

Novel Machine Learning Methods for MHC Class I Binding Prediction

Christian Widmer^{1,*}, Nora C. Toussaint^{2,*}, Yasemin Altun³,
Oliver Kohlbacher², and Gunnar Rätsch¹

¹ Friedrich Miescher Laboratory, Max Planck Society, Spemannstr. 39, 72076
Tübingen, Germany

² Center for Bioinformatics Tübingen, Eberhard-Karls-Universität, Sand 14, 72076
Tübingen, Germany

³ Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076
Tübingen, Germany

Abstract. MHC class I molecules are key players in the human immune system. They bind small peptides derived from intracellular proteins and present them on the cell surface for surveillance by the immune system. Prediction of such MHC class I binding peptides is a vital step in the design of peptide-based vaccines and therefore one of the major problems in computational immunology. Thousands of different types of MHC class I molecules exist, each displaying a distinct binding specificity. The lack of sufficient training data for the majority of these molecules hinders the application of Machine Learning to this problem.

We propose two approaches to improve the predictive power of kernel-based Machine Learning methods for MHC class I binding prediction: First, a modification of the Weighted Degree string kernel that allows for the incorporation of amino acid properties. Second, we propose an enhanced Multitask kernel and an optimization procedure to fine-tune the kernel parameters. The combination of both approaches yields improved performance, which we demonstrate on the IEDB benchmark data set.

1 Introduction

Despite the success of traditional whole-organism vaccines in the last century there is still a lack of effective vaccines for many diseases, for example AIDS and cancer. A fairly new approach to vaccination, the peptide-based vaccines, shows great promise here. Peptide-based vaccines utilize peptides, i.e. small protein fragments, derived from, e.g., viral proteins to induce immunity. In order for a peptide to trigger an immune response from inside a host's cell, it has to bind to a major histocompatibility complex class I (MHC-I) molecule. The resulting peptide/MHC-I complex will be transported to the cell surface where it can be recognized by specific immune system cells, the T cells (Fig. 1A), and thereby induce an immune response. Thus, MHC-I binding is a prerequisite for peptide immunogenicity. Furthermore, identifying peptides with a high affinity to

* Authors contributed equally.

MHC-I molecules is generally considered the best way to identify immunogenic peptides. Since only immunogenic peptides are suitable candidates for inclusion in a peptide-based vaccine, the prediction of peptides binding to MHC-I is of great interest in the field of vaccine design.

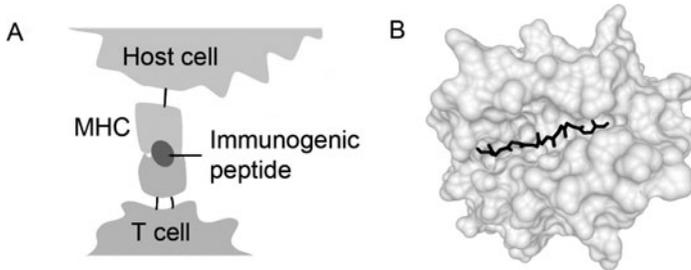


Fig. 1. Peptide/MHC-I complex. A) An MHC-I molecule presents an immunogenic peptide on the cell surface where it is recognized by a T cell. B) The structure of a nonameric peptide complexed with an MHC-I molecule. The binding groove is closed at both ends and the peptide is bound in an extended conformation. (PDB-ID: 3L3D (<http://www.pdb.org>) [2], plotted with BALLView [11])

MHC-I molecules are membrane-bound proteins with a closed binding groove that holds peptides in an extended conformation (Fig. 1B). They typically bind peptides that contain eight to ten amino acids (AAs) with a preference for nine AAs. The corresponding gene complex is highly polymorphic. As of today, more than 3,000 different MHC-I alleles are known, each coding for an MHC-I molecule binding a specific range of peptides. Any human has up to six different types of MHC-I molecules. This implies that a peptide that is capable of inducing an immune response in one individual might never be presented on the cell surface in another. In order to design vaccines effective for a given population, it is therefore necessary to accurately predict MHC-I binding for a wide range of different MHC alleles [22].

