Chapter 22
Improved Prediction of MHC Class I Binders/Non-Binders Peptides Through Artificial Neural Network Using Variable Learning Rate: SARS Corona Virus, a Case Study

Sudhir Singh Soam, Bharat Bhasker, and Bhartendu Nath Mishra

Abstract Fundamental step of an adaptive immune response to pathogen or vaccine is the binding of short peptides (also called epitopes) to major histocompatibility complex (MHC) molecules. The various prediction algorithms are being used to capture the MHC peptide binding preference, allowing the rapid scan of entire pathogen proteomes for peptide likely to bind MHC, saving the cost, effort, and time. However, the number of known binders/non-binders (BNB) to a specific MHC molecule is limited in many cases, which still poses a computational challenge for prediction. The training data should be adequate to predict BNB using any machine learning approach. In this study, variable learning rate has been demonstrated for training artificial neural network and predicting BNB for small datasets. The approach can be used for large datasets as well. The dataset for different MHC class I alleles for SARS Corona virus (Tor2 Replicase polyprotein 1ab) has been used for training and prediction of BNB. A total of 90 datasets (nine different MHC class I alleles with tenfold cross validation) have been retrieved from IEDB database for BNB. For fixed learning rate approach, the best value of AROC is 0.65, and in most of the cases it is 0.5, which shows the poor predictions. In case of variable learning rate, of the 90 datasets the value of AROC for 76 datasets is between 0.806 and 1.0 and for 7 datasets the value is between 0.7 and 0.8 and for rest of 7 datasets it is between 0.5 and 0.7, which indicates very good performance in most of the cases.

Keywords Variable learning rate · Artificial neural network · SARS Corona virus · MHC class I binder/non-binder · Epitope prediction · Vaccine designing · T-cell immune response

B.N. Mishra (✉)
Department of Biotechnology, Institute of Engineering & Technology,
UP Technical University, Lucknow, India
e-mail: profbnmishra@gmail.com

H.R. Arabnia and Q.-N. Tran (eds.), *Software Tools and Algorithms for Biological Systems*, Advances in Experimental Medicine and Biology 696,
DOI 10.1007/978-1-4419-7046-6_22, © Springer Science+Business Media, LLC 2011

1 Introduction

Cytotoxic T cells of the immune system monitor cells for infection by viruses, or intracellular bacteria through scanning their surface for peptides bound to MHC class I molecules [1]. The cells that present peptides derived from non-self, e.g., from viruses or bacteria (after binding to MHC molecule), can trigger a T-cell immune response, which leads to the destruction of such cells [2]. T cells do not recognize soluble native antigen but rather recognize antigens that has been processed into antigenic peptides, which are presented in combination with MHC molecules. It has been observed that peptides of nine amino acid residues (9-mers) bind most strongly; peptides of 8–11 residues also bind but generally with lower affinity than nonamers [3, 4]. Binding of a peptide to a MHC molecule is prerequisite for recognition by T cells and, hence is fundamental to understanding the basis of immunity and also for the development of potential vaccines and design of immunotherapy [5].

The SARS coronavirus, sometimes abbreviated to SARS-CoV, is the virus that causes severe acute respiratory syndrome (SARS). On 12 April 2003, scientists working at the Michael Smith Genome Sciences Centre in Vancouver, British Columbia, finished mapping the genetic sequence of a coronavirus believed to be linked to SARS. Passive immunization with convalescent serum has been tested as a way to treat SARS. Control of SARS is most likely to be achieved by vaccination [6]. In this chapter, various MHC class I alleles for SARS coronavirus (Tor2 Replicase polyprotein 1ab) have been used as a case study.

2 Review and Motivation of Present Work

The algorithms for prediction of MHC-binding peptides are based on two concepts: (1) algorithms based on identifying the patterns in sequences of binding peptides, e.g., binding motif, quantitative matrices, and artificial neural networks, and (2) algorithms based on three-dimensional structures for modeling peptide/MHC interactions [7, 8]. The second approach, i.e., based on structures corresponds to techniques with distinct theoretical lineage and includes the use of homology modeling, docking and 3D-threading techniques.

