American Journal of Bioinformatics Research 2012, 2(2): 1-10
DOI: 10.5923/j.bioinformatics.20120202.01

Predicting MHC Class II Epitopes Using Separated Constructive Clustering Ensemble

Hossam Fathy ElSemellawy[1,*], Amr Badr[2], Mostafa Abdel Aziem[3]

[1]Information System Department, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt
[2]Computer Science Department, Faculty of Computers and Information (Cairo University), Giza, Egypt
[3]Computer Science Department, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt

Abstract Successful Prediction for MHC Class II epitopes is an essential step in designing Genetic Vaccines[1]. MHC Class II epitopes are short peptides with length between 9 and 25 amino acids which are bound by MHC. These epitopes are recognized by T-Cell Receptors and leads to activation of cellular and humoral immune system and, ultimately, to effective destruction of pathogenic organism. Successful prediction of MHC class II epitopes is more difficult than MHC class I epitopes due to open binding groove at both ends in class II molecules, this structure leads to variable length for MHC II epitopes and complicating the task for detecting the core binding 9-mer. Large efforts have been exerted in developing algorithms to predict which peptides will bind to a given MHC class II molecules. In this paper we presented a novel classification algorithm for predicting MHC Class II epitopes using Multiple Instance Learning technique. Separated Constructive Clustering Ensemble (SCCE) is our new version for Constructive Clustering Ensemble (CCE)[27]. This method was used for converting multiple instance learning problem into normal Single Instance Problem. Most of the processing in this method lies mainly in vector preparation step before using classifier; Support Vector Machine (SVM) has been used as a method with proven performance in a wide range of practical applications[38]. SCCE integrated many algorithms like Genetic Algorithm, K medoid clustering, Ensemble learning and Support vector machine in an orchestration to predict the MHC II epitopes. SCCE was tested over three benchmark data sets and proved to be very competitive with the state of art regression methods. SCCE achieved these results using only binder and non binder flags; without need for regression data. An implementation of MHCII-SCCE as an online web server for predicting MHC-II Epitopes is freely accessible at http://www.fci.cu.edu.eg:8080/MHCII_Server/MHCII SCCE Server 1.0.htm.

Keywords Major Histocompatibility Complex (MHC), Multiple Instance Learning (MIL), Genetic Algorithm (GA), Support Vector Machine (SVM)

1. Introduction

Epitopes or antigenic peptides are set of amino acids from the pathogenic organism DNA which bound with MHC to be presented by Antigen Presenting Cells (APCs)[2] for inspection by T cells. Humans' MHC is often called Human Leukocyte Antigen (HLA). The binding and presentation operations are considered the central recognition process occurring in the immune system as without them the immune system would be almost ineffective. These operations lead to activation of the T cell, which then signals to the wider immune system that a pathogen has been encountered.

The proteins of the MHC are grouped into two classes[3]. Class I molecules present endogenous peptides, Class II molecules generally present exogenous peptides. MHC class I ligands are of 8 to 11 amino acids long while MHC class II ligands are of 9 to 25 amino acids. Class I molecules have a binding cleft which is closed at both ends while MHC class II molecules have groove which is opened at both ends this allows much larger peptides to bind and also allows the bound peptide to have significant "dangling ends", thus the prediction of which peptides will bind a specific MHC class II complex constitutes an important step in identifying potential T cell epitopes. These epitopes are suitable as vaccine candidate. Figure 1 displays MHC II tertiary structure bounded with MHC II epitope.

The currently available methods for identifying MHC II binding peptides are split into main categories:

(1) Qualitative methods: These methods try to identify binder and non binder peptides despite its binding affinity, like methods use a position weight matrix to model ungapped multiple sequence alignment of MHC binding peptides[4-7], other methods use Artificial Neural Networks (ANN)[10,11] and support vector machines (SVM) like[12,13].

(2) Quantitative methods: these methods try to predict binding affinity for peptide like PLS-ISC[15], MHCPred[16],

* Corresponding author:
Hossam.ElSemellawy@gmail.com (Hossam Fathy ElSemellawy)
Published online at http://journal.sapub.org/bioinformatics

SVRMHC[17], ARB[18], NetMHCII[19], MHCMIR[20] and NN-align[14].

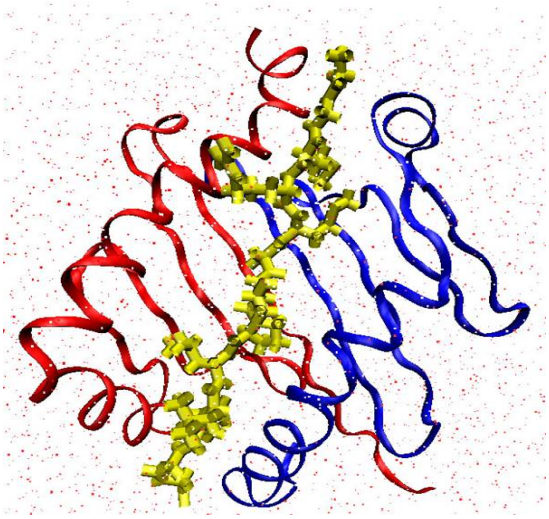

Figure 1. Example of tertiary structure of peptide binding to MHC class II. It can be seen that the binding groove is open in the ends in contrast to MHC class I

Several prediction methods depend on detecting the binding 9-mer core peptide like Gibbs sampling[22], RankPEP[5], Gibbs[7], NetMHCII[4], MHCMIR[20] and NN-align[14]. Chang et al.[23] showed the value of using potentially useful predictive information that may be available outside the 9-mer core peptide by incorporating peptide length as one of the inputs to improve the performance of the predictor in compare with one that uses only the features derived from the 9-mer core. Nielsen et al.[4] have demonstrated that including peptide flanking residues among inputs improves the performance of their SMM-align method on 11 out of 14 MHC-II allele specific data sets.

