

Support Vector Machine prediction of MHC-peptide binding

Rebecca Watson

Department of Computer Science & Software Engineering

The University of Western Australia

35 Stirling Highway

Crawley WA 6009

Australia

email: rebeccaw@cs.uwa.edu.au

*This report is submitted as partial fulfilment
of the requirements for the Honours Programme of the
Department of Computer Science & Software Engineering,
The University of Western Australia,
2001*

Abstract

Advances in surgical ability now mean that the success of organ donor transplants depend almost entirely upon our ability to match donor and recipient. Exact tissue matches however are not required due to the availability of immunosuppressive drugs such as Cyclosporine. The development of immunosuppressive drugs requires an understanding of our immune system and in particular the process of MHC restriction which involves MHC-peptide binding. However, limited data is available regarding specific MHC-peptide pairs and prediction algorithms can be applied to bridge the gap between available data and the required information.

This work investigates the use of support vector machines (SVM) to 'learn' rules from available experimental data on MHC-peptide binding in order to predict the binding ability of given MHC-peptide pairs. A number of different kernel functions were investigated, including: dot product, radial basis function (RBF), several polynomial kernels, and an engineered kernel. Building examples from an existing repository for peptide binding data, a database was constructed and populated with 1170 entries obtained from recent experiments available in the literature. Importantly, this database includes much information on non-binding MHC-peptide pairs. An XML implementation of this data has been constructed and this information was used to train SVM for different MHC molecules.

Overall prediction performance of the trained SVM varied, depending upon the data available for the MHC-peptide pairs. Particularly promising results were obtained for allelic products A*0101, A*2601, A*2602, B*0801, B*2702, B*2704, B*2705, B*2706, and B*3503 where prediction ability ranged between about 80% to 100%, exceeding the performance of other predictive techniques. Promising results were also achieved for the B*27 group of alleles, significantly improving the individual performances of every B*27 subtype (B*2701-6).

A web interface has also been developed so that the trained SVM are available world-wide. Additionally, the web site allows automatic submission of new peptide-binding data to the database.

Keywords: SVM, support vector machine, MHC-peptide binding, MHCPEP, canonical binding, organ transplant, immunosuppressive drug, peptide blocker

CR Categories: I:2:1 [Artificial Intelligence]:Applications and Expert Systems - Medicine and science

I:2:6 [Artificial Intelligence]:Learning - Connectionism and neural nets

Acknowledgements

I would like to thank my supervisor Associate Professor Gareth Chelvanayagam for his help and support, and my family and friends who have also provided invaluable support throughout the year. Finally, I would also like to thank Professor Grundy of Columbia University who supplied the SVM implementation.

List of Tables

4.1	SVM performance of A*0101 allele. ^a Number of correctly classified binding entries. ^b Number of incorrectly classified non-binding entries. ^c Number of correctly classified non-binding entries. ^d Number of incorrectly classified binding entries. ^e Sensitivity; the proportion of correctly classified binding examples. ^f Specificity; the proportion of correctly classified non-binding examples. ^g Predictive (Binding); the probability that an entry classified by SVM as binding is correct. ^h Predictive (Non-Binding); the probability that an entry classified by SVM as non-binding is correct. ⁱ Overall; the proportion of correctly classified entries.	24
4.2	Individual alleles that illustrated high prediction accuracy. [†] Table fields as per Table 4.1.	25
4.3	Groups of alleles that illustrated high prediction accuracy. [†] Table fields as per Table 4.1.	27
4.4	A comparison of SVM and ANN classifiers. [†] Table fields as per Table 4.1.	29
4.5	Effect of weak features on SVM performance. [†] Table fields as per Table 4.1.* SVM trained using weak features.	30
4.6	Questionable MHCPEP_ <i>R</i> entries. ^a The unique entry identification number from MHCPEP_ <i>R</i> . ^b Allele contained in the entry. ^c Peptide contained in the entry.	31
4.7	Analysis of non-binding prediction algorithm. [†] Fields as per Table 4.1. [*] <i>NEG</i> field contains the proportion of predicted non-binding entries classified as non-binding by SVM.	32

List of Figures

1.1	Amino Acid Structure. Amino acids differ in their side-chain group illustrated by the ‘X’ section of the model. This side chain is unique for each amino acid, and is responsible for the unique structure and physiochemical properties of each amino acid.	2
1.2	Each person receives one set of HLA molecules from each parent. There are a number of each type of these molecules (shown in brackets) resulting in a considerable set of possible haplotypes (an individual’s particular set of the MHC molecules). Chromosome 6 codes for these regions and positions 1-180 in each of these segments code for the alpha-1 and alpha-2 domains of the allele. .	3
1.3	The alpha-1 and alpha-2 domains of the allele form the cleft in which the peptide binds. Contours in the cleft (that vary between allelic product via polymorphic positions) determine whether or not a particular peptide can bind to the allele.	3
1.4	A model of MHC-peptide binding illustrating the peptide in the cleft of the allele.	4
1.5	Canonical binding model consists of a nine-mer peptide, a simplified view of this peptide is shown in (a). Amino acids in positions 2 and 9 of the peptide ‘anchor’ in pockets 2 and 9 of the allele (b).	5
2.1	Non-linear mapping from input space to feature space via the function Φ (implied by the kernel function). An optimal linear hyperplane is determined in feature space, resulting in a corresponding non-linear decision boundary in input space.	9
2.2	An example of a hyperplane separating two classes.	10
2.3	Two classes separated by an oriented hyperplane in input space. Support vectors are data points closest to the hyperplane (on the margin) that represent the hyperplane implicitly in input space. .	11

4.1	The B*27 group of alleles performance. Performance for each individual allele in the group (B*2701–6) were determined (subtype number appears on x-axis). The red bars represent the performance of individual alleles in the group, while the blue bars represent the performance of the individual alleles (using peptide features only). Almost all alleles increased in performance (over all performance measures) except B*2701. Although B*2701's Sensitivity and Predictive(NB) decreased, this is more than compensated for by the increase in both Sensitivity and Predictive(B). B*2703 showed a marked increase in prediction ability.	28
-----	--	----

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
2 Review of Previous Work	7
2.1 Support Vector Machines	7
2.1.1 Support Vector Machine Theory	8
2.1.2 Support Vector Machine Development and Application . .	11
2.1.3 Support Vector Machine Availability	14
2.2 Biological Research	14
2.2.1 Peptide Motifs	14
2.2.2 Prediction Algorithms	15
2.2.3 Binding Matrices	15
2.2.4 Artificial Neural Network	15
2.3 Biological Resources	16
2.3.1 MHCPEP Database	16
2.3.2 AAindex Database	16
2.3.3 HLA Informatics Group	16
3 Constructing MHCPEP_<i>R</i>	17
3.1 Data Collection	17
3.2 Prediction Algorithm	18
3.2.1 Amino Acid Similarity Measure	18
3.2.2 Non-Binding Prediction	18
3.2.3 Prediction Analysis	19

4	Peptide binding prediction with Support Vector Machine	21
4.1	Performance Measures	21
4.2	Experimental Results and Discussion	22
4.2.1	Kernel Experimentation	23
4.2.2	Generalisation Performance	26
4.2.3	Feature Selection	29
4.2.4	MHCPEP_ <i>R</i> Data Analysis	30
5	Web Interface	33
5.1	MHCPEP_ <i>R</i>	33
5.2	Support Vector Machines	33
5.3	Data Deposition	33
6	Conclusions and Future Work	34
A	Original Honours Proposal	36
A.1	Background	36
A.1.1	Biological Resources	36
A.1.2	Support Vector Machine (SVM) Resources	36
A.2	Aim	37
A.3	Method	37
A.3.1	SVM specification	37
A.3.2	Code SVM	38
A.3.3	Training and Testing SVM	39
A.3.4	GUI Specification and Construction	39
A.4	Software and Hardware Requirements	40

CHAPTER 1

Introduction

In 1823, German surgeon Carl Bunger performed the first reported transplant, grafting skin from a woman's thigh to her nose. French physiologist Paul Bert discovered that tissues transplanted from one person to another were rejected. The cause of this rejection, the immune system, was discovered over forty years later by German biologist Carl O. Jensen. Though surgical techniques had developed further by the turn of the 20th century, rejection remained a problem. French immunologist Jean Dausset discovered a system for tissue matching in 1958, which minimised donor-recipient differences. Combined with the first immunosuppressive drugs, transplants were first made possible in the 1950's. Transplants however remained rare until the discovery of Cyclosporine (an immunosuppressive drug) in 1972, marketed in 1983. Surgical advances have now progressed to the point where organ donor success rates are determined almost entirely by our ability to understand and elude the body's complex immune system.

Using *Tissue Serology* to match donor and recipient, combined with immunosuppressive drugs, the chance of a successful transplant today is generally around 80%. However, the success rate of Tissue Serology is only around 10–15%, and immunosuppressive therapies have proven to be a crucial factor in the acceptance of transplanted organs. Developing an accurate method of predicting *MHC-peptide binding* (resulting in the formulation of immunosuppressive *peptide blockers*) would add another stage to the process of organ transplantation, theoretically increasing the organ transplant success rate.

At the center of the immune system is the process of *MHC restriction*, discovered in the early seventies by researchers at the Australian National University who received a Nobel Prize in recognition of their work. MHC restriction involves MHC-peptide binding, leading to the recognition of foreign proteins (such as a virus or bacteria) by our immune system. In order to predict this binding, an understanding of the process of MHC-peptide binding (and the canonical binding model) is required.

A peptide is a short fragment of protein and proteins form much of the cell.

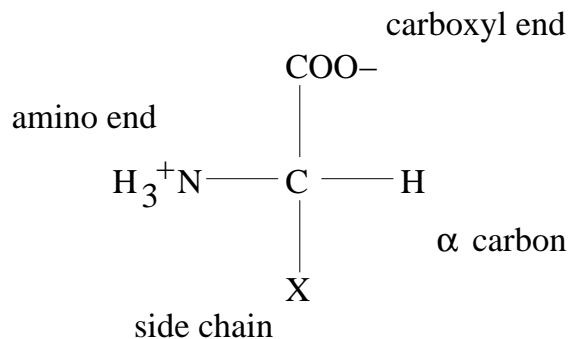


Figure 1.1: Amino Acid Structure. Amino acids differ in their side-chain group illustrated by the ‘X’ section of the model. This side chain is unique for each amino acid, and is responsible for the unique structure and physiochemical properties of each amino acid.

