Introduction to the Peptide Binding Problem of Computational Immunology: New Results

Wen-Jun Shen · Hau-San Wong · Quan-Wu Xiao · Xin Guo · Stephen Smale

Abstract We attempt to establish geometrical methods for amino acid sequences. To measure the similarities of these sequences, a kernel on strings is defined using only the sequence structure and a good amino acid substitution matrix (e.g. BLOSUM62). The kernel is used in learning machines to predict binding affinities of peptides to human leukocyte antigen DR (HLA-DR) molecules. On both fixed allele (Nielsen and Lund in BMC Bioinform. 10:296, 2009) and pan-allele (Nielsen et al. in Immunome Res. 6(1):9, 2010) benchmark databases, our algorithm achieves the state-of-the-art performance. The kernel is also used to define a distance on an HLA-DR allele set

Communicated by Teresa Krick.

W.-J. Shen · H.-S. Wong
Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

W.-J. Shen
e-mail: wenjshen-c@my.cityu.edu.hk

H.-S. Wong
e-mail: cshswong@cityu.edu.hk

Q.-W. Xiao
Microsoft Search Technology Center Asia, Beijing, China
e-mail: qwxiao@live.com

X. Guo
Department of Statistical Science, Duke University, Durham, NC, USA
e-mail: xg20@stat.duke.edu

S. Smale (✉)
Department of Mathematics, City University of Hong Kong, Kowloon, Hong Kong
e-mail: smale@cityu.edu.hk

based on which a clustering analysis precisely recovers the serotype classifications assigned by WHO (Holdsworth et al. in Tissue Antigens 73(2):95–170, 2009; Marsh et al. in Tissue Antigens 75(4):291–455, 2010). These results suggest that our kernel relates well the sequence structure of both peptides and HLA-DR molecules to their biological functions, and that it offers a simple, powerful and promising methodology to immunology and amino acid sequence studies.

Keywords String kernel · Peptide binding prediction · Reproducing kernel Hilbert space · Major histocompatibility complex · HLA DRB allele classification

Mathematics Subject Classification 62P10 · 68T05 · 92B15 · 92C40

1 Introduction

Large scientific and industrial enterprises are engaged in efforts to produce new vaccines from synthetic peptides. The study of peptide binding to appropriate MHC II molecules is a major part of this effort. Our goal here is to support the use of a certain "string kernel" for peptide binding prediction as well as for the classification of supertypes of the major histocompatibility complex (MHC) alleles. We hope that this will contribute to the prediction of vaccines and the understanding of organ transplants. This work actually could be applied to any other receptor-binding system where large binding sets are available.

Part of our innovation and philosophy is our attitude and our relationship towards multiple sequence alignments and the associated gaps with their introduction of errors. This paper is alignment-free except for the use of BLOSUM62 (see below). Note that Waterman, who originally played a big role in alignments, is now writing papers about alignment-free theory [49].

Our point of view is that some key biological information is contained in just two places: first, in a similarity kernel (or substitution matrix) on the set of the fundamental amino acids; and second, on a good representation of the relevant alleles as strings of these amino acids.

This is achieved with great simplicity and predictive power. Along the way we find that emphasizing peptide binding as a real-valued function rather than a binding/non-binding dichotomy clarifies the issues. We use a modification of BLOSUM62 followed by a Hadamard power. We also use regularized least squares (RLS) in contrast to support vector machines as the former is consistent with our regression emphasis.

We next briefly describe the construction (more details also in Sect. 2) of our main kernel $\hat{K}^3$ on amino acid sequences, inspired by local alignment kernels (see e.g. [36, 47]) as well as an analogous kernel in vision (see [44]) begins.

For the purposes of this paper, a kernel K is a symmetric function $K : X \times X \to \mathbb{R}$ where X is a finite set. Given an order on X, K may be represented as a matrix (think of X as the set of indices of the matrix elements). Then it is assumed that K is positive definite (in such a representation).

Let $\mathscr{A}$ be the set of the 20 basic (for life) amino acids. Every protein has a representation as a string of elements of $\mathscr{A}$.

Step 1. Definition of a kernel $K^1 : \mathscr{A} \times \mathscr{A} \to \mathbb{R}$.

BLOSUM62 is a similarity (or substitution) matrix on $\mathscr{A}$ used in immunology [16]. In the formulation of BLOSUM62, a kernel $Q : \mathscr{A} \times \mathscr{A} \to \mathbb{R}$ is defined using blocks of aligned strings of amino acids representing proteins. One can think Q as the "raw data" of BLOSUM62. It is symmetric, positive-valued, and it is a probability measure on $\mathscr{A} \times \mathscr{A}$. (We have in addition checked that it is positive definite.)

Let p be the marginal probability defined on $\mathscr{A}$ by Q. That is,

$$p(x) = \sum_{y \in \mathscr{A}} Q(x, y).$$

Next, we define the BLOSUM62-2 matrix, indexed by the set $\mathscr{A}$, as

$$[\text{BLOSUM62-2}](x, y) = \frac{Q(x, y)}{p(x) p(y)}.$$

We list the BLOSUM62-2 matrix in Appendix. Suppose $\beta > 0$ is a parameter, usually chosen about $\frac{1}{8}$ or $\frac{1}{10}$ (still mysterious). Then a kernel $K^1 : \mathscr{A} \times \mathscr{A} \to \mathbb{R}$ is given by

$$K^1(x, y) = \big([\text{BLOSUM62-2}](x, y)\big)^\beta. \tag{1}$$

Note that the power in (1) is of the matrix entries, not of the matrix.

Step 2. Let $\mathscr{A}^1 = \mathscr{A}$ and define $\mathscr{A}^{k+1} = \mathscr{A}^k \times \mathscr{A}$ recursively for any $k \in \mathbb{N}$. We say s is an amino acid sequence (or string) if $s \in \bigcup_{k=1}^{\infty} \mathscr{A}^k$, and $s = (s_1, \ldots, s_k)$ is a k-mer if $s \in \mathscr{A}^k$ for some $k \in \mathbb{N}$ with $s_i \in \mathscr{A}$. Consider

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

where u, v are amino acid strings of the same length k, $u = (u_1, \ldots, u_k)$, $v = (v_1, \ldots, v_k)$; u, v are k-mers. K_k^2 is a kernel on the set of all k-mers.

Step 3. Let $f = (f_1, \ldots, f_m)$ be an amino acid sequence. Denote by $|f|$ the length of f (so here $|f| = m$). Write $u \subset f$ whenever u is of the form $u = (f_{i+1}, \ldots, f_{i+k})$ for some $1 \leq i + 1 \leq i + k \leq m$. Let g be another amino acid sequence, then define

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

for f and g in any finite set X of amino acid sequences. Here, and in all of this paper, we abuse the notation to let the sum count each occurrence of u in f (and of v in g). In other words we count these occurrences "with multiplicity". While u and v need to have the same length, not so for f and g. Replacing the sum by an average gives a different but related kernel.

We define the correlation kernel $\hat{K}$ normalized from any kernel K by

$$\hat{K}(x, y) = \frac{K(x, y)}{\sqrt{K(x, x)K(y, y)}}.$$

In particular, let $\hat{K}^3$ be the correlation kernel of K^3.

Remark 1 $\hat{K}^3$ is a kernel (see Sect. 2.2). It is symmetric, positive definite, positive-valued; it is basic for the results and development of this paper. We sometimes say string kernel. The construction works for any kernel (at the place of K^1) on any finite alphabet (replacing $\mathscr{A}$).

Remark 2 For some background see [15, 19, 23, 35, 37, 51]. But we use no gap penalty or even gaps, no logarithms, no implied round-offs, and no alignments (except the BLOSUM62-2 matrix which indirectly contains some alignment information).

Remark 3 For complexity reasons one may limit the values of k in Step 3 with a small loss of accuracy, or even choose the k-mers at random.

Remark 4 The chains (sequences) we use are proteins and peptides. Peptides are short sequence fragments of proteins.

Associated to a gene are a number of variants called alleles.[1] An allele could be said to be the representative of the gene in an individual. The alleles give different characteristics the these individuals, for example, resistance to diseases. Thus in WHO data bases, [28], one has WHO nomenclature of an allele; before the asterisk is the gene and after the asterisk is the detailed code of the allele.

MHC II and MHC I are sets of alleles which are associated with immunological responses to viruses, bacteria, peptides and related entities. See [13, 26] for good introductions. In this paper we only study HLA II, the MHC II in human beings. HLA-DRB (or simply DRB) describes a subset of HLA II alleles which play a central role in immunology, as well as in this paper. Alleles have representations as amino acid sequences.

1.1 First Application: Binding Affinity Prediction

Peptide binding to a fixed HLA II (and HLA I as well) molecule (or an allele) a is a crucial step in the immune response of the human body to a pathogen or a peptide-based vaccine. Its prediction is computed from data of the form $(x_i, y_i)_{i=1}^m$, $x_i \in \mathscr{P}_a$ and $y_i \in [0, 1]$, where $\mathscr{P}_a$ is a set of peptides (i.e. sequences of amino acids; in this paper we study peptides of length 9 to 37 amino acids, usually about 15) associated to an HLA II allele a. Here y_i expresses the strength of the binding of x_i to a. The peptide binding problem occupies much research. We may use our kernel $\hat{K}^3$ described above for this problem since peptides are represented as strings of amino acids. Our

---

[1]Allele: an alternative form of a gene that occurs at a specified chromosomal position (locus) [22].

prediction thus uses only the amino acid sequences of the peptides, a substitution matrix, and some existing binding affinities (as "data").

Following RLS supervised learning with kernel $K = \hat{K}^3$, the main construction is to compute

$$f_a = \arg \min_{f \in \mathscr{H}_K} \sum_{i=1}^{m} \big(f(x_i) - y_i\big)^2 + \lambda \|f\|_K^2. \tag{2}$$

Here $\lambda > 0$ and the index $\beta > 0$ in $\hat{K}^3$ are chosen by a procedure called leave-one-out cross-validation (defined in Sect. 2.3, see also [12]). Also $\mathscr{H}_K$ is the space of functions spanned by $\{K_x : x \in \mathscr{P}\}$ (where $K_x(y) := K(x, y)$) on a finite set $\mathscr{P}$ of peptides containing $\mathscr{P}_a$. An inner product on $\mathscr{H}_K$ is defined on the basis vectors as $\langle K_x, K_y \rangle_K = K(x, y)$, then in general by linear extension. The norm of $f \in \mathscr{H}_K$ induced by this inner product is denoted by $\|f\|_K$. In (2), f_a is the predicted peptide binding function. We refer to this algorithm as "KernelRLS".

For the set of HLA II alleles, with the best data available we have Table 1. The area under the receiver operating characteristic curve (area under the ROC curve, AUC, see Sect. 2.3 for definition) is the main measure of accuracy used in the peptide binding literature. NN-W refers to the algorithm which up to now has achieved the most accurate results for this problem, although there are many previous contributions as [10, 24, 50]. In Sect. 2 there is more detail.

The following results compare ours with [29], "The method is evaluated on a large-scale benchmark consisting of six independent data sets covering 14 human MHC class II alleles, and is demonstrated to outperform other state-of-the-art MHC class II prediction methods", directly quoted from the abstract of [29]. Since ours compares very well with [29], see Table 1, we do not need to compare ours with the other algorithms, including for example string kernel methods etc. Similar considerations apply to the pan-allele case using reference [31].

We note the simplicity and universality of the algorithm that is based on $\hat{K}^3$, which itself has this simplicity with the contributions from the substitution matrix (i.e. BLOSUM62-2) and the sequential representation of the peptides. There is an important generalization of the peptide binding problem where the allele is allowed to vary. Our results on this problem are detailed in Sect. 3.

We have found the third decimal point useful for comparison of algorithms, and understanding nuances of algorithms, but always keeping in mind the danger of over-fitting. What is a good AUC score has a very large range depending on the setting.

Since our manuscript was submitted, Andreatta [1] writes that he has tried our kernel method on MHC I peptide binding. We quote from his page 80, the "kernel method showed performance comparable to state-of-the-art methods for MHC class I prediction".

