Prediction Of Peptides Binding To MHC Class I Alleles By Partial Periodic Pattern Mining

Cem Meydan, Uğur Sezerman
Biological Sciences and Bioengineering Dept.
Sabancı University
Istanbul, Turkey
cemmeydan@su.sabanciuniv.edu
ugur@sabanciuniv.edu

Hasan Otu
BIDMC Genomic Center
Harvard Medical School
Boston, MA, USA
hotu@bidmc.harvard.edu

***Abstract*— MHC (Major Histocompatibility Complex) is a key player in the immune response of an organism. It is important to be able to predict which antigenic peptides will bind to a specific MHC allele and which will not, creating possibilities for controlling immune response and for the applications of immunotherapy. However, a problem for MHC class I is the presence of bulges and loops in the peptides, changing the total length. Most machine learning methods in use today require the sequences to be of same length to successfully mine the binding motifs. We propose the use of time-based data mining methods in motif mining to be able to mine motifs position-independently. Also, the information for both binding and non-binding peptides is used on the contrary to the other methods which only rely on binding peptides. The prediction results are between 60-95% for the tested alleles.**

Keywords-motif mining, periodic pattern mining, major histocompatibility complex, machine learning

I. Introduction

MHC (Major Histocompatibility Complex) is a large gene family with an important part of the immune system, autoimmunity and reproduction. MHC molecules take role in destruction of pathogens and diseased cells by showing self and non-self antigen peptides on their surface and coordinating the T-cells which identify these peptides. The T-Cells recognize the infected cell upon binding to the antigenic peptide-MHC complex and trigger the immune response to foreign bodies by a cascade of events. Since they have a key role in immune response, MHCs are critical in many diseases, and they can be used for controlling specific processes by creating peptides to bind to specific MHC alleles. This binding affinity to specific peptides may be exploited for creating peptide vaccines, suppressing specific alleles in organ transplants, and many other possible areas in immunotherapy.

The peptide binding groove in the MHC molecules binds peptides with high promiscuity; it is estimated that each HLA (human leukocyte antigen system) class I protein can bind over 1000 peptides. Thus it is difficult to find specific motifs for experimental studies, and the large number of possible structures makes it infeasible to find them by experiments alone. Computational determination of binding specificity of a given peptide to specific alleles is an important problem in bioinformatics. Although many methods have been proposed, still the accuracy is not near what can be expected for such short motifs. The most state of art prediction servers can predict alleles with 75-95% accuracy for easy classes and 50-65% for hard classes [1, 2] leaving space for improvement.

Various methods are employed for MHC binding peptide prediction [3]. These methods usually depend on identification of 2 to 3 specific anchors. ANN, quantitative matrices, most binding motif miners and other methods relying on sequence information requires the peptides to be in the same length, with appropriate aminoacids aligned to be in the same position. However the difficult classes of peptides show bulges and loops in their structure, changing the length of the peptide from the optimal length of 9. Since these methods cannot handle length variation, they require pre-processing and complex alignment of the data to get reasonable results. Newer methods use results of the sampling of random insertions for elongation and deletions for shortening, meant for fitting the peptide into the 9-length window, thus the 9 limitation is still present in the core.

The required pre-processing step may not be always feasible or give good results on the training set, especially for such short and variable peptides. For this reason, we propose a method which does not require the peptides to be of same length and the anchor positions to be specific, using partial periodic pattern mining. We aim to include a novel method for extracting the motifs which include bulges that can be used on difficult sets which is based on application of sequence mining domain of data mining for ordered episodes. These temporal mining algorithms are usually used in intrusion detection and other future prediction methods, which try to capture the patterns which occur in an order but not necessarily consecutively. Another novel aspect is the use of both binding and non-binding motif information concurrently.

II. Methods

A. *Motif Mining*

The motif mining algorithm is based on the apriori algorithm that is used in frequent itemset discovery. Apriori algorithm uses the principle that all subsets of a frequent itemset must also be frequent. Accordingly, it has a bottom-up approach where the shorter frequent itemsets are extended to create longer candidates, which are then filtered by frequency of occurrence [4-6]. The longer frequent itemsets are also extended and this iterative extension process continues until no frequent itemsets of a length can be found.

Our motif mining method is similar to temporal event mining in time-related databases. In general, the partial periodic pattern mining algorithms for time series data will try to find frequently co-occurring events, or causality relationships between them. In the domain of protein motifs, the aminoacids become the "events", and the causality/future prediction aspects become the motifs that are sought [7]. In the approach, each sequence is taken as a separate time series, with many parallel events occurring at the same time. In these time series, if an event happens frequently after another event occurs, within a given time window, it is considered an episode of events, a motif.

