also returns the performance of the method as estimated by cross-validation, and provides links to a scatter-plot showing the correlation between measured and predicted values, as well as the complete alignment core on the training data. A sequence logo gives a visual representation of the identified sequence motif, which can also be viewed in a log-odds position-specific scoring matrix format. If any evaluation data has been provided at the time of method training, a section of the results will report the predictions of this evaluation set.

A few example applications illustrating the power of the *NNAlign* method are presented in the following sections. First, the method is applied to examples of pre-aligned peptide data using examples of MHC class I binding. Next, the alignment problem is included using MHC class II binding data, showing the ability of the method to identify at the same time the correct length of the motif, the binding register, and the sequence motif itself.



CENTERFO RBIOLOGI CALSEQU ENCEANA LYSIS CBS

**NNAlign output**

**Technical University of Denmark**

Run ID: **180135**
Run Name: **DRB1_0101.th08.lg9**

**Training data**

Trained ANNs on 6427 sequences
View data distribution
*(See Instructions for optimal data distribution)*
Pre-processing: Linear rescale

**Neural network architecture**

Motif length: 9
Flanking region size: 3
Number of hidden neurons: 20
Encode peptide length: Yes
Encode flank region length: Yes
Neural network encoding: Blosum
Number of training cycles: 500
Number of NN seeds: 10
Number of networks in final ensemble: 20
Stop training on best test-set performance: No
Cross-validation method: Fast
Subsets for cross-validation: Hobohm clustering (thr=0.8)

**RESULTS**

**Motif length = 9**

**Sequence motif**

Cores realigned with offset correction

Click here if you have problems visualizing this image

**Figure:** Visualization of the sequence motif using the WebLogo program

View a Log-odds matrix representation of the motif

**Performance measures**

Folds for cross-validation = 5
RMSE = 0.194155
Pearson correlation coefficient = 0.6877
Spearman rank coefficient = 0.6832

View scatterplot of predicted vs. observed values
Download complete alignment core on the training data

Save the trained MODEL. You may use this model for a new submission

**Figure 2.1. Example of output from the *NNAlign* server trained on MHC class II binding data for allele HLA-DRB1*01:01.** Links on the results page (in pink) redirect to additional files and figures relevant for the analysis. Run ID is a sequential identifier for the current job, and Run Name a user-defined prefix that is added to all files of the run. The view data distribution link shows the transformation applied to the data in preprocessing, which can be either a linear or logarithmic transformation. In this case the method was trained with a motif length of 9, including a PFR of size 3 to both ends of the peptide, and encoding in the network input layer peptide length and PFR length. The hidden layer was made of a fixed number of 20 neurons. Peptides were presented to the networks using a Blosum encoding to account for amino acid similarity, for 500 hundred iterations per peptide without stopping on the best test set performance. At each cross-validation step, 10 networks were trained starting from 10 different initial configurations. The subsets for cross-validation were constructed using a Hobohm1 method that groups in the same subset sequences that align with more than 80% identity (thr = 0.8). The model can be downloaded to disk using the dedicated link, and can be resubmitted to *NNAlign* to find occurrences of the learned pattern in new data. The estimated performance of the trained method is expressed in terms of Root Mean Square Error, Pearson and Spearman correlation. A visual representation of the correlation can be obtained from the scatterplot of predicted versus observed values. The complete alignment core link allows downloading the prediction values in cross-validation for each peptide, and where the core was placed within the peptides. Next follows a section on the sequence logo, showing a logo representation of the binding motif learned by the network ensemble. If the relative option is selected, links to logos for the individual networks in the final ensemble are also listed here. Finally, if an evaluation set is uploaded, an additional section shows performance measures and core alignment for these data.

An important output from the *NNAlign* method is a sequence logo representing the identified binding motif. Such sequence logos provide a highly intuitive representation of single-receptor specificities (as is the case for MHC class I and II binding data). Finally, to illustrate how the method is capable of handling and guide the semi-expert user in interpreting large-scale data sets, *NNAlign* is applied to data generated by a large-scale peptide microarray technology.

**MHC class I**

Binding of peptides to MHC class I molecules is highly specific, with only 1-5% of a set of random natural peptides binding to any given MHC molecule [5]. Moreover, in the vast majority of cases only peptides with length 8-10 amino acids can fit in the binding pocket of MHC class I molecules. The predictive performance of *NNAlign* on 12 human MHC class I alleles from data by Peters et al. [57] is shown in Table 2.1 (see the table footnote for the parameters used). The benchmark data sets contain quantitative binding data of a given length (9 amino acids) covering the whole spectrum from non-binding to strong-binding peptides, hence serving as a perfect illustration of the strength of the *NNAlign* method to handle pre-aligned peptide data. The overall performances of the three methods are comparable demonstrating that *NNAlign* competes with state-of-the-art methods designed specifically for MHC class I prediction.

**MHC class II**

As opposed to MHC class I binding, which is mostly limited to peptides of similar length, the MHC class II molecule interacts with peptides of a wide length distribution and high compositional diversity [60]. Binding of a peptide to an MHC class II molecule is primarily determined by a core of normally 9 amino acids, but the composition of the regions flanking the binding core (peptide flanking region, PFR) has been shown to also affect the binding strength of a peptide [61, 62]. Identifying the binding motif and binding register for MHC class II binding peptides is thus a problem that inherently requires simultaneous alignment and binding affinity identification. Here, an MHC class II benchmarking was obtained from the recent publication by Wang et al. [37]. The performance was estimated for each allele using a 5 fold cross validation, where at each step 4/5 of the data were used to train the neural networks, and 1/5 were left out for evaluation. For cross-validation, we preserved the same data partitioning as used in the original publication. In Table 2.1, the performance of *NNAlign* on the Wang set is compared to other publicly available methods for MHC class II prediction. These include *SMM-align* [53], *ProPred/Tepitope* [58, 59], as well as the original version of the *NN-align* algorithm [36]. The *NN-align*-based methods outperform their competitors on all alleles, confirming the ability of the neural networks in dealing with alignment problems. The difference with the original *NN-align* method, which is due to differences in network architecture, is small and not

**Table 2.1.** Predictive performance in AUC on 12 human HLA MHC class I alleles (Peters data set) and on 14 HLA-DR MHC class II alleles (Wang similarity reduced SR dataset).

| **MHC class I** | | | | | **MHC class II** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **ALLELE** | **#** | **SMM** | **ANN** | **NNAlign** | **ALLELE** | **#** | **NN-align** | **SMM-align** | **Propred** | **NNAlign server** |
| A*0101 | 1157 | 0.980 | **0.982** | 0.980 | DRB1*0101 | 3504 | 0.763 | 0.756 | 0.692 | **0.794** |
| A*0201 | 3089 | 0.952 | 0.957 | **0.959** | DRB1*0301 | 1136 | **0.829** | 0.808 | 0.669 | 0.816 |
| A*0203 | 1443 | 0.916 | 0.921 | **0.922** | DRB1*0401 | 1221 | 0.734 | 0.721 | 0.711 | **0.736** |
| A*2402 | 197 | 0.780 | **0.825** | 0.772 | DRB1*0404 | 474 | **0.803** | 0.789 | 0.753 | 0.782 |
| A*0301 | 2094 | 0.940 | 0.937 | **0.941** | DRB1*0405 | 1049 | 0.794 | 0.767 | 0.742 | **0.808** |
| A*1101 | 1985 | 0.948 | 0.951 | **0.952** | DRB1*0701 | 1175 | 0.811 | 0.796 | 0.75 | **0.845** |
| A*2902 | 160 | 0.911 | **0.935** | 0.920 | DRB1*0802 | 1017 | 0.698 | 0.689 | 0.641 | **0.714** |
| A*3101 | 1869 | 0.930 | 0.928 | **0.931** | DRB1*0901 | 1042 | 0.713 | 0.696 | | **0.745** |
| A*6801 | 1141 | **0.885** | 0.883 | 0.881 | DRB1*1101 | 1204 | 0.847 | 0.829 | 0.779 | **0.853** |
| B*0702 | 1262 | 0.964 | **0.965** | 0.961 | DRB1*1302 | 1070 | 0.732 | 0.754 | 0.577 | **0.775** |
| B*3501 | 736 | **0.889** | 0.875 | 0.876 | DRB1*1501 | 1171 | 0.756 | 0.741 | 0.703 | **0.765** |
| B*5301 | 254 | 0.882 | **0.899** | 0.875 | DRB3*0101 | 987 | **0.798** | 0.78 | | 0.784 |
| **Ave** | | 0.914 | 0.922 | 0.914 | DRB4*0101 | 1011 | 0.789 | 0.762 | | **0.808** |
| | | | | | DRB5*0101 | 1198 | 0.795 | 0.776 | 0.711 | **0.798** |
| | | | | | **Ave** | | 0.776 | 0.762 | 0.703 | 0.787 |

For MHC class I no significant difference is found in predicted performance between the *NNAlign*, *SMM* and *ANN* method (p>0.5, binomial test). The values for the *SMM* and *ANN* methods were taken from Peters et al. [57]. The method was trained using a fixed motif length of 9 corresponding to the peptide length, and constructing a network ensemble with multiple architectures using respectively 2, 3, 4, 5 and 7 hidden neurons. Performance was measured in cross-validation, training each network for a fixed number of 500 iterations per sequence.
The different MHC class II prediction methods are *NN-align* [36], *SMM-align* [53], and *ProPred* [58, 59]. *NNAlign* server is the method described here. Performance values for first 4 methods are taken from [37]. *NNAlign* was trained with a motif length of 9, flanking regions of 3 amino acids, Blosum encoding including peptide length and flanking region length, and an ensemble of 2, 3, 5, 9 and 12 hidden neurons for each of 10 initial random configurations.
In bold is highlighted the best performing method for each MHC allele. The column # gives the number of the peptides in the data set for the given allele.

significant (p>0.2, binomial test). For this example involving unaligned data, the *NNAlign* server competes with comparable state-of-the-art methods.

### Choosing the optimal motif length

Different positions in a binding motif can be more or less informative, and the ends of a motif can often not be clearly delineated. This prompts the question of how many positions are necessary and sufficient to represent a given motif and how the length of a motif is defined. *NNAlign* allows searching for the optimal motif length in a quantitative peptide data set. Here, the best motif length is the one that yields, in a cross-validation experiment, the lowest root mean square error (RMSE) between observed and predicted values. By this token, a terminal position is included in the motif if it contributes with information at a level above what could be considered to be noise. In contrast, if the inclusion of a putative terminal position does not lead to a reduction in the RSME then it can be concluded that it does not add useful parameters to the model; rather, it lowers the predictive performance and should be omitted. This approach was used to suggest the motif length of the 14 MHC class II HLA-DR alleles, which were searched for optimal predictive performance by scanning through possible lengths from 6 to 11 amino acids. *NNAlign* will report the length associated with the lowest RMSE value as the optimal motif length (see Figure 2.2, left hand panel). Nonetheless, the user is advised to inspect the sequence logo as well as the performance plot of the RMSE as a function of the motif length to evaluate whether the dependence upon length appears significant. As defined here and illustrated in Figure 2.2 right

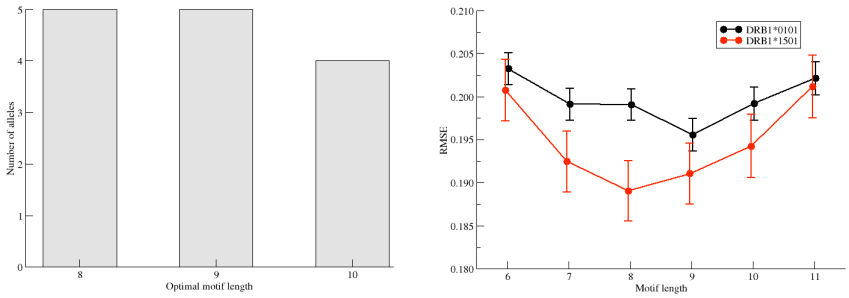



**Figure 2.2. Identification of optimal motif length using the *NNAlign* method. Left panel:** Histogram of the optimal motifs lengths for the 14 HLA-DR molecules in the Wang dataset as identified by the *NNAlign* method. **Right panel:** Predictive performance measured in terms of the root mean square error (RMSE) between observed and predicted values as a function of the motif length for the two molecules DRB1*01:01 and DRB1*15:01. *NNAlign* was trained using the same parameters settings described in Figure 2.4. At each motif length are shown the mean and standard error of the mean RMSE as estimated by bootstrap sampling. For DRB1*01:01 a single consistent optimal motif length of 9 amino acids is found. For DRB1*15:01 all motif lengths 8-11 had statistically indistinguishable performance (paired t-test).

panel, the 9-mer preference of HLA-DRB1*01:01 is significant, whereas the apparent 8-mer preference of HLA-DRB1*15:01 is not significant. In fact, for the 14 HLA-DR molecules included in the benchmark, only one was found to have a single consistent optimal motif length (DRB1*01:01 with a motif length of 9 amino acids). For all other molecules the method did identify more than one possible optimal motif length. However, all motif lengths fell in the range of 7 to 10 amino acids, and in all cases a 9-mer motif was compatible with being the optimal motif length.

**Improving the LOGO sequence motif representation by an offset correction**

In order to enhance predictive performance, the *NNAlign* method exploits an ensemble of neural networks [36, 63], which have been trained on different subsets of the data, and/or from alternative configurations of the network architecture (i.e. different number of hidden neurons and/or encoding schemes). As a consequence of different architectures and starting conditions, individual networks might disagree on the exact boundaries of the motif. This disagreement would complicate the visualization of the motif if this was represented as a simple overlay of the individual motifs as exemplified in Figure 2.3, where sequence logos for four different networks from the ensemble trained on HLA*DRB1-04:01 binding data are shown in panels A through D. The individual networks agree on identifying the same strong primary anchor residues and positions, however, each single network identifies different ends (i.e. suggests different registers of the same motif; *in casu* starting at positions 1, 2, 2 and 3 of the predicted nonamer peptide). The weak C-terminal primary anchor residue of HLA*DRB1-04:01 probably explains why the boundaries are difficult to determine. A simple overlay of the predictions from individual networks would result in a muddled motif as depicted in Figure 2.3, panel E. Implementing a Gibbs sampler approach, where matrix representations of the core motifs of different networks are aligned, we introduced an offset correction for each network aiming at maximizing the information content of a combined logo representation of the motif. This approach led to a considerable improvement in the visual logo representation of the binding motif (Figure 2.3, panel F). Offset correction is included as an integral part of the method to enhance motif visualization.

**Characterizing the binding motif of HLA-DR molecules using the *NNAlign* method**

To illustrate the power of the *NNAlign* method to capture the binding motifs within unaligned quantitative peptide data, we applied the method to derive sequence logo representations of the 14 MHC class II HLA-DR molecules included in the Wang dataset. *NNAlign* was trained with a binding motif length of 9 amino acids, Blosum encoding, including peptide length and flanking region length, and PFRs of 3 amino acids, homology clustering at

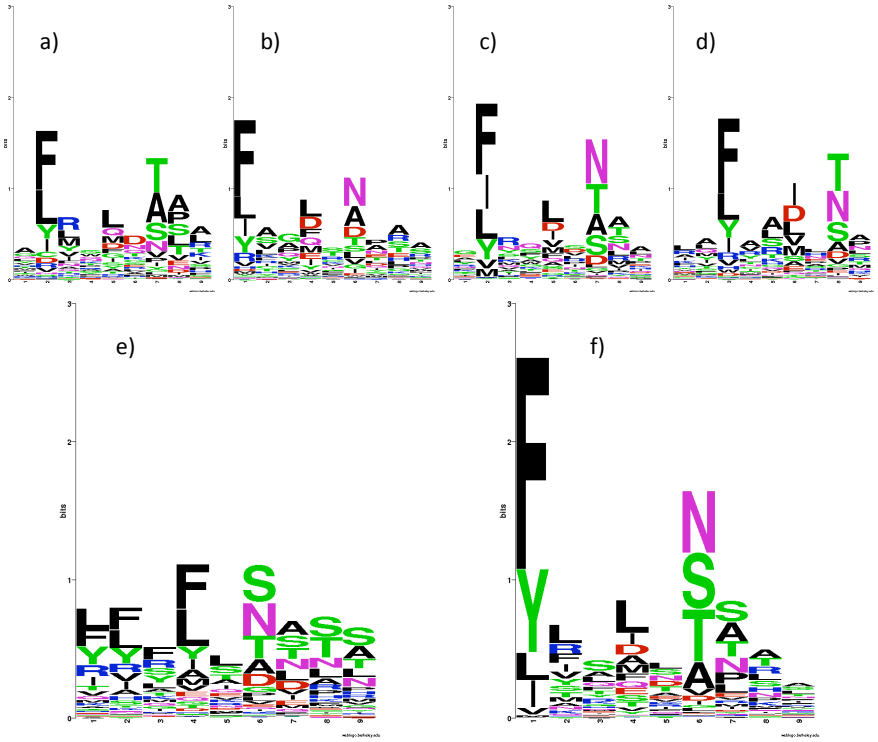



**Figure 2.3. Sequence logos for HLA*DRB1-04:01.** In panels **a)** to **d)** are shown sequence logos for 4 single networks from the network ensemble created with *NNAlign*. The fundamental pattern appears in all these networks, but they place the anchors at different position of the core. **e)** shows the core of the 20 networks ensemble without offset correction; in **f)** offset correction was used to realign the logos to a common register.

threshold 0.8 using all data points, 20 hidden neurons and a 5-fold cross-validation without stopping at the best test set performance. These parameters were found to be optimal in the original *NN-align* paper for MHC class II binding prediction [36], with the only difference that here we choose a single value for hidden layer size for a matter of prediction speed. Individual networks are aligned to a common register using the offset correction strategy previously described. The sequence logos obtained are shown in Figure 2.4. The sequence logos reflect the overall consensus of the binding motifs for HLA-DR molecules, namely a prominent P1 anchor with strong amino acids preference towards hydrophobic amino acids in general, and aromatic amino acids as F and Y in particular, and the presence of two or more additional anchors at P4, P6 and/or P9 each with a unique amino acid preference. Even though most of these motifs exhibit a strong preference for hydrophobic and neutral amino acids at most anchor positions, some dramatic deviations from this general pattern exist. Examples of this are the motifs of DRB1*03:01 and DRB1*11:01 molecules that have strong preferences for charged amino acids at P4 and P6, respectively.

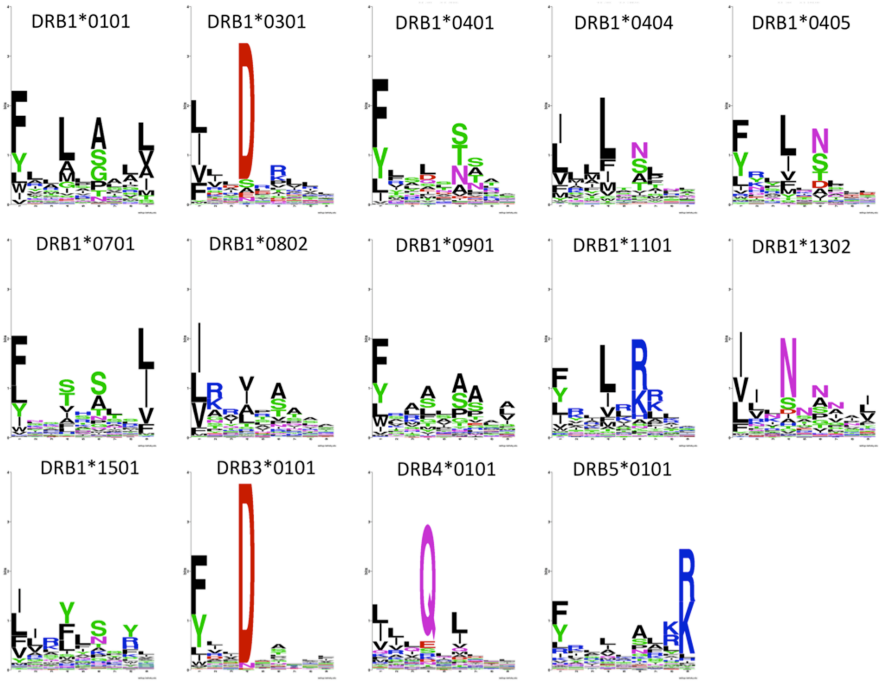



**Figure 2.4. Sequence logo representation of the binding motifs for the 14 HLA-DR molecules contained in the Wang MHC class II data set.** *NNAlign* was trained with Blosum encoding, including peptide length and flanking region length, PFRs of 3 amino acids, homology clustering at threshold 0.8 using all data points, 20 hidden neurons and a 5-fold cross-validation without stopping on the best test set performance.. Sequence logos are calculated as described in material and methods and visualized using the WebLogo program [28].

### Handling large data sets exemplified by protease recognition of high-density peptide microarrays

A peptide microarray containing a total of >100,000 peptides (49,838 of which were unique) was digested with the protease trypsin. The peptide sequences had been synthesized using the theme Ac-GAGAXXXXXGAGA, where Ac- is acetyl blocking the peptide alpha-amino group prior to digestion, and X represents amino acids chosen randomly from the 20 natural amino acids (except lysine, as this residue contains an epsilon-amino group, which even without digestion would be detectable (see Materials and Methods for details)). As a result, free amino groups can only be expressed by trypsin cleaved peptides, which can then be labeled with Dylight549 and quantitated by fluorescence microscopy. A fluorescence microscopy picture of such a digested and stained peptide microarray (Figure 2.5a) demonstrates both the resolution of the photolithographic peptide synthesis strategy and the dynamic range of the free amino group detection strategy. The resulting data was log-transformed and rescaled to obtain a data distribution covering the

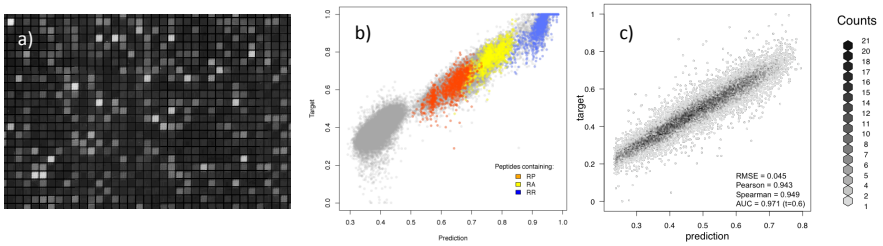



**Figure 2.5. Analysing high-density peptide array data with *NNAlign*. a)** Fluorescence microscopy picture of a peptide microarray. The image is a magnified segment of the peptide chip used in the trypsin cleavage analysis. **b)** Trypsin peptide-chip data. The normalized observed (target) likelihood of cleavage as a function of the prediction score for the trypsin data set. Localizations of peptides containing the pairs of amino acids RP, RA or RR are highlighted in the plot. Proline (P) is known to prevent cleavage after arginine (R), whereas cleavage is observed with other amino acids such as R and A. **c)** Chymotrypsin peptide-chip data. Correlation plot between predicted and measured (target) data from the chymotrypsin data set. Values are binned by their x,y proximity, so that the scatterplot represents the density of data in each bin. *NNAlign* was trained with linear rescaling of the quantitative data, a motif length of 4 amino acids without inclusion of PFR encoding, Blosum encoding of peptide sequences, a combination of 3,7,15 hidden neurons, 10 initial seeds, 5-fold exhaustive cross-validation, training was stopped on the best test set performance.

spectrum between 0 and 1 which, along with the corresponding peptide sequences encoded as Blosum scores without flanking regions, were used to train the *NNAlign* method. Training was done with a motif length of 5, a fixed number of 3 hidden neurons, 5-fold exhaustive validation, and stopping at the best test set performance.

The prediction method yielded a Pearson correlation between measured values and predictions of r = 0.971, a Spearman correlation of $\rho = 0.910$, and receiver operating characteristic (ROC) area under the curve (AUC) of 0.997 (using a target threshold of t = 0.5). The very high performance measures of the resulting *NNAlign* method demonstrate both that the recorded peptide digestion data contains a consistent and intelligible signal, and that the *NNAlign* method is capable of deciphering and predicting this extraordinary large number of sequence-dependent peptide signals. The correlation scatterplot feature of the *NNAlign* web-server output, which compares predicted vs. observed values, further supports the validity of both the peptide microarray and of the *NNAlign* method. The correlation scatterplot for the trypsin digestion data reveals two major populations of peptides, one composed of non-degradable, non-predicted peptides and one containing weakly to strongly degradable, predicted peptides (Figure 2.5b). Few (0.7%) of the former peptides contained Arginine, whereas most (97.1%) of latter peptides contained Arginine. This is exactly what one would have expected from a peptide digestion with trypsin, which is known to cleave at the C-terminal side of amino acids Arginine (and Lysine, which has been excluded here, see above) [64]. For illustration purposes, Figure 2.5b includes a color-enhanced visualization of certain dipeptide sequences (note, this is not a

standard feature of the *NNAlign* server) showing that RP sequences are resistant, RA sequences are quite susceptible, and RR sequences appear extremely susceptible to trypsin digestion. Thus, the known trypsin resistance of RP sequences is both demonstrated by the peptide microarray and subsequently captured by the *NNAlign* method. Note that both the peptide microarray and the *NNAlign* generate a continuous set of measurements and predictions showing that trypsin cleavage involves a more complex interaction than a simple recognition solely of an Arginine residue (and by inference a Lysine residue), which would have resulted in a cleaved/non-cleaved classification [65]. It is also important to note that the detection strategy employed here does not reveal where the protease cleavage has occurred, but merely that the protease has recognized the peptide as a substrate and cleaved it somewhere.

A similar high-density peptide microarray driven approach was next used to address the specificity of the protease chymotrypsin, which is known to preferentially cleave at the C-terminal of tyrosine, phenylalanine and tryptophan (albeit not if followed by a proline). A high-density peptide microarray containing about 50,000 peptides (16,526 unique peptides) was generated according to the theme Ac-GAGAXXXXGAGA, treated with chymotrypsin, labeled with TAMRA and quantitated by fluorescence microscopy. The resulting data was used to train an *NNAlign* method (using the settings described in Figure 2.5c). The correlation scatterplot of the measured versus predicted values exhibits a very strong linear correlation with a Pearson of r = 0.943 demonstrating that the peptide microarray data contains a consistent signal that reliably has been captured by the *NNAlign* method.

### 2.1.3 Discussion

The amount of data deposited in genomic and proteomic databases has been growing exponentially for many years [66]. Due to recent technological advances that have enabled whole-genome sequencing and made whole-proteome analysis a realistic goal, sequence data will accumulate at an even faster pace in the future where single laboratories, even single experiments, can generate data at the "omics" level. This is amply illustrated here where a high-density peptide microarray technology allowed the parallel synthesis of more than 100,000 discrete peptide sequences per array, and the collection of a corresponding number of quantitative peptide-receptor interaction data − all within a single experiment.

The biggest hurdle of future "omics" research may easily become that of making sense of such large-scale biologic sequence data [67]. Presently, the "omics" experimentalist requires assistance from specialized and highly trained bioinformaticians capable of large-scale data handling and interpretation. Ideally, however, he or she should not only be armed with high-throughput data-generation technologies, but also with reasonably easy and robust bioinformatics methods allowing the experimentalist to analyze his or her own data. This would permit an immediate analysis of experimental

results and assist in rational designs of next generation experiments aimed at extending the original analysis e.g. providing *in silico* tools for searches that potentially could encompass entire proteomes. Enabling the same person to do large-scale experiments and analysis should result in a better integration between design, experiment, and interpretation and eventually support the development of new hypotheses. Unfortunately, suitable bioinformatics resources aimed at the non-expert user are currently scarce, and rarely web-based. In our experience, open source software packages such as Weka [68] are not capable of performing concurrent alignment and motif identification, and are not suited for treating large-scale data sets. A widely used method for motif discovery, MEME [69], can perform searches for ungapped sequence patterns in DNA or protein sequences, and offers a user-friendly online server to the untrained user. However, this method is not designed for use in quantitative data, such as peptide-MHC binding or peptide microarray data.

