Computational Ensemble Approach for Immune System Study: Conformational B-cell Epitope Prediction

Yuh-Jyh Hu[1,2*], Shun-Ning You[1] and Chu-Ling Ko[1]

[1] Department of Computer Science, National Chiao Tung University, 1001 University Rd, Hsinchu, Taiwan
[2] Institute of Biomedical Engineering, National Chiao Tung University, 1001 University Rd, Hsinchu, Taiwan

Abstract

Various tools have been developed to predict B-cell epitopes. We proposed a multistrategy approach by integrating two ensemble learning techniques, namely bagging and meta-decision tree, with a threshold-based cost-sensitive method. By exploiting the synergy among multiple retrainable inductive learners, it directly learns a tree-like classification architecture from the data, and is not limited by a prespecified structure. In addition, we introduced a new three-dimensional sphere-based structural feature to improve the window-based linear features for increased residue description. We performed independent and cross-validation tests, and compared with previous ensemble meta-learners and state-of-the-art B-cell epitope prediction tools using bound-state and unbound-state antigens. The results demonstrated the superior performance of the bagging meta-decision tree approach compared with single epitope predictors, and showed performance comparable to previous meta-learners. The new approach—requiring no predictions from other B-cell epitope tools—is more flexible and applicable than are previous meta-learners relying on specific pretrained B-cell epitope prediction tools.

Keywords

B-cell; Epitope; Meta-decision tree; Ensemble learning

Correspondence to:

Yuh-Jyh Hu
Department of Computer Science, National Chiao Tung University,
1001 University Rd, Hsinchu, Taiwan.
E-mail: yhu@cs.nctu.edu.tw

EJBI 2017; 14(1):04-15
received: September 01, 2017
accepted: October 26, 2017
published: November 06, 2017

1 Introduction

B-cell epitopes are specific regions on proteins recognized as antigen-binding sites by the antibodies of B cells. A detailed understanding of the interaction between antibodies and epitopes facilitates the development of diagnostics and therapeutics as well as rational design of preventive vaccines [1, 2, 3]. Therefore, generation of potent antibodies through reverse immunological approaches requires precise knowledge of epitopes. According to their structure and interaction with antibodies, epitopes can be classified as conformational and linear epitopes. A linear epitope is formed by a continuous sequence of amino acids, whereas a conformational epitope comprises discontinuous sections of the antigen's primary sequence; the discontinuous sections are close together in the three-dimensional (3D) space and interact with an antibody together. Approximately 10% of B-cell epitopes are linear, whereas the remaining 90% are conformational [4, 5].

The increasing availability of protein structures has facilitated the development of computational prediction tools by exploiting protein antigen structures. The following are some of the knowledge that has been used to elucidate these structures for epitope prediction: (a) spatial neighborhood information and a surface measure [6]; (b) local spatial context, accessible surface area (ASA) propensity and consolidated amino acid index [7]; and (c) the B-factor to detect atomic fluctuation [8]. Some studies have either adopted a hybrid approach combining structural and physicochemical features [9, 10], proposed a ensemble meta-learner incorporating consensus results from multiple prediction servers by using a voting mechanism [11], applied an ensemble of classifiers using various input features [12], or employed meta-learning based on stacking and cascade generalization [13].

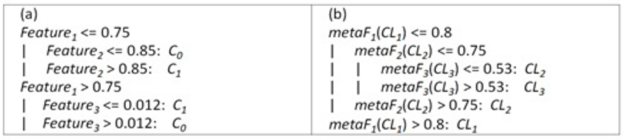
In this paper, we propose the use of a meta-decision tree (MDT) approach [14] for B-cell epitope prediction. The previous meta-learners have relied on the classifications of other B-cell epitope prediction tools, which are trained and may not be retrained conveniently by the user from new data; nevertheless, the current proposed meta-classifier is independent of these types of pretrained B-cell epitope prediction servers. Consequently, our approach is more applicable and flexible than the previously developed meta-learning methods for B-cell epitope prediction. Our goal is not to develop a specific classifier for B-cell epitope prediction, but instead we intend to advocate the applicability of a generic improved ensemble meta learning approach over current B-cell epitope classifiers, and to show its competitive performance with those of other methods. To evaluate the new B-cell epitope meta-learner, we performed cross-validation (CV) with previous meta-classifiers and compared them with major epitope predictors for the same test data sets used previously for consistency. The results indicate that the proposed MDT approach outperforms commonly used single prediction servers considerably, as well as exhibiting performance comparable to previously developed meta-learners.

2 Materials and Methods

2.1 MDTs

MDTs [14] are used for meta-learning that applies multiple base classifiers to a single data set by exploiting the classification results of the base classifiers as a type of meta-knowledge. The structure of an MDT is identical to that of an ordinary decision tree, in that both have internal nodes and leaves, and have the same computational complexity O (m n log n) where m is the number of attributes and n is the number of examples [15]; however, in an MDT, the attributes associated with the internal nodes and the meaning indicated by the leaves differ from those of an ordinary decision tree.

In both MDT and ordinary decision trees, an internal node specifies a test on an attribute value. For an ordinary decision tree, the attribute selected for the internal node must be one of the base attributes used to describe the data instances, for instance, the hydrophilic scale. By contrast, the attribute at an internal node of an MDT is a meta-attribute derived from the output of the base classifiers. Notably, although the base classifiers used in an MDT are standard inductive learners (e.g., artificial neutral network and naïve Bayes classifier), they differ from the B-cell epitope prediction servers (e.g., SEPPA 2.0 [7] and DiscoTope 2.0 [16]) used as the base predictors by other meta-learning methods [11, 13]. Unlike these servers, the base classifiers used in MDTs can be retrained from new training data if required. As for the leaves, a leaf of an ordinary decision tree corresponds to a predicted class, whereas that of an MDT specifies a particular base classifier for class prediction. Figure 1 illustrates examples of an ordinary decision tree and an MDT. The ordinary decision tree in Figure 1A includes three internal nodes and four leaf nodes; each internal node specifies a test on a particular base attribute value [e.g., $Feature_1 \leq 0.75$ (or > 0.75)], and each leaf indicates the predicted class (e.g., C_1). The MDT in Figure 1B also has three internal nodes and four leaf nodes; unlike in the ordinary decision tree, each internal node in this MDT specifies a test on a particular meta-attribute derived from the output of a base classifier [e.g., $metaF_1(CL_1)$ in Fig 1B], and rather than predicting the class, each leaf node predicts the base classifier most suitable for classification (e.g., CL_1). The advantage of an MDT is that it combines and exploits the classifications from multiple base classifiers to improve the accuracy of the final prediction.



(a)
$Feature_1$ <= 0.75
| $Feature_2$ <= 0.85: C_0
| $Feature_2$ > 0.85: C_1
$Feature_1$ > 0.75
| $Feature_3$ <= 0.012: C_1
| $Feature_3$ > 0.012: C_0

(b)
$metaF_1(CL_1)$ <= 0.8
| $metaF_2(CL_2)$ <= 0.75
| | $metaF_3(CL_3)$ <= 0.53: CL_2
| | $metaF_3(CL_3)$ > 0.53: CL_3
| $metaF_2(CL_2)$ > 0.75: CL_2
$metaF_1(CL_1)$ > 0.8: CL_1

(A) An ordinary decision tree, (B) A meta decision tree.
Figure 1: Sample ordinary decision tree and MDT.

2.2 Base Attributes and Meta-attributes

In the framework of machine learning, we translate an epitope prediction problem into an inductive learning problem. With a given training set of antigens with known epitope and nonepitope regions, the goal is to learn a classifier from a training set of antigens and apply the learned classifier to novel antigens for epitope detection.

