Predicting Peptides Binding to MHC Class II Molecules Using Boosted Decision Tree

Haneen Tartory
Master of Informatics
Palestine Polytechnic University
Abu Roman St.,P.O.198,Hebron,Palestine
haneen@student.ppu.edu

Hashem Tamimi
Biotechnology Research Center
Palestine Polytechnic University
Abu Roman St.,P.O.198,Hebron,Palestine
htamimi@ppu.edu

Yaqoub Ashhab
Biotechnology Research Center
Palestine Polytechnic University
Abu Roman St.,P.O.198,Hebron,Palestine
yashhab@ppu.edu

May 17, 2012

Abstract

Prediction of MHC class II binding peptides represents a challenging problem in machine learning. Many researchers applied different machine learning tool for the prediction, these tools are: SVM, neural network, genetic programming, and HMM, but each has its own strengths and weaknesses. In this paper we used the Boosted decision tree algorithm for the prediction using two different methods in representing the peptide sequences. The experiments results show that the boosted decision tree algorithm can be developed to give a good algorithm for MHC II prediction problem.

1 Introduction

The Major Histocompatibility Complex (MHC) is a large genomic region or gene family found in most vertebrates that encodes MHC molecules[1], that plays an important role in the immune system and autoimmunity [1]. Only a small fraction of the possible peptides that can be generated from proteins actually generate an immune response[1]. MHC molecules act as receptors for peptides derived from foreign antigens as well as self peptides and enable the long-term display of antigens on the cell surface [2].

There are two major types of MHC molecules are involved in the peptide binding process; The class I MHC molecules found on almost all cell types present antigens to T cells, whereas Class II MHC molecules on antigen presenting cells present antigens to T helper cells [2].

Prediction of peptide-MHC binding represents an important goal in bioinformatics, because of their role in the immune system. Prediction of peptides binding to a MHC class II molecule is more difficult than MHC class I due to different length of the binding peptides is longer than 9mer [3].

Recently, many studied focused on prediction of peptide binding to MHC II depending on different machine learning tools, such as neural networks, geneting programming, and hidden markov model (HMM), support vector machine (SVM), and others. This paper aims at predicting peptides binding to MHC class II molecules using boosted decision tree.

In this paper, three major physicochemical properties (size, charge and hydrophobicity) [4] were used

depending on compatibility matrices, in order to represent the sequence of epitope, in addition to the characters sequence. The experiments results show that the boosted decision tree algorithm can be developed to give good results for MHC II prediction problem.

The rest of this paper is organized as follows. Sec. 2 introduces the boosted decision tree algorithms. Experimental results are given in Sec. 3. Sec. 4 concludes this paper. Sec. 5 introduces future work.

2 Boosted Decision Tree

Boosting refers to a general and effective method of producing a very accurate prediction rule by combining rough and moderately inaccurate rules of thumb (Freund, et al., 1999) [5]. The first practical boosting algorithm was the AdaBoost algorithm, that proposed by Freund and Schapire 1995, to solved many of the practical difficulties of the earlier boosting algorithms (Freund, et al., 1997) [6]. It is a classification algorithm; uses weak classifier (classifier that gives more than 50% correct result, better than random), and finally combines them in one strong classifier. Below algorithm shows the adaboost algorithm, where h represents the weak classifier and y is the class (-1 or 1) [6], Figure 1 shows the Adaboost algorithm.

