Journal of Immunological Methods 400–401 (2013) 30–36

Contents lists available at ScienceDirect

Journal of Immunological Methods

journal homepage: www.elsevier.com/locate/jim

Research paper

MHC-NP: Predicting peptides naturally processed by the MHC

Sébastien Giguère [a,*], Alexandre Drouin [a], Alexandre Lacoste [a], Mario Marchand [a], Jacques Corbeil [b], François Laviolette [a]

[a] Department of Computer Science and Software Engineering, Pavillon Adrien-Pouliot, 1065, av. De la Médecine, Université Laval, Québec, Québec, G1V 0A6, Canada
[b] Department of Molecular Medicine, Université Laval, Québec, Québec, G1V 0A6, Canada

ARTICLE INFO

Article history:
Received 31 May 2013
Accepted 5 October 2013
Available online 18 October 2013

Keywords:
Machine learning
Kernel
Immunology
Epitope
Vaccinology
MHC

ABSTRACT

We present MHC-NP, a tool for predicting peptides naturally processed by the MHC pathway. The method was part of the 2nd Machine Learning Competition in Immunology and yielded state-of-the-art accuracy for the prediction of peptides eluted from human HLA-A*02:01, HLA-B*07:02, HLA-B*35:01, HLA-B*44:03, HLA-B*53:01, HLA-B*57:01 and mouse H2-D^b and H2-K^b MHC molecules. We briefly explain the theory and motivations that have led to developing this tool. General applicability in the field of immunology and specifically epitope-based vaccine are expected. Our tool is freely available online and hosted by the Immune Epitope Database at http://tools.immuneepitope.org/mhcnp/.

1. Introduction

Epitope-based vaccines show great promise for diseases for which current approaches, such as pathogen attenuation, are not easily feasible or efficacious. Such vaccines have the advantage of being simpler to produce and induce a very specific immune response by targeting the immunogenic region of an antigen and specific MHC alleles of the host (Toussaint and Kohlbacher, 2009). Computer assisted methods for the identification of immunogenic epitopes represent an important step in facilitating the creation of these next generation vaccines. Ideally, such vaccines could be adapted to target specific portions of the population, such as pregnant women and/or immunocompromised individuals, for which attenuated vaccines may present greater risks.

The major histocompatibility complex (MHC) is responsible for specific antigen recognition and inducing an appropriate cellular response. The MHC molecules are part of a complex pathway responsible for presenting antigens on the surface of antigen-presenting cells (APCs). Such antigens are presented under the form of MHC–peptide complexes, where the peptide is a fragment of the antigen protein's sequence. Once MHC–peptide complexes are presented on an APC's surface, T cells recognise specific complexes and trigger an appropriate immune response. MHC molecules are categorised in two classes, MHC-class I, present antigens for CD8 T-cell driven responses and MHC-II, in contrast, present antigens for CD4 T-cell responses. Both pathways are complex and the binding of a peptide to a MHC molecule can be verified by *in vitro* or *in silico* methods.

Numerous approaches have been proposed for identifying MHC–peptide complexes (Zhang et al., 2012; Lundegaard et al., 2008; Giguère et al., 2013). Most of these methods have focused on predicting the binding affinity of a given peptide and a MHC molecule. Unfortunately, the binding of a peptide and a MHC molecule is insufficient to ensure that the peptide will be processed to the surface of the cell. Indeed, only a

* Corresponding author.
E-mail addresses: sebastien.giguere.8@ulaval.ca (S. Giguère), alexandre.drouin.8@ulaval.ca (A. Drouin), alexandre.lacoste.1@ulaval.ca (A. Lacoste), mario.marchand@ift.ulaval.ca (M. Marchand), jacques.corbeil@crchul.ulaval.ca (J. Corbeil), francois.laviolette@ift.ulaval.ca (F. Laviolette).

0022-1759/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jim.2013.10.003

subset of the peptides that bind to an MHC molecule can be naturally processed. Predicting such peptides is a difficult task.