EL-Manzalawy et al.[20] has introduced using Multiple Instance regression (MHCMIR) by representing flexible length peptides by a bag of 9-mer subsequences. The label associated with each bag could be either binary label (binder – not binder) or continuous binding affinity of the peptide. This method like[14] does not require the 9-mer cores in each binding peptide to be identified prior to training the predictor. The learning algorithm also detected the 9-mers core peptides.

EL-Manzalawy et al.[20] adapted MILES (multiple instance learning via embedded selection)[24] to fit with the MHC II epitopes prediction by using BLOSUM62 matrix instead of Euclidean distance for calculating distance between peptides; and replacing the 1-norm SVM classifier by a support vector regression (SVR) model[26]. According to the evaluation results, both Multiple Instance Learning and Multiple Instance regression are provided to be promising in predicting MHC II binding affinity.

Both[14, 21] have used ensemble learning whether between different MHC II classifiers like[21] or between classifiers use the same method but with different parameters like[14]. NN-align[14] made ensemble for number of Neural Network classifiers with different number of nodes in the hidden layer. Both[14, 21] succeeded in achieving much better results than using single classifier.

In this paper we used multiple instance learning for predicting the MHC II binding epitopes. We followed the same methodology like[20] for representing MHC II flexible length peptides by a bag of 9-mer subsequences and assign a binary label indicating whether the corresponding peptide is an MHC-II binder or not.

The main contributions in this paper to the current state-of-the-art methods are the following:

1. New enhanced version for Constructive Clustering Ensemble method[27] has been developed for resolving the multi-instance learning problem, the new enhancement has been proved to achieve better results than the original algorithm. Constructive clustering ensemble method focused mainly on converting flexible length bags into a fixed vector to be able to convert multi-instance learning problem to a normal classification problem.

2. Replacing K-means clustering algorithm used in the constructive clustering[27] with k-medoid algorithm to fit with MHC II epitopes prediction besides changing distance function.

3. Using Genetic algorithm for selecting ensemble classifiers parameters. All parameters are selected using training data only and tested with unseen test data.

We showed how enhanced version for constructive clustering ensemble has a competitive performance with state of art methods.

In this paper, we first defined all used benchmark datasets, then how to calculate the performance, introduction to constructive clustering ensemble and enhancements added to this method, calculating performance comparison between original CCE method and enhanced version SCCE. Genetic algorithms were used to specify ensemble learning parameters; finally, performance comparison and conclusion.

2. Data

The first one is currently the largest data set which published by Wang et al.[39] for human HLA DR, DP and DQ molecules binding affinities. The data set comprises 26 HLA-DR, DP and DQ alleles. Alleles included in this data set were selected for their high frequency in the human population so it reached to 99% from the human coverage.

Peptides with measured IC50 < 1000 nM are considered binders and others are non binders. Wang et al. also partitioned the data into five folds used for training and testing at the following URL http://tools.immuneepitope.org/analyze/html/download_MHC_II.html. Wang et al.[39] published comparison between set of methods by cross validation using this five folds data set. In this paper we compared the results published in Wang et al.[39] with results collected from SCCE using the same folds.

The second one is IEDB HLA-DR restricted peptide-binding data set which published by Nielsen et al.[28]. As SCCE depends on qualitative input so we converted this

data set from quantitative to qualitative, The peptide binding affinity IC50 has been converted to either 1 or 0 by converting all peptides with binding affinity greater than 0.426 to binder (1) and all peptides less than or equal to 0.426 to non binder. The data set comprises 13 HLA-DR alleles each characterized by at least 420 and up to 5166 peptide binding data points, only one allele data has been removed as we retrieved only allele data which have more than 100 instances in both binder and non binder class. NN-align[14] published the five folds used for training and testing on http://www.cbs.dtu.dk/suppl/immunology/NetMHCII-2.0.php. NN-align[14] also published results on this data set for set of methods using five folds cross validation. We compared SCCE using five folded cross validation with the methods presented in NN-align[14].

The third data set is El-Manzalawy benchmark data set[29]. El-Manzalawy et al. has introduced four degrees of similarity reduction for the data sets extracted from MHCPEP[40] and MHCBN[41] besides the original UPDS data set extracted from the source. El-Manzalawy et al.[29] published the results for three methods using this data set by five folds cross validation.

The first degree is MHCPEP-SRDS1 and MHCBN-SRDS1 which derived from the corresponding UPDS datasets. The new data set don't have two peptides share a 9-mer subsequence.

The second degree is MHCPEP-SRDS2 and MHCBN-SRDS2 which derived from the corresponding SRDS1 datasets. The new data set ensured that the sequence identity between any pair of peptides is less than 80%.

The third degree is MHCPEP-SRDS3 and MHCBN-SRDS3 which derived from the corresponding UPDS datasets using similarity reduction introduced by Raghava[42].

The forth degree is MHCPEP-WUPDS and MHCBN-WUPDS which derived from the corresponding UPDS datasets, the new data set assigned weight to a peptide, this weight is inversely proportional to the number of peptides that are similar to it.