1.2 Second Application: Clustering and Supertypes

We consider the classification problem of DRB (HLA-DR β sequence) alleles into groups called supertypes as follows. The understanding of DRB similarities is very important for the designation of high population coverage vaccines. An HLA gene can generate a large number of allelic variants and this polymorphism guarantees a

Table 1 The algorithm performance of RLS on each fixed allele in the benchmark [29]. If a is the allele in column 1, then the number of peptides in $\mathscr{P}_a$ is given in column 2. The root-mean-square error (RMSE) scores are listed (see Sect. 2.3). The AUC scores of the RLS and the NN-W algorithm are listed for comparison, where a common threshold $\theta = 0.426$ is used [29] in the final thresholding step into binding and non-binding (see Sect. 2.3 for the details). The best AUC in each row is marked in bold. In all the tables the weighted average scores are given by the weighting on the size $\#\mathscr{P}_a$ of the corresponding peptide sets $\mathscr{P}_a$

List of alleles, a	$\#\mathscr{P}_a$	KernelRLS		NN-W in [29]
		RMSE	AUC	AUC
DRB1*0101	5166	0.18660	**0.85707**	0.836
DRB1*0301	1020	0.18497	**0.82813**	0.816
DRB1*0401	1024	0.24055	**0.78431**	0.771
DRB1*0404	663	0.20702	0.81425	**0.818**
DRB1*0405	630	0.20069	**0.79296**	0.781
DRB1*0701	853	0.21944	0.83440	**0.841**
DRB1*0802	420	0.19666	**0.83538**	0.832
DRB1*0901	530	0.25398	**0.66591**	0.616
DRB1*1101	950	0.20776	**0.83703**	0.823
DRB1*1302	498	0.22569	0.80410	**0.831**
DRB1*1501	934	0.23268	**0.76436**	0.758
DRB3*0101	549	0.15945	0.80228	**0.844**
DRB4*0101	446	0.20809	0.81057	**0.811**
DRB5*0101	924	0.23038	**0.80568**	0.797
Average		0.21100	0.80260	0.798
Weighted average		0.20451	0.82059	0.810

population from being eradicated by a single pathogen. See Sect. 1.2 of [26]. Furthermore, there are no more than twelve HLA II alleles in each individual [20] and each HLA II molecule binds only to specific peptides [40, 53]. As a result, it is difficult to design an effective vaccine for a large population. It has been demonstrated that many HLA molecules have overlapping peptide binding sets and there have been several attempts to group them into supertypes accordingly [3, 6, 25, 32, 39, 42, 43]. The supertypes are designed so that the HLA molecules in the same supertype will have a similar peptide binding specificity.

The Nomenclature Committee of the World Health Organization (WHO) [28] has given extensive tables on serological type assignments to DRB alleles which are based on the works of many organizations and labs throughout the world. In particular the HLA dictionary 2008 by Holdsworth et al. [17] acknowledges especially the data from the WHO Nomenclature Committee for Factors of the HLA system, the International Cell Exchange and the National Marrow Donor Program. The text in Holdsworth et al., 2008 [17] indicates also the ambiguities of such assignments especially in certain serological types.

We define a set $\mathscr{N}$ of DRB alleles as follows. We downloaded 820 DRB allele sequences from the IMGT/HLA Sequence Database [33].[2] Then 14 non-expressed alleles were excluded and there remained 806 alleles. We use two markers "RFL" and "TVQ" (which are strings in the standard alphabetical code of amino acids given in Table 9), each of which consists of three amino acids to identify the polymorphic part of a DRB allele. For each allele, we only consider the amino acids located between the markers "RFL" (the location of the first occurrence of "RFL") and "TVQ" (the location of the last occurrence of "TVQ"). One reason is the majority of polymorphic positions occur in exon 2 of the HLA class II genes [14], and the amino acids located between the markers "RFL" and "TVQ" constitute the whole exon 2 [46]. The DRB alleles are encoded by six exons. Exon 2 is the most important component constituting an HLA II–peptide binding site. The other reason is in the HLA pseudo-sequences used in the NetMHCIIpan [30], all positions of the allele contacting with the peptide occur in this range.

Thus each allele is transformed into a normal form. We should note that two different alleles may have the same normal form. For those alleles with the same normal form, we only consider the first one. The order is according to the official names given by WHO. We collect the remaining 786 alleles with no duplicate normal forms into a set, we call $\mathscr{N}$. This set not only includes all alleles listed in the tables of [17], but also contains all new alleles since 2008 until August 2011.

Thus $\mathscr{N}$ may be identified with a set of amino acid sequences. Next impose the kernel $\hat{K}^3$ above on $\mathscr{N}$ where $\beta = 0.06$, we call the kernel $\hat{K}^3_{\mathscr{N}}$.

On $\mathscr{N}$ we define a distance derived from $\hat{K}^3_{\mathscr{N}}$ by

$$D_{L^2}(x, y) = \left(\frac{1}{\#\mathscr{N}} \sum_{z \in \mathscr{N}} \left(\hat{K}^3_{\mathscr{N}}(x, z) - \hat{K}^3_{\mathscr{N}}(y, z) \right)^2 \right)^{1/2}, \quad \forall x, y \in \mathscr{N}. \quad (3)$$

Here and in the sequel we denote $\#A$ the size of a finite set A.

The DRB1*11 and DRB1*13 families of alleles have been the most difficult to deal with by WHO and for us as well. Therefore we will exclude the DRB1*11 and DRB1*13 families of alleles in the following cluster tree construction with the evidence that clustering of these two groups is ineffective. See [26] for a discussion. They are left to be analyzed separately.[3]

The set $\mathscr{M}$ consists of all DRB alleles except for the DRB1*11 and DRB1*13 families of alleles. $\mathscr{M}$ is a subset of the set $\mathscr{N}$. We produce a clustering of $\mathscr{M}$ based on the L^2 distance defined on $\mathscr{N}$ in (3), D_{L^2}, restricted to $\mathscr{M}$, and use the OWA (Ordered Weighted Averaging) [52] based linkage (see Sect. 4.1 for definition) instead of the "single" linkage in the hierarchical clustering algorithm.

This clustering uses no previous serological type information and no alignments. We have assigned supertypes labeled ST1, ST2, ST3, ST4, ST5, ST6, ST7, ST8, ST9, ST10, ST51, ST52 and ST53 to certain clusters in the Tree shown in Fig. 1 based on

---

[2]ftp://ftp.ebi.ac.uk/pub/databases/imgt/mhc/hla/DRB_prot.fasta.

[3]We have found from a number of different experiments that "they do not cluster". (Perhaps the geometric phenomenon here is in the higher dimensional scaled topology, i.e. the Betti numbers $\beta_i > 0$, for $i > 0$ [4].)
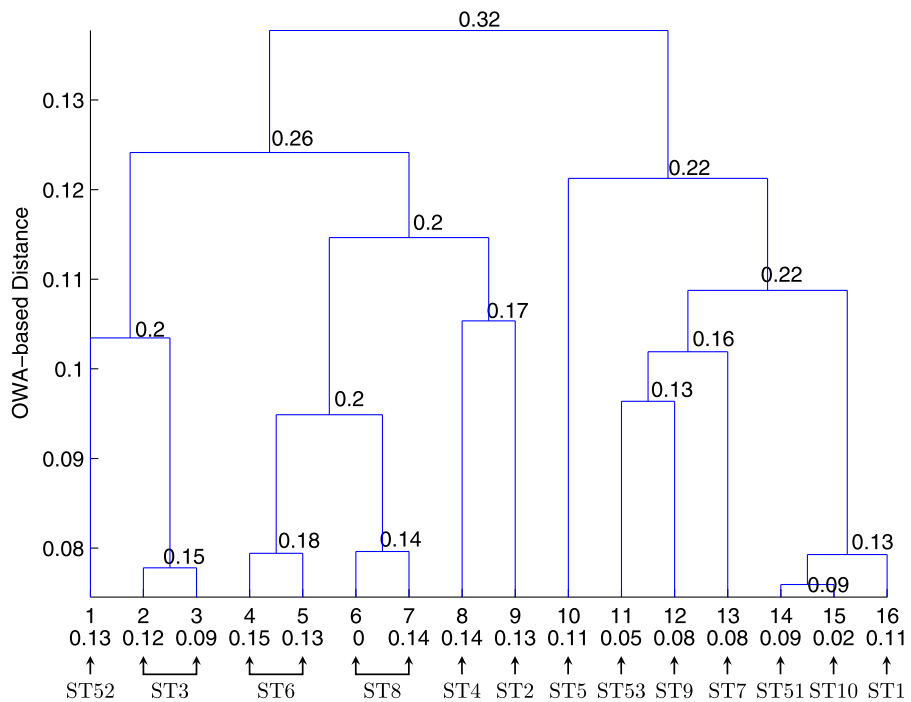
Fig. 1 Cluster tree on 559 DRB alleles. The diameters of the leaf nodes are given at the bottom of the figure. The numbers given in the figure are the diameters of the corresponding unions of clusters

contents of the clusters described in Table 6. Peptides have played no role in our model. Differing from the artificial neural network method [17, 27], no "training data" of any previously classified alleles are used in our clustering. We make use of the DRB amino acid sequences to build the cluster tree. Only making use of these amino acid sequences, our supertypes are in exact agreement with WHO assigned serological types [17], as can be seen by checking the supertypes against the clusters in Table 6.

For a cluster, i.e. leaf of the tree, one retrieves the serotype by checking the alleles in that cluster and then use the WHO labeling. See Sect. 4 for details.

This second application is given in some detail in Sect. 4.

2 Kernel Method for Binding Affinity Prediction

In this section we describe in detail the construction of our string kernel. The motivation is to relate the sequence information of strings (peptides or alleles) to their biological functions (binding affinities). A kernel works as a measure of similarity and supports the application of powerful machine learning algorithms such as RLS which we use in this paper. For a fixed allele, binding affinity is a function on peptides with values in [0, 1]. The function values on some peptides are available as the data, according to which RLS outputs a function that predicts for a new peptide the binding

affinity to the MHC II molecule. The method is generalized in the next section to the pan-allele kernel algorithm that takes also the allele structure into account.

2.1 Kernels

We suppose throughout the paper that X is a finite set. We now give the definition of a kernel, of which an important example is our string kernel.

Definition 1 A symmetric function $K : X \times X \to \mathbb{R}$ is called a kernel on X if it is positive definite, in the sense that by choosing an order on X, K can be represented as a positive definite matrix $(K(x, y))_{x,y \in X}$.

Kernels have the following properties [2, 7, 41].

Lemma 1

(i) *If K is a kernel on X then it is also a kernel on any subset X_1 of X.*
(ii) *If K_1 and K_2 are kernels on X, then $K : X \times X \to \mathbb{R}$ defined by*

$$K(x, x') = K_1(x, x') + K_2(x, x')$$

is also a kernel.
(iii) *If K_1 is a kernel on X_1 and K_2 is a kernel on X_2, then $K : (X_1 \times X_2) \times (X_1 \times X_2) \to \mathbb{R}$ defined by*

$$K\big((x_1, x_2), (x_1', x_2')\big) = K_1(x_1, x_1') \cdot K_2(x_2, x_2')$$

is a kernel on $X_1 \times X_2$.
(iv) *If K is a kernel on X, and f is a real-valued function on X that maps no point to zero, then $K' : X \times X$ defined by*

$$K'(x, x') = f(x)K(x, x')f(x')$$

is also a kernel.
(v) *If K is a kernel on X, then the correlation normalization $\hat{K}$ of K given by*

$$\hat{K}(x, x') = \frac{K(x, x')}{\sqrt{K(x, x)K(x', x')}} \tag{4}$$

is also a kernel.

Proof (i), (ii), and (iv) follows the definition directly. (iii) follows the fact that the Kronecker product of two positive definite matrices is positive definite; see [18] for details. The positive definiteness of a kernel K guarantees that $K(x, x) > 0$ for any x in X, so (v) follows (iv). □

Remark 5 Notice that with correlation normalization we have $\hat{K}(x, x) = 1$ for all $x \in X$. This is a desired property because the kernel function is usually used as a similarity measure, and with $\hat{K}$ we can say that each $x \in X$ is similar to itself.

Define the real-valued function K_x on X by $K_x(y) = K(x, y)$. The function space $\mathscr{H}_K = \mathrm{span}\{K_x : x \in X\}$ is a Euclidean space with inner product $\langle K_x, K_y \rangle_K = K(x, y)$, extended linearly to $\mathscr{H}_K$. The norm of a function f in $\mathscr{H}_K$ is denoted as $\|f\|_K$.

Remark 6 The kernel can be defined even without assuming X is finite; in this general case the kernel is referred to as a reproducing kernel [2]. If X is finite then a reproducing kernel is equivalent to our "kernel". The theory of reproducing kernel Hilbert spaces plays an important role in learning [7, 38, 48].

On a finite set X there are two notions of distance derived from a kernel K. The first one is the usual distance in $\mathscr{H}_K$, that is,

$$D_K(x, x') = \|K_x - K_{x'}\|_K,$$

for two points $x, x' \in X$. The second one is the L^2 distance defined by

$$D_{L^2}(x, x') = \left(\frac{1}{\#X} \sum_{t \in X} \big(K(x, t) - K(x', t)\big)^2 \right)^{1/2}.$$

Important examples of the kernels discussed above are our kernel K^3 and its normalization $\hat{K}^3$, both defined on any finite $X \subset \bigcup_{k \geq 1} \mathscr{A}^k$.

2.2 Kernel on Strings

We start with a finite set $\mathscr{A}$ called the alphabet. In the work here $\mathscr{A}$ is the set of 20 amino acids, but the theory in this section applies to any other finite set. For example, as the name suggests, it can work on text for semantic analysis with a similar setting. See also [44] for the framework in vision.