In an effort to refine epitopic peptide identification methods, we propose a method to predict if a peptide is naturally processed by the MHC pathway. Our method has been trained on *in vivo* and *in vitro* data provided by D.K. Crockett and V. Brusic (Zhang and Brusic, 2013). The proposed method is based on statistical learning. Therefore, given that training data is available, it can be used for both MHC-I and MHC-II pathways. In addition, if used in conjunction with any binding affinity prediction tool, our method can validate if peptides can be processed by the MHC pathway. In the context of the 2012 Machine Learning Competition in Immunology (MLI), we have obtained empirical results which demonstrated that our method will have promising utility in immunology, vaccinology, and transplant rejection.

2. Material and methods

2.1. Data

The datasets used to train our method were those provided to the participants of the 2012 Machine Learning Competition in Immunology. These datasets comprised eight MHC-I molecules, composed of six human molecules (HLA-A*02:01, HLA-B*07:02, HLA-B*35:01, HLA-B*44:03, HLA-B*53:01, and HLA-B*57:01) and two mouse molecules (H2-D^b and H2-K^b) (Zhang and Brusic, 2013). For each target molecule, three sets of peptides were provided: binding peptides, non-binding peptides and peptides obtained by elution. The peptides obtained by elution are naturally processed by the MHC-I pathway. Our method was tested using a set of data composed of binding, non-binding and naturally processed peptides. This testing data were provided by the organisers of the MLI competition and made publicly available http://bio.dfci.harvard.edu/DFRMLI/HTML/natural.php.

2.1.1. Discussion

From a machine learning point of view (see Hastie et al. (2001) for an introductory book), the training data were challenging in many ways. First, most learning algorithms receive data under the form of real valued feature vectors, although the training examples consisted of sequences of amino acids. Second, the length of training peptides varied from 8 to 11 amino acids. Most learning algorithms need to compute vector operations which are only defined for vectors of the same length. Therefore, any real valued vector representation of a peptide that depends on its length would ultimately fail. When facing such a problem, some authors have preferred to group peptides by length and to define independent learning problems for each length. This solution inevitably leads to inferior accuracy, since this severely reduces the number of training examples for each problem. The next section will introduce the concept of string kernels, which bring a solution to these two problems.

The particular case of identifying naturally processed peptides cannot be described as a typical binary or multiclass classification problem. Indeed, in this classification scheme, each example may only belong to one of the pre-defined classes. Yet, in our context, naturally processed peptides are all known to bind the MHC, whereas only 5–15% of the binding peptides are naturally processed. Moreover, the problem has inherent noise in the data, due to the fact that a peptide can be a known binder, but not yet identified by elution. This implies that the eluted peptide class is a subset of the binding peptide class as illustrated in Fig. 1. We therefore need to use a learning algorithm that is robust to noise and that accounts for the relationship between these classes.

2.1.2. Preprocessing the data

The provided data were obtained through experimental processes and, therefore, required a small amount of preprocessing to ensure that it contained no inconsistencies. First, for each MHC molecule, peptides that were simultaneously in the non-binding and binding sets were discarded. Second, peptides that were simultaneously in the eluted and binding sets were removed from the binding set. Note that eluted peptides also bind to MHC molecules, although we preferred having the eluted peptides in a distinct set. These two steps ensured that the three sets of peptides were disjoint and therefore that the peptides comprised in each set shared one common characteristic (eluted-binder, non-eluted-binder, non-binder). We will respectively refer to these three sets of peptides as ‘E’, ‘B’ and ‘N’.

2.2. Learning approach

Machine learning is a method of choice for building predictive software when facing complex phenomena. Classification is the task of assigning one of possibly many pre-defined classes to each example produced by a phenomena of interest. In this paper, we will restrict ourselves to the case of binary classification, where each example can only belong to two classes.