The forth database is The Wang et al.[21] benchmark data set; consists of quantitative and qualitative binding data to 14 HLA-DR alleles plus IAB data set. We used this data set for comparing between the original version for CCE and our enhanced version SCCE using 10 folds cross validation.

For all data sets evaluation, training data was used as an input for genetic search to find the best parameters for the training data then built ensemble classifier using these parameters and test the classifier against unseen test data.

3. Methods

3.1. Multi-Instance Learning Using Constructive Clustering Ensemble (CCE)

The term multi-instance learning was defined by Dietterich et al.[33] when they were investigating the problem of drug activity prediction. The activity prediction objective is to predict whether a candidate drug molecule will bind strongly to a target protein or not. Not all drug molecules can bind well to all proteins. Drug molecule shape is the most important factor in determining whether a drug molecule will bind the target protein or not. However, drug molecules are flexible, so they can adopt a wide range of shapes. A binder molecule can take many shapes but at least one of them can bind to the target protein while the non binder molecule does not adapt to any shape that can bind to the protein.

Multiple instance learning formulates this problem[30] by representing each candidate molecule by a bag of instances, each instance in the bag representing a unique shape adapted by the molecule. The bag is positive if and only if at least one of the instances in the bag can bind to the protein and negative if none of the instances in the bag can bind to the protein. Not like supervised learning[27] where all training instances have known labels, in multi-instance learning the labels of the training instances are unknown; and in contrast to unsupervised learning where all training instances are without known labels, in multi-instance learning the labels of the training bags are known. In multi-instance learning each instance in the bag has its own features vectors while the label is for the whole bag not for each instance.

[30, 31, 32, 33] proposed a solution to the MIL problem by adapting single supervised learning algorithms to multi-instance learning as long as their focuses are shifted from the discrimination between instances to the discrimination between bags. Constructive Clustering based Ensemble (CCE) method[27] for resolving the multi-instance learning problems takes opposite direction. CCE adapted the multi-instance representation to fit with the single supervised learning algorithms, so each bag is represented in a single features vector instead of vector for each instance and this vector takes bag label.

For building one vector for the whole bag instances, CCE collected all instances from all bags despite its label and placed them all in a one list, then cluster these instances into d groups. CCE represented each bag by a vector of d features (one for each cluster), the vector values are either 1 or 0; 1 if there is an instance in the bag related to this cluster and 0 if not. Now each bag is represented by only one vector containing d-dimensional binary feature vector so we could use normal single instance supervised classifiers to distinguish the bags. CCE proposed using support vector machines for classification and K-means for clustering.

As there is no method for specifying the best number of clusters, CCE[27] created many classifiers using different cluster count and combined their prediction so the method utilized the power of ensemble learning to achieve strong generalization capability[34].

CCE doesn't need to store any of the training data as only clusters centroids are stored to be used for building training and testing vectors. Calculating distance between only clusters centroids and test bags instances minimizes the number of comparisons as we compare with a limited number of points not with the whole training data like Citation-kNN and Bayesian-kNN[35].

CCE results were very competitive[27] whether in the

MUSK data sets[33] or Generalized MI Data Sets[36] but there are some major problems in the CCE which are the following:

1. There is no methodology for selecting the best number of clusters or even upper and lower boundaries.
2. In case that there are shared instances between positive and negative bags which is a common case in MIL, some of the resultant clusters will be shared between positive and negative bags. These clusters will not help in distinguishing between positive and negative bags as its features have value 1 in both cases.
3. Depending on binary features vector prevented any variation in the distance between bag instances and clusters centroids.

SCCE method for predicting MHC II epitopes addressed all these issues plus adapting it to fit with MHC II epitopes prediction.

Comparisons between SCCE and original version for CCE are displayed to show the effect of these enhancements.

3.2. Separated Constructive Clustering Ensemble Method (SCCE)

SCCE is an enhanced version for CCE to solve multi-instance learning problem. SCCE converted bag of Multiple Instances vectors into a single vector and uses Support vector Machines (SVM) to distinguish between binder and non binder bag.

First each peptide was represented by bag of 9-mer subsequences (Figure 2), then assigned a binary label whether 1 for binder bags and 0 for non binder bags. These 9-mers in each bag instances represent a candidate binding core; if the bag has at least one of these binding cores, then it is a binder bag.

DLQDRTAQDKSVVNMQQRY	DLQDRTAQD	1
	LQDRTAQDK	
	QDRTAQDKS	
	DRTAQDKSV	
	RTAQDKSVV	
	TAQDKSVVN	
	AQDKSVVNM	
	QDKSVVNMQ	
	DKSVVNMQQ	
	KSVVNMQQR	
	SVVNMQQRY	

Figure 2. representing MHC II binder peptide into bag of 9-mers subs strings, each instance is a candidate to be core 9-mers[20]

According to Nielsen et al.[4], including peptide flanking residues PFR among inputs improves their SMM-align method performance on 11 out of 14 MHC-II allele specific data sets; EL-Manzalawy et al.[20] tried to apply this finding by representing each peptide as a bag of 10, 11, or 12-mers extracted from it. According to their findings we could deduce that the binding core may not exist at the beginning of the peptide and may be delayed for one or two positions as the first position would be reserved for the PFR. As a result, one of the parameters for our method is the starting position for extracting the 9-mers bag instances from the peptide.

Chang et al.[23] showed the value of using potentially useful predictive information that may be available outside the 9-mer, SCCE tried to apply this finding by representing the whole bag into one vector so all useful predictive information appears in the resultant vector.