Many computational methods for the classification of peptides into MHC-I binders or non-binders have been proposed: ranging from matrices [14, 17] to machine learning [3, 1]. All of these methods require a certain amount of experimental binding data for each allele under consideration. However, a major problem in MHC-I binding prediction is the lack of data: for the vast majority of the known alleles there is no or only little experimental binding data available yielding the development of accurate prediction methods for most alleles rather challenging. In 2008, Laurent and Vert [8] proposed a kernel-based approach that tries to overcome the lack of training data by sharing binding information across alleles.

In this work, we propose two approaches to improve MHC-I binding prediction. First, we consider an improved string kernel, which takes AA properties into account and thereby allows more accurate predictions for alleles with little binding data. Second, we consider an enhanced Multitask learning algorithm,

which can be used to improve prediction performance for an allele by utilizing binding data of similar alleles. We are able to show that the combination of both approaches outperforms existing methods.

2 Improved String Kernels for MHC-I Binding Prediction

Background. String kernels are a very powerful tool for machine learning in bioinformatics due to their capability to exploit the sequential structure of AA or nucleotide sequences. They have been successfully applied to various problems in computational biology, ranging from protein remote homology detection [10], to gene identification [16, 20], to drug design [8].

MHC-I molecules bind peptides in an extended conformation (Fig. 1B). Within the complex the peptide’s side chains interact with surrounding side chains of the MHC and also with each other. Each of the peptide’s side chains contributes to the binding affinity. The respective contribution is influenced by the position of a side chain within the peptide sequence as well as by the AA types of its neighboring side chains. A string kernel is very well suited to handle such data is the *Weighted Degree (WD) kernel* [15]. The WD kernel considers sequences of fixed length L and counts co-occurring substrings in both sequences at the same position. It is defined as

$$K_{\ell}^{\text{wd}}(\mathbf{x}, \mathbf{z}) = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} \mathbf{I}(\mathbf{x}_{[i:i+d]} = \mathbf{z}_{[i:i+d]}) \quad (1)$$

where $\beta_d = 2^{\frac{\ell-d+1}{\ell^2+\ell}}$ is the weighting of the substring lengths.

A major downside to using string kernels on AA sequences is that prior knowledge on properties of individual AAs, e.g., their size, hydrophobicity, charge, cannot be easily incorporated. Especially when dealing with small training data sets as common in MHC-I-binding prediction, inclusion of this information in the sequence representation would be beneficial.

A straightforward approach to utilizing this knowledge is to consider a representation of the sequence as vector of the physico-chemical properties of all sequence elements, i.e. AAs. One may then use a standard kernel to compute sequence similarities, as, e.g., done in [24, 13]. This approach, however, ignores the sequential nature of the underlying data.

Here, we propose to combine the benefits of standard string kernels with the ones of physico-chemical descriptors for AAs.

Idea. As string kernels in general, the WD kernel (1) compares substrings of length ℓ between the input sequences \mathbf{x} and \mathbf{z} . We can rewrite the corresponding term $\mathbf{I}(\bar{\mathbf{x}} = \bar{\mathbf{z}})$ as:

$$\mathbf{I}(\bar{\mathbf{x}} = \bar{\mathbf{z}}) = \langle \Phi_{\ell}(\bar{\mathbf{x}}), \Phi_{\ell}(\bar{\mathbf{z}}) \rangle,$$

where $\bar{\mathbf{x}}, \bar{\mathbf{z}} \in \Sigma^{\ell}$ and $\Phi_{\ell} : \Sigma^{\ell} \rightarrow \mathbb{R}^{|\Sigma^{\ell}|}$.