Peptides and proteins are constructed from a set of 20 building blocks: the amino acids. Peptides consist of (and can be expressed as) a linear sequence of amino acids. For example, the peptide FRDYVDRFY has 9 amino acids (termed a nine-mer peptide) and contains an Arginine (R) in position 2 of the peptide. The ‘side chain’ of an amino acid differs between amino acids (see Figure 1.1). In a peptide sequence, each amino acid’s side chain protrudes from the linear peptide structure and the properties of the side-chains along the peptide structure affect the peptide’s ability to bind to a particular MHC molecule.

Major histocompatibility complex (MHC) molecules (alleles) are cell surface proteins that bind to peptides, presenting them on the surface of our host cells. Peptides derived from foreign proteins (such as a virus or bacteria) should then be recognised by T-cell receptors that will trigger an immune response.

In humans, there are three class I MHC molecules that present peptides: HLA-A; HLA-B; and HLA-C. There are a number of different variants of HLA-A(120), HLA-B(250), and HLA-C(70) molecules. Each person has two of each of the HLA types of MHC molecules in their immune system, one set from each parent (see Figure 1.2). Therefore the number of distinct MHC molecules per person ranges from three (homozygous: the copy from both parents were the same for *HLA-A*, *HLA-B*, and *HLA-C*) and six (heterozygous: the copy from the parents were different at *HLA-A*, *HLA-B*, and *HLA-C*). A person’s *haplotype* is their specific set of MHC molecules. People vary as much at a genetic level as in their physical appearance. While similar physical characteristics are exhibited in ethnic groups, so too are the ‘flavours’ (haplotypes) of MHC molecules.

The structure of an allele (MHC molecule) resembles a hot-dog bun with a

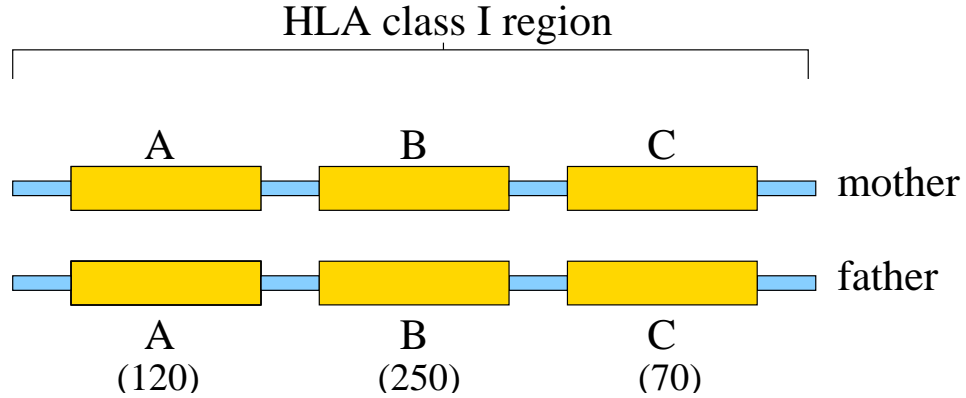


Figure 1.2: Each person receives one set of HLA molecules from each parent. There are a number of each type of these molecules (shown in brackets) resulting in a considerable set of possible haplotypes (an individual's particular set of the MHC molecules). Chromosome 6 codes for these regions and positions 1-180 in each of these segments code for the alpha-1 and alpha-2 domains of the allele.

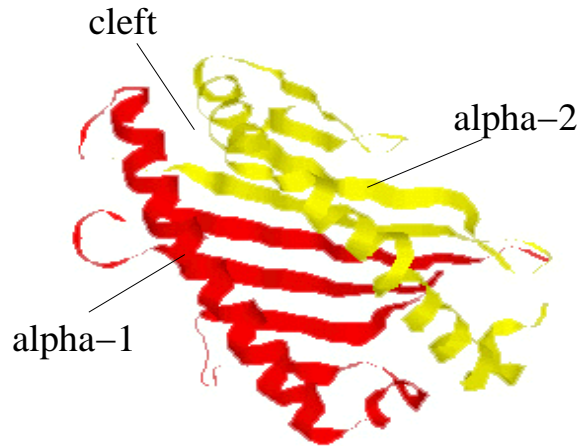


Figure 1.3: The alpha-1 and alpha-2 domains of the allele form the cleft in which the peptide binds. Contours in the cleft (that vary between allelic product via polymorphic positions) determine whether or not a particular peptide can bind to the allele.

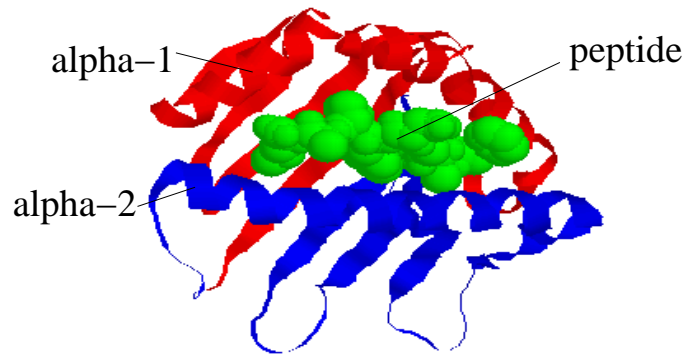


Figure 1.4: A model of MHC-peptide binding illustrating the peptide in the cleft of the allele.

long cleft that ‘locks around’ the peptide (see Figure 1.3). Figure 1.4 illustrates the peptide binding in this cleft. Chromosome 6 codes for the alpha-1 and alpha-2 domains (the HLA region) that contains polymorphic positions (those positions in which amino acids may differ between alleles). Amino acid positions between 1 and 180 in each HLA region code for the alpha-1 and alpha-2 domains of the allele. These positions therefore result in the different variants of HLA alleles (polymorphic allelism) and the unique shape or contour of the allele’s cleft. The different variants of alleles are referred to as types, for example, HLA-A2 is a type of HLA-A allele. Further, HLA-A0201 (denoted A*0201) is a sub-type of HLA-A2 and represents a specific HLA-A allele.

The allele cleft consists of several *pockets* (volumes of space), the characteristics of a pocket determine that pocket’s binding preferences for particular amino acid side chains. The peptide ligand’s side chains interact with pockets (in the corresponding position), the preferences of each pocket either increasing or interfering with the overall binding affinity of a particular peptide [28]. A well accepted model of binding is the canonical binding model. Canonical binding of peptides to class I HLA alleles constitutes the following (see Figure 1.5):

1. The peptide is of length nine amino acids, termed a nine-mer peptide.
2. The backbone of amino acids in positions 1–3 and 7–9 of the peptide adopt specific angles around interatomic bonds, effectively fixing the directions in which their side chains project.
3. As the backbone has a certain structure, both amino acids two and nine

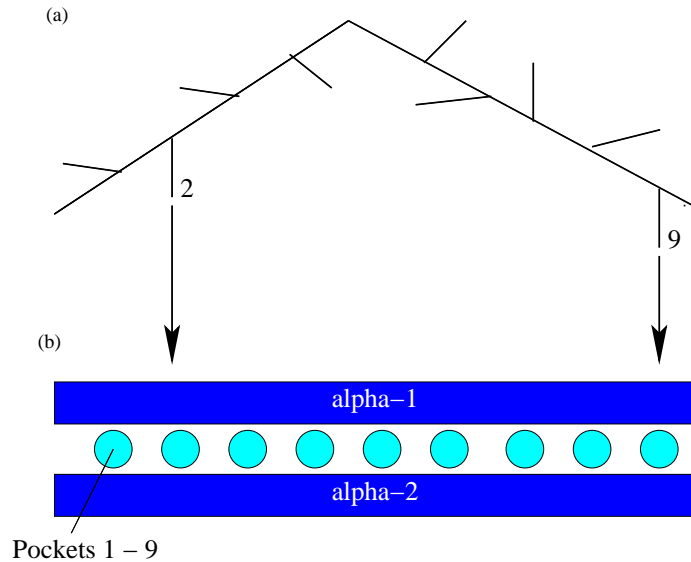


Figure 1.5: Canonical binding model consists of a nine-mer peptide, a simplified view of this peptide is shown in (a). Amino acids in positions 2 and 9 of the peptide ‘anchor’ in pockets 2 and 9 of the allele (b).

project directly into the cleft of the allele (into pockets two and nine). However, through allelic variation, the contours within the cleft change, and other ‘anchor’ positions in the peptide can become important.

4. Positions 4–6 tend to loop out, forming the flexible portion of the peptide bound structure. This may be necessary, as it is this section of the peptide that the T-cell receptor recognises.

Using Tissue Serology, it is possible to determine the haplotypes (an individual's set of MHC molecules) of the recipient required of the donor. If we could determine which peptides canonically bind to these types of MHC molecules, then we could design *peptide blockers*. A peptide blocker is a solution of peptides that will bind to the MHC molecules (of the donor's haplotype), effectively ‘blocking’ these sites from presenting an antigen (to the recipient's T-cell receptors) that will trigger an immune response. Guery et al. found the use of such peptide blockers for MHC class II molecules could “efficiently and selectively prevent the induction of T cell-dependent primary and secondary in vivo antibody responses by blocking antigen presentation to class II-restricted T helper cells.” [17] Therefore peptide blockers could be applied as an immunosuppressive therapy. MHC-peptide binding prediction has a number of other applications

including the proposed peptide-bound vaccines (similar to those implemented by Apton [13]).

Support vector machines (SVM) are a relatively new type of supervised machine learning that have proven to be particularly attractive to biological analysis due to their ability to handle noise, large data sets, and large input spaces. SVM application to the problem of MHC-peptide binding prediction is particularly attractive due to SVM's performance and reliability when compared to other classifiers [9, 31]. In this work, we apply SVM to the problem of predicting which peptides are likely to bind to which class I MHC molecules. Our analysis is restricted to that subset of class I molecules for which there is peptide binding/non-binding data available in the literature.

CHAPTER 2

Review of Previous Work

Due to initiatives such as the Human Genome Project, a large amount of biological data is becoming available and the need for bioinformatic tools that can handle large data sets is becoming increasingly apparent. Molecular biologists have recognised the potential of sophisticated machine learning methods in biological identification or prediction problems [14]. Support vector machines (SVM) have illustrated a number of characteristics that make them attractive to biological analysis. The theory of SVM and their application to multiple areas of biological analysis is explored in Section 2.1. Section 2.2 then considers past and current research into the prediction of MHC-peptide binding. A brief overview in Section 2.3 outlines the biological resources required to apply SVM to MHC-peptide binding prediction.