In SCCE; first we separated between instances from binder bags and non binder bags; and removed shared 9-mers between them then applied clustering on each list separately. By this way, two unrelated lists of the clusters were created for each side. The resultant clusters centroids are combined in one list and used to build bags vectors by representing each bag by a vector of d features (one for each cluster). Final vector values are the shortest distance between each cluster centroid and the instances in the bag. Our vector has continuous values instead of the binary values used in CCE.

For clustering set of strings, k-means can't be used as it depends on building clusters centroids using the mean value for each attribute in the cluster members; to overcome this problem k-medoids algorithm has been used[37]. The main positive point in k-medoids over k-means is its robustness to noise and outliers as it minimizes the sum of dissimilarities instead of a sum of square Euclidean distances, the medoid can be defined as the instance of cluster whose average dissimilarity to all instances in the cluster is minimal, i.e. it is the most centrally located point in the cluster. K-medoids steps are the following:

1. Initialize: randomly select d instances from the n instances as a medoids.
2. Associate each instance to the closest medoid (selecting the closest medoid is done by collecting the distance using distance function which is based on the BLOSUM62 amino acid substitution matrix[25]).
3. Using BLOSUM62 for distance calculation gives us larger distance for nearly similar 9-mers and lower values or negative values for non similar 9-mers; so all distances will be multiplier by -1 to have a smaller values for similar 9-mers and vice versa.
4. For each cluster try to swap the medoid point with non medoid point and computer the cost for this swap, select instance that have the lowest cost as the new medoid.
5. Repeat steps 2, 3 until there is no change in the medoids.

After complete running K-medoid clustering algorithm, two lists of clusters are generated. These clusters medoids contain the most important patterns in both negative and positive bags.

New parameter for minimum cluster size has been added to remove the noise clusters or clusters containing non core 9-mers. All final clusters are added in one list and used to build bags vectors by representing each bag by a vector of d features (one for each cluster). Bags vector values are the shortest distance (using BLUSOM62[25]) between cluster centroid and the instances in the bag.

After building all bags vectors, supervised single classifier (Support Vector Machine) was trained using training data. When unseen peptide presented to SCCE classifier, it is converted into a bag of 9-mers substrings and use clusters centroids generated during training phase to build unseen

peptide vector then send this vector to SCCE classifier to retrieve the result whether binder or not. Algorithm 1 showed the pseudo-code for the SCCE.

Algorithm 1 Training SCCE

```
Function BuildSCCEClassifier (parameters: PosBags, NegBags, PosClustersCount, NegClustersCount, PosClustersMinSize, NegClustersMinSize)
Begin
  set PosInstances to null            #Positive Instances
  set NegInstances to null            #Negative Instances
  set PosClusters to null             #Positive Clusters
  set NegClusters to null             #Negative Clusters
  For integer i loops 1 to size(PosBags) stepping 1
    Foreach instance x Є PosBags_i do
        PosInstances := PosInstances U {x}
 #copy instances from positive bags into PosInstances list
    End
  For integer i loops 1 to size(NegBags) stepping 1
    Foreach instance Є NegBags_i do
        NegInstances := NegInstances U {x}
 #copy instances from negative bags into NegInstances list
    End
  Call RemoveSharedInstances(PosInstances,NegInstances)
  PosClusters := Cluster (PosInstances, PosClustersCount)
  # cluster PosInstances into PosClustersCount groups.
  Call RemoveSmallClusters(PosClusters, PosClustersMinSize)
  NegClusters := Cluster (NegInstances, NegClustersCount)
  # cluster NegInstances into NegClustersCount groups.
  Call RemoveSmallClusters(NegClusters, NegClustersMinSize)
  AllClustersList := PosClusters U NegClusters
  d := Size(AllClustersList)
  Set FinalVector to null
  For integer i loops 1 to size(PosBags) stepping 1
  Begin
    For integer k loops 1 to d stepping 1
    Begin
    Yk = arg min dist (AllClustersListk, Pi).
#we got the minimum distance between the cluster medoid and the Bag Instances and add it in the vector.
    End
    S := S U {Y1, Y2….Yk}
  End
  For integer i loops 1 to size(NegBags) stepping 1
  Begin
    For integer k loops 1 to d stepping 1
    Begin
    Yk = arg min dist (AllClustersListk, Ni).
#we got the minimum distance between the cluster medoid and the Bag Instances and add it in the vector.
    End
    S := S U {Y1, Y2….Yk}
  End
  Classifier := TrainSVM(S)
  Return Classifier
  End
Function dist (Parameters: s1, s2)
Begin
    For integer i loops 1 to 9 stepping 1
    Begin
    d+ = BLOSUM62(s1[i]; s2[i])
    End
    d = d * -1
    Return d
End
```

Table 1 displayed performance comparison on Wang et al.[21] benchmark data set by calculating Area Under Curve (AUC) for the original CCE after adapting it to fit with MHC II epitopes prediction and different versions for SCCE. The first Column is the normal CCE; in this case all instances from both positive and negative bags are added in one list and clustered into 160 clusters, with minimum cluster size 10. For the other columns, positive and negative instances are separated then applied K-medoid clustering using three different configurations, first one; 80 clusters are created for positive instances only with minimum cluster size 10, the second, 80 clusters are created also but for negative instances only with minimum cluster size 10, the third one which achieved the best results; 80 clusters are generated for positive instances and another 80 clusters for negative instances with minimum cluster size for both cases 10 Instances.