$\Phi_\ell(\bar{\mathbf{x}})$ can be indexed by a substring $s \in \Sigma^\ell$ and is defined as $\Phi_\ell(\bar{\mathbf{x}})_s = 1$, if $s = \bar{\mathbf{x}}$, and 0 otherwise. Using $\Phi_1 : \Sigma \mapsto \{0, 1\}$, a simple encoding of the letters into $|\Sigma|$ -dimensional unit vectors, the substring comparison can be rewritten as

$$\mathbf{I}(\bar{\mathbf{x}} = \bar{\mathbf{z}}) = \prod_{l=1}^{\ell} \langle \Phi_1(\bar{\mathbf{x}}_l), \Phi_1(\bar{\mathbf{z}}_l) \rangle,$$

Φ_1 ignores the relations between the letters in the alphabet. Since this is a problem when considering AAs, we replace Φ_1 with a feature map Ψ that takes relations between the AAs into account. This leads to the following kernel on AA substrings:

$$K_\ell^\Psi(\bar{\mathbf{x}}, \bar{\mathbf{z}}) = \prod_{l=1}^{\ell} \langle \Psi(\bar{\mathbf{x}}_l), \Psi(\bar{\mathbf{z}}_l) \rangle. \quad (2)$$

Using the feature representation corresponding to this kernel, we can now recognize sequences of AAs that have certain properties (e.g. first AA: hydrophobic, second AA: large, third AA: positively charged, etc.): For every combination of products of features involving exactly one AA property per substring position, there is one feature induced in the kernel. A richer feature space including combinations of several properties from every position can be obtained using the following two formulations. The first is based on the polynomial kernel:

$$K_{\ell,d}^\Psi(\bar{\mathbf{x}}, \bar{\mathbf{z}}) = \left(\sum_{l=1}^{\ell} \langle \Psi(\bar{\mathbf{x}}_l), \Psi(\bar{\mathbf{z}}_l) \rangle \right)^d, \quad (3)$$

and the second on the RBF kernel:

$$K_{\ell,\sigma}^\Psi(\bar{\mathbf{x}}, \bar{\mathbf{z}}) = \exp \left(- \frac{\sum_{l=1}^{\ell} \|\Psi(\bar{\mathbf{x}}_l) - \Psi(\bar{\mathbf{z}}_l)\|^2}{\sigma^2} \right). \quad (4)$$

Improved WD Kernel. Replacing the substring comparison $\mathbf{I}(\bar{\mathbf{x}} = \bar{\mathbf{z}})$ in (1) with one of the formulations in (2), (3), or (4) together with a set of features $\Psi(a)$ for each letter $a \in \Sigma$ (i.e. for each AA), directly leads to a generalized form of the WD kernel:

$$K_\ell^{\text{wd},\Psi}(\mathbf{x}, \mathbf{z}) = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} K_d^\Psi(\mathbf{x}_{[i:i+d]}, \mathbf{z}_{[i:i+d]}). \quad (5)$$

$K_\ell^{\text{wd},\Psi}$ is a linear combination of kernels and therefore a valid kernel [18]. It can be computed efficiently, with a complexity comparable to that of the original WD kernel.

The combination of the WD kernel with the RBF substring kernel (4) is particularly interesting:

$$K_{\ell,\sigma}^{\text{wd},\Psi}(\mathbf{x}, \mathbf{z}) = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} \exp \left(- \frac{\sum_{j=1}^d \|\Psi(\mathbf{x}_j) - \Psi(\mathbf{z}_j)\|^2}{\sigma^2} \right). \quad (6)$$

For a bijective encoding Ψ and $\sigma \rightarrow 0$, this *WD-RBF kernel* corresponds to the WD kernel: the RBF substrng kernel will be one only for identical substrings, otherwise it will be zero. Thus, employing the WD-RBF kernel will, at least in theory, always yield equal or better performances than the original WD kernel.