To measure a similarity among the 20 amino acids, Henikoff and Henikoff [16] collect families of related proteins, align them and find conserved regions (i.e. regions that do not mutate frequently or greatly) as blocks in the families. The occurrence of each pair of amino acids in each column of every block is counted. A large number of occurrences indicate that in the conserved regions the corresponding pair of amino acids substitute each other frequently or, in other words, that they are similar. A symmetric matrix Q indexed by $\mathscr{A} \times \mathscr{A}$ is eventually obtained by normalizing the occurrences, so that $\sum_{x, y \in \mathscr{A}} Q(x, y) = 1$ and $Q(x, y)$ indicates the frequency of occurrences. See [16] for details. The BLOSUM62 matrix is constructed accordingly.

Define $K^1 : \mathscr{A} \times \mathscr{A} \to \mathbb{R}$ as

$$K^1(x, y) = \left(\frac{Q(x, y)}{p(x)p(y)} \right)^{\beta}, \quad \text{for some } \beta > 0,$$

where $p : \mathscr{A} \to [0, 1]$ given by

$$p(x) = \sum_{y \in \mathscr{A}} Q(x, y),$$

is the marginal probability distribution on $\mathscr{A}$. When $\beta = 1$, recall that the matrix $(K^1(x, y))_{x,y \in \mathscr{A}}$ is BLOSUM62-2 (one takes logarithm with base 2, scales it with factor 2, and rounds the obtained matrix to integers to obtain the BLOSUM62 matrix). Notice that if one chooses simply $Q = \frac{1}{m} I_{m \times m}$, then one obtains the matrix $I_{m \times m}$ as the analogue of the BLOSUM62-2, and the corresponding K^3 of the Introduction is called the spectrum kernel [23].

In matrix language K^1 is the Hadamard power of the BLOSUM62-2 matrix, where for a matrix $M = (M_{i,j})$ with positive entries and a number $\beta > 0$, we denote $M^{\circ \beta}$ as the βth Hadamard power of M and $\log^{\circ} M$ as the Hadamard logarithm of M, and their (i, j) entries are, respectively,

$$\left(M^{\circ \beta}\right)_{i,j} := (M_{i,j})^{\beta}, \qquad \left(\log^{\circ} M\right)_{i,j} := \log(M_{i,j}).$$

Theorem 1 (Horn and Johnson [18]) *Let A be an $m \times m$ positive-valued symmetric matrix. The Hadamard power $A^{\circ \beta}$ is positive definite for any $\beta > 0$ if and only if the Hadamard logarithm $\log^{\circ} A$ is conditionally positive definite (i.e. positive definite on the space $V = \{v = (v_1, \ldots, v_m) \in \mathbb{R}^m : \sum_{i=1}^{m} v_i = 0\}$).*

Proposition 1 *Every positive Hadamard power of BLOSUM62-2 is positive definite. Thus the above defined K^1 is a kernel for every $\beta > 0$.*

Proof One just shows the eigenvalues of the Hadamard logarithm on V are all positive. One checks this by computer (see Appendix for data). $\qquad \square$

Theorem 2 *Based on any kernel K^1, the functions K_k^2, K^3, and $\hat{K}^3$ defined as in the Introduction are all kernels.*

Proof The fact that K_k^2 is a kernel for $k \geq 1$ follows from Lemma 1(iii). We now prove that K^3 is positive definite on any finite set X of strings, which then implies the same for $\hat{K}^3$ by Lemma 1(v). From Lemma 1(i) it suffices to verify the cases that $X = X_k = \bigcup_{i=1}^{k} \mathscr{A}^i$ for $k \geq 1$. When $k = 1$, K^3 is just K^1 and hence positive definite. We assume now that K^3 is positive definite on X_k with $k = n$.

We claim that the matrices indexed by X_{n+1},

$$K_{i,X_{n+1}}^3(f, g) = \begin{cases} \sum_{\substack{u \subset f, v \subset g \\ |u|=|v|=i}} K^2(u, v) & \text{if } |f|, |g| \geq i, \\ 0 & \text{if } |f| < i \text{ or } |g| < i, \end{cases}$$

are all positive semi-definite. In fact, for any $1 \leq i \leq n$,

$$K_{i,X_{n+1}}^3 = P_i K_i^2 P_i^{\mathrm{T}}, \tag{5}$$

where K_i^2 is the matrix $(K_i^2(u, v))_{u,v \in \mathscr{A}^i}$, and P_i is a matrix with X_{n+1} as the row index set and $\mathscr{A}^i$ as the column index set, and for any $f \in X_{n+1}$ and $u \in \mathscr{A}^i$, $P_i(f, u)$ counts the number of times u occurs in f. Let us explain Eq. (5) a little more. For f

and g in X_{n+1}, from the definition of P_i we have

$$\left(P_i K_i^2 P_i^T\right)(f, g) = \sum_{u,v \in \mathscr{A}^i} P_i(f, u) P_i(g, v) K_i^2(u, v)$$

$$= \sum_{\substack{u \subset f, v \subset g \\ |u|=|v|=i}} K_i^2(u, v), \quad \forall i. \tag{6}$$

Summing Eq. (6) above over $i \in \mathbb{N}$ gives the definition of $K^3(f, g)$.

For $i = n + 1$, we have

$$K_{n+1, X_{n+1}}^3 (f, g) = \begin{cases} 0 & f \notin \mathscr{A}^{n+1} \text{ or } g \notin \mathscr{A}^{n+1}, \\ K_{n+1}^2(f, g) & \text{otherwise.} \end{cases}$$

Therefore $K_{n+1, X_{n+1}}^3$ is positive definite on $\mathscr{A}^{n+1}$, and is zero elsewhere. Since

$$K^3(f, g) = \sum_{i=1}^{n} K_{i, X_{n+1}}^3 (f, g), \quad \forall f, g \in X_n,$$

we know that the sum of $K_{i, X_{n+1}}^3$ with $i = 1, \ldots, n$ are positive definite on X_n, and positive semi-definite on X_{n+1}. Because

$$K^3(f, g) = \sum_{i=1}^{n+1} K_{i, X_{n+1}}^3 (f, g), \quad \forall f, g \in X_{n+1},$$

we see that K^3 is positive definite on X_{n+1}. $\square$

Corollary 1 *Our kernels K_k^2, K^3 and $\hat{K}^3$ are discriminative. That is, given any two strings f, g in the domain of K, as long as $f \neq g$, we have $D_K(f, g) > 0$. Here K stands for any of the three kernels.*

2.3 First Application: Peptide Affinities Prediction

We first briefly review the RLS algorithm inspired by learning theory. Let K be a kernel on a finite set X. Write $\mathscr{H}_K$ to denote the inner product space of functions on X defined by K. Suppose $\bar{z} = \{(x_i, y_i)\}_{i=1}^m$ is a sample set (called the training set) with $x_i \in X$ and $y_i \in \mathbb{R}$ for each i. The RLS uses a positive parameter $\lambda > 0$ and $\bar{z}$ to generate the output function $f_{\bar{z}, \lambda} : X \to \mathbb{R}$, defined as

$$f_{\bar{z}, \lambda} = \arg \min_{f \in \mathscr{H}_K} \left\{ \frac{1}{\#\bar{z}} \sum_{(x_i, y_i) \in \bar{z}} \left(f(x_i) - y_i\right)^2 + \lambda \|f\|_K^2 \right\}. \tag{7}$$

Since $\mathscr{H}_K$ is of finite dimension, one solves (7) by representing f linearly by functions K_x with $x \in X$ and finding the coefficients. See [7, 38] for details.

Remark 7 The RLS algorithm (7) is independent of the choice of the underlying space X where the function space $\mathscr{H}_K$ is defined, in the sense that the predicted values $f_{\bar{z},\lambda}(x)$ at $x \in X$ will not be changed if we extend K onto a large set $X' \supset X$ and re-run (7) with the same $\bar{z}$ and λ. This is guaranteed by the construction of the solution. See, e.g. [7, 38].

Five-fold cross-validation is a procedure to evaluate the performance of an algorithm. Suppose $\bar{z} = \{(x_i, y_i)\}_{i=1}^m$ is partitioned into five divisions (we assume $m \geq 5$, which is always the case in this paper). Five-fold cross-validation is the procedure that validates an algorithm (with fixed parameters) as follows. We choose one of the five divisions of the data for testing, train the algorithm on the remaining four divisions, and predict the output function on the testing division. We do this test for five rounds so that each division is used in one round as the testing data and thus every sample x_i is labeled with both y_i from data and the predicted value $\tilde{y}_i$. The algorithm performance is obtained by comparing the two values over all the sample set. Similarly one defines the n-fold cross-validation for any $n \leq m$. As an important special instance, the m-fold case is also referred to as leave-one-out cross-validation. Cross-validations are also used to tune parameters.

One important step of RLS is parameter selection. As for KernelRLS, we have two parameters, the power β used to define our kernel, and λ in (7). They are selected from an optional set Λ, by leave-one-out cross-validation on the training data. We never use testing data for parameter selection which is under the risk of over-fitting. We refer to the whole procedure, both selecting the parameter, and finding the minimizer of (7) with the best parameter, as the training of KernelRLS. (A similar notion is used also for KernelRLSPan, which will be studied later.)

Binding affinity measures the strength that a peptide binds to an MHC II molecule, and is represented by the IC50 score (see [21] for more details). Usually an IC50 score lies between 0 and 50,000 (nano molar). A widely used IC50 threshold determining binding and non-binding is 500 ("binding" if the IC50 value is less than 500). The bioinformatics community usually normalizes the scores by the function $\psi_b : (0, +\infty) \to [0, 1]$ with a base $b > 1$,

$$\psi_b(x) := \begin{cases} 0 & x > b, \\ 1 - \log_b x & 1 \leq x \leq b, \\ 1 & x < 1. \end{cases} \tag{8}$$

Without introducing any ambiguity we will in the sequel refer to the normalized IC50 value as the binding affinity using an appropriate value of b.

We evaluate KernelRLS on the IEDB benchmark data set published in [30]. The data set covers 14 DRB alleles, each allele a with a set $\mathscr{P}_a$ of peptides. For any $p \in \mathscr{P}_a$, its sequence representation and the $[0, 1]$-valued binding affinity $y_{a,p}$ to the molecule a are both given. On this data set we compare KernelRLS with the state-of-the-art NN-align algorithm proposed in [29]. In [29] for each allele a, the peptide set $\mathscr{P}_a$ was divided into five parts for validating the performance.[4]

---

[4]Both the data set and the 5-fold partition are available at http://www.cbs.dtu.dk/suppl/immunology/NetMHCII-2.0.php.

Now fix an allele a. Set $X = \mathscr{P} \supset \mathscr{P}_a$ (Remark 7 shows that one may select any finite $\mathscr{P}$ that contains $\mathscr{P}_a$ here). Define the kernel $\hat{K}^3$ on X through the steps in the Introduction (leaving the power index β to be fixed). We use the same 5-fold partition $\mathscr{P}_a = \bigcup_{t=1}^5 \mathscr{P}_{a,t}$ as in [30], and use five-fold cross-validation to test KernelRLS (7) with $K = \hat{K}^3$. In the tth test ($t = 1, \ldots, 5$) four parts of $\mathscr{P}_a$ are merged to be the training data, denoted as $\mathscr{P}_a^{(t)} = \mathscr{P}_a \backslash \mathscr{P}_{a,t}$, and $\mathscr{P}_{a,t}$ is left as the testing data. For fixed t and a, we train KernelRLS on $\mathscr{P}_a^{(t)}$ with Λ being the product space of the geometric sequence $\{0.001, \ldots, 10\}$ of length 30 (for β), and the geometric sequence $\{e^{-17}, \ldots, e^{-3}\}$ of length 15 (for λ). After five rounds of tests on allele a, for each $t = 1, \ldots, 5$, each peptide p in the division $\mathscr{P}_{a,t}$ has a predicted affinity $f_{\mathscr{P}_a^{(t)}, \lambda_a^{(t)}, \beta_a^{(t)}}(p)$. Since $\mathscr{P}_a = \cup_{t=1}^5 \mathscr{P}_{a,t}$, we combine all these predictions and denote by $\tilde{y}_{a,p}$ the predicted affinity for each $p \in \mathscr{P}_a$.

The RMSE score is therefore evaluated as

$$\mathrm{RMSE}_a = \sqrt{\frac{1}{\#\mathscr{P}_a} \sum_{p \in \mathscr{P}_a} (\tilde{y}_{a,p} - y_{a,p})^2}.$$

A smaller RMSE score indicates a better algorithm performance. Since the affinity labels in this data set are transformed with $\psi_{b=50,000}$, there is a threshold $\theta = \psi_{50,000}(500) \approx 0.426$ in [29] dividing the peptides $p \in \mathscr{P}_a$ into "binding" if $y_{a,p} > \theta$ and "non-binding" otherwise, to the molecule a. Denote $\mathscr{P}_{a,B} = \{p \in \mathscr{P}_a : y_{a,p} > \theta\}$ and $\mathscr{P}_{a,N} = \mathscr{P}_a \backslash \mathscr{P}_{a,B}$. Then the AUC index is defined to be

$$\mathrm{AUC}_a = \frac{\#\{(p, p') : p \in \mathscr{P}_{a,B},\ p' \in \mathscr{P}_{a,N},\ \tilde{y}_{a,p} > \tilde{y}_{a,p'}\}}{(\#\mathscr{P}_{a,B})(\#\mathscr{P}_{a,N})} \in [0, 1]. \tag{9}$$

A higher AUC indicates a better performance.