Decision tree is considered a weak classifier that can be boosted. There are different methods can be used to apply boosted decision tree. One of these methods uses a decision stump, which is simply a decision tree with a single branch (Rennie, 2003)[7]. A single decision stump is a weak learner it does not perform particularly well but an ensemble of decision stumps can perform as well as or better than a full-blown decision tree, which is a sequence of binary splits of the data (Rennie, 2003)[7]. Boosted recombined weak classifier (Rodr?'guez, et al., 2008) [8]is an example of how we can use decision stumps to create a weak classifier. In their work each time a new decision stump is constructed, a tree is obtained from that decision stump and the decision stumps from previous iterations. The method has a parameter, the level of reuse. It is the number of classifiers from the former iterations that are going to be used [8]. Other method, is depending on the varying the weight for each tree (ROE, et al., 2006) [9]. ROE, et al., 2006 explained how we can build a decision tree, depending on finding the best variable and splitting point which gives the best separation using gini index [9]. To apply boosting decision tree as proposed by ROE, et al., 2006, for each tree iteration, same set of training sample are used but the weights of misclassified events in previous iteration are increased (boosted). Events with higher weights have larger impact on Gini index values and Criterion values. The use of boosted weights for misclassified sample makes them possible to be correctly classified in succeeding trees [9]. In their experiment they find that the boosted decision tree is 20%-80% better than that with ANN on MiniBooNE partical identification variables (PID)[9].

Initialization step: for each example x, set
$D(x)=\frac{1}{N}$, where N is the number of examples

Iteration step (for t=1…T):

1.Find best weak classifier $h_t(x)$ using weights $D_t(x)$

2. Compute the error rate ε_t as

$$\varepsilon_t = \sum_{i=1}^{N} D(x_i).I[y_i \neq h_t(x_i)]$$

3. assign weight α_t to classifier $h_t(x)$ in the final hypothesis

$$\alpha_t = \log((1-\varepsilon_t)/\varepsilon_t)$$

4. For each x_i, $D(x_i) = D(x_i).\exp(\alpha_t.I[y_i \neq h_t(x_i)])$

5.Normalize $D(x_i)$ so that $\sum_{i=1}^{N} D(x_i) = 1$

$$f_{final}(x) = sign\ [\sum \alpha_t\ h_t(x)]$$

Figure 1: The boosting algorithm AdaBoost.

3 Experiments and Results

3.1 Data Source

Peptide datasets used in this study are available from the NetMHCII 2.2 server (http://www.cbs.dtu.dk/suppl/immunology/NetMHCII-2.0.php, Nielsen M and Lund O, 2009)[10]. The

dataset was used in this paper is DRB1*0101 datasets which contains 5166 epitope sequences divided into 5 fold.

When classifying the peptides into binders and non-binders, a threshold value is used. This means that peptides with binding affinity values greater than 0.426 are classified as binders [10].

The main characteristics of this data set, that it contains a 5166 epitope sequences, 1656 are non binders, and 3510 are binders. That mean the non binders sequences represent one third of the binders. The second characteristic is that the longest epitope sequence contains 37 amino acid.

3.2 Boosted decision tree experiment and result using physicochemical properties

In our first experiment we used three physicochemical properties to represent each amino acid(Biro, 2006)[4]; hydrophobic, charged and size. Hydrophobic which is a measure of how strongly the side chains are pushed out of water (Eisenber, et al., 1982) [11]. In the charge, the opposite charges attract and similar charges repel each other. The charge of a molecule is PH dependent (Biro, 2006)[4]. There is a considerable variation in the sizes of amino acids. Theoretically, there might be size complementarily between amino acids, similar to nucleic acid base pairs (Biro, 2006) [4]. Table 1 shows these three physicochemical properties (hydrophobic, charged, and size) for each amino acid.

Table 1: physicochemical properties of amino acids [4]

Amino acid	Size	Charged	Hydrophobic
R	156	10.8	-7.5
K	128	9.7	-4.5
D	115	2.8	-3
Q	128	5.7	-2.9
N	114	5.4	-2.7
E	129	3.2	-2.6
H	137	7.6	-1.7
S	87	5.7	-1.1
T	101	5.9	-0.8
P	97	6.5	-0.3
Y	163	5.7	0.1
Y	163	5.7	0.1
C	103	5.1	0.2
G	57	6	0.7
A	71	6	1
M	131	5.7	1.1
W	186	6	1.5
L	113	6	2.2
V	99	6	2.3
F	147	5.5	2.5
I	113	5.9	3.1

In boosted decision tree, the number of sample from each class should be approximately equals, and because the in previous dataset the non binders sequences represent one third of the binders, so we apply boosted decision tree many times and each time we choose from binders random sample equals non binders sequences.