In our method, first the frequent itemsets of size 1, F(1), are found. The first step is straightforward, only the aminoacid counts at different positions within the sequences are counted, and if their frequency (support of the rule) is below the given threshold, they are filtered out. Then the candidate set of size 2, C(2) is created from the aminoacids by F(1) → F(1); a motif of length 2 which is created by concatenating every aminoacid (frequent motif of length 1) to each other, creating motifs such as Leu→Val. The sequences in the dataset are checked for whether Leu is followed by Val within a window. A specific window is defined as being between at least *(minimum space)* away and at most *(maximum space)* away. If the aminoacids co-occur within this window by a specific order, at least (*minimum support* x *sequence count*) times, then the motif is considered frequent. By this method, all of the candidate motifs are filtered by the minimum support and confidence values given, creating F(2). Thus, iteratively F(n) is created from filtering of C(n)=F(n-1)→F(1).

In the motif mining context, the frequent rules are not association rules as in a shopping basket analysis; they have a time value which is used for relations such as "before"/"after". Then the episodes become, "if A occurs in a given position, B will likely to occur within n to m positions after A with probability of p and confidence of c". There are two parameters, the slack length (s), which is the length after an event within which we do not look for a rule, and the window size (w), in which the consequent event may occur. Thus, n=s+1 and m=s+w-1 in the above definition, and the rule is given as A→B (p, c) for parameters (s, w). The rule may also consist of 3 or longer events, such as A→B→C.

While experimenting, we used window size of 3 and slack length of 0 to 8, which produced different rulesets. For s=0, the rules that consist of consecutive/nearby aminoacids are mined whereas for larger values of s, the motifs consisting of aminoacids at separate ends of the peptide are found. Since the anchor positions of MHC motifs may be different, different slack lengths are needed to mine them all.

B. *Prediction*

Once the rules are mined, these rules are used in the prediction and scoring process. Before prediction, rules from both the binding and non-binding sequences are mined separately. During classification of an unknown peptide, the peptide is scored independently by both of the binding and non-binding rules. The simplest classification method is the direct comparison of the scores for binding/non-binding by summing the support values of the rules that occur in the given peptide. However, the binding and non-binding datasets are usually not balanced due to the very low count of non-binding peptides resulting in the support values thus the rule count for negative class to be substantially higher. To overcome this problem, sum of both classes are normalized. Hence, for two rules with the same support value, one that is found in the dataset with the lower count of rules has a higher score, considering that rule is much important for that class separation than the other. For a training dataset an optimal multiplier for both binding and non-binding may be found that separates the scores with the greatest threshold. We added an optimization step for the weights for positive and negative classes and also the best cut-off value to use as a threshold for class separation.

III. Results

A. *Data Set*

The dataset used is MHCBN from Raghava et al. [8]. The total database consists of 25860 peptides, 20717 binders and 4022 non-binders. The alleles HLA-A*0201, HLA-A*2, H-2Kb and HLA-B*3501 that have sufficient binder/non-binder data are used in testing. The binding affinity values of high/medium/low are combined to create the binder dataset and the rest are taken as non-binder for a binary value. The actual affinity values are not used in the mining/scoring process.

B. *Experiments*

Unbalanced datasets reduce the accuracy dramatically. However, resampling the non-binding peptides or undersampling the binding peptides does not increase the accuracy and sometimes decrease it as well [9]. To overcome this problem, we used the binding peptides to generate non-binding samples. While the patterns for non-binding can be mined by looking at what occurs in non-binding sequences, they can also be mined by looking at what does not happen in the binding peptides. Since the binding peptide count is high, the distribution of the aminoacids on a specific position was found and a new sequence was generated with aminoacids inversely proportional to the ones found in the binding sequence, i.e. for every position *i*, a random aminoacid is placed, with probability of the aminoacid *R* being selected inversely proportional to the occurrence of *R* in position *i* in all of the binding sequences. Thus, for example, if none or very few of the peptides binding allele HLA-A*0201 have { D, E, R, K } in position 3, then it is likely that these aminoacids are negatively affecting the binding affinity of the peptide [10]. Since it is possible that the non-binding peptides are not varied enough to capture this pattern, newly generated non-binder sequences can help in this process. However care must be taken to not suppress the actual non-binding sequences since there is no guarantee that the generated sequences actually have patterns that help in the classification.

For HLA-A*0201, the ratio of positive to negative class was about 23 to 1, to balance this ratio to more acceptable levels without under-representing the actual non-binder data, an additional of 100 synthetic non-binder peptides were created to compare the effects with and without these synthetic peptides. Each allele is tested by dividing the data into 80% training 20% testing sets randomly, a total of 25 times for an allele. The average, maximum and minimum results for the 4 datasets, with both training and testing set accuracies can be found in Table 1.

