Chapter 12

Machine Learning-Based Methods for Prediction of Linear B-Cell Epitopes

Hsin-Wei Wang and Tun-Wen Pai

Abstract

B-cell epitope prediction facilitates immunologists in designing peptide-based vaccine, diagnostic test, disease prevention, treatment, and antibody production. In comparison with T-cell epitope prediction, the performance of variable length B-cell epitope prediction is still yet to be satisfied. Fortunately, due to increasingly available verified epitope databases, bioinformaticians could adopt machine learning-based algorithms on all curated data to design an improved prediction tool for biomedical researchers. Here, we have reviewed related epitope prediction papers, especially those for linear B-cell epitope prediction. It should be noticed that a combination of selected propensity scales and statistics of epitope residues with machine learning-based tools formulated a general way for constructing linear B-cell epitope prediction systems. It is also observed from most of the comparison results that the kernel method of support vector machine (SVM) classifier outperformed other machine learning-based approaches. Hence, in this chapter, except reviewing recently published papers, we have introduced the fundamentals of B-cell epitope and SVM techniques. In addition, an example of linear B-cell prediction system based on physicochemical features and amino acid combinations is illustrated in details.

Key words B-cell epitope, Machine learning, Support vector machine, Propensity scale, Kernel function

1 Introduction of B-Cell Epitopes

The immune system is a collection of organs, tissues, cells, and molecules that work together to protect the body from various foreign pathogens such as bacteria, viruses, parasites, and fungi. This defense system against pathogens has been divided into two main strategies in vertebrates: innate immunity and adaptive immunity mechanisms. The innate immune system is considered as the first defending process against invading pathogens, while the adaptive immune system of the second defending layer creates immunological memories after an initial response to a specific pathogen and induces an enhanced response to subsequent encounters regarding the same pathogen. The latter adaptive immunity is classified into two branches of immune responses including cellular

Rajat K. De and Namrata Tomar (eds.), *Immunoinformatics*, Methods in Molecular Biology, vol. 1184,
DOI 10.1007/978-1-4939-1115-8_12, © Springer Science+Business Media New York 2014

immunity mediated by T-cell lymphocytes that eliminate infected cells and humoral immunity mediated by B-cell lymphocytes secreting antibodies which neutralize pathogens in the body fluid. Epitopes or antigenic determinants are defined as clusters of amino acid segments located on the surface of an antigen that bind to antigen-specific membrane receptors on lymphocytes or to secreted antibodies, and which elicit either cellular or humoral immune response and are recognized by specific antibodies [1]. Due to expensive and time-consuming factors of biomedical and immunological experiments, in silico epitope prediction and analysis prior to biological experiments become practical and standard strategies for both biomedical researchers and immunologists regarding various immunology-related applications such as epitope-based vaccine design and disease prevention, diagnosis, and treatment. There are several good review articles for both T-cell and B-cell epitope prediction analysis based on computational approaches as well as several useful epitope databases [2–8]. Among all published papers, epitope prediction methods can be simply categorized into four major types: sequence-based, structure-based, hybrid of sequence-based and structure-based, and consensus methods. It is in general expected that the prediction accuracy could be improved if an antigen structure has been determined. This is mainly due to easy validation of the surface characteristics of candidate epitopes on an antigen from the resolved structure. Hence, combination of sequence and structure features simultaneously should provide better prediction results than using sequence-based or structure-based along methods. Furthermore, combining several prediction methods and summarizing all individual prediction result through a voting mechanism could be anticipated to achieve an even better prediction accuracy since each prediction method held its own strength. Nevertheless, due to limited numbers of determined antibody–antigen complex structures and integrating difficulties for various computational limitations, there is yet no such a successfully integrated system for both B-cell and T-cell epitope prediction. Most of the prediction systems still focus on identifying one specific type of epitope according to its own characteristics.