In this paper, we used the boosted recombined weak classifier [8] method to apply boosted decision tree. We built the decision stumps to apply this method using a suitable threshold for each physicochemical property after normalization, our thresholds was 0.6981 for hydrophobic property, 0.5556 for charged property, and 0.6129 for size property. We apply thresholds for each position in the epitope sequence (37 position), where each position have three values (physicochemical values).

If the three values for each epitope position value larger than the thresholds, it classified to 1 if the largest sample has in the actual classifier 1 more than 0, other samples took 0 values, so we have a 37*3 decision stumps. Then, we combined these decision stumps to create 111 weak classifier, these classifiers entered to adaboost algorithm to compute alpha values in order to use it in the test samples.

The boosted recombined weak classifier has one parameter which is r (the level of reuse). The suitable value of r using this method as shown in Figure 2 was 5 because it gave good results. Three fold cross validation was used, and we applied the boost-

ing algorithm 10 times, each time we took random sample from binders sample equals non-binders. The ROC analysis (sensitivity, specificity, accuracy, negative predictive value (NPV), positive predictive value (PPV), Area under the ROC curve (AUC) (Fawcett, 2006)[12] was used to evaluate the performance of this experiment. The score values we used in this paper to draw the ROC curve was the result of final step of the adaboost algorithm (f (final)) without the sign (Kawakita, et al., 2005) [13]

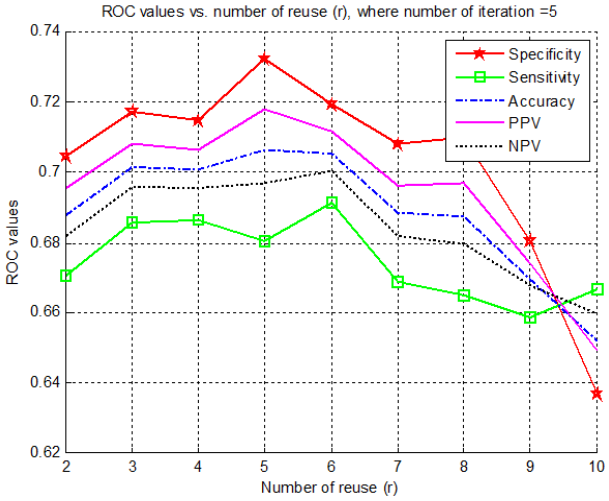

Figure 2: The sensitivity, specificity, accuracy, PPV and NPV values according to change in r for the first experiment.

Table 2 shows the results for this experiment, when r equal 5 and number of iterations equal 10.

Table 2: Results for the physicochemical properties experiment for 10 iteration.

Average sensitivity	0.7131
Average specificity	0.71
Average accuracy	0.706522
Average PPV	0.71820
Average NPV	0.697168
Average area under the ROC curve	0.7768

Figure 3 shows the ROC curve and the cutoff point for the first iteration in this experiment.