2.1 Support Vector Machines

Support vector machines are a form of supervised computer learning developed by Vladimir Vapnik and colleagues at Bell Laboratories [1]. Vapnik combined disciplines and ideas that had been around since the 1960's, stemming from roots in both the neural information processing community, and in computational learning theory [14]. Having combined two key ideas; an optimum margin classifier and kernel function, SVM are often seen as an extension of artificial neural networks (due to the similarity between the optimum margin classifier utilised by SVM and artificial neural network's perceptron algorithm). SVM however, are able to incorporate domain knowledge via the use of a kernel function and are not dependent on the initial choice of weights.

SVM are built upon the foundation of statistical learning theory, designed to minimise the structural (error) risk (reducing the task to an optimisation problem). A major advantage of SVM is therefore that the classification problem can be solved reliably, even when we are required to find a complicated decision boundary [20]. Another major advantage of SVM learning, is SVM's ability to

condense information in the training data and provide a sparse representation using a subset of this data (support vectors) [4]. SVM are therefore able to classify datasets with a very large number of features (high dimensional input space).

Flach [14] highlights the recent trend towards the combination of approaches from separate research communities. SVM are able to incorporate existing algorithms in terms of their kernel methods. For example, radial basis function networks have been implemented using the radial basis function (RBF) kernel. Therefore the strengths of existing algorithms can be incorporated into the SVM architecture. However, in order to become successful, SVM will need to illustrate their ability to ‘fine tune’ to specific domain areas. This highlights the need for researchers and machine learners to understand the other’s field at the research level.

2.1.1 Support Vector Machine Theory

Often we will wish to classify data into two distinct categories. Formally this involves finding a classification function f , where

$$f : R^n \rightarrow \{\pm 1\} \tag{2.1}$$

based on training data such that f will correctly classify unseen examples [26].

Training data are considered to be points in n dimensional space and are labeled with a binary classification. A hyperplane, that separates this data into the two labeled categories, can then be used to classify a new datum point depending on the datum’s position relative to this hyperplane. Unfortunately, a linear hyperplane does not exist for most real-world data.

SVM overcome this problem by mapping the data points into higher dimensional space, called feature space F (an element of Hilbert space). SVM then simply define an linear hyperplane P in this space [20]. This mapping to higher order space is achieved implicitly using a function Φ , implied by a *kernel* function (Figure 2.1). A kernel function represents the relationship between two input data as a real number:

$$K : R^n \times R^n \rightarrow R \tag{2.2}$$

For example, the simple dot product kernel is defined as:

$$K(x, y) = (\Phi(x) \cdot \Phi(y)) \tag{2.3}$$

As F is not represented explicitly, feature space can theoretically have an infinite dimension. Kernel functions must be symmetric, and satisfy a general positivity constraint: Mercer’s theorem.

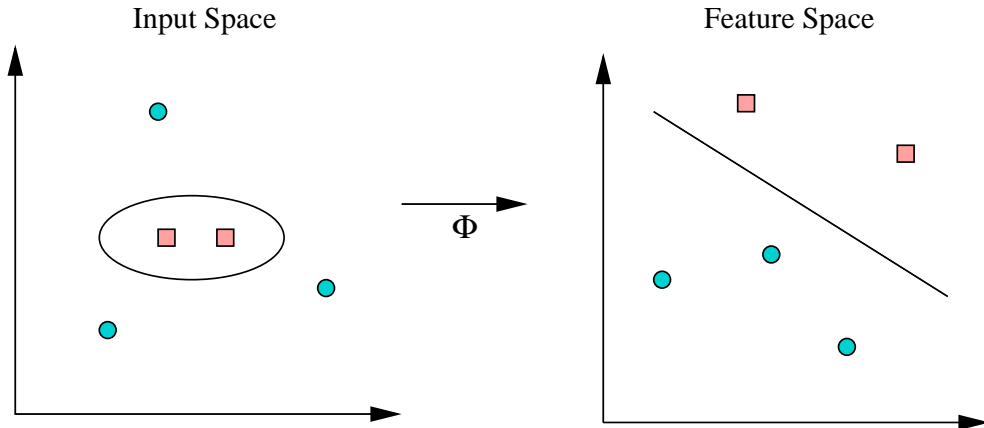


Figure 2.1: Non-linear mapping from input space to feature space via the function Φ (implied by the kernel function). An optimal linear hyperplane is determined in feature space, resulting in a corresponding non-linear decision boundary in input space.

The class of hyperplanes are those satisfying the constraint:

$$(w \cdot x) + b = 0, \tag{2.4}$$

where $w \in R^n$ (vector normal to the hyperplane), and $b \in R$ (the minimum distance between the training data and the hyperplane: *margin*). The corresponding decision functions are therefore:

$$f(x) = \text{sign}((w \cdot x) + b) \tag{2.5}$$

The hyperplane, P is chosen to maximise the margin. Thus P is often referred to as the maximal marginal hyperplane,

$$\max_{w,b} \min\{\|x - x_i\| : x \in R^n, (w \cdot x) + b = 0, i = 1, \dots, \ell\} \tag{2.6}$$

where ℓ is the number of data points (see Figure 2.2).

Support vectors are the data points found on the margin, and are used to represent the hyperplane implicitly (see Figure 2.3). The solution therefore consists of a subset of the training data, condensing the information available in all training examples.

Noisy data can introduce training errors, and a separating hyperplane may not exist due to the overlap in areas. Accuracy can be sacrificed however in order to gain better overall predictive power [15]. *Slack variables* can be introduced into

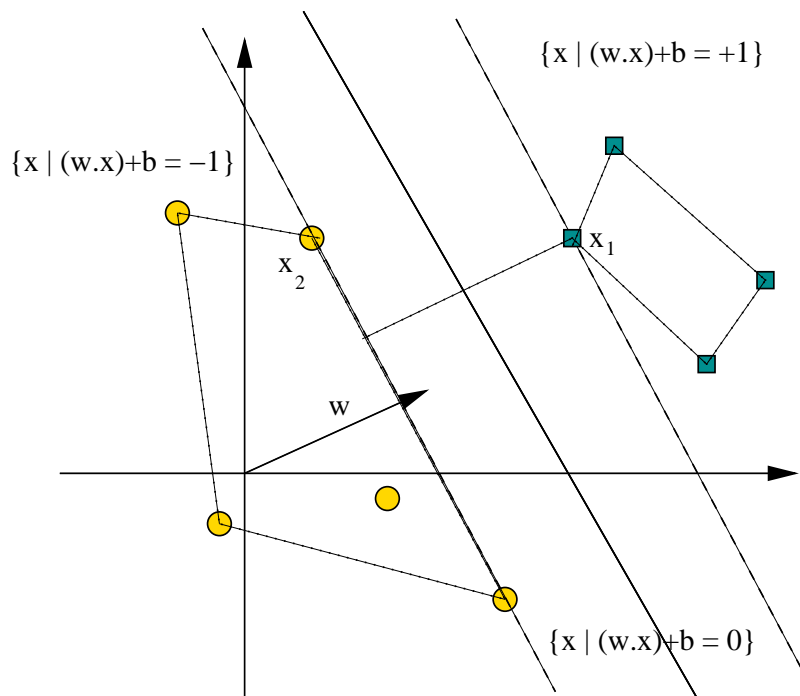


Figure 2.2: An example of a hyperplane separating two classes.

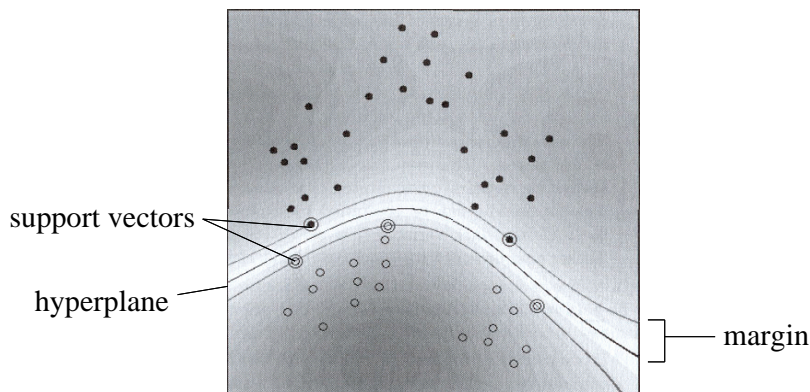


Figure 2.3: Two classes separated by an oriented hyperplane in input space. Support vectors are data points closest to the hyperplane (on the margin) that represent the hyperplane implicitly in input space.

the optimisation equation which allow for the possibility of examples violating the hyperplane barrier [26]. The resulting margin accomodates noisy data and is termed a *soft margin*. One technique for introducing a soft margin is the use of a diagonal factor d , a constant, applied to the kernel function during the training phase:

$$K \leftarrow K + dI \quad (2.7)$$

where I is the identity matrix (the original kernel function is still used during classification and testing) [15].

Statistical learning theory or VC (Vapnik-Chervonenkis) theory, illustrates the importance of choosing a function f (defined implicitly by the oriented hyperplane) such that the risk (test error), is minimised [26]. SVM therefore involve solving an optimisation problem: minimising the upper bound of the risk, achieved by maximising the margin between the separating hyperplane and the data [4]. During training, this equates to solving a constrained quadratic programming problem, guaranteeing that SVM will determine the global maximum.

2.1.2 Support Vector Machine Development and Application

SVM have performed well in multiple areas of biological analysis, including the recognition of translation initiation sites (TIS) [31], analysis of microarray gene expression data [9], prediction of protein-protein interactions [7], and multi-class protein fold recognition [22].

The n features selected for biological data (whose numerical values are used to

construct the n dimensional vector for input space) will affect the performance of SVM. Both Bock et al. [7] and Ding et al. [22] applied SVM to biological analysis of proteins using features of the amino acids in the protein sequences. Bock et al. applied SVM to the prediction of protein-protein interactions, classifying four out of five potential interactions correctly. Bock et al. employed a different approach than those of earlier protein-protein interaction prediction algorithms, concentrating entirely on the “primary structure and associated physiochemical properties” of the proteins during feature selection. Similarly, Ding et al. applied SVM to multi-class protein fold recognition, using feature sets based upon the physiochemical properties of amino acids in the protein sequence. Ding et al. illustrated SVM’s superiority to artificial neural networks (ANN) over a number of different protein-fold datasets.

SVM are able to determine the features that significantly contribute to the classification of a datum point. Furey et al. [15] applied SVM to the classification and validation of microarray expression data, selecting features based upon weights implied by SVM. Furey et al. however, concluded that the use of the highest ranked features alone did not affect the performance of the SVM, and that new methods of feature selection were required. Such work has since been conducted by Chow et al. [27] who used *feature relevance experts* to identify significant features in a feature set (though these feature subsets did not effect the SVM’s classification performance). Thus Furey et al. and Chow et al. both concluded that the use of weak features (in addition to strong features) did not alter SVM performance.