To the best of our knowledge, *NNAlign* is the first web-based bioinformatics solution that allows non-expert users to discover short sequence motifs in quantitative peptide data. As shown here, *NNAlign* easily competes with state-of-the-art methods for identifying peptide-binding motifs of aligned (exemplified by MHC class I) as well as unaligned (exemplified by MHC class II) quantitative peptide sequence data. Further, demonstrating the general utility of *NNAlign*, we have used it to characterize the cleavage specificities of proteases from high-throughput peptide array data. If a sufficient number of training examples can be generated, including negative instances, we could envision applying the method also on data generated by phage display peptide libraries. Other instances of recognition of short specific peptide motifs occurs frequently in biology where they are involved in molecular interaction, recognition, signaling, internalization, modification etc (e.g. phosphorylation, dephosphorylation, trafficking motifs, SH2 and SH3 domains, glycosylation, lipidation, etc.). In contrast to domain recognition, short linear peptide sequences are thought to be particularly difficult to identify due to their unordered structure [70]. *NNAlign* appears to be ideally suited to identify such short linear peptide targets. Due to its simple interface and robust performance, we believe the method to constitute a significant tool providing the non-bioinformatician end-user with the ability to perform advanced bioinformatics-driven analysis of large-scale peptide data sets.

### 2.1.4 Materials and Methods

#### MHC class I data set

The data set of quantitative peptide-MHC class I binding affinity data published by Peters et al. [57] contains data from 48 different human, mouse, macaque and chimpanzee alleles. We selected 12 representative human alleles, and extracted binding data for 9-mer peptides maintaining the subsets of

the original benchmark. This allows comparing the performance of *NNAlign* to the other methods presented in the paper by Peters et al. .

**MHC class II data set**

A large set of over 17,000 HLA-peptide binding affinities was published by Wang et al. [37] containing data from several different human alleles including HLA DR, DP and DQ alleles. For each allele, the predictive performance of various methods was estimated on the similarity reduced (SR) data set, where sequence similarity is minimized in order to avoid overlap between cross-validation subsets. We preserved the same subsets for our cross-validation, for easy comparison of the results and predictive performances.

**Peptide arrays**

Peptide arrays were synthesized by Schafer-N, Copenhagen, Denmark using a maskless photolithographic technique [71] in which 365 nm light is projected onto NPPOC-photoprotected [72, 73] amino groups on a glass surface in patterns corresponding to the synthesis fields. Details of the technique will be published elsewhere, but briefly, the patterns were generated using digital micromirrors and projected onto the synthesis surface using UV-imaging optics. In each layer of amino acids, the relevant amino acids were coupled successively to predefined fields after UV-induced removal of the photoprotection groups in those fields. The couplings were made using standard Fmoc-amino acids activated with HBTU/DIEA in NMP. After coupling of the last Fmoc-amino acid in each layer, all Fmoc-groups were removed in 20% piperidine in NMP and replaced by NPPOC groups coupled as the chloroformate in DCM with 0.1 M DIEA. The procedure was then repeated until all amino acids had been added to the growing peptide chains. Final cleavage of side protection groups was performed in TFA:1,2-ethanedithiol:water 94:2:4 v/v/v for 2 h at room temperature.

**Trypsin data.** Peptide arrays were incubated for 30 min at room temperature with 0.1 g/L bovine Trypsin (Sigma T9201) dissolved in 0.1 M Tris/Acetate pH 8.0. After washing in the same buffer containing 0.1% SDS, the slides were washed with deionized water and air-dried. Staining of amino groups exposed by enzyme cleavage was made by incubation the slide for 30 min in 0.1 mg/mL Dylight549-NHS (Thermo Scientific) in 9:1 v/v n-methyl pyrrolidone:0.1M n-methyl morpholine/HCl pH 8 for 10 minutes.

**Chymotrypsin.** Peptide arrays were incubated for 30 min at room temperature with 0.1 g/L bovine Chymotrypsin (Sigma C4129) dissolved in 0.1 M Tris/Acetate pH 8.0. After washing in the same buffer containing 0.1% SDS, the slides were washed with deionized water and air-dried. Staining of amino groups exposed by enzyme cleavage was made by incubation of

the slides for 10 min in 1 mM 5(6)-TAMRA (carboxytetramethylrhodamine, Fluka 21953) activated with 1 eq HBTU, 2 eq DIEA in n-methylpyrrolidone.

**Recording of signals from peptide arrays.** After incubation with activated fluorochromes, the peptide array slides were washed in the incubation buffer without fluorochrome followed by washings in n-methylpyrrolidone and dichloromethane and air-dried. Images of the arrays were recorded using a MVX10 microscope equipped with a MT10_D fluorescence illumination system and a XM10 CCD camera (all from Olympus). The excitation wavelength was 530-550 nm and the emission filter was 575-625 nm. The images were analyzed using the PepArray analysis program (Schafer-N, Copenhagen Denmark).

### The *NNAlign* Web Server

**Data pre-processing.** The quantitative peptide data entered by the user is rescaled to be between 0 and 1 before being fed to the neural network. The user is also given the option to apply a logarithmic transformation to the raw data, if its distribution appears to be too squashed towards low values. Outliers deviating more than 3 standard deviations from the average, which after rescaling would produce sparse regions in the spectrum with no data, are set at a value of exactly 3 standard deviations. This procedure produces ideal data for artificial neural network (ANN) training, with all values in the range [0:1] and the bulk of the data in the central region of the spectrum. The parameters for the rescaling function are defined separately on each of the training sets used in cross-validation, and then also applied to rescale their relative test sets.

**Subsets for cross-validation.** In a *n*-fold cross-validation, *n* subsets are created from the complete dataset, and at each step *n*-1 subsets are used for training and 1 subset for testing. *NNAlign* offers three alternatives to create the subsets: *i)* random, splits the data into *n* subsets randomly; *ii)* homology clustering, uses a Hobohm 1 algorithm [21] to identify sequences that share an ungapped alignment with more than a specified fraction of matches; *iii)* common motif clustering, looks for stretches of identical amino acid between pairs of sequences as described by Nielsen et al. [53]. For both methods ii) and iii) similar sequences are grouped together in the same subset, but it is possible to choose to only include one representative for each group and disregard the other sequences from training. In this phase, if the input data contains repeated flanks (as might be the case in peptide array experiments, where linker sequences can be attached at the extremities of all peptides), these flanks are discarded, as they would affect the overlap estimation. If the user reckons that the repeated flanks might contain meaningful biological signal, an option allows retaining them in the training data. Note that in common motif clustering, the motif length is taken as the smallest in the interval of length given by the user. Thus, depending on the selected interval

the subsets might be constructed in a different way and that could influence the cross-validated performance.

**Neural network training.** The neural network training is performed as described by Nielsen et al. [36]. Initially, all network weights are assigned random values. From the current network configuration, the method selects the optimal n-mer core (and potential peptide flanking residues) for each of the peptides within the training set. The network weights are next updated, to lower the sum of squared errors between the observed and predicted score, the cores are redefined based on the new network configuration, and the procedure is iterated.

An ensemble of ANNs is trained on the cross-validation subsets, with architecture parameters specified by the user. The motif length, encoding of flanks and peptide length determine the size of the input layer. If the motif length is given as an interval of values, multiple runs of ANN training are performed on the different lengths, and the length that produces the best cross-validated performance in terms of root mean square error (RMSE) is chosen for the final ensemble. The number of hidden neurons may be specified as a list of multiple values, so that an ensemble of networks is constructed with hidden layers of different sizes. Each architecture is trained multiple times, starting from different initial random configurations, to avoid as much as possible choosing sub-optimal solutions producing local minima. Sequences can be presented to the network either with Sparse or Blosum encoding. In Sparse encoding, a vector of length N represents each amino acid, where all values are identical apart from the one representing the observed amino acid. Blosum encoding, on the other hand, takes into account amino acids similarity and partially allows substitutions of similar amino acids while penalizing very dissimilar ones [22].

**Performance measures.** Cross-validation allows estimating a method performance without the need of external evaluation data. The subsets reserved as test-sets are run through the network trained in the same cross-validation step, and Pearson's correlation, RMSE and Spearman correlation are calculated between observed and predicted values.

It is possible to use the internal subsets to stop the training phase on the best test set performance in terms of RMSE. In this mode, performance can be estimated in an exhaustive or in a fast way. Exhaustive $n$-fold cross-validation (CV) consists of a nested CV procedure. At each step, 1 subset is left out as evaluation set, and the remaining subsets are used to generate a network ensemble in an $n$-1 CV training. In this CV training, the selected network configuration is the one that gives the minimum RMSE on the stopping set. Next the predictions for the evaluation data are estimated as a simple average of the prediction values for each network in the training ensemble. The exhaustive CV procedure adds one level to the cross-validation and increases greatly the running time. In alternative, the fast evaluation skips one nested level by using the same subset for stopping and evaluating performance, for a quicker but likely less accurate performance estimation.

**Final network ensemble.** With cross-validated ANN training, each network has been evaluated on data not included in the training. The networks can then be ranked by performance, and only the top N for each cross-validation step will be included in the final ensemble, with N specified by the user. The final network ensemble can be downloaded to local disk, and used for predictions on new data by loading it to the *NNAlign* server submission page.

**Sequence motif logo.** A list of 100,000 random naturally occurring peptides with length $L$ = motif length + 2 × flank length, generated from random UniProt [19] sequences, is presented to the individual networks in the ensemble. For each network, the 1% peptides that obtain the highest prediction scores are used to create a position specific scoring matrix (PSSM) that represents the motif captured by the neural network. Using a Gibbs sampler approach, all PSSMs are aligned to maximize the information content of the combined matrix. This "offset correction" step is obtained by repeatedly attempting to shift the starting position of randomly chosen PSSMs, and accepting/rejecting the move according to the conventional Metropolis Monte Carlo probability relation [74]:

$$P_{\text{accept}} = \min(1, e^{\Delta I / T}) \tag{2.1}$$

Where $\Delta I$ is the change in information content between the new and old offset configuration and $T$ is a scalar that is lowered during the calculation. The process assigns to each PSSM, and to its relative network, an offset value that quantifies the shift distance from other networks (see section 2.2 for details). The re-aligned cores from the 1% scoring of 100,000 peptides are finally used to generate a combined sequence logo with the WebLogo program [28]. The offset correction can be skipped if the user chooses to, and in this case the logo is simply created by presenting the list of random peptides to the ANN final ensemble and selecting the 1% peptides that obtain the overall best score.

**Evaluation data.** Additional data not included in the training can be uploaded to the *NNAlign* server as an evaluation set. Evaluation data must be a list of peptides, with or without associated values, or a file in FASTA format. In the first case, all peptides are run through the trained network ensemble, and scored accordingly to their best alignment core. If values are provided together with peptides, they are assumed to be target values for validation purposes, and statistical measures between these values and predictions are calculated. In the case a FASTA file is loaded as evaluation set, the sequences therein contained are cut into peptides of length $L$ = motif length + 2 × flank length, shifting the starting position of one amino acid at a time. The generated peptides are all fed to the network to identify those that most closely match the motif learned by the ANNs. The results are sorted by prediction value, so that the best candidates are displayed at the top of the list.

### Making sequence logos

Sequence logos were introduced by Schneider et al. [26] as a way to represent graphically the pattern in a set of aligned sequences. The height $R_i$ of each column $i$ in the logo is given as the information content in bits of the alignment at that particular position, and for a sufficiently large number of sequences and a 20-letter alphabet it is calculated as:

$$R_i = \log_2 20 + \sum_a f_{a,i} \log_2 f_{a,i} \tag{2.2}$$

where $f_{a,i}$ is the frequency of amino acid $a$ at position $i$. The relative height $h_{a,i}$ of amino acid $a$ at position $i$ is:

$$h_{a,i} = R_i f_{a,i} \tag{2.3}$$

The value of $R_i$ varies between 0, for a position with maximum entropy, to $\log_2 20$, for a completely conserved position in the alignment. Thus, the height of a column in the sequence logo indicates the importance of a certain position in defining the motif, and the height of each letter in the column the amino acid preference at that position. Amino acid letters are colored according to their chemical properties: polar amino acids (C, G, S, T, Y) are shown in green and (N, Q) pink, basic (K. R, H) in blue, acidic (D, E) in red, and hydrophobic (A, L, I, V, F, M, P, W) in black.

### Author Contributions

Conceived and designed the experiments: MA MN SB. Performed the experiments: MA. Analyzed the data: MA MN. Contributed reagents/materials/analysis tools: CS. Wrote the paper: MA MN OL CS SB.

## 2.2 Offset correction for ANN ensembles

A neural network ensemble is a collection of networks aiming to describe the same problem, but which generally differ in their architecture (e.g. number of hidden neurons or encoding scheme), their initial configuration, or the subset of the data used for training. Ensembles have been shown to be superior to individual network in generalizing the description of many problems [75, 76]. After training a number of ANNs, their outputs must be combined into a single prediction for the ensemble. For regression tasks, predictions are often combined by simply averaging the individual networks output values. For classification tasks (when examples are to be assigned to a discrete number of categories), the consensus classification is commonly chosen by a majority vote. Many other methods exists (reviewed in [77]).

The *NNAlign* method presented in section 2.1 performs two tasks at the same time: identification of the alignment core and prediction of binding affinity. The first is a classification task, the second is a regression task. Following the paradigm discussed above, the optimal alignment core for a given sequence is chosen by majority vote. Among all the possible alignment registers of a peptide, a democratic agreement of the ANNs determines the consensus solution. As for the regression task, the final prediction value is the arithmetic mean of the outputs of all individual networks in the ensemble.

Ensembling networks was shown to give high performance in terms of peptide-MHC class II binding prediction [36]. However, the identification of a common alignment for all the components in the ensemble presents some challenges. As a consequence of different training conditions (distinct subsets, architecture or initial configuration), distinct networks may disagree on the precise boundaries of the alignment core. Although most or all networks may identify the fundamental pattern contained in the data, they may place the motif anchors in different core registers, especially if any of the extreme positions of the binding motif is weak. This situation is exemplified in figure 2.3, where panels A to D show the sequence logos of four different networks trained on HLA*DRB1-04:01 binding data. The salient features of the peptide-MHC motif were captured by all networks, but were placed in different registers within the 9 amino acid core. Therefore, a direct overlay of the individual motifs results in a muddled motif (panel E) and fails to create a consensus motif for the ensemble. The networks, from the point of view of the binding core, are unaligned.

Here, we describe a simple algorithm to calculate the extent of this networks misalignment, which we quantify with an "offset" value for each network, and by that construct a consensus motif for the complete network ensemble.

### 2.2.1 PSSM representation of a network

The first step is to translate the motif captured by each ANN into a position-specific scoring matrix (PSSM). A large number of random natural peptides

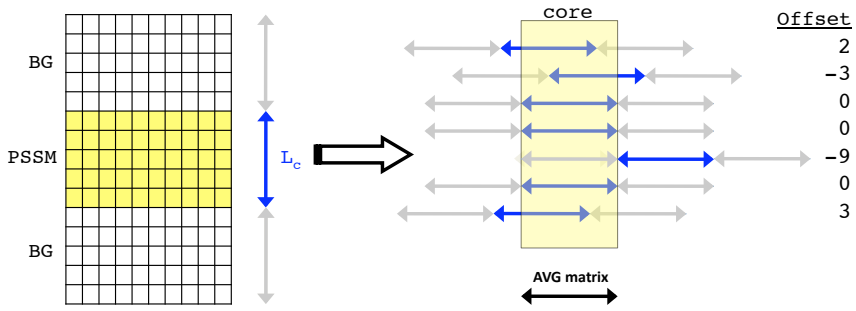



**Figure 2.6. A schematic of the PSSM alignment algorithm.** Each matrix, composed of $L_c$ rows and 20 columns, is elongated with flanks composed of the background frequencies (BG) of each amino acid. Then, the algorithm attempts to find the optimal alignment that maximizes the information content of the average matrix over a core of $L_c$ positions. The numerical values quantifying the shift from the original core position of each matrix are called the offset of each matrix.

(commonly 100,000) are presented separately to each network in the ensemble. For each network, every peptide is assigned a prediction score and an optimal binding core of length $L_c$. The core of the top 1% scoring peptides is used to calculate the frequency of all 20 amino acids in each position of the core. The result is a collection of frequency matrices, each corresponding to a network, with size $L_c \times 20$. Finally, the matrices are all elongated of $L_c$ positions at both extremes, setting the amino acid frequencies at these positions to the background frequencies (i.e. the relative occurrence of each amino acid in natural sequences, see Table 1.1). The function of these flanks will be clear in the next section. The final representation of each network is thus a matrix composed of 20 columns (one per amino acid), and $3 \times L_c$ rows, containing the frequency of a given amino acid at a given position in the alignment.

### 2.2.2 Alignment of PSSMs

The algorithm for PSSM alignment is based on Gibbs sampling. Initially, all matrices are assigned a random offset between $-L_c$ and $+L_c$. The offset value determines from which position, compared to the initial matrix without flanks, the core of the alignment starts for a certain matrix. For instance, if at a given time a matrix has offset $= 2$, the first position in the alignment core for this matrix is position 2, and the core extends to position $L_c + 2$, thus including 2 rows composed of background frequencies. The notation and an example alignment is shown in figure 2.6.

Then, the algorithm performs a number of iterations to find the optimal alignment (the optimal set of offset values) for the set of matrices. At each iteration, the algorithm attempts one of two possible moves: *i)* a single shift move or *ii)* a phase shift move. In the single shift move, one PSSM is chosen randomly, and it is shifted to the right or to the left by assigning a new

random offset value to the matrix. In the phase shift move, the entire alignment is shifted a random number of positions to the left or right. A move therefore produces a new candidate set of offset values. In the new configuration of offsets, a consensus matrix is calculated by averaging the amino acid frequencies of all matrices on the alignment core. We can then evaluate the information content of this average matrix using

$$E = \sum_{i=1}^{L_c} \sum_{A=1}^{20} p_{i,A} \log \frac{p_{i,A}}{q_A} \tag{2.4}$$

where $p_{i,A}$ is the frequency of amino acid $A$ at position $i$, and $q_A$ is the background frequency of amino acid $A$ as in table 1.1.

Whether the move was favorable or not to the energy of the system, can be estimated by comparing the information content before and after the move, obtaining

$$\Delta E = E_1 - E_0 \tag{2.5}$$

The probability of accepting a move is given by the conventional Metropolis Monte Carlo relationship

$$P_{\text{accept}} = \min(1, e^{\frac{\Delta E}{T}}) \tag{2.6}$$

where $T$ is a scalar progressively lowered during the iterations, commonly called the "temperature" of the system. If $\Delta E$ is positive, a move will always be accepted. Moves that lower the energy of the system ($\Delta E < 0$) are only accepted with a certain probability, which depends on the value of $T$. Initially when the temperature $T$ is high many moves, also unfavorable ones, will be accepted. But as $T$ is lowered and the system gets colder, fewer and fewer moves with $\Delta E < 0$ are accepted and the system converges to an energy maximum.

Eventually, when the temperature reaches its minimum and the iterations are over, each matrix has assigned an optimal offset value that maximize the information content of the alignment (Figure 2.6). The offset value quantifies, for each matrix, the relative distance from the optimal alignment core.

### 2.2.3 Including offset in ANN predictions

Once the offset values have been calculated using the PSSM alignment discussed in the previous section, incorporation of the offsets into the neural networks ensemble is straightforward. Each network inherits the offset values of its corresponding PSSM. At prediction time, the starting position in the core for a given query sequence is simply shifted by $o_n$ positions, with $o_n$ being the offset value for network $n$ in the ensemble. In the next chapter we show how offset correction, by combining signals from distinct neural networks, contributed to enhance the characterization of molecules with very weak binding motifs.

# Chapter 3

# The binding motifs of HLA-DP and DQ molecules

In the previous chapter, we demonstrated the power of artificial neural networks in identifying weak sequence motifs in unaligned peptide data. Here, after discussing briefly the function and current understanding of HLA class II molecules, we show how the *NNAlign* method was applied to characterize, at an unprecedented level of detail, the binding specificities of 5 HLA-DP and 6 HLA-DQ molecules among the most common in the human population.

## 3.1 HLA class II molecules

Human major histocompatibility complex (MHC) molecules are surface receptors encoded in a large genomic region of chromosome 6, commonly referred to as the human leukocyte antigen (HLA) system (Figure 3.1). MHC class II molecules, in particular, present antigens in form of short peptides derived from extracellular proteins. Such peptide antigens, potentially originated from extracellular pathogens, are brought to the cell membrane for presentation to $CD4^+$ (helper) T-cells. If the antigen is recognized as foreign, the helper T-cells contribute in initiating an immune cascade aimed to destroy the pathogen.

The three HLA class II loci DR, DP and DQ are among the most genetically variable loci in the human genome, resulting in a very large number of allelic variations and phenotypes in the world population [78]. In HLA-DR molecules the polymorphism is limited to the $\beta$ chain, so that variants are commonly simply identified by their DRB genetic locus (e.g. DRB1*01:01).

Conversely, in HLA-DP and DQ both the $\alpha$ and the $\beta$ chain are highly polymorphic, and both the A1 and B1 loci are used to identify the serotype (e.g. DQA1*01:01-DQB1*05:01).

The binding specificity of HLA-DR molecules (i.e. which peptides are recognized by the MHC molecule) are relatively well studied and characterized. Among hundreds of known HLA-DR variants, only for a small fraction there is a considerable amount of measured peptide binding affinities. However, even in absence of experimental data, pan-specific bioinformatics methods have successfully been applied to predict the binding specificity of most known HLA-DR alleles [79, 80]. Much less is known about the peptide-binding repertoire of HLA-DP and DQ molecules. Most studies have targeted only specific variants suspected to be associated with disease, and have generally been performed on a small scale. Given such scarcity of data, a detailed characterization of the binding specificity of these molecules has not been thus far possible. Only recently larger data sets of binding data for HLA-DP and DQ molecules have been published. In particular, Wang et al. [37] released an unprecedentedly large set of measured MHC class II affinities covering 26 allelic variants, including a total of about 17,000 affinity measurements for 5 DP and 6 DQ molecules. In the following section, we show how the *NNAlign* method was employed to finely characterize the binding specificities of these molecules, and display them as sequence logos. To the best of our knowledge, no previous studies could provide such a quantitative measure of the importance of each position, and relative importance of different amino acids, in the binding core of the HLA molecules.

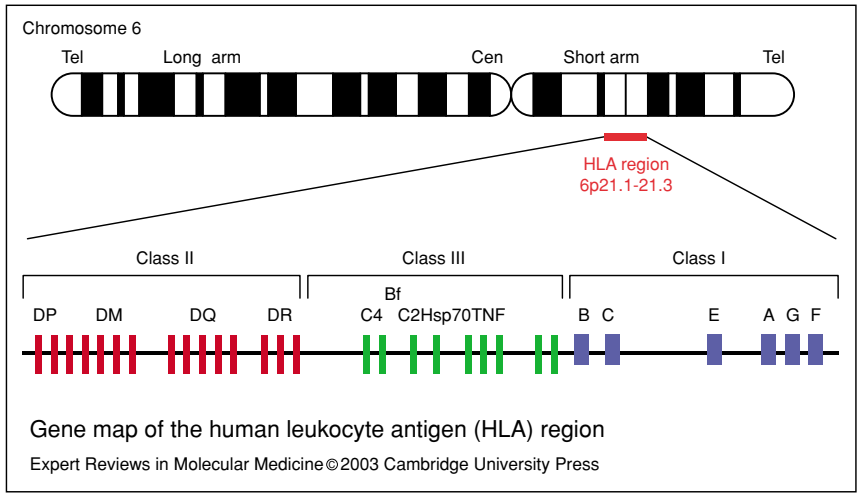



**Figure 3.1. Gene map of the human leukocyte antigen (HLA) region.** HLA class I molecules present peptides derived from cytosolic proteins; HLA class II molecules sample peptides from extracelllular proteins; HLAs corresponding to MHC class III mainly encode components of the complement system. Adapted from *Mehra & Kaur* [81]

## 3.2 Paper II

# Characterizing the binding motifs of 11 common human HLA-DP and HLA-DQ molecules using *NNAlign*

*Immunology*, July 2012 vol. 136(3): 306-311. doi: 10.1111/j.1365-2567.2012.03579.x

Massimo Andreatta*, and Morten Nielsen

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

* Corresponding author - massimo@cbs.dtu.dk

**Abstract**

Compared with HLA-DR molecules, the specificities of HLA-DP and HLA-DQ molecules have only been studied to a limited extent. The description of the binding motifs has been mostly anecdotal and does not provide a quantitative measure of the importance of each position in the binding core and the relative weight of different amino acids at a given position. The recent publication of larger data sets of peptide-binding to DP and DQ molecules opens the possibility of using data-driven bioinformatics methods to accurately define the binding motifs of these molecules. Using the neural network-based method *NNAlign*, we characterized the binding specificities of five HLA-DP and six HLA-DQ among the most frequent in the human population. The identified binding motifs showed an overall concurrence with earlier studies but revealed subtle differences. The DP molecules revealed a large overlap in the pattern of amino acid preferences at core positions, with conserved hydrophobic/aromatic anchors at P1 and P6, and an additional hydrophobic anchor at P9 in some variants. These results confirm the existence of a previously hypothesized supertype encompassing the most common DP alleles. Conversely, the binding motifs for DQ molecules appear more divergent, displaying unconventional anchor positions and in some cases rather unspecific amino acid preferences.

### 3.2.1 Introduction

The MHC performs an essential role in the cellular immune system, and regulates immune responses through presentation of processed antigens to T lymphocytes. The MHC is also widely studied because of its association with many autoimmune and inflammatory diseases, including type I diabetes, rheumatoid arthritis, multiple sclerosis and Crohn's disease, and certain MHC alleles have been linked to susceptibility to infectious diseases such as malaria and HIV (reviewed in [82]).

Unlike MHC class I, which samples peptides from cytosolic proteins, MHC class II molecules present short peptide sequences derived from extracellular proteins. Human MHC class II molecules are heterodimers consisting of an $\alpha$-chain and a $\beta$-chain encoded on chromosome 6 in one of three HLA loci: DR, DP and DQ. Compared with DR molecules, the specificities of DP and DQ molecules have only been studied to a limited extent, and their binding motifs are poorly characterized and understood. The scarcity of binding data for DP and DQ molecules is mainly the result of the relative difficulty, compared with HLA-DR, of obtaining experimental binding data for these molecules, but the common assumption that DR molecules are more important in mediating immune responses has exacerbated the lack of information on DP and DQ. However, a growing number of reports associate certain DP and DQ alleles with several diseases, such as type I diabetes and coeliac disease [82, 83, 84], as well as in cancer [85, 86, 87].

This gap in knowledge between DR and the other class II molecules has only recently begun to be filled, with the publication of larger sets of binding

data for HLA DP and DQ molecules. In particular, a recent study by Wang et al. [37] describes the release of an unprecedentedly large set of measured MHC class II binding affinities covering 26 allelic variants, including a total of about 17 000 affinity measurements for five DP and six DQ molecules. The same study also compared the predictive performance of some of the best available bioinformatics methods on these data, and found that it was possible to obtain reliable binding predictions for DP and DQ at levels comparable to those for DR molecules. The same group, in two additional publications [88, 89] attempted to characterize the binding specificities of a number of DP and DQ molecules using a matrix method called ARB (average relative binding) [90]. However, this method has been shown to perform significantly worse than other comparable approaches for MHC class II binding prediction, such as the *NN-align* method [36]. In this report, we applied the latest version of the *NN-align* algorithm, implemented as the *NNAlign* web-server [91], to exploit the newly available large data sets of peptide binding affinity to DP and DQ molecules and finely characterize the binding specificities of 11 DP and DQ molecules.