3 A New Multitask Kernel for MHC-I Binding Prediction

We will build upon a kernel-based formulation of Multitask Learning, as proposed by [4]:

$$\begin{aligned} \max_{\alpha} & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \tilde{K}((\mathbf{x}_i, s), (\mathbf{x}_j, t)) + \sum_{i=1}^n \alpha_i & (7) \\ \text{s.t.} & \alpha^T \mathbf{y} = 0, \quad 0 \leq \alpha_i \leq C \quad \forall i \in \{1, n\}, \end{aligned}$$

where s and t correspond to the tasks associated with examples x_i and x_j , respectively.

$$\tilde{K}((\mathbf{x}_i, s), (\mathbf{x}_j, t)) = K(\mathbf{x}_i, \mathbf{x}_j) + K^{\text{dirac}}(s, t) \cdot K(\mathbf{x}_i, \mathbf{x}_j), \quad (8)$$

where K denotes the base kernel that captures the interactions between examples from all tasks and $K^{\text{dirac}}(s, t)$ is defined as

$$K^{\text{dirac}}(s, t) = \begin{cases} 1, & \text{if } t = s \\ 0, & \text{else} \end{cases}. \quad (9)$$

It was shown in previous work [7] that it pays off to use multitask learning methods for the problem of MHC-I binding prediction. In particular, a multitask kernel based on the product of allele (i.e. task) similarity and peptide (i.e. instance) similarity was used:

$$K^{\text{MT}}((x, s), (z, t)) = K^{\text{all}}(s, t) \cdot K^{\text{pep}}(x, z),$$

which is a generalization of the kernel presented in Equation (8). Here, the similarity between tasks is explicitly taken into account, instead of solely setting a higher default similarity for in-domain comparisons. In the case of MHC-I molecules, the pseudo sequence (i.e. the AAs in the binding groove of the MHC that interact with the bound peptide) is used as task-feature. Clearly, the more similar the pseudo sequences are the more similar we expect the tasks to be. Furthermore, [8] considered several combinations of kernels for K^{all} and K^{pep} . The best performing combination employed a polynomial kernel on top of a string kernel of degree $d = 1$ for both, K^{all} and K^{pep} .

We now aim at improving the above multitask kernel K^{MT} as follows. First, we introduce additional parameters, that allow the specialization of the trade-off between the in-domain kernel components (i.e. $s = t$) and the out-of-domain kernel components (i.e. $s \neq t$) dependent on the task.

While K^{MT} captures how closely related two tasks are, according to some task kernel K^{all} , it does not take into account how well information from the other tasks boosts performance. Clearly, transferring information from other tasks will become increasingly relevant if only little training data is available. If there is an abundance of training data for a particular task, it is most likely sufficient to set a stronger focus on in-domain data.

The above leads us to the following kernel formulation, which introduces a new multitask kernel composed of a linear combination of two multitask kernels with two mixing coefficients $\beta_{s,1}$ and $\beta_{s,2}$ that have to be adjusted for each task s independently. Details on how the $\beta_{s,k}$ are tuned are given in the following section.

$$K^{\text{MT-WD}}((x, s), (z, t)) = \beta_{s,1} K^{\text{WD}}(s, t) \cdot K^{\text{WD}}(x, z) + \beta_{s,2} K^{\text{dirac}}(s, t) \cdot K^{\text{WD}}(x, z) \quad (10)$$

Finally, by combining both lines of work, we propose a multitask kernel that uses the enhanced WD Kernel $K^{\text{WD-RBF}}$ (see Equation 6) from the previous section to compute the similarity between instances. We arrive at the following formulation:

$$K^{\text{MT-WD-RBF}}((x, s), (z, t)) = \beta_{s,1} K^{\text{WD}}(s, t) \cdot K^{\text{WD-RBF}}(x, z) + \beta_{s,2} K^{\text{dirac}}(s, t) \cdot K^{\text{WD-RBF}}(x, z) \quad (11)$$

In summary, the new kernel formulation consists of three parts. First, we formulate the kernel as a combination of a task specific component and a multitask kernel component. Second, we introduce task-specific parameters that can be tuned for each task independently. Third, we combine the previous two ideas with the novel WD-RBF kernel.