T-cell epitopes are defined as peptide sequences presented on the surface of an antigen-presenting cell, and they are bound to major histocompatibility complex (MHC) class I and II molecules. Known as a structural basis for peptide binding to MHC molecules, T-cell epitopes are typically composed by continuous amino acids ranging from 9 to 11 in length for MHC class I binding and a length ranging from 13 to 25 amino acids for MHC class II binding [9, 10, 4]. For B-cell epitopes, it is generally categorized into two types: linear epitope (LE), a segment composed of a continuous stretch of amino acid residues, and conformational epitope (CE) constituted by several sequentially discontinuous segments that are dispersed among discontinuous regions, but become

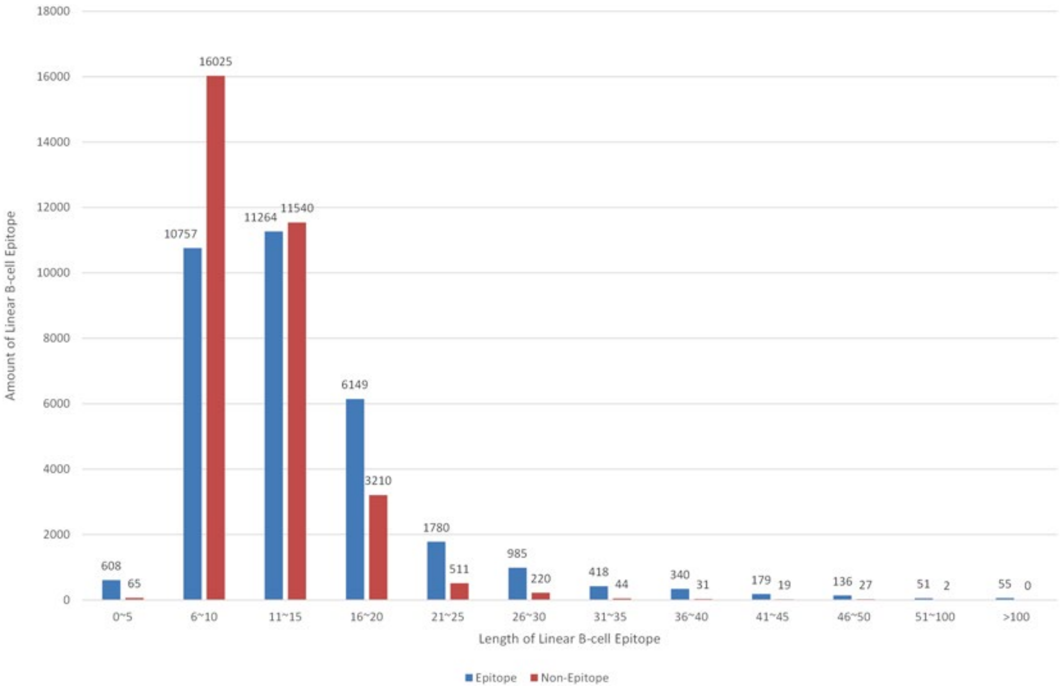

Fig. 1 Length distribution of linear B-cell epitopes and non-epitopes collected from IEDB database (version 2.4)

aggregated on the protein surface [11, 12]. Compared to continuous T-cell epitopes, linear B-cell epitopes possess significantly various peptide lengths from 2 to 829 residues from verified LE data statistics (IEDB: http://www.eiedb.org/) [13]. Length distribution of verified linear B-cell epitopes from IEDB database is shown in Fig. 1. Near 95 % of verified linear B-cell epitopes possess flexible lengths ranging from 6 to 30 residues. Even several annotated epitopes are with lengths larger than 100 residues. It was also reported that the proportion of LEs is considered with only 10 % of all B-cell epitopes [11], while the majority of B-cell epitope belongs to the discontinuous CE type with epitope size ranging from 6 to 29 residues [14]. However, in contrast to less complex features of T-cell epitope prediction systems and superior achievement for T-cell epitope prediction, the performance of predicting B-cell epitopes is yet to be satisfied and all proposed approaches still face a lot of challenges in computational immunology. Besides, only a small set of verified CEs are curated, a small set of resolved antibody–antigen complex structures, and not many convincible CE prediction systems are available. Therefore, in this chapter, we mainly discuss most of the published linear B-cell epitope prediction methods, and demonstrate how to adopt machine learning-based approaches for linear B-cell epitope prediction. It is also noticed that the support vector machine (SVM)-based learning method is one of the most popular approaches in recent reports.

In addition, the SVM-based system provided better performance compared to other machine learning methods. To demonstrate the usage of in silico prediction on linear B-cell epitopes through machine learning approaches, we choose to introduce the SVM classifier and hope that readers can fully understand the complete procedures and fundamental knowledge of linear B-cell epitope prediction.

Currently, various computational approaches and software for linear B-cell epitope prediction have been boomingly proposed in the last decade. Table 1 shows available methods, applicable websites, and kernel methods applied for LE prediction in a chronological order.