### 3.2.2 Materials and Methods

*NNAlign* is a neural network-based method specifically designed to identify short linear motifs contained in large peptide data sets. As a direct result of the method, it identifies a core of consecutive amino acids within the peptide sequences that constitutes an informative motif. The method has been shown to perform significantly better than any other publicly available method for MHC class II binding prediction, including HLA-DP and HLA-DQ molecules [37]. One of the strengths of this approach is the use of multiple neural networks, trained with different architectures and initial conditions, to reduce stochastic factors and at the same time combine information from the different networks in the ensemble to obtain a prediction that is better than what can be obtained from the individual networks. Although this ensemble approach has earlier proved to be highly effective in terms of improving the accuracy for binding affinity predictions [36], it has been demonstrated that the use of network ensembles could lead to a loss in accuracy when it comes to identification of the motif binding core [91]. However, using an offset correction algorithm implemented in *NNAlign*, this problem is resolved allowing not only improved predictive performance for network ensembles but also a more accurate representation of the identified sequence motif.

In this report, we applied *NNAlign* to peptide MHC class II binding data for five HLA-DP and six HLA-DQ molecules to characterize their specificities and binding motifs. The binding data were obtained from the publication by Wang et al. [37]. They comprise a total of 17092 measured peptide MHC affinities, with an average of over 1500 measurements per allelic variant. Each data set was split in five random subsets and, each time excluding one subset, a network was trained on the remaining four subsets. We set the motif length to nine amino acids, and for all the remaining parameters we used

the default values of the *NNAlign* web server: sequences were presented to the networks using Blosum encoding [22], hidden layers were composed of three neurons, training lasted 500 iterations per training example, starting from five different initial configurations for each cross-validation fold, subsets for cross-validation were created using a homology clustering at 80% to reduce similarity between subsets, using the best four networks for each cross-validation step.

The resulting 20 networks in each ensemble, trained on different subsets of the data and from alternative initial conditions, capture motifs that can be different from each other to some extent. They often place the alignment core in a different register, and might disagree on the exact boundaries of the motif. The offset correction algorithm described by Andreatta et al. [91] proved extremely efficient in correcting for this disagreement, allowing realignment of different networks to a common core. This alignment procedure creates a position-specific scoring matrix (PSSM) representation of the motif of each network, and then aligns the matrices to maximize the information content of the combined core. We used a slightly modified version of the algorithm described in detail in a previous publication [91], where PSSMs are extended at both ends with background frequencies before alignment, so allowing the PSSMs to be aligned on a window of the same length as the matrices. This process assigns to each PSSM, and its relative network, an offset value that quantifies the shift distance from other networks. Note that the alignment procedure does not guarantee that the final combined register corresponds to the biologically correct register (in the case of peptide MHC binding, the nine-amino-acid stretch bound in the MHC binding cleft), but rather to the window with the maximum information content. In most of the cases informative positions are also biologically important positions, so the core register would be in the correct place. However, if either terminal of the core has very weak information content (i.e. no particular amino acid preference at terminal positions), the sequences might possibly, although aligned correctly, all be shifted by one or more positions with respect to the biologically correct core register. This is an aspect to keep in mind when interpreting the results, and possibly adjust the register based on previous knowledge about the location of the motif anchors.

An effective way of visualizing the receptor-binding motif is by using sequence logos. Sequence logos were introduced by Schneider and Stephens [26] to graphically represent the sequence motif contained in a set of aligned sequences, where at each position, the frequency of all amino acids is displayed as a stack of letters. The height of a column in the logo is given as the information content in bits of the alignment at that particular position, and the relative height of individual letters is proportional to the frequency of the corresponding amino acid at that position. In this paper, we use such sequence logos to display the HLA-DP and HLA-DQ binding specificities identified by *NNAlign*.

### 3.2.3 Results and Discussion

#### HLA-DP

The five HLA-DP allelic variants were chosen by Wang et al. [37] to cover a high percentage of the human population. Only considering the $\beta$-chain, more polymorphic than the $\alpha$-chain and the main determinant for HLA-DP binding [92, 93], the allele choice provides coverage of about 92% of the average population at the DPB1 locus [89].

The sequence motifs identified by *NNAlign* for the five HLA-DP molecules are shown in Figure 3.2. In general, all variants share a common pattern characterized by anchors at positions P1 and P6, with strong preferences for phenylalanine (F) and other aromatic or hydrophobic amino acids. Additionally, some molecules appear to have a hydrophobic preference at P9 especially for leucine (L). This P9 anchor was previously described for DPB1*04:02 [94], but here we observe it also for other variants such as DPA1*02:01-DPB1*01:01 and DPA1*02:01-DPB1*05:01. In some instances, and notably for DPA1*03:01-DPB1*04:02, the residues at position P7 appear to have influence on the binding specificity of the molecule. This has not been described in previous reports. Another small exception to the P1-P6 hydrophobic/aromatic pattern is observed in the allelic variant DPA1*02:01-DPB1*05:01, where the positively charged amino acids R and K are moderately preferred at P1 together with hydrophobic ones, as was also previously noted [89].

Taken as a whole, there appears to be a large overlap in the peptide-binding specificities of the five DP molecules, characterized by strong hydrophobic/aromatic anchors at P1 and P6, with the few exceptions noted above. Consistent with these observations, previous studies have found considerable overlaps in the peptide repertoires that can bind different DP alleles, and suggested the existence of a DP supertype encompassing the most common variants [89, 94]. Greenbaum et al. [95] on the basis of shared binding repertoires, suggested the presence of two DP supertypes: a main DP supertype (composed of DPB1*01:01, 05:01 and 04:02) and a DP2 supertype (DPB1*02:01 and 04:01). These two subgroups correspond, in our analysis, to molecules with a strong P9 anchor (main DP) as opposed to molecules with weak or no P9 hydrophobic preference (DP2).

#### HLA-DQ

Most efforts in characterizing HLA-DQ binding specificities have been directed towards a few selected molecules, such as DQA1*05:01-DQB1*02:01 (also known as DQ2) or DQA1*03:01-DQB1*03:02 (DQ8) because of their association with disease [96, 97, 98]. The data published by Wang et al. [37] aim to be more comprehensive in terms of human population coverage, and they include binding data for the six most common allelic variants across different ethnicities.

The HLA-DQ sequence motifs identified by *NNAlign* are shown in Figure 3.3. In contrast to the DP variants, which appear to share a common

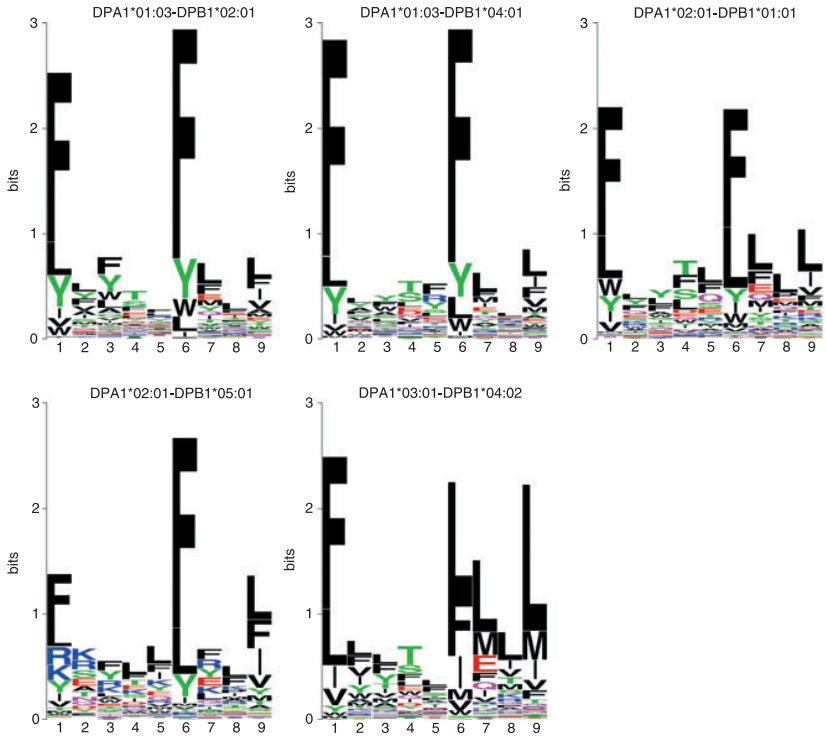



**Figure 3.2. Sequence logos for five HLA-DP molecules.** Hydrophobic amino acids are shown as black, acidic amino acids as red, basic amino acids as blue, neutral and polar amino acids as green and pink. All variants appear to share two main hydrophobic/aromatic anchors at P1 and P6, with an additional P9 anchor for some variants.

supertypical pattern, the DQ molecules show very little overlap in specificity. There do not appear to be common amino acid preferences, and the anchors are found at different positions within the 9-mer core. In particular, DQA1*01:01-DQB1*05:01 shows a strong preference for aromatic residues (F, W, Y) at P5, and secondary anchors at P6 and P7. The only previous report addressing the binding motif of this molecule [88] also found a dominant anchor characterized by a preference for W and F, but placed this anchor at P4, and is generally in disagreement with our findings on other positions. The binding motif for DQA1*01:02-DQB1*06:02 appears loose, with several amino acids allowed at most positions. Previous reports [99, 100] identified mainly a P4-P6-P9 anchor spacing, with small and hydrophobic residues at P4, hydrophobic/aliphatic amino acids such as I, L, M, V at P6, and small residues like A and S at P9. Similar amino acid preferences are reflected in the binding motif detected by *NNAlign*, with additional anchors at P3 and P7. The only pair of molecules that appear to have a somewhat similar specificity is composed of DQA1*03:01-DQB1*03:02 and DQA1*04:01-DQB1*04:02. Both show a dominant anchor at P9, with preference for the

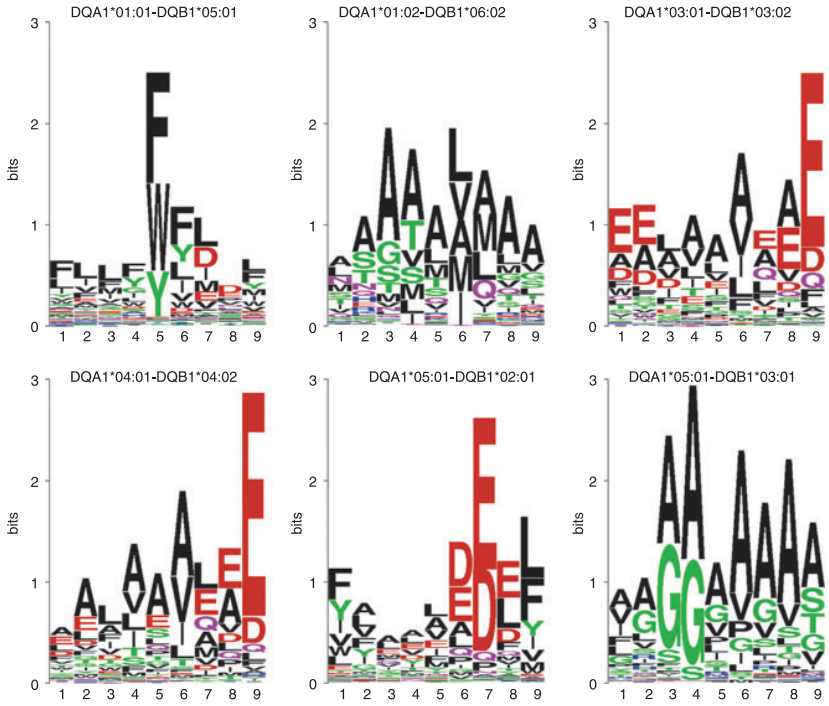



**Figure 3.3. Sequence logos for six HLA-DQ molecules.** Most of the variants display unique anchor positions and spacing, and very diverse amino acid preferences.

acidic residues E and D. Additionally, they both show a preference for hydrophobic amino acids at P6, and mainly for A or E at P8. The strong acidic anchor at P9 was observed before [96, 101]. In the case of DQA1*05:01-DQB1*02:01, previous studies describe a motif with P1 and P9 binding pockets with hydrophobic/aromatic preferences, and acidic residues in the centre of the core, particularly at P4, P6 and P7 [88, 101, 102, 103, 104, 105]. Besides the hydrophobic/aromatic P1-P9, *NNAlign* places the strongest anchor at P7, but with preferences for glutamic acid (E) also at P6 and P8. Finally, the somewhat peculiar sequence motif of DQA1*05:01-DQB1*03:01 seems to just prefer small amino acids such as A, G and S, especially on the central positions of the core, in agreement with the motif previously suggested for this molecule [88].

It is evident that the peptide-binding specificities for HLA-DQ variants are much more diverse than for HLA-DP variants. In particular, the strong hydrophobic/aromatic P1 anchor that generally characterizes all known HLA-DR and DP molecules is not observed here. There appears to be no general pattern in the spacing of the anchors, as well as in the kinds of permitted amino acids. In particular, we find a preference for acidic amino acids close to or at the C-terminal of the binding motif for three of the six molecules,

and generally, the motifs seem rather promiscuous, with several residues allowed in the binding groove of the MHC.

### 3.2.4 Conclusions

In this report, we applied a state-of-the-art neural network-based method, *NNAlign*, to characterize the binding specificities of five HLA-DP and six HLA-DQ molecules. The allelic variants are among the most common human MHC class II molecules at the two HLA loci DP and DQ, covering a large percentage of the human population [88, 89].

For what concerns HLA-DP, there appears to be a common pattern in all the five variants under consideration, with primary anchor positions at P1 and P6 with preference for hydrophobic and aromatic residues. Some variants show an additional hydrophobic anchor at P9 and other minor differences, but in general there appears to be a consistent overlap in the binding specificities of all five molecules. The same cannot be said for HLA-DQ, where most of the molecules have very different anchor positions, anchor spacing and amino acid preferences. Hence, there does not seem to be a supertypical mode of binding for DQ, and each variant appears to be characterized by a distinct binding specificity. The most striking observation for the DQ loci binding motifs is the preference for acidic amino acids close to or at the C-terminal of the binding groove. Such an amino acid preference has not, to the best of our knowledge, previously been described for any HLA class I molecules, and has only sporadically been reported for HLA class II molecules. Binding predictions (including identification of the binding core) for any peptide sequence to all the alleles described in this report can be obtained at the NetMHCII server (`http://www.cbs.dtu.dk/services/NetMHCII`).

The binding motifs described in this work confirm most of the observations brought up by previous studies, but also highlight some interesting differences. Importantly, the sequence logo representation provides a quantitative measure of the relevance of each position in the binding core, and the relative importance of each amino acid, in determining the specificities of a given molecule, a differentiation that was not obtained in previous studies. The study first and foremost demonstrates the power of the *NNalign* method to, in a fully automated manner, identify and characterize the receptor-binding motif from a set of peptide-binding data. Secondly, it underlines the importance of generating such peptide data sets to carry out receptor-binding motif characterizations, gain insights into the peptide-binding repertoire of MHC molecules and reveal details about which amino acids and amino acid positions are critical for binding and, potentially, for peptide immunity.

# Chapter 4

# Identifying multiple specificities in peptide data

Chapters 2 and 3 addressed the problem of identifying sequence motifs in unaligned peptide data. In these chapters, it was shown how artificial neural networks could be used to find the optimal local alignment in sets of peptide data and at the same time produce quantitative predictions for new occurrences of the identified sequence motif. As discussed in section 1.4, ANNs have the power of capturing higher-order correlations between positions in a motif. Thus, if presence of a given residue at some position influences the amino acid preference at another position, such correlation can be effectively captured by ANNs with hidden neurons.

Non-linear models such as ANNs are very powerful for learning and predicting patterns. However, in presence of positional correlations this non-linearity poses some difficulties when it comes to visualizing and interpreting the identified sequence motifs. A sequence logo can only show amino acid preferences in a linear fashion and assumes that positions are independent from each other. For example, imagine that a certain receptor displays a preference for ligands with a charged residue at one of two adjacent positions, but not at both positions at the same time. The relationship regulating this interaction is a XOR (or "exclusive disjunction"), where the binding event occurs only if exactly one of the two positions presents a charged amino acid. This relationship, though rather simple, is not linear and cannot be represented by a sequence logo. Attempting to plot instances of the binding motif with a linear device such as a sequence logo would give the erroneous impression that charged residues are allowed simultaneously at both positions, while they should be mutually exclusive.

The problem can be made linear by considering the two mutually exclusive conditions as two separate "classes" of binders. That is, creating two

separate clusters containing peptides with one or the other mode of binding. The logo representation of the two clusters would now truthfully specify the preference for a charged residue at one of the two positions, but not at both positions. Generalizing from this simple example, we can imagine representing a binding motif that contains positional correlations by subdividing its positive instances into a number of clusters devoid of positional correlations.

In a recent study Gfeller et al. [24] showed that positional correlations are widespread in peptide recognition domains, and that such positional correlations originate from the multiple specificity of the receptors. The specificities of these domains can be represented by multiple PSSMs, allowing the visualization of the different binding modes by multiple sequence logos [24, 106]. Signals of higher order correlation between different amino acids have been found in MHC binding peptides [25], and in these cases the investigation of sub-specificities for individual molecules may elucidate their mechanism of binding.

In the following paper, we propose a method for the identification of multiple specificities in peptide data sets. The novelty of the Gibbs clustering algorithm lies in its ability to simultaneously align and cluster peptide data. Other available methods are limited to one of these aspects at a time, dealing only with single specificities or requiring the data to be pre-aligned prior to clustering [33, 91, 24, 69, 107]. We show applications of the method to a range of different benchmark data sets, including pre-aligned and unaligned peptide data covering the MHC class I, MHC class II and SH3 domain systems.

## 4.1 Paper III

# Simultaneous alignment and clustering of peptide data using a Gibbs sampling approach

Accepted with minor revision in *Bioinformatics*, September 2012

Massimo Andreatta*[1], Ole Lund[1], and Morten Nielsen[1,2]

[1]Center for Biological Sequence Analysis, Department of Systems Biology, Technical University of Denmark, DK-2800 Lyngby, Denmark [2]Instituto de Investigaciones Biotecnológicas, Universidad de San Martín, CP(1650) San Martín, Argentina

* Corresponding author - massimo@cbs.dtu.dk

**Abstract**

**Motivation:** Proteins recognizing short peptide fragments play a central role in cellular signaling. As a result of high-throughput technologies, peptide-binding protein specificities can be studied using large peptide libraries at dramatically lower cost and time. Interpretation of such large peptide data sets however is a complex task, especially when the data contain multiple receptor binding motifs, and/or the motifs are found at different locations within distinct peptides.
**Results:** The algorithm presented in this paper, based on Gibbs sampling, identifies multiple specificities in peptide data by performing two essential tasks simultaneously: alignment and clustering of peptide data. We apply the method to de-convolute binding motifs in a panel of peptide data sets with different degrees of complexity spanning from the simplest case of pre-aligned fixed-length peptides, to cases of unaligned peptide data sets of variable length. Example applications described in this paper include mixtures of binders to different MHC class I and class II alleles, distinct classes of ligands for SH3 domains, and sub-specificities of the HLA-A*02:01 molecule.
**Availability:** The Gibbs clustering method is available online as a web server at `http://www.cbs.dtu.dk/services/GibbsCluster`
**Contact:** mniel@cbs.dtu.dk
**Supplementary information:** Supplementary Data are available at *Bioinformatics* online.

### 4.1.1 Introduction

Peptides are short amino acid sequences occurring ubiquitously in biological processes, such as metabolism, signal transduction and immune response. They are also extensively used in research to mimic functional or (linear) structural aspects of proteins and protein interactions. The advantage of using peptides lies in the relative ease in generating large libraries of sequences, such as in phage display technologies [3, 108]. More recently, developments in high-throughput peptide microarrays have allowed producing large-scale data sets of peptide-ligand interactions, and have been applied to various problems including antibody-antigen interactions, peptide-MHC binding, kinase binding motifs and other receptor-ligand interactions [4, 13, 42, 109].

Identifying receptor-ligand binding motifs within peptide data sets is a highly challenging task for at least two major reasons which we term alignment and poly-specificity. The alignment problem arises because most receptor motifs are weak and short making identification of the binding register within the ligands not trivial [33]. The poly-specificity problem arises because receptor-ligand data sets often contain multiple motifs either due to the experimental setup or to the actual poly-specificity of the receptor [110]. Several bioinformatics methods have been developed attempting to deal with these challenges and detect subtle sequence signals in peptide data sets, including motif alignment [69], Gibbs sampling [34], Hidden Markov

Models [35] and artificial neural networks [36]. In particular, artificial neural networks (ANNs) have shown a high performance on this kind of data [37, 91]. Significant correlations between residues have been found in peptide interaction domains [24]. Although positional correlations can be accurately captured by ANNs, the specificities of such domains can in many cases more intuitively be represented by multiple position-specific scoring matrices (PSSM) [24, 106, 107]. Multiple PSSMs allow visualizing poly-specificities as sequence logos of the different binding modes.

While the above methods attempt to deal with the challenges involved in motif identification in peptide data sets, they all suffer from the limitations of only dealing with single specificities or requiring the input data to be prealigned to a common motif. In this paper, we describe a novel approach for effective alignment and clustering of peptide data going beyond these limitations. In the Gibbs clustering method, alignment and specificity clustering are performed simultaneously by sampling the space of possible solutions using a Gibbs sampling strategy. Each cluster is represented by a PSSM, and the method aims at maximizing the information content of individual matrices while minimizing the overlap between distinct clusters.

### 4.1.2 Data sets

#### MHC class I data

The MHC class I data set was extracted from the Immune Epitope Database [111]. We selected representative alleles for each of the 12 HLA supertypes identified by Lund et al. [8], and considered as binders all peptides with affinity < 500nM. The representative alleles for each supertype are: HLA-A0101 (A1); HLA-A0201 (A2); HLA-A0301 (A3); HLA-A2403 (A24); HLA-A2601 (A26); HLA-B0702 (B7); HLA-B0801 (B8); HLA-B2705 (B27); HLA-B3901 (B39); HLA-B4001 (B44); HLA-B5801 (B58); HLA-B1501 (B62). The 12 MHC class I molecules have rather divergent binding motifs, and cross-binding across alleles is limited, with ≈90% of the measured binders in IEDB being exclusive binders of a single molecule. This should ensure a data set where the actual number of specificities is known, making it ideal to benchmark the method with respect to the real number of clusters.

#### MHC class II data

HLA-DR binding data for the two molecules HLA-DRB1*03:01 and HLA-DRB1*04:01 was taken from the large data set published by Wang et al. [37]. As in the case of MHC class I, we selected peptides with affinity < 500nM as binders. HLA-DR alleles are highly promiscuous, and often the same peptide cross-binds to several alleles. In order to reduce this promiscuity and obtain an orthogonal data set we excluded, for each allele, peptides with predicted cross-binding. Cross-binding was predicted using the NetMHCIIpan method [79] with a strict binding threshold of 50% rank score (that is a peptide with a predicted binding affinity equal to or higher than 50% rank score

is assumed to cross-bind). The data set was composed of respectively 202 and 201 binders to the molecules HLA-DRB1*03:01 and HLA-DRB1*04:01.

#### SH3 domain binding data

We used the phage display data for the Src SH3 domain from the recent publication by Kim et al. [107]. The raw data set consists of 2,457 unique peptide sequences identified to bind to the Src SH3 domain. All the peptides are 12 amino acids long and are unaligned with respect to the binding motif to the SH3 domain. We extended all sequences with artificial flanks composed of X (any amino acid) to both extremes, to give the alignment algorithm freedom of movement in the search of the optimal alignment window.

#### HLA-A*02:01 affinity and stability data

In a recent study, Harndahl et al. [112] published a data set of measurements of binding affinity and stability to HLA-A*02:01 for 739 peptides. We limited the data set to peptides with binding affinity stronger than the conventional 500nM, resulting in 650 peptides with measured affinity and stability.

### 4.1.3 Methods

The Gibbs clustering algorithm attempts to group the input peptide data into a number of clusters and for each cluster identify the optimal local sequence alignment based on the optimization of the fitness of the system in terms of Kullback-Leibler distance (KLD) sum of the alignments. The KLD allows measuring the information gain of an observed amino acid distribution compared to a background distribution (the frequency of each amino acid in random protein sequences). A given alignment can be represented by a log-odds (LO) weight matrix, which summarizes the amino acid preferences for each column of the alignment. Throughout the paper, we graphically represent LO matrices using the sequence logo visualization tool Seq2Logo [32].

#### Log-odds matrices

A log-odds weight matrix is calculated as $\log(p_{A,j}/q_A)$, where $p_{A,j}$ is the frequency of amino acid $A$ at position $j$, and $q_A$ is the background frequency of $A$. These frequencies are calculated as described in Nielsen et al. [33], including heuristic sequence weighting and pseudo-count correction. To avoid the creation of small highly specialized clusters, we introduce an additional term to the log-odds matrix calculation to account for the size of the alignment. In our scheme, terms in the PSSM are calculated using:

$$\mathrm{LO}_{A,j} = \frac{n}{n+\sigma} \log \frac{p'_{A,j}}{q_A} \tag{4.1}$$

where $n$ is the number of peptides in the alignment, $\sigma$ is a weight on small clusters, and $p'_{A,j}$ is the pseudo-count corrected frequency. The function of

$\sigma$ is to flatten the log-odds matrix when the alignment is composed of few sequences ($n$ small), but its effect is minor when the matrix is constructed on many data points ($n$ large). Practically, it avoids the creation of small and specialized alignments, favoring instead larger and more general ones.

A peptide $x$ can be scored simply by adding the LO values for the amino acid found at each position in $x$:

$$S = \sum_j \mathrm{LO}'_{A,j} \tag{4.2}$$

where $j$ is the index over the positions in the alignment core, and $A$ is the amino acid found at position $j$ in $x$. However, when evaluating the fitness of a given sequence $x$ in an alignment (where $x$ is part of the alignment), we must take the precaution of excluding $x$ from the matrix calculation before doing the evaluation. We call $\mathrm{LO}'_{A,j}$ the log-odds matrix made without sequence $x$.

**Scoring function**

In the general case, a Gibbs clustering solution is composed of $g$ clusters, each with a corresponding alignment and LO matrix. When evaluating a clustering solution, we aim to maximize the intra-cluster fitness of the alignment while minimizing the similarity between different clusters. In other words, the distance between points in the same cluster should be as small as possible, whilst the distance between points in different groups should be maximal. In the Gibbs clustering algorithm, we implement this maximization using the relationship:

$$S_i^* = S_i - \lambda \max_{\substack{1 \le n \le g \\ n \ne i}} (S_n, 0) \tag{4.3}$$

where $S_i$ is the score of a given peptide to the log-odds matrix $\mathrm{LO}_{A,j}$ of cluster $i$. Note that, as discussed above, the log-odds matrix of group $i$ is calculated excluding the peptide to be scored. The *max()* part of the equation determines the inter-cluster similarity, i.e. which cluster is the closest to cluster $i$. If we imagine to have, besides the $g$ clusters given by the data, and additional cluster composed of the universe of natural peptides, the amino acid frequencies $p_{A,j}$ in this extra group would be equal to the background frequencies $q_A$ for any amino acid $A$. Thus $\log(q_A/q_A) = 0$ in equation 4.1, leading to a $\mathrm{LO}_{A,j}$ matrix composed of zeros which gives scores $S_{\mathrm{BG}} = 0$ for all sequences. This justifies the zero in equation 4.3, and provides a generalization for the case where there is only one cluster, with $S_i^* = S_i$ .