4 Fine Tuning the Kernel with Multiple Kernel Learning

We propose to use Multiple Kernel Learning (MKL) [9] to learn the weights $\beta_{s,k}$ of the individual components (see Equation 11) along with the respective classifiers. In particular, we employ a variant of MKL, which was shown to work well in the domain of computer vision [5]. Here, the setup is slightly different from standard MKL, as we first obtain one classifier f_i for each kernel K_i (i.e. $f_i(x) = \sum_j \alpha_j y_j K_i(x, x_j)$) and then find an optimal linear combination of the learned functions in a second step (i.e. $f(x, s) = \sum_i \beta_{s,i} f_i(x)$). In [5], the authors propose to use LPBoost for the combination of classifiers. However, LPBoost yields a sparse solution in terms of kernel weights, which is not what we are interested in. Therefore, we propose a formulation based on the nu-SVM [19] to combine the classifiers f_i .

$$\begin{aligned}
& \min_{\beta, \xi, \rho} \frac{1}{2} \|\beta - \mathbf{1}\|^2 + \sum_{i=1}^N \xi_i - \rho\nu & (12) \\
& \text{s.t.} \quad y_i \sum_{j=1}^M \beta_j f_j(\mathbf{x}_i) + \xi_i \geq \rho \quad \forall i \in [1, \dots, N], \\
& \quad \beta_i \geq 0 \quad \forall i \in [1, \dots, M] \\
& \quad \xi_i \geq 0 \quad \forall i \in [1, \dots, N]
\end{aligned}$$

From preliminary experiments, we observed that $\beta_{s,k} = 1 \quad \forall s \forall k$ often yields a good solution. We use this as prior knowledge by regularizing the parameter vector β to be close to a vector of ones $\mathbf{1}$. Intuitively speaking, only the training error measured by the loss term will cause the $\beta_{s,k}$ to differ from 1.

For the training procedure, the training data is split into two parts. The first part containing $\frac{3}{4}$ of training examples is used to obtain the initial f_i . Subsequently, the second part of the training data is used in the loss term of Equation (12), which is solved for each task s individually. After having obtained the β , we retrain the f_i on the entire training data set and use the determined β for the final linear combination $f(x, s) = \sum_{i=1}^2 \beta_{s,i} f_i(x)$.

5 Experimental Methods

Data. The IEDB benchmark data set from Peters *et al.* [12] contains quantitative binding data (IC₅₀ values) for various MHC alleles, including 35 human MHC alleles. Splits for a 5-fold cross-validation are given. We evaluate the performance of the proposed methods on a subset of this data set: binding data of nonameric peptides with respect to human MHC. Peptides with IC₅₀ values greater than 500 nM were considered non-binders, all others binders.

Amino acid descriptors. A wide range of physico-chemical and other descriptors of AAs have been published. Within this work we use encode each AA by 20 descriptors corresponding to the respective entries of the Blosum50 substitution matrix [6].

Performance evaluation procedure. For performance evaluation we employ a two times nested 5-fold cross-validation, i.e. two nested cross-validation loops. The inner loop is used for model selection (kernel and regularization parameters) and the outer loop for performance estimation. Performance is measured by averaging the area under the ROC curve (auROC).

Learning curve analysis. To assess the performance dependence on the amount of training data, WD kernel and WD-RBF kernel performances were analyzed on allele A*0201 in 100 cross-validation runs to average over different data splits to reduce random fluctuations of the performance values. In each run, 30% of the available data was used for testing. From the remaining data, training sets of different sizes (20, 31, 50, 80, 128, 204, 324, 516, 822, 1,308) were selected randomly.

Performance analysis of the improved WD kernel. Performances of the WD and the WD-RBF kernel were analyzed on all 35 human MHC alleles contained in the IEDB benchmark.

Performance analysis of the multitask kernel approach. Performances of three multitask learning approaches using a) the WD kernel, b) the WD-RBF kernel, and c) the WD-RBF kernel with an additional optimization step were also analyzed on all 35 human MHC alleles contained in the IEDB benchmark.

SVM computations. We used the freely available large scale machine learning toolbox Shogun [21] for all SVM computations. All used kernels are implemented as part of the toolbox and will be part of Shogun-0.9.3.

Results and Discussion

The main goal of this work is to present novel ideas for kernel-based MHC-I binding prediction, namely an enhanced string kernel [23] and a refined model for multitask learning.