To describe each amino acid on a protein antigen, we first adopted 14 base attributes: epitope propensity, secondary structure, residue accessibility, B factor, solvent-excluded surfaces, solvent-accessible surfaces, protein chain flexibility, hydrophilicity, position-specific scoring matrix (PSSM), atom volume, accessible surface area, side chain polarity, hydropathy index, and antigenic propensity. Descriptions and references of the base attributes have been reported by Hu et al. [13]. In addition, we considered five other physicochemical properties: surface probability [17], turns [18], exposed surface [19], and two types of polarities defined by Ponnuswamy et al. [20] and Grantham [21]. All the attributes were derived from the protein sequences or from the structural information provided by PDB. To keep the consistency, we prepared the attributes and their values for the training and testing on MDT in the same way without any discrepancy. Furthermore, we extended the idea of Ansari and Raghava [22] and created the 3D sphere-based attributes from the base attributes; the authors only considered amino acids in a linear window to generate one-dimensional (1D) window-based attributes. By contrast, we considered the amino acids in a constrained 3D spherical space and analyzed the values of different base attributes. To generate a 3D-based attribute value for an amino acid, we first created a surrounding sphere with its central carbon atom C_α as the center. We then computed the average value for a base attribute of all the amino acids within this 3D sphere and used the average as the 3D sphere-based attribute value for the center amino acid. Figure 2 illustrates a 3D spherical neighborhood for an amino acid. Because 90%

of the B-cell epitopes are conformational (discontinuous), 3D sphere-based attributes are more informative than 1D window-based attributes. By varying the radius of the sphere, we could define different 3D sphere-based attributes. With the base and 3D sphere-based attributes, we represented the protein antigens in the training set and trained the base classifiers from the base training data set.

A meta-attribute is defined over the output of the trained base classifiers. We used RFs [23], SVMs [24], C4.5 [25], k-NN [26], PART [27], BN [28], JRip [29], and VP [30] as the base classifiers. Furthermore, the majority vote of the base classifiers was also included as a base classification. According to Todorovski and Dzeroski [14], we calculated the properties of the class probability distributions predicted by the base classifiers, reflecting the certainty and confidence of the predictions. Here, we defined three meta-attributes: *epi_prob*(x,B), *entropy*(x,B), and *vote_epi_prop*(x), where x is a data instance and B a base classifier. The meta-attribute *epi_prob*(x,B) is the probability of epitope predicted by the base classifier B for the amino acid x. The meta-attribute *entropy*(x,B) is the entropy of the class probability distribution predicted by the base classifier B for the amino acid x. The meta-attribute *vote_epi_prop*(x) is the proportion of the epitope class predicted by all base classifiers for the amino acid x. These meta-attributes reflect the certainty of the base classifier in predicting the class, and they characterize the confidence variedly.

We computed the meta-attribute values for each data instance, namely the amino acid, on the basis of the output of the base classifiers and combined them to form a meta training data set. We then trained an MDT from a training set of data described by the meta-attributes. The metadata preparation process is illustrated in Figure 3. The meta training data can be obtained offline and independently of MDT, and consequently do not affect the training of MDT directly.

2.3 MDT Construction

MDT construction is identical to that of an ordinary decision tree. It involves a greedy, top-down, recursive search for the most suitable decision tree from a training data set. The core algorithm selects the most suitable attribute for an internal node and partitions the data available at the node into subsets according to the attribute values to create the descending nodes. This process is repeated for the data associated with each descendant to select the next attribute to grow the tree until some stopping criterion is satisfied.

Rather than employing the measures of impurity reduction commonly used for ordinary decision trees, such as information gain, gain ratio [25], and Gini [31], the focus

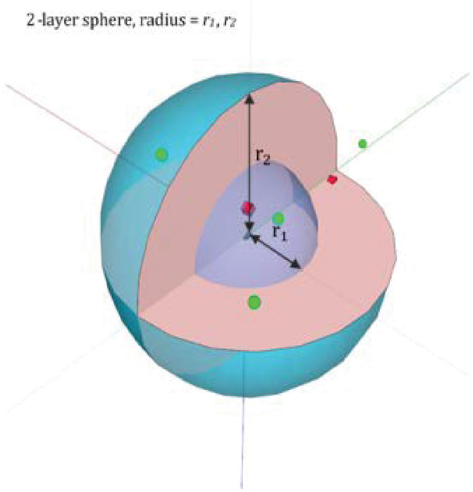


Figure 2: Two-layer 3D spherical neighborhood. A 3D sphere-based structural feature is defined for an amino acid (AA) based on its 3D spherical neighborhood specified by radius, r. The center of the sphere is the C_α of AA, and every other AA whose C is within the distance of r is considered a neighbor of AA. By varying r, we can specify different 3D spherical neighborhoods to define different 3D sphere-based structural features.

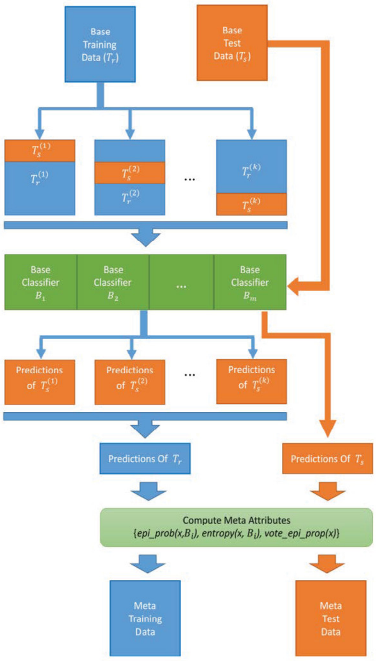


Figure 3: Metadata preparation process.Base training data *Tr* are divided into k folds. In each iteration, (k-1) folds of *Tr* are fed to all the base classifiers for training and the trained base classifiers are tested on the remaining fold to obtain the predictions. After repeating the same procedure k times, the predictions of the entire *Tr* are obtained from all base classifiers. To prepare meta training data, for each data instance x, the values of its three meta-attributes, *epi_prob*(x,B_i), *entropy*(x,B_i), and *vote_epi_prop*(x), are computed; these values together form the meta training data. Meta test data are prepared similarly.

of MDTs is the accuracy of each base classifier in predicting the data instance *S* available at an internal node. We defined the new information measure as follows:

$$info(S) = 1 - max_{B \in Base}\ Accuracy(B,S) \qquad (1)$$

Where, *B* is a base classifier, *Base* is the set of all base classifiers, *S* is the data available at an internal node, and *Accuracy* is the classification accuracy of *B* on *S*. We selected the attribute that maximized the decrease in *info* of the subsets of *S* after the partition according to the values of the selected attribute compared with the original *info* of *S*. We repeated the same selection–partition process to grow the tree until the accuracy of some base classifier was 100% on the current subset or the size of the current subset was lower than a prespecified threshold. The classifier at the leaf node with the maximum accuracy was used to predict new instances after the tree grew completely.

2.4 Bagging and Decision Threshold

Three factors affect the performance of a predictor: (a) training d1ata, (b) representation bias, and (c) search bias. First, learning algorithms for predictors may produce different hypotheses from a small data set, and each has the same prediction accuracy on the same training data. Nevertheless, no single hypothesis can cover the entirety, or a sufficiently large portion, of the hypothesis space. Consequently, the overfitting problem can arise. Second, in most real-world applications, the true target concept may not be represented by any single hypothesis in the hypothesis space because of the limitations of representations, constraining the applicability of the learned hypothesis. Third, most learning algorithms adopt a local search strategy to prevent computational explosion during learning (e.g. the commonly used greedy partition rule for growing a decision tree); however, greedy local search of any form causes higher variance and can be stuck in local optima, thus failing to identify the true target.