The parameter $\lambda$ modulates the weight of inter-cluster similarity on the final sequence score. For $\lambda = 0$ overlap between clusters is not penalized, leading to tight but promiscuous clusters. Large $\lambda$ values put emphasis on inter-cluster similarity, at the expense of consistency within the same group.

Equation 4.3 defines the energy function of a single sequence in the alignment. The overall score of the alignment/clustering is given by the average

score of all sequences in the data set. The fitness of the system can be thought of as the relative entropy or Kullback-Leibler distance (KLD) from the background model made on random peptides.

**Moves of the algorithm**

Initially, peptides are distributed randomly in $g$ clusters. Then the algorithm proceeds with a number of moves to align and cluster the sequences and optimize the KLD of the alignment/clustering. The probability of accepting a move is given by:

$$P = \min(1, e^{dE/T}) \tag{4.4}$$

where $dE$ is the energy change as a result of the move, and $T$ is a scalar commonly known as the temperature of the system, lowered by discrete steps during the iterations.

The algorithm consists of 3 different moves: *i) Single sequence move:* in this move, we attempt to transfer a peptide $x$ from one group $G_o$ to a destination group $G_d$ chosen at random. The score $S_o^*$ of $x$ in its original cluster is calculated using equation 4.3, selecting the core register that gives the highest score. In the same way, $S_d^*$ is obtained for the destination group. The move is then accepted or rejected following equation 4.4, where $dE = S_d^* - S_o^*$. *ii) Simple shift:* this move attempts to move a peptide $x$ within a group, by applying a random shift to the alignment core of $x$. The score of $x$ is calculated before and after the shift, and the $dE$ between the two configurations determines whether the move is accepted or rejected according to equation 4.4. *iii) Phase shift:* the entire alignment of a group $G_o$ is shifted a random number of positions to the left or to the right. This move may be important if the alignment reaches a local minimum where the sequences are optimally aligned to each other but the core window is not centered on the most informative motif. As in the other moves, the configurations before and after the move are compared to calculate whether the move is favorable or unfavorable, and accepted/rejected following equation 4.4.

The *simple shift* and *phase shift* moves have been described before for multiple sequence alignment [33, 34]. The new feature of the Gibbs clustering method is the additional *single sequence* move, which allows transferring sequences between different clusters. The three moves are generally performed with different frequency. The *simple shift* move, with the lowest impact among the three moves, is attempted at each iteration. *Single sequence* moves are performed every $F_r$ iterations. *Phase shifts*, which affect at the same time all peptides in a given clusters, would generally be the least frequent and occur every $F_s$ iterations, with $F_s > F_r > 1$. Throughout the paper these parameters are fixed to $F_r = 10$ and $F_s = 1,000$. The default cooling schedule uses 20 linear temperature steps starting from an initial $T$ of 0.8 down to $10^{-5}$.

### Trash cluster to collect spurious sequences

The algorithm allows including an additional cluster, called "trash-cluster", to collect the peptides that appear not to match any of the motifs being identified. The behavior of the trash-cluster is identical to any of the other clusters, with the difference that sequences in the trash cluster do not contribute to the overall score of the system. The trash-cluster can be thought of as the universe of all natural peptides (i.e. the background model) and peptides can be moved in and out from the trash-cluster with probability defined by the Monte Carlo relationship (equation 4.4), where the score to the trash-clusters is always equal to the background baseline (zero by default, but can be set to different values to adjust the levels of sensitivity and specificity).

### Measures of clustering quality

As a measure of clustering quality, we used the Adjusted Rand Index (ARI). This measure is based on the well-known Rand index [113], but corrected for chance and class size. We implemented the ARI corrected for chance as in Hubert and Arabie [114]. As a term of comparison, we also used a modified version of the Matthews correlation coefficient (MCC) extended to more than the conventional two classes (positives and negatives). In the general case where $A$ mixed specificities are grouped in $C$ clusters, a MCC is initially calculated for each cluster. The true positives (TP) for group $C_i$ are given by the class $A_i$ with highest number of sequences in $C_i$, the false positives (FP) by the number of sequences in $C_i$ not belonging to $A_i$, the false negatives (FN) by the number of sequences labeled $A_i$ not found in $C_i$, and the true negatives (TN) are all the remaining sequences. The MCC for the entire matrix is then

| Cluster | $C_1$ | $C_2$ | ... | ... | $C_C$ | Row sum |
|---|---|---|---|---|---|---|
| $R_1$ | $n_{1,1}$ | $n_{1,2}$ | ... | ... | $n_{1,C}$ | $n_{1\bullet}$ |
| $R_2$ | $n_{2,1}$ | $n_{2,2}$ | ... | ... | $n_{2,C}$ | $n_{2\bullet}$ |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| $R_R$ | $n_{R,1}$ | $n_{R,2}$ | ... | ... | $n_{R,C}$ | $n_{R\bullet}$ |
| Column sum | $n_{\bullet 1}$ | $n_{\bullet 2}$ | ... | ... | $n_{\bullet C}$ | |

**Figure 4.1. Confusion matrix and notation for calculation of ARI and MCC.** In the case of a clustering problem, the two partitions being compared are the predicted clusters (rows) and the labels for the actual classes (columns). Row sums and column sums are used in the calculation of the ARI. The MCC for the entire matrix is calculated as the average of the MCC of each row. Highlighted in different colors are the 4 classes of prediction for cluster $R_2$, assuming that $n_{2,2}$ is the highest value in $R_2$: yellow - true positives; blue false negatives; green false positives; red true negatives.

calculated as the average MCC of each cluster. The notation for ARI and MCC calculation is also illustrated in Figure 4.1.

#### Training from multiple initial seeds

Gibbs sampling is a heuristic rather than a rigorous optimization procedure. Therefore, it cannot guarantee that the most optimal solution is always reached from any starting configuration. A common procedure to boost performance is to repeat the sampling from a number of initial random configurations, and select the solution that appears to be optimal in terms of the fitness function that governs the system. Clearly, this is a sound procedure only if optimal fitness (KLD) corresponds to optimal clustering of the data. We investigated the correlation between fitness and quality of the clustering on MHC class I data sets containing different number of specificities. Binders to different alleles were combined to obtain mixtures of 5 to 8 alleles, and then the Gibbs clustering algorithm was used to recover the distinct motifs. For each allele combination, we ran the algorithm from 40 random initial configurations, measuring for each the fitness in terms of KLD and the clustering quality in terms of ARI.

In general, we observe that both KLD and ARI tend to decrease as the number of alleles in the mixture increases (Figure 4.2). Yet, in the case of MHC class I where motifs are very strong and distinct from each other, it is possible to reconstruct with high accuracy even up to 8 different specificities. The same considerations can be made if we measure clustering quality in terms of MCC instead of ARI, which correlates in very similar fashion to KLD (Figure 4.3). These results show that, only based on the KLD, it is possible to filter out sub-optimal solutions. By running the algorithm from different starting conditions, and selecting solutions with high KLD, the method achieves a higher classification performance. Multiple seeding and automatic selection of the optimal solution are integrated in the Gibbs clustering algorithm.

### 4.1.4 Results

The Gibbs clustering algorithm performs two essential tasks simultaneously: alignment and clustering of peptide data. Here, we use the method to deconvolute binding motifs in a panel of different peptide data sets with different degrees of complexity spanning from the simplest case of pre-aligned fixed-length peptides, to cases of unaligned peptide data sets of variable length.

#### Pre-aligned data - Mixtures of binders to MHC class I alleles

In order to benchmark the clustering aspect of the Gibbs algorithm, we used a set of pre-aligned fixed-length peptides with experimentally confirmed binding to representatives of the 12 MHC class I supertypes (see Data sets section). These 12 MHC molecules all have highly specific binding motifs

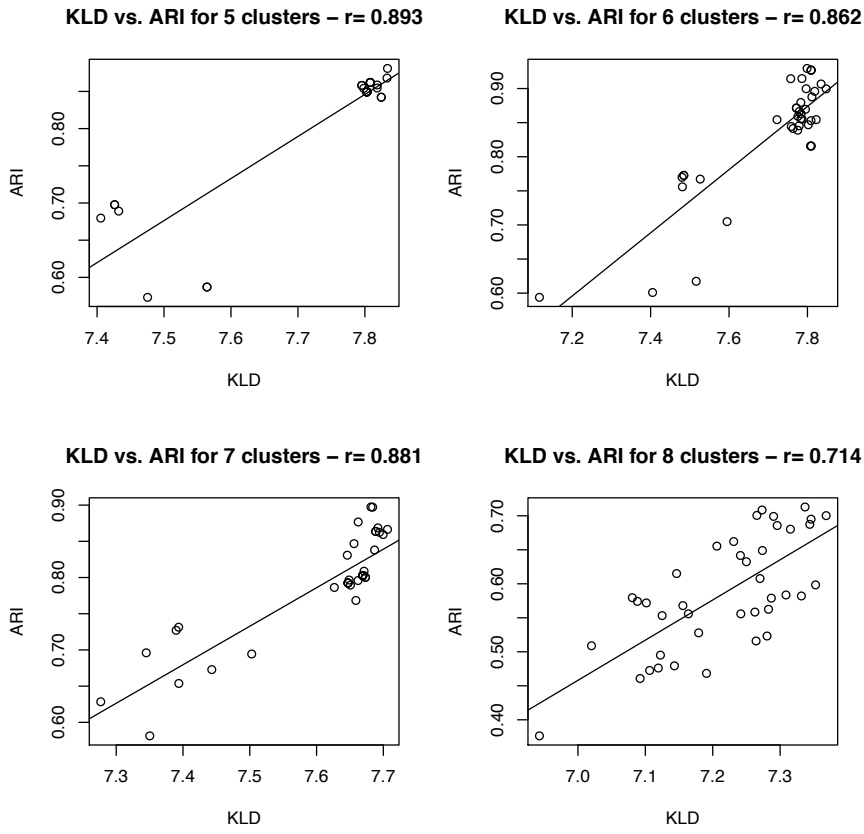



**Figure 4.2. Correlation between KLD and quality of clustering (in ARI) for mixtures of 5, 6, 7 and 8 MHC class I alleles.** The ARI is a measure of how well the algorithm reconstructed the original classes, the KLD represents the energy of the alignments including distance within and between clusters. The individual data points were collected by starting the algorithm from 40 random initial conditions. The plots show that, only based on the KLD, it is possible to filter out most of the sub-optimal solutions and achieve higher performances by selecting solutions with high KLD.

with limited mutual overlap [8]. For each number of alleles $n$ = {1,2,. . . ,8}, 10 different combinations of $n$ alleles were constructed randomly from the pool of the 12 MHC molecules. For each data set, the algorithm was used to cluster the peptides into $c$ = {1, 2,. . . , 12} groups and the $c$ with optimal KLD score was recorded. Figure 4.4 shows the results of this calculation. For $\lambda = 0.5$, the number of predicted motifs correlates well with the actual number of alleles in the data set. With smaller values of $\lambda$, the method tends to over-estimate the number of motifs, while for larger $\lambda$ clusters with shared similarities are more heavily penalized and are merged into fewer clusters. The predictions are most consistent (lowest variations in the optimal number of clusters) on mixtures of few alleles. This is a natural consequence of

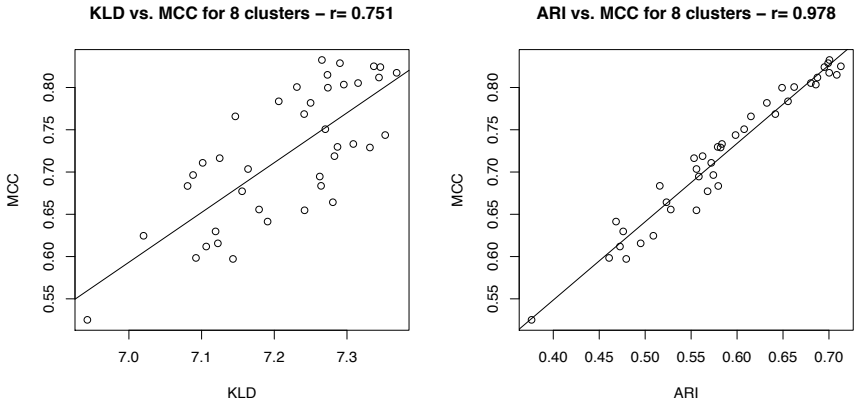



**Figure 4.3. Correlation between KLD and two measures of clustering quality (MCC and ARI).** MCC and ARI are measures of how well the algorithm reconstructed the original classes, the KLD represents the energy of the alignments including distance within and between clusters. The left panel shows that most of the best solutions (high MCC) have the highest values of KLD. ARI and MCC correlate very strongly (right panel) and appear to be equally good measures for cluster quality.

both the increased complexity of the search space as the number of alleles is increased, and the promiscuity of MHC binding peptides. Although the 12 MHC class I molecules share very limited overlap in specificity, a larger collection of alleles increases inevitably the chance of including cross-binding peptides in the data set.

### Unaligned data - Mixtures of binders to MHC class II alleles

To demonstrate the performance of the Gibbs clustering method on data sets of unaligned peptides of variable length, we turned to the MHC class II system. As opposed to MHC class I molecules, which in the vast majority of cases interact only with peptides of length between 8 and 10 amino acids, MHC class II molecules can bind peptides of highly variable length [60]. Binding of a peptide to a MHC class II molecule is primarily determined by a core of 9 amino acids, but the location of the 9-mer core within the peptide is not known *a priori*. Therefore, MHC class II binding data is by nature unaligned with respect to the binding core.

The Gibbs clustering algorithm was applied to identify motifs in a set of binders to the MHC class II HLA-DRB1*03:01 and HLA-DRB1*04:01 molecules. Compared to MHC class I, class II alleles share a high degree of overlap in their binding specificities. This promiscuity between different MHC class II molecules complicates the performance evaluation of the clustering algorithm, as a peptide may match the motif of multiple alleles, in which case it is not clear in what cluster the sequence should be rightfully placed. To lower this potential degree of cross-binding, the data set was constructed to include experimentally confirmed binders with weak predicted

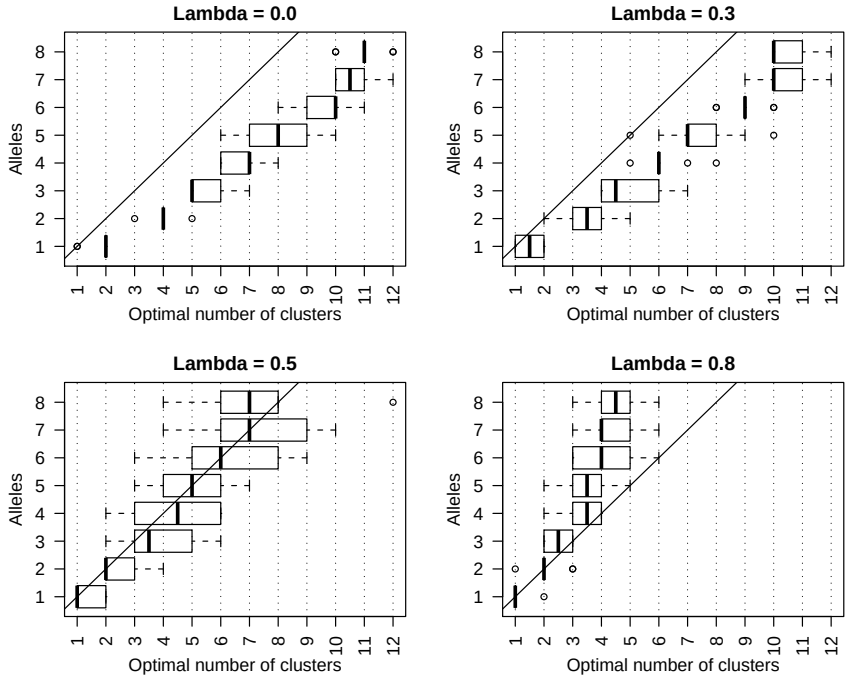



**Figure 4.4. Box-and-whisker plot showing the optimal number of clusters on mixtures of different MHC class I alleles.** The algorithm was run on 10 different random combinations of $n$ alleles, where $n = \{1 \ldots 8\}$, starting with $c = \{1 \ldots 12\}$ clusters for each combination. The optimal number of clusters of each of the 10 combinations is the $c$ with highest KLD of the system. The four panels show the predicted number of clusters for four different values of $\lambda$ for a fixed value of $\sigma = 10$. With $\lambda = 0.5$ the correlation between number of alleles in the data set and predicted number of clusters falls approximately on a straight line with slope = 1.

cross-binding potential (for details refer to Data sets section). We maintained the same parameters used for the MHC class I benchmark, except for $\lambda$ which was increased to 0.8 to avoid the creation of excessively small and specialized clusters (running the algorithm with $\lambda = 0.5$ resulted, in particular, in the DRB1*03:01 peptides being subdivided into several small and highly specialized clusters). Additionally, since HLA-DR molecules are known to prefer hydrophobic amino acids at position P1, we imposed a preference for this kind of amino acids in the Gibbs sampling moves as proposed by Nielsen et al. [33]. The algorithm was run multiple times to create 1-4 clusters, each started from 5 different random configurations. For each cluster size, the solution with the highest KLD score was recorded. The optimal solution indicated the presence of two clusters (Figure 4.5), and the corresponding motifs are shown in Figure 4.6. The main distinctive feature in the logos of Figure 4.6 is the acidic (D) anchor at position P4 and a basic (K/R) anchor at position 6 of the first motif, which are absent in the second logo. These

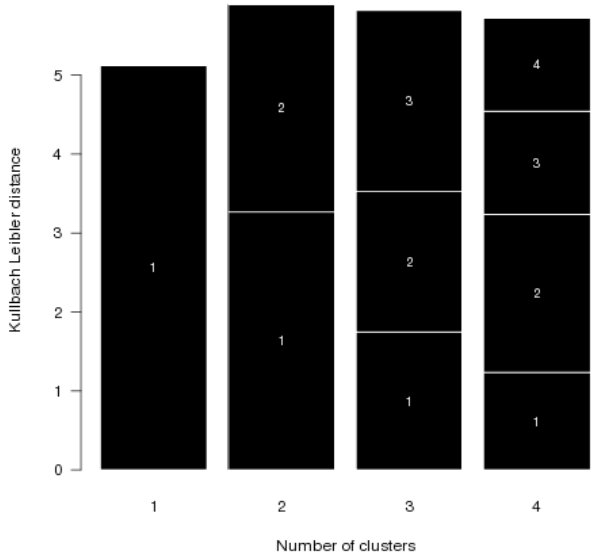



**Figure 4.5. KLD of the alignment/clustering on a mixture of 2 MHC class II molecules depending on the initial number of clusters.** Each block represents a cluster, and its size is proportional to the number of sequences in the cluster. The global optimal solution (highest KLD) was found with 2 clusters, and the corresponding sequence motifs are shown in Figure 4.6.

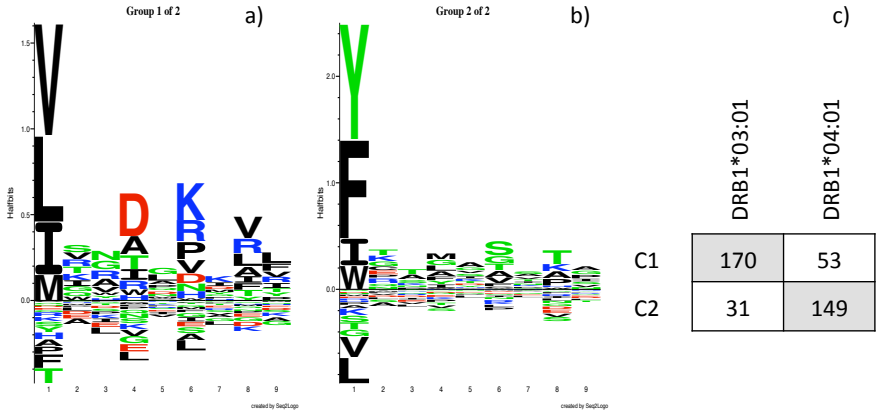



**Figure 4.6. Reconstructed binding motifs from a mixture of binders to 2 MHC class II alleles.** The data set was composed of respectively 202 and 201 binders to the molecules HLA-DRB1*03:01 and HLA-DRB1*04:01. In **a)** and **b)** are shown the logos of the two motifs identified by the algorithm, with the first cluster predominantly composed of DRB1*03:01 binders and the second of DRB1*04:01 binders. **c)** confusion matrix for the two classes of binders, the correlation coefficient is MCC = 0.59.

preferences characterize the binding motif of HLA-DRB1*03:01. The classification of the peptides in the two groups (Figure 4.6c) demonstrates that most peptides are clustered correctly, with an accuracy of 79% and MCC of 0.59.

**Gibbs clustering as a tool to remove noise from data**

In the previous examples, we assumed that all sequences belong to one cluster or another. However, experimental data often contain some level of noise and hence peptides which may not fit in any of the motifs. The Gibbs clustering algorithm allows, by the inclusion of a trash-cluster, a very simple yet highly effective manner to detect such spurious peptides and remove them from the motif identification (see Methods for the implementation).

In Figure 4.7 is shown the effect of the trash cluster on mixtures of 1, 2, 3 and 4 MHC class I alleles polluted with 50 random peptides. We observed that the majority of the random peptides were placed into the trash-cluster, but that an average of about 5 peptides were assigned to one of the clusters. This fits the overall expectation as 1-5% of random natural peptides are estimated to bind to a given MHC class I molecules [5, 115]. Furthermore, most of the random peptides that were inserted into one of the clusters had consistently lower scores than the actual binders (Figure 4.8). The Gibbs clustering algorithm allows obtaining different levels of sensitivity and specificity by varying the threshold to assign a peptide to the trash cluster. Increasing this threshold would remove more noise (peptides with low cluster score) from the data set, but at the same time would increase the number of binders placed in the trash. In the experiments with noisy data (Figure 4.7), a few sequences measured to be binders to a given allele are assigned to the trash (2 for the 1 clusters case, 2 for 2 clusters, 2 for 3 clusters, 4 for 4 clusters). Interestingly, none of these peptides appear to match the binding motifs of the alleles they were measured to bind to. Using the state-of-the-art MHC class I binding prediction method NetMHCcons [116], these

**Table 4.1.** Measured, predicted and re-tested binding affinities (in nM) for peptides assigned to the trash cluster.

| Peptide | HLA | IEDB[a] | Predicted[b] | Validated[c] |
|---|---|---|---|---|
| DHHFTPQII | A*01:01 | 62 | 28485 | 24822 |
| SQTSYQYLI | B*07:02 | 248 | 24349 | 49928 |
| NAFGWENAY | B*07:02 | 350 | 24481 | - |
| TVFKGFVNK | B*27:05 | 235 | 13723 | - |
| ELPIVTPAL | B*40:01 | 314 | 15208 | - |
| ADKNLIKCS | B*40:01 | 316 | 33324 | 76190 |

[a] Binding affinity deposited in the Immune Epitope Database.
[b] Predicted binding affinities using NetMHCcons.
[c] Re-tested binding affinities after detection as outliers.
As a rule of thumb, generally affinity<50nM identifies a strong binder, 50nM<affinity<500nM a weak binder, affinity>500nM non-binders.

| 1 cluster | HLA-B0702 | Random |
|---|---|---|
| g0 | 198 | 5 |
| trash | 2 | 45 |

| 2 clusters | HLA-A0101 | HLA-B0702 | Random |
|---|---|---|---|
| g0 | 199 | 2 | 2 |
| g1 | 0 | 197 | 5 |
| trash | 1 | 1 | 43 |

| 3 clusters | HLA-A0101 | HLA-B0702 | HLA-B4001 | Random |
|---|---|---|---|---|
| g0 | 0 | 197 | 1 | 5 |
| g1 | 1 | 1 | 196 | 8 |
| g2 | 199 | 2 | 1 | 2 |
| trash | 0 | 0 | 2 | 35 |

| 4 clusters | HLA-A0101 | HLA-B0702 | HLA-B2705 | HLA-B4001 | Random |
|---|---|---|---|---|---|
| g0 | 1 | 2 | 195 | 0 | 6 |
| g1 | 1 | 1 | 2 | 196 | 9 |
| g2 | 198 | 2 | 1 | 1 | 2 |
| g3 | 0 | 195 | 1 | 1 | 6 |
| trash | 0 | 0 | 1 | 2 | 27 |

**Figure 4.7. Clustering of MHC class I data polluted with 50 random sequences using a trash-cluster.** In the 4 examples 200 binders for respectively 1, 2, 3 and 4 alleles are mixed together and then reconstructed using the Gibbs clustering. Clustering was repeated from 10 initial random seeds, choosing the solution with highest KLD. Despite the noise nearly all sequences are clustered correctly. Most random sequences are collected by the trash-cluster, although a few obtain scores > 0 to some cluster and are retained in the main groups.

peptides all show extremely low predicted binding affinity to their respective HLA restriction element (>10,000 nM, see Table 4.1). Furthermore, an experimental re-examination of three of these peptides confirmed that they are indeed non-binders to their respective HLA molecule (J. Sidney, personal communication). The method was thus able, whilst grouping distinct specificities into different clusters, to also identify false positives that most likely correspond to erroneous measurements in the experimental assay. Introducing the trash-bin for the MHC class II benchmark also led to an improved clustering performance, removing two outlier peptides, maintaining the optimal solution to consist of two clusters and enhancing the performance to MCC=0.62 (data not shown).

### SH3 domains

The Src Homology 3 domain (SH3 domain) is a small protein interaction module abundantly found in eukaryotes. SH3 domains consist of about 60

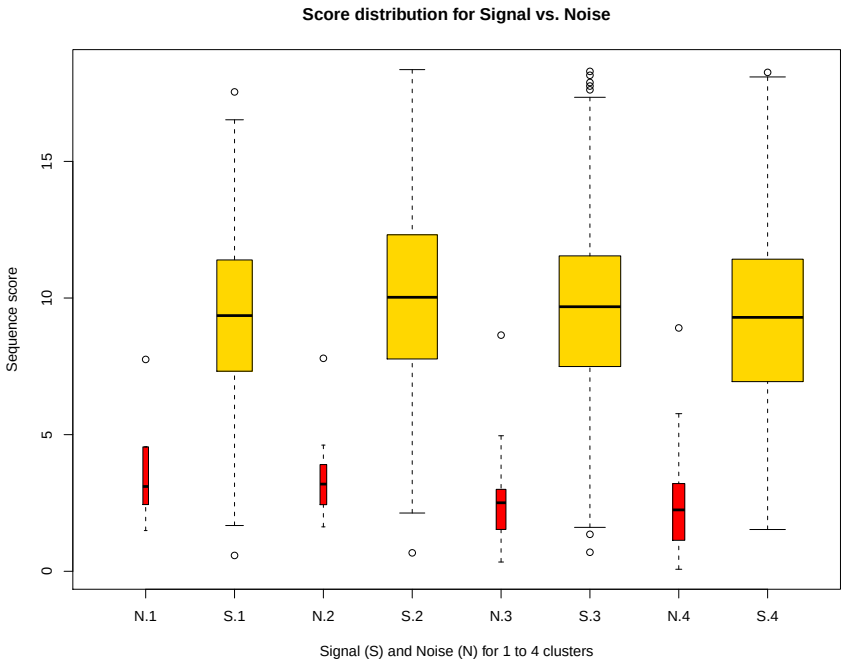



**Figure 4.8. Scores of binders and random sequences in clustering 1 to 4 alleles polluted with random peptides.** The 4 $N.x$ and $S.x$ columns refer to the 4 clustering solutions in Figure 4.7, where $x$ is the number of alleles in the mixture. $N$ (red) columns show the KLD scores for random peptides assigned non-trash clusters, $S$ (yellow) columns depict the scores for binding peptides in non-trash clusters. The KLD scores for binding peptides (yellow) are consistently higher than the scores of random peptides (red). By increasing the threshold for the trash cluster more noise can be removed from the data set, but at the same time an increasing number of binders would be placed in the trash. The width of the boxes in the boxplot is proportional to the square root of the number of sequences in a given group.

amino acids and have been shown to mediate protein-protein interactions by preferentially binding to short proline-rich sequences [117]. The minimal consensus sequence for SH3 domain binding is composed of two prolines located two amino acids apart (PxxP), but it is commonly recognized that there exist two main classes of binders: class I ligands having a general consensus sequence +xΦPxΦP and class II ligands with consensus sequence ΦPxΦPx+ (where + is a positively charged amino acid, usually R, Φ is a hydrophobic amino acid, and x any amino acid) [118]. However, there are a few exceptions to the these predominant motifs, and a number of non-consensus ligands have been identified (reviewed in [119, 120]).