However, Barzilay et al. [6] (prior to Chow et al.’s research), concluded that the use of a large number of weak features reduced the performance of SVM. Barzilay et al. created a synthetic database of normally distributed vectors with a linear decision boundary. The classification performance of SVM with relatively small numbers of features (10–50) outperformed those trained on larger numbers of features (300–900). Whether or not the use of a large number of features will impact on the performance of the SVM remains uncertain, though apparently it depends on the nature of the dataset and the specific SVM implementation. Regardless, SVM’s ability to identify deterministic features remains an important characteristic, especially to biological analysis.

Brown et al. [9] applied SVM to the analysis of microarray gene expression data, comparing the performance of SVM to four other classifiers; decision trees (two kinds), Parzen windows, and Fisher’s linear discriminant. Parzen windows can be regarded as a generalisation of k-nearest neighbour techniques and are closely related to SVM, also utilising a decision function based on a kernel function. Fisher’s linear discriminant is similar to SVM in that it involves a maximization

sation of the variance between classes. Fisher’s linear discriminant does not map to higher order space however, but instead onto a line, applying a classification function in this one-dimensional space. Brown et al. showed that SVM were superior to these four classifiers over a number of different datasets.

Whilst comparing different classifiers, Brown et al. also compared the performance of alternative kernel functions. Popular kernel functions to date include the dot product, radial basis function (RBF), and polynomial kernel functions. Brown et al. concluded that the choice of kernel function altered the performance of the SVM. Zien et al. [31], who applied SVM to the recognition of translation initiation sites (TIS), found that performance was affected by the choice of kernel function. Recognition of TIS by SVM using a simple polynomial kernel outperformed the existing TIS recognition approaches, namely: ANN and the Salzberg method. While all SVM classifiers outperformed these approaches, the kernels applied can be considered naive as they do not incorporate any domain knowledge. Zien et al. developed an engineered kernel that incorporated prior biological knowledge that performed the best of all kernel functions and 26% better than existing approaches.

One of the engineered kernels that Zien et al. developed considered the dependencies between distant data points to be less or non existent. The concept of creating a ‘local’, or ‘neighbourhood’ kernel was also explored by Brailovsky et al. [8], and Barzilay et al. [6]. Both Brailovsky et al. and Barzilay et al. illustrated the increased performance of such local kernels compared to other ‘naive’ kernels.

Brailovsky et al. [8] combined global and local kernel approaches to create an engineered kernel function $K^*(u, v)$ which classifies data based on those points close to it (locality) while still providing a general classification rule for new, unseen data. Brailovsky et al. outlined the equation for such a kernel, based upon a global kernel $K(u, v)$:

$$K^*(u, v) = K(u, v) \times n(u, v) \tag{2.8}$$

where $n(u, v)$ is the number of windows (spheres in input space that may include any number of input data) including both u and v . Note that if windows are not overlapping, and cover the entire domain (input) space, this kernel will be equivalent to the global kernel $K(u, v)$. It is also worth noting that the engineered kernel can be based upon any global kernel.

A number of algorithms for window selection were discussed by Brailovsky et al. including the use of the *Infi-K-TSS* algorithm. The *Infi-K-TSS* algorithm calculates windows such that each window, anchored at a TSS (training sample set) has a radius such that it includes at least k neighbours. One additional win-

dow is also required (an ‘overall window’) which includes all input data, ensuring that all data points are considered. Letting k equal the size of TSS, results in the $K * (u, v)$ kernel reducing to the global kernel $K(u, v)$, multiplied by the TSS size. The selection of the parameter k was not discussed by Brailovsky.

2.1.3 Support Vector Machine Availability

A number of SVM implementations are available to download via the Internet. The SVM software utilised by Furey [15] was written by Professor Grundy of Columbia University. This SVM implementation enables the user to define a number of parameters, as well as the choice of a kernel function, including: dot product; radial basis function (RBF); and varying degree polynomial kernels.

2.2 Biological Research

The current, predominant method of determining an immune response between different tissue samples is Tissue Serology; a test that detects the presence of antibodies to a particular antigen. This method is only 10–15% successful since the antibodies used are often unable to discriminate between closely related allelic products. DNA typing is far more reliable. Elution studies and other assays can be used to determine peptide binding, but these experiments are very costly and time consuming. Recently however, a number of approaches for MHC-peptide binding prediction have become available which are briefly outlined in the following sections.

2.2.1 Peptide Motifs

Rammensee et al. [24] developed *peptide motifs* as a means to predict T-cell epitopes within proteins of known sequences, and analyse MHC-peptide interaction. This approach involves searching a database for possible binding peptides based upon known anchor residues and preferred residues for a given allele in any number of the peptides’ positions. This list of possible candidates is then screened to exclude known weak binders and, if possible, candidates are screened for non-anchor residues detrimental or optimal for binding. This results in a list of example ligands or T-cell epitopes which are predicted to bind to a given allele.

2.2.2 Prediction Algorithms

Andersen et al. [5] compared the performance of peptide motifs to two computer algorithms, namely: the KP-algorithm (BIMAS [21]), and the HGR-algorithm (SYFPEITHI [23]). The KP-algorithm predicts HLA-peptide binding by ranking potential peptides based on a predicted half-time of dissociation to HLA class I molecules. The HGR-algorithm is based upon motif scanning of primary anchor residues. Andersen et al. concluded that there was no strong correlation between the actual and predicted binding of these two algorithms, neither of which outperformed peptide motifs.

2.2.3 Binding Matrices

Polymorphic residues (amino acids found in the polymorphic positions) contribute to the variation in cleft shape, and hence binding affinity of alleles to a given peptide. Pocket specificity can therefore be represented functionally by “substituting the corresponding peptide position with all natural amino acid residues and quantifying their effects on binding [“pocket profiles”]” [28]. A binding matrix for a given allele is then created which is formed from the sum of all pocket profiles. Prediction of binding affinity of a given peptide to the allele can then be performed based upon mathematical processing of a peptide sequence (utilising the binding matrix). The determination of such a binding matrix however, requires hundred of individual peptides and hence thousands of peptide binding assays.

2.2.4 Artificial Neural Network

Brusic et al. [10] applied ANN to the prediction of MHC-peptide binding in 1994. A database of 4000 peptide sequences was compiled. The peptide sequences were known to bind to HLA-A2 (human) or H-2Kb (mouse) MHC molecules, whereas non-binding examples were predicted. A fully-connected 3-layer back-propagation network correctly classified 78% of HLA-A2 and 88% of H-2Kb MHC molecules.

2.3 Biological Resources

2.3.1 MHCPEP Database

Since Brusica et al.'s ANN work, a large, comprehensive database of over 13,000 MHC binding peptides has been constructed, namely: MHCPEP [11]. MHCPEP is available in a flat-file format, allowing information retrieval via simple string comprehension.

2.3.2 AAindex Database

The AAindex [18] contains amino acid indices and mutation matrices. There are 20 kinds of amino acids, so each amino acid index contains 20 numerical values that represent a biochemical or physiochemical measure for each of the amino acids. Such measurements include size, charge, weight, propensity to appear in a helix, and affinity for water.

2.3.3 HLA Informatics Group

The Anthony Nolan Bone Marrow Trust oversees a number of research groups including the HLA Informatics Group [2]. The HLA Informatics Group funds the HLA Sequencing Data (available at [3]), and the IMGT/HLA database [25].

Constructing MHCPEP_*R*

In order to predict MHC-peptide binding, both binding and non-binding examples are required. As MHCPEP contained only binding examples, the non-binding examples were determined using two approaches. The first approach, data collection, involved the compilation of experimental findings published in a variety of journals. This resulted in 1170 entries that were compiled into the database. The second approach applied a prediction algorithm (based upon the canonical binding model) to the existing peptide motifs and binding data.

The additional entries were compiled with the original entries from MHCPEP, creating a supplementary database of MHC-peptide entries, namely: MHCPEP Revised (MHCPEP_*R*). MHCPEP_*R* consists of 13432 original MHCPEP entries, 1170 additional experimental entries, and 2000 predicted non-binding entries. Additional entries in MHCPEP_*R* are consistent with the original MHCPEP entry format. All of the additional entries consisted of HLA-A (580), HLA-B (568), and HLA-C (22) alleles. The MHCPEP_*R* database was converted to XML format and users can access the database via a web interface (see Chapter 5) that also allows for online submission of additional entries.

3.1 Data Collection

A literature search was conducted and 1170 experimental entries were compiled (464 of which were non-binding entries). Compiling these examples proved challenging, as six binding measurements exist. Further, three of these measurements are relative to peptides in the study, while researchers are currently disputing the degree of binding illustrated by the other three measurements. Though an attempt to normalise the three independent measurements was made, no correlation was determined.

In order to provide additional functionality to the database, and prevent the loss of precision, a number of additional entry fields were included, namely:

‘RAW’, ‘RES BINDING’, and ‘MHC UPDATE’. The existing ‘METHOD’ field stores the binding measurement, while ‘RAW’ contains the corresponding raw data measure (if available). As researchers dispute the degree of binding represented by the binding measurements, the field ‘RES BINDING’ contains the researcher’s (who reported the entry) determination of binding affinity. The ‘MHC UPDATE’ field was included to allow users to cross reference new entries with the original entries (by recording the unique entry identification numbers of the corresponding original entries). Of the 1170 additional entries, 224 referenced original MHCPEP entries.

3.2 Prediction Algorithm

3.2.1 Amino Acid Similarity Measure

There are twenty types of amino acids, each represented by a single letter. Experimental and theoretical research have characterised different properties of these amino acids, representing these properties in numerical indices. The AAindex [18] contains over 400 amino acid indices. Nakai et al. [19] concluded that the most deterministic features for amino acids were size, hydrophobicity, polarity, refractive index, and beta propensity term. Therefore amino acid indices for each of these features were taken from AAindex with accession numbers: DAWD720101, ARGP820101, GRAR740102, MCMT640101, and KIMC930101 respectively.

Each of the twenty amino acids were mapped to five dimensional space (using the 5 feature measures), where the euclidean (straight line) distance between any pair of amino acids represented a measure of the pair’s ‘similarity’.

Alleles binding to amino acids R, E, D, or K in any pocket will tend not to bind to amino acids F, I, L, M, or V in this pocket. The minimum euclidean distance between any pair between these subsets equaled 32.21, therefore 30 was considered the ‘cut-off’ value for this similarity measure. A matrix, *AAmatrix30* was determined representing distances between any pair of amino acids (where distances greater than 30 represent ‘different’ amino acids).