Most of the LE prediction focused on sequence contents and their corresponding propensity scales including surface accessibility [35], hydrophilicity [36], flexibility [37], and secondary structure [38] have been heavily considered in epitope predictive algorithms. The distinguishing characteristics among currently available programs such as BEPITOPE [17], PEOPLE [16], and BcePred [18] are mainly dealing with computation of different weighting scales over a sliding window along a query protein sequence. However, Blythe and Flower hypothesized that "single-scale amino acid propensity profiles cannot be used to predict epitope locations reliably" [39], a conclusion based on the observation that in the field of epitope prediction, even the best combinations of physicochemical propensity scales were not accurate enough to estimate and predict qualified B-cell epitopes. Therefore, several methods integrating the concept of amino acid propensity scales with machine learning technologies were proposed. For example, Saha and Raghava used recurrent artificial neural networks based on amino acid sequence information in ABCPred [19]; Larsen employed hidden Markov model (HMM) in BepiPred [20]; Chen et al. adopted SVM classifier on amino acid pairs [22]; Söllner and Mayer utilized a molecular operating environment with the decision tree and nearest neighbor approaches [21]; El-Manzalawy et al. developed BCPred [23] and FBCPred [24] employing SVM with a subsequence kernel for both fixed and flexible length epitopes; Sweredoski and Baldi developed COBEpro [26]; Wang et al. designed LEPS [30]; and Gao et al. presented BEST [31]; the last three approaches applied an SVM classifier in a two-step system to predict LEs based on an improved propensity scale approach; similarly, the BEEPro system designed by Lin et al. [33] and the LBtope system provided by Singh et al. [34] also adopted SVM classifiers by combining different propensity scales to enhance the prediction accuracies.

In the ABCPred system, two artificial neural network methods were developed, feed-forward (FNN) and recurrent neural network (RNN), for the prediction of continuous B-cell epitopes. Both FNN and RNN networks were used to achieve B-cell epitope prediction

Table 1
Linear B-cell epitope prediction methods

Name	URL	Method	Year	Reference
Antigenic	http://www.emboss.bioinformatics.nl/cgi-bin/emboss/antigenic	Physicochemical properties, occurrence of amino acid residues	1990	[15]
PEOPLE	n/a	Physicochemical properties	1999	[16]
BEPITOPE	Stand-alone program can be obtained freely to academics jlpellequer@cea.fr	Physicochemical properties	2003	[17]
BcePred	http://www.imtech.res.in/raghava/bcepred/	Physico-chemical properties	2004	[18]
ABCpred	http://www.imtech.res.in/raghava/abcpred/	ANN	2006	[19]
BepiPred	http://www.cbs.dtu.dk/services/BepiPred/	HMM	2006	[20]
Söllner	n/a	MOE, KNN, Decision tree	2006	[21]
Chen	n/a	SVM, AAP	2007	[22]
BCPred	http://www.ailab.cs.iastate.edu/bcpreds/	SVM, String kernel	2008	[23]
FBCPred	http://www.ailab.cs.iastate.edu/bcpreds/	SVM, String kernel	2008	[24]
LEPD	http://www.lepd.cs.ntou.edu.tw/	Physicochemical properties, mathematical morphology	2008	[25]
COBEpro	http://www.ics.uci.edu/~baldig/scratch/index.html	SVM	2009	[26]
Epitopia	http://epitopia.tau.ac.il	Naïve Bayes classifier	2009	[27, 28]
BayesB	http://www.immunopred.org/bayesb/index.html	SVM, Bayes feature extraction	2010	[29]
LEPS	http://leps.cs.ntou.edu.tw/	Physicochemical properties, mathematical morphology, SVM	2011	[30]
BEST	http://biomine.ece.ualberta.ca/BEST/	SVM	2012	[31]
SVMTriP	http://sysbio.unl.edu/SVMTriP/	SVM, tripeptide similarity and propensity	2012	[32]
BEEPro	n/a	Physicochemical properties, SVM, PSSM	2013	[33]
LBtope	http://crdd.osdd.net/raghava/lbtope/	SVM, binary profile, dipeptide composition, AAP	2013	[34]