The sequence of ideas leads to Table 1. We simply take the weighted average of all the optimal β's,

$$\beta_{\mathrm{peptide}}^* := \frac{1}{\sum_a \#\mathscr{P}_a} \sum_a \left\{ (\#\mathscr{P}_a) \left(\frac{1}{5} \sum_{t=1}^5 \beta_a^{(t)} \right) \right\} = 0.11387, \tag{10}$$

and use it to define kernel for peptides in the next section.

Remark 8 We take the point of view that peptide binding is a matter of degree and hence is better measured by a real number, rather than the binding–non-binding dichotomy. Thus RMSE is a better measure than AUC. The results in Table 1 also demonstrate that the regression-based learning model works well.

The following issue is not standard in the subject and is not intended to yield predictions. It is a preliminary step in giving some confidence to the peptide binding results dealt with in this paper. We are concerned with the "well-posedness" of the problem and give a context to the question: if the input (peptide) varies by a small amount, then does the output (predicted binding function) vary by a small amount?

Table 2 The module of continuity of the predicted values

Allele a	Ω_a	Allele a	Ω_a	Allele a	Ω_a
DRB1*0101	1.7047	DRB1*0301	1.2950	DRB1*0401	1.4663
DRB1*0404	1.2905	DRB1*0405	1.0745	DRB1*0701	1.4940
DRB1*0802	1.2767	DRB1*0901	1.5940	DRB1*1101	1.3537
DRB1*1302	0.9970	DRB1*1501	1.3017	DRB3*0101	1.0696
DRB4*0101	1.4039	DRB5*0101	1.4092		

Remark 9 Our philosophy is that there is a kernel structure on the set of amino acid sequences related to their biological functions (e.g. the correspondent distances on peptides relates to their affinities to each MHC II molecule). The kernel should not depend on the alignment information, which is a source of noise. The performance of our kernel $\hat{K}^3$ is reflected in the modulus of continuity of the predicted values, namely,

$$\Omega_a := \max_{p, p' \in \mathscr{P}_a} \frac{|\tilde{y}_{a,p} - \tilde{y}_{a,p'}|}{d(p, p')},$$

where

$$d(p, p') = \|\hat{K}_p^3 - \hat{K}_{p'}^3\|_{\hat{K}^3} = \sqrt{2 - 2\hat{K}^3(p, p')}$$

is the distance in the space $\mathscr{H}_{\hat{K}^3}$ on peptides, and the kernel $\hat{K}^3$ is defined with $\beta = \beta_{\text{peptide}}^*$. We list the values of Ω_a for the 14 alleles in Table 2.

The modulus of continuity can be extended to a bigger peptide set $\mathscr{P}'$ which contains the neighborhood of each peptide $p \in \mathscr{P}$ with respect to the metric d.

3 Kernel Algorithm for pan-Allele Binding Prediction

We now define a pan-allele kernel on the product space of alleles and peptides. The binding affinity data is thus a subset of this product space. The main motivation is that by the pan-allele kernel we predict affinities to those MHC II molecules with few or no binding data available: this is often the case because the MHC II alleles form a huge set (the phenomenon is often referred to as MHC II polymorphism), and the job of determining experimentally peptide affinities to all the alleles is immense. Also, in the pan-allele setting, one puts the binding data to different molecules together to train the RLS. This makes the training data set larger than that was available in the fixed allele setting, and thus helps to improve the algorithm performance. This is verified in Table 4.

Let $\mathscr{L}$ be a finite set of amino acid sequences representing the MHC II alleles. Using a positive parameter β_{allele} we define a kernel $\hat{K}_{\mathscr{L}}^3$ on $\mathscr{L}$ following the steps in the Introduction. Let $\mathscr{P}$ be a set of peptides. In the sequel we denote by β_{peptide}

Table 3 The performance of KernelRLSPan. For comparison we list the AUC scores of NetMHCIIpan-2.0 [31]. The best AUC in each row is marked in bold

| Allele, a | #$\mathscr{P}_a$ | KernelRLSPan | | NetMHCIIpan-2.0 |
		RMSE	AUC	AUC
DRB1*0101	7685	0.20575	0.84308	**0.846**
DRB1*0301	2505	0.18154	0.85095	**0.864**
DRB1*0302	148	0.21957	0.71176	**0.757**
DRB1*0401	3116	0.19860	0.84294	**0.848**
DRB1*0404	577	0.21887	0.80931	**0.818**
DRB1*0405	1582	0.17459	**0.86862**	0.858
DRB1*0701	1745	0.17769	**0.87664**	0.864
DRB1*0802	1520	0.18732	**0.78937**	0.780
DRB1*0806	118	0.23091	0.89214	**0.924**
DRB1*0813	1370	0.18132	**0.88803**	0.885
DRB1*0819	116	0.18823	**0.82706**	0.808
DRB1*0901	1520	0.19741	**0.82220**	0.818
DRB1*1101	1794	0.16022	**0.88610**	0.883
DRB1*1201	117	0.22740	0.87380	**0.892**
DRB1*1202	117	0.23322	0.89440	**0.900**
DRB1*1302	1580	0.19953	0.82298	**0.825**
DRB1*1402	118	0.20715	**0.86474**	0.860
DRB1*1404	30	0.18705	0.64732	**0.737**
DRB1*1412	116	0.26671	**0.89967**	0.894
DRB1*1501	1769	0.19609	**0.82858**	0.819
DRB3*0101	1501	0.15271	0.82921	**0.85**
DRB3*0301	160	0.26467	**0.86857**	0.853
DRB4*0101	1521	0.16355	**0.87138**	0.837
DRB5*0101	3106	0.18833	0.87720	**0.882**
Average		0.20035	0.84109	0.846
Weighted average		0.19015	0.84887	0.849

specifically the parameter used to define the kernel $\hat{K}^3_{\mathscr{P}}$ on $\mathscr{P}$. We define the pan-allele kernel on $\mathscr{L} \times \mathscr{P}$ as

$$\hat{K}^3_{\mathrm{pan}}\big((a, p), (a', p')\big) = \hat{K}^3_{\mathscr{L}}(a, a')\hat{K}^3_{\mathscr{P}}(p, p'). \tag{11}$$

Let there be given a set of data $\{(p_i, a_i, r_i)\}_{i=1}^m$. Then for each i, $a_i \in \mathscr{L}$, $p_i \in \mathscr{P}$, and $r_i \in [0, 1]$ is the binding affinity of p_i to a_i. The RLS is applied as in Sect. 2. The output function $F : \mathscr{L} \times \mathscr{P} \to \mathbb{R}$ is the predicted binding affinity.

Remark 10 When we choose $\mathscr{L} = \{a\}$ for a certain allele a, the setting and the algorithm reduce to the fixed allele version studied in Sect. 2.

Table 4 The performance of KernelRLSPan on the fixed allele data. For defining AUC, the transform $\psi_{50,000}$ is used as in Table 1

Allele, a	RMSE	AUC	allele, a	RMSE	AUC
DRB1*0101	0.17650	0.86961	DRB1*0301	0.16984	0.85601
DRB1*0401	0.20970	0.82359	DRB1*0404	0.17240	0.88193
DRB1*0405	0.18425	0.84078	DRB1*0701	0.17998	0.90231
DRB1*0802	0.16734	0.88496	DRB1*0901	0.23562	0.71057
DRB1*1101	0.17073	0.91022	DRB1*1302	0.23261	0.75960
DRB1*1501	0.21266	0.80724	DRB3*0101	0.16011	0.79778
DRB4*0101	0.18751	0.84754	DRB5*0101	0.18904	0.89585

Average: RMSE 0.18916, AUC 0.84200

Weighted average: RMSE 0.18496, AUC 0.85452

We test the pan-allele kernel with RLS (we call the algorithm "KernelRLSPan") on Nielsen's NetMHCIIpan-2.0 data set (we also denote by this name the algorithm published in [31] with the data set), which contains 33,931 peptide-allele pairs. For peptides, amino acid sequences are given, and for alleles, DRB names are given so that we can find out the sequence representations in $\mathcal{N}$ as defined in Sect. 1.2. Each pair is labeled with a [0, 1]-valued binding affinity. There are 8083 peptides and 24 alleles in $\mathcal{N}$ in total that appear in these peptide-allele pairs. The whole data set is divided into 5 parts in [31].[5]

We choose the following setting. Let $\mathscr{L} = \mathcal{N}$ and $\mathscr{P}$ be a peptide set large enough to contain all the peptides in the data set. We use $\beta_{\mathsf{peptide}}^{*} = 0.11387$ as suggested in (10) to construct $\hat{K}_{\mathscr{P}}^{3}$ and leave the power index β_{allele} for $\hat{K}_{\mathcal{N}}^{3}$ to be fixed later. This defines $\hat{K}_{\mathsf{pan}}^{3}$. We test the RLS algorithm by five-fold cross-validation according to the 5-part division in [31]. In each test we merge four parts of the samples as the training data and leave the other part as the testing data. We train KernelRLSPan with the optional parameter set $\Lambda = \{(\beta_{\mathsf{allele}}, \lambda) : \beta_{\mathsf{allele}} \in \{0.02 \times n : n = 1, 2, \ldots, 8\}, \lambda \in \{e^{n} : n = -17, -16, \ldots, -9\}\}$. The procedures are the same as used in Sect. 2.3 except we now do cross-validation for the peptide-allele pairs. In all the five tests, the pair $\beta_{\mathsf{allele}} = 0.06$ and $\lambda = e^{-13}$ uniformly achieves the best performance in the training data. We now use the threshold $\theta = \psi_{15,000}(500) \approx 0.3537$ to evaluate the AUC score, because the affinity values in the data set are obtained by the transform $\psi_{15,000}$. The results of these computations are shown in Table 3, where we see that the two algorithms are comparable. We do not assert on this test that the ours is better.

We implement KernelRLSPan on the fixed allele data set used in Table 1. Recall that the data set is normalized with $\psi_{50,000}$ and has the five-fold division defined by [30]. The performance is listed in Table 4, which is better than that of KernelRLS as listed in Table 1.

---

[5]Both the data set and the 5-part partition are available at http://www.cbs.dtu.dk/suppl/immunology/NetMHCIIpan-2.0.

Next, we use the whole NetMHCIIpan-2.0 data set for training, and test the algorithm performance on a new data set. A set of 64798 triples of MHC II–peptide binding data is downloaded from IEDB.[6] We pick from the set the DRB alleles, having IC50 scores, and having explicit allele names and peptide sequences. Those items that also appear in the NetMHCIIpan-2.0 data set are deleted. For the duplicated items (same peptide-allele pair and same affinity) only one of them are kept. All the pieces with the same peptide-allele pair yet different affinities are deleted. We deleted those with peptide length less than 9. (The KernelRLSPan can handle these peptides, while the NetMHCIIpan-2.0 cannot. The short peptides therefore are deleted to make the two algorithms comparable.) For some alleles the data in the set is insufficient to define the AUC score (i.e. the denominator in (9) becomes zero), so we delete tuples containing them. Eventually we obtained 11334 peptide-allele pairs labeled with IC50 binding affinities, which are further normalized by $\psi_{15,000}$ as in the NetMHCIIpan-2.0 data set.

Now define $\hat{K}_{\mathsf{pan}}^3$ on $\mathcal{N} \times \mathcal{P}$ as in (11) with $\beta_{\mathsf{allele}} = 0.06$ as suggested by the above computation and $\beta_{\mathsf{peptide}} = 0.11387$ as suggested in (10). We train on the NetMHCIIpan-2.0 data set both KernelRLSPan and NetMHCIIpan-2.0.[7] We train KernelRLSPan with the optional parameter set $\Lambda = \{\lambda = \mathrm{e}^n, n = -18, \ldots, -8\}$ (since other parameters are fixed). (The result shows that $\lambda = \mathrm{e}^{-13}$ uniformly performs best.) The algorithm performance of the two algorithms are compared on Table 5.

In this section KernelRLSPan is tested. Tables 3, 4, and 5 suggest that compared with KernelRLS, KernelRLSPan performs much better. Also, the kernel method uses only the substitution matrix and the sequence representations without direct alignment information but yields comparable performance with the state-of-the-art NetMHCIIpan-2.0 algorithm.

4 Clustering and Supertypes

In this section, we describe in detail the construction of our cluster tree and our classification of DRB alleles into supertypes. We compare the supertypes identified by our model with the serotypes designated by WHO and analyze the comparison results in detail.