After comparing the results between original CCE and these different configurations for the algorithm, creating clusters for positive instances and negative instances separately has the best average results over any other method; as it facilities building clusters that purely represent positive or negative instances. Once new peptide presents to classifier; if it is related to positive bags it will be more close to clusters created from instances from positive bags and vice versa. Although original CCE has the second top rank between the four methods but if we look at alleles results one by one; we will find that original CCE has the best result in only one allele from 14 alleles; while the other methods have the best results in the other 13 alleles.

Selecting parameters for clusters count and minimum cluster size will be the most important part for building SCCE classifier, in some cases build clusters for positive instances only gives the best results while in another cases cluster negative instances only gives the best results and also for balancing between negative and positive instances.

Another Important point is using ensemble learning; original CCE made ensemble between set of classifiers but after changing clusters count so the vector for each classifier will be different, CCE didn't specify how to select suitable number of clusters. CCE also didn't build the ensemble classifier according to training data. According to table 2; the number of clusters for instances can't be the same for all alleles as they vary in the features of each case.

Table 1. Comparison between the CCE original method and different configurations for SCCE method.

	No Separation	Cluster Only Positive Instances	Cluster Only Negative Instances	Cluster separated Positive and negative Instances
DRB1-0101	0.81	0.81	0.785	0.811
DRB1-0301	0.72	0.741	0.647	0.72
DRB1-0401	0.64	0.69	0.534	0.682
DRB1-0404	0.77	0.76	0.515	0.783
DRB1-0405	0.705	0.657	0.491	0.746
DRB1-0701	0.758	0.805	0.495	0.79
DRB1-0802	0.648	0.47	0.506	0.72
DRB1-0901	0.633	0.627	0.517	0.653
DRB1-1101	0.797	0.813	0.587	0.793
DRB1-1302	0.677	0.522	0.604	0.69
DRB1-1501	0.743	0.751	0.617	0.75
DRB3-0101	0.706	0.537	0.727	0.69
DRB4-0101	0.773	0.673	0.502	0.801
DRB5-0101	0.813	0.807	0.583	0.79
Average	0.728	0.69	0.58	0.744

3.3. Genetic Algorithm for Selecting Parameters

According to previous section we need to have a methodology for specifying the parameters. The number of parameters are duplicated many times according to classifiers used in the ensemble, selecting ensemble parameters can't be the same for all alleles specially that they vary in the number of binders and non binders, binding cores,..

SCCE depends on GA for selecting ensemble parameters. First, training data were split into training and validation data set. Genetic Algorithm runs using training data and after each generation the best chromosome was tested with validation data set to make sure that the enhancements achieved in recognizing training data didn't lead to over fitting. Once the results are going down; generations creation terminates and returns the best chromosome. Finally, we train the classifier using the whole training data with the final parameters.

Genetic Algorithm chromosome contains all ensemble parameters; gene for each parameter, so for each classifier in the ensemble there is a gene for positive clusters count, negative clusters count, minimum size for positive cluster, minimum size for negative cluster and the starting position in the peptide to extract the 9-mers bag instances. GA fitness function is 3 fold cross validation result. Genetic algorithm run for specifying the parameters for 20 ensemble classifiers, we started GA with initial population of 20 randomly generated chromosomes; each chromosome has the parameters for 20 ensemble classifiers. After completing all steps for generating new generation from the initial population like selection, cross over and mutation; the best chromosome was tested using validation data; if there is an enhancements in the results then GA Search continues in generating new generation, if the result is going down on validation data set or GA search reached to maximum number of generations, GA terminates and return best chromosome.

4. Results

4.1. The Wang et al. data set[39]

SCCE was evaluated using Wang et al.[39] data set using five folds cross validation. The data set was split into five folds and used to compare the results between several MHC Class II epitopes prediction methods. Wang et al.[39] published the results using this data set for five methods using 5-fold cross validation.

SCCE classifier was created by ensemble 15 classifiers; training data has been split into training and validation. Training data was used for selecting classifiers parameters, and validation data set has been used to make sure that the new GA generations are not going to over fitting. After completing GA iterations; all training data was used for building the ensemble classifiers and tested using testing data.

From the results in Table 2; SCCE is ranked number 2 after NN-align but SCCE has an advantage over NN-align which is its ability to work without need for binding affinity information. SCCE was mainly designed to work with classification data where only binary labels (binder or not binder) are available. Another advantage in SCCE is the ability to adapt it in the future to use any high generalization capability classifier rather than SVM, Currently SVM is one of the best classification methods but we can switch to any other new classifier if it proves better generalization capability.

4.2. IEDB Benchmark Data Set

Quantitative IEDB benchmark data set from Nielsen et al.[28] has been converted to qualitative using a threshold of 500 nM. This means that peptides with log50k transformed binding affinity values greater than 0.426 are classified as binders and peptides with binding affinity values less than or equal 0.426 as classifier as non binders. The data has been split into five folds; SCCE was evaluated using five folds cross validation.

Table 3 displayed the results for SCCE using five folds cross validation with IEDB data set[28]. The first three methods results were collected from NN-align[14] using five cross validation. NN-align[14] published the five folds used for training and testing on http://www.cbs.dtu.dk/suppl/immunology/NetMHCII-2.0.php.

SCCE is very competitive with the state of art methods.

Both SMM and NN-align collected its results after training using regression data while SCCE and TEPITOPE depend only on qualitative data. SCCE achieved big difference than TEPITOPE and competitive performance with SMM and NN-align without need for quantitative training data.