A learning algorithm should minimize the task loss. Here, the task is to distinguish naturally processed peptides from all other peptides (non-eluted-binders and non-binders). Therefore, while developing our method, we have focussed on learning a predictor for peptides that are naturally processed by the MHC pathway. Consequently, we have only considered the following two classes: peptides that are naturally processed

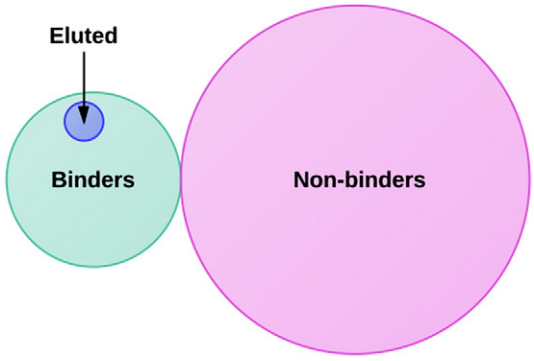

Fig. 1. An illustration of the different peptide classes contained in the datasets.

(E) and peptides that are not naturally processed (B and N). This predictor was presented to the MLI 2012 competition, ultimately yielding excellent prediction accuracy. Throughout this paper, we will refer to this model as (E vs BN).

In an attempt to further improve this method, we have, since the MLI competition, developed an alternate model. This model consists in learning two predictors: a predictor to distinguish binding peptides (E and B) from non-binding peptides (N) and a predictor to distinguish eluted peptides (E) from binding peptides (B). The prediction is done in a decision tree fashion: the (EB vs N) predictor is first applied, then, only if the peptide is predicted to be a binder, the (E vs B) predictor is applied to determine if the peptide is naturally processed. Throughout this paper, we will refer to this model as (EB vs N + E vs B).

For comparison purposes, a binding/non-binding predictor was learned to distinguish binding peptides (E + B) from non-binding peptides (N). This predictor purposely does not distinguish naturally processed peptides from binding peptides. Such a predictor allows to establish a direct comparison to current tools, which rely on MHC-peptide binding affinity. This predictor also allows to measure the capability of our method to discriminate naturally processed peptides from binding peptides.

2.2.1. Learning algorithm

To learn the predictors described in the previous section, we have used the soft-margin support vector machine (SVM) algorithm originally proposed by Cortes and Vapnik (1995). This supervised learning algorithm is considered to be the state of the art for classification problems.

The input of the SVM algorithm consists in a set of pairs. For each of these pairs, the first element is an example and the second element is a class that has been attributed to the example by an expert. First, each learning example is mapped to a possibly high dimensional vector space called the feature space. Then, the SVM finds a separating hyperplane that splits the feature space in two halves, such that examples from the same class lie on the same side. Also, the hyperplane is such that all learning examples have the greatest Euclidian distance (also known as the geometric margin) from it. The soft margin SVM provides a parameter allowing a trade-off between the number of classification errors and the size of the geometric margin on the correctly classified examples. This allows SVM to tolerate that certain examples of a given class be on the wrong side of the hyperplane and thus prevent the hyperplane from being affected by noise in the data. This parameter is generally tuned by cross-validation techniques that are described in Section 3.

To perform predictions, new testing examples are first mapped to the same space as the training examples. Then, their classes are assigned depending on their location with respect to the hyperplane. The greater the geometric margin for an example, the more confident the predictor is about its predicted class. This notion of margin can be further transformed into a probability estimate using techniques such as the Platt scaling (Platt, 1999). Our proposed tool: MHC-NP, outputs, for each new peptide, a probability estimate of the peptide being naturally processed by the specified MHC pathway.

2.2.2. From sequences to feature vectors

The kernel trick (Shawe-Taylor and Cristianini, 2004) is a method to map complex structures, such as strings, to a high dimensional vector space in which the dot product is defined. A kernel function is a function that implicitly computes the dot product in this high dimensional space, without explicitly mapping the structures to this space. Such functions therefore avoid the prohibitive computational cost of computing dot products in high dimensional vector spaces. Recall that every algorithm that constructs a hyperplane heavily depends on dot product operations. In addition, as for dot products, kernels can be interpreted as similarity functions. The more similar two examples are, the greater the kernel function value. Kernel functions have been proposed for a variety of biological structures including protein primary structure (Saigo et al., 2004) and protein tertiary structure (Qiu et al., 2007).