ANN artificial neural network, *HMM* hidden Markov model, *MOE* molecular operating environment, *KNN* k-nearest neighbor, *PSSM* position-specific scoring matrix, *n/a* not applicable using different window lengths from 10 to 20 amino acids, and the best performance of 66 % accuracy evaluated on a dataset of 700 B-cell epitopes and 700 non-epitopes was obtained by adopting an RNN trained on peptides of 16 amino acids in length. The BepiPred combined two amino acid propensity scales and an HMM trained on LEs to gain a slightly improved prediction accuracy rate over the propensity scale only-based methods by Parker et al. and Levitt et al. on the Pellequer dataset of 14 proteins and 83 epitopes. In Chen's approach, the observed certain amino acid pairs (AAPs) tend to appear more frequently in known B-cell epitopes than in non-epitope peptides. They utilized an AAP propensity scale based on such observation and trained with an SVM classier to increase an improved prediction accuracy rate of 71 % from the datasets of 872 B-cell epitopes and 872 non-epitopes. In the method of Söllner and Mayer, each epitope is represented using a set of propensity scales, neighborhood matrices, and respective probability and likelihood values. This approach combined several parameters previously associated with antigenicity, and included novel parameters based on frequencies of amino acids and amino acid neighborhood propensities. In their report, the best performance of 72 % was achieved utilizing a nearest-neighbor classifier with feature selection from datasets of 1,211 B-cell epitopes and 1,211 non-epitopes. For the BCPred developed by El-Manzalawy et al., they applied five different kernel methods to evaluate SVM classifiers on a homology-reduced dataset of 701 linear B-cell epitopes and 701 non-epitopes, and they demonstrated that the BCPred outperformed the ABCPred and Chen's methods. In addition to BCPred, El-Manzalawy et al. also developed another FBCPred for predicting flexible length linear B-cell epitopes using the subsequence kernels. Two machine learning approaches were adopted in their study: one approach utilized four sequence kernels for determining a similarity score between any arbitrary pair of variable length sequences, and the other approach applied four different methods of mapping a variable length sequence into a fixed length feature vector. The FBCPred was demonstrated with an improved performance of 73 % accuracy rate on the homology-reduced dataset of flexible length linear B-cell epitopes. In the COBEpro system, Sweredoski applied SVM to make predictions on short peptide fragments within the query antigen sequence and calculated an epitopic propensity score for each residue based on the fragment predictions. The accuracy rates and AUC values of COBEpro possessed better performance than Chen, BCpred, and BepiPred regarding different benchmark datasets. The LEPS system designed by Wang et al. combined improved propensity scale method, local high antigenicity profile, occurring frequencies of amino acid segments (AASs), and SVM classifier to predict LEs with flexible length. Using several benchmark datasets, LEPS has shown its competitive performance comparing to BepiPred, ABCPred, BCPred, and FBCPred. For the BEEPro developed by Lin et al., authors have claimed that both linear and conformational epitopes could be predicted by the SVM-based system which employed the features mainly based on evolutionary information, amino acid ratio propensity scale, and 14 specifically selected physicochemical propensity scales. The results have shown a superior performance compared to BepiPred, ABCPred, BCPred, FBCPred, and LEPS. For the BEST system presented by Gao et al., authors constructed an SVM training architecture based on features of averaging selected propensity scores by a 20-mer sliding window, sequence similarity score, predicted secondary structure, and solvent accessibility. The prediction performance was compared to Chen, BCPred, COBEpro, BayseB, and CBTOPE with an accuracy rate around 74 % for fragment-based LE prediction. For the latest LBtope system, authors provided five various training datasets, and they emphasized on experimentally verified non-epitope datasets compared to previously random peptides used in other studies. In this study, they applied SVM and K-nearest-neighbor learning models using various physicochemical propensity scales and amino acid composition-transition-distribution properties, and the LBtope prediction system obtained accuracy rates ranged from 54 to 86 % on the created datasets. Since most of machine learning-based approaches applied SVM classifiers to improve the performance of B-cell epitope prediction and the results showed that SVM-based methods possessed a better performance than other approaches, here we will briefly introduce basic theories of SVM in the next section for readers interested in related fields. An example of prediction system will also be applied to illustrate the combination of propensity scales and machine learning kernel method for LE prediction.

2 A Supervised Learning Method: SVM Classifier

Machine learning is a subfield of applied statistics, which trains on a collected sample dataset and generalizes rules from previous experiences. The training data with unknown probability distribution is usually applied to extract some general principles and perhaps the distribution for future predictions on new testing data. There are several types of machine learning algorithm based on trained inputs or desired outcomes, such as supervised, unsupervised, semi-supervised, and reinforcement learning mechanisms. Recently, one of the most popular computer algorithms for a variety of biological applications including epitope prediction is the SVM kernel method, a supervised learning model and learned by known epitope contents to predict novel epitopes within a query protein sequence [40]. To build an epitope prediction model, users have to provide a set of training examples including two classes, named as true epitopes and non-epitopes. The constructed SVM model is a representation of the trained examples as points in the selected feature space, and these sample points are divided by a hyperplane with a separable margin as wide as possible.