4.3. El-Manzalawy Benchmark Data Set

El-Manzalawy et al.[29] introduced new benchmark data set by using four different similarity reduction methods to create four data sets beside the original data. El-Manzalawy et al. proved that the performance for MHC II prediction method depends mainly on inherent peptide similarity in the training data and will be affected according to the similarity reduction level.

SCCE performance has been compared with MHC class II epitopes prediction methods (5-spectrum, LA, and CTD) included in the El-Manzalawy et al.[29] using five folds cross validation. This comparison enabled us measure to what extend the result will be affected by similarity reduction.

Table 2. comparison between set of known MHC II epitopes prediction methods and SCCE using five folds cross validation on IEDB HLA-DR, DP and DQ data set published by Wang et al.[39]

Allele	ARB	SMM-align	PROPRED	combinatorial library	NN-align	SCCE
HLA-DPA1*0103-DPB1*0201	0.832	0.921		0.84	0.943	0.93
HLA-DPA1*01-DPB1*0401	0.847	0.93		0.833	0.947	0.94
HLA-DPA1*0201-DPB1*0101	0.824	0.909		0.849	0.944	0.94
HLA-DPA1*0201-DPB1*0501	0.859	0.923		0.867	0.956	0.94
HLA-DPA1*0301-DPB1*0402	0.821	0.932		0.864	0.949	0.94
HLA-DQA1*0101-DQB1*0501	0.871	0.93		0.809	0.945	0.92
HLA-DQA1*0401-DQB1*0402	0.845	0.896		0.681	0.922	0.891
HLA-DQA1*0501-DQB1*0201	0.855	0.901		0.586	0.932	0.91
HLA-DQA1*0501-DQB1*0301	0.844	0.91		0.802	0.927	0.91
HLA-DRB1*0301	0.753	0.852	0.699		0.887	0.84
HLA-DRB1*0401	0.731	0.781	0.737		0.813	0.79
HLA-DRB1*0404	0.707	0.816	0.769		0.823	0.803
HLA-DRB1*0405	0.771	0.822	0.767		0.87	0.841
HLA-DRB1*0701	0.767	0.834	0.773	0.762	0.869	0.85
HLA-DRB1*0802	0.702	0.741	0.647		0.796	0.772
HLA-DRB1*0901	0.747	0.765		0.572	0.81	0.8
HLA-DRB1*1101	0.8	0.864	0.804		0.9	0.87
HLA-DRB1*1302	0.727	0.797	0.6		0.814	0.783
HLA-DRB1*1501	0.763	0.796	0.743		0.852	0.831
HLA-DRB3*0101	0.709	0.819		0.655	0.856	0.8
HLA-DRB4*0101	0.785	0.816		0.697	0.87	0.84
HLA-DRB5*0101	0.76	0.832	0.728		0.886	0.83
H-2-IAb	0.8	0.855			0.858	0.85
Avarage	0.7858	0.853	0.7267	0.752	0.885	0.861
Min	0.702	0.741	0.6	0.572	0.796	0.772
Max	0.871	0.932	0.804	0.867	0.956	0.94

Table 3. comparison between set of known MHC II epitopes prediction and SCCE on IEDB HLA-DR data set published by Nielsen et al.[28].

Allele	TEPITOPE	SMM	NN-align	SCCE
DRB1*0101	0.72	0.802	0.837	0.83
DRB1*0301	0.664	0.795	0.808	0.79
DRB1*0401	0.716	0.75	0.767	0.75
DRB1*0404	0.77	0.8	0.815	0.8
DRB1*0405	0.759	0.789	0.771	0.742
DRB1*0701	0.761	0.812	0.844	0.82
DRB1*0802	0.766	0.787	0.826	0.782
DRB1*0901		0.655	0.623	0.67
DRB1*1101	0.721	0.796	0.822	0.803
DRB1*1302	0.652	0.785	0.822	0.77
DRB1*1501	0.686	0.727	0.754	0.73
DRB4*0101		0.793	0.811	0.77
DRB5*0101	0.68	0.761	0.789	0.77
Avg	0.72	0.77	0.79	0.77

Table 4. Comparison between set of MHC II epitopes prediction methods and SCCE using El-Manzalawy et al.[29] benchmark data set.

Data Set Name	Similarity Reduction Method	5-spectrum	LA	CTD	SCCE
MHCPEP[40]	UPDS	0.885	0.886	0.9	0.91
	SRDS1	0.702	0.789	0.77	0.81
	SRDS2	0.575	0.709	0.659	0.73
	SRDS3	0.684	0.751	0.702	0.76
	WUPDS	0.722	0.741	0.719	0.804
MHCBN[41]	UPDS	0.758	0.797	0.775	0.84
	SRDS1	0.451	0.697	0.719	0.753
	SRDS2	0.36	0.636	0.673	0.71
	SRDS3	0.68	0.756	0.703	0.77
	WUPDS	0.609	0.718	0.694	0.763

Results show clearly that SCCE outperformed all methods besides its robustness in case of low similarity between the data.

5. Discussion

Predicting MHC Class II binder epitopes is an essential step in developing Genetic Vaccines[1], MHC II epitopes predictions is much more complicated than predicting MHC I binders epitopes according to the open binding groove at both ends. MHC II structure leads to variable length epitopes and complicating the task for detecting the core binding 9-mer.