The Gibbs clustering algorithm was run on a large data set of 2,457 peptides binding to the Src SH3 domain. The peptides are 12 amino acids long and unaligned with respect to the binding motif(s) to the SH3 domain. As

the data set may contain non-consensus ligands as well as noise, we performed the alignment/clustering with the addition of a trash-cluster, which collects peptides that do not match any of the main motifs. To ensure the removal of non-consensus sequences that may only partially match the major motifs, the baseline for the trash cluster was set to a relatively high value of 10. The sequence motifs identified by the Gibbs clustering are shown in Figure 4.9. Aligning all sequences into a single cluster (Figure 4.9a) showed the characteristic PxxP pattern, in this case preceded by a leucine (L) and arginine/proline (R/P) three positions back. Clustering the peptides into two groups revealed the two sequence motifs shown in Figure 4.9b. They correspond very well to the two known classes of SH3 domain ligand, one with the PxΦPxRN pattern (class II) and the other with pattern RxΦPxΦP (class I). Dividing further the data set and creating 3 clusters led to the emergence of a new subset of specificity (panel c) besides the two described in the 2-clusters case. Although several exceptions to the two main classes have been

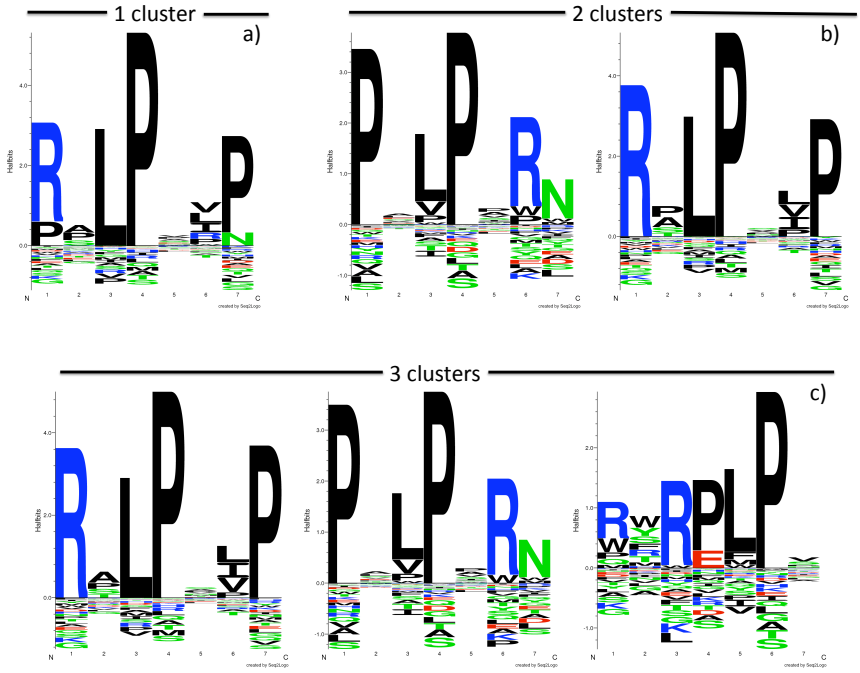



**Figure 4.9. Sequence motifs on SH3 domain binding data clustered in 1 to 3 clusters.** **a)** Sequence motif of the data set aligned in one single cluster. The cluster contains 2,360 peptides, 97 peptides were discarded to the trash cluster. **b)** Sequence motifs for SH3 domain data split in two clusters. The two groups are in strong agreement with the canonical class I (right, 1,892 peptides) and class II (left, 498 peptides) types of SH3 domain ligands. 67 peptides were moved to the trash cluster. **c)** Sequence motifs when the data is split in 3 clusters. The clusters have sizes of respectively 1,606, 490 and 305 peptides, with 56 peptides discarded to the trash cluster.

discovered [120], this RxRPΦP pattern has not, to the best of our knowledge, been described before. Splitting the data set further to more than 3 clusters does not show new specificities besides those described here.

The two motifs displayed in Figure 4.9b agree strongly with the results obtained in a previous study [107], where the MUSI method was applied to the same phage display data set. The Gibbs clustering method however has the strong advantage, compared to MUSI, in that the data do not need to be aligned prior to clustering. Instead, in the Gibbs clustering method alignment and clustering are performed simultaneously. In the specific case of SH3 domain binding, where both motifs share a strong common PxxP pattern, a pre-alignment strategy to a common motif like the one implemented in MUSI can be successful. However, in the general case, the different motifs will be weak and will not share a common pattern. On such data, it becomes difficult if not impossible to accurately identify the binding core within the peptide data set using alignment techniques [33]. For instance, by applying the MUSI method on the MHC class II data set from above, we found the solution with two motifs being suboptimal compared to a solution with a single motif. Forcing MUSI to generate two clusters, the overall performance was MCC=0.21, which is significantly lower than what was obtained using the Gibbs clustering method ($p<0.01$, bootstrap test).

**Sub-specificities of MHC class I molecules**

Peptide binding to MHC molecules is one of the most selective steps in determining MHC class I-restricted CTL responses. The strength of this interaction is commonly measured in terms of binding affinity between peptide and MHC complex. However, not all peptides with high affinity are immunogenic, indicating the presence of other factors determining an effective response [121]. Some studies have suggested that the stability of the MHC-peptide complex is a major player in determining immunogenicity [122, 123, 112].

By means of the Gibbs clustering algorithm, we investigated if there exist sub-specificities for MHC class I binding, and whether these sub-specificities correlated with different levels of affinity and/or stability. For this purpose, we used a data set recently published by Harndahl et al. [112] consisting of 650 peptides binding with affinity stronger than 500nM to HLA-A*02:01 for which also the peptide stability had been measured. We applied the Gibbs clustering algorithm to split the data set in two clusters using default parameters and investigated the properties of the sequences in the two groups.

The sequence motifs for the resulting clusters are shown in Figure 4.10. The first cluster (G1), composed of 441 sequences, was highly specific in terms of amino acid preference, with [LIM] at P2 and [VLI] at P9. The contribution from other positions is secondary. The second cluster (G2) is more promiscuous at both anchor positions P2 and P9, especially at P2 where several amino acids other than L, I and M are allowed. The peptides in the two groups had a median binding affinity of 6 nM and 9 nM, for G1 and G2

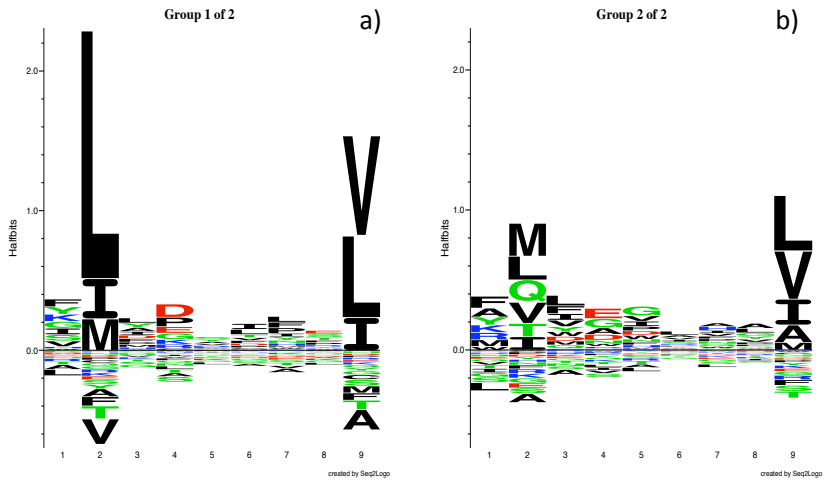



**Figure 4.10. Sub-motifs of HLA-A*02:01 binding specificity.** The peptides in the two clusters have similar affinity but differ significantly in stability. The sequence logo in the left panel is composed mainly of stable peptides ($T_h \approx 5.7$ hours) whereas peptides in the second group have lower stability ($T_h \approx 2.1$ hours).

respectively. This difference is not significant (p=0.095, Wilcoxon rank-sum test). In contrast, we observed that peptides in G1 have a significantly higher stability compared to G2 ($p<10^{-6}$, Wilcoxon rank-sum test): the median half-life of the MHC-peptide complex in G1 is $T_h \approx 5.7$ hours, whereas in G2 it is only $T_h \approx 2.1$ hours.

From these results, we can conclude that the method identified subtle differences between the binders to HLA-A*02:01 that appear to differentiate stable binders from unstable binders. In particular, as previously noted peptide-HLA-A*02:01 complexes appear to be destabilized by a suboptimal amino acid in just one of the two anchor positions and in particular position P2 [112].

### 4.1.5 Discussion

We proposed an efficient algorithm to identify multiple specificities in peptide data sets. The applications of the method are numerous, ranging from the deconvolution of poly-specificities contained in a data set, to the analysis of sub-specificities within a known binding motif. The algorithm aims at identifying the solution (the set of clusters and corresponding alignments) that optimally fits the peptide data set. The optimal solution is automatically selected and the identified binding motifs are visualized as individual sequence logos. Using a panel of benchmark data sets, we have demonstrated the power of the Gibbs clustering method in deconvoluting poly-specificities contained both in pre-aligned and unaligned peptide data sets covering the MHC class I, MHC class II and human SH3 domain systems.

Gibbs sampling is a powerful approach to explore large spaces of possible solutions. In the case of amino acid sequences, there are immense possible ways of aligning and clustering them as soon as the number of sequences becomes bigger than a handful. The probabilistic nature of Gibbs sampling allows efficient sampling of the search space and convergence towards a state of high fitness of the system. Compared to other motif identification methods, Gibbs clustering is unique in that it incorporates alignment and clustering in a set of alternative sampling moves, allowing for simultaneous identification of clusters and optimal sequence alignment. This property makes the method capable of identifying subtle and relatively weak binding motifs (as demonstrated for the case of MHC class II binding motifs) but it comes at the price of computational speed. Analyzing the 400 peptides in the MHC class II binding data set takes a little more than 5 minutes using Gibbs clustering. This running time is reduced to 15 seconds using the MUSI algorithm [107] yet at the cost a of dramatic and significant drop in accuracy.

In a general situation, it is not known a priori how many motifs are contained in a data set. When presented with a set of experimental data, the investigator ideally wants a definitive answer to the question: How many motifs are contained in my data? Unfortunately the answer is not unambiguous, not so much for a fault of mathematical and computational methods, rather for the ambiguity of the question. The answer depends on the level of resolution that is expected for the particular problem at hand. If the goal is a rough classification of sequences based on global differences then the resulting number of clusters will be small. Conversely, more partitions would be produced if we were searching for subtler distinguishing sequence characteristics. The true number of clusters is therefore not an objective answer but depends on the kind of biological question that is being asked. In the Gibbs clustering algorithm, we introduce a parameter $\lambda$ that aims to modulate the degree of resolution required by the user. High $\lambda$ penalizes overlap between clusters, and tends to create coarser clusters, whereas low $\lambda$ results in smaller and specialized clusters. For example, we showed that for a certain value of $\lambda$, we could accurately identify the number of MHC class I molecules contained in a data set of mixed specificities. In another example, we split one of these very same specificities into sub-motifs, and looked for subtle differences in a rather homogenous population of peptides. And these are not the extremes: one could conceive partitioning the data further into more specialized sub-populations, as well as obtaining a coarser picture of similarities between alleles. The same data may have different levels of resolution depending on the aim of the analysis, and the investigator should keep this in mind when using a classification method like the one presented here. The Gibbs clustering method in its current form is limited to handle situations where motifs are of uniform length. Likewise, the method can only handle amino acid input data. The reason for this limitation is that most of its unique features like pseudo-count estimates from Blosum substitution matrices and sequence weighting of are specific for amino acid data.

In conclusion, we believe the Gibbs clustering method to be both a highly accurate and very user-friendly tool that will allow researchers to interpret

peptide data sets in terms of receptor specificities in a highly intuitive manner. Therefore, we expect it to become an important tool as large-scale peptide chip technologies grow to be a cost-effective and accessible platform for investigation of protein-ligand interactions. The method is highly customizable and publicly available as an online web-server at `http://www.cbs.dtu.dk/services/GibbsCluster`.

**Acknowledgements**

We thank John Sidney (La Jolla Institute for Allergy and Immunology, California, USA) for the binding affinity validation of the predicted MHC class I outliers.

*Funding:* this work was supported by the European Union 7th Framework Program FP7/2007--2013 [grant number 222773].

# Chapter 5

# String kernels for binding affinity prediction

In a recent publication, Smale and coworkers [124] proposed a kernel function to measure distances between amino acid sequences, and applied the method to the prediction of peptide-MHC class II binding affinity. The method was shown to achieve performances comparable to *NNAlign* (described in chapter 2) on a benchmark of HLA-DR molecules. We implemented the kernel method according to the manuscript to investigate its strengths and weaknesses compared to other available methods. After briefly introducing the kernel functions and the RLS algorithm, we show its application for the prediction of peptide binding affinity to HLA class I molecules, and particularly how the method can benefit by training on peptides of multiple lengths.

## 5.1 Kernel functions

The kernel function for peptide-peptide distances is constructed by defining three functions of increasing complexity:

- $K^1$, defining similarities between pairs of amino acids, based on BLOSUM scores;
- $K^2$, between amino acid stretches of equal length, based on kernel $K^1$;
- $K^3$, between pairs of amino acid chains of any length, based on $K^2$.

The construction of the kernels starts from the BLOSUM substitution frequencies between amino acids [22]. The BLOSUM odds matrix $B$ can be cal-

culated for each pair of amino acids using:

$$B_{xy} = \frac{p_{xy}}{q_x q_y} \tag{5.1}$$

where $p_{xy}$ is the substitution frequency between amino acids $x$ and $y$, and $q_a$ is the background frequency of amino acid $a$ in naturally occurring proteins. The BLOSUM matrices commonly used for sequence alignment are obtained taking the logarithm of the odds matrix $B$, then rounding the score to the closest integer.

The basic kernel $K^1$ between a pair of amino acids $(x, y)$ is derived directly from the BLOSUM62 odds matrix $B$ using:

$$K^1(x, y) = (B_{xy})^\beta \tag{5.2}$$

where $\beta$ is the Hadamard power (power of single entries) of the matrix. $\beta$ is one of the parameters of the method to be optimized.

The kernel $K^2$ between a pair of k-mers $(u, v)$ follows by multiplication of the $K^1$ on all amino acid pairs composing the two k-mers:

$$K^2_k(u, v) = \prod_{i=1}^{k} K^1(u_i, v_i) \tag{5.3}$$

where $u_i$ denotes the amino acid found at position $i$ in the k-mer $u$.

Finally, the similarity between a pair of amino acid sequences $(f, g)$ is calculated by combining the $K^2$ between any pair of substrings $u$ and $v$ of respectively $f$ and $g$:

$$K^3(f, g) = \sum_{\substack{u \subset f, v \subset g \\ |u|=|v|=k \\ \text{all } k=1,2,\ldots}} K^2_k(u, v) \tag{5.4}$$

A normalized version of the $K^3$ kernel, which returns a value of 1 for any pair of identical sequences, can be obtained using:

$$\hat{K}^3(f, g) = \frac{K^3(f, g)}{\sqrt{K^3(f, f) K^3(g, g)}} \tag{5.5}$$

**Examples** Calculation of $K^1$ between pairs of amino acids is trivial: it corresponds to the BLOSUM odds score for the two amino acids, to the power of $\beta$. For example, assuming $\beta = 0.1$, one obtains $K^1(R, G) = 0.922$ and $K^1(R, K) = 1.076$, indicating that substitutions between R and K are more frequent than between R and G.

Pairwise $K^2_k$ entries are calculated by multiplication of $K^1$ on all the amino acid pairs composing the k-mers. For example, for the pair of 3-mers TQA and WQP, we obtain:

$$K^2_3(\text{TQA}, \text{WQP}) = K^1(T, W) \times K^1(Q, Q) \times K^1(A, P) = 1.081$$

On full sequence level, $K^3$ combines in a sum the $K_k^2$ on all possible k-mer pairs contained in the two sequences to be compared. Suppose we want to calculate the similarity based on $K^3$ between the short peptides FFTQA and WQPE. First we have to calculate $K_1^2$ an all 1-mer pairs {(F,W), (F,Q), (F,P), ...}, then $K_2^2$ on all 2-mers {(FF,WQ), (FF,QP), ...}, then $K_3^2$ on all 3-mers {(FFT,WQP), (FFT,QPE), ...}, and finally $K_4^2$ on the two possible 4-mer pairs {(FFTQ,WQPE), (FTQA,WQPE)}. Summing all the contributions one obtains $K^3(\text{FFTQA}, \text{WQPE}) = 37.826$. Normalizing this score with equation 5.5 one obtains:

$$\hat{K}^3(\text{FFTQA}, \text{WQPE}) = \frac{K^3(\text{FFTQA}, \text{WQPE})}{\sqrt{K^3(\text{FFTQA}, \text{FFTQA}) K^3(\text{WQPE}, \text{WQPE})}} = 0.813$$

For a more thorough and detailed description of the kernel and its properties refer to the original publication by Smale and coworkers [124].

## 5.2 Regularized Least Squares (RLS) learning

Given a data set of $n$ peptide sequences, the kernel function $\hat{K}^3$ allows calculating pairwise distances between all pairs of peptides. We can define a function $f$ to combine the elements of the kernel as a system of linear equations:

$$f(\cdot) = \sum_{i=1}^{n} w_i \hat{K}^3(x_i, \cdot) \tag{5.6}$$

where $x$ is the vector of $n$ peptide sequences in the training set, and $w_i$ the $i^{th}$ coefficient (or weight) in the linear equations of the system. The RLS algorithm aims at finding the optimal set of weights $w$ that minimize the error between the outcome of the $f$ function and the desired output $y$ for each data point (the vector $y$ may be for example the set of measured peptide-MHC binding affinities):

$$f = \arg\min \left( \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|^2 \right) \tag{5.7}$$

where $\lambda$ is a regularization parameter that controls the absolute value of the coefficients $w$ and by that avoids over fitting on the training data.

By combining equation 5.6 and 5.7 and imposing $\frac{\partial f}{\partial w} = 0$, one obtains:

$$(\hat{K}^3 + n\lambda I)w = Gw = y \tag{5.8}$$

where $G$ is simply the kernel on peptides $\hat{K}^3$ with a constant additive term $n\lambda$ on the diagonal. The optimization problem is now expressed in terms of the system of linear equations $Gw = y$ which can be solved to find the set of weights $w$ as follows.

The properties of $\hat{K}^3$ (and consequently $G$) guarantee that $G$ can be decomposed into a product of a lower and an upper triangular matrix $G = LL^T$

where $L$ is a lower triangular matrix and $L^T$ its transposed version. We implemented the decomposition using the Cholesky Banachiewicz algorithm, with a computational complexity of $O(n^3)$. After calculating $L$, we can solve $Lb = y$ for the vector $b$, and finally obtain the set of coefficients by solving $L^T w = b$ for $w$. These solutions are trivial to calculate, as on the triangular matrices we can iteratively descend the diagonal and calculate one coefficient of $b$ (and then $w$) at a time.

The trained model based on the kernel $\hat{K}^3$ and its associated weights $w$ can be used to predict any new data point $x^*$ using equation 5.6, or explicitly:

$$f(x^*) = \sum_{i=1}^{n} w_i \hat{K}^3(x_i, x^*) \tag{5.9}$$

## 5.3 Predicting MHC class I binding affinity

After confirming the correctness of our algorithm implementation by reproducing the results of the original publication [124] for MHC class II binding prediction (data not shown), we applied the method to a new problem: prediction of MHC class I binding affinity.

As discussed in section 1.1.1, MHC class I molecules can accommodate in their binding groove peptide of lengths between 8 and 11 amino acids. Most prediction methods, including NN-align [36] and the stabilized matrix method (SMM) [125] require that a separate model is trained for each different peptide length. We found particularly interesting that the Kernel method does not suffer from this limitation, and can be trained on multiple peptide lengths at the same time. Here, we investigate whether such property can be

**Table 5.1.** The human MHC class I 12 alleles used in the benchmark, and number of peptides with measured binding affinity for each allele.

| Allele | #9 | #9bind | #10 | #10bind |
|---|---|---|---|---|
| HLA-A*01:01 | 1157 | 103 | 56 | 18 |
| HLA-A*02:01 | 3089 | 1181 | 1316 | 526 |
| HLA-A*02:03 | 1443 | 639 | 1055 | 470 |
| HLA-A*03:01 | 2094 | 517 | 1082 | 410 |
| HLA-A*11:01 | 1985 | 693 | 1093 | 471 |
| HLA-A*24:02 | 197 | 99 | 78 | 28 |
| HLA-A*29:02 | 160 | 68 | 55 | 8 |
| HLA-A*31:01 | 1869 | 427 | 1057 | 315 |
| HLA-A*68:01 | 1141 | 498 | 1055 | 440 |
| HLA-B*07:02 | 1262 | 208 | 205 | 78 |
| HLA-B*35:01 | 736 | 211 | 177 | 46 |
| HLA-B*53:01 | 254 | 106 | 177 | 47 |

#9 and #10 refer to the number of 9-mer and 10-mer peptides with measured binding affinity, #9bind and #10bind to the number of peptides with binding affinity < 500nM.

an advantage in terms of predictive performance, especially in cases where training data for a particular peptide length are scarce.

For this benchmark, we used 12 HLA class I alleles from the data set published by Peters et al. [57], which contains binding data for 9-mer and 10-mer peptides. The size of the data sets and relative number of binding peptides vary greatly among the different alleles, and are listed in Table 5.1. We train and evaluate predictive performance using a 5-fold cross-validation, preserving the same partitions as in Peters et al. [57].

First, we estimated an optimal value for the regularization parameter $\lambda$. On a cross-validation experiment on the 9-mer training sets, $\lambda$ was found to give highest performance around $\lambda = e^{-10}$. We fixed the value of the parameter $\beta$ (see equation 5.2) to $\beta = 0.11387$, found to be optimal for class II prediction in the paper by Smale et al. [124].

### 5.3.1 Enriching 9-mer data with 10-mers

We investigated whether training a method on 9-mer and 10-mer peptides leads to improved predictions for 9-mer peptides. For this purpose, we enriched the 9-mer training sets (still preserving the partitions defined in Peters et al. [57]) with 10-mer sequences. In order to ensure that the 10-mers have informative value and are not merely an elongation of existing 9-mers, we excluded 10-mers that are identical to some 9-mer after removing any of the amino acids between P3 and P9 in the 10-mers (for most alleles the anchors are at P2 and P9 for 9-mers, and P2 and P10 for 10-mers).

**Table 5.2.** Prediction performance of the kernel method for prediction of binding affinity of 9-mer peptides

| Allele | AUC | | RMSE | |
|---|---|---|---|---|
| | K9 | K9/10 | K9 | K9/10 |
| HLA-A*01:01 | 0.9659 | **0.9668** | **0.1173** | 0.1177 |
| HLA-A*02:01 | 0.9473 | **0.9479** | 0.1720 | **0.1711** |
| HLA-A*02:03 | 0.9111 | **0.9143** | 0.1848 | **0.1805** |
| HLA-A*03:01 | 0.9191 | **0.9218** | 0.1678 | **0.1649** |
| HLA-A*11:01 | 0.9301 | **0.9372** | 0.1748 | **0.1702** |
| HLA-A*24:02 | 0.8048 | **0.8352** | 0.2096 | **0.2016** |
| HLA-A*29:02 | 0.9377 | **0.9420** | 0.1754 | **0.1725** |
| HLA-A*31:01 | 0.9249 | **0.9262** | 0.1650 | **0.1616** |
| HLA-A*68:01 | 0.8674 | **0.8752** | 0.1988 | **0.1928** |
| HLA-B*07:02 | 0.9624 | **0.9653** | 0.1354 | **0.1344** |
| HLA-B*35:01 | 0.8786 | **0.8913** | 0.1889 | **0.1843** |
| HLA-B*53:01 | 0.8646 | **0.8764** | 0.2384 | **0.2297** |
| AVG | 0.9095 | **0.9166** | 0.1773 | **0.1734** |

The column K9 refers to the kernel method trained on 9-mer peptides, K9/10 to the method trained on 9-mers enriched with 10-mer peptides. In bold is the best performing method for a given allele. Data set sizes for each allele are listed in Table 5.1

The addition of 10-mer peptides lead to improved performance on all alleles based on AUC, and 11/12 alleles based on RMSE (see Table 5.2). The increase in performance is significant ($p<0.01$, binomial test) and appeared more pronounced on alleles with few data points, such as A*24:02, B*35:01 and B*53:01. The average performance on the 12 alleles in AUC is 0.917, comparable to SMM [125] (AUC=0.915) and ANN [25] (AUC=0.922) on the same 9-mer data sets.

### 5.3.2 Enriching 10-mer data with 9-mers

Compared to 9 amino acid long peptides, there are relatively few data for 10-mers (see Table 5.1). For example, in the class I benchmark data set there were only 56 10-mer peptides with measured binding affinities for HLA-A*01:01, compared to over a thousand data points for 9-mer peptides. We investigated whether the large 9-mer data sets could be used to improve the binding predictions for 10-mers peptides. The enriched data sets were prepared in a similar manner as described above for 9-mers, only including enriching peptides that differ from 10-mers after removing any of the amino acids between P3 and P9 in the 10-mers.