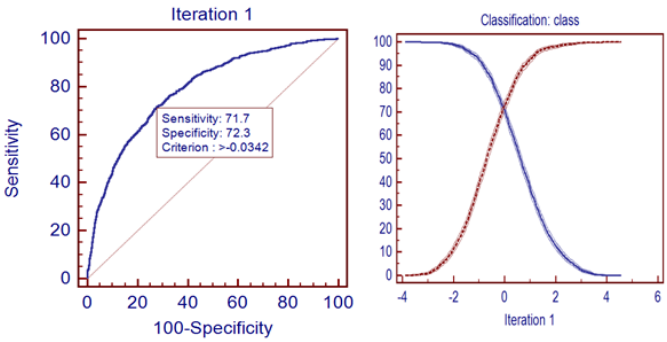

Figure 3: ROC curve for the first iteration, and the cutoff point

3.3 Boosted decision tree experiment and result using characters sequence

If we want to search about all amino acids at each location, we should build a 740 (37*20) decision stumps, which is considered more complex and time consuming, so to decrease this complexity the physicochemical character instead of the alphabetical sequences method was used (Tomita et al., 2008)[14]. In their work, Tomita et al. 2008, classified amino acids into five categories using representative properties. These groups are: bulky (WY), small (AG), hydrophobic (IVLFCM), positively charged (RKH), negatively charged (DE), and they excluded (PNQST) amino acids which represent the middle amino acids of the physicochemical properties. In this paper, we used these groups of amino acids and we considered the excluded amino acids a new group, so we had 6 groups of amino acids. In this experiment we decoded the epitope sequence into those 6 groups and completed the sequences into 37 using 0 values. To create decision stumps, for each epitope we find if the position 1 have group 1, it classified to 1 if the largest sample have in the actual classifier 1 more than 0, other samples took 0 values, so we have a 37*6 decision stumps. Then, we combined these decision

stumps to create 222 weak classifier, these classifiers entered to adaboost algorithm to compute alpha values in order to use it in the test samples. In this experiment the suitable value or r that gave a good results was 8, as we can see in Figure 4, where three fold cross validation was used, and we applied the boosting algorithm 5 times, each time we took random sample from binders sample equals non-binders. The ROC analysis [12] was used to evaluate the performance of this experiment.

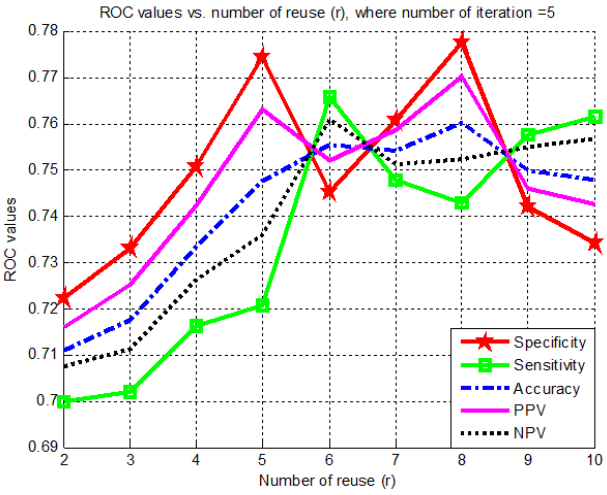

Figure 4: The sensitivity, specificity, accuracy, PPV and NPV values according to change in r for the first experiment.

Table 3 shows the results for this experiment, when r equal 8 and number of iterations equal 5.

Table 3: Results for the physicochemical characters experiment for 5 iteration.

Average sensitivity	0.7414
Average specificity	0.7828
Average accuracy	0.7601
Average PPV	0.7701
Average NPV	0.7522
Average area under the ROC curve	0.838

Figure 5 shows the ROC curve and the cutoff point for the second iteration in this experiment.

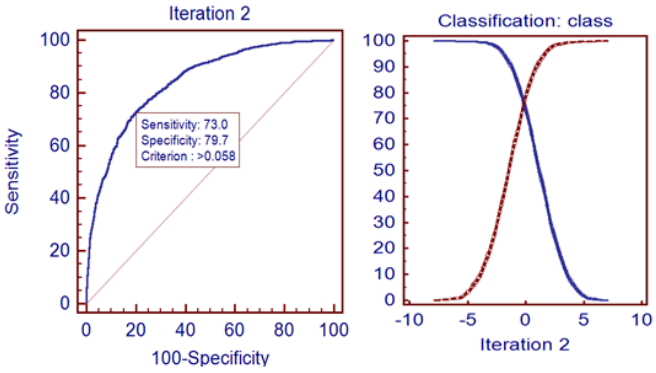

Figure 5: ROC curve for the second iteration, and the cutoff point

4 Conclusion

We present two experiments for predicting MHC class II predicting using boosted decision tree, one using the physicochemical properties values and the other for characters sequences. Our results show that the using of characters sequence using boosted decision tree have a good results than the physicochemical values. Table 4 shows the comparison of AUC values on allele DRB1-0101 as mentioned in Nielsen, et al., 2009 [15],and Wang, et al., 2008 [16], from this table we can notice that the boosted decision tree can be developed to give a good results for predicting MHC II binding .