SCCE has been inspired from both[20] and[27]. EL-Manzalawy et al.[20] has introduced converting MHC class II binding epitopes prediction into multiple instance learning; Construct Clustering Ensemble (CCE)[27] converted Multiple Instance Learning problem where each instance in the bag has its own feature vector into normal single classifier where each bag has one feature vector representing all bag instances and assign the bag label to this new feature vector. Combining both[20] and[27] enabled us to get benefit from the knowledge outside binding core and build one vector for all bag instances. SCCE has adapted CCE to MHC Class II epitopes prediction by using K-medoid clustering algorithm and BLOSUM62 amino acid substitution matrix for distance calculation.

SCCE introduced separating between instances from positive and negative bags and remove shared instances between each other; then create clusters for each group separately. By this way the new clusters represent unique 9-mer in positive instances or negative instances separately and neglect representing shared 9-mers, the resultant clusters are used to build single vector for the whole bag instances. Ensemble learning has been generated between many classifiers; each one has different clusters count; by this way we used different ways for representing features and better generalization capability.

SCCE also introduced using Genetic Algorithm for specifying ensemble classifiers parameters, the parameters include positive and negative clusters count and minimum cluster size. GA iterations terminate when reach to the maximum number of iterations or validation data set results is going down.

SCCE used Support Vector Machine (SVM) for classification as it is currently one of the most robust methods[38] and has a very good generalization capability.

6. Conclusions

In this paper new classification method for predicting MHC Class II epitopes was presented, this method has been tested against three main benchmark data sets and proved to be competitive with the state of art methods, SCCE is a very flexible method that can change its input vector size automatically according to patterns in training data. GA enabled SCCE to represent all feature in the training data by selecting the best number for positive and negative clusters count plus selecting the minimum cluster size. SCCE can be developed in the future to use any classifier rather than SVM or even make ensemble between SVM and any other classifiers.

ACKNOWLEDGEMENTS

We are very Grateful to Dr Yasser EL-Manzalawy (from IOWA State University) for his valuable support during our work.

REFERENCES

[1] C. Janeway, P. Travers et al, Immunobiology: The Immune System in Health and Disease, 6th ed. Garland Pub, 2004.

[2] Castellino F, Zhong G, N GR: Antigen presentation by MHC class II molecules: invariant chain function, protein trafficking, and the molecular basis of diverse determinant capture. Hum Immunol 1997, 54:159-169.

[3] Yewdell JW, Bennink JR: Immunodominance in major histocompatibility complex class I-restricted T lymphocyte responses. Annual review of immunology 1999, 17:51-88.

[4] M. Nielsen, C. Lundegaard, and O. Lund, Prediction of MHC class II binding affinity using SMM-align, a novel stabiliza-

tion matrix alignment method. BMC Bioinformatics, vol. 8, p. 238, 2007.

[5] P. Reche, J. Glutting, H. Zhang, and E. Reinherz, Enhancement to the RANKPEP resource for the prediction of peptide binding to MHC molecules using profiles. Immunogenetics, vol. 56, no. 6, pp. 405–419, 2004.

[6] H. Singh and G. Raghava, ProPred: prediction of HLA-DR binding sites. Bioinformatics, vol. 17, no. 12, pp. 1236–1237, 2001.

[7] M. Nielsen, C. Lundegaard, P. Worning, C. Sylvester-Hvid, K. Lamberth, S. Buus, S. Brunak, and O. Lund, Improved prediction of MHC class I and II epitopes using a novel Gibbs sampling approach. Bioinformatics, vol. 20, pp. 1388–97, 2004.

[8] M. Rajapakse, B. Schmidt, L. Feng, and V. Brusic, Predicting peptides binding to MHC class II molecules using multi-objective evolutionary algorithms. BMC Bioinformatics, vol. 8, no. 1, p. 459, 2007.

[9] H. Mamitsuka, Predicting peptides that bind to MHC molecules using supervised learning of Hidden Markov Models. PROTEINS: Structure, Function, and Genetics, vol. 33, pp. 460–474, 1998.

[10] M. Nielsen, C. Lundegaard, P. Worning, S. Lauemøller, K. Lamberth, S. Buus, S. Brunak, and O. Lund, Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Science, vol. 12, pp. 1007–1017, 2003.

[11] S. Buus, S. Lauemoller, P. Worning, C. Kesmir, T. Frimurer, S. Corbet, A. Fomsgaard, J. Hilden, A. Holm, and S. Brunak, Sensitive quantitative predictions of peptide-MHC binding by a'Query by Committee' artificial neural network approach. Tissue Antigens, vol. 62, no. 5, pp. 378–384, 2003.

[12] J. Cui, L. Han, H. Lin, H. Zhang, Z. Tang, C. Zheng, Z. Cao, and Y. Chen, Prediction of MHC-binding peptides of flexible lengths from sequence-derived structural and physicochemical properties. MolImmunol, 2006.

[13] J. Salomon and D. Flower, Predicting Class II MHC-Peptide binding: a kernel based approach using similarity scores, BMC Bioinformatics, vol. 7, no. 1, p. 501, 2006.

[14] M. Nielsen and O. Lund, NN-align. An artificial neural networkbased alignment algorithm for MHC class II peptide binding prediction. BMC bioinformatics, vol. 10, no. 1, p. 296, 2009.

[15] I. Doytchinova and D. Flower, Towards the in silico identification of class II restricted Ts-cell epitopes: a partial least squares iterative self consistent algorithm for affinity prediction. pp. 2263–2270, 2003.

[16] C. Hattotuwagama, P. Guan, I. Doytchinova, C. Zygouri, and D. Flower, Quantitative online prediction of peptide binding to the major histocompatibility complex. Journal of Molecular Graphics and Modelling, vol. 22, no. 3, pp. 195–207, 2004.