4.1 Identification of DRB Supertypes

We classify DRB alleles into disjoint subsets by using DRB amino acid sequences and the BLOSUM62 substitution matrix. No peptide binding data or X-ray 3D structure data are used in our clustering. We obtain a classification in this way into subsets (a partition) which we call supertypes.

---

[6]The data set was downloaded from http://www.immuneepitope.org/list_page.php?list_type=mhc&measured_response=&total_rows=64797&queryType=true, on May 23, 2012.

[7]The code is published in http://www.cbs.dtu.dk/cgi-bin/nph-sw_request?netMHCIIpan.

Table 5 The performance of KernelRLSPan and NetMHCIIpan-2.0 trained on the NetMHCIIpan-2.0 benchmark data set, tested on a new data set downloaded from the IEDB. The best performance of both AUC and RMSE scores of each row is marked in bold

Allele, a	$\#\mathscr{P}_a$	kernelRLSpan		NetMHCIIpan-2.0	
		RMSE	AUC	RMSE	AUC
DRB1*0101	1024	0.25519	0.79717	**0.24726**	**0.82988**
DRB1*0102	7	**0.39748**	**0.58333**	0.62935	**0.58333**
DRB1*0103	41	0.33159	**0.83333**	**0.32204**	**0.83333**
DRB1*0301	883	**0.21760**	0.80276	0.23975	**0.82384**
DRB1*0401	1122	0.19610	0.79930	**0.19363**	**0.82456**
DRB1*0402	48	**0.23912**	**0.67321**	0.27352	0.65714
DRB1*0403	43	0.16381	**0.70443**	**0.15868**	0.66995
DRB1*0404	494	0.21689	0.79344	**0.20219**	**0.82517**
DRB1*0405	462	0.19617	0.78941	**0.19387**	**0.80611**
DRB1*0406	14	0.19516	0.53846	**0.19497**	**0.61538**
DRB1*0701	724	0.20853	0.80876	**0.20039**	**0.84786**
DRB1*0801	24	0.37281	**0.72500**	**0.34767**	0.71250
DRB1*0802	404	0.17403	0.80407	**0.17181**	**0.81085**
DRB1*0901	335	0.21204	0.79524	**0.21029**	**0.80489**
DRB1*1001	20	0.28082	0.74000	**0.24335**	**0.92000**
DRB1*1101	811	0.24195	0.83219	**0.23838**	**0.85071**
DRB1*1104	10	**0.43717**	**0.76190**	0.57082	0.57143
DRB1*1201	795	0.25786	**0.83178**	**0.24984**	0.82685
DRB1*1301	147	**0.27014**	0.65077	0.30202	**0.70722**
DRB1*1302	499	0.22194	0.82118	**0.21284**	**0.84258**
DRB1*1501	856	0.21580	0.83563	**0.20869**	**0.84902**
DRB1*1502	3	**0.013186**	**1.00000**	0.20061	**1.00000**
DRB1*1601	16	0.19556	**0.84615**	**0.18740**	0.76923
DRB1*1602	12	0.32238	**0.68571**	**0.30431**	0.60000
DRB3*0101	437	**0.16568**	0.74058	0.17860	**0.77182**
DRB3*0202	750	**0.16021**	0.82543	0.16453	**0.84191**
DRB4*0101	563	**0.20594**	**0.80575**	0.21383	0.78734
DRB5*0101	774	0.25934	0.78701	**0.25849**	**0.81950**
DRB5*0202	16	**0.23013**	**0.71429**	0.40554	0.57143
Average		**0.24046**	0.76987	0.25947	**0.77151**
Weighted average		0.21853	0.80309	**0.21816**	**0.82216**

In Sect. 3, we have defined the allele kernel on $\mathscr{N}$ as $\hat{K}^3_{\mathscr{N}}$; the L^2 distance derived from $\hat{K}^3_{\mathscr{N}}$ is defined as

$$D_{L^2}(x, y) = \left(\frac{1}{\#\mathscr{N}} \sum_{z \in \mathscr{N}} \left(\hat{K}^3(x, z) - \hat{K}^3(y, z) \right)^2 \right)^{1/2}, \quad \forall x, y \in \mathscr{N}.$$

The authors have found very generally that the L^2 distance is preferable to the simple kernel distance and the two give different results. For one thing the L^2 distance exploits a measure, perhaps even uniform. But this is a long story and this is not the place for that story.

The OWA-based linkage, defined as follows is used to measure the proximity between clusters X and Y.[8] Let $U = (d_{xy})_{x \in X, y \in Y}$, where $d_{xy} = D_{L^2}(x, y)$. After ordering (with repetitions) the elements of U in descending order, we obtain an ordered vector $V = (d'_1, \ldots, d'_n), n = |U|$. A weighting vector $W = (w_1, \ldots, w_n)$ is associated with V, and the proximity between clusters X and Y is defined as

$$D_{\mathsf{OWA}}(X, Y) = \sum_{i=1}^{n} w_i d'_i.$$

Here the weights W are defined as follows [34]:

$$w'_i = \frac{e^{i/\mu}}{\mu}, \quad i = 1, 2, \ldots, n,$$

$$w_i = \frac{w'_i}{\sum_{j=1}^{n} w'_j}, \quad i = 1, 2, \ldots, n,$$

where $\mu = \gamma(1 + n)$ and γ is chosen to be 0.1 as suggested by experiments and the work of [34]. This weighting gives more importance to pairs (x, y) which have smaller distance.

Hierarchical clustering [8] is applied to build a cluster tree. A cluster tree is a tree on which every node represents the cluster of the set of all leaves descending from that node. The L_2 distance D_{L^2} is used to measure the distance between alleles x and y, $x, y \in \mathcal{M}$ and OWA-based linkage is used to measure the proximity between clusters X and Y, $X, Y \subseteq \mathcal{M}$ instead of "single" linkage. This algorithm is a bottom-up approach. At the beginning, each allele is treated as a singleton cluster, and then successively one merges two nearest clusters X and Y into a union cluster, the process stopping when all unions of clusters have been merged into a single cluster.

This cluster tree, associated to $\mathcal{M}$, has thus 559 leaves. We cut the cluster tree at 16 clusters, an appropriate level to separate different families of alleles. The upper part of this tree is shown in Fig. 1. The contents of the clusters are given in Table 6. We assign supertypes to certain clusters in the cluster tree based on the contents of the clusters described in Table 6. A supertype is based on one or two clusters in Table 6. If two clusters in Table 6 are closest in the tree, and the alleles in which are in the same family, they are assigned an identical supertype. Thirteen supertypes are defined in this way, which we name ST1, ST2, ST3, ST4, ST5, ST6, ST7, ST8, ST9, ST10, ST51, ST52, and ST53. The corresponding cluster diameters are 0.11, 0.13, 0.15, 0.14, 0.11, 0.18, 0.08, 0.14, 0.08, 0.02, 0.09, 0.13, and 0.05, respectively.

The diameter of a cluster Z is defined as

$$\text{diameter}(Z) = \max_{x, y \in Z} D_{L^2}(x, y). \tag{12}$$

---

[8] Another way of measuring distance between clusters is the Hausdorff distance.

Table 6 Overview of clusters of HLA-DR alleles with split serological types assigned by WHO

Super-type	Allele	Sero-type	Allele	Sero-type	Allele	Sero-type
ST52	Cluster 1					
	DRB3*0101(2)	DR52	DRB3*0108(U.)	–	DRB3*0212(U.)	–
	DRB3*0106(s.s.)	DR52	DRB3*0102(s.s.)	–	DRB3*0226	–
	DRB3*0110(s.s.)	DR52	DRB3*0112	–	DRB3*0222(U.)	–
	DRB3*0301	DR52	DRB3*0105(U.)	–	DRB3*0204(U.)	–
	DRB3*0209	DR52	DRB3*0103(s.s.)(U.)	–	DRB3*0213(U.)	–
	DRB3*0302(s.s.)	DR52	DRB3*0113	–	DRB3*0215(U.)	–
	DRB3*0107(s.s.)	DR52	DRB3*0111(U.)	–	DRB3*0218(U.)	–
	DRB3*0203(s.s.)	DR52	DRB3*0114	–	DRB3*0205(U.)	–
	DRB3*0211	DR52	DRB3*0303	–	DRB3*0225	–
	DRB3*0201(2)	DR52	DRB3*0109(U.)	–	DRB3*0219(U.)	–
	DRB3*0202(2)	DR52	DRB3*0206(s.s.)	–	DRB3*0216(U.)	–
	DRB3*0210	DR52	DRB3*0220(U.)	–	DRB3*0221(U.)	–
	DRB3*0208(s.s.)	DR52	DRB3*0223	–	DRB3*0227	–
	DRB3*0207(s.s.)	DR52	DRB3*0217(U.)	–		
	DRB1*0338	–	DRB3*0214(U.)	–		
ST3	Cluster 2					
	DRB1*0323	DR3	DRB1*0334	–	DRB1*0358	–
	DRB1*0301(2)	DR17	DRB1*0364	–	DRB1*0308	–
	DRB1*0305	DR3	DRB1*0361	–	DRB1*0326	–
	DRB1*0311	DR17	DRB1*0332	–	DRB1*0313	–
	DRB1*0304	DR17	DRB1*0328	–	DRB1*0360	–
	DRB1*0306	DR3	DRB1*0362	–	DRB1*0324	–
	DRB1*0307	DR3	DRB1*0346	–	DRB1*0352	–
	DRB1*0314	DR3	DRB1*0336	–	DRB1*0365	–
	DRB1*0315	DR3	DRB1*0357	–	DRB1*0329	–
	DRB1*0312(s.s.)	DR3	DRB1*0339	–	DRB1*0327	–
	DRB1*0302	DR18	DRB1*0333	–	DRB1*0353	–
	DRB1*0303	DR18	DRB1*0319	–	DRB1*0321	–
	DRB1*0310	DR17	DRB1*0348	–	DRB1*0343	–
	DRB1*0342	–	DRB1*0363	–	DRB1*0330	–
	DRB1*0345	–	DRB1*0322	–	DRB1*0325	–
	DRB1*0355	–	DRB1*0309	–	DRB1*0344	–
	DRB1*0359	–	DRB1*0337	–	DRB1*0331	–
	DRB1*0354	–	DRB1*0351	–	DRB1*0335	–
	DRB1*0320	–	DRB1*0347	–	DRB3*0115	–
	DRB1*0356	–	DRB1*0318	–	DRB1*0316(s.s.)	–
	Cluster 3					
	DRB1*1525	–	DRB1*0340	–	DRB1*0317	–
	DRB1*0349	–	DRB1*0341	–		

Table 6 (*Continued*)

Super-type	Allele	Sero-type	Allele	Sero-type	Allele	Sero-type
ST6	Cluster 4					
	DRB1*1410	DR14	DRB1*1482	–	DRB1*1472	–
	DRB1*1401(4)	DR14	DRB1*1462	–	DRB1*14101	–
	DRB1*1426	DR14	DRB1*1470	–	DRB1*1434	–
	DRB1*1407	DR14	DRB1*1438	–	DRB1*1423	–
	DRB1*1460	DR14	DRB1*14112	–	DRB1*1445	–
	DRB1*1450	DR14	DRB1*1490	–	DRB1*1443	–
	DRB1*1404	DR1404	DRB1*1486	–	DRB1*1456	–
	DRB1*1449	DR14	DRB1*1497	–	DRB1*14103	–
	DRB1*1411	DR14	DRB1*1435	–	DRB1*1444	–
	DRB1*1408	DR14	DRB1*1455	–	DRB1*1496	–
	DRB1*1414	DR14	DRB1*1431	–	DRB1*14100	–
	DRB1*1405	DR14	DRB1*1493	–	DRB1*1436	–
	DRB1*1420	DR14	DRB1*1428	–	DRB1*1465	–
	DRB1*1422	DR14	DRB1*1471	–	DRB1*1464	–
	DRB1*1416	DR6	DRB1*1468	–	DRB1*1495	–
	DRB1*1439	–	DRB1*1432	–	DRB1*1459	–
	DRB1*1499	–	DRB1*14111	–	DRB1*1491	–
	DRB1*1461	–	DRB1*14104	–	DRB1*1441	–
	DRB1*14117	–	DRB1*1458	–	DRB1*1437	–
	DRB1*1487	–	DRB1*1473	–	DRB1*1457	–
	DRB1*1475	–	DRB1*1479	–	DRB1*14105	–
	DRB1*1488	–	DRB1*14107	–	DRB1*1474	–
	DRB1*14110	–	DRB1*1476	–		
	Cluster 5					
	DRB1*1419	DR14	DRB1*1452	–	DRB1*1433	–
	DRB1*1402	DR14	DRB1*14108	–	DRB1*1424	–
	DRB1*1429	DR14	DRB1*1483	–	DRB1*14109	–
	DRB1*1406	DR14	DRB1*1481	–	DRB1*14115	–
	DRB1*1418	DR6	DRB1*1494	–	DRB1*1467	–
	DRB1*1413	DR14	DRB1*1447	–	DRB1*1498	–
	DRB1*1421	DR14	DRB1*1451	–	DRB1*1463	–
	DRB1*1417	DR6	DRB1*14106	–	DRB1*1485	–
	DRB1*1427	DR14	DRB1*1489	–	DRB1*1478	–
	DRB1*1403	DR1403	DRB1*1430	–	DRB1*1448	–
	DRB1*1412	DR14	DRB1*1409	–		
	DRB1*1446	–	DRB1*1480	–		
ST8	Cluster 6					
	DRB1*1442(U.)	–				
	Cluster 7					