Many machine learning algorithms like the SVM have been kernelized by substituting the dot product operation of their objective and prediction functions by kernel function evaluations. Such kernelized algorithms can work with complex data, such as strings, given that there exists a kernel defined for this data type. Kernels that compute the similarity between two strings are called *string kernels*. There is no doubt that string kernels have been partly responsible for the increased use of machine learning algorithms in biology.

2.2.3. The generic string kernel

As highlighted in the previous section, string kernels are similarity functions that can be used with kernelized learning algorithms. A common approach is to compare strings by their common k-mers for a chosen value of k. For example, “LFQLITA” and “LFQRPPLI” can be split into their 2-mers, which are respectively (LF, FQ, QL, LI, TA) and (LF, FQ, QR, RP, PP, PL, LI). Then, the similarity value between these two sequences is given by the number of common 2-mers. Here, the similarity is 3, since the sequences have LF, FQ, and LI as common 2-mers. This similarity function corresponds to the Spectrum kernel (Leslie et al., 2002). Note that this kernel allows the comparison of strings of different lengths and corresponds to a dot product in a high dimensional vector space. Therefore, such kernels alleviate the need for considering peptides of different lengths separately and allow better usage of the data.

Comparing peptides using k-mer is a simple approach that gives a broad estimate of their similarity. However, for most proteins, notably for MHC, peptide–protein interactions occur at a very specific location known as the binding site. The relative position of residues in this binding site and their physicochemical properties are key aspects that drive the interaction. Nevertheless, the Spectrum kernel does not account for these two important biological features.

In an effort to consider the relative position of residues in similarity function, Meinicke et al. (2004) proposed the Oligo kernel, which was the first string kernel to account for the relative position of k-mers. Instead of counting the number of common k-mer, the Oligo kernel assigns a weight to each common k-mer depending on their relative positions in the two peptides. For example, in the peptides “LFQLITA” and “LFQRPPLI”, the 2-mers “LF” and “FQ” share the same position. In contrast, the 2-mer “LI” is respectively at position 4 and 7. For this reason, the contribution of “LF” and “FQ” to the similarity should be greater than the contribution of “LI”.

In this sense, the author of the Oligo kernel proposed to weight the contribution of common k-mers by using a function inversely proportional to their distance in the peptides.

Moreover, in an attempt to account for the physicochemical properties of the residues in the binding site, Toussaint et al. (2010) proposed to weight the contribution of k-mers as function of their physicochemical properties. This is based on the fact that an amino acid in a peptide can sometimes be substituted by another amino acid with similar properties, such as hydrophobicity, charge or molecular weight, without affecting the peptide's binding affinity. Toussaint and collaborators proposed to incorporate such knowledge in the kernel function by using the physicochemical properties of their amino acids to compare k-mers.

Inspired by the ideas of Meinicke et al. (2004) and Toussaint et al. (2010), Giguère et al. (2013) proposed the generic string (GS) kernel. This kernel accounts for the physicochemical properties and the relative position of amino acids in the comparison of k-mers. The GS kernel was shown to outperform state-of-the-art prediction methods on single-target and pan-specific peptide–MHC-II binding affinity prediction benchmark datasets and three Quantitative Structure Affinity Model benchmark datasets. Giguère et al. (2013) have also proposed a dynamic programming algorithm for the fast computation of their kernel and have shown that the GS kernel induces a dot product in a high dimensional vector space. The GS kernel has four parameters, namely, two for controlling the importance of comparing the physico-chemical properties of amino acids, one for setting the maximum length of k-mers and one controlling the penalty enquired due to the relative distance of k-mers. As described in the next section, these parameters can be tuned by cross-validation.

2.2.4. Implementation details

In order to ensure reproducibility, all experimentations were conducted using the SVM implementation of the Scikit-Learn library (Pedregosa et al., 2011) and the GS kernel (Giguère et al., 2013), available at http://graal.ift.ulaval.ca/gs-kernel/, both free and open source softwares.