Improved WD Kernel

The more data is available, the easier it will be to infer the relation of the AAs from the sequences in the training data alone. Therefore, the incorporation of additional information can be expected to especially improve prediction accuracy in cases where less training data is available. We chose the allele with the highest number of peptides, A*0201, to perform a learning curve analysis for WD and WD-RBF. Mean auROCs with confidence intervals (σ/\sqrt{n}) over 100 cross-validation runs are shown in Figure 2. It can clearly be seen, that the fewer examples are available for learning, the stronger is the improvement of the WD-RBF over the WD kernel.

In a more comprehensive comparison, we assessed the performance of WD and WD-RBF kernels on all 35 human MHC alleles from the IEDB benchmark. WD-RBF outperforms WD for 24 alleles (Fig. 3). This is significant with respect to the binomial distribution: Assuming equal performance of WD and WD-RBF, the probability of WD-RBF outperforming WD 24 out of 35 times is ≈ 0.01 .

Improved Multitask Learning Kernel

From the results in Figure 4, we can make several important observations. First, in accordance with the results of [7], we clearly see that multitask learning *MTL* (*WD*) greatly improves performance compared to learning individual models *Plain* (*WD*). Second, we observe a slightly improved performance of *Plain*, when using the WD-RBF instead of the WD, which is consistent with the results from the previous section. In accordance with Figure 2, improvements using the new kernel are rather small as this dataset contains relatively many examples. Third, Figure 4 shows that employing the enhanced multitask Kernel *MTL*

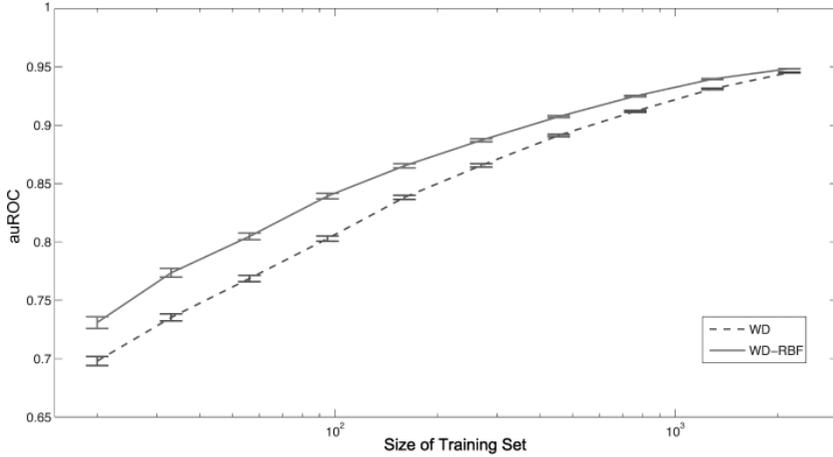


Fig. 2. Learning curve analysis on MHC allele A*0201. Shown are areas under the ROC curves averaged over 100 different test splits (30%) and for increasing numbers of training examples (up to 70%). The training part was used for training and model selection using 5-fold cross-validation.

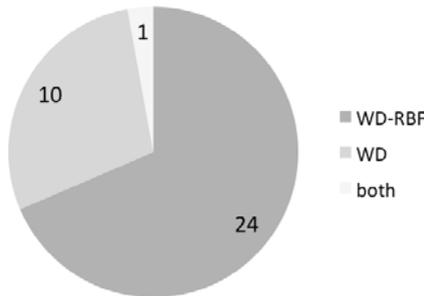


Fig. 3. Performance of WD and WD-RBF kernels on human MHC alleles from the IEDB benchmark data set: The pie chart displays the number of alleles for which the WD (green) and the WD-RBF (red) performed best, respectively, and the number of alleles for which they performed equally (blue).

(*WD-RBF*) introduced in Equation (11) improves performance compared to the regular multitask learning kernel using the WD kernel. Note, that here, the $\beta_{s,k}$ (see Equation 11) are all set to $\beta_{s,k} = 1$. Lastly, we observe that the tuning the $\beta_{s,k}$ using Equation 12 further improves performance up to $auROC = 0.909$, leaving us with the best performing method in our experiments, which slightly outperforms the method presented in [7], who reported $auROC = 0.903$ for this dataset.