Table 6 (*Continued*)

Super-type	Allele	Sero-type	Allele	Sero-type	Allele	Sero-type
	DRB1*0809	DR8	DRB1*1477	–	DRB1*0808	–
	DRB1*1415	DR8	DRB1*1440	–	DRB1*0844	–
	DRB1*0814	DR8	DRB1*1484	–	DRB1*0835	–
	DRB1*0812	DR8	DRB1*0846	–	DRB1*0836	–
	DRB1*0803	DR8	DRB1*0848	–	DRB1*0847	–
	DRB1*0810	DR8	DRB1*0819	–	DRB1*0825	–
	DRB1*0817	DR8	DRB1*0827	–	DRB1*0834	–
	DRB1*0811	DR8	DRB1*0829	–	DRB1*0828	–
	DRB1*0801	DR8	DRB1*0837	–	DRB1*0845	–
	DRB1*0807	DR8	DRB1*0839	–	DRB1*0830	–
	DRB1*0806	DR8	DRB1*0822	–	DRB1*0824	–
	DRB1*0805	DR8	DRB1*0815	–	DRB1*0820(U.)	–
	DRB1*0818	DR8	DRB1*0840	–	DRB1*14116	–
	DRB1*0816	DR8	DRB1*0838	–	DRB1*14102	–
	DRB1*0802	DR8	DRB1*0826	–	DRB1*0842	–
	DRB1*0804	DR8	DRB1*0843	–	DRB1*0841	–
	DRB1*0813	DR8	DRB1*0833	–	DRB1*1425	–
	DRB1*0821	–	DRB1*0823	–	DRB1*1469	–
ST4	Cluster 8					
	DRB1*0420(s.s.)	DR4	DRB1*0438	–	DRB1*0490	–
	DRB1*0401	DR4	DRB1*0434	–	DRB1*0487	–
	DRB1*0464	DR4	DRB1*0475	–	DRB1*0430	–
	DRB1*0408	DR4	DRB1*0476	–	DRB1*0448	–
	DRB1*0416	DR4	DRB1*0472	–	DRB1*0467	–
	DRB1*0426	DR4	DRB1*0435	–	DRB1*0483	–
	DRB1*0442	DR4	DRB1*0443	–	DRB1*0480	–
	DRB1*0432(s.s.)	DR4	DRB1*0479	–	DRB1*0462	–
	DRB1*0423	DR4	DRB1*0440	–	DRB1*0457	–
	DRB1*0404	DR4	DRB1*0470	–	DRB1*0497	–
	DRB1*0413	DR4	DRB1*0444	–	DRB1*0463	–
	DRB1*0431	DR4	DRB1*0456	–	DRB1*0498	–
	DRB1*0403	DR4	DRB1*0455	–	DRB1*0449	–
	DRB1*0407(2)	DR4	DRB1*0433	–	DRB1*04102	–
	DRB1*0429	DR4	DRB1*0439	–	DRB1*0441	–
	DRB1*0424	DR4	DRB1*0460	–	DRB1*0446	–
	DRB1*0409	DR4	DRB1*0450	–	DRB1*0485	–
	DRB1*0405	DR4	DRB1*0496	–	DRB1*0478	–
	DRB1*0410	DR4	DRB1*0451	–	DRB1*0465	–
	DRB1*0428	DR4	DRB1*0471	–	DRB1*0491	–
	DRB1*0417	DR4	DRB1*04100	–	DRB1*0468	–

Table 6 (*Continued*)

Super-type	Allele	Sero-type	Allele	Sero-type	Allele	Sero-type
	DRB1*0411	DR4	DRB1*0488	–	DRB1*0477	–
	DRB1*0422	DR4	DRB1*0493	–	DRB1*0484	–
	DRB1*0406	DR4	DRB1*0427	–	DRB1*0447	–
	DRB1*0421	DR4	DRB1*0452	–	DRB1*0436	–
	DRB1*0419	DR4	DRB1*04101	–	DRB1*0454	–
	DRB1*0425(s.s.)	DR4	DRB1*0474	–	DRB1*0437	–
	DRB1*0414	DR4	DRB1*0495	–	DRB1*0453	–
	DRB1*0402	DR4	DRB1*0459	–	DRB1*0418	–
	DRB1*0415	DR4	DRB1*0473	–	DRB1*0458	–
	DRB1*0499	–	DRB1*0461	–	DRB1*0486	–
	DRB1*0482	–	DRB1*0445	–	DRB1*0412	–
	DRB1*0466	–	DRB1*0489	–	DRB1*0469	–
ST2	Cluster 9					
	DRB1*1501(2)	DR15	DRB1*1533	–	DRB1*1548	–
	DRB1*1505	DR15	DRB1*1553	–	DRB1*1512	–
	DRB1*1506	DR15	DRB1*1524	–	DRB1*1515	–
	DRB1*1503	DR15	DRB1*1509	–	DRB1*1557	–
	DRB1*1508	DR2	DRB1*1549	–	DRB1*1511	–
	DRB1*1502(2)	DR15	DRB1*1541	–	DRB1*1538	–
	DRB1*1504	DR15	DRB1*1540	–	DRB1*1529	–
	DRB1*1507	DR15	DRB1*1523	–	DRB1*1545	–
	DRB1*1602	DR16	DRB1*1518	–	DRB1*1554	–
	DRB1*1605(s.s.)	DR16	DRB1*1537	–	DRB1*1510	–
	DRB1*1601	DR16	DRB1*1514	–	DRB1*1521	–
	DRB1*1609	DR16	DRB1*1544	–	DRB1*1612	–
	DRB1*1603	DR2	DRB1*1526	–	DRB1*1617	–
	DRB1*1604	DR16	DRB1*1539	–	DRB1*1611	–
	DRB1*1528	–	DRB1*1530	–	DRB1*1614	–
	DRB1*1535	–	DRB1*1531	–	DRB1*1618	–
	DRB1*1532	–	DRB1*1556	–	DRB1*1610	–
	DRB1*1542	–	DRB1*1555	–	DRB1*1608	–
	DRB1*1551	–	DRB1*1516	–	DRB1*1615	–
	DRB1*1552	–	DRB1*1522	–	DRB1*1607	–
	DRB1*1536	–	DRB1*1546	–	DRB1*1616	–
	DRB1*1520	–	DRB1*1547	–	DRB1*1527	–
	DRB1*1543	–	DRB1*1558	–	DRB1*1534	–
ST5	Cluster 10					
	DRB1*1202	DR12	DRB1*1215	–	DRB1*1230	–
	DRB1*1201(4)	DR12	DRB1*1219	–	DRB1*1207	–
	DRB1*1203	DR12	DRB1*1216	–	DRB1*1229	–

Table 6 (*Continued*)

Super-type	Allele	Sero-type	Allele	Sero-type	Allele	Sero-type
	DRB1*1205	DR12	DRB1*1221	–	DRB1*1234	–
	DRB1*1220	–	DRB1*1208	–	DRB1*1222	–
	DRB1*1233	–	DRB1*1212	–	DRB1*1223	–
	DRB1*1218	–	DRB1*1225	–	DRB1*1227	–
	DRB1*1213	–	DRB1*1211	–	DRB1*1209	–
	DRB1*1232	–	DRB1*1228	–	DRB1*1204	–
	DRB1*1226	–	DRB1*1214	–	DRB1*0832(U.)	–
ST53	Cluster 11					
	DRB4*0101(3)	DR53	DRB4*0104(U.)	–	DRB4*0107(U.)	–
	DRB4*0105(s.s.)	DR53	DRB4*0102(s.s.)(U.)	–	DRB4*0108	–
ST9	Cluster 12					
	DRB1*0901	DR9	DRB1*0912	–	DRB1*0915	–
	DRB1*0905	DR9	DRB1*0906	–	DRB1*0911	–
	DRB1*0910	–	DRB1*0908	–	DRB1*0914	–
	DRB1*0916	–	DRB1*0904	–	DRB5*0112(U.)	–
	DRB1*0907	–	DRB1*0903	–	DRB1*0902	–
	DRB1*0909	–	DRB1*0913	–		
ST7	Cluster 13					
	DRB1*0703	DR7	DRB1*0721	–	DRB1*0708	–
	DRB1*0701	DR7	DRB1*0716	–	DRB1*0711	–
	DRB1*0709	DR7	DRB1*0713	–	DRB1*0717	–
	DRB1*0704	DR7	DRB1*0714	–	DRB1*0707	–
	DRB1*0715	–	DRB1*0712	–	DRB1*0706	–
	DRB1*0719	–	DRB1*0720	–		
	DRB1*0705	–	DRB1*0718	–		
ST51	Cluster 14					
	DRB5*0101	DR51	DRB5*0104(U.)	–	DRB5*0106(U.)	–
	DRB5*0102	DR51	DRB5*0103(U.)	–	DRB5*0111(U.)	–
	DRB5*0107(s.s.)	DR51	DRB5*0113(U.)	–	DRB5*0204(U.)	–
	DRB5*0202	DR51	DRB5*0109(s.s.)(U.)	–	DRB5*0203(U.)	–
	DRB5*0105(U.)	–	DRB5*0114	–	DRB5*0205(U.)	–
ST10	Cluster 15					
	DRB1*1001	DR10	DRB1*1003	–	DRB1*1002	–
ST1	Cluster 16					
	DRB1*0107	DR1	DRB1*0120	–	DRB1*0135	–
	DRB1*0101	DR1	DRB1*0127	–	DRB1*0111	–
	DRB1*0102	DR1	DRB1*0112	–	DRB1*0117	–
	DRB1*0104	DR1	DRB1*0128	–	DRB1*0118	–
	DRB1*0109	DR1	DRB1*0136	–	DRB1*0115	–

Table 6 (*Continued*)

Super-type	Allele	Sero-type	Allele	Sero-type	Allele	Sero-type
	DRB1*0103	DR103	DRB1*0131	–	DRB1*0106	–
	DRB1*0113	DR1	DRB1*0132	–	DRB1*0126	–
	DRB1*0122	–	DRB1*0119	–	DRB1*0137	–
	DRB1*0124	–	DRB1*0130	–	DRB1*0123	–
	DRB1*0110	–	DRB1*0121	–	DRB1*0108	–
	DRB1*0129	–	DRB1*0105	–	DRB1*0114	–
	DRB1*0134	–	DRB1*0125	–	DRB1*0116	–

The DRB alleles in the first ten supertypes are gathered from the DRB1 locus. The DRB alleles in the ST51, ST52 and ST53 supertypes are gathered from the DRB5, DRB3, and DRB4 loci, respectively.

4.2 Serotype Designation of HLA-DRB Alleles

There is a historically developed classification, based on extensive works of medical labs and organizations, that groups alleles into what are called serotypes. This classification is oriented to immunology and diseases associated to gene variation in humans. It uses peptide binding data, 3D structure, X-ray diffraction and other tools. When the confidence level is sufficiently high, WHO assigns a serotype to an allele as in Table 6 where a number prefixed by DR follows the name of that allele.

There are four DRB genes (DRB1/DRB3/DRB4/DRB5) in the HLA-DRB region [20]. The DRB1 gene/locus is much more polymorphic than the DRB3/DRB4/DRB5 genes/loci [5]. More than 800 allelic variants are derived from the exon 2 of the DRB genes in humans [11]. The WHO Nomenclature Committee for Factors of the HLA System assigns an official name for each identified allele sequence, e.g. DRB1*01:01. The characters before the separator "*" describe the name of the gene, the first two digits correspond to the allele family and the third and fourth digits correspond to a specific HLA protein. See Table 6 for examples of how the alleles are named. If two HLA alleles belong to the same family, they often correspond to the same serological antigen, and thus the first two digits are meant to suggest serological types. So for those alleles which are not assigned serotypes by WHO, WHO has suggested serotypes for them according to their official names or allele families.

4.3 Comparison of Identified Supertypes to Designated Serotypes

In Sect. 4.1, we have identified 13 supertypes and in Sect. 4.2 we have introduced the WHO assigned serotypes. In the following, we compare these two classifications.