3.2.2 Non-Binding Prediction

The prediction algorithm determined *complementary* allele pairs; in which binding examples for one allele could be used as non-binding examples for the other allele. Complementary pairs were determined based upon the canonical binding

model; so that the pairs exhibited ‘different’ (determined by AAmatrix30) anchor residues for pockets 2 and 9.

Anchor residues of alleles for pockets 2 and 9 were taken from Chelvanayagam [12]. Two sets of alleles were formed for each pocket, where sets contained alleles with ‘similar’ (determined by AAmatrix30) anchor residues for that pocket. Allele pairs were considered to be complementary if they appeared in opposite subsets for both pockets 2 and 9.

For each complementary pair, the binding peptides for each allele were predicted as non-binding examples for the other allele. That is, if a given peptide binds to one of these alleles, it is unlikely to canonically bind to the other allele. This resulted in 4213 predicted non-binding examples.

3.2.3 Prediction Analysis

The predicted examples were cross referenced against the known experimental examples. Of the 4213 predicted entries, 14 were represented in the experimental data and surprisingly, only half (7) were correct. However, this does not suggest the prediction algorithm was only 50% correct. Peptides that would bind to a particular allele form a small proportion of available peptides, therefore this increases the probability that a predicted non-binding example is correct.

In order to reduce the perceived error of the prediction algorithm a simple analysis of the source entries (the entries used to predict the non-binding examples) for each of the incorrect cases was conducted. In each case, the peptide residues in positions 2 and 9 of the source allele did not conform to the anchor residues reported for the allele in pocket 2 and 9 respectively. For example the peptide FRYNGLIHR was reported to bind to A*0302, however anchor residues for A*0302 are reported as V/L/M and K, for pockets 2 and 9 respectively. Though R is considered to be ‘similar’ to K, R is considered ‘different’ from V/L/M. Thus either positions 2 and 9 are not anchor positions for A*0302 or B*2705, or one anchor is sufficient to bind. The referenced articles for each incorrect entry confirmed that the entry was correctly recorded in MHCPEP_R.

All entries for each HLA allele were thus analysed to determine whether or not the amino acids in positions 2 and 9 in the peptide were ‘similar’ for each class set (binding and non-binding). Alleles that illustrated inconsistencies were removed from the prediction algorithm subsets, namely: A*0204, A*0302, A*1101, A*2602, A*3303, B*2701, B*2702, B*2703, B*2704, B*2705, and B*2706.

The prediction algorithm was repeated, resulting in 2000 unique predicted non-binding examples which were included in MHCPEP_R . These predicted

peptides are analysed further in section 4.2.4.

Peptide binding prediction with Support Vector Machine

4.1 Performance Measures

The performance of a SVM can be measured in a number of ways. A standard basis for measurement however, involves the classification of test input into one of the four following categories:

- TP – true positive; both the MHCPEP_*R* and the SVM classed the peptide as binding,
- TN – true negative; both classed the peptide as non-binding,
- FP – false positive; classed as binding by the SVM and non-binding by MHCPEP_*R*, and
- FN – false negative; classed as non-binding by the SVM and binding by MHCPEP_*R* [9].

The number in each category for leave-one-out-cross-validation (LOOCV) or ‘jack-knife’ testing on each datum can then be analysed in a number of ways to measure performance. The following performance measures were recorded:

1. Sensitivity [10, 31]; defined as the rate of correctly predicted positive data (true positives) to all positive data,

$$\textit{Sensitivity} = TP/(TP + FN) \quad (4.1)$$

2. Specificity [10, 31]; the rate of correctly predicted negative data to all predicted negative data.

$$\textit{Specificity} = TN/(TN + FP) \quad (4.2)$$

3. Predictive binding [10]; represents the probability that a predicted positive (binding) example is in fact positive.

$$Predictive(binding) = TP/(TP + FP) \quad (4.3)$$

4. Predictive non-binding [10]; represents the probability that a predicted negative (binding) example is in fact negative.

$$Predictive(non - binding) = TN/(TN + FN) \quad (4.4)$$

5. Overall; represents the probability that a predicted example will be correctly classified, expressed in terms of the total number of examples (TOTAL).

$$Overall = (TN + TP)/(TOTAL) \quad (4.5)$$

4.2 Experimental Results and Discussion

Bock et al. [7] found that protein-protein interactions could be accurately predicted by SVM using the structure and physiochemical properties of the proteins alone. Therefore MHC-peptide binding prediction was based upon selection of features that represented such properties for both the peptide and allele.

SVM prediction of peptide binding for 25 individual alleles was conducted. Section 4.2.1 outlines the relative performance of a number of kernel functions, namely: dot product; RBF; varying degree polynomial; and an engineered kernel function. Peptide features alone were able to differentiate between binding and non-binding data as allele features remained constant.

Bock et al. however, predicted protein-protein interaction using features that represented the structural and physiochemical properties of both proteins. Experimentation on individual alleles failed to take into account the structure and properties of the allele. Therefore experimentation involving groups of alleles (in which subtypes were combined) was conducted which required additional features to represent allelic variation. These experimental results are discussed in Section 4.2.2.

Previous work in SVM application considered the effect of weak features on classification performance. Both Chow et al. [27] and Furey et al. [15] concluded that the selection of dominant features (conversely, the inclusion of weak features) did not affect classification performance. Barzilay et al. however used a synthetic database of normalised vectors to demonstrate that large numbers of weak features decrease SVM performance. Therefore the effect of weak features

on SVM performance appeared to be data dependent. Section 4.2.3 discusses whether or not such a large feature set, or the presence of weak features in this set will affect performance of SVM when predicting MHC-peptide binding.

Section 4.2.4 discusses the quality of data available in MHCPEP_*R* and hence the process of identifying and analysing possible outliers in the data. The prediction algorithm used (see Section 3.2) is also analysed using trained SVM to classify a number of predicted non-binding class sets (providing a crude approximation to the algorithm's performance).

When training the SVM, Chow et al. [27] found that the selection of training examples affected the performance of the SVM. The best performance was achieved through the use of 'leave-one-out-cross-validation tests' (LOOCV tests) during training. This involves training the SVM with all but one of the test inputs, which is then classified by the SVM. All experimentation used LOOCV testing, the performance for each SVM overall was measured as outlined in Section 4.1.

In order to accommodate noise in the data, a soft margin was used for all SVM via options available in the SVM implementation.

4.2.1 Kernel Experimentation

SVM performance largely depends on the choice of kernel function. Although there are no current theories concerning how to choose a suitable kernel function, the use of engineered local kernels, (that are data dependent) have been shown to perform well [6, 8, 31]. The SVM implementation we used allowed the use of dot product, radial basis function (RBF), and varying degree polynomial kernels. Utilising parameters available in the SVM implementation it was also possible to implement the Infi-K-TSS algorithm outlined by Brailovsky et al. [8] (see Section 2.1.2).

Initially, experimentation involving individual alleles only was conducted. For each allele, experimentation was conducted using the dot, RBF, polynomial (degree 2–6), and the engineered kernel. Experimentation was confined to the individual alleles for which both binding and non-binding data were available in MHCPEP_*R*. Note however, predicted non-binding entries were not used as the performance of the prediction algorithm was unknown at this stage. 25 alleles were selected, namely: A*0101, A*0201, A*0202, A*0203, A*0204, A*0205, A*0206, A*0302, A*1101, A*2601, A*2602, A*2603, A*2604, A*2605, A*2606, A*3303, B*0702, B*0801, B*2701, B*2702, B*2703, B*2704, B*2705, B*2706, B*3501, B*3502, B*3503, and B*5401. Nine-mer peptides (peptides of length nine) only were considered (due to the canonical binding model) and reported

Allele	Kernel	TP ^a	FP ^b	TN ^c	FN ^d	Sens ^e	Spec ^f	P(B) ^g	P(NB) ^h	Overall ⁱ
A*0101	DOT	51	16	23	10	0.8361	0.5897	0.7612	0.6970	0.74
A*0101	RBF	57	16	23	4	0.9344	0.5897	0.7808	0.8519	0.80
A*0101	POLY(2)	41	14	25	20	0.6721	0.6410	0.7455	0.5556	0.66
A*0101	POLY(3)	52	12	27	9	0.8525	0.6923	0.8125	0.75	0.79
A*0101	POLY(4)	50	11	28	11	0.8197	0.7179	0.8197	0.7179	0.78
A*0101	POLY(5)	50	15	24	11	0.8197	0.6154	0.7692	0.6857	0.74
A*0101	POLY(6)	51	15	24	10	0.8361	0.6154	0.7727	0.7059	0.75

Table 4.1: SVM performance of A*0101 allele.^aNumber of correctly classified binding entries. ^bNumber of incorrectly classified non-binding entries. ^cNumber of correctly classified non-binding entries. ^dNumber of incorrectly classified binding entries. ^eSensitivity; the proportion of correctly classified binding examples. ^fSpecificity; the proportion of correctly classified non-binding examples. ^gPredictive (Binding); the probability that an entry classified by SVM as binding is correct. ^hPredictive (Non-Binding); the probability that an entry classified by SVM as non-binding is correct. ⁱOverall; the proportion of correctly classified entries.

peptide sequences in which one or more residues were unknown were not considered.

The performance of each SVM was analysed using the measures outlined in section 4.1. Although ‘Overall’ performance was calculated, this measure is not as important as the sensitivity, specificity, predictive binding and predictive non-binding measures. All four of these measures need to be considered when comparing the performance of two SVM. For example, Table 4.1 illustrates the performance of SVM when applied to the A*0101 allele, the polynomial (degree 4) kernel illustrated the highest performance (taken as being the minimum of all four measures).

The SVM features for these individual alleles did not include the allele’s features (as these features remained constant) and therefore peptide features were sufficient to differentiate between entries. For each nine-mer peptide, the physiochemical and structural properties of the peptide were represented by the physiochemical and structural properties of each amino acid in the peptide sequence. Five amino acid measures were used for each amino acid that were outlined by Nakai et al. [19] (discussed in section 3.2.1). This resulted in 45 features selected for each entry.