3. Theory/calculation

3.1. Assessing model performance

To assess to prediction accuracy of the different approaches, we used the area under the ROC curve (AUC) and the F1 score (Bradley, 1997). In addition, we have used the sensitivity and the specificity to analyse the performance of our method.

The Receiver Operating Characteristic (ROC) curve provides a graphical illustration of a predictor's recall and specificity by plotting the recall and (1 — specificity) as a function of the threshold. This type of curve can be used to select a threshold with respect to some trade-off between recall and specificity. The AUC is a threshold independent metric obtained by computing the area under the ROC curve. This metric is closely related to the one used by (Zhang and Brusic, 2013) to assess the performance of the methods submitted to the 2012 MLI competition, which is given by

$$\sqrt{Sensitivity} + \sqrt{Specificity} + Sensitivity \cdot 10^{-5}. \quad (1)$$

Indeed, Brusic et al. (2013) ranked the methods by selecting the threshold maximizing, thus making their metric threshold independent.

In the field of machine learning, the accuracy is a threshold dependent metric that is used to evaluate prediction methods. Unfortunately, the datasets used to train our method are unbalanced, which means that the number of eluted peptide is much smaller than the number of non-eluted peptides. Thus, using the accuracy to evaluate the performance of our method could be misleading. For example, if only 5% of the peptides contained in a dataset were eluted, a predictor could abstain from predicting any peptide as being eluted and would nevertheless achieve an accuracy of 95%. For this reason, we propose to use the F1 score to evaluate our method's discriminative power. The F1 score is given by Eq. (2) and its value is comprised between 0 and 1. This metric has the advantage of being unaffected by class imbalance.

$$F1\ score = 2 \cdot \frac{precision \cdot sensitivity}{precision + sensitivity} \quad (2)$$

It is important to mention that small changes in the decision threshold can lead to important differences in threshold dependent metrics such as the F1 score. However, it is crucial to compare the ability of a method to estimate a good threshold using the training data. Without a threshold, a method can only provide a confidence score to indicate how confident it is about an example belonging to a target class. Although, for most real world applications, a binary decision must be made. Thus, a threshold is required to distinguish between a positive and a negative decision. Moreover, recent learning paradigms, such as transductive learning (Joachims, 1999), domain adaptation (Jiang, 2008) and positive and unlabelled learning (Elkan and Noto, 2008) aim at learning models specifically designed for a target task by using abundant unlabelled data. These new approaches have been shown to outperform supervised learning when few labelled training examples are available. The additional discrimination power of such methods could be undetected by using only threshold independent metrics. To ensure that improvements made by future methods are discernable, we recommend using threshold dependent metrics such as the F1 score.

3.2. Algorithm parameter selection

For all models, 10-fold cross-validation (see Hastie et al., 2001) on the training set was used for selecting the SVM and the GS kernel parameters. All the metric values reported in the Results and discussion section were computed on the (independent) testing set provided by the organisers of the 2012 MLI competition.

3.3. Advanced parameter tuning

In the cross-validation method, a single parameter is chosen to produce a good predictor with a limited amount of data. This parameter is estimated based on the ability of the algorithm to yield an accurate predictor. However, with a limited amount of data, the uncertainties in the estimations tell us that predictors obtained by using other parameters

values should also be considered. In this case, Bayesian theory suggests to use a (probabilistic) combination of many predictors obtained with different parameter values (Lacoste et al., unpublished). Such a combination allows to take into account the uncertainty on determining which predictor is truly the most accurate. Elaborated repeated resampling techniques allow to estimate the probability of each predictor being the most accurate. These probabilities are then used to weight the contribution of each predictor in the final decision.

In the context of MLI, a preliminary version of this method was used, yielding outstanding results for the HLA-B*07:02, H2-D^b and H2-K^b alleles. However, additional experiments showed that this approach does not significantly improve the AUC or F1 scores.