5 Future work

There are many possible directions for future work. For the physicochemical properties method we can propose at least two thresholds on each physicochemical property, and we can use additional amino acids physicochemical properties not only size, charge, and hydrophobic. Another direction in the future work, in using the characters sequence, we can use all amino acids not groups of them. In the future we can use

Table 4: Table 4:Comparison of AUC values on allele DRB1-0101.

An artificial neural network-based alignment algorithm including data redundancy step-size rescaling and P1-PSSM encoding (NN-W-P1) [15]	0.88
An artificial neural network-based alignment algorithm including data redundancy step-size rescaling (NN-W) [15]	0.87
SVRMHC (support vector regression) [16]	0.69
MHC2Pred (support vector machine) (Wang, et al., 2008) [16]	0.67
Boosted decision tree (using physicochemical properties)	0.78
Boosted decision tree (using characters sequence)	0.84

the sequence to feature method [17] to represent the sequences, and also we can apply another algorithm of boosted decision tree such as algorithm mentioned in ROE, et al., 2006 [11], in order to improvement the results.

References

[1] L. G. Tussey and A. J. McMichael, *General introduction to the MHC*. Cambridge Books Online, 1995.

[2] L. Handunnetthi, S. V. Ramagopalan, G. C. Ebers, and J. C. Knight, "Regulation of mhc class ii gene expression, genetic variation and disease," *UKPMC Author Manuscripts*, pp. 99–112, 2009.

[3] M. Rajapakse, B. Schmidt, L. Feng, and V. Brusic, "Predicting peptides binding to mhc class ii molecules using multi-objective evolutionary algorithms," *BMC Bioinformatics*, 2007.

[4] J. Biro, "Amino acid size, charge, hydropathy indices and matrices for protein structure analysis," *Theor Biol Med Model*, 2006.

[5] Y. Freund and R. E. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, pp. 771–780, 1999.

[6] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[7] J. D. M. Rennie, "Boosting with decision stumps and binary features.." `http://people.csail.mit.edu/jrennie/writing`, 2003.

[8] J. s. M. Juan J. Rodrguez, "Boosting recombined weak classifiers," *ScienceDirect*, vol. 29, p. 10491059, 2008.

[9] B. P. ROE, "Boosted decision trees, a powerful event classifier," *World Scientific Publishing Co.*, pp. 139–142, 2006.

[10] C. Immunological Bioinformatics, "Supplementary material." `http://www.cbs.dtu.dk/suppl/immunology/NetMHCII-2.0.php`.

[11] D. EISENBERG, R. M. WEISS, T. C. TERWILLIGER, and W. WILCOX, "Hydrophobic moments and protein structure," *Faraday Symp. Chem. Soc.*, pp. 109–120, 1982.

[12] T. Fawcett, "An introduction to roc analysis," *ScienceDirect*, vol. 27, pp. 861–874, 2006.

[13] M. Kawakitaa, M. Minamia, S. Eguchia, and C. Lennert-Codyc, "An introduction to the predictive technique adaboost with a comparison to generalized additive models," *ScienceDirect*, vol. 76, pp. 328–343, 2005.

[14] Y. Tomitaa, R. Katoa, M. Okochia, and H. Honda, "A motif detection and classification method for peptide sequences using genetic programming," *Journal of Bioscience and Bioengineering*, vol. 106, pp. 154–161, 2008.

[15] M. Nielsen and O. Lun, "Nn-align. an artificial neural network-based alignment algorithm for mhc class ii peptide binding prediction," *BMC Bioinformatics*, 2009.

[16] P. Wang, J. Sidney, C. Dow, B. Mothe, A. Sette, and B. Peters, "A systematic assessment of mhc class ii peptide binding predictions and evaluation of a consensus approach," *PLoS Comput Biol*, 2008.

[17] Y. EL-Manzalawy, "Predicting flexible length linear b-cell epitopes," *World Scientific Publishing Co.*, vol. 7, pp. 121–132, 2008.