The benchmark calculations, shown in Table 5.3, demonstrate that prediction of binding affinity for 10-mers benefited greatly from inclusion of 9-mers in the training set. Training a kernel method on 10-mers enriched with 9-mers was optimal on 10 out of 12 alleles (11/12 based on RMSE), and in the other cases the best method was trained only on 9-mers. In no instance

**Table 5.3.** Prediction performance of the kernel method for prediction of binding affinity of 10-mer peptides

| Allele | AUC | | | RMSE | | |
|---|---|---|---|---|---|---|
| | K9 | K10 | K9/10 | K9 | K10 | K9/10 |
| HLA-A*01:01 | 0.9518 | 0.9298 | **0.9678** | 0.2243 | 0.1857 | **0.1744** |
| HLA-A*02:01 | 0.9056 | 0.9003 | **0.9195** | 0.1863 | 0.1824 | **0.1726** |
| HLA-A*02:03 | 0.8363 | 0.8375 | **0.8555** | 0.1932 | 0.1860 | **0.1765** |
| HLA-A*03:01 | 0.8347 | 0.8582 | **0.8673** | 0.2048 | 0.1894 | **0.1830** |
| HLA-A*11:01 | 0.8675 | 0.8896 | **0.9001** | 0.2021 | 0.1900 | **0.1820** |
| HLA-A*24:02 | 0.8543 | 0.8786 | **0.8986** | 0.2016 | 0.1983 | **0.1787** |
| HLA-A*29:02 | **0.8245** | 0.5213 | 0.7926 | **0.2100** | 0.2425 | 0.2155 |
| HLA-A*31:01 | 0.8554 | 0.8654 | **0.8787** | 0.1774 | 0.1730 | **0.1663** |
| HLA-A*68:01 | 0.8285 | 0.8527 | **0.8682** | 0.2044 | 0.1901 | **0.1808** |
| HLA-B*07:02 | 0.8200 | 0.8068 | **0.8426** | 0.2233 | 0.2212 | **0.2091** |
| HLA-B*35:01 | **0.8609** | 0.8211 | 0.8604 | 0.2035 | 0.1946 | **0.1797** |
| HLA-B*53:01 | 0.7779 | 0.7131 | **0.7781** | 0.2200 | 0.2223 | **0.2085** |
| AVG | 0.8514 | 0.8229 | **0.8691** | 0.2042 | 0.1980 | **0.1856** |

The column K9 refers to the kernel method trained on 9-mer peptides, K10 trained on 10-mers, K9/10 to the method trained on 10-mers enriched with 9-mer peptides. In bold is the best performing method for a given allele.

training uniquely on 10-mers was optimal for the prediction of 10-mers. It is interesting to note that a rather accurate method, with average AUC = 0.8514, could be obtained by only training on 9-mers. In other words, we could predict binding of 10-mer peptides with a method that had never seen a 10-mer peptide.

### 5.3.3 SMM with 9-mer approximation

An attempt to deal with the scarcity (or lack) of data for certain peptide lengths was proposed by Lundegaard et al. for the NetMHC method [126]. Here, the authors trained the method on 9-mers and extrapolated predictions for other lengths by artificially shortening or elongating the evaluation sequences. In particular, it was shown that the 9-mer approximation performed better than ANNs trained on 10-mers in 12 out of 16 alleles.

The matrix-based SMM method [125], which as mentioned in section 5.3.1 reaches comparable performance to the Kernel method on MHC class I data, can be readily adapted to employ the 9-mer approximation for binding affinity prediction of 10-mers. After training SMM on 9-mers from the above data set of 12 MHC class I molecules, we evaluated its performance in predicting 10-mers compared to the Kernel method.

The first observation is that the 9-mer approximation gives on average better performance in terms of AUC than the method trained directly on 10-mer data (Table 5.4). Only on three alleles the SMM trained on 10-mers is superior to the approximation, in agreement with the findings of Lundegaard

**Table 5.4.** Comparison between Kernel method and SMM method for the prediction of 10-mer peptides.

| Allele | AUC | | | RMSE | | |
|---|---|---|---|---|---|---|
| | K9/10 | SMM-9 | SMM-10 | K9/10 | SMM-9 | SMM-10 |
| HLA-A*01:01 | 0.9678 | **1.0000** | 0.9488 | **0.1744** | 0.1964 | 0.1805 |
| HLA-A*02:01 | **0.9195** | 0.9060 | 0.9035 | **0.1726** | 0.1852 | 0.1858 |
| HLA-A*02:03 | **0.8555** | 0.8164 | 0.8157 | **0.1765** | 0.1952 | 0.1912 |
| HLA-A*03:01 | **0.8673** | 0.8497 | 0.8635 | **0.1830** | 0.1939 | 0.1845 |
| HLA-A*11:01 | **0.9001** | 0.8868 | 0.8987 | **0.1820** | 0.1925 | 0.1854 |
| HLA-A*24:02 | **0.8986** | 0.8679 | 0.8043 | **0.1787** | 0.2204 | 0.2485 |
| HLA-A*29:02 | **0.7926** | 0.7766 | 0.6250 | **0.2155** | 0.2608 | 0.2313 |
| HLA-A*31:01 | **0.8787** | 0.8562 | 0.8642 | **0.1663** | 0.1744 | 0.1729 |
| HLA-A*68:01 | 0.8682 | **0.8713** | 0.8589 | **0.1808** | 0.1932 | 0.1910 |
| HLA-B*07:02 | 0.8426 | **0.8727** | 0.8404 | **0.2091** | 0.2212 | 0.2202 |
| HLA-B*35:01 | 0.8604 | **0.8951** | 0.8511 | **0.1797** | 0.2049 | 0.1877 |
| HLA-B*53:01 | 0.7781 | **0.7982** | 0.7719 | 0.2085 | 0.2157 | **0.2020** |
| AVG | **0.8691** | 0.8664 | 0.8372 | **0.1856** | 0.2045 | 0.1984 |

The column K9/10 refers to the kernel method trained on 10-mers enriched with 9-mer peptides, SMM-9 to the SMM method trained on 9-mers and using the 9-mer approximation, SMM-10 to the SMM method trained on 10-mer peptides.

et al. [126], where 2 of these 3 alleles (A*03:01 and A*31:01) also appeared not to benefit from the 9-mer approximation. Secondly, if performance is judged upon AUC values, the 9-mer approximation shows to be very powerful and reaches performances comparable to the Kernel method trained on both 9-mers and 10-mers. The kernel prevails on 7 out of 12 alleles, while the SMM trained on 9-mers is best on 5 alleles. However, when considering the RMSE between target and predicted values, the Kernel method performs significantly better than the SMM ($p<0.01$, binomial test).

### 5.3.4 Combining Kernel and SMM in a consensus method

A consensus method combines predictions between two or more methods on the same data with the aim of boosting predictive performance. Systematic benchmarks have shown that, for MHC class I and class II binding prediction, consensus methods are generally superior to any of the individual methods included in the benchmark [127, 128]. In particular, NetMHCcons [116] is a method that defines the optimal combination of prediction methods for MHC class I binding in several species, and combines them in a consensus predictor with enhanced performance. The general idea behind consensus strategies is that distinct methods, especially if they exploit different aspects of the data, distribute their errors in different areas of the evaluation space. A polled opinion from two or more methods may be able to correct, or at least reduce, such errors.

Observing that the Kernel and SMM methods appear to have comparable performance, with each of them prevailing on different alleles, we investigated whether a combination of the two methods could lead to a consensus with higher predictive performance. SMM was trained on 9-mer data (SMM-9), whereas the Kernel method was trained on both 9-mer and 10-mer peptides (K9/10). The two methods were then applied to predict, in a 5-fold cross-validation, the binding affinity of 9-mer and 10-mer peptides (in the case of SMM, using the 9-mer approximation described above). The consensus was obtained simply by averaging the prediction values of the two methods for each evaluation sequence.

Prediction performances for the individual methods and their consensus are shown in Table 5.5. In terms of AUC, the consensus method is significantly better than any of the two individual methods on 19 out of 24 instances ($p < 0.0005$, pairwise binomial tests). However there appears to be no significant improvement ($p > 0.5$, binomial test) in terms of RMSE from K9/10 to the consensus method (K9/10 is best in 10 cases, Cons is best in 13 cases, they have same RMSE in 1 case). This is likely due to the rather mediocre RMSE of SMM-9, although the same method achieves AUC values comparable to K9/10. Understandably, if one of the methods in the consensus is consistently worse than the other, then including such method will not result beneficial, or be even deleterious, to the overall performance of the consensus method.

**Table 5.5.** Performance of Kernel and SMM methods for prediction of 9-mer and 10-mer peptides, and consensus of the two methods.

| Allele | L | AUC | | | RMSE | | |
|---|---|---|---|---|---|---|---|
| | | SMM-9 | K9/10 | Cons | SMM-9 | K9/10 | Cons |
| HLA-A*01:01 | 9 | 0.9685 | 0.9668 | **0.9734** | 0.1272 | **0.1177** | 0.1195 |
| HLA-A*02:01 | 9 | 0.9459 | 0.9479 | **0.9503** | 0.1817 | **0.1711** | 0.1733 |
| HLA-A*02:03 | 9 | 0.9133 | 0.9143 | **0.9233** | 0.1843 | 0.1805 | **0.1759** |
| HLA-A*03:01 | 9 | 0.9270 | 0.9218 | **0.9292** | 0.1678 | 0.1649 | **0.1634** |
| HLA-A*11:01 | 9 | 0.9380 | 0.9372 | **0.9415** | 0.1743 | 0.1702 | **0.1689** |
| HLA-A*24:02 | 9 | 0.7748 | **0.8352** | 0.8209 | 0.2307 | **0.2016** | 0.2066 |
| HLA-A*29:02 | 9 | 0.9197 | 0.9420 | **0.9421** | 0.1909 | 0.1725 | **0.1707** |
| HLA-A*31:01 | 9 | 0.9282 | 0.9262 | **0.9318** | 0.1654 | 0.1616 | **0.1600** |
| HLA-A*68:01 | 9 | 0.8818 | 0.8752 | **0.8871** | 0.1907 | 0.1928 | **0.1862** |
| HLA-B*07:02 | 9 | 0.9595 | 0.9653 | **0.9655** | 0.1419 | **0.1344** | 0.1346 |
| HLA-B*35:01 | 9 | 0.8800 | 0.8913 | **0.8990** | 0.1946 | 0.1843 | **0.1836** |
| HLA-B*53:01 | 9 | 0.8517 | 0.8764 | **0.8775** | 0.2428 | 0.2297 | **0.2261** |
| HLA-A*01:01 | 10 | **1.0000** | 0.9678 | 0.9898 | 0.1964 | **0.1744** | 0.1801 |
| HLA-A*02:01 | 10 | 0.9060 | 0.9195 | **0.9210** | 0.1852 | **0.1726** | 0.1740 |
| HLA-A*02:03 | 10 | 0.8164 | **0.8555** | 0.8483 | 0.1952 | **0.1765** | 0.1767 |
| HLA-A*03:01 | 10 | 0.8497 | 0.8673 | **0.8738** | 0.1939 | **0.1830** | **0.1830** |
| HLA-A*11:01 | 10 | 0.8868 | 0.9001 | **0.9061** | 0.1925 | 0.1820 | **0.1816** |
| HLA-A*24:02 | 10 | 0.8679 | 0.8986 | **0.9021** | 0.2204 | **0.1787** | 0.1870 |
| HLA-A*29:02 | 10 | 0.7766 | 0.7926 | **0.8085** | 0.2608 | **0.2155** | 0.2183 |
| HLA-A*31:01 | 10 | 0.8562 | 0.8787 | **0.8801** | 0.1744 | 0.1663 | **0.1651** |
| HLA-A*68:01 | 10 | 0.8713 | 0.8682 | **0.8840** | 0.1932 | 0.1808 | **0.1789** |
| HLA-B*07:02 | 10 | **0.8727** | 0.8426 | 0.8681 | 0.2212 | **0.2091** | 0.2095 |
| HLA-B*35:01 | 10 | **0.8951** | 0.8604 | 0.8900 | 0.2049 | 0.1797 | **0.1785** |
| HLA-B*53:01 | 10 | 0.7982 | 0.7781 | **0.8097** | 0.2157 | 0.2085 | **0.1966** |
| AVG | 9 | 0.9074 | 0.9166 | **0.9201** | 0.1827 | 0.1734 | **0.1724** |
| AVG | 10 | 0.8664 | 0.8691 | **0.8818** | 0.2045 | **0.1856** | 0.1858 |

The column L represents the length of the peptides being evaluated. SMM-9 refers to SMM method trained on 9-mers, K9/10 to the kernel method trained on 10-mers enriched with 9-mer peptides, and Cons is the consensus prediction of the two methods combined. Predictions for SMM-9 on 10-mer peptides were obtained using the 9-mer approximation described in section 5.3.3.

## 5.4 Discussion

Kernel methods have gained popularity mainly because they are at the base of Support Vector Machines (SVM), mapping data from input space to high-dimensional feature spaces. If the mapping defined by the kernel is non-linear, then the separating hyperplane of the SVM can also be made non-linear with respect to the original input space [129].

However, kernels can also be used directly as functions to measure distances between strings, such as amino acid strings. For example, string kernels have been applied to MHC binding prediction [130], detection of pro-

tein homology [131] and fold recognition [132]. The kernel implementation used in this chapter was originally applied to the prediction of MHC class II binding and to define similarities between a large set of DRB molecules [124]. Combining the two aspects of peptide-peptide and MHC-MHC similarities, the authors also developed a pan-specific method enabling binding predictions for any peptide-MHC class II combination. Here, we applied the kernel functions, coupled with RLS learning, to the prediction of MHC class I binding.

The kernel method showed performance comparable to state-of-the-art methods for MHC class I prediction. Notably, improved performance could be obtained by combining peptides of different length in the same training set. Such property is given by the nature of the $K^3$ kernel, which combines similarities between shorter stretches of amino acids and by that implicitly allows "gaps" in the pairwise alignments. The positive contribution of several pairs of sub-sequences was observed previously in the *Local Alignment Kernel* [130], where incorporating sub-optimal alignments into similarity scores contributed to improved predictions. Furthermore, the method is essentially assumption-free, not being based on any specific knowledge of the binding pocket of the molecules or of the binding core length. These properties make it easily adaptable to various biological problems based on sequence similarity.

Since the method does not produce a multiple sequence alignment in the proper sense, it cannot directly provide a sequence motif. This is a serious limitation, if the alignment and/or binding motif were desirable. Another possible restriction may derive from the computational complexity of the algorithm: the decomposition of the $n \times n$ kernel matrix is a function of $O(n^3)$ on $n$ training sequences, which may become prohibitive with more than a few thousand data points. Finally, specialized substitution matrices may be employed depending on the problem at hand $-$ the current kernels are based on BLOSUM62 scores, but matrices specific for certain systems exist, such as the PMBEC matrix for peptide-MHC binding [133].

At any rate, our results confirmed the power and usefulness of the method beyond the cases presented in the original publication, attesting it as a flexible and promising approach for sequence alignment problems. As string kernels exploit different aspects of the data compared to alignment-based methods such as *NNAlign*, we can anticipate improved predictions on quantitative peptide data by combining these approaches in consensus methods.

# Chapter 6

# Epitope prediction from peptide pool-based ELISPOT and ICS assays

T-CELLS play an essential role in antigen-specific recognition. T-cells rearrange the genes encoding the T-cell receptor (TCR) to recognize with very high specificity antigens in form of peptide-MHC complexes. The diversity generated by somatic rearrangement of TCR genes is immense, with estimates in the order of $\sim 10^9$ TCR receptor variants [134, 135]. Because the pool of receptors is so enormous, T-cells specific for individual antigens are present at low frequency in blood even after clonal expansion. In order to detect such rare cells, techniques for measuring T-cell response must be endowed with very high levels of sensitivity.

The Enzyme-linked immunosorbent spot (ELISPOT) assay has been widely applied to monitor immune responses for the very reason of its exquisite sensitivity in detecting rare cell populations. In ELISPOT, T-cell response is measured in terms of quantity of cytokines secreted by activated T-cells. In a typical ELISPOT assay (Figure 6.1), T-cells are plated in presence and absence of antigen. The constitutive production of cytokines in absence of antigen constitutes the "background" level that must be subtracted from the antigen-induced signal. As the cytokines are bound by detection antibodies in the location where they are secreted, each spot that develops in the assay corresponds to an individual reactive T-cell, thus providing a quantitative measure of the response.

An alternative to ELISPOT is the Intracellular Cytokine Staining (ICS) assay. As the name suggests, it is based on the detection of cytokines within the endoplasmic reticulum after cell stimulation. By adding an inhibitor of

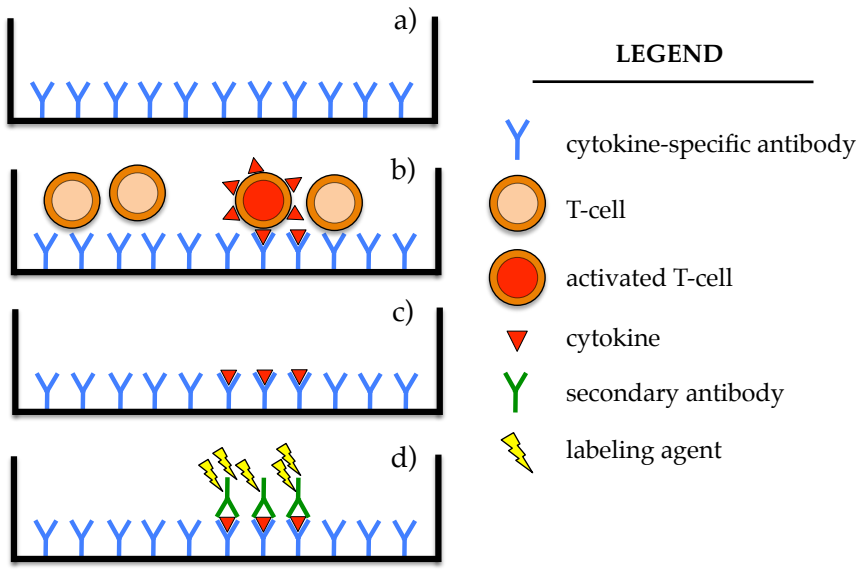



**Figure 6.1. A simple schematic of the ELISPOT assay.** **a**) The bottom of the plate is coated with cytokine-specific antibodies. **b**) Then, T-cells are added to the wells, with and without antigen. **c**) Antigen-stimulated cells secrete cytokines, which remain bound to their specific antibodies after the cells are washed off. **d**) Finally, a secondary antibody is added together with a labeling agent (e.g. avidin) to visually show the presence of cytokine.

protein transport, cytokines produced by an activated T-cell are retained inside the cell, and can be detected by a labelled antibody. Although not as sensitive as ELISPOT, ICS assays have the advantage of allowing detection of the responding cell type [136]. The two methods are often used in conjunction, first screening T-cell responses using an ELISPOT assay, and then using ICS for validation and to identify whether the responses are mediated by $CD4^+$ or $CD8^+$ T-cells. A further validation technique is tetramer staining [137], where HLA molecules are oligomerised (in a 4-molecule complex) to increase their avidity for T-cells, and can be used to study T-cell activation *in vitro* following antigen recognition by the MHC tetramer complex.

## 6.1 T-cell epitope mapping using peptide pools

Both in ELISPOT and ICS assays, the most precise approach to define the determinants of T-cell response is to test individual peptides, each in its own well. However, when the number of peptides is large, for instance when scanning a whole viral genome, testing individual peptides becomes unfeasible. A sharp reduction in the number of tests can be obtained by the use of peptide matrix pools, where each pool contains a mixture of several peptides. Matrices are organized into row pools and column pools, where each

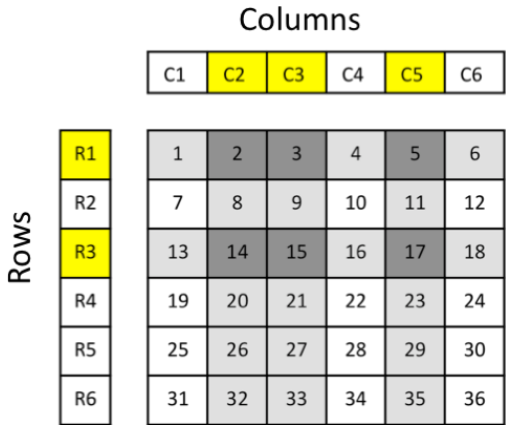



**Figure 6.2. Structure of a peptide pool matrix.** Pools are organized in rows and columns, with each peptide contained exactly in one row-pool and one column-pool. In this example, the matrix is composed of 12 pools (6 rows and 6 columns). If the assay on the 12 pools revealed that three columns (C2, C3, C5) and 2 rows (R1, R3) give a positive response, this leaves only 6 peptides as possible epitopes in this donor, those in the intersections of the responding pools (dark grey).

particular peptide is contained exactly in one row and one column (see for example [138, 139]).

In the example of Figure 6.2, a small matrix of 6x6 pools can be constructed for 36 peptides. Each peptide is present in one row and one column pool, so that a given row-column intersection identifies unequivocally one particular peptide. For example, peptide 11 is contained in row R2 and column C5, and no other peptide is found in this particular row-column combination. Instead of testing individual peptides for T-cell response, the row and column pools are tested. In the case of a 6x6 matrix, the 12 pools would be tested instead of the 36 individual peptide sequences. Then, the matrix intersection between rows and columns that gave a positive response pinpoint the peptides that could be responsible for the response. For instance, if rows R1 and R3, and columns C2, C3 and C5 produced response, the list of possible epitopes is narrowed down to the peptides found in the intersections of these pools (2,3,5,14,15,17). The 6 peptides would normally have to be then tested individually to determine the actual epitopes.

If the response is limited to very few pools, this approach is extremely effective, and reduces greatly the number of tests. However, matrices are commonly in the range of 10x10 up to 30x30, and there might be several epitopes contained in the peptide pools. In this case, many pools would produce a positive response, leading to a large number of intersections in the matrix and consequently prohibitive numbers of peptides to be tested individually. Furthermore, different individuals are characterized by different HLA phenotypes, and consequently they would generally respond to different epitopes depending on the MHC alleles involved in the response. The

matrices of each donor would need to be deconvoluted to obtain a reliable picture of the peptides consistently generating immune response in several donors, and spanning the HLA polymorphism of the cohort.

The following sections of this chapter describe a strategy to reduce the number of tests needed to identify epitopes, or rather rank the candidate peptides that are most likely to produce a T-cell response, based both on the matrix structure and allelic composition of the donors. Then we show how this method was used to identify T-cell epitopes from a large cohort of patients vaccinated for Yellow Fever, based on both ELISPOT and ICS data.

## 6.2 Filters and scoring

The first step in the analysis is to establish what should be considered as a positive T-cell response in an ELISPOT assay, as opposed to a negative response. There is no universally accepted rule for setting such a threshold on ELISPOT signal, but it is rather sensibly chosen based on experimentalists experience. Generally, this threshold should depend on the donor-specific background ($BG$), where the background is the ELISPOT signal detected in absence of antigen. For example, we may design a threshold $t_p$ for separating positive and negative responses according to the function:

$$t_p(BG) = \begin{cases} BG + 25 & \text{if } BG < 25 \\ BG + 2 \times BG & \text{if } 25 \leq BG < 50 \\ BG + 100 & \text{if } BG \geq 50 \end{cases} \tag{6.1}$$

Once a threshold $t_p$ is set, for each donor in the cohort there will be a number of positive pools, i.e. pools with ELISPOT score $S > t_p$. What is unknown is: *1)* which peptide(s) in the pool determined the response, and *2)* the restriction of the recognized peptide(s) to the donor's HLA alleles. Two filters are applied to the potential epitopes, and then results from different donors are combined to raise evidence for epitope candidates.

### Filter I - Row/Column intersections

The first filter aims at reducing the number of possible peptides that could have determined a T-cell response in a given donor. It exploits the structure of the pools matrix, where each peptide is present in exactly one row and one column pool, so that a given row-column intersection identifies unequivocally one particular peptide. As also described above and illustrated in Figure 6.2, potential epitopes in a specific donor are only the peptides found in intersections where both row and column produced a response. This procedure of identifying positive intersections is applied to all donors, with the outcome of a list of epitope candidates for each donor. Whilst prior to applying this filter all peptides were possible epitopes, now the candidates (for a certain donor) are limited to those in the positive intersections.

### Filter II - HLA allele restriction

The second filter exploits the knowledge on the HLA phenotype of the donors. Only peptides that are predicted to bind to any of the alleles of a given donor are retained as potential epitopes. This step clearly depends on the quality of the predictions, which differs between MHC molecules: MHC class I predictions are very accurate, with NetMHCcons [116] being currently the best method available for the task. Not nearly as good are methods for MHC class II binding, but the best publicly available is NetMHCIIpan [79]. However, it can only be applied to HLA-DRB molecules, and currently it has not been trained for HLA-DP and HLA-DQ molecules. For the lattest two, we may use NetMHCII [36], although this method is only limited to 6 HLA-DP and 6 HLA-DQ alleles.

To illustrate how the different filters operate consider the following example. Suppose that, in a given donor, the ELISPOT analysis produced positive scores for 7 row pools and 2 column pools, leading to 14 possible epitopes given by the row/column intersections (Filter I):

```
Peptide          Row   Col
MSGRKAQGKTLGVNM  R:2   C:7
KTLGVNMVRRGVRSL  R:9   C:7
QIGNRPGPSRGVQGF  R:11  C:7
RPGPSRGVQGFIFFF  R:14  C:7
QGFIFFFLFNILTGK  R:19  C:7
FFFLFNILTGKKITA  R:24  C:7
FNILTGKKITAHLKR  R:26  C:7
TGKKITAHLKRLWKM  R:2   C:26
GLAVLRKVKRVVASL  R:9   C:26
LRKVKRVVASLMRGL  R:11  C:26
ASLMRGLSSRKRRSH  R:14  C:26
RGLSSRKRRSHDVLT  R:19  C:26
SRKRRSHDVLTVQFL  R:24  C:26
RSHDVLTVQFLILGM  R:26  C:26
```

Then let us assume that this donor has the following phenotype for HLA-A, -B and -C:

```
HLA-A03:01,HLA-A11:01,HLA-B07:02,HLA-B08:02,HLA-C04:01,HLA-C07:01
```

Using NetMHCcons, we predict which of the 6 MHC class I molecules of the donor can potentially bind peptides in positive intersections, considering as binders peptides with predicted binding affinity stronger than 500nM (log-affinity > 0.426), or ranked among the top 2% peptides for a given allele. On the sequences above, we obtain (Filter II):

```
Peptide          Allele       LogAff  Rank   Epitope
FFFLFNILTGKKITA  HLA-A*03:01  0.629   0.25   FLFNILTGK
ASLMRGLSSRKRRSH  HLA-A*03:01  0.718   0.05   SLMRGLSSRK
KTLGVNMVRRGVRSL  HLA-A*11:01  0.655   0.80   KTLGVNMVR
KTLGVNMVRRGVRSL  HLA-B*07:02  0.749   0.08   MVRRGVRSL
RPGPSRGVQGFIFFF  HLA-B*07:02  0.605   0.40   RPGPSRGVQGF
```

For this particular donor, after the two filters the possible peptides producing a T-cell response are reduced to those listed above. Not only the list of candidates is narrowed down to fewer peptides, but in the process the restriction elements and minimal peptide epitopes are also predicted. Note that one peptide might be associated with several HLA alleles.

### Combining predictions from different donors

After the two filters, we obtain a list of possible peptide/allele combinations *for each donor*. Predictions for the different donors are combined in a very straightforward manner: simply by counting the number of occurrences $C_{p,a}$ of each peptide/allele pair $p, a$ over all the donors. For example, the peptide/allele association `FFFLFNILTGKKITA/HLA-A*03:01` might be observed in $C_{p,a} = 9$ donors. The next question is whether this number is or is not significantly different from random. It clearly depends to a great extent on the frequency of that allele across the donors: e.g. if only 10 donors in total carry the HLA-A*0301 allele, then there is a good chance that the peptide is an epitope; on the other hand, if A*0301 is found in 100 donors, then the 9 positive occurrences are likely to be due to other peptides in the same pools as `FFFLFNILTGKKITA`. In other words, if an allele is very frequent, then there is a higher chance to obtain peptide associations to it just by random, because the allele is found in so many donors. To quantify this effect, we calculate a Z-score based on a background frequency as described next.

### Background counts and Z-scores

The allele-dependent background counts are estimated by random permutations of the pool matrices. Maintaining the same matrix structure, positive pools are scrambled randomly to obtain new, permuted matrices. The filters can then be applied on such permuted matrices, and then calculate a count $BG_{p,a}$ for each peptide/allele pair. Note that in a scrambled matrix, each row-column pool intersection (and consequently any peptide) has the same probability of being positive, therefore the background count can be more conveniently expressed as $BG_a$ to only depend on the allele. The background count $BG_a$ expresses the probability of obtaining a positive association to allele $a$ after the filters given the number of positive pools in the matrices and the HLA allele distribution of the donors. Evidently, more common alleles obtain higher background counts simply because they have higher probability of generating random intersections.