We would like to point out that while the improvement over this previous method is rather small (0.6% auROC), the ideas presented in this paper have the potential to contribute to greater improvements for two reasons. First, [7] used

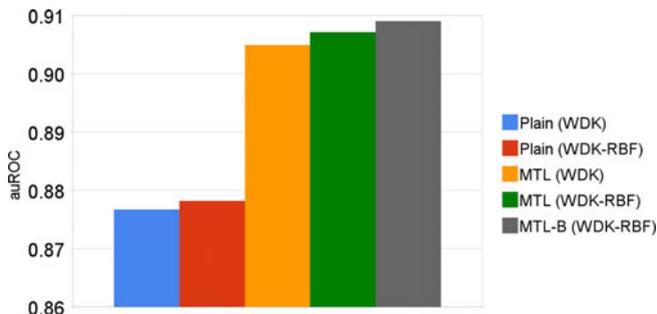


Fig. 4. Performance (averaged over alleles) measured on the IEDB benchmark data set for several methods. In *Plain (WD)*/*(WD-RBF)* classifiers are trained individually for each task using the WD kernel, or the WD-RBF, respectively. *MTL (WD)* employs a multitask kernel based on the WD, *MTL (WD-RBF)* compares instances using the WD-RBF and *MTL-B (WD-RBF)* employs an additional optimization step (see Equation 12) to fine tune kernel components.

a different *base* kernel. Finding out, whether using this kernel as starting point to our proposed improvements further boosts performance is subject to future experiments. Second, the formulation presented in Equation 12 is extensible to an arbitrary number of kernel components. With more insight into the problem domain, it might be possible to carefully engineer a multitask kernel with more than two meaningful components, which could then be tuned using the proposed formulation.

6 Conclusion

We have proposed two approaches to improve kernel-based Machine Learning methods for MHC class I binding prediction. First, a modification of the Weighted Degree string kernel that allows for the incorporation of amino acid properties. Second, we present an improved multitask learning approach based on a new multitask kernel. Finally, we combine these two approaches, which gives rise to further improvements.

Due to their high dimensional feature space, string kernels require a sufficient number of examples during training to learn relationships between amino acids. Standard kernels employing physico-chemical descriptors of amino acids, on the other hand, cannot exploit the sequential structure of the input sequences and implicitly generate many features, numerous of which will be biologically implausible. Here, one also needs many examples to learn the subset of features that is needed for accurate discrimination. The lack of training data for a large fraction of all known MHC class I alleles, however, calls for approaches that perform well even when training data is scarce. We could show, that incorporation of physico-chemical amino acid descriptors into the Weighted Degree kernel yields significant improvements in the prediction of MHC-binding peptides. This improvement is particularly strong when data is less abundant.

We confirmed that multitask learning methods are beneficial for MHC class I binding prediction. Furthermore, we presented an enhanced multitask kernel that incorporates the improved WD kernel and that has additional hyper-parameters, which are in turn tuned using a variant of the nu-SVM.

Our results show that incorporation of prior knowledge of amino acid properties as well as a sophisticated approach to fine tuning the multitask kernel yields improvements in kernel-based MHC-I binding prediction. While this work focused on the classification into binders and non-binders, the proposed methods show promise also for the quantitative prediction of peptide/MHC class I binding affinity.

Acknowledgments. This work was partly supported by Deutsche Forschungsgemeinschaft (SFB 685, project B1).