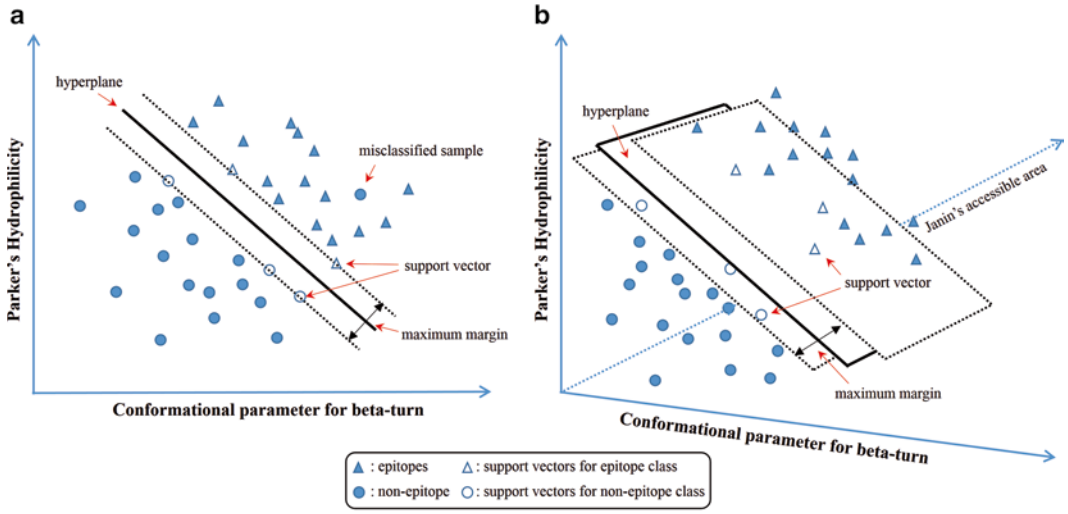

Fig. 2 Two examples of two-class SVM classifiers. (**a**) The first example of two-dimensional feature space and two classes were separated by a straight line with the maximum margin. (**b**) The second example of adding one more feature to a three-dimensional feature space and the two class samples were separated by a hyperplane with maximum margin. Each *circle* and *triangle* element represents samples from two different classes, and *empty circle* and *triangle* objects represent the support vectors for each class. The hyperplane was defined with a maximum margin between two planes constructed by support vectors

Query protein sequence segments are then mapped into the same feature space and assigned to one of the two defined categories based on the locations of the testing segment.

2.1 The Hyperplane of an SVM Model

Figure 2a shows a simple example of mapped points in a two-dimensional feature space. In this example, it is assumed that each peptide was calculated and mapped into a corresponding feature point by two selected feature values: secondary structure (conformation parameter for beta turn) and hydrophilicity (Parker's parameters [36]). The feature profile of each known epitope or non-epitope peptide is calculated according to the residue contents and the feature values are mapped into the two-dimensional space and represented by triangle and circle objects, respectively. In this case, it is quite easy to draw a line between two clusters geometrically, and an unknown data point could be predicted easily according to the query feature point falling on the epitope or the non-epitope sides of this separating line. If we add one more different feature such as Janin's accessible area to classify a peptide into two clusters, the feature space becomes a three-dimensional space, and we need a plane to divide the space into two parts as shown in Fig. 2b. Definitely, similar procedures could be extended to higher dimensions by adding more features. Hence, the original straight line in two-feature space can be extended to a hyperplane in a higher dimensional space which represents the border line to separate two clusters.