For prediction of T-cell epitope, ANN has been used with genetic algorithms [9, 10] and evolutionary algorithm [11]. Support vector machine has also been used to predict the binding peptides [12, 13]. For improving prediction of MHC class I binding peptides, probability distribution functions have also been used [14]. Threading methods [15] and Gibbs motif sampler [16] approach have also been used for prediction of MHC-binding peptides. In many cases, the number of known binders and non-binders to specific MHC alleles are limited; therefore, the convergence to optimal weights of ANN has to be improved. In current study, the variable learning rate has been used to improve convergence taking various MHC alleles for SARS corona virus as case study.

3 Methodologies

3.1 Variable Learning Rate for ANN Training

The values of learning rate are taken between 0.0 and 1.0. Back propagation network learns using a method of gradient descent to search for a set of weights that can model the given classification problem, so as to minimize the mean squared distance between the network's class prediction and the actual class label of the samples. The learning rate helps to avoid getting stuck at local minimum in the decision space (i.e., where the weights appear to converge, but are not the optimum solution) and encourages finding the global minimum. If the learning rate is too small, then learning will occur at a very slow pace. If the learning rate is too large, then oscillation between inadequate solutions may occur [17, 18]. In gradient descent, learning rate determines the magnitude of the change to be made in the parameters, i.e., weights and a bias of nodes as per (1) and (2); furthermore, the updated weights and biases are given by (3) and (4). The value of error (Err_j) at output node and at internal node is given by (5) and (6), respectively. The input and output to each j-th unit are given by (7) and (8), respectively. For a given training set of input vectors, the learning rate L is kept fixed which leads to poor convergence in case of a small dataset. To improve the convergence, variable learning rate (i.e., the learning rate is updated after each input vector in a given training set of input vectors) has been used as per (9). The error is calculated using (5) after each training vector, and the learning rate is increased by a value a, if the error on the subsequent training vector decreases. It is decreased geometrically by value $b\eta$, if the error on subsequent training vector increases. The value of ΔL has to be calculated after each input vector in the given training set as per (9), and used to update the learning rate, L. The updated learning rate, $L+\Delta L$, has been used for further training to calculate the values of the weights and biases of the nodes as per (1)–(4).

$$\Delta w_{ij} = (L)\mathrm{Err}_j O_i \tag{1}$$

$$\Delta \theta_j = (L)\mathrm{Err}_j \tag{2}$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \tag{3}$$

$$\theta_j = \theta_j + \Delta \theta_j \tag{4}$$

$$\mathrm{Err}_j = O_j(1-O_j)(T_j - O_j) \tag{5}$$

$$\mathrm{Err}_j = O_j(1-O_j)\sum_k \mathrm{Err}_k w_{jk} \tag{6}$$

$$I_j = \sum_i w_{ij} O_i + O_j \tag{7}$$

$$O_j = \frac{1}{1+e^{-I_j}} \tag{8}$$

$$\Delta L = \left\{ \begin{array}{ll} +a & \text{if } E^{t+\Gamma} \langle E^t \\ -b\eta & \text{otherwise} \end{array} \right\} \tag{9}$$

where Δw_{ij} is the change in the weight w_{ij}, $\Delta\theta_j$ is the change in the bias t_j, W_{ij} is the weight of the connection from unit i to a unit j in the next higher layer, Err_j is the error for unit j at the output layer, Err_j is the error for unit j at hidden layer, I_j is the net input to unit j, O_j is the output of the unit j, T_j is the true output, ΔL is the change in the learning rate, L is the current learning rate, a and b are coefficients E^t is the error at the node in output layer in the previous learning input vector, $E^{t+\Gamma}$ is the error at the node in output layer for current learning input vector.

3.2 Evaluation Parameters

The predictive performance of the model has been evaluated using receiver operating characteristics (ROC) analysis. The area under the ROC curve (AROC) provides a measure of overall prediction accuracy: AROC $< 70\%$ for poor, AROC $> 80\%$ for good, and AROC $> 90\%$ for excellent prediction. The ROC curve is generated by plotting sensitivity (SN) as a function of 1-specificity (SP). The sensitivity, $\mathrm{SN} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$ and $\mathrm{SP} = \mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$, gives percentage of correctly predicted binders and non-binders, respectively. The $\mathrm{PPV} = [(\mathrm{TP})/(\mathrm{TP}+\mathrm{FP})] \times 100$ and $\mathrm{NPV} = [(\mathrm{TN})/(\mathrm{FN}+\mathrm{TN})] \times 100$ give the positive probability value, i.e., the probability that a predicted binder will actually be a binder, and negative probability value, i.e., the probability that a predicted non-binder will actually be a non-binder. Tenfold cross validation has been used for training and prediction. The terms TP, FP, TN, and FN related to threshold T are true positive, false positive, true negative, and false negative, respectively. A web-based tool has been used to calculate the area under the ROC curve available at (www.rad.jhmi.edu/jeng/javarad/roc/JROCFITi.html).