By using the cluster tree given in Fig. 1 and the contents of the clusters described in Table 6, we have named our supertypes with prefix "ST" paralleled to the serotype names. The detailed information of DRB alleles and serological types for these 13 supertypes is given in Table 6. Our supertype clustering recovers the WHO serotype classification and provides further insight into the classification of DRB alleles which

are not assigned serotypes. There are 559 DRB alleles in Table 6, and only 138 DRB alleles have WHO assigned serotypes. Table 7 gives the relationship between the broad serological types and the split serological types. As shown in Tables 6 and 7, our supertypes assigned to these 138 DRB alleles are in exact agreement with the WHO assigned broad serological types (see Table 7). Extensive medical/biological information was used by WHO to assign serological type whereas solely DRB amino acid sequences were used in our supertype clustering. All alleles with WHO assigned DR52, DR3, DR6, DR8, DR4, DR2, DR5, DR53, DR9, DR7, DR51, DR10, and DR1-serotype are classified, respectively, into the ST52, ST3, ST6, ST8, ST4, ST2, ST5, ST53, ST9, ST7, ST51, ST10, and ST1-supertype. The other 461 alleles in Table 6 are not assigned serotypes by WHO in [17]. However, WHO has suggested serotypes for them according to their official names or allele families; that is, if two DRB alleles are in the same family, they belong to the same serotype. Our clustering confirms that this suggestion is reasonable, as can be checked from the clusters in Table 6.

We make some remarks on Fig. 1 and Table 6 as follows.

ST52: This supertype consists of exactly the DRB3 alleles with the exception of DRB1*0338 (a new allele and unassigned by WHO [17]).

ST3: This supertype consists of cluster 2 and cluster 3 in the cluster tree and contains 63 DRB1*03 alleles with two exceptions: DRB3*0115 and DRB1*1525. The DRB3*0115 is grouped with the DRB1*03 alleles in a number of different experiments done by us, and the DRB1*1525 is a new allele and unassigned by WHO. Here, the DR3-serotype is a broad serotype which consists of three split serotypes, DR3, DR17 and DR18 (see Table 7).

ST6: This supertype consists of cluster 4 and cluster 5 and consists of exactly 102 DRB1*14 alleles. Here, the DR6-serotype is a broad serotype which consists of five split serotypes, DR6, DR13, DR14, DR1403 and DR1404.

ST8: This supertype consists of cluster 6 and cluster 7 and mainly contains 46 DRB1*08 alleles (The serological designation of DRB1*1415 is DR8 by WHO.). The unassigned alleles DRB1*1425, DRB1*1440, DRB1*1442, DRB1*1469, DRB1*1477, and DRB1*1484 are DRB1*14 alleles, but they are classified into the ST8 supertype. Both DRB1*14116 and DRB1*14102 are new allele sequences that do not exist in the tables of [17, 28] and they are classified into the ST8 supertype too.

Supertypes 52, 4, 2, 5, 53, 9, 7, 51, 10, and 1 correspond, respectively, to clusters 1, 8, 9, 10, 11, 12, 13, 14, 15, and 16 in the cluster tree.

ST4: This supertype consists of exactly 99 DRB1*04 alleles.

ST2: This supertype consists of 53 DRB1*15 alleles and 16 DRB1*16 alleles. Here, the DR2-serotype is a broad serotype which consists of three split serotypes, DR2, DR15, and DR16.

ST5: This supertype contains exactly 29 DRB1*12 alleles. The DRB1*0832 is undefined by experts in [17], but its serological designation by the neural network algorithm [27] is DR8 or DR12. We classify it into the ST5 supertype. The DR5-serotype is a broad serotype which consists of two split serotypes, DR11 and DR12.

ST53: This supertype consists of exactly the DRB4 alleles.

ST9: This supertype contains exactly the DRB1*09 alleles with the exception of DRB5*0112. The DRB5*0112 is undefined by experts in [17]. And from a number of

Table 7 Overview of the broad serological types in connection with the split serological types assigned by WHO. The serological type information listed in this table was extracted from the Tables 4 and 5 given in [17]. This table summarizes the allele and serotype information given in the first and third columns of Tables 4 and 5

HLA-DRB1 serological families		
Broad Serotype	Split serotype	Alleles
DR1	DR1	DRB1*01
	DR103	DRB1*0103
DR2	DR2	DRB1*1508, *1603
	DR15	DRB1*15
	DR16	DRB1*16
DR3	DR3	DRB1*0305, *0306, *0307, *0312, *0314, *0315, *0323
	DR17	DRB1*0301, *0304, *0310, *0311
	DR18	DRB1*0302, *0303
DR4	DR4	DRB1*04
DR5	DR11	DRB1*11
	DR12	DRB1*12
DR6	DR6	DRB1*1416, *1417, *1418
	DR13	DRB1*13, *1453
	DR14	DRB1*14, *1354
	DR1403	DRB1*1403
	DR1404	DRB1*1404
DR7	DR7	DRB1*07
DR8	DR8	DRB1*08, *1415
DR9	DR9	DRB1*09
DR10	DR10	DRB1*10

DRB3/4/5 serological families	
Serotype	Alleles
DR51	DRB5*01,02
DR52	DRB3*01,02,03
DR53	DRB4*01

different experiments done by us, DRB5*0112 is clustered with the DRB1*09 family of alleles.

ST7: This supertype consists of exactly 19 DRB1*07 alleles.

ST51: This supertype consists of exactly 15 DRB5 alleles.

ST10: This supertype is the smallest supertype and consists of exactly 3 DRB1*10 alleles.

ST1: This supertype consists of exactly 36 DRB1*01 alleles. Here, the DR1-serotype is a broad serotype which consists of two split serotypes, DR1 and DR103.

For the DRB alleles, there are 13 broad serotypes given by WHO, and our clustering classifies all alleles which are assigned the same broad serotype to the same supertype. For the alleles which are not assigned serotypes, our supertypes confirm the nomenclature of WHO.

As can be seen from Fig. 1, the ST52 supertype is closest to the ST3 supertype. The ST53 supertype is closest to the ST9 and ST7 supertypes. The ST51 supertype is closest to the ST10 and ST1 supertypes.

4.4 Previous Work in Perspective

In 1999, Sette and Sidney asserted that all HLA I alleles can be classified into nine supertypes [39, 43]. This classification is defined based on the structural motifs derived from experimentally determined binding data. The alleles in the same supertype comprise the same peptide binding motifs and bind to largely overlapping sets of peptides. Essentially, the supertype classification problem is to identify peptides that can bind to a group of HLA molecules. Besides many works on HLA class I supertype classification, some works have been proposed to identify supertypes for HLA class II. In 1998, through analyzing a large set of biochemical synthetic peptides and a panel of HLA-DR binding assays, Southwood et al. [45] asserted that seven common HLA-DR alleles, e.g. DRB1*0101, DRB1*0401, DRB1*0701, DRB1*0901, DRB1*1302, DRB1*1501, and DRB5*0101 had similar peptide binding specificity and should be grouped into one supertype. Lund et al. [25] used the position specific scoring matrices (PSSM) from the TEPITOPE method to do a functional clustering of 50 HLA-DR alleles. Both of these studies used peptide binding data and this resulted in the limited number of DRB alleles available for classification. The work of Doytchinova and Flower [9], classified 347 DRB alleles into five supertypes by the use of both protein sequences and 3D structural data. Ou et al. [32] defined seven supertypes based on similarity of function rather than on sequence or structure. To our knowledge, our study is the first to identify HLA-DR supertypes solely based on DRB amino acid sequence data. However, the clustering of HLA-DR molecules into functional groups has been made earlier using the NetMHCIIpan pan-specific prediction methods [30]. This method is not limited to a fixed number of HLA-DR alleles, and can be used to predict the binding specificity of any HLA-DR molecule with known protein sequence, and has been used to correctly classify HLA-DR molecules into groups of molecules with large functional overlap (including the mixed groups of HLA-DRB1*08/HLA-DRB1*11, HLA-DRB1*11/HLA-DRB1*13 molecules).

The split serological types are obtained from [17]. The left column indicates the supertypes defined by the cluster tree. Remark on the labels for the alleles: "(U.)" stands for "undefined" marked by the experts in [17]; "(s.s.)" indicates that the normal forms of the allele is shorter than 81 amino acids; "(n)" with $n = 2, 3, \ldots$ indicates that the normal form is shared by n alleles.

We are far from claiming to have any definitive answers or final statements on these questions of peptide binding and serotype clustering. Many problems here are left unresolved. For example, the serotype clustering result is more provocative than otherwise and further studies are needed. One could look at more automatic choice of the supertypes, or develop comparative schemes. One could also study problems

🌱 Springer

of phylogenetic trees from this point of view as those of H5N1. Extending the framework to 3D structures of proteins, instead of just amino acid sequences is suggested. We intend to study these questions ourselves and hope that our study will persuade others to think about these kernels on amino acid sequences. We make no claim that our results are superior to what could be done with conventional alignment/phylogeny based methods.

Appendix: The BLOSUM62-2 Matrix

We list the whole BLOSUM62-2 matrix in Table 8. Table 9 explains the amino acids denoted by the capital letters.

From the Introduction, we see that the matrix Q can be recovered from the BLOSUM62-2 once the marginal probability vector p is available. The latter vector is obtained by

$$p = \left([\text{BLOSUM62-2}]\right)^{-1} v_1,$$

Table 8 The BLOSUM62-2 matrix

	A	R	N	D	C	Q	E	G	H	I
A	3.9029	0.6127	0.5883	0.5446	0.8680	0.7568	0.7413	1.0569	0.5694	0.6325
R	0.6127	6.6656	0.8586	0.5732	0.3089	1.4058	0.9608	0.4500	0.9170	0.3548
N	0.5883	0.8586	7.0941	1.5539	0.3978	1.0006	0.9113	0.8637	1.2220	0.3279
D	0.5446	0.5732	1.5539	7.3979	0.3015	0.8971	1.6878	0.6343	0.6786	0.3390
C	0.8680	0.3089	0.3978	0.3015	19.5766	0.3658	0.2859	0.4204	0.3550	0.6535
Q	0.7568	1.4058	1.0006	0.8971	0.3658	6.2444	1.9017	0.5386	1.1680	0.3829
E	0.7413	0.9608	0.9113	1.6878	0.2859	1.9017	5.4695	0.4813	0.9600	0.3305
G	1.0569	0.4500	0.8637	0.6343	0.4204	0.5386	0.4813	6.8763	0.4930	0.2750
H	0.5694	0.9170	1.2220	0.6786	0.3550	1.1680	0.9600	0.4930	13.5060	0.3263
I	0.6325	0.3548	0.3279	0.3390	0.6535	0.3829	0.3305	0.2750	0.3263	3.9979
L	0.6019	0.4739	0.3100	0.2866	0.6423	0.4773	0.3729	0.2845	0.3807	1.6944
K	0.7754	2.0768	0.9398	0.7841	0.3491	1.5543	1.3083	0.5889	0.7789	0.3964
M	0.7232	0.6226	0.4745	0.3465	0.6114	0.8643	0.5003	0.3955	0.5841	1.4777

Table 8 (*Continued*)

	A	R	N	D	C	Q	E	G	H	I
F	0.4649	0.3807	0.3543	0.2990	0.4390	0.3340	0.3307	0.3406	0.6520	0.9458
P	0.7541	0.4815	0.4999	0.5987	0.3796	0.6413	0.6792	0.4774	0.4729	0.3847
S	1.4721	0.7672	1.2315	0.9135	0.7384	0.9656	0.9504	0.9036	0.7367	0.4432
T	0.9844	0.6778	0.9842	0.6948	0.7406	0.7913	0.7414	0.5793	0.5575	0.7798
W	0.4165	0.3951	0.2778	0.2321	0.4500	0.5094	0.3743	0.4217	0.4441	0.4089
Y	0.5426	0.5560	0.4860	0.3457	0.4342	0.6111	0.4965	0.3487	1.7979	0.6304
V	0.9365	0.4201	0.3690	0.3365	0.7558	0.4668	0.4289	0.3370	0.3394	2.4175