An approach applicable for mitigating the aforementioned problems is ensemble learning. Various forms of ensemble learning have been developed. To further enhance the immunity of MDTs to the preceding problems, we applied a bagging-like strategy [32] to MDTs. By constructing an MDT ensemble from multiple random samples, we assumed that the final prediction based on the predictions of the MDTs ensemble can further reduce the variance among different MDTs, providing a more accurate approximation to the true target.

Most learning algorithms assume balanced class distributions and equal misclassification costs, which limit their applicability to epitope prediction because the B-cell epitopes are severely underrepresented. We adopted the undersampling strategy to decrease the number of nonepitopes in the training data to mitigate the class imbalance problem with B-cell epitope prediction. The appropriate epitope-to-nonepitope ratio for undersampling was first determined by either CV on training data or prior knowledge of class distribution in test data. We then performed multiple undersampling runs according to the selected ratio to create multiple random data sets to train an MDT ensemble.

We computed the probability for an amino acid of being epitope or nonepitope on the basis of the predictions of the MDTs trained from the random samples. By using *m* MDTs, we defined the scores of epitope, $Score_{E,}$ and nonepitope, $Score_{N}$, for an amino acid *AA* as follows:

$$Score_E = \alpha \cdot \sum_{i=1}^{m} w_i \cdot \lceil e_i - 0.5 \rceil + (1-\alpha) \cdot \sum_{i=1}^{m} w_i \cdot e_i \qquad (2)$$

$$Score_N = \alpha \cdot \sum_{i=1}^{m} w_i \cdot \lceil n_i - 0.5 \rceil + (1-\alpha) \cdot \sum_{i=1}^{m} w_i \cdot n_i \qquad (3)$$

In equations (2) and (3), e_i (or n_i) is the probability of being an epitope (or nonepitope) according to the prediction of the *i*th MDT. To predict the class of *AA*, we traversed each MDT to a leaf that specified a base classifier to make the prediction. We obtained the proportion of a base classifier used to make predictions from the *m* MDTs and denoted the proportion for each base classifier by using w_i. A higher w_i value indicates a stronger weight of that base classifier exerted on the score; when w_i is set to 1, all the base classifiers are treated equally. The first term in equations (2) and (3) considers only the count of classifications by the *m* MDTs, whereas the second term considers the class probabilities. We used a control parameter α to balance the effects of the two scoring mechanisms, and its value could be determined through CV. We defined the probability for the amino acid *AA* of being epitope or nonepitope as follows:

$$P_N(AA) = \frac{Score_N}{Score_E + Score_N} \qquad (4)$$

$$P_N(AA) = \frac{Score_N}{Score_E + Score_N} \qquad (5)$$

To appropriately address the imbalanced class distribution in B-cell epitopes, we also set a probability threshold for the final classification as follows:

$$Class(AA) = \begin{cases} non-epitope, & P_E(AA) < \theta \\ epitope, & P_E(AA) \ge \theta \end{cases} \qquad (6)$$

Where θ is a threshold. A carefully selected on the basis of CV or prior knowledge warrants a reasonable performance of the class-sensitive bagging MDT approach. Figure 4 illustrates the entire flow of this system.

2.5 Data Sets and Performance Measures

We collected the training and test data used in DiscoTope 2.0 [16], SEPPA 2.0 [7], Bpredictor [33], ElliPro [34], CBTOPE [22], EPMeta [10], and B-cell meta-classifiers [13] and by Zhao et al. [35]; we then combined these with the data in the Epitome database [36] and Immune Epitope Database

(IEDB) [37] to prepare the data set for the comparative study. After removing the duplicates and filtering out the antigens without epitope residues annotated in either Epitope Information or B cell Assay Information in the IEDB, we obtained a total of 363 bound-state antigens. Because the epitope predictors used in our study were web-based servers or software packages that could not be retrained using different training data, to conduct a consistent and unbiased comparative analysis of the prediction performances of these predictors, we created an independent data set of antigens with known epitopes. We divided this data set of 363 protein antigens into a test data set and a training data set. To ensure fair comparison between different prediction methods, we selected 18 antigens that were not used before to train any of the predictors in comparison for testing, and used the remaining 345 antigens that were used previously to train these predictors for MDT training. Table 1 lists the total 363 antigens, and Table 2 lists the 18 test antigens.

While most of the studies of epitope prediction and feature analysis are focused on bound-state antigen structures [6, 7, 16, 34, 35], epitopes in bound states show different characteristics and reveal more binding information than unbound epitopes [38], which can raise two issues in epitope prediction. One is that explicit binding information in the bound-state structures can bias the prediction performance; the second is that an antigen possibly bound by multiple antibodies can cause more false negatives because only the epitope to the antibody in the bound structure is considered a true epitope, and all remaining epitopes to other antibodies are labeled as nonepitopes. We adopted a set of unbound-state antigens, listed in Table 3, recently constructed and annotated by Ren et al. [39] to evaluate MDT's performance for unbound-state epitope prediction, and compare it with other epitope predictors.

We evaluated prediction performances by using several measures: TPR (i.e., sensitivity), FPR, precision (i.e., positive predictive value), percentage accuracy, F-score, and MCC. Table 4 lists the definitions of these measures. In general, correlation exists between the TPR and FPR produced by a predictor. Typically, the FPR increases with the TPR. We prepared ROC curves to summarize the results on the different thresholds.

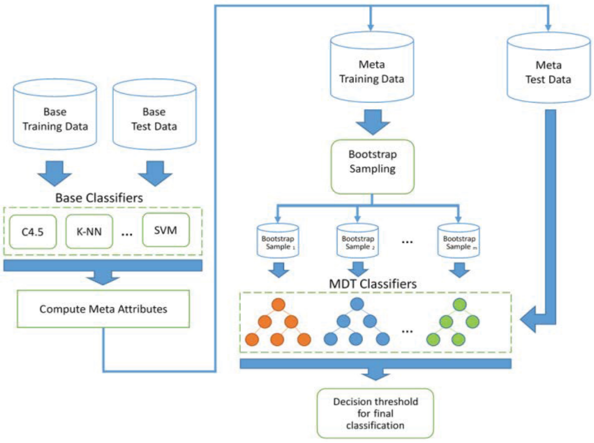


Figure 4: System flow of bagging MDT.

2.6 Correlation Analysis of Base Classifiers and Ablation Analysis of Base Attributes

An MDT can be constructed from an arbitrary number of different base classifiers, and its overall performance depends on these learning components. If the learning components have complementary predictive strengths, an MDT can search various hypotheses in the hypothesis space and provide superior generalizations for novel test data to those of a single-component learner. We used the ARI [40] to measure the strength of the relationship between the predictions produced by two base classifiers. Although the ARI was initially designed to measure agreement between two clustering results, in our case, a higher ARI value could indicate greater agreement between the two classifiers. If P is the partition of the amino acids into epitopes and nonepitopes for a given data set of antigens, according to the predictions of the classifier A, and Q is the partition produced by the classifier B, a lower ARI value between P and Q suggests a higher probability that the two classifiers have complementary strengths. After evaluating several different indices for the measurement of the agreement between two partitions, Milligan and Cooper [41] recommended the use of the ARI. Therefore, in this study, we adopted the ARI for evaluating the correlation between the classifiers. The results from the ARI analysis provided a basis for selecting the appropriate base classifiers in MDTs.