It can be seen that the predictions have acceptable accuracy values of 70 to 80%. For HLA-A*0201, the false positive rate is the result of low non-binding count. If we look at the peptides that are classified as binding, when in fact non-binding, they carry very strong binding patterns, such as the L→{L-I-V} pattern in anchor positions of HLA-A*0201. These aminoacids in the 2-9 positions are accepted by the literature as good binders. The peptides that are classified as false positives carry these patterns and other strong patterns. It is obvious that they carry another part that suppresses the affinity of the binding motif to the MHC allele. While the method marks some peptides with good positive scores as non-binding due to the presence of a non-binding signal, it cannot catch them all, possibly due to the lower count of the negative dataset. However an important point to consider is that the non-binding accuracies given do not reflect the whole domain of the non-binders, since the dataset has an experimental bias. The sequences tested and marked as non-binding are either poly-alanine sequences or known binding sequences, on which mutations are carried out repeatedly to find the binding position and rules. Our prediction method will accurately find non-binders that do not carry the binding patterns, which are under-represented in this dataset, showing lower negative prediction accuracy than the actual value. To solve these problems, the method of rule cascade was proposed. Basically, the algorithm will try to cascade the rules for the optimal decision making. If a binding motif in a peptide is followed by a non-binding signal, it would most likely non-binding, if the non-binding signal is strong to inhibit binding process. A decision tree like structure and a SVM classifier was built on the presence of the different signals in each class to assist in classification. Predictor vectors to be used in these classifiers are created by the binary values of rules in a given peptide. If a peptide contains a motif, it is 1 for that column, otherwise 0. When coupled with the class value of binder (1) and non-binder (0), the dataset can be classified with different classifiers.

The accuracy values for different classifiers are given in Table 2. The classifiers were all tested with 5-fold cross validation. However some of them were tested by creating the whole vector set, i.e. whole dataset was used to mine the rules, and then when the dataset is created, it is tested by

TABLE 1: The results for the prediction of 4 MHC class I alleles, with 80% training, 20% testing set separation repeated for 25 times.

DataSet		Accuracy			Sensitivity			Specificity			Precision		
		Avg	*Max*	*Min*	*Avg*	*Max*	*Min*	*Avg*	*Max*	*Min*	*Avg*	*Max*	*Min*
HLA-A*0201 (1390 Pos, 60 Neg)	***Train***	0.806	0.838	0.757	0.808	0.843	0.754	0.756	0.854	0.646	0.987	0.992	0.984
	Test	0.794	0.852	0.762	0.802	0.860	0.773	0.620	0.917	0.333	0.980	0.995	0.964
HLA-A*0201 + 100 Synthetic Neg	***Train***	0.807	0.876	0.712	0.808	0.890	0.708	0.771	0.854	0.708	0.988	0.992	0.978
	Test	0.795	0.883	0.728	0.804	0.914	0.720	0.607	0.917	0.417	0.979	0.995	0.962
HLA-A*2 (682 Pos, 222 Neg)	***Train***	0.720	0.751	0.684	0.714	0.772	0.655	0.739	0.853	0.684	0.897	0.919	0.883
	Test	0.747	0.808	0.676	0.809	0.891	0.715	0.556	0.733	0.378	0.849	0.888	0.803
H-2Kb (255 Pos, 43 Neg)	***Train***	0.639	0.769	0.492	0.601	0.775	0.422	0.871	0.971	0.735	0.967	0.992	0.942
	Test	0.580	0.750	0.400	0.577	0.824	0.333	0.596	0.889	0.222	0.890	0.972	0.854
HLA-B*3501 (295 Pos, 22 Neg)	***Train***	0.811	0.941	0.680	0.812	0.958	0.657	0.798	1.000	0.647	0.983	1.000	0.972
	Test	0.775	0.941	0.609	0.793	0.958	0.593	0.563	1.000	0.200	0.957	1.000	0.891

TABLE 2: The results for HLA-A0201 dataset with different classifiers. As it can be seen, binder and non-binder sample count being unbalanced affects the specificity negatively. SVM on the first column, even being tested on the test data instead of the training and is expected to be less accurate, shows greater accuracy and specificity than any of the other methods.