Based upon these features, 9 of the 25 alleles were considered to perform well. The performance of these 9 alleles is shown in Table 4.2. Generally higher order

Allele	Kernel	TP	FP	TN	FN	Sens	Spec	P(B)	P(NB)	Overall [†]
A*0101	POLY(4)	50	11	28	11	0.8197	0.7179	0.8197	0.7179	0.78
A*2601	POLY(6)	16	6	26	5	0.7619	0.8125	0.7273	0.8387	0.7925
A*2602	POLY(5)	5	1	14	3	0.625	0.9333	0.8333	0.8235	0.8261
B*0801	DOT POLY(2-6)	11	0	4	0	1.0	1.0	1.0	1.0	1.0
B*2702	DOT POLY(3-6)	13	1	4	0	1.0	0.8	0.9286	1.0	0.9444
B*2704	RBF POLY(3-6)	15	1	10	1	0.9375	0.9090	0.9090	0.9375	0.9090
B*2705	RBF	89	5	23	6	0.9368	0.8214	0.9468	0.7931	0.9106
B*2706	POLY(3)	16	4	11	4	0.8	0.7333	0.8	0.7333	0.7714
B*3503	RBF	5	1	6	1	0.8333	0.8571	0.8333	0.8571	0.8462

Table 4.2: Individual alleles that illustrated high prediction accuracy.[†]Table fields as per Table 4.1.

kernels (RBF or polynomial kernel of degree greater than 3) performed the best of all kernels applied. The polynomial kernel however was considered to be the best performing kernel in this application due to the RBF kernel’s sensitivity to uneven distribution of data and noise in the class sets.

Although 9 of the 25 alleles performed well, this does not alone suggest that SVM’s can accurately predict MHC-peptide binding based upon the features selected. Therefore all 25 allele class sets were analysed to determine whether or not the data suggested a trend which would result in these 9 alleles performing well. For each allele, the proportion of binding examples was determined, as well as the proportion of noise present in each class set (binding and non-binding). The level of noise present in class sets was approximated by the number of suspected outliers (determined in Section 4.2.4). For each of the 25 alleles, SVM performance was dependent on:

- the distribution of entries in class sets; and
- the proportion of noise present in class sets.

If the distribution of entries was relatively even (that is, there was a relatively even number of binding and non-binding entries) then SVM performance was high. However, uneven distribution of data caused SVM to misclassify entries of the smaller class set as members of the larger class set. This *data skewing* was also observed by Zien et al. [31]. Further, if the proportion of noise was high in

a class set, this also caused the data to skew. For the majority of alleles, the proportion of binding examples was high, so that noise present in the non-binding class resulted in SVM skewing data towards the larger, binding class set.

Based upon the above findings, it is reasonable to conclude that the peptide features selected are strong features that are able to represent the structural and physiochemical properties of the peptide that result in individual allele binding preferences. Therefore if quality data is available (that is, an even distribution of class sets that contain small proportions of noise), accurate prediction of MHC-peptide binding is possible for individual alleles.

Peptide sequences for those entries that were ‘consistently’ misclassified were analysed (see Section 4.2.4). A number of these peptides sequences exhibited amino acids in positions 2 and 9 that did not conform to the reported anchor residues for pockets 2 and 9 of the allele. As human inspection of the peptide sequence also identified these questionable entries, SVM misclassification of these entries could be considered ‘reasonable errors’.

Brown et al [9] suggested that data skewing could be corrected via the use of a diagonal factor (that is, experimentation involving the soft margin). Experimentation on A*0302, B*3501, and B*5401 alleles was conducted using diagonal factors ranging from 5 to 30. Although performance increased in some specific cases, for other cases the performance decreased. Therefore the diagonal factor can be applied to specific domain applications when ‘fine-tuning’ a SVM, though no general rule could be determined.

4.2.2 Generalisation Performance

Experiments considered thus far utilised peptide features only to represent the physiochemical and structural properties that determine binding affinity of a particular MHC-peptide pair. However, Bock et al. [7] predicted protein-protein interactions, representing the structural and physiochemical properties of both proteins. Therefore, we suspected that additional information was represented by allelic variation that would increase SVM performance.

Polymorphic positions in the alleles were used to represent the structure and physiochemical properties of the alleles. The polymorphic positions for HLA alleles were determined using a simple program which accessed sequence alignments for all known HLA-A, HLA-B, and HLA-C alleles extracted from HLA Sequencing Data (see Section 2.3.3). This program identified 67 polymorphic positions for HLA alleles between positions 1–180 (the positions responsible for alpha-1 and alpha-2 domains and hence binding preferences).

Allele	Kernel	TP	FP	TN	FN	Sens	Spec	P(B)	P(NB)	Overall [†]
A*020(1-6)	POLY(6)	549	22	27	13	0.9767	0.5510	0.9615	0.675	0.9427
A*260(1-3)	POLY(6)	24	8	59	8	0.75	0.8806	0.75	0.8806	0.8384
B*270(1-6)	POLY(15)	169	10	61	8	0.9548	0.8592	0.9441	0.8841	0.9274
B*350(1-3)	RBF	165	27	23	5	0.9706	0.46	0.8594	0.8214	0.8545

Table 4.3: Groups of alleles that illustrated high prediction accuracy.[†]Table fields as per Table 4.1.

The amino acids for each HLA allele at these polymorphic positions were determined from protein sequences taken from the IMGT database (see Section 2.3.3). Feature selection was similar to that used for peptide features, thus five amino acid measures were used to represent each of the polymorphic positions of the allele, resulting in 335 features identified for each allele (and hence 380 features identified for each MHC-peptide entry).

Experiments were conducted for groups of alleles, combining a number of subtypes for each type present in MHCPEP_*R*. Allele groups were formed for the A*02 type (A*0201-6), A*26 type (A*2601-3), B*27 type (B*2701-6), and the B*35 type (B*3501-3). Table 4.3 contains the best performing kernels for each group.

In all cases the higher order kernels, RBF and polynomial (degree 6) were the best performers. However, SVM application to individual alleles (based upon peptide features alone) required polynomial kernel of degree greater than 3. Therefore higher order input spaces appear to require higher order kernels to achieve a reasonable level of classification performance.

The data from each group was divided into the individual subtypes to determine the performance of the individual alleles in the group. For all groups the poor performers of the group experienced an increase in performance (when combined in the group) while not significantly affecting the better performers. This ‘redistribution’ of performance can be attributed to the reduction in noise levels for the allele once combined in the group relative to the individuals.

A number of individuals in the A*02, A*26, and B*35 group contained large proportions of noise, whereas each of the B*27 alleles performed well individually (apart from B*2703). For the B*27 group each allele experienced a significant increase in performance as illustrated in Figure 4.1. As it was observed that higher order kernels were required for higher order input spaces, additional experiments for the B*27 group of alleles were conducted and the highest performance was achieved for a polynomial kernel of degree 15. The significant increases in perfor-

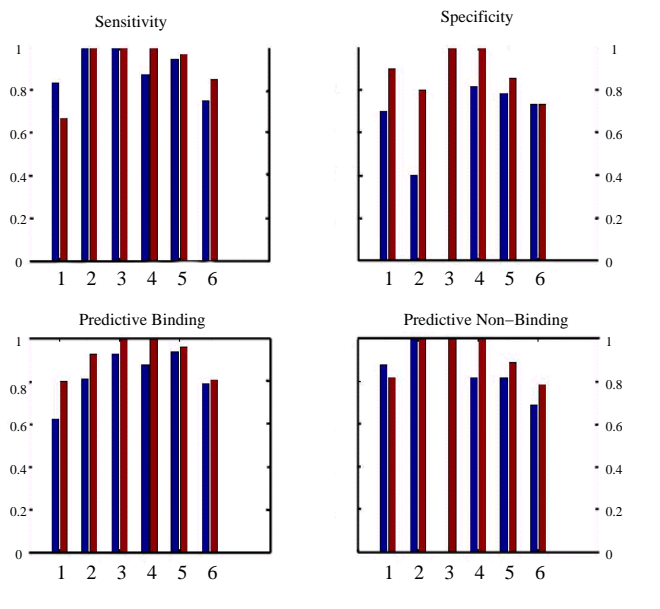


Figure 4.1: The B*27 group of alleles performance. Performance for each individual allele in the group (B*2701–6) were determined (subtype number appears on x-axis). The red bars represent the performance of individual alleles in the group, while the blue bars represent the performance of the individual alleles (using peptide features only). Almost all alleles increased in performance (over all performance measures) except B*2701. Although B*2701’s Sensitivity and Predictive(NB) decreased, this is more than compensated for by the increase in both Sensitivity and Predictive(B). B*2703 showed a marked increase in prediction ability.

mance exhibited by all alleles in the B*27 group implies that there is additional information available in the allele’s structure and physiochemical properties.

Further, the B*2703 allele possessed 27 binding and 2 non-binding examples resulting in the 2 non-binding examples being misclassified during experimentation involving peptide features only. However, the B*27 group SVM correctly predicted all B*2703 entries. Therefore the information available in the other alleles non-binding data appears to have provided the necessary information required to classify these 2 non-binding entries. A number of alleles are well studied and possess large data sets. These alleles could therefore benefit SVM prediction of those alleles that possess smaller data sets.

Brusic et al [10] applied ANN to the prediction of MHC-peptide binding. The performance of ANN compared to that of SVM, for the A*02 group of alleles,

Classifier	TP	FP	TN	FN	Sens	Spec	P(B)	P(NB)	Overall [†]
ANN	44	12	56	16	0.7333	0.8235	0.7857	0.8235	0.7813
SVM	165	27	23	5	0.9706	0.46	0.8594	0.8214	0.9427

Table 4.4: A comparison of SVM and ANN classifiers.[†]Table fields as per Table 4.1.

is illustrated in Table 4.4. Although Brusica et al.’s ANN outperformed SVM in this case, it is unlikely that this is indicative of ANN superiority to MHC-peptide binding prediction. Uneven distribution of data and the presence of noise significantly affects classification performance of SVM and the majority of the individual alleles in the A*02 group performed poorly (using peptide features only) due to these factors. Therefore, SVM performance when applied to the A*02 group is not indicative of the performance of SVM when applied to MHC-peptide binding prediction.

4.2.3 Feature Selection

SVM have illustrated the ability, through feature relevance experts [27], to identify highly ranked features. However, whether or not the inclusion of the weak features will affect SVM performance appeared to be data dependent. [27, 15, 6, 22]. Barzilay et al [6] found that a large number of features decreased SVM performance [6]. Furey et al. [15] and Chow et al [27] however, found that SVM performance was not affected by large feature sets, or the inclusion of weak features.

Experimentation was therefore conducted in order to ascertain whether or not the inclusion of weak features would affect the performance of SVM applied to MHC-peptide binding prediction. Allele features were included as features for an individual allele’s training set (as they would not contribute to the classification of data).