In addition to assessing the complementary prediction strengths of the base classifiers by using the ARI, we conducted an ablation analysis of different base attribute types to measure their contribution to the MDT. We classified the base attributes into three categories: (a) sequence, (b) structure, and (c) 3D sphere-based. We compared the relevance of the three categories of base attributes to MDT by their removal or addition and estimated their effects on the meta-classification by the amount of decrease or increase in prediction performance.

3 Results

To keep the consistency in evaluating MDT's performances, we followed the same protocol to prepare the attributes and their values for MDT's training as well as its testing without any discrepancy. All the attributes were derived from the protein sequences or from the structural information provided by PDB.

3.1 Prediction Correlations Between Base Classifiers

For a meta-learning method to perform effectively, the base classifiers must have complementary predictive capabilities,

Table 1: Data set of 363 protein antigens.

1A2Y_C	1ADQ_A	1AFV_A	1AHW_C	1AHW_F	1AR1_B	1BGX_T	1BJ1_V	1BJ1_W	1BQL_Y
1BVK_C	1BVK_F	1BZQ_A	1C08_C	1CIC_C	1CIC_D	1CZ8_V	1CZ8_W	1DQJ_C	1DVF_A
1DVF_B	1DZB_X	1DZB_Y	1E6J_P	1EGJ_A	1EO8_A	1EZV_E	1FBI_X	1FDL_Y	1FJ1_F
1FNS_A	1FSK_A	1FSK_D	1FSK_G	1FSK_J	1G7H_C	1G7I_C	1G7J_C	1G7L_C	1G7M_C
1G9M_G	1G9N_G	1GC1_G	1HYS_A	1HYS_B	1I9R_A	1IAI_H	1IAI_L	1IC4_Y	1IC5_Y
1IC7_Y	1J1O_Y	1J1P_Y	1J1X_Y	1J5O_B	1JHL_A	1JPS_T	1JRH_I	1JTO_L	1JTO_M
1JTP_L	1JTP_M	1JTT_L	1KB5_A	1KB9_E	1KEN_A	1KIP_C	1KIQ_C	1KIR_C	1KXQ_A
1KXT_A	1KXV_A	1KYO_E	1LK3_A	1LK3_B	1MEL_L	1MEL_M	1MHP_B	1MLC_E	1MLC_F
1N5Y_B	1N6Q_B	1N8Z_C	1NBY_C	1NBZ_C	1NDG_C	1NDM_C	1NSN_S	1OAK_A	1OAZ_A
1OP9_B	1ORQ_C	1ORS_C	1OSP_O	1OTS_A	1OTS_B	1OTT_A	1OTT_B	1OTU_A	1OTU_B
1P2C_C	1P84_E	1PG7_L	1PKQ_J	1QFW_A	1QLE_B	1R0A_B	1R3I_C	1R3J_C	1R3K_C
1R3L_C	1RI8_B	1RJC_B	1RJL_C	1RVF_1	1RVF_2	1RVF_3	1RZJ_G	1S78_B	1SQ2_L
1T6V_L	1T6V_M	1TPX_A	1TQB_A	1TQC_A	1TZH_V	1TZH_W	1TZI_V	1UA6_Y	1UAC_Y
1UJ3_C	1V7M_V	1VFB_C	1W72_A	1WEJ_F	1XGP_C	1XGQ_C	1XGR_C	1XGT_C	1XGU_C
1XIW_A	1XIW_E	1YJD_C	1YQV_Y	1YY9_A	1YYM_G	1Z3G_A	1Z3G_B	1ZA3_R	1ZTX_E
1ZV5_L	2AEP_A	2ARJ_Q	2B2X_A	2BDN_A	2BOB_C	2BOC_C	2DD8_S	2DQC_Y	2DQD_Y
2DQE_Y	2DQF_C	2DQF_F	2DQG_Y	2DQH_Y	2DQI_Y	2DQJ_Y	2DWD_C	2DWE_C	2EIZ_C
2EKS_C	2FJG_V	2FJG_W	2GHW_A	2H9G_R	2H9G_S	2HMI_B	2I25_L	2I25_M	2I26_L
2I26_M	2I26_Q	2I60_G	2I60_P	2I9L_I	2IFF_Y	2ITD_C	2J4W_D	2J5L_A	2JEL_P
2NXY_A	2NXZ_A	2NY0_A	2NY1_A	2NY2_A	2NY3_A	2NY4_A	2NY5_G	2NY6_A	2NY7_G
2OZ4_A	2P42_A	2P42_C	2P43_A	2P44_A	2P45_A	2P46_A	2P46_C	2P47_A	2P48_A
2P49_A	2P4A_A	2P4A_C	2Q8A_A	2Q8B_A	2QQK_A	2QQN_A	2R29_A	2R4R_A	2R4S_A
2R56_A	2UZI_R	2VH5_R	2VIR_C	2VIS_C	2VIT_C	2VXQ_A	2VXS_A	2VXS_B	2VXT_I
2W9E_A	2XQB_A	2XQY_A	2XQY_E	2XTJ_A	2XWT_C	2YBR_C	2YBR_F	2YBR_I	2YC1_C
2YC1_F	2YSS_C	2ZJS_Y	2ZNW_Y	2ZNW_Z	2ZNX_Y	2ZNX_Z	2ZUQ_A	3A67_Y	3A6B_Y
3A6C_Y	3B9K_B	3BDY_V	3BE1_A	3BGF_A	3BGF_S	3BQU_B	3BSZ_E	3BSZ_F	3C09_A
3CVH_A	3CVH_M	3D85_C	3D9A_C	3DVG_Y	3DVN_V	3EO1_C	3EO1_F	3FMG_A	3G04_C
3G6D_A	3GB7_C	3GBM_B	3GBN_B	3GI8_C	3GI9_C	3GRW_A	3H42_A	3H42_B	3HFM_Y
3HI6_A	3HI6_B	3HMX_A	3I50_E	3IDX_G	3IGA_C	3J1S_A	3K3Q_B	3K3Q_C	3KJ4_A
3KJ6_A	3KR3_D	3L5W_I	3L5X_A	3LD8_A	3LDB_A	3LEV_A	3LH2_S	3LH2_T	3LH2_U
3LH2_V	3LHP_S	3LHP_T	3MA9_A	3MJ9_A	3MXW_A	3NGB_G	3NH7_A	3O0R_B	3O0R_C
3O2D_A	3PGF_A	3Q3G_E	3QA3_E	3QA3_G	3QA3_I	3QA3_L	3QWO_P	3R1G_B	3RU8_X
3RVV_A	3RVW_A	3RVX_A	3SDY_A	3SDY_B	3SE8_G	3SE9_G	3SKJ_E	3SKJ_F	3SOB_B
3SQO_A	3T2N_A	3THM_F	3TJE_F	3U2S_C	3U2S_G	3U30_D	3UC0_A	3UC0_B	3UX9_A
3UX9_C	3VG9_A	4AEI_A	4AEI_B	4AEI_C	4AL8_C	4ALA_C	4DGI_A	4DKE_A	4DKE_B
4DKF_A	4DKF_B	4DN4_M	4DTG_K	4ETQ_C	4F2M_E	4F2M_F	4F3F_C	4FQI_B	4GMS_A
4GMS_C	4GMS_E	4HKX_E							

Table 2: Independent test data set of 18 protein antigens.

1BZQ_A	1KXT_A	1KXV_A	1W72_A	2I9L_I	2J4W_D	2J5L_A	2OZ4_A	2R4R_A	2R4S_A
2ZJS_Y	3B9K_B	3BQU_B	3BSZ_E	3BSZ_F	3DVN_V	3KJ4_A	3KJ6_A		

Table 3: Data set of unbound-state antigens.