[17] W. Liu, X. Meng, Q. Xu, D. Flower, and T. Li, Quantitative prediction of mouse class I MHC peptide binding affinity using support vector machine regression (SVR) models. BMC Bioinformatics, vol. 7, no. 1, p. 182, 2006.

[18] H. Bui, J. Sidney, B. Peters, M. Sathiamurthy, A. Sinichi, K. Purton, B. Moth´e, F. Chisari, D. Watkins, and A. Sette, Automated generation and evaluation of specific MHC binding predictive tools: ARB matrix applications. Immunogenetics, vol. 57, no. 5, pp. 304–314, 2005.

[19] M. Nielsen, C. Lundegaard, and O. Lund: Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method. BMC Bioinformatics, vol. 8, p. 238, 2007.

[20] Yasser EL-Manzalawy, Drena Dobbs, and Vasant Honavar: Predicting MHC-II binding affinity using multiple instance regression. IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS.

[21] P. Wang, J. Sidney, C. Dow, B. Moth´e, A. Sette, and B. Peters: A Systematic Assessment of MHC Class II Peptide Binding Predictions and Evaluation of a Consensus Approach. PLoS Computational Biology, vol. 4, no. 4, 2008.

[22] C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald, and J. Wootton: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science, vol. 262, no. 5131, pp. 208–214, 1993.

[23] S. Chang, D. Ghosh, D. Kirschner, and J. Linderman: Peptide length based prediction of peptide-MHC class II binding. Bioinformatics, vol. 22, no. 22, p. 2761, 2006.

[24] Y. Chen, J. Bi, and J. Wang, MILES: Multiple-instance learning via embedded instance selection. IEEE Trans Pattern Anal Mach Intell, vol. 28, no. 12, pp. 1931–1947, 2006.

[25] S. Henikoff and J. Henikoff, Amino Acid Substitution Matrices from Protein Blocks. Proceedings of the National Academy of Sciences of the United States of America, vol. 89, no. 22, pp. 10 915–10 919, 1992.

[26] S. Shevade, S. Keerthi, C. Bhattacharyya, and K. Murthy: Improvements to the SMO Algorithm for SVM Regression. IEEE Transactions on Neural Networks, vol. 11, no. 5, p. 1189, 2000.

[27] Zhi-Hua Zhou, Min-Ling Zhang: Solving Multi-Instance Problems with Classifier Ensemble Based on Constructive Clustering. Knowledge and Information Systems, 11(2):155-170, 2007.

[28] Nielsen M, Lundegaard C, Blicher T, Peters B, Sette A, Justesen S, Buus S, Lund O: Quantitative predictions of peptide binding to any HLADR molecule of known sequence: NetMHCIIpan. PLoS Comput Biol 2008, 4(7):e1000107.

[29] El-Manzalawy Y, Dobbs D, Honavar V: On evaluating MHC-II binding peptide prediction methods. PLoS One 2008, 3(9):e3268.

[30] R. H. Dietterich, T. G.; Lathrop and T. Lozano-Perez, Solving the multiple-instance problem with axis parallel rectangles. Artificial Intelligence, vol. 89(1-2), pp. 31–71, 1997.

[31] Zhou Z-H, Zhang M-L (2003) Ensembles of multi-instance learners. In Lavra·c N, Gamberger D, Blockeel H, Todorovski L (eds). Lecture Notes in Artificial Intelligence 2837, Springer, Berlin, pp 492-502.

[32] S. Andrews, I. Tsochantaridis, and T. Hofmann, Support vector machines for multiple-instance learning. Advances in Neural Information Processing Systems, vol. 15, 2002.

[33] T. Gartner, P. Flach, A. Kowalczyk, and A. Smola, Mul-

ti-instance kernels. Proceedings of the 19th International Conference on Machine Learning, pp. 179–186, 2002.

[34] Dietterich TG (2000) Ensemble methods in machine learning. In Kittler J, Roli F (eds). Lecture Notes in Computer Science 1867, Springer, Berlin, pp 1-15.

[35] Wang J, Zucker J-D (2000) Solving the multiple-instance problem: A lazy learning approach. In Proceedings of the 17th International Conference on Machine Learning, San Francisco, CA, 2000, pp 1119-1125.

[36] Weidmann N, Frank E, Pfahringer B (2003) A two-level learning method for generalized multi-instance problem. In Lavra·c N, Gamberger D, Blockeel H, Todorovski L (eds). Lecture Notes in Artificial Intelligence 2837, Springer, Berlin, pp 468-479.

[37] Jiawei Han & Micheline Kamber (2001), Data Mining Concepts and Techniques, pp351.

[38] Cristianini N, Shawe-Taylor J: An introduction to support vector machines and other kernel-based learning methods. Cambridge, UK, Cambridge University Press; 2000.

[39] Peng Wang, John Sidney, Yohan Kim, Alessandro Sette, Ole Lund, Morten Nielsen, Bjoern Peters: Peptide binding predictions for HLA DR, DP and DQ molecules. BMC Bioinformatics 2010, 11:568.

[40] Brusic V, Rudy G, Harrison L, Journals O. MHCPEP a database of MHCbinding peptides: update 1997. Nucleic Acids Res 26: 368–371.

[41] Bhasin M, Singh H, Raghava G (2003) MHCBN: a comprehensive database of MHC binding and non-binding peptides. Bioinformatics 19: 665–666.

[42] Raghava G. MHCBench: Evaluation of MHC Binding Peptide Prediction Algorithms. Available at http://www.imtech.res.in/raghava/mhcbench/.