4. Results and discussion

4.1. Selecting a model for eluted peptide prediction

In this paper, we have presented two eluted peptide prediction models: the one predictor approach (E vs BN) and the two predictor approach (EB vs N + E vs B). In order to determine which model is the most accurate, we computed their AUC and F1 score on the testing set of each allele. The results on all eight alleles are shown in Table 1. For the AUC and F1 score results, we observe that the (EB vs N + E vs B) approach outperforms the (E vs BN) one on six out of eight alleles. Unfortunately, the number of alleles was insufficient to compute p-values, therefore no statistical analysis of the results was made. Nevertheless, the (EB vs N + E vs B) method clearly promises better prediction accuracy than the (E vs BN) method.

This result is interesting from a machine learning point of view, since it indicates that it is better to handle the binding affinity prediction task and the eluted peptide prediction task separately.

Both methods have parameters that are tuned by cross-validation to fit closely to the prediction task. Among the most important parameters to tune, are those of the kernel function which effectively define a specialized similarity function. Generally, theses parameters vary greatly depending on the learning task.

The two predictor approach (EB vs N + E vs B) was tuned twice, once to distinguish binding from non-binding peptides, and a second time, to isolate eluted peptides. The optimal parameters found for each task were considerably different. This result was expected, since both tasks are significantly different and, thus, require different similarity functions. We believe that this is one of the main reasons why the two predictor approach outperforms the single predictor approach. In addition, note that the former exploits the structure of the problem by taking into account that eluted peptides are also binders.

Due to its superior accuracy, we have chosen to use the (EB vs N + E vs B) method to elaborate our eluted peptide prediction tool, MHC-NP. This tool is hosted by the Immune Epitope Database (Peters et al., 2005) and publicly available at http://tools.immuneepitope.org/mhcnp/.

4.2. Comparison to binding affinity prediction methods

The purpose of the 2012 Machine Learning Competition in Immunology was to assess the ability of computational methods for predicting peptides naturally processed by the MHC-I pathway. It is known that there exists a correlation between the binding of peptides and their immunogenicity (Toussaint and Kohlbacher, 2009). Therefore, strong binders are more likely to be naturally processed by the MHC pathway. For this reason, we have chosen to compare our most accurate eluted peptide predictor, the (EB vs N + E vs B) approach, to two state-of-the-art MHC-I binding affinity prediction methods. The first method, called (EB vs N), is inspired by the peptide–protein binding affinity work of Giguère et al. (2013). Note that this method is equivalent to performing the first prediction task of the (EB vs N + E vs B) approach. This comparison allows to estimate the additional discriminative power that follows from using a (E vs B) predictor in combination with a binding affinity predictor. For the sake of completeness, we also compare our approach to the popular NetMHC-3.2 (Lundegaard et al., 2008), which was used as a benchmark method in the 2012 MLI competition.

Table 2 shows AUC results on the testing sets for the three methods. Our approach outperforms both the (EB vs N) approach and NetMHC-3.2 on five out of eight alleles. Also, MHC-NP achieves an average AUC of 0.8602, which is greater

Table 1
Comparison of the two eluted peptide prediction models. For each allele, the best results are shown in bold. Alleles followed by a star symbol are those for which our method performed the best in the MLI competition.

MHC allele	AUC		Fl score	
	E vs BN	EB vs N + E vs B	E vs BN	EB vs N + E vs B
HLA-A*02:01	0.8573	**0.8806**	0.4114	**0.4795**
HLA-B*07:02⋆	0.9157	**0.9236**	0.5644	**0.5992**
HLA-B*35:01	0.9187	**0.9367**	0.6720	**0.7059**
HLA-B*44:03	**0.8178**	0.7947	**0.5833**	0.5443
HLA-B*53:01⋆	0.8401	**0.8515**	**0.6368**	0.5758
HLA-B*57:01	0.8017	**0.8258**	**0.6623**	0.6447
H2-D^b⋆	**0.8614**	0.8437	0.3125	**0.3724**
H2-K^b⋆	0.8185	**0.8251**	0.3212	**0.3421**
Average	0.8539	**0.8602**	0.5205	**0.5330**

Table 2
Comparison of the MHC-NP (EB vs N + E vs B) eluted peptide prediction method and two binding affinity prediction methods using the area under the ROC curve. For each allele, the best result is shown in bold. Alleles followed by a star symbol are those for which our method performed the best in the MLI competition.