Method	Pos #	Neg #	Accuracy	Sensitivity	Specificity	Precision
SVM (20% test set accuracy, with Synthetic Non-binders)	278	112	0.962	0.975	0.929	0.971
SVM (whole set accuracy)	1390	60	0.951	0.979	0.300	0.970
Naïve Bayesian	1390	60	0.823	0.836	0.533	0.976
Naïve Bayesian Multinomial	1390	60	0.888	0.902	0.567	0.980
Decision Tree	1390	60	0.916	0.939	0.383	0.972

cross-validation. This introduces a bias into the accuracy, because the rule mining process is not external to the test set. Other samples were tested by creating the vectors from the rules being mined from the 80% training set, classifiers being trained, again, on the training set and finally tested on the test set. This method does not carry a bias. However, the results did not show specifically any deviation from the whole-set and test-set accuracies, so only the first row is shown from those sets.

Our aim was to show if the more complex methods would give better results. As it can be seen in Table 2, the unbalanced counts of binders and non-binders negatively affected the classifiers as well. For all of the trials, the predictions were biased to the positive dataset. The classifiers try to minimize the error rate, and choose to err on the side of negatives (which are 1/23 in ratio to the binders in HLA-A*0201), thus giving greater accuracy but very low specificity. Balancing the training set by the addition of 100 synthetic non-binders improved the specificity dramatically (first row in Table 2). It can be seen that SVM gives 97.5% sensitivity and 92.9% specificity, much better than both other classifiers and the previous methods in Table 1. This accuracy is given by the training/testing separated data; even though it is expected to be lower than the other methods, the balanced set improves the prediction strength. Note that, the representative strength of the correctly classified synthetic non-binders is open to debate.

IV. Conclusion

We developed a method that uses sequential pattern mining schemes for finding the most probable binding motif, with position- independent information that can be applied to the peptides of arbitrary length to accommodate for the sequences with insertions and loops between the anchor positions. The frequent partial periodic rules that can explain most of the peptides are mined from the training set using different windows for position-independent episodes. For the same allele, the non-binding peptide information is also used for mining motifs for non-binding, since the mined episodes may contain arbitrary episodes that are not related to the binding affinity. Also, some additional peptides in the non-binding aminoacid positions may cause a previously binding peptide to become non-binding. Thus, we mined frequent rules for both binding and non-binding peptides, and use the exclusive set of the two for scoring the peptides. The peptides are scored according to the support and confidence of the frequent episodes they contain. From this study, position independent motifs mined with representing the aminoacid sequence as time series data proved to be usable for prediction of the binding peptides to MHC class I proteins. Although the accuracy of the algorithm is not state-of-the-art, it is in the same range. The pattern mining method can be improved upon to include some position dependency as anchor points or windows, and by the addition of rule merging/splitting for better class separation.

References

[1] H. H. Lin, S. Ray, S. Tongchusak, E. L. Reinherz, and V. Brusic, "Evaluation of MHC class I peptide binding prediction servers: applications for vaccine research," *BMC Immunol*, vol. 9, p. 8, 2008.

[2] B. Peters, H. H. Bui, S. Frankild, M. Nielson, C. Lundegaard, E. Kostem, D. Basch, K. Lamberth, M. Harndahl, W. Fleri, S. S. Wilson, J. Sidney, O. Lund, S. Buus, and A. Sette, "A community resource benchmarking predictions of peptide binding to MHC-I molecules," *PLoS Comput Biol,* vol. 2, p. e65, Jun 9 2006.

[3] B. Trost, M. Bickis, and A. Kusalik, "Strength in numbers: achieving greater accuracy in MHC-I binding prediction by combining the results from multiple prediction tools," *Immunome Res,* vol. 3, p. 5, 2007.

[4] R. Srikant and R. Agrawal, "Mining generalized association rules," *Future Generation Computer Systems,* vol. 13, pp. 161-180, Nov 1997.

[5] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc., 1994.

[6] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data* Washington, D.C., United States: ACM, 1993.

[7] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining and Knowledge Discovery,* vol. 1, pp. 259-289, Nov 1997.

[8] M. Bhasin, H. Singh, and G. P. Raghava, "MHCBN: a comprehensive database of MHC binding and non-binding peptides," *Bioinformatics,* vol. 19, pp. 665-6, Mar 22 2003.

[9] A. P. Sales, G. D. Tomaras, and T. B. Kepler, "Improving peptide-MHC class I binding prediction for unbalanced datasets," *BMC Bioinformatics,* vol. 9, p. 385, 2008.

[10] J. G. Houbiers, H. W. Nijman, S. H. van der Burg, J. W. Drijfhout, P. Kenemans, C. J. van de Velde, A. Brand, F. Momburg, W. M. Kast, and C. J. Melief, "In vitro induction of human cytotoxic T lymphocyte responses against peptides of mutant and wild-type p53," *Eur J Immunol,* vol. 23, pp. 2072-7, Sep 1993.