For more reliable estimates of the $BG_a$ values, multiple versions (e.g. 1000) of the scrambled matrices can be calculated. This allows drawing a distribution of $BG_a$ values, with mean $\mu_a$ and standard deviation $\sigma_a$. Within this distribution, where is the observed count $C_{p,a}$ located? This question can be answered quantitatively by means of a Z-score, that indicates by how many standard deviations $C_{p,a}$ is above or below the mean $\mu_a$:

$$Z_{p,a} = \frac{C_{p,a} - \mu_a}{\sigma_a} \tag{6.2}$$

### Ranking epitope predictions

Higher Z-score means higher significance of a certain peptide-allele association, and can therefore be used to rank epitope candidates. The predicted strength of MHC binding $\text{aff}^L_{p,a}$ (in log-transformed units, $\text{aff}^L = \frac{1-\log(\text{aff})}{\log(50{,}000)}$ ) for peptide $p$ to allele $a$ can also be included in the score, using:

$$Z^*_{p,a} = Z_{p,a} \times \text{aff}^L_{p,a} \tag{6.3}$$

As $\text{aff}^L_{p,a}$ varies between 0 for a non-binder to 1 for the strongest binder, its inclusion in equation 6.3 penalizes in a linear manner weak binders compared to strong binders. In summary, ranking potential epitopes by $Z^*_{p,a}$ favors peptides found in several intersections, where the donors have a particular HLA phenotype and the peptides are preferentially predicted to be strong binders to their MHC molecule.

## 6.3 Predicting CD8$^+$ T-cell epitopes for YF virus

One of the most common vaccines against yellow fever (YF) is based on a 3,411 amino acids-long polyprotein (GenBank AF052437 [140]) from the 17D-204 yellow fever virus strain. A total of 875 peptides, mostly of length 15 amino acids and overlapping by 11 amino acids, were generated to cover the whole length of the construct, and distributed into a 30x30 matrix composed of 30 rows + 30 columns = 60 peptide pools. As discussed in section 6.1 each row-column intersection identifies unambiguously one and only one peptide. T-cell response for the 60 pools was measured using ELISPOT on blood sample from 92 donors, fully typed for HLA phenotype, who received vaccination against YF.

### 6.3.1 ELISPOT analysis

Using the rules of equation 6.1 to define positive responses, the matrices appeared very dense with 58% of row pools and 65% of column pools giving signal above the threshold. On such data, up to several hundreds of peptides for each donor should be individually tested to identify epitopes, a huge effort that defies the utility of using pools to reduce the number of tests. In order to reduce the complexity of this data set, we applied the computational filters introduced in the previous section to combine information from different donors and compile a prioritized list of candidate epitopes.

The predicted epitopes for three among the most common alleles (HLA-A*01:01, A*02:01 and B*07:02) are shown respectively in Tables 6.1, 6.2 and 6.3. Peptides are ranked according to $Z^*$ (see equation 6.3), limiting the list to peptides with $Z^* > 2$. We note that only few peptides are predicted as likely epitopes for A*01:01 and B*07:02, as opposed to A*02:01 where many peptides survive the filters with significant $Z^*$. A*02:01, compared to other HLA-I alleles, has a more promiscuous binding motif, with several residues

**Table 6.1.** Top predicted epitopes for HLA-A*01:01 from ELISPOT matrices.

| Peptide sequence | Count | Predicted Epitope | Z-score | Affinity | Z* | P/T |
|---|---|---|---|---|---|---|
| YTDYLTVMDRYSVDA | 17 | YTDYLTVMDRY | 3.957 | 0.934 | 3.695 | 5/6 |
| MSNPLTSPISCSYSL | 19 | LTSPISCSY | 4.935 | 0.670 | 3.307 | 0/3 |
| GQEKYTDYLTVMDRY | 16 | YTDYLTVMDRY | 3.467 | 0.934 | 3.238 | 5/6 |
| ERIKSEYMTSWFYDN | 16 | KSEYMTSWFY | 3.467 | 0.712 | 2.469 | 3/4 |

Green: predicted epitopes that give positive response in experimental validation (the fraction of positive responding donors is in column P/T). Red: peptides not giving response in any donor when tested individually. The HLA-A*01:01 background count for Z-score calculation is $\mu = 8.91$.

**Table 6.2.** Top predicted epitopes for HLA-A*02:01 from ELISPOT matrices.

| Peptide sequence | Count | Predicted Epitope | Z-score | Affinity | Z* | P/T |
|---|---|---|---|---|---|---|
| SLLWNGPMAVSMTGVK | 40 | LLWNGPMAV | 8.306 | 0.838 | 6.96 | 31/33 |
| MVLAGWLFHVRGARR | 30 | VLAGWLFHV | 4.823 | 0.913 | 4.404 | 7/9 |
| SRGVQGFIFFFLFNIKK | 31 | FIFFFLFNI | 5.171 | 0.819 | 4.235 | NB |
| MLMTGGVTLVRKNRW | 30 | MLMTGGVTLV | 4.823 | 0.860 | 4.148 | 0/2 |
| PSELQMSWLPLCVRL | 30 | LQMSWLPLCV | 4.823 | 0.805 | 3.883 | 0/1 |
| VYMDAVFEYTIDCDG | 28 | YMDAVFEYTI | 4.127 | 0.876 | 3.615 | 1/14 |
| QGFIFFFLFNILTGK | 28 | FIFFFLFNI | 4.127 | 0.819 | 3.38 | NB |
| ELNLLDKRQFELYKR | 31 | LLDKRQFEL | 5.171 | 0.643 | 3.325 | 1/6 |
| FLDPASIAARGWAAH | 28 | FLDPASIAA | 4.127 | 0.773 | 3.190 | 10/18 |
| KKNGGDAMYMALIAAFS | 29 | AMYMALIAA | 4.475 | 0.704 | 3.150 | 0/6 |
| MYMWLGARYLEFEAL | 28 | YMWLGARYL | 4.127 | 0.762 | 3.145 | 0/7 |
| AMYMALIAAFSIRPGK | 27 | YMALIAAFSI | 3.778 | 0.81 | 3.061 | |
| SGSAASMVNGVIKIL | 30 | SMVNGVIKI | 4.823 | 0.624 | 3.010 | 0/12 |
| DEAHFLDPASIAARG | 27 | FLDPASIAA | 3.778 | 0.773 | 2.921 | 10/16 |
| MVTMLSPMLHHWIKV | 27 | MLSPMLHHWI | 3.778 | 0.741 | 2.800 | |
| TGVMRGNHYAFVGVM | 28 | GVMRGNHYAFV | 4.127 | 0.676 | 2.79 | 0/6 |
| ALYEKKLALYLLLAL | 27 | ALYEKKLALYL | 3.778 | 0.729 | 2.755 | 0/7 |
| NLYKLHGGHVSCRVK | 29 | KLHGGHVSCRV | 4.475 | 0.61 | 2.730 | |
| RGNHYAFVGVMYNLW | 29 | YAFVGVMYNL | 4.475 | 0.554 | 2.479 | |
| LASVAMCRTPFSLAE | 28 | AMCRTPFSL | 4.127 | 0.594 | 2.451 | 0/6 |
| ILMTATPPGTSDEFP | 27 | ILMTATPPGT | 3.778 | 0.638 | 2.411 | 0/4 |
| VIIMDEAHFLDPASI | 25 | IIMDEAHFL | 3.082 | 0.77 | 2.373 | 1/11 |
| QTKIQYVIRAQLHVG | 28 | YVIRAQLHV | 4.127 | 0.564 | 2.327 | 0/5 |
| SLDISLETVAIDRPA | 27 | SLDISLETV | 3.778 | 0.608 | 2.297 | 3/10 |
| SMSLFEVDQTKIQYV | 27 | SLFEVDQTKI | 3.778 | 0.607 | 2.294 | |
| GKATLECQVQTAVDFKK | 28 | ATLECQVQTAV | 4.127 | 0.551 | 2.274 | |
| KILTYPWDRIEEVTR | 29 | KILTYPWDRI | 4.475 | 0.498 | 2.229 | |
| VRNGKKLIPSWASVK | 24 | KLIPSWASV | 2.734 | 0.796 | 2.176 | 0/5 |
| IVDRQWAQDLTLPWQ | 29 | RQWAQDLTL | 4.475 | 0.481 | 2.152 | 0/4 |
| NSFQIEEFGTGVFTT | 24 | FQIEEFGTGV | 2.734 | 0.78 | 2.132 | 0/2 |

Green: predicted epitopes that give positive response in experimental validation (the fraction of positive responding donors is in column P/T). Red: peptides not giving response in any donor when tested individually. Yellow: there could not be established binding between the MHC molecule and the peptide. The HLA-A*02:01 background count for Z-score calculation is $\mu = 16.15$.

allowed at its anchor positions, and many of these residues being among the most frequent in natural protein sequences. Out of the 875 peptides in the pools, 363 are predicted to have binding affinity < 500nM for A*02:01, as opposed to only 46 and 173 for A*01:01 and B*07:02 respectively. Predicted epitopes were then validated by tetramer staining on individual vaccinated donors. For HLA-A*01:01, for example, 3 out of 4 predicted epitopes were validated by tetramer staining (green rows in Table 6.1). Several epitopes were also confirmed for A*02:01, although a considerable number of false positives were found. Three epitopes were confirmed for B*07:02.

There are two major factors in the experimental setup that may lead to false positives. Firstly, the size of the peptide pools. Using a 30x30 matrix

**Table 6.3.** Top predicted epitopes for HLA-B*07:02 from ELISPOT matrices.

| Peptide sequence | Count | Predicted Epitope | Z-score | Affinity | Z* | P/T |
|---|---|---|---|---|---|---|
| LWSPRERLVLTLGAA | 21 | SPRERLVLTL | 4.048 | 0.811 | 3.283 | 7/8 |
| DDCVVRPIDDRFGLA | 21 | RPIDDRFGL | 4.048 | 0.690 | 2.793 | 9/9 |
| MPRSIGGPVSSHNHI | 19 | MPRSIGGPV | 3.174 | 0.831 | 2.638 | 0/5 |
| VRPIDDRFGLALSHL | 19 | RPIDDRFGLAL | 3.174 | 0.799 | 2.536 | 9/9 |
| MSNPLTSPISCSYSL | 19 | SPISCSYSL | 3.174 | 0.759 | 2.409 | |
| KTLGVNMVRRGVRSL | 19 | MVRRGVRSL | 3.174 | 0.749 | 2.377 | 0/3 |
| PPHAATIRVLALGNQ | 22 | HAATIRVLAL | 4.485 | 0.514 | 2.305 | |
| GKNLVFSPGRKNGSF | 20 | SPGRKNGSF | 3.611 | 0.634 | 2.289 | |
| VFSPGRKNGSFIIDG | 20 | SPGRKNGSF | 3.611 | 0.634 | 2.289 | |
| TILPLMALLTPVTMA | 21 | LPLMALLTPV | 4.048 | 0.558 | 2.259 | |

Green: predicted epitopes that give positive response in experimental validation (the fraction of positive responding donors is in column P/T). Red: peptides not giving response in any donor when tested individually. The HLA-B*07:02 background count for Z-score calculation is $\mu = 11.74$.

setup implies that each pool is composed of 30 peptides, and if at least one peptide can generate a response to any of the HLA alleles of the donor the pool will be positive. Indeed, we observed a very high rate of positive responses (> 50% of the pools), and consequently very dense matrices, difficult to deconvolute. Secondly, ELISPOT cannot distinguish between CD4$^+$ and CD8$^+$ responses. In the ELISPOT analysis we assumed that the T-cell response was CD8-restricted and ignored the other kind of response. However, a considerable portion of positive pools are likely due to CD4-restricted responses, introducing a high level of noise in the data. This may have a particularly large impact on A2, which has a motif characterized by hydrophobic anchors similarly to the prevalent HLA-DR molecules. For example the predicted A2 epitope FIFFFLFNI, which did not bind to the HLA-I molecule (table 6.2), is contained in a 15-mer shown to be a dominant DRB1*01:01 epitope. A subsequent analysis based on Intracellular Cytokine Staining (ICS), described next, aims to address both these sources of noise, by reducing the sizes of the matrices and differentiating the type of T-cell response.

### 6.3.2 ICS analysis

The ICS assay allows monitoring which T-cell type is responding to a given antigen. Therefore we can study CD8$^+$ responses separately from CD4$^+$ responses, and by that remove one of the sources of noise. The yellow fever peptides were arranged into four 15x15 matrices, with the advantage of having smaller pools (15 peptides), but the drawback of a larger number of pools to be tested for each donor ($15 \times 2 \times 4 = 120$). ICS assays on the complete set of pools were performed on a total of 20 donors. As in the case of the ELISPOT data, we applied the filters to predict and rank CD8-restricted T-cell epitopes. The matrices appear less dense than in the ELISPOT experiment, with 28% of the row pools and 37% of the column pools being positive. From the observation that most of the validated epitopes were strong binders, here we applied a stronger threshold for the filter on predicted binding affinity, limiting the NetMHCcons predictions to affinity < 50nM or %rank < 0.5.

**Table 6.4.** Top predicted epitopes for HLA-A*01:01 from ICS matrices.

| Peptide sequence | Count | BG | Predicted Epitope | Z-score | Affinity | Z* |
|---|---|---|---|---|---|---|
| YTDYLTVMDRYSVDA | 6 | 0.660 | YTDYLTVMDRY | 6.736 | 0.934 | 6.291 |
| GQEKYTDYLTVMDRY | 5 | 0.660 | YTDYLTVMDRY | 5.474 | 0.934 | 5.113 |
| ERIKSEYMTSWFYDN | 4 | 0.660 | KSEYMTSWFY | 4.213 | 0.712 | 3.000 |
| MSNPLTSPISCSYSL | 4 | 0.660 | LTSPISCSY | 4.213 | 0.670 | 2.823 |

Green: validated epitopes. Red: false positives.

**Table 6.5.** Top predicted epitopes for HLA-A*02:01 from ICS matrices.

| Peptide sequence | Count | BG | Predicted Epitope | Z-score | Affinity | Z* |
|---|---|---|---|---|---|---|
| SLLWNGPMAVSMTGVK | 8 | 1.037 | LLWNGPMAV | 6.962 | 0.838 | 5.834 |
| KKEGNTSLLWNGPMAVS | 8 | 1.037 | LLWNGPMAV | 6.962 | 0.838 | 5.834 |
| IVLASAALGPLIEGN | 9 | 1.037 | VLASAALGPLI | 7.961 | 0.654 | 5.207 |
| KKPFALLLVLAGWLFHV | 4 | 1.037 | VLAGWLFHV | 2.963 | 0.913 | 2.705 |
| LLVLAGWLFHVRGAR | 4 | 1.037 | VLAGWLFHV | 2.963 | 0.913 | 2.705 |
| NMEVRGGMVAPLYGV | 4 | 1.037 | GMVAPLYGV | 2.963 | 0.852 | 2.524 |
| RGGMVAPLYGVEGTK | 4 | 1.037 | GMVAPLYGV | 2.963 | 0.852 | 2.524 |
| NALSMMPEAMTIVML | 4 | 1.037 | SMMPEAMTIV | 2.963 | 0.826 | 2.447 |
| TMAEVRLAAMFFCAVKK | 4 | 1.037 | RLAAMFFCAV | 2.963 | 0.816 | 2.417 |
| LMALLTPVTMAEVRL | 4 | 1.037 | ALLTPVTMAEV | 2.963 | 0.775 | 2.296 |
| IWYMWLGARYLEFEAKK | 4 | 1.037 | YMWLGARYL | 2.963 | 0.762 | 2.257 |

Green: validated epitopes. Red: false positives.

**Table 6.6.** Top predicted epitopes for HLA-B*07:02 from ICS matrices.

| Peptide sequence | Count | BG | Predicted Epitope | Z-score | Affinity | Z* |
|---|---|---|---|---|---|---|
| LWSPRERLVLTLGAA | 4 | 0.713 | SPRERLVLTL | 3.995 | 0.811 | 3.240 |
| VRPIDDRFGLALSHL | 4 | 0.713 | RPIDDRFGLAL | 3.995 | 0.799 | 3.192 |
| NGPMAVSMTGVMRGN | 4 | 0.713 | GPMAVSMTGVM | 3.995 | 0.727 | 2.904 |
| SAALGPLIEGNTSLL | 4 | 0.713 | GPLIEGNTSL | 3.995 | 0.585 | 2.337 |
| GPLIEGNTSLLWNGP | 4 | 0.713 | GPLIEGNTSL | 3.995 | 0.585 | 2.337 |
| PEMPALYEKKLALYL | 3 | 0.713 | MPALYEKKLAL | 2.780 | 0.769 | 2.137 |
| FFMSPKGISRMSMAM | 3 | 0.713 | SPKGISRMSM | 2.780 | 0.753 | 2.093 |

Green: validated epitopes.

The predicted epitopes for the HLA alleles A*01:01, A*02:01 and B*07:02 with $Z^* > 2$ are listed in Tables 6.4, 6.5 and 6.6 respectively. We note that for A*01:01 the same 4 peptides as in ELISPOT are predicted to be epitopes, with 3 of them validated experimentally. Similarly, 2 out of 3 B*07:02 epitopes found with ELISPOT are replicated in ICS, additionally a new one is found and there are no false positives. As for A*02:01, most of the false positives predicted on ELISPOT data disappear. However, at the same time several validated A2 epitopes are not predicted with significant Z-scores on ICS data. For example, both 15-mers containing the 9-mer FLDPASIAA obtained high ranking on ELISPOT data, and were tetramer-validated on more than half of the donors. In the ICS assay, the intersections identifying these peptides responded positive only in 2 donors out of the 10 carrying A*02:01, clearly not a significant ratio. Similar arguments can be made for the other false negatives.

### 6.3.3 Discussion

Only a small fraction of the 875 YF peptides were tested using tetramers thus far: 3 positive and 1 negative for A*01:01, 25 positive and 13 negative for A*02:01, 7 positive and 5 negative for B*07:02. Although one can safely assume that the large majority of the 875 peptides do not contain epitopes for a given HLA-I molecule, we are far from having a picture of the totality of CD8$^+$ epitopes in the YF proteome. However, based on these preliminary results, there appears to be a large overlap in the epitope predictions based on ELISPOT and ICS. A number of epitopes were suggested by both approaches, and several of these candidates were later confirmed by tetramer staining.

On the validated epitopes, we estimated the predictive power of different ranking methods by evaluating the sensitivity as a function of the %rank. That is, based on a given epitope ranking strategy, how many peptides should be tested to discover a certain fraction of known epitopes. Figure 6.3 compares three ranking criteria for each of the 3 class I alleles: predicted binding affinity of the optimal 9-mer, $Z^*$-scores based on ELISPOT data, and $Z^*$-scores on ICS data. In the case of A*01:01, the 3 validated epitopes are effectively recovered by all methods, in particular by the $Z^*$-scores where they were all found among the top 4 ranking candidates.

For what concerns A*02:01, about half of the validated epitopes could have been found by testing only a handful of peptides based on any of the three proposed rankings. However, when it came to recover the totality of the known epitopes, there were major differences between the methods. In particular, it appeared that a number of epitopes were completely missed by the Z-score-based methods. Examining the experimental data for these peptides, we observed that they did not give positive response significantly more often than random in the ELISPOT and ICS experimental pools. For example, the validated A*02:01 epitope GLYGNGILV was contained in two 15-mer peptides: VIGLYGNGILVGDNS and RNGEVIGLYGNGILV. In the ELISPOT assay, respectively 18 and 17 of the 43 donors carrying A*02:01 responded to the two peptides, not significantly higher than $\mu = 16.15$ expected by chance given the frequency of the allele ($Z = 0.644$ and $Z = 0.296$, respectively). Moreover, on ICS matrices none of the 10 A2 donors responded positive to either of the two 15-mer peptides. Based on such evidence, nothing would suggest that these two peptides contain an epitope. We can make similar arguments for the remaining epitopes not identified by the matrix-based methods.

Out of the 7 validated B*07:02 epitopes, all three ranking strategies identified rapidly 4 epitopes among the highest ranking peptides. However, compared to the other two approaches, the $Z^*$-score based on ICS matrices fails to identify the remaining 3 validated epitopes. As in the case of the undetected A2 epitopes, the explanation can be found in the peptide pool responses − there were few or no positive intersections for these peptides in the ICS matrices.

In summary, it appears that predicted binding affinity is the most important factor to consider when ranking potential epitopes. The experimental

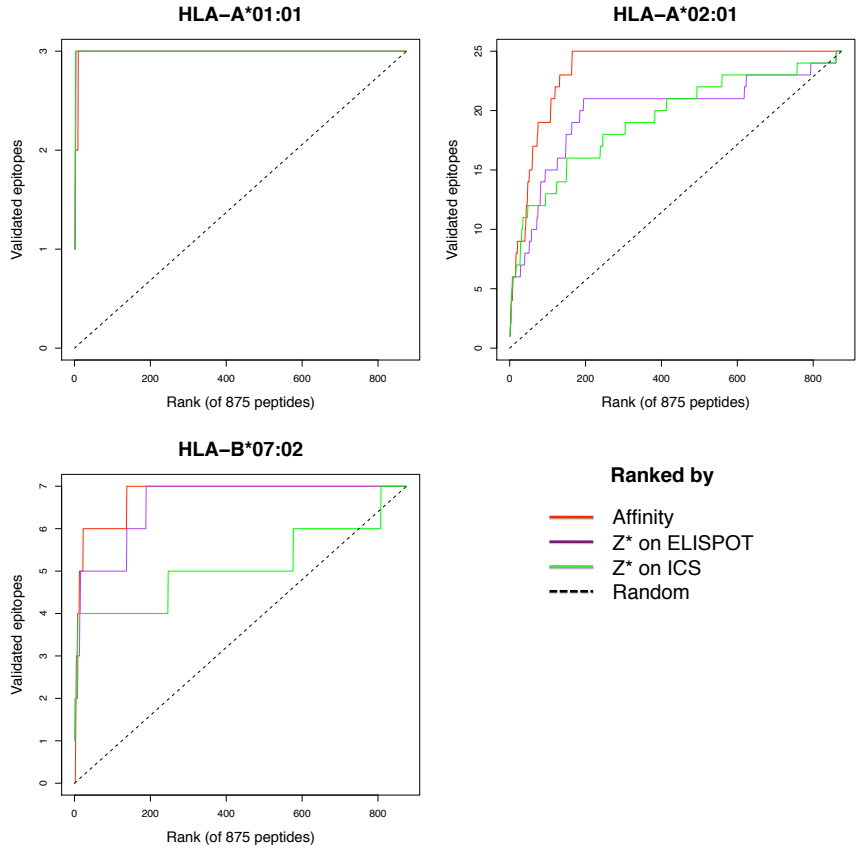



**Figure 6.3. Rank vs. sensitivity for the three HLA-I alleles A*01:01, A*02:01, B*07:02.** The 875 YF peptides were ranked, for each allele, using three different schemes: by predicted binding affinity of the optimal 9-mer, by Z-score from ELISPOT matrices, by Z-score from ICS matrices. The dashed diagonal lines represent the expected rank by a random ordering of peptides.

peptide pool-based approaches, though effectively identifying a number of $CD8^+$ epitopes, could not find the totality of the validated epitopes. One aspect to consider is that peptide pools were composed mainly of 15-mer peptides, which would have to undergo digestion and processing during an ELISPOT or ICS assay before being presented to the MHC class I molecules in the form of a 9-mer (or 10/11-mer). As such trimming may not always happen, some epitopes may be overlooked in these assays. In the case of ICS, perhaps stronger evidence could have been collected with a larger number of donors. The present data set contained peptide-pool measurements for only 20 donors, and even the most common HLA molecule (A*02:01) was present in only 10 donors. Statistical significance on such small data sets is difficult to achieve, and it would certainly benefit from a larger number of observations. Finally, as discussed previously only a small fraction of the 875 YF

peptides were tetramer-validated on their restriction element, and these peptides were not singled out for validation randomly. A complete benchmark of the method would require an unbiased mapping of all epitopes in the data set. Yet, these preliminary results suggest that by employing computational techniques it is possible to reduce greatly the number of tests necessary to identify T-cell epitopes. Experimental analysis of all possible peptides, even in a peptide-pools setup, can be extremely time-consuming and automated ranking strategies are necessary to rationally guide epitope discovery.

# Chapter 7

# Epilogue

The research presented in this thesis was conducted with one major objective: to provide computational methods for the analysis of peptide data. In the era of "big data", bioinformatics techniques are indispensable to extract in an automatic manner meaningful patterns from extremely large experimental data sets. In this thesis, I presented a series of different algorithms and tools to identify sequence motifs in peptide data, tackling the problem from different angles and applying these methods to several biological problems.

First, we demonstrated the power of artificial neural networks in identifying sequence motifs in large-scale peptide data sets. The *NNAlign* method presented in chapter 2, based on ANNs, was benchmarked on different kinds of peptide-based data sets, but also applied for motif discovery to characterize the binding motifs of HLA molecules to an unprecedented level of detail. Part of the success of ANNs derives from their ability to exploit the quantitative nature of peptide data. In the example of peptide-MHC binding used widely in this thesis, peptide binding can be considered as a quantitative event, for example in terms of the ligand concentration required for at least half of the peptides to be bound to the MHC ($IC_{50}$ binding affinity). In other words, one can measure with some numerical scale the strength of the interaction. Regression methods such as neural networks are capable of capturing these quantitative aspects.

Although ANNs are able to pick up higher-order correlations, they do not provide an ideal framework for the detection of multiple specificities in peptide data. In chapter 4 we suggested a new approach, based on multiple position-specific scoring matrices, that facilitates the interpretation of peptide data in terms of poly-specificity of the receptor. It was applied to deconvolute mixtures of MHC class I and class II molecules, to identify diverse classes of binders to the SH3 domain, and in the characterization of

sub-motifs of the HLA-A*02:01 molecule. The Gibbs clustering method has the limitation that it only exploits the qualitative nature of the data, i.e. only positive instances of a given biological event are considered to derive the sequence motif(s). In future perspective, it is possible to envision the incorporation of quantitative aspects into the poly-specificity framework, for example by training multiple ANNs in parallel, each "responsible" for one of the multiple specificities.

Yet another method for pattern analysis is presented in chapter 5, which constructs kernel functions for sequence similarity based uniquely on the BLOSUM62 substitution matrix. In particular, we employed this algorithm to overcome one of the limitations of current methods for prediction of MHC class I binding: that training sequences must be of equal length. The method combines peptides of different length to achieve predictive performance higher than when trained on the individual peptide lengths. This may be particularly useful for prediction of peptide-MHC interactions where experimental data for a certain peptide length is scarce.

Finally, chapter 6 outlined a computational strategy to aid the discovery of T-cell epitopes from ELISPOT and ICS experiments based on peptide-pool matrices. We demonstrated that the method reduced greatly the number of tests necessary to discover T-cell epitopes in the yellow fever proteome.

There are numerous directions in which the work in this thesis could be continued, and room for improvement. It would be pretentious to state otherwise. But I remain with the hope that it advanced ever so little our understanding of sequence alignment problems and that it may serve as inspiration for others.