MHC allele	MHC-NP (EB vs N + E vs B)	EB vs N	NetMHC-3.2
HLA-A*02:01	0.8806	0.9078	**0.9310**
HLA-B*07:02⋆	**0.9236**	0.9075	0.9042
HLA-B*35:01	**0.9367**	0.9307	0.9090
HLA-B*44:03	0.7947	0.7778	**0.8104**
HLA-B*53:01⋆	**0.8515**	0.7817	0.6651
HLA-B*57:01	0.8258	**0.8438**	0.8181
H2-D^b⋆	**0.8437**	0.8031	0.7641
H2-K^b⋆	**0.8251**	0.8106	0.8098
Average	**0.8602**	0.8454	0.8265

Table 3
Comparison of the MHC-NP (EB vs N + E vs B) eluted peptide prediction method and two binding affinity prediction methods using the F1 score. For each allele, the best result is shown in bold. Alleles followed by a star symbol are those for which our method performed the best in the MLI competition.

MHC allele	MHC-NP (EB vs N + E vs B)	EB vs N	NetMHC-3.2
HLA-A*02:01	0.4795	0.4452	**0.5833**
HLA-B*07:02⋆	**0.5992**	0.4791	0.5239
HLA-B*35:01	0.7059	**0.7432**	0.7417
HLA-B*44:03	**0.5443**	0.4655	0.4697
HLA-B*53:01⋆	**0.5758**	0.5030	0.4568
HLA-B*57:01	0.6447	**0.7085**	0.3006
H2-D^b⋆	**0.3724**	0.3008	0.0833
H2-K^b⋆	**0.3421**	0.3408	0.3000
Average	**0.5330**	0.4983	0.4324

than both binding affinity prediction methods who respectively obtained 0.8454 and 0.8265.

As seen in Table 3, our method outperforms the EB vs N approach and NetMHC-3.2 on five out of eight alleles. Our approach also achieved the best average F1 score: 0.5330, in comparison to 0.4983 and 0.4324 respectively obtained by the EB vs N and the NetMHC-3.2 method.

Considering our results, it is unclear why, for some alleles, the simpler approach of predicting MHC–peptide binding affinity outperforms the MHC-NP method at predicting eluted peptides. Part of the 2012 MLI competition was to assess if the latter task was learnable. Our results show that, for most alleles, this task can be learned with good accuracy. The performance of the NetMHC-3.2 tool for the HLA-A*02:01 allele and the (EB vs N) approach for the HLA-B*35:01 and HLA-B*57:01 alleles could be attributed to noise in the naturally processed peptides data.

Finally, Table 4 reports the sensitivity and specificity of MHC-NP on all alleles. The proposed approach achieves an average sensitivity of 0.4716 and an impressive average specificity of 0.948. Recall that sensitivity and specificity depend on a decision threshold. Considering the high specificity and low sensitivity of MHC-NP, it is reasonable to think that the decision threshold of MHC-NP could be selected to allow a better specificity/sensitivity trade-off. Given the AUC results reported in Table 2, such a threshold clearly exists.

Table 4
Sensitivity and specificity of the MHC-NP (EB vs N + E vs B) method on all alleles. Alleles followed by a star symbol are those for which our method performed the best in the MLI competition.