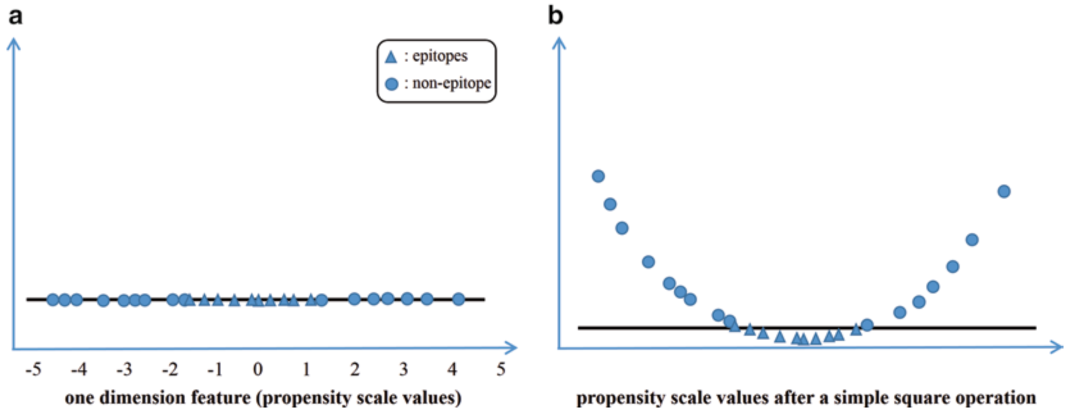

Fig. 3 An example of applying degree-2 polynomial kernel on all data points. (**a**) One-dimensional feature space and hard to find a single line to separate all data into two classes. (**b**) Applying square operation on all data points, and a clear hyperplane could separate all data points into two classes

2.2 Maximum Margins of a Hyperplane

It is obvious that the hyperplanes are not unique in an SVM model. How to select an optimal hyperplane between two clusters is the main goal of adopting SVM predictor and it serves as the key factor of a successful SVM classifier. Based on general statistical assumptions and the definition of a margin as the distance between the hyperplane to the nearest points (support vectors) within one cluster, the SVM model could find an optimal hyperplane possessing the maximal margin from any one of the training data points within two clusters. Hence, the selected hyperplane could maximize the performance of the SVM classifier to predict query samples. Nevertheless, several outlier data points might reside in wrong clusters from real applications and are called misclassified samples, and it might be solved by introducing an ε-insensitive loss function [41] which balances the number of hyperplane violations and the size of the margin.

2.3 Selection of Kernel Functions

Sometimes a tolerant margin could not support to find an optimal hyperplane to separate two clusters since the data points are crossly distributed in a feature space. In that case, there might exist a kernel function which provides a solution by adding an additional dimension for the data points. The original points could be transferred by a kernel function in order to find a better hyperplane to separate two clusters in a higher dimensional feature space. For an example shown in Fig. 3, the one-dimensional feature points could apply a simple square operation to transfer all data points into a two-dimensional space, and therefore an optimal hyperplane could be observed clearly. There are several frequently applied standard kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid which can help to transfer the data points into a higher dimension to find a better hyperplane [42, 43]. However, it should be noticed that a very high-dimensional kernel function may cause overfitting problems and generally lead poor predictive performance. To avoid too many irrelevant dimensions, the selection of types and degrees of kernel functions should be carefully considered. Nevertheless, the traditional way to find a better kernel function is usually achieved by a trial-and-error approach and verified by cross-validation processes. But the selected so-called best kernel function still does not guarantee the optimal performance.

3 A Practical Example of Predicting Liner Epitope Based on SVM Classifier

To demonstrate how to apply SVM in predicting LEs, we selected the features used by Wang et al. [30], including physicochemical and AAS propensity scales. The first step is to discover all segments with global high or local high antigenicities according to the corresponding physicochemical properties. Once the potential segments were identified, the frequently appeared AASs were evaluated according to previously identified LE candidates. Based on the SVM method and the constructed models, the potential candidate segments were classified into epitopes or non-epitopes. Here we go through more details and learn how to apply machine learning technology intuitively in the application of LE prediction.

3.1 Antigenicity Analysis

An antigenic peptide possesses physicochemical properties of hydrophilicity, polarity, charge, flexibility, accessibility, secondary structure, and some other miscellaneous factors. For each category of specific physicochemical property, the individual score was given by sliding a window of a specified length along the query protein sequence from the *N* to *C* terminal direction and applying respective assigned weighting coefficients to each residue. The mean value of the assigned physicochemical feature within a sliding window was then calculated, and the average value was considered as a representative score at the midpoint of the window [44]. The boundary problems will be faced at both the *N*- and *C*-termini since the length of the neighboring residues was not sufficient to be considered within a fixed sliding window size. Hence, only the covered neighboring amino acids were applied to calculate the antigenicity value. Once an individual scale for each physicochemical feature was determined, a combination of different weighted coefficients on various scales at each position was calculated to achieve a final antigenicity profile. Different weighting coefficient assignment definitely affects the final antigenicity profile at a certain level. Users could assign the weightings according to his/her special concerns or simply apply equally distributed weightings. Here we applied different weighting coefficients to enhance and distinguish the importance of antigenic features with respect to LE prediction. In this example, we applied the beta turn [45], hydrophilicity [46], flexibility [47], and surface accessibility [48] with weighting coefficients of 0.4, 0.3, 0.15, and 0.15, respectively [16] (*see* **Note 1**).