	L	K	M	F	P	S	T	W	Y	V
A	0.6019	0.7754	0.7232	0.4649	0.7541	1.4721	0.9844	0.4165	0.5426	0.9365
R	0.4739	2.0768	0.6226	0.3807	0.4815	0.7672	0.6778	0.3951	0.5560	0.4201
N	0.3100	0.9398	0.4745	0.3543	0.4999	1.2315	0.9842	0.2778	0.4860	0.3690
D	0.2866	0.7841	0.3465	0.2990	0.5987	0.9135	0.6948	0.2321	0.3457	0.3365
C	0.6423	0.3491	0.6114	0.4390	0.3796	0.7384	0.7406	0.4500	0.4342	0.7558
Q	0.4773	1.5543	0.8643	0.3340	0.6413	0.9656	0.7913	0.5094	0.6111	0.4668
E	0.3729	1.3083	0.5003	0.3307	0.6792	0.9504	0.7414	0.3743	0.4965	0.4289
G	0.2845	0.5889	0.3955	0.3406	0.4774	0.9036	0.5793	0.4217	0.3487	0.3370
H	0.3807	0.7789	0.5841	0.6520	0.4729	0.7367	0.5575	0.4441	1.7979	0.3394
I	1.6944	0.3964	1.4777	0.9458	0.3847	0.4432	0.7798	0.4089	0.6304	2.4175
L	3.7966	0.4283	1.9943	1.1546	0.3711	0.4289	0.6603	0.5680	0.6921	1.3142
K	0.4283	4.7643	0.6253	0.3440	0.7038	0.9319	0.7929	0.3589	0.5322	0.4565
M	1.9943	0.6253	6.4815	1.0044	0.4239	0.5986	0.7938	0.6103	0.7084	1.2689
F	1.1546	0.3440	1.0044	8.1288	0.2874	0.4400	0.4817	1.3744	2.7694	0.7451
P	0.3711	0.7038	0.4239	0.2874	12.8375	0.7555	0.6889	0.2818	0.3635	0.4431
S	0.4289	0.9319	0.5986	0.4400	0.7555	3.8428	1.6139	0.3853	0.5575	0.5652
T	0.6603	0.7929	0.7938	0.4817	0.6889	1.6139	4.8321	0.4309	0.5732	0.9809
W	0.5680	0.3589	0.6103	1.3744	0.2818	0.3853	0.4309	38.1078	2.1098	0.3745
Y	0.6921	0.5322	0.7084	2.7694	0.3635	0.5575	0.5732	2.1098	9.8322	0.6580
V	1.3142	0.4565	1.2689	0.7451	0.4431	0.5652	0.9809	0.3745	0.6580	3.6922

Table 9 The list of the amino acids

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

where $v_1 = (1, \ldots, 1) \in \mathbb{R}^{20}$ is a vector with all its coordinate being 1. The matrix Q can be obtained precisely from http://www.ncbi.nlm.nih.gov/IEB/ToolBox/CPP_DOC/lxr/source/src/algo/blast/composition_adjustment/matrix_frequency_data.c#L391.

References

1. M. Andreatta, Discovering sequence motifs in quantitative and qualitative peptide data. Ph.D. thesis, Center for Biological Sequence Analysis, Department of systems biology, Technical University of Denmark, 2012.
2. N. Aronszajn, Theory of reproducing kernels, *Trans. Am. Math. Soc.* **68**, 337–404 (1950).
3. A. Baas, X.J. Gao, G. Chelvanayagam, Peptide binding motifs and specificities for HLA-DQ molecules, *Immunogenetics* **50**, 8–15 (1999).
4. L. Bartholdi, T. Schick, N. Smale, S. Smale, A.W. Baker, Hodge theory on metric spaces, *Found. Comput. Math.* **12**(1), 1–48 (2012).
5. E.E. Bittar, N. Bittar (eds.), *Principles of Medical Biology: Molecular and Cellular Pharmacology* (JAI Press, London, 1997).
6. F.A. Castelli, C. Buhot, A. Sanson, H. Zarour, S. Pouvelle-Moratille, C. Nonn, H. Gahery-Ségard, J.-G. Guillet, A. Ménez, B. Georges, B. Maillère, HLA-DP4, the most frequent HLA II molecule, defines a new supertype of peptide-binding specificity, *J. Immunol.* **169**, 6928–6934 (2002).
7. F. Cucker, D.X. Zhou, *Learning Theory: An Approximation Theory Viewpoint* (Cambridge University Press, Cambridge, 2007).
8. W.H.E. Day, H. Edelsbrunner, Efficient algorithms for agglomerative hierarchical clustering methods, *J. Classif.* **1**(1), 7–24 (1984).
9. I.A. Doytchinova, D.R. Flower, In silico identification of supertypes for class II MHCs, *J. Immunol.* **174**(11), 7085–7095 (2005).
10. Y. El-Manzalawy, D. Dobbs, V. Honavar, On evaluating MHC-II binding peptide prediction methods, *PLoS ONE* **3**, e3268 (2008).
11. M. Galan, E. Guivier, G. Caraux, N. Charbonnel, J.-F. Cosson, A 454 multiplex sequencing method for rapid and reliable genotyping of highly polymorphic genes in large-scale studies, *BMC Genom.* **11**(296) (2010).
12. G.H. Golub, M. Heath, G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, *Technometrics* **21**, 215–224 (1979).
13. D. Graur, W.-H. Li, *Fundamentals of Molecular Evolution* (Sinauer Associates, Sunderland, 2000).
14. W.W. Grody, R.M. Nakamura, F.L. Kiechle, C. Strom, *Molecular Diagnostics: Techniques and Applications for the Clinical Laboratory* (Academic Press, San Diego, 2010).
15. D. Haussler, Convolution kernels on discrete structures. Tech. report, 1999.
16. S. Henikoff, J.G. Henikoff, Amino acid substitution matrices from protein blocks, *Proc. Natl. Acad. Sci. USA* **89**, 10915–10919 (1992).
17. R. Holdsworth, C.K. Hurley, S.G. Marsh, M. Lau, H.J. Noreen, J.H. Kempenich, M. Setterholm, M. Maiers, The HLA dictionary 2008: a summary of HLA-A, -B, -C, -DRB1/3/4/5, and -DQB1 alleles and their association with serologically defined HLA-A, -B, -C, -DR, and -DQ antigens, *Tissue Antigens* **73**(2), 95–170 (2009).
18. R.A. Horn, C.R. Johnson, *Topics in Matrix Analysis* (Cambridge University Press, Cambridge, 1994).
19. L. Jacob, J.-P. Vert, Efficient peptide–MHC-I binding prediction for alleles with few known binders, *Bioinformatics* **24**(3), 358–366 (2008).
20. C.A. Janeway, P. Travers, M. Walport, M.J. Shlomchik, *Immunobiology*, 5th edn. (Garland Science, New York, 2001).
21. N. Jojic, M. Reyes-Gomez, D. Heckerman, C. Kadie, O. Schueler-Furman, Learning MHC I–peptide binding, *Bioinformatics* **22**(14), e227–e235 (2006).
22. T.J. Kindt, R.A. Goldsby, B.A. Osborne, J. Kuby, *Kuby Immunology* (Freeman, New York, 2007).
23. C. Leslie, E. Eskin, W.S. Noble, The spectrum kernel: a string kernel for SVM protein classification, in *Pacific Symposium on Biocomputing*, vol. 7 (2002), pp. 566–575.
24. H.H. Lin, G.L. Zhang, S. Tongchusak, E.L. Reinherz, V. Brusic, Evaluation of MHC-II peptide binding prediction servers: applications for vaccine research, *BMC Bioinform.* **9**(Suppl 12), S22 (2008).

25. O. Lund, M. Nielsen, C. Kesmir, A.G. Petersen, C. Lundegaard, P. Worning, C. Sylvester-Hvid, K. Lamberth, G. Røder, S. Justesen, S. Buus, S. Brunak, Definition of supertypes for HLA molecules using clustering of specificity matrices, *Immunogenetics* **55**(12), 797–810 (2004).

26. O. Lund, M. Nielsen, C. Lundegaard, C. Keşmir, S. Brunak, *Immunological Bioinformatics* (MIT Press, Cambridge, 2005).

27. M. Maiers, G.M. Schreuder, M. Lau, S.G. Marsh, M. Fernandes-Vi na, H. Noreen, M. Setterholm, C.K. Hurley, Use of a neural network to assign serologic specificities to HLA-A, -B and -DRB1 allelic products, *Tissue Antigens* **62**(1), 21–47 (2003).

28. S.G.E. Marsh, E.D. Albert, W.F. Bodmer, R.E. Bontrop, B. Dupont, H.A. Erlich, M. Fernández-Vi na, D.E. Geraghty, R. Holdsworth, C.K. Hurley, M. Lau, K.W. Lee, B. Mach, M. Maiersj, W.R. Mayr, C.R. Müller, P. Parham, E.W. Petersdorf, T. SasaZuki, J.L. Strominger, A. Svejgaard, P.I. Terasaki, J.M. Tiercy, J. Trowsdale, Nomenclature for factors of the HLA system, 2010, *Tissue Antigens* **75**(4), 291–455 (2010).

29. M. Nielsen, O. Lund, NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction, *BMC Bioinform.* **10**, 296 (2009).

30. M. Nielsen, C. Lundegaard, T. Blicher, B. Peters, A. Sette, S. Justesen, S. Buus, O. Lund, Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan, *PLoS Comput. Biol.* **4**(7), e1000107 (2008).

31. M. Nielsen, S. Justesen, O. Lund, C. Lundegaard, S. Buus, NetMHCIIpan-2.0: improved pan-specific HLA-DR predictions using a novel concurrent alignment and weight optimization training procedure, *Immunome Res.* **6**(1), 9 (2010).

32. D. Ou, L.A. Mitchell, A.J. Tingle, A new categorization of HLA DR alleles on a functional basis, *Hum. Immunol.* **59**(10), 665–676 (1998).

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

34. R. Sadiq, S. Tesfamariam, Probability density functions based weights for ordered weighted averaging (OWA) operators: an example of water quality indices, *Eur. J. Oper. Res.* **182**(3), 1350–1368 (2007).

35. H. Saigo, J.-P. Vert, N. Ueda, T. Akutsu, Protein homology detection using string alignment kernels, *Bioinformatics* **20**(11), 1682–1689 (2004).

36. H. Saigo, J.P. Vert, T. Akutsu, Optimizing amino acid substitution matrices with a local alignment kernel, *BMC Bioinform.* **7**, 246 (2006).

37. J. Salomon, D.R. Flower, Predicting class II MHC-peptide binding: a kernel based approach using similarity scores, *BMC Bioinform.* **7**, 501 (2006).

38. B. Schölkopf, A.J. Smola, *Learning with Kernels* (MIT Press, Cambridge, 2001).

39. A. Sette, J. Sidney, Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B polymorphism, *Immunogenetics* **50**(3–4), 201–212 (1999).

40. A. Sette, L. Adorini, S.M. Colon, S. Buus, H.M. Grey, Capacity of intact proteins to bind to MHC class II molecules, *J. Immunol.* **143**(4), 1265–1267 (1989).

41. J. Shawe-Taylor, N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University Press, Cambridge, 2004).

42. J. Sidney, H.M. Grey, R.T. Kubo, A. Sette, Practical, biochemical and evolutionary implications of the discovery of HLA class I supermotifs, *Immunol. Today* **17**(6), 261–266 (1996).

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

44. S. Smale, L. Rosasco, J. Bouvrie, A. Caponnetto, T. Poggio, Mathematics of the neural response, *Found. Comput. Math.* **10**(1), 67–91 (2010).

45. S. Southwood, J. Sidney, A. Kondo, M.F. del Guercio, E. Appella, S. Hoffman, R.T. Kubo, R.W. Chesnut, H.M. Grey, A. Sette, Several common HLA-DR types share largely overlapping peptide binding repertoires, *J. Immunol.* **160**(7), 3363–3373 (1998).

46. G. Thomson, N. Marthandan, J.A. Hollenbach, S.J. Mack, H.A. Erlich, R.M. Single, M.J. Waller, S.G.E. Marsh, P.A. Guidry, D.R. Karp, R.H. Scheuermann, S.D. Thompson, D.N. Glass, W. Helmberg, Sequence feature variant type (SFVT) analysis of the HLA genetic association in juvenile idiopathic arthritis, in *Pacific Symposium on Biocomputing'2010* (2010), pp. 359–370.

47. J.-P. Vert, H. Saigo, T. Akustu, Convolution and local alignment kernel, in *Kernel Methods in Computational Biology*, ed. by B. Schoelkopf, K. Tsuda, J.-P. Vert (MIT Press, Cambridge, 2004), pp. 131–154.

48. G. Wahba, *Spline Models for Observational Data* (SIAM, Philadelphia, 1990).

49. L. Wan, G. Reinert, F. Sun, M.S. Waterman, Alignment-free sequence comparison (II): theoretical power of comparison statistics, *J. Comput. Biol.* **17**(11), 1467–1490 (2010).
50. P. Wang, J. Sidney, C. Dow, B. Mothé, A. Sette, B. Peters, A systematic assessment of MHC class II peptide binding predictions and evaluation of a consensus approach, *PLoS Comput. Biol.* **4**, e1000048 (2008).
51. C. Widmer, N.C. Toussaint, Y. Altun, O. Kohlbacher, G. Rätsch, Novel machine learning methods for MHC class I binding prediction, in *Pattern Recognition Bioinformatics*, vol. 6282, ed. by T.M.H. Dijkstra, E. Tsivtsivadze, E. Marchiori, T. Heskes (Springer, Berlin, 2010), pp. 98–109.
52. R.R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Trans. Syst. Man Cybern.* **18**(1), 183–190 (1988).
53. J.W. Yewdell, J.R. Bennink, Immunodominance in major histocompatibility complex class I-restricted T lymphocyte responses, *Annu. Rev. Immunol.* **17**, 51–88 (1999).