3TGT	4GXX	4OIE	2I5V	1J95	2QTW	1D7P	1B1I	1YG9
4I53	4M4Y	2C36	2G7C	1JVM	1KF3	1F45	1DOK	1FCQ
4JPJ	1F8D	4OSN	1POH	1DKK	1ZVM	2ILK	1KEX	1BV1
4DKP	3NN9	2GHV	1UB4	1HHL	3OIW	2NVH	3M1N	3PX8
3O3X	4NN9	3K7B	1WHO	1JSE	4KZN	1MF7	3MJ6	
4IPY	3IRC	4E9O	1I4M	2NWD	4EFV	1MJN	3NCL	
3NTE	3VTT	3EJC	3Q27	2VB1	1BOY	2ICA	3S26	
1HGH	3WE1	1W8K	4NX7	1HX0	1ATZ	3FCU	4GNY	
3KU3	3GGQ	1Z40	3KVD	3ZKG	1AUQ	2YXF	1AHO	
3ZP0	2HG0	3WKL	1EY0	3Q6O	1IJB	3O1Y	3D6S	
4FNK	2P5P	2WK0	1IGD	2GBC	1BIO	4JNI	3F5V	

Table 4: Definitions of performance measures.

Performance Measure	Definition
TPR[a]	TP/(TP+FN)
FPR	FP/(FP+TN)
Precision[b]	TP/(TP+FP)
Accuracy	(TP+TN)/(TP+TN+FP+FN)
F-score	2×TPR×Precision/(TPR+Precision)
MCC	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

[a]TPR is also known as sensitivity or recall.
[b]Precision is also known as positive predictive value.

which can be reflected by relatively low correlation among their predictions. We selected eight base classifiers to construct an MDT. They are random forests (RFs) [23], support vector machines (SVMs) [24], C4.5 [25], k-nearest neighbors (k-NN) [26], PART [27], Bayes Net (BN) [28], JRip [29] and Voted Perceptron (VP) [30].

We measured the correlation between two base classifiers by the adjusted Rand index (ARI) [40] of their classifications. Table 5 lists the ARI values of all pairs of the base classifiers for an independent test data set of 18 antigens (see Materials and methods). The mean ± standard deviation ARI values for the test data sets were 0.238 ± 0.084; the ARI value is relatively low, indicating a relatively weak correlation among the base classifiers.

The 18-antigen independent test data set contained a total of 243 epitope and 3,760 nonepitope residues. A base classifier was trained using the training sets of 345 antigens; for the test set, it classified a protein residue as an epitope or a nonepitope. For each epitope and nonepitope residue, we counted the number of base tools that correctly classified the residue as epitope or nonepitope. Figure 5 presents the distributions of epitope and nonepitope residues for all test proteins, according to the number of base tools with the same prediction, for indicating the degree of agreement in classification among the base tools. For instance, as shown in Fig 5, we observed only 1.6% of the epitope residues were correctly and unanimously predicted by all base classifiers, whereas 32.5% of the epitopes could not be detected by any base predictor. By contrast, more than 65% of the epitope residues were correctly classified by 1–7 base predictors. Compared with epitopes, a markedly higher percentage of nonepitopes was correctly classified by all base predictors. Despite the higher percentage of unanimous predictions, more than 30% of the nonepitopes were classified variedly, and they were correctly predicted by 1–7 base classifiers. Taken together, these results indicate that base classifiers do not always agree when predicting epitopes and that they may have complementary strengths, suggesting that a meta-learner built based on these base learners can demonstrate synergy in their predictive capabilities.

3.2 Independent Test of MDT and other B-cell Epitope predictors

Five representative conformational epitope predictors [7, 16, 22, 33, 34], and three meta-classifiers [11, 13] were considered for comparison in the test. We compared MDT with current B-cell epitope predictors on a test data set of 18 antigens selected from a total of 363 bound antigens that were previously used to train and test epitope prediction tools (see Materials and methods). We did not perform k-fold cross-validation (CV) based on the 363 antigens because some epitope predictors, such as DiscoTope 2.0 and SEPPA 2.0 in this comparative study, had been pretrained using different antigens from the 363 antigens, and the antigens previously used for training can overlap the test data folds in the iterative process of CV. This violates the principles of CV, and is likely to cause overestimated performance. To avoid the bias, we performed an independent test instead.

We trained a bagging MDT model from 354 antigens, which were used previously to train the predictors in comparison, and compared its performance with the performances of other B-cell epitope predictors for the same test set of 18 antigens. The 18 test antigens were not used before to train any of the predictors in comparison to ensure a fair comparison. For each predictor in comparison, we selected the parameter values of their best-performing models for the training data set individually; these values were used in the tests to ensure fair comparison. Table 6 shows that the meta-classifiers, cascade, stacking, and bagging MDT, all considerably outperformed the five representative conformational epitope predictors for ACC, F-score, and MCC. The results revealed that DiscoTope 2.0 [16], ElliPro [34], and Bpredictor [33] presented high true positive rates (TPRs) of prediction; nevertheless, they also had high false positive rates (FPRs). By contrast, SEPPA 2.0 [7] and CBTOPE [22] showed lower TPRs as well as FPRs. Compared with these single predictors, the meta-classifiers, cascade, stacking, and bagging MDT led to a more favorable balance between TPRs and FPRs, and consequently demonstrated a higher F-score and MCC. Figure 6 shows their ROC curves, indicating the trade-off between the amounts of true positives (TP) and false positives (FP) produced by the classifiers. In general, these observations suggest that the performance of bagging MDT is superior to that of other current B-cell epitope prediction methods. In addition, the bagging MDT approach is comparable with previously reported epitope meta-classifiers, such as EPMeta [11], cascade, and stacking [13], while unlike previous meta-classifiers, it does not depend on the prediction output of specific pretrained B-cell epitope predictors such as DiscoTope 2.0 [16] and SEPPA 2.0 [7]. Notably, when bagging MDT also employed the output of the eight prediction tools, e.g. SEPPA 2.0, as used in cascade and stacking [13], it produced the highest performance with regard to ACC, F-score, and MCC, as presented in the final row of Table 6.

Table 5: Correlation analysis of base classifiers based on 18 antigens.

18 Ags	C4.5	KNN	Voted Perceptron	PART	Random Forest	Bayes Net	JRip
KNN	0.198	-	-	-	-	-	-
Voted Perceptron	0.189	0.256	-	-	-	-	-
PART	0.290	0.259	0.282	-	-	-	-
Random Forest	0.157	0.245	0.166	0.164	-	-	-
BayesNet	0.232	0.120	0.126	0.240	0.050	-	-
JRip	0.251	0.239	0.281	0.306	0.237	0.191	-
SVM	0.248	0.359	0.382	0.386	0.361	0.125	0.335