3.2 Mathematical Morphology and Local Peak Determination

Most of the LE prediction systems focused on identifying the global high antigenicity segments. However, Wang et al. found that some of the experimental verified epitopes are located within the local high antigenicity profile. They applied some filtering processes to identify segments with global or local high antigenicity as epitope candidates. Their proposed filtering processes were completed employing mathematical morphology algorithm which is a nonlinear filter for signal analysis built on lattice theory and topology with applications to one-, two-, or n-dimensional signals. An antigenicity profile was interacted with a predetermined structuring element under three basic operations: erosion, dilation, and opening. Details of operating descriptions could be referred to refs. [49–52]. Nevertheless, the segments with local high or global high antigenicity were detected and extracted for next processes. It should be noticed that the default settings of window size for calculation of antigenicity scale, extraction of local peaks, and filtering of minimal size of epitope candidates played an important role at the initial stage. These default window sizes were selected according to the optimal performance in terms of accuracy analysis from known datasets [53]. The global antigenicity was defined as the average of the whole protein sequence antigenicity, and the low-to-moderate antigenicity meant the antigenicity of a predicted peptide lower than that of global antigenicity. Once the antigenic scale of each amino acid was calculated applying a running mean window by default settings, all epitope candidates were extracted when the average antigenicity of residues was continuously higher than that of the entire sequence or when the residues were located within peptides with locally high antigenicity compared to their neighboring segments. For fragments with globally or locally high antigenic residues, a merging function was performed to identify the candidates of LEs. All extracted segments with either globally or locally high antigenicity scales would be further filtered by the next classifier according to the SVM learning model based on previously statistical features. One example for extracting all possible epitope candidates is shown in Fig. 4. The figure shows a set of identified epitope candidates by mathematical morphology approaches on the *P30* protein. The original antigenicity profile according to the default weighting coefficient settings was shown in Fig. 4a. Eroded antigenicity profile by an erosion operator was shown in Fig. 4b, and a following dilation filter was applied on the previously eroded profile and an opened antigenicity profile was obtained and shown in Fig. 4c. All local peaks in Fig. 4d could be detected by taking the difference between the original and opened antigenicity profiles at the corresponding positions. These local peak segments were further filtered by a scanning window and the filtered segments were regarded as initially predicted candidates as shown in Fig. 4e. Finally, the predicted candidate LEs were obtained according to the locally high antigenic characteristics as shown in Fig. 4f.