Experimentation was conducted on each of the 25 individual alleles using the dot and RBF kernels. The inclusion of these weak features significantly affected SVM performance, using the dot product kernel, for 5 of the 25 alleles, illustrated in Table 4.5. Using peptide features only, these 5 alleles had performed reasonably well, while the other alleles had performed poorly. However, the inclusion of these weak features did not affect the performance of the RBF kernel for any of the 25 alleles. This supports the ability of higher order kernels to classify higher order input space.

Barzilay et al. [6] conducted experimentation on feature selection using both

Allele	Kernel	TP	FP	TN	FN	Sens	Spec	P(B)	P(NB)	Overall [†]
A*0101	DOT	51	16	23	10	0.8361	0.5897	0.7612	0.6970	0.74
	DOT*	37	14	25	24	0.6066	0.6410	0.7255	0.5102	0.62
A*0201	DOT	404	17	14	23	0.9461	0.4516	0.9596	0.3784	0.9127
	DOT*	244	11	20	183	0.5714	0.6452	0.9568	0.0985	0.5764
A*1101	DOT	85	12	7	17	0.8333	0.3684	0.8763	0.2917	0.7603
	DOT*	59	5	14	43	0.5784	0.7368	0.9219	0.2456	0.6033
A*2601	DOT	15	11	21	6	0.7143	0.6563	0.5769	0.7778	0.6792
	DOT*	8	13	19	13	0.3810	0.5937	0.3809	0.5937	0.5094
B*3501	DOT	140	18	15	21	0.8696	0.4545	0.8861	0.4167	0.7990
	DOT*	92	17	16	69	0.5714	0.4848	0.8440	0.19	0.5567

Table 4.5: Effect of weak features on SVM performance.[†]Table fields as per Table 4.1.* SVM trained using weak features.

dot and RBF kernels, finding that both kernels reduced in performance when applied to data containing a larger number of weak features. Therefore this would suggest that the order of the input space in this case was greater than that able to be classified by the dot or RBF kernel. During our application however, the order of the input space did not increase above that able to be classified by the RBF kernel.

4.2.4 MHCPEP_*R* Data Analysis

SVM have illustrated the ability to identify possible outliers; questionable data that may require verification. In order to identify possible outliers in MHCPEP_*R*, a subset of the data was analysed. Possible outliers were identified as those datum that were consistently misclassified by the dot, RBF, and polynomial (degree 2) kernels.

Of the 1620 data utilised by the 25 alleles during individual experimentation (see Section 4.2.1), 106 were identified as possible outliers. The majority of these outliers were non-binding examples, a result of the data skewing towards the larger, binding class set. Therefore, analysis of the possible outliers was constrained to the positive examples only. Further, as the entries reporting a low level of binding were most likely to be misclassified, only low binding datum were analysed. Table 4.6 illustrates those entries that were considered unusual upon inspection of the original entries in MHCPEP_*R*.

For example, entry RHUM136CD contained a peptide with residues R and

ENTRY ID ^a	Allele ^b	Peptide ^c
RHUM136CD	A*0101	FRDYVDRFY
RHUM13740	A*0302	FRYNGLIHR
HUM11429	B*2701	SRHKKLMFK
HUM1142A	B*2701	QRHGSKYLA
HUM1142D	B*2701	SRFSWGAEG

Table 4.6: Questionable MHCPEP_*R* entries. ^aThe unique entry identification number from MHCPEP_*R*. ^bAllele contained in the entry. ^cPeptide contained in the entry.

Y in positions 2 and 9 respectively, while reported residues are T or S, and Y respectively. Although the motifs need not be the same for binding to occur, a certain level of similarity is expected between the motif and residue. The R in position 2 of the peptide however will present a very large, positively charged side chain while T and S present small polar side chains. This example is therefore unusual as the shape and nature of pocket 2 for the allele would be unlikely to bind to both types of side chains. Further, experimentation with this entry may prove that the peptide does indeed bind to the A*0101 allele. In such a case, one anchor may have been sufficient to cause binding (non-canonical binding).

The quality of the predicted non-binding examples generated from the prediction algorithm (outlined in Section 3.2) was also considered. It is difficult to analyse the predicted non-binding entries as no entries occurred in MHCPEP_*R*. Therefore classification of the negative examples by trained SVM was considered a crude approximation of the algorithm’s performance. Analysis was thus limited to those alleles for which predicted non-binding data, and trained SVM were available (Section 4.2.1). Alleles A*0101, A*0205, and A*2601 were selected as all three alleles possessed 18 predicted non-binding examples.

The proportion of correctly classified negative examples is shown in Table 4.7. A high proportion of predicted negative examples was correctly classified for alleles A*0205 and A*2601. For these alleles it would appear that the prediction algorithm has performed well (while performing poorly for A*0101).

Such varied performance can be attributed to the nature of the algorithm. The prediction algorithm was based upon primary anchor positions in pockets 2 and 9 (the canonical binding model). For alleles with these anchor positions, the algorithm should therefore perform well. As the algorithm is based upon the canonical binding model it should perform well for the majority of alleles in the database. However, for individual allele predictions, application of the algorithm should be based upon the residues of that allele in order to guarantee a good level

Allele	Kernel	Sens	Spec	P(B)	P(NB)	NEG ^{†*}
A*0101	POLY(4)	0.8197	0.7179	0.8197	0.7179	0
A*0205	POLY(6)	1	0.2857	0.8780	1	0.8889
A*2601	POLY(6)	0.7619	0.8125	0.7273	0.8387	1

Table 4.7: Analysis of non-binding prediction algorithm. [†]Fields as per Table 4.1. **NEG* field contains the proportion of predicted non-binding entries classified as non-binding by SVM.

of performance. Further, as discussed in Section 3.2, there is a high proportion of non-binding peptides. Therefore, this increases the probability of correctly predicting non-binding peptides.

CHAPTER 5

Web Interface

5.1 MHCPEP_*R*

The MHCPEP_*R* database is available online [30] to download or query. Queries available include peptide searches and/or allele searches of the database. More advanced queries are also available, including an option to create a specific flat-file database sent to the email address specified by the user.

5.2 Support Vector Machines

The web interface provides access to the nine alleles' SVM that performed well (see Section 4.2.1), and to the B*27 group's SVM that outperformed the individual B*27 alleles [29]. The SVM classify a given MHC-peptide pair; returning a discrimination value of this classification (measure of certainty), and report the performance of the trained SVM.

5.3 Data Deposition

The web interface for MHCPEP_*R* also provides a form for adding new entries to the database.

Conclusions and Future Work

This work involved the first application of SVM to MHC-peptide binding prediction and we have shown it to be an effective tool for this area of biological analysis. For alleles where there is a balanced data set (that is not over burdened with noise) SVM illustrate high prediction accuracy. In particular the SVM trained for A*0101, A*2601, A*2602, B*0801, B*2702, B*2704, B*2705, B*2706, and B*3503 illustrated accuracy ranging between about 80% to 100%. This level of accuracy has not been demonstrated by any other prediction algorithm to date.

It is apparent that the SVM is able to differentiate between binding and non-binding entries using features based upon the peptide's structure and physiochemical properties. However, the structure and physiochemical properties of the allele cleft (represented via the polymorphic positions of the alpha-1 and alpha-2 domains of the allele) can provide further information for allele binding preferences than those provided by the peptide's features alone (as demonstrated by the B*27 group of alleles).

We have also developed and extended an existing database: MHCPEP, converting the database to XML format. A web interface was also developed allowing users to query the database (a feature that was not provided by MHCPEP). The web interface also allows for automatic submission of additional entries into the database. The predicted non-binding entries in the database should be considered only in the case where anchor positions are 2 and 9 for the given allele.

Experimentation demonstrated a number of important SVM features. Higher order kernels, such as the RBF or polynomial kernels are required to accurately predict high order input space. The polynomial kernel of degree 3 provided good accuracy for the majority of experiments involving peptide features only. Higher order degree polynomials (degree 6) are required when allele features were also included. In general however, the polynomial kernel proved to be the best kernel for SVM application to MHC-peptide binding prediction. The RBF kernel proved to be the most sensitive to the uneven data sets and the noise level in the available data. SVM also demonstrated their ability for 'fine tuning' via the

choice of kernel function, diagonal factor, and a number of other parameters.

Performance of the engineered kernel was not discussed as this kernel performed poorly. The Infi-K-TSS algorithm incorporated local domain knowledge, based upon the notion of ‘locality’ (so that closer points are allocated a higher weighting). However, SVM were applied to uneven distributions of data that resulted in this engineered kernel dramatically skewing the data towards the larger class set. Therefore engineered kernels that incorporate ‘global’ domain knowledge are required. Currently only local engineered kernels have been developed [6, 8, 31]. Further, as biological data often involves high order input space, work involving the development of higher order kernels would aid in SVM application to multiple areas of biological analysis.

Future work in this area could also include the use of *feature relevance experts*, such as those applied by Chow [27] to experiment with other features available in AAindex database. The use of such *feature relevance experts* could also aid in biological analysis. For example, anchor residues for each allele could be predicted based upon the amino acid positions that exhibit the most deterministic features.

In conclusion, SVM are an effective means of predicting MHC-peptide binding and further work in this area is warranted based upon our findings.

Original Honours Proposal

Title: Support Vector Machine prediction of MHC-peptide binding

Author: Rebecca Watson

Supervisor: Associate Professor Gareth Chelvanayagam

A.1 Background

A.1.1 Biological Resources

MHCPEP is a database of MHC binding peptides. Over 9000 peptide sequences known to bind to MHC molecules are held in the database which consist of the actual peptide sequence, its MHC specificity, and other selected features [11].

A.1.2 Support Vector Machine (SVM) Resources

SVM Theory

A SVM; like an artificial neural network (ANN); is a supervised computer learning network as they utilize prior knowledge of expression data to categorise input data [9]. SVMs were originally introduced by Vapnik and co-workers, whose work has since been successfully expanded by many researchers [15]. Research involving SVMs have illustrated promising results which are comparable to ANN and other algorithms.

Often we will wish to classify data into two distinct categories. This project aims to construct a SVM which can be trained to classify specific allele-peptide combinations as binding; a positive result, or as non-binding; a negative result. Thus the SVM will effectively aim at finding a classification function,

$$f : R^n \rightarrow \{\pm 1\} \tag{A.1}$$

using labeled training data from R^n [31]. The SVM could then be applied to the classification of unknown allele-peptide binding combinations. Implemented SVMs are available online [16].

Artificial Neural Network (ANN) Research

Supervised machine learning classification of MHC binding peptides has previously been researched using ANN [10]. This research will allow a relatively direct comparison of SVM and ANN classification capabilities.