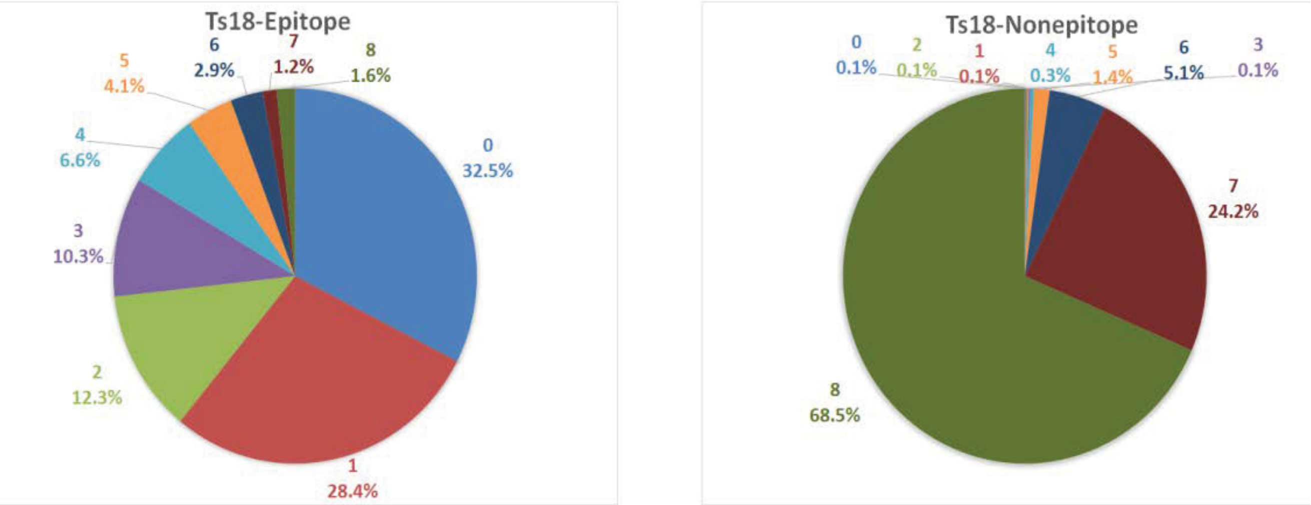


Figure 5: Pie charts of classification agreement. Pie charts showing the degree of agreement among base tools in epitope prediction of 18 test protein antigens. Distributions of the counts of (A) epitope and (B) nonepitope residues, according to the number of base tools with the same prediction.

Table 6: Results of independent test.

Classifier	TPR	FPR	PPV	ACC	F-score	MCC
SEPPA	0.140	0.047	0.162	0.904	0.150	0.100
DiscoTope	**0.959**	0.756	0.076	0.287	0.140	0.115
ElliPro	0.807	0.558	0.085	0.464	0.155	0.120
Bpredictor	0.856	0.634	0.080	0.396	0.147	0.111
CBTOPE	0.029	**0.001**	0.636	**0.940**	0.055	0.127
EPMETA	0.757	0.522	0.086	0.495	0.154	0.112
Cascade	0.222	0.016	0.466	0.937	0.301	0.293
Stacking	0.243	0.015	**0.513**	**0.940**	**0.330**	**0.326**
F19_BaggingMDT_BaseOnly[a]	0.148	0.017	0.356	0.932	0.209	0.199
F19_BaggingMDT_plus3Dsphere[b]	0.255	0.019	0.466	0.937	**0.330**	0.315
F19_BaggingMDT_plus3Dsphere_8T[c]	*0.272*	*0.014*	*0.555*	*0.943*	*0.365*	*0.362*

[a] Bagging MDT used only 19 base attributes and no 3D sphere-based attributes.

[b] Bagging MDT used 19 base attributes and the derived 3D sphere-based attributes.

[c] Bagging MDT used 19 base attributes, the derived 3D sphere-based attributes, as well as the output of the eight prediction tools, e.g. SEPPA, used in Cascade and Stacking.

3.3 Individual Comparisons Between MDT and Other B-cell Epitope Predictors

In addition to the tests of bagging MDT and the commonly used epitope predictors on the same test data, we compared bagging MDT with the epitope predictors separately by using different data sets. We tested six representative epitope predictors: SEPPA 2.0 [7], DiscoTope 2.0 [16], ElliPro [34], Bpredictor [33], CBTOPE [22], and EPMeta [11]. All had been trained and tested by different data sets. In each test, we selected one epitope predictor

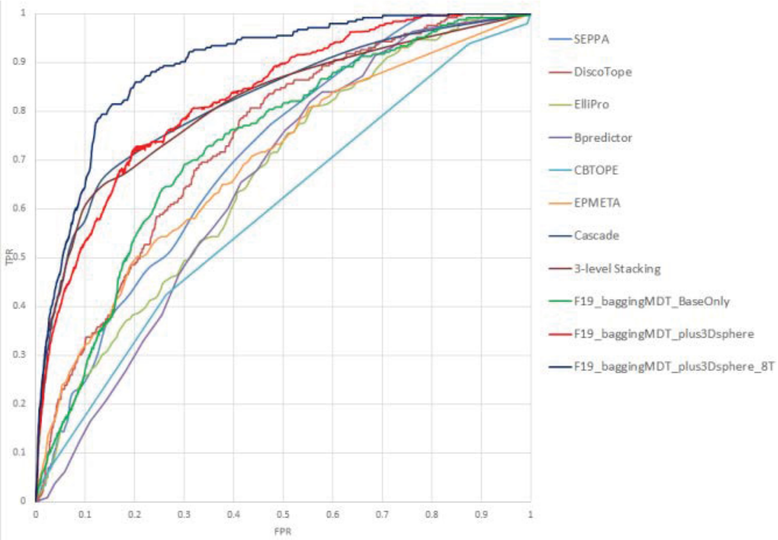


Figure 6: ROC curves of predictors. ROC curves of epitope predictors and meta-classifiers based on the test data set of 18 antigens.

for comparison. To make a fair and consistent comparison, we only trained and tested bagging MDT on the same data sets that had been used specifically to train and test the predictor selected for comparison. The results of the individual comparisons are presented in Table 7. Bagging MDT considerably outperformed SEPPA 2.0, ElliPro, Bpredictor, CBTOPE, and EPMeta in the individual tests except DiscoTope 2.0. As shown in Table 7, in the individual comparison with DiscoTope 2.0, bagging MDT was overconservative, as indicated by its significantly lower TPR and FPR, which caused weaker performance than DiscoTope 2.0. In general, these results demonstrate that the synergy in multiple base classifier effects in MDT can result in performance superior to other current epitope predictors. Furthermore, unlike previous meta-classifiers, the new bagging MDT approach does not require the prediction output of other epitope predictors [11] or a prespecified classification architecture [13], and is thus more flexible and applicable.

3.4 Evaluation of MDT based on Unbound-State Antigens

In addition to the independent test for bound-state antigens, we compared MDT with other epitope predictors by a set of unbound-state antigens. To maintain the consistency with the previous study in [39], we used the same unbound-state antigen data and also conducted antigen-based 10-fold CV by dividing the same unbound-state structures into 10 subsets randomly. The overall performance was used as the average of the results obtained from all iterations of three 10-fold CVs. The experimental results in Table 8 demonstrated that MDT outperformed the structure-based predictors DiscoTope 2.0, SEPPA 2.0 and ElliPro markedly for MCC, and showed comparable performance in F-score. Table 8 also showed that MDT was competitive with a two-stage SVM-based unbound epitope predictor PUPre [39]. In addition, the significantly higher precision and lower recall of MDT suggested that MDT was more conservative than the other tools for prediction of epitopes in unbound-state antigens.

3.5 Ablation Analysis

An ablation study provides insight into the effects of base learners on the prediction performance of a meta classifier. However, the time required for a complete ablation analysis increases exponentially with the number of base learners. To avoid computational explosion, following [13], we adopted a greedy approach for the ablation study. We used the same 94 antigens in [13] for training, and tested the trained classifiers on an independent set of 69 antigens.

We adopted two greedy iterative approaches, backward elimination and forward selection, to evaluate the contributions of available base classifiers. The greedy iterative backward elimination approach started with the maximal MDT built upon all available base learners. In each iteration, we identified the base classifier in MDT such that the performance of MDT decreased the most after the removal of this classifier. By contrast, the greedy iterative forward selection approach started with the minimal MDT built without any base classifier. In each iteration, we identified the base classifier available such that the addition of this classifier improved MDT's performance the most. We compared the relevance of the base classifier to MDT by the order of their removal or addition, and estimated their effects on MDT by the amount of decrease or increase in prediction performance. We show the results in Tables 9 and 10.