4 SARS Corona Virus: A Case Study

4.1 Data Resources

The datasets used for training and testing of (BNB) have been obtained from IEDB Beta 2.0 database (http://www.immuneepitope.org) for HLA-A*0201, HLA-A*0301, HLA-A*1101, HLA-A*0202, HLA-A*0203, HLA-A*0206, HLA-A*2902, HLA-A*3002 and HLA-B*4002 MHC Class I alleles. The strong binders have been retrieved for IC50 $<$ 500. All 9-mers have been filtered after removing the duplicates. For strong non-binders, the records with IC50 $>$ 5000 have been retrieved. The duplicates from binders and non-binders sets have been removed. Furthermore, to keep the ratio of binders and non-binders nearly 1:1, so as to reduce the biasness in learning, the additional 9-mer non-binders were retrieved through EBI-Expasy protein database available at: http://www.expasy.ch. Final sets of binders and non-binders for various alleles have been shown in Table 1.

Table 1 The number of binders and non-binders for various alleles for SARS coronavirus

	Binders IC < 500			Non-binders IC > 5,000		
MHC Alleles	Retrieved	9-mer(s)	Final	Retrieved	9-mer(s)	Final*
A*0201	136	103	103	0	0	104
A*0301	192	116	116	38	23	117
A*1101	202	128	128	16	8	129
A*0202	132	101	101	1	0	101
A*0203	124	96	96	0	0	96
A*0206	135	102	10	0		102
A*2902	178	100	100	80	68	100
A*3002	136	100	77	127	98	77
B*4002	91	36	36	37	9	36

*Additional 9-mer non-binders added

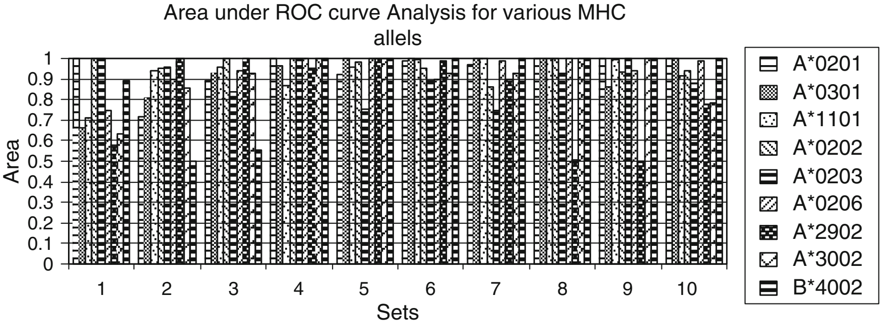

Fig. 1 Analysis of area under ROC curve

5 Results and Discussion

The tenfold cross validation has been used to validate the results. The ANN has been trained ten times for each MHC allele using adaptive learning, each time leaving one of the subsets out of 10, and using the left out subset for prediction. The area under ROC curve has been shown in Fig. 1.

We assembled a dataset of binders and non-binders for various MHC class I alleles to study the impact of the variable learning rate to train the ANN for small datasets. The ten set of binders and non-binders of nearly equal size have been used for tenfold cross validation. The average value of AROC for HLA-A*0201, HLA-A*0301, HLA-A*1101, HLA-A*0202, HLA-A*0203, HLA-A*0206, HLA-A*2902, HLA-A*3002, and HLA-B*4002 MHC Class I alleles is 0.9485, 0.922, 0.9333, 0.9615, 0.8989, 9405, 0.82, 0.906, and 0.8945, respectively, indicating excellent predictions for most of the cases.

The average values for various parameters viz. sensitivity, specificity, accuracy, PPV, and NPV for MHC alleles have been calculated. The average value of sensitivity, i.e., the percent of binders that are correctly predicted as binders, is 91.08.

Higher sensitivity means that almost all of the potential binders will be included in the predicted results. The average specificity, i.e., the percent of correctly predicted as non-binders, is 82.85. The average PPV value is 85.50. It shows that the probability that a predicted binder will actually be a binder is 85.50%. The average NPV is 89.47. It indicates that the probability that a predicted non-binder will actually be a non-binder is 89.47%.