MHC allele	Sensitivity	Specificity
HLA-A*02:01	0.5000	0.9589
HLA-B*07:02⋆	0.5259	0.9805
HLA-B*35:01	0.5926	0.9795
HLA-B*44:03	0.4725	0.9618
HLA-B*53:01⋆	0.5229	0.9149
HLA-B*57:01	0.6504	0.8091
H2-D^b⋆	0.2784	0.9868
H2-K^b⋆	0.2301	0.9925
Average	0.4716	0.948

5. Conclusions

We proposed a new method, MHC-NP, for the prediction of peptides naturally processed by the MHC pathway. We showed that MHC-NP outperforms state-of-the-art approaches based on MHC binding affinity. Moreover, the results support the hypothesis that peptides that strongly bind the MHC molecule have greater propensity of being naturally processed to the cell surface. In absence of eluted peptide data, the prediction of MHC binding affinity remains a reasonable approach to identify naturally processed peptides. Furthermore, the proposed approach is amenable to be used in conjunction with state-of-the-art pan-specific MHC binding tools to improve its prediction accuracy. Finally, the approach could be further improved by using additional data on molecules that contribute to the MHC pathway.

Acknowledgements

We thank G.L. Zhang, V. Brusic and their collaborators for organising the MLI 2012 competition and providing the data. We thank the Immune Epitope Database for hosting MHC-NP. Computations were performed on the GPC supercomputer at the SciNet HPC Consortium. SciNet is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund — Research Excellence; and the University of Toronto. We also thank Calcul Quebec and Laval University for their support. JC acknowledges the support of the Canada Research Chair in Medical Genomics. This work was supported in part by the Fonds de recherche du Québec - Nature et technologies (FL, MM & JC: 2013-PR-166708) and the NSERC Discovery Grants (FL: 262067, MM: 122405).

References

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recog. 30, 1145.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273.
Elkan, C., Noto, K., 2008. Learning classifiers from only positive and unlabeled data. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, p. 213.
Giguère, S., Marchand, M., Laviolette, F., Drouin, A., Corbeil, J., 2013. Learning a peptide–protein binding affinity predictor with kernel ridge regression. BMC Bioinforma. 14, 82.
Hastie, T., Tibshirani, R., Friedman, J.J.H., 2001. The Elements of Statistical Learning, vol. 1. Springer, New York.
Jiang, J., 2008. A literature survey on domain adaptation of statistical classifiers (online, Mar.).
Joachims, T., 1999. Transductive inference for text classification using support vector machines. Machine Learning: Proceedings of the Sixteenth International Conference (ICML'99), Bled, Slovenia, June 27–30, 1999. Morgan Kaufmann Pub, p. 200.
Leslie, C., Eskin, E., Noble, W.S., 2002. The spectrum kernel: a string kernel for SVM protein classification. Proceedings of the Pacific Symposium on Biocomputing, vol. 7, pp. 564–575. http://dx.doi.org/10.1142/9789812799623_0053 (Hawaii, USA).
Lundegaard, C., Lamberth, K., Harndahl, M., Buus, S., Lund, O., Nielsen, M., 2008. NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8–11. Nucleic Acids Res. 36, W509.
Meinicke, P., Tech, M., Morgenstern, B., Merkl, R., 2004. Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. BMC Bioinformatics 5, 169.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825.

Peters, B., Sidney, J., Bourne, P., Bui, H.-H., Buus, S., Doh, G., Fleri, W., Kronenberg, M., Kubo, R., Lund, O., 2005. The immune epitope database and analysis resource: from vision to blueprint. PLoS Biol. 3, e91.

Platt, J., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers, 10, p. 61.

Qiu, J., Hue, M., Ben-Hur, A., Vert, J.-P., Noble, W.S., 2007. A structural alignment kernel for protein structures. Bioinformatics 23, 1090.

Saigo, H., Vert, J.-P., Ueda, N., Akutsu, T., 2004. Protein homology detection using string alignment kernels. Bioinformatics 20, 1682.

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

Toussaint, N.C., Kohlbacher, O., 2009. Towards in silico design of epitope-based vaccines.

Toussaint, N., Widmer, C., Kohlbacher, O., Rätsch, G., 2010. Exploiting physico-chemical properties in string kernels. BMC Bioinformatics 11, S7.

Zhang, L., Udaka, K., Mamitsuka, H., Zhu, S., 2012. Toward more accurate pan-specific MHC–peptide binding prediction: a review of current methods and tools. Brief. Bioinform. 13, 350.