4 Conclusion

Numerous approaches can be used for predicting linear and conformational B-cell epitopes. Of these, the approach based

Table 7: Results of individual comparisons.

Classifier	TPR	FPR	PPV	ACC	F-score	MCC
SEPPA[a]	0.138	0.040	0.161	0.916	0.148	0.105
F19_BaggingMDT_plus3Dsphere	0.277	0.044	0.260	**0.920**	**0.268**	**0.226**
DiscoTope[a]	0.891	0.589	0.091	0.441	**0.166**	**0.149**
F19_BaggingMDT_plus3Dsphere	0.045	0.006	0.350	**0.935**	0.080	0.107
ElliPro[b]	0.740	0.519	0.125	0.505	0.214	0.128
F19_BaggingMDT_plus3Dsphere	0.606	0.025	0.707	**0.941**	**0.653**	**0.623**
Bpredictor Ts=122[c]	0.093	0.049	0.125	0.892	0.107	0.051
F19_BaggingMDT_plus3Dsphere	0.486	0.028	0.566	**0.938**	**0.523**	**0.492**
CBTOPE[d]	0.647	0.164	0.236	0.823	0.346	0.314
F19_BaggingMDT_plus3Dsphere	0.785	0.012	0.834	**0.973**	**0.809**	**0.795**
EPMeta Ts=149[e]	0.871	0.579	0.096	0.450	0.173	0.148
F19_BaggingMDT_plus3Dsphere	0.343	0.021	0.533	**0.937**	**0.417**	**0.396**

[a]Bagging MDT was trained and tested using the same data sets used specifically to train and test SEPPA 2.0 (or DiscoTope 2.0), excluding the antigens with missing feature values. All classifiers were tested on the same test data to conduct a consistent comparison.

[b]ElliPro only provided the test data but no training data. Consequently, we trained bagging MDT from our training data used in this study, excluding the antigens in ElliPro's test data set. We compared bagging MDT and ElliPro on the same test data previously used by ElliPro, excluding the antigens with missing feature values, to maintain consistency.

[c]Bagging MDT was trained on the same data previously used to train Bpredictor, excluding the antigens with missing feature values. Although Bpredictor provided its original test data set, the data lacked the epitope residues annotated in the IEDB. Alternatively, from the 363 antigens in this study we selected 122 antigens that were not used for Bpredictor training as the test data.

[d]Bagging MDT was trained and tested using the 60% nonredundant benchmark dataset previously used to evaluate CBTOPE, excluding the antigens with missing feature values. Following CBTOPE, we adopted 5-fold CV to compare the performances. We selected the parameter value of the best-performing CBTOPE on the training data set. All the classifiers, including CBTOPE, were tested on the same test data for consistent comparison.

[e]EPMeta only provided the training data but no test data. Consequently, we trained bagging MDT from the same training data used previously to train EPMeta, excluding the antigens with missing feature values, and performed a comparison between bagging MDT and EPMeta on the same test data set of 149 antigens selected from the 363 antigens in this study.

Table 8: Results of cross-validation on unbound-state antigens.

Classifier	Recall	Precision	F-score	MCC
SEPPA	0.48	0.16	0.24	0.14
DiscoTope	0.26	0.17	0.21	0.11
ElliPro	0.68	0.12	0.20	0.08
PUPre	0.71	0.18	0.28	0.21
F19_baggingMDT_plus3Dsphere[a]	0.13	0.53	0.21	0.23

[a]Bagging MDT used 19 base attributes and the derived 3D sphere-based attributes.

Table 9: Results of backward ablation analysis.

MDT*	TPR	FPR	PPV	ACC	F-score	MCC
F19_BaggingMDT_plus3Dsphere	0.463	0.011	0.791	0.947	0.584	0.581
\ SVM	0.483	0.024	0.633	0.936	0.548	0.520
\ Voted Perceptron	0.522	0.034	0.573	0.931	0.546	0.509
\ KNN	0.525	0.039	0.539	0.926	0.532	0.492
\ Random Forest	0.461	0.041	0.492	0.919	0.476	0.432
\ JRip	0.501	0.056	0.438	0.909	0.467	0.419
\ PART	0.394	0.031	0.524	0.923	0.450	0.414
\ C4.5 (Only Bayes Net)	0.538	0.178	0.208	0.800	0.300	0.241

*Classifiers tested in iterative backward ablation analysis. The first classifier in the first row is the MDT that employs all of the 8 base classifiers. The remaining classifiers are listed in the order in which they were selected to be removed iteratively from MDT for ablation study. '\' indicates "removed." For example, the second classifier is the MDT after SVM was removed, and the third classifier is the MDT after SVM and Voted Perceptron were removed from MDT. The MDT in the final row applied only Bayes Net after C4.5 was removed.

Table 10. Results of forward ablation analysis

MDT*	TPR	FPR	PPV	ACC	F-score	MCC
+SVM	0.368	0.006	0.847	0.944	0.513	0.536
+ Voted Perceptron	0.417	0.008	0.824	0.946	0.553	0.563
+ Random Forest	0.455	0.011	0.789	0.947	0.577	0.575
+ KNN	0.455	0.010	0.790	0.947	0.577	0.575
+ C4.5	0.466	0.011	0.789	0.948	0.586	0.582
+ PART	0.464	0.011	0.779	0.947	0.581	0.576
+ BayesNet	0.469	0.012	0.779	0.947	0.586	0.580
+ JRip (F19_BaggingMDT_plus3Dsphere)	0.463	0.011	0.791	0.947	0.584	0.581

*Classifiers tested in iterative forward ablation analysis. The first classifier in the first row is the MDT that employs only SVM. The remaining classifiers are listed in the order in which they were selected to be added iteratively to MDT for ablation study. '+' indicates "added." For example, the second classifier is the MDT after SVM and Voted Perceptron were added. The MDT in the final row applied all 8 base classifiers.

on meta-learning, which exploits the synergy among various prediction tools, demonstrated prediction performance superior to that of single epitope predictors. Nevertheless, the current meta-learning approach to epitope prediction depends heavily upon the predictive strength of other pretrained conformational and linear epitope predictors that users cannot retrain directly. This limits the applicability and flexibility of meta-classifiers. Here, we proposed a cost-sensitive bagging MDT approach, which combines two ensemble learning techniques with a cost-sensitive method. This method does not employ the predictions of any pretrained single epitope predictor, making it independent of multiple epitope prediction tools and capable of learning a meta-classification architecture from different given data, rather than restricting it through a prespecified and fixed hierarchy. This method applies the bagging mechanism to reduce the variance in the results of MDTs and considers the misclassification cost to adjust the final prediction and address the imbalanced class distribution in B-cell epitopes.

The structural characteristics of epitopes are different between bound and unbound states. Unlike previous studies of epitope prediction that conducted experiments mainly on bound-state structures [6, 7, 16, 34], we evaluated the performance of MDT for bound and unbound epitope prediction, respectively. While most of the epitope predictors have been trained from and tested on different bound-state antigens, to draw a fair comparison between MDT and other predictors for bound-state antigen prediction, instead of performing k-fold CV, we ran an independent test and also made individual comparisons. The results of the independent test and individual comparisons both demonstrated that our proposed meta-learning MDT approach outperformed the single base tools and other recently developed meta-learning epitope predictors for bound-state epitope prediction. By contrast, we conducted antigen-based 10-fold CV for unbound epitope prediction evaluation between MDT and other epitope predictors. The 10-fold CV was conducted on a recently constructed unbound structure data set [39]. It had not been used yet to pretrain any of the epitope predictors in comparison, and consequently, it was adopted to compare the performance for unbound epitope prediction. The results demonstrated the superior performance of MDT in comparison with three commonly used structure-based epitope predictors, and showed a marginally but noticeably higher performance than that of an unbound-state epitope predictor.