# Bibliography


[1] Gould CM, Diella F, Via A, Puntervoll P, Gemünd C, et al. (2010) ELM: the status of the 2010 eukaryotic linear motif resource. Nucleic acids research 38: D167-D180. 1

[2] Hooda Y, Kim PM (2012) Computational structural analysis of protein interactions and networks. Proteomics 12: 1697-1705. 1

[3] Bratkovič T (2010) Progress in phage display: evolution of the technique and its applications. Cellular and molecular life sciences 67: 749-767. 1, 52

[4] Uttamchandani M, Yao SQ (2008) Peptide microarrays: next generation biochips for detection, diagnostics and high-throughput screening. Current pharmaceutical design 14: 2428-2438. 1, 4, 5, 17, 52

[5] Rao X, Costa AICAF, van Baarle D, Keşmir C (2009) A comparative study of HLA binding affinity and ligand diversity: implications for generating immunodominant CD8+ T cell responses. The Journal of Immunology 182: 1526-1532. 3, 21, 63

[6] Nielsen M, Lund O, Buus S, Lundegaard C (2010) MHC class II epitope predictive algorithms. Immunology 130: 319-328. 3, 18

[7] Sidney J, Peters B, Frahm N, Brander C, Sette A (2008) HLA class I supertypes: a revised and updated classification. BMC immunology 9: 1. 3

[8] Lund O, Nielsen M, Kesmir C, Petersen AG, Lundegaard C, et al. (2004) Definition of supertypes for HLA molecules using clustering of specificity matrices. Immunogenetics 55: 797-810. 3, 53, 59

[9] Kirksey TJ, Pogue-Caley RR, Frelinger JA, Collins EJ (1999) The structural basis for the increased immunogenicity of two HIV-reverse transcriptase peptide variant/class I major histocompatibility complexes. Journal of Biological Chemistry 274: 37259-37264. 4

[10] Painter CA, Negroni MP, Kellersberger KA, Zavala-Ruiz Z, Evans JE, et al. (2011) Conformational lability in the class II MHC 310 helix and adjacent extended strand dictate HLA-DM susceptibility and peptide exchange. Proceedings of the National Academy of Sciences 108: 19329-19334. 4

[11] Schrödinger, LLC (2010) The PyMOL molecular graphics system, version 1.2r1. 4

[12] Sette A, Adorini L, Colon SM, Buus S, Grey HM (1989) Capacity of intact proteins to bind to MHC class II molecules. The Journal of Immunology 143: 1265. 4

[13] Schutkowski M, Reineke U, Reimer U (2005) Peptide arrays for kinase profiling. Chembiochem 6: 513-521. 4, 17, 52

[14] Hecker M, Lorenz P, Steinbeck F, Hong L, Riemekasten G, et al. (2012) Computational analysis of high-density peptide microarray data with application from systemic sclerosis to multiple sclerosis. Autoimmunity reviews 11: 180-190. 4, 6

[15] Reimer U, Pawlowski N, Seznec J, Knaute T, von Hoegen P, et al. (2012) Universal mapping of humoral immune response using a versatile high-content and high-density peptide microarray. Retrovirology 9: P17. 4

[16] Thiele A, Posel S, Spinka M, Zerweck J, Reimer U, et al. (2011) Profiling of enzymatic activities using peptide arrays. Mini-Reviews in Organic Chemistry 8: 147-156. 4

[17] Buus S, Rockberg J, Forsström B, Nilsson P, Uhlen M, et al. (2012) High-resolution mapping of linear antibody epitopes using ultrahigh-density peptide microarrays. Molecular & Cellular Proteomics, in press . 4

[18] Panicker RC, Sun H, Chen GYJ, Yao SQ (2009) Peptide-based microarray. Microarrays : 139-167. 5

[19] UniProt Consortium (2008) The universal protein resource (UniProt). Nucleic Acids Res 36: 61-79. 6, 33

[20] Henikoff S, Henikoff JG (1994) Position-based sequence weights. Journal of molecular biology 243: 574-578. 7, 11

[21] Hobohm U, Scharf M, Schneider R, Sander C (1992) Selection of representative protein data sets. Protein Science 1: 409-417. 8, 31

[22] Henikoff S, Henikoff JG (1992) Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences 89: 10915. 8, 32, 44, 71

[23] Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic acids research 25: 3389-3402. 8, 11

[24] Gfeller D, Butty F, Wierzbicka M, Verschueren E, Vanhee P, et al. (2011) The multiple-specificity landscape of modular peptide recognition domains. Molecular systems biology 7. 9, 50, 53

[25] Nielsen M, Lundegaard C, Worning P, Lauemøller SL, Lamberth K, et al. (2003) Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Science 12: 1007-1017. 9, 50, 76

[26] Schneider TD, Stephens RM (1990) Sequence logos: a new way to display consensus sequences. Nucleic acids research 18: 6097-6100. 11, 34, 44

[27] Schneider TD, Stormo GD, Gold L, Ehrenfeucht A (1986) Information content of binding sites on nucleotide sequences. Journal of molecular biology 188: 415-431. 11

[28] Crooks GE, Hon G, Chandonia JM, Brenner SE (2004) WebLogo: a sequence logo generator. Genome research 14: 1188-1190. 11, 12, 26, 33

[29] Workman CT, Yin Y, Corcoran DL, Ideker T, Stormo GD, et al. (2005) enoLOGOS: a versatile web tool for energy normalized sequence logos. Nucleic acids research 33: W389-W392. 11

[30] Vacic V, Iakoucheva LM, Radivojac P (2006) Two Sample Logo: a graphical representation of the differences between two sets of sequence alignments. Bioinformatics 22: 1536-1537. 11

[31] Colaert N, Helsens K, Martens L, Vandekerckhove J, Gevaert K (2009) Improved visualization of protein consensus sequences by iceLogo. Nature methods 6: 786-787. 11

[32] Thomsen MCF, Nielsen M (2012) Seq2Logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. Nucleic Acids Research 40: W281-W287. 11, 12, 54

[33] Nielsen M, Lundegaard C, Worning P, Hvid CS, Lamberth K, et al. (2004) Improved prediction of MHC class I and class II epitopes using a novel Gibbs sampling approach. Bioinformatics 20: 1388-1397. 13, 18, 50, 52, 54, 56, 61, 67

[34] Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, et al. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science 262: 208-214. 13, 52, 56

[35] Noguchi H, Kato R, Hanai T, Matsubara Y, Honda H, et al. (2002) Hidden Markov model-based prediction of antigenic peptides that interact with MHC class II molecules. Journal of bioscience and bioengineering 94: 264-270. 13, 53

[36] Nielsen M, Lund O (2009) NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. BMC bioinformatics 10: 296. 13, 18, 21, 22, 24, 25, 32, 35, 43, 53, 74, 85

[37] Wang P, Sidney J, Kim Y, Sette A, Lund O, et al. (2010) Peptide binding predictions for HLA DR, DP and DQ molecules. BMC bioinformatics 11: 568. 13, 18, 21, 22, 30, 40, 43, 45, 53

[38] James W (2001) Nucleic acid and polypeptide aptamers: a powerful approach to ligand discovery. Current opinion in pharmacology 1: 540-546. 16

[39] Hoppe-Seyler F, Crnkovic-Mertens I, Tomai E, Butz K (2004) Peptide aptamers: specific inhibitors of protein function. Current molecular medicine 4: 529-538. 16

[40] Lin J, Bardina L, Shreffler WG, Andreae DA, Ge Y, et al. (2009) Development of a novel peptide microarray for large-scale epitope mapping of food allergens. Journal of Allergy and Clinical Immunology 124: 315-322. 17

[41] Han X, Yamanouchi G, Mori T, Kang JH, Niidome T, et al. (2009) Monitoring protein kinase activity in cell lysates using a high-density peptide microarray. Journal of biomolecular screening 14: 256-262. 17

[42] Halperin RF, Stafford P, Johnston SA (2011) Exploring antibody recognition of sequence space through random-sequence peptide microarrays. Molecular & Cellular Proteomics 10: M110 000786. 17, 52

[43] Masch A, Zerweck J, Reimer U, Wenschuh H, Schutkowski M, et al. (2010) Antibody signatures defined by high-content peptide microarray analysis. Methods in Molecular Biology 669: 161-172. 17

[44] Christensen JK, Lamberth K, Nielsen M, Lundegaard C, Worning P, et al. (2003) Selecting informative data for developing peptide-MHC binding predictors using a query by committee approach. Neural computation 15: 2931-2942. 17

[45] Dyrløv Bendtsen J, Nielsen H, von Heijne G, Brunak S (2004) Improved prediction of signal peptides: SignalP 3.0. Journal of molecular biology 340: 783-795. 17

[46] Juncker AS, Willenbrock H, Von Heijne G, Brunak S, Nielsen H, et al. (2003) Prediction of lipoprotein signal peptides in Gram-negative bacteria. Protein Science 12: 1652-1662. 17

[47] Kim JH, Lee J, Oh B, Kimm K, Koh I (2004) Prediction of phosphorylation sites using SVMs. Bioinformatics 20: 3179-3184. 17

[48] Blom N, Gammeltoft S, Brunak S (1999) Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. Journal of molecular biology 294: 1351-1362. 17

[49] Obenauer JC, Cantley LC, Yaffe MB (2003) Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs. Nucleic acids research 31: 3635-3641. 17

[50] Schwartz D, Gygi SP (2005) An iterative statistical approach to the identification of protein phosphorylation motifs from large-scale data sets. Nature biotechnology 23: 1391-1398. 17

[51] Lundegaard C, Lund O, Buus S, Nielsen M (2010) Major histocompatibility complex class I binding predictions as a tool in epitope discovery. Immunology 130: 309-318. 17

[52] Bhasin M, Raghava GPS (2004) SVM based method for predicting HLA-DRB1* 0401 binding peptides in an antigen sequence. Bioinformatics 20: 421-423. 18

[53] Nielsen M, Lundegaard C, Lund O (2007) Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method. BMC bioinformatics 8: 238. 18, 21, 22, 31

[54] Castelletti D, Fracasso G, Righetti S, Tridente G, Schnell R, et al. (2004) A dominant linear B-cell epitope of ricin A-chain is the target of a neutralizing antibody response in Hodgkin's lymphoma patients treated with an anti-CD25 immunotoxin. Clinical & Experimental Immunology 136: 365-372. 18

[55] Hua R, Zhou Y, Wang Y, Hua Y, Tong G (2004) Identification of two antigenic epitopes on SARS-CoV spike protein. Biochemical and biophysical research communications 319: 929-935. 18

[56] Sompuram SR, Vani K, Hafer LJ, Bogen SA (2006) Antibodies immunoreactive with formalin-fixed tissue antigens recognize linear protein epitopes. American journal of clinical pathology 125: 82-90. 18

[57] Peters B, Bui HH, Frankild S, Nielsen M, Lundegaard C, et al. (2006) A community resource benchmarking predictions of peptide binding to MHC-I molecules. PLoS computational biology 2: e65. 21, 22, 29, 75

[58] Singh H, Raghava GPS (2001) ProPred: prediction of HLA-DR binding sites. Bioinformatics 17: 1236-1237. 21, 22

[59] Sturniolo T, Bono E, Ding J, Raddrizzani L, Tuereci O, et al. (1999) Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. Nature biotechnology 17: 555-561. 21, 22

[60] Rammensee HG, Bachmann J, Emmerich NPN, Bachor OA, Stevanović S (1999) SYFPEITHI: database for MHC ligands and peptide motifs. Immunogenetics 50: 213-219. 21, 60

[61] Lovitch SB, Pu Z, Unanue ER (2006) Amino-terminal flanking residues determine the conformation of a peptide-class II MHC complex. The Journal of Immunology 176: 2958-2968. 21

[62] Godkin AJ, Smith KJ, Willis A, Tejada-Simon MV, Zhang J, et al. (2001) Naturally processed HLA class II peptides reveal highly conserved immunogenic flanking region sequence preferences that reflect antigen processing rather than peptide-MHC interactions. The Journal of Immunology 166: 6720-6727. 21

[63] Petersen TN, Lundegaard C, Nielsen M, Bohr H, Bohr J, et al. (2000) Prediction of protein secondary structure at 80% accuracy. Proteins: Structure, Function, and Bioinformatics 41: 17-20. 24

[64] Olsen JV, Ong SE, Mann M (2004) Trypsin cleaves exclusively C-terminal to arginine and lysine residues. Molecular & Cellular Proteomics 3: 608-614. 27

[65] Schilling O, Huesgen PF, Barré O, auf dem Keller U, Overall CM (2011) Characterization of the prime and non-prime active site specificities of proteases by proteome-derived peptide libraries and tandem mass spectrometry. Nature protocols 6: 111-120. 28

[66] Lathe W, Williams J, Mangan M, Karolchik D (2008) Genomic data resources: challenges and promises. Nature Education 1. 28

[67] Nilsson T, Mann M, Aebersold R, Yates JR, Bairoch A, et al. (2010) Mass spectrometry in high-throughput proteomics: ready for the big time. Nature methods 7: 681-685. 28

[68] Frank E, Hall M, Trigg L, Holmes G, Witten IH (2004) Data mining in bioinformatics using Weka. Bioinformatics 20: 2479-2481. 29

[69] Bailey TL, Williams N, Misleh C, Li WW (2006) MEME: discovering and analyzing DNA and protein sequence motifs. Nucleic acids research 34: W369-W373. 29, 50, 52

[70] Neduva V, Russell RB (2006) DILIMOT: discovery of linear motifs in proteins. Nucleic acids research 34: W350-W355. 29

[71] Singh-Gasson S, Green RD, Yue Y, Nelson C, Blattner F, et al. (1999) Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array. Nature biotechnology 17: 974-978. 30

[72] Hasan A, Stengele KP, Giegrich H, Cornwell P, Isham KR, et al. (1997) Photolabile protecting groups for nucleosides: Synthesis and photodeprotection rates. Tetrahedron 53: 4247-4264. 30

[73] Bhushan KR, DeLisi C, Laursen RA (2003) Synthesis of photolabile 2-(2-nitrophenyl) propyloxycarbonyl protected amino acids. Tetrahedron letters 44: 8585-8588. 30

[74] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. The journal of chemical physics 21: 1087-1092. 33

[75] Hansen LK, Salamon P (1990) Neural network ensembles. Pattern Analysis and Machine Intelligence, IEEE Transactions on 12: 993-1001. 35

[76] Opitz DW, Shavlik JW (1996) Actively searching for an effective neural network ensemble. Connection Science 8: 337-354. 35

[77] Zhou ZH, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all. Artificial intelligence 137: 239-263. 35

[78] Robinson J, Waller MJ, Fail SC, McWilliam H, Lopez R, et al. (2009) The IMGT/HLA database. Nucleic acids research 37: D1013-D1017. 39

[79] Nielsen M, Lundegaard C, Blicher T, Peters B, Sette A, et al. (2008) Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan. PLoS computational biology 4: e1000107. 40, 53, 85

[80] Rapin N, Hoof I, Lund O, Nielsen M (2008) MHC motif viewer. Immunogenetics 60: 759-765. 40

[81] Mehra NK, Kaur G (2003) MHC-based vaccination approaches: progress and perspectives. Expert reviews in molecular medicine 2: 1-17. 40

[82] Jones EY, Fugger L, Strominger JL, Siebold C (2006) MHC class II proteins and disease: a structural perspective. Nature Reviews Immunology 6: 271-282. 42

[83] Fernando MMA, Stevens CR, Walsh EC, De Jager PL, Goyette P, et al. (2008) Defining the role of the MHC in autoimmunity: a review and pooled analysis. PLoS genetics 4: e1000024. 42

[84] Fallang LE, Bergseng E, Hotta K, Berg-Larsen A, Kim CY, et al. (2009) Differences in the risk of celiac disease associated with HLA-DQ2. 5 or HLA-DQ2. 2 are related to sustained gluten antigen presentation. Nature immunology 10: 1096-1101. 42

[85] Mandic M, Castelli F, Janjic B, Almunia C, Andrade P, et al. (2005) One NY-ESO-1-derived epitope that promiscuously binds to multiple HLA-DR and HLA-DP4 molecules and stimulates autologous CD4+ T cells from patients with NY-ESO-1-expressing melanoma. The Journal of Immunology 174: 1751-1759. 42

[86] Qian F, Gnjatic S, Jäger E, Santiago D, Jungbluth A, et al. (2004) Th1/Th2 CD4+ T cell responses against NY-ESO-1 in HLA-DPB1* 0401/0402 patients with epithelial ovarian cancer. Cancer Immun 4: 12. 42

[87] Kamatani Y, Wattanapokayakit S, Ochi H, Kawaguchi T, Takahashi A, et al. (2009) A genome-wide association study identifies variants in the HLA-DP locus associated with chronic hepatitis B in asians. Nature genetics 41: 591-595. 42

[88] Sidney J, Steen A, Moore C, Ngo S, Chung J, et al. (2010) Divergent motifs but overlapping binding repertoires of six HLA-DQ molecules frequently expressed in the worldwide human population. The Journal of Immunology 185: 4189-4198. 43, 46, 47, 48

[89] Sidney J, Steen A, Moore C, Ngo S, Chung J, et al. (2010) Five HLA-DP molecules frequently expressed in the worldwide human population share a common HLA supertypic binding specificity. The Journal of Immunology 184: 2492-2503. 43, 45, 48

[90] Bui HH, Sidney J, Peters B, Sathiamurthy M, Sinichi A, et al. (2005) Automated generation and evaluation of specific MHC binding predictive tools: ARB matrix applications. Immunogenetics 57: 304-314. 43

[91] Andreatta M, Schafer-Nielsen C, Lund O, Buus S, Nielsen M (2011) NNAlign: A web-based prediction method allowing non-expert end-user discovery of sequence motifs in quantitative peptide data. PloS one 6: e26781. 43, 44, 50, 53

[92] Bugawan TL, Horn GT, Long CM, Mickelson E, Hansen JA, et al. (1988) Analysis of HLA-DP allelic sequence polymorphism using the in vitro enzymatic DNA amplification of DP-alpha and DP-beta loci. The Journal of Immunology 141: 4024. 45

[93] Díaz G, Amicosante M, Jaraquemada D, Butler RH, Guillén MV, et al. (2003) Functional analysis of HLA-DP polymorphism: a crucial role for DP$\beta$ residues 9, 11, 35, 55, 56, 69 and 84-87 in T cell allorecognition and peptide binding. International immunology 15: 565-576. 45

[94] Castelli FA, Buhot C, Sanson A, Zarour H, Pouvelle-Moratille S, et al. (2002) HLA-DP4, the most frequent HLA II molecule, defines a new supertype of peptide-binding specificity. The Journal of Immunology 169: 6928. 45

[95] Greenbaum J, Sidney J, Chung J, Brander C, Peters B, et al. (2011) Functional classification of class II human leukocyte antigen (HLA) molecules reveals seven different supertypes and a surprising degree of repertoire sharing across supertypes. Immunogenetics 63: 325-335. 45

[96] Lee KH, Wucherpfennig KW, Wiley DC (2001) Structure of a human insulin peptide-HLA-DQ8 complex and susceptibility to type 1 diabetes. Nature immunology 2: 501-507. 45, 47

[97] Megiorni F, Mora B, Bonamico M, Barbato M, Nenna R, et al. (2009) HLA-DQ and risk gradient for celiac disease. Human immunology 70: 55-59. 45

[98] Hovhannisyan Z, Weiss A, Martin A, Wiesner M, Tollefsen S, et al. (2008) The role of HLA-DQ8 $\beta$ 57 polymorphism in the anti-gluten T-cell response in coeliac disease. Nature 456: 534-538. 45

[99] Siebold C, Hansen BE, Wyer JR, Harlos K, Esnouf RE, et al. (2004) Crystal structure of HLA-DQ0602 that protects against type 1 diabetes and confers strong susceptibility to narcolepsy. Proceedings of the National Academy of Sciences of the United States of America 101: 1999. 46

[100] Ettinger RA, Kwok WW (1998) A peptide binding motif for HLA-DQA1* 0102/DQB1* 0602, the class II MHC molecule associated with dominant protection in insulin-dependent diabetes mellitus. The Journal of Immunology 160: 2365-2373. 46

[101] Sidney J, del Guercio MF, Southwood S, Sette A (2002) The HLA molecules DQA1*0501 / B1*0201 and DQA1*0301 / B1*0302 share an extensive overlap in peptide binding specificity. The Journal of Immunology 169: 5098-5108. 47

[102] van de Wal Y, Kooy YMC, Drijfhout JW, Amons R, Papadopoulos GK, et al. (1997) Unique peptide binding characteristics of the disease-associated DQ ($\alpha$1*0501, $\beta$1*0201) vs the non-disease-associated DQ ($\alpha$1*0201, $\beta$1*0202) molecule. Immunogenetics 46: 484-492. 47

[103] Quarsten H, Paulsen G, Johansen BH, Thorpe CJ, Holm A, et al. (1998) The P9 pocket of HLA-DQ2 (non-Asp$\beta$57) has no particular preference for negatively charged anchor residues found in other type 1 diabetes-predisposing non-Asp$\beta$57 MHC class II molecules. International immunology 10: 1229-1236. 47

[104] Stepniak D, Wiesner M, de Ru AH, Moustakas AK, Drijfhout JW, et al. (2008) Large-scale characterization of natural ligands explains the unique gluten-binding properties of HLA-DQ2. The Journal of Immunology 180: 3268-3278. 47

[105] Vartdal F, Johansen BH, Friede T, Thorpe CJ, Stevanović S, et al. (1996) The peptide binding motif of the disease associated HLA-DQ ($\alpha$ 1*0501, $\beta$ 1*0201) molecule. European journal of immunology 26: 2764-2772. 47

[106] Bailey TL, Elkan C (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization. Machine learning 21: 51-80. 50, 53

[107] Kim TH, Tyndel MS, Huang H, Sidhu SS, Bader GD, et al. (2011) MUSI: an integrated system for identifying multiple specificity from very large peptide or nucleic acid data sets. Nucleic Acids Research : 1-8. 50, 53, 54, 67, 69

[108] Koivunen E, Arap W, Rajotte D, Lahdenranta J, Pasqualini R (1999) Identification of receptor ligands with phage display peptide libraries. Journal of nuclear medicine 40: 883. 52

[109] Soen Y, Chen DS, Kraft DL, Davis MM, Brown PO (2003) Detection and characterization of cellular immune responses using peptide-MHC microarrays. PLoS biology 1: e65. 52

[110] Gfeller D (2012) Uncovering new aspects of protein interactions through analysis of specificity landscapes in peptide recognition domains. FEBS letters, in press . 52

[111] Vita R, Zarebski L, Greenbaum JA, Emami H, Hoof I, et al. (2010) The immune epitope database 2.0. Nucleic acids research 38: D854-D862. 53

[112] Harndahl M, Rasmussen M, Roder G, Dalgaard Pedersen I, Sørensen M, et al. (2012) Peptide-MHC class I stability is a better predictor than peptide affinity of CTL immunogenicity. European Journal of Immunology 42: 1405-1416. 54, 67, 68

[113] Rand WM (1971) Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association : 846-850. 57

[114] Hubert L, Arabie P (1985) Comparing partitions. Journal of classification 2: 193-218. 57

[115] Yewdell JW, Bennink JR (1999) Mechanisms of viral interference with MHC class I antigen processing and presentation. Annual review of cell and developmental biology 15: 579-606. 63

[116] Karosiene E, Lundegaard C, Lund O, Nielsen M (2012) NetMHCcons: a consensus method for the major histocompatibility complex class I predictions. Immunogenetics 64: 177-186. 63, 78, 85

[117] Yu H, Chen JK, Feng S, Dalgarno DC, Brauer AW, et al. (1994) Structural basis for the binding of proline-rich peptides to SH3 domains. Cell 76: 933-945. 65

[118] Mayer BJ (2001) SH3 domains: complexity in moderation. Journal of Cell Science 114: 1253-1263. 65

[119] Carducci M, Perfetto L, Briganti L, Paoluzi S, Costa S, et al. (2012) The protein interaction network mediated by human SH3 domains. Biotechnology advances 30: 4-15. 65

[120] Saksela K, Permi P (2012) SH3 domain ligand binding: What s the consensus and where s the specificity? FEBS letters 586: 2609-2614. 65, 67

[121] Assarsson E, Sidney J, Oseroff C, Pasquetto V, Bui HH, et al. (2007) A quantitative analysis of the variables affecting the repertoire of T cell specificities recognized after vaccinia virus infection. The Journal of Immunology 178: 7890-7901. 67

[122] Busch DH, Pamer EG (1998) MHC class I/peptide stability: implications for immunodominance, in vitro proliferation, and diversity of responding CTL. The Journal of Immunology 160: 4441-4448. 67

[123] Geironson L, Røder G, Paulsson K (2012) Stability of peptide-HLA-I complexes and tapasin folding facilitation-tools to define immunogenic peptides. FEBS letters 586: 1336-1343. 67

[124] Shen WJ, Wong HS, Xiao QW, Guo X, Smale S (2012) Towards a mathematical foundation of immunology and amino acid chains. Arxiv preprint arXiv:12056031 . 71, 73, 74, 75, 80

[125] Peters B, Sette A (2005) Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method. BMC bioinformatics 6: 132. 74, 76, 77

[126] Lundegaard C, Lamberth K, Harndahl M, Buus S, Lund O, et al. (2008) NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8-11. Nucleic acids research 36: W509-W512. 77, 78

[127] Wang P, Sidney J, Dow C, Mothé B, Sette A, et al. (2008) A systematic assessment of MHC class II peptide binding predictions and evaluation of a consensus approach. PLoS Computational Biology 4: e1000048. 78

[128] Moutaftsi M, Peters B, Pasquetto V, Tscharke DC, Sidney J, et al. (2006) A consensus epitope prediction approach identifies the breadth of murine $T_{CD8+}$-cell responses to vaccinia virus. Nature biotechnology 24: 817-819. 78

[129] Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. ACM, pp. 144-152. 79

[130] Salomon J, Flower DR (2006) Predicting class II MHC-peptide binding: a kernel based approach using similarity scores. BMC bioinformatics 7: 501. 79, 80

[131] Saigo H, Vert JP, Ueda N, Akutsu T (2004) Protein homology detection using string alignment kernels. Bioinformatics 20: 1682-1689. 80

[132] Rangwala H, Karypis G (2005) Profile-based direct kernels for remote homology detection and fold recognition. Bioinformatics 21: 4239-4247. 80

[133] Kim Y, Sidney J, Pinilla C, Sette A, Peters B (2009) Derivation of an amino acid similarity matrix for peptide: MHC binding and its application as a Bayesian prior. BMC bioinformatics 10: 394. 80

[134] Arstila TP, Casrouge A, Baron V, Even J, Kanellopoulos J, et al. (1999) A direct estimate of the human $\alpha\beta$ T cell receptor diversity. Science 286: 958-961. 81

[135] Naylor K, Li G, Vallejo AN, Lee WW, Koetz K, et al. (2005) The influence of age on T cell generation and TCR diversity. The Journal of Immunology 174: 7446. 81

[136] Meddows-Taylor S, Shalekoff S, Kuhn L, Gray GE, Tiemessen CT (2007) Development of a whole blood intracellular cytokine staining assay for mapping CD4+ and CD8+ T-cell responses across the HIV-1 genome. Journal of virological methods 144: 115-121. 82

[137] Altman JD, Moss PAH, Goulder PJR, Barouch DH, McHeyzer-Williams MG, et al. (1996) Phenotypic analysis of antigen-specific T lymphocytes. Science 274: 94-96. 82

[138] Anthony DD, Lehmann PV (2003) T-cell epitope mapping using the ELISPOT approach. Methods 29: 260-269. 83

[139] Radošević K, Wieland CW, Rodriguez A, Weverling GJ, Mintardjo R, et al. (2007) Protective immune responses to a recombinant adenovirus type 35 tuberculosis vaccine in two mouse strains: CD4 and CD8 T-cell epitope mapping and role of gamma interferon. Infection and immunity 75: 4105-4115. 83

[140] Xie H, Cass AR, Barrett ADT (1998) Yellow fever 17D vaccine virus isolated from healthy vaccinees accumulates very few mutations. Virus research 55: 93-99. 87