Also, a disadvantage of neural networks is their “black box” approach. That is, they are unable to be very ‘transparent’ in their calculations or show the steps involved in researching their conclusions. It may in fact be possible to provide more ‘rationale’ when using SVM that would prove to be an added bonus in the use of SVMs.

A.2 Aim

MHCPEP is a database consisting of peptides that bind to class I or II MHC molecules [11]. The ability to predict the binding of specific allele - peptide pairs is directly proportional to the ability to predict immunological compatibilities. A Support Vector Machine (SVM) could be applied to this prediction, and also in the identification of possibly misclassified binding pairs whose binding compatibilities are not supported by expression data.

The aim of this project is to produce means of accurately classifying allele-peptide-binding pairs.

A.3 Method

A.3.1 SVM specification

Time Required: one month, 22nd April 2001

In order to specify a SVM, we must specify two parameters; the kernel function (the map $f : R^n \rightarrow \{\pm 1\}$ from input space to feature space), and the “penalty for violating the soft margin” [9]. The choice of these two parameters depends on the test data to be classified.

Kernel Function

The kernel function maps data from the input space to feature space -1, +1. The choice of the kernel function will depend on the properties of the peptide data from MHCPEP. These properties include probability distribution of features, level of noise in data, and the balance of positive and negative examples.

Popular kernel functions to date are the simple dot product, polynomial kernels of varying degree and the radial basis kernel.

Margin

The margin is the minimum distance, in feature space, between training points and the separating surface (the hyperplane) [31]. The chosen hyperplane, for a given set of training data, is the one that is of maximum distance to the training data, that is, of maximum margin. This hyperplane is termed the “maximal marginal hyperplane” [15].

Noisy data can introduce training errors. Due to noisy data a separating hyperplane may not exist as classes may overlap in areas. We can however sacrifice some training accuracy to gain better predictive power [15]. To compensate for noise we can introduce “slack variables” which allow for the possibility of examples violating the hyperplane barrier [26]. This results in a ‘soft margin’.

Possible Feature Selection

SVM can be used to identify features in the two resultant classes that differ. We can then weight these features in our kernel functions, the highest weight being the top feature [15].

Prior to training the SVM however, we can place any suitable weights on features known to differentiate between the two resultant classes.

Training Set Selection

The choice of the training set used is also very important. Training data must be an “unbiased sample from the same source as the test data” so that the training and test data will have the “same underlying probability distribution” [31].

Secondly, the capacity of the SVM; the size of the class of functions that the kernel can be chosen from; must be suitable to the size of available training data [26]. Thus the capacity is “limited according to the statistical theory of learning from small samples” [31].

A.3.2 Code SVM

Time Required: 1.5 months, 8th June 2001

The available implementation written by Professor Grundy was coded in C [16]. Many other implementations are available online and I plan to draw on these implementations as well as on written material outlining SVM implementation.

Code will be needed to form the interface between the SVM and the downloaded MHCPEP database. In the final stages of the project a GUI will also be implemented which will allow a user to input specific allele-peptide combinations and query their binding compatibility.

A.3.3 Training and Testing SVM

Time required; 2 months, 10th August 2001

Experimentation

Experimentation involving different kernel functions, feature set selections, diagonal factors and other variables will be conducted. Optimisation of these variable combinations will hopefully create classification results comparable to the performance of ANN research in this area. The trained SVM can also be compared to other algorithms including matrix methods, modelling and energetics (evaluating force fields).

Performance

The use of “hold-one-out cross validation tests” [2] (where the SVM is trained using all but one data item, which is then classified by the trained SVM), can be used to measure the performance of the SVM.

Each of the input items will result in an output that is one of:

- TP - true positive; both the MHCPEP database and the SVM classed the peptide as binding,
- TN - true negative; both classed the peptide as non-binding,
- FP - false negative; classes as binding by SVM and non binding by MHCPEP,
- FN - false negative; classes as non-binding by SVM and binding by MHCPEP [9].

These classifications can then be applied in a number of ways to measure the performance of the SVM.

Identification of Possible Outliers

SVM have illustrated the ability to identify outliers or possibly misclassified instances in test data [15]. The possibility of misclassified data, though small, must be considered in training and testing the SVM.

A.3.4 GUI Specification and Construction

Time Required: 3 weeks, 31st August 2001

The overall goal of training a SVM is of course to classify unknown input data as members or non-members of a class. The GUI will allow any combination of allele - peptide pairs to be queried and classified as binding or non-binding (using the “optimally” trained SVM).

A.4 Software and Hardware Requirements

The machines available in the labs should be adequate in terms of software and hardware requirements. When training the SVM however, long periods of time may be needed on the same machine. The training and testing period will take approximately two months and so the provision of a separate machine may be necessary depending on the “Standing time” required during training and testing.

Bibliography

- [1] Bell Laboratories [Online]. <http://www.lucent.com/> [2001, 25 November].
- [2] HLA infomatics group. Available: <http://www.anthonynolan.org.uk/HIG/index.html> [2001, 18 September].
- [3] HLA sequencing data, funded by Anthony Nolan Bone Marrow Trust. Available: <http://www.anthonynolan.com/HIG/data.html> [2001, 18 September].
- [4] AMARI, S., AND WU, S. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks* 12 (1999), 783–789.
- [5] ANDERSEN, M. H., TAN, L., SONDERGAARD, I., ZEUTHEN, J., ELLIOTT, T., AND HAURUM, J. S. Poor correspondence between predicted and experimental binding of peptides to class I MHC molecules. *Tissue Antigens* 55 (2000), 519–531.
- [6] BARZILAY, O., AND BRAILOVSKY, V. L. On domain knowledge and feature selection using a support vector machine. *Pattern Recognition Letters* 20 (1999), 475–484.
- [7] BOCK, J. R., AND A., G. D. Predicting protein-protein interactions from primary structure. *Bioinformatics* 17, 5 (2001), 455–460.
- [8] BRAILOVSKY, V. L., BARZILAY, O., AND SHAHAVE, R. On global, local, mixed and neighborhood kernels for support vector machines. *Pattern Recognition Letters* 20 (1999), 1183–1190.
- [9] BROWN, M. P. S., GRUNDY, W. N., LIN, D., CRISTIANINI, N., SUGNET, C. W., FUREY, T. S., ARES JR., M., AND HAUSSLER, D. Knowledge-based analysis of microarray gene expression data by using support vector machines. *PNAS* 97 (2000), 262–297.
- [10] BRUSIC, V., RUDY, G., AND HARRISON, L. C. Prediction of MHC binding peptides using artificial neural networks. In *Complex Systems: Mechanism of Adaption*, R. J. Stonier and X. H. Yo, Eds. IOS Press, Oxford, 1994, pp. 253–260.
- [11] BRUSIC, V., RUDY, G., KYNE, A. P., AND HARRISON, L. C. MHCPEP, a database of MHC-binding peptides: Update 1997,. *Nucleic Acids Research* 26, 1 (1997), 368–371.
- [12] CHELVANAYAGAM, G. A roadmap for HLA-A, HLA-B and HLA-C peptide binding specificities. *Immunogenetics* 45, 1 (1996), 15–26.
- [13] CORPORATION, A. Web site [Online]. <http://www.aphton.com/> [2001, 15 September].

- [14] FLACH, P. A. On the state of the art in machine learning: A personal review. *Artificial Intelligence* 131 (2001), 199–222.
- [15] FUREY, T. S., CRISTIANINI, N., DUFFY, N., BEDNARSKI, D. W., SCHUMMER, M., AND HAUSSLER, D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16 (2000), 906–914.
- [16] GRUNDY, P. B. SVM software [Online]. Available: <http://www.cs.columbia.edu/~bgrundy/svm> [2001, 3 June].
- [17] GUERY, J. C., NEAGU, M., RODRIGUEZ-TARDUCHY, G., AND ADORINI, L. Selective immunosuppression by administration of major histocompatibility complex class II-binding peptides. *J Exp Med* 177, 5 (May 1993), 1461–8.
- [18] KAWASHIMA, S., AND KANEHISA, M. AAindex: Amino acid index database. *Nucleic Acids Research* 28, 1 (2000), 374.
- [19] NAKAI, K., KIDERA, A., AND KANEHISA, M. *Protein Eng.* 2 (1988), 93–100.
- [20] OPPER, M., AND URBANCZIK, R. Universal learning curves of support vector machines. *Physical Review Letters* 86, 19 (May 2001).
- [21] PARKER, K. C., BEDNAREK, M. A., AND COLIGAN, J. E. Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. *J. Immunol.* 152, 163.
- [22] Q., D. C. H., AND DUBCHAK, I. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* 17, 4 (2001), 349–358.
- [23] RAMMENSEE, H. G., BACHMANN, J., EMMERICH, N. N., BACHOR, O. A., AND STEVANOVIC, S. SYFPEITHI: database for MHC ligands and peptide motifs. *Immunogenetics* 50 (1999), 213–219.
- [24] RAMMENSEE, H. G., FRIEDE, T., AND STEVANOVIC, S. MHC ligands and peptide motifs: first listing. *Immunogenetics* 41 (1995), 178–228.
- [25] ROBINSON, J., WALLER, M. J., PARHAM, P., BODMER, J. G., AND MARSH, S. G. E. IMGT/HLA database - a sequence database for the human major histocompatibility complex. *Nucleic Acids Research* 29 (2001), 210–213.
- [26] SCHOLKOPF, B., BURGESS, C. J. C., AND SMOLA, A. J., Eds. *Advances in Kernel Methods, Support Vector Learning*. MIT Press, England, 1999.
- [27] SHOW, M. L., MOLER, E. J., AND MIAN, I. S. Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. *Physiol Genomics* 5 (2001), 99–111.

- [28] STUNIOLO, T., BONO, E., DING, J., RADDRIZZANI, L., TUERECI, O., SAHIN, U., BRAZENTHALER, M., GALLAZZI, F., PROTTI, M. P. SINGAGLIA, F., AND HAMMER, J. Generation of tissue-specific and promiscuous hla ligand databases using DNA microarrays and virtual HLA class II matrices. *Nature Biotechnology* 17 (June 1999).
- [29] WATSON, R. MHC-Peptide Prediction [Online]. <http://www.cs.uwa.edu.au/rebeccaw/> [2001, 15 September].
- [30] WATSON, R. MHCPEP_R[Online]. <http://www.cs.uwa.edu.au/rebeccaw/> [2001, 15 September].
- [31] ZIEN, A., RATSCH, G., MIKA, S., SCHOLKOPF, B., LENGAUER, T., AND MULLER, K. R. Engineering support vector machine kernels that recognize translation initiation sites. *BioInformatics* 16 (2000), 799–807.