Acknowledgement

This work was partially supported by Ministry of Science and Technology (MOST) of Taiwan, MOST 106-2221-E-009-184.

References

1. Meloen RH, Puijk WC, Langeveld JP, Langedijk JP, Timmerman P. Design of synthetic peptides for diagnostics. Curr Protein Pept Sci. 2003; 4: 253-260.
2. Tanabe S. Epitope peptides and immunotherapy. Curr Protein Pept Sci. 2007; 8: 109-118.
3. Naz RK, Dabir P. Peptide vaccines against cancer, infectious diseases, and conception. Front Biosci. 2007; 12: 1833-1844.
4. Benjamin DC, Berzofsky JA, East IJ, Gurd FR, Hannum C, Leach SJ, et al. The antigenic structure of proteins: a reappraisal. Annu Rev Immunol. 1984; 2: 67-101.
5. Pellequer JL, Westhof E, Van Regenmortel MH. Predicting location of continuous epitopes in proteins from their primary structures. Methods Enzymol. 1991; 203: 176-201.
6. Andersen PH, Nielsen M, Lund O. Prediction of residues in discontinuous B-cell epitopes using protein 3D structures. Protein Sci. 2006; 15: 2558-2567.
7. Qi T, Qiu T, Zhang Q, Tang K, Fan Y, Qiu J, et al. SEPPA 2.0-more refined server to predict spatial epitope considering species of immune host and subcellular localization of protein antigen. Nucleic Acids Res. 2014; 42: W59-63.
8. Karplus PA, Schulz GE. Prediction of Chain Flexibility in Proteins – a Tool for the Selection of Peptide Antigens. Naturwissenschaften. 1985; 72: 212-213.
9. Rubinstein ND, Mayrose I, Martz E, Pupko T. Epitopia: a web-server for predicting B-cell epitopes. BMC Bioinformatics. 2009; 10: 287.
10. Zhang W, Liu J, Zhao M, Li Q. Predicting linear B-cell epitopes by using sequence-derived structural and physicochemical features. Int J Data Min Bioinform. 2012; 6: 557-569.

11. Liang S, Zheng D, Standley DM, Yao B, Zacharias M, Zhang C. EPSVR and EPMeta: prediction of antigenic epitopes using support vector regression and multiple server results. BMC Bioinformatics. 2010; 11: 381.

12. Zhang W, Niu Y, Xiong Y, Zhao M, Yu R, Liu J. Computational prediction of conformational B-cell epitopes from antigen primary structures by ensemble learning. PLoS One. 2012; 7: e43575.

13. Hu Y-J, Lin SC, Lin YL, Lin KH, You SN. A meta-learning approach for B-cell conformational epitope prediction. BMC Bioinformatics. 2014; 15: 378.

14. Todorovski L, Dzeroski S. Combining Multiple Models with Meta Decision Trees. Lecture Notes in Computer Science. 2002; 1910: 54-64.

15. Witten IH, Frank E. Data Mining: Practical Machine Learning Tools and Techniques. San Francisco: Morgan Kaufmann Publishers. 2nd edition. 2005. 560.

16. Kringelum JV, Lundegaard C, Lund O, Nielsen M. Reliable B Cell Epitope Predictions: Impacts of Method Development and Improved Benchmarking. PLoS Comput Biol. 2012; 8: e1002829.

17. Emini EA, Hughes JV, Perlow DS, Boger J. Induction of hepatitis A virus-neutralizing antibody by a virus-specific synthetic peptide. J Virol. 1985; 55: 836-839.

18. Pellequer JL, Westhof E, Van Regenmortel MH. Correlation between the location of antigenic sites and the prediction of turns in proteins. Immunol Lett. 1993; 36: 83-99.

19. Janin J, Wodak S, Levitt M, Maigret B. Conformation of amino acid side-chains in proteins. J Mol Biol. 1978; 125: 357-386.

20. Ponnuswamy PK, Prabhakaran M, Manavalan P. Hydrophobic Packing and Spatial Arrangement of Amino-Acid-Residues in Globular-Proteins. Biochimica Et Biophysica Acta. 1980; 623: 301-316.

21. Grantham R. Amino acid difference formula to help explain protein evolution. Science. 1974; 185: 862-864.

22. Ansari HR, Raghava GP. Identification of conformational B-cell Epitopes in an antigen from its primary sequence. Immunome Res. 2010; 6: 6.

23. Breiman L. Random forests. Machine Learning. 2001; 45: 5-32.

24. Chang CC, Lin CJ. LIBSVM: A Library for Support Vector Machines. ACM Trans Intell Syst Technol. 2011; 2: 27.

25. Quinlan JR. C4. 5: Programs for Machine Learning. San Francisco: Morgan Kaufmann Publishers; 1993.

26. Cover TM, Hart PE. Nearest neighbor pattern classification. IEEE Trans Inf Theory. 1967; 13: 21-27.

27. Frank E, Witten IH. Generating Accurate Rule Sets Without Global Optimization. Proceedings of the Fifteenth International Conference on Machine Learning. 1998; 144-151.

28. Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. San Francisco: Morgan Kaufmann Publishers; 1988.

29. Cohen WW. Fast Effective Rule Induction. Proceedings of the Fifteenth International Conference on Machine Learning. 1995; 115-123.

30. Freund Y, Schapire RF. Large margin classification using the perceptron algorithm. Machine Learning. 1999; 37: 277-296.

31. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. CRC press; 1984.

32. Breiman L. Bagging predictors. Machine Learning. 1996; 24: 123-140.

33. Zhang W, Xiong Y, Zhao M, Zou H, Ye XE, Liu J. Prediction of conformational B-cell epitopes from 3D structures by random forests with a distance-based feature. BMC Bioinformatics. 2011; 12: 341.

34. Ponomarenko J, Bui HH, Li W, Fusseder N, Bourne PE, Sette A, et al. ElliPro: a new structure-based tool for the prediction of antibody epitopes. BMC Bioinformatics. 2008; 9: 514.

35. Zhao L, Wong L, Li J. Antibody-specified B-cell epitope prediction in line with the principle of context-awareness. IEEE/ACM Trans Comput Biol Bioinform. 2011; 8: 1483-1494.

36. Schlessinger A, Ofran Y, Yachdav G, Rost B. Epitome: database of structure-inferred antigenic epitopes. Nucleic Acids Res. 2006; 34: D777-D780.

37. Ponomarenko, J, Papangelopoulos N, Zajonc DM, Peters B, Sette A, Bourne PE. IEDB-3D: structural data within the immune epitope database. Nucleic Acids Res. 2011; 39: D1164-D1170.

38. Ren J, Liu Q, Ellis J, Li J. Tertiary structure-based prediction of conformational B-cell epitopes through B factors. Bioinformatics. 2014; 30: 264-273.

39. Ren J, Liu Q, Ellis J, Li J. Positive-unlabeled learning for the prediction of conformational B-cell epitopes. BMC Bioinformatics. 2015; 16: S12.

40. Hubert L, Arabie P. Comparing partitions. J Classif. 1985; 2: 193-218.

41. Milligan GW, Cooper MC. A study of the comparability of external criteria for hierarchical cluster analysis. Multivariate Behavioral Research. 1986; 21: 441-458.