The values used for training the ANN for various MHC alleles are: learning rate L (0.48–0.67), coefficient a (0.3–0.51) and coefficient b (0.0117–0.0725). Area under ROC curve using fixed learning rate for various MHC alleles was found to be 0.5 which indicates very poor prediction.

The modules for the training, classification, and results have been implemented in C using pointers, to improve the efficiency of training and classification through artificial neural network and variable leaning rate.

6 Conclusion

Overall, the study shows that the quality of the prediction of binders and non-binders can be substantially improved using the variable learning rate for artificial neural network training for small datasets. The approach can also be used for various other applications, where the datasets for training are limited. The approach is also useful for the large datasets. The only drawback of the approach is that the value of the parameters is to be adjusted as per the application.

Acknowledgments The authors acknowledge the support of Dr. D. S. Yadav, Assistant Professor in the Department of Computer Science and Engineering, Institute of Engineering and Technology, U P Technical University, Lucknow, and Dr. Firoz Khan, Scientist at CIMAP, Lucknow, India, in reading the entire manuscript and providing the useful suggestions.

References

1. Harriet L Robinson et al. “T cell vaccines for microbial infections”, Nature Medicine, vol. 11, no. 4, pp. S25–S32, 2005
2. Anne S De Groot et al. “Genome-derived vaccines”, Expert Review of Vaccines, vol. 3, no. 1, pp. 59–76, 2004
3. Anne S De Groot et al. “Immuno-informatics: mining genomes for vaccine components”. Immunology and Cell Biology, vol. 80, pp. 255–269, 2002
4. Rino Rappuoli “Reverse vaccinology, a genome based approach to vaccine development”, Vaccine, vol. 19, pp. 2688–2691, 2001
5. Tamas G Szabo et al. “Critical role of glycosylation in determining the length and structure of T-cell epitopes – As suggested by a combined in silico systems biology approach”, Immunome Research, vol. 5, p. 4, 2009
6. Kathryn V Holmes “SARS coronavirus: a new challenge for prevention and therapy”, Journal of Clinical Investigations, vol. 111, pp. 1605–1609, 2003
7. Joo Chuan Tong et al. “Methods and protocols for prediction of immuno-genic epitopes”, Briefings in Bioinformatics, vol. 8, no. 2, pp. 96–108, 2006

8. Bing Zhao et al. "MHC-binding peptide prediction for epitope based vaccine design", International Journal of Integrative Biology, vol. 1, no. 2, pp. 127–140, 2007
9. Giuliano Armano et al. "A hybrid genetic-neural system for predicting protein secondary structure", BMC Bioinformatics, vol. 6, no. 4, S3, 2005
10. Yeon-Jin Cho1 et al. "Prediction rule generation of MHC class I binding peptides using ANN and GA", L. ICNC 2005, LNCS 3610, pp. 1009–1016, 2005
11. V Brusic, G Rudy, M Honeyman J Hammer, L Harrison "Prediction of MHC class II-binding peptides using an evolutionary algorithm and artificial neural network", Bioinformatics, vol. 14, no. 2, pp. 121–130, 1998
12. Henning Riedesel et al. "Peptide binding at class I major histo-compatibility complex scored with linear functions and support vector machines", Genome Informatics, vol. 15, no. 1, pp. 198–212, 2004
13. M Bhasin, G P S Raghava "A hybrid approach for predicting promiscuous MHC class I restricted T cell epitopes", Journal of Bioscience, vol. 32, no. 1, pp. 31–42, 2007
14. S S Soam et al. "Prediction of MHC class I binding peptides using probability distribution functions", Bioinformation, vol. 3, no. 9, pp. 403–408, 2009
15. S P Singh et al. "Evaluation of threading based method for prediction of peptides binding to MHC class I alleles", International Journal of Integrative Biology, vol. 4, no. 1, pp. 16–20, 2008
16. S P Singh et al. "Prediction of MHC binding peptides using Gibbs motif sampler, weight matrix and artificial neural network", Bioinformation vol. 3, no. 4, pp. 150–155, 2008
17. T Tollenaere "SuperSAB: fast adaptive back-propagation with good scaling properties", Neural Networks, vol. 3, no. 5, pp. 561–573, 1990
18. Enrique Castillo et al. "A very fast learning method for neural networks based on sensitivity analysis", Journal of Machine Learning Research, vol. 7 pp. 1159–1182, 2006