References

1. Adams, H.P., Koziol, J.A.: Prediction of binding to MHC class I molecules. *Journal of Immunological Methods* 185(2), 181–190 (1995)
2. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. *Nucleic Acids Research* 28, 235–242 (2000)
3. Dönnes, P., Elofsson, A.: Prediction of MHC class I binding peptides, using SVMHC. *BMC Bioinformatics* 3, 25 (2002)
4. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Kim, W., Kohavi, R., Gehrke, J., DuMouchel, W. (eds.) *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, Washington, USA, August 22–25, pp. 109–117. ACM, New York (2004)
5. Gehler, P., Nowozin, S.: Infinite kernel learning. In: *NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels* (2008)
6. Henikoff, S., Henikoff, J.G.: Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America* 89(22), 10915–10919 (1992)
7. Jacob, L., Bach, F., Vert, J.P.: Clustered Multi-Task Learning: A Convex Formulation. In: *NIPS*, pp. 745–752. MIT Press, Cambridge (2009)
8. Jacob, L., Vert, J.P.: Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics* 24(3), 358 (2008)
9. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A., Laskov, P., Müller, K.R.: Efficient and accurate LP-norm MKL. In: *Advances in Neural Information Processing Systems*, vol. 22 (2009)
10. Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., Leslie, C.: Profile-based string kernels for remote homology detection and motif extraction. In: *Proceedings IEEE Computational Systems Bioinformatics Conference* (2004)
11. Moll, A., Hildebrandt, A., Lenhof, H., Kohlbacher, O.: BALLView: an object-oriented molecular visualization and modeling framework. *J. Comput. Aided Mol. Des.* 19(11), 791–800 (2005)
12. Peters, B., Bui, H.H., Frankild, S., Nielsen, M., Lundegaard, C., Kostem, E., Basch, D., Lamberth, K., Harndahl, M., Fleri, W., Wilson, S.S., Sidney, J., Lund, O., Buus, S., Sette, A.: A Community Resource Benchmarking Predictions of Peptide Binding to MHC-I Molecules. *PLoS Comput. Biol.* 2(6), e65 (2006)

13. Pfeifer, N., Kohlbacher, O.: Multiple Instance Learning Allows MHC Class II Epitope Predictions Across Alleles. In: Crandall, K.A., Lagergren, J. (eds.) WABI 2008. LNCS (LNBI), vol. 5251, pp. 210–221. Springer, Heidelberg (2008)
14. Rammensee, H., Bachmann, J., Emmerich, N.P., Bachor, O.A., Stevanovic, S.: SYFPEITHI: Database for MHC ligands and peptide motifs. *Immunogenetics* 50, 213–219 (1999)
15. Rätsch, G., Sonnenburg, S.: Accurate Splice Site Detection for *Caenorhabditis elegans*. In: Schölkopf, B., Vert, K.T. (eds.) Kernel Methods in Computational Biology, pp. 277–298. MIT Press, Cambridge (2004)
16. Rätsch, G., Sonnenburg, S., Srinivasan, J., Witte, H., Müller, K.R., Sommer, R.J., Schölkopf, B.: Improving the *Caenorhabditis elegans* genome annotation using machine learning. *PLoS Comput. Biol.* 3(2), e20 (2007)
17. Reche, P.A., Glutting, J.P., Reinherz, E.L.: Prediction of MHC class I binding peptides using profile motifs. *Hum. Immunol.* 63(9), 701–709 (2002)
18. Schölkopf, B., Burges, C., Smola, A. (eds.): *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge (1999)
19. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. *Neural Computation* 12(5), 1207–1245 (2000)
20. Schweikert, G., Zien, A., Zeller, G., Behr, J., Dieterich, C., Ong, C.S., Philips, P., De Bona, F., Hartmann, L., Bohlen, A., Krüger, N., Sonnenburg, S., Rätsch, G.: mGene: accurate SVM-based gene finding with an application to nematode genomes. *Genome Res.* 19(11), 2133–2143 (2009)
21. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research* 7, 1531–1565 (2006)
22. Toussaint, N.C., Kohlbacher, O.: Towards in silico design of epitope-based vaccines. *Expert Opinion on Drug Discovery* 4(10) (2009)
23. Toussaint, N.C., Widmer, C., Kohlbacher, O., Rätsch, G.: Exploiting physicochemical properties in string kernels. *BMC Bioinformatics* (submitted, 2010)
24. Tung, C.-W., Ho, S.-Y.: POPI: predicting immunogenicity of MHC class I binding peptides by mining informative physicochemical properties. *Bioinformatics* 23(8), 942–949 (2007)