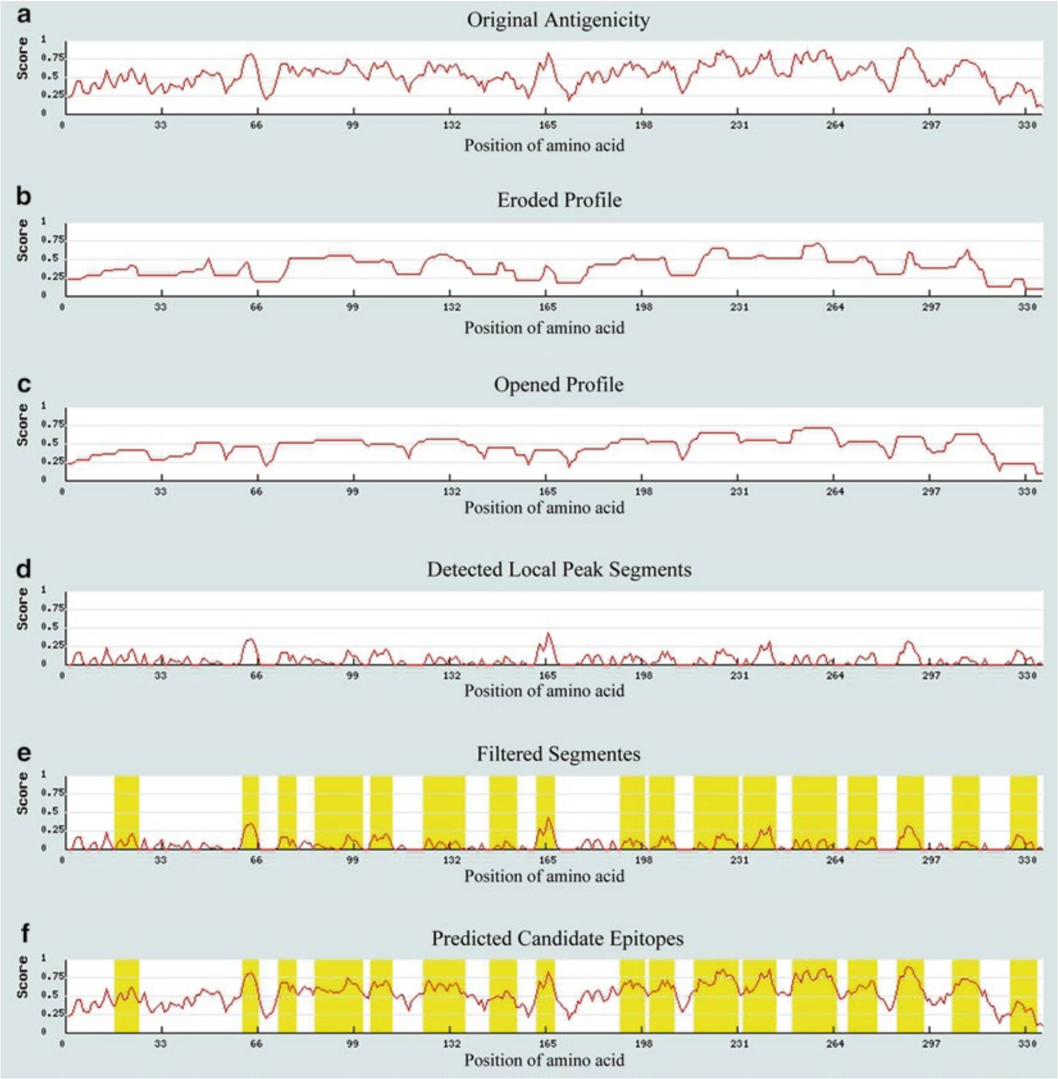

Fig. 4 An example of applying combination of morphological filters to extract segments with globally or locally high antigenicity characteristics on the *P30* protein. (**a**) The original antigenicity profile for *P30* protein. (**b**) Eroded antigenicity profile by an erosion operator. (**c**) Followed by a dilation filter and an opened antigenicity profile was obtained. (**d**) All local peaks detected by taking the difference between (**a**) and (**c**) at the corresponding positions. (**e**) Filtering local peaks with a default scanning window and the highlighted segments were considered as the candidate epitope locations. (**f**) All predicted candidate LEs for *P30* protein based on selected physicochemical propensity scales

3.3 SVM Classifier on Candidate Epitopes

The processes of adopting an SVM machine learning-based approach for epitope prediction usually comprise two major stages. The first stage requires a collection of training datasets and selected features. The training dataset includes samples in two categories: positive samples (true epitope segments) and negative samples (non-epitope segments). These samples will be trained to construct an SVM model according to the selected feature set. It should be noticed that a successful collection of training samples leads to good performance for all machine learning-based classifiers. In other words, lacking verified knowledge for collecting either positive or negative class samples affects the performance of the classifier dramatically and yields a biased estimation on evaluating system performance. As we know, most of the machine learning-based classification applications in biological fields fall short in negative samples. To balance the collection of both positive and negative class training samples, sometimes the generation of artificial negative samples is required [54]. Here, we adopted the Chen's dataset [22] containing 872 epitopes and 872 non-epitopes, for training the SVM classifier. All epitopes and non-epitopes within this dataset were restricted to a length of 20 residues. For the feature selection problem, since the physicochemical properties were already considered in the previous epitope candidate selection, here we only choose the amino acid combination propensity scales as the training features. We evaluated the statistical characteristics that determined the frequencies of occurrence of AASs with various lengths from another B-cell LE dataset, Bcipep [55], and the Chen's non-epitope dataset. Next, an SVM model was built based on the statistical features of the epitopes and non-epitopes. It should be noticed that the requirement of fixed window size for training and prediction sometimes considered as a deficiency in the machine learning-based approaches. Here, all collected epitopes and non-epitopes within the training dataset were restricted to a length of 20 residues. These verified epitopes were retrieved employing a "truncation-extension treatment." That is, when the length of an LE was longer than 20 residues, an equal number of superfluous residues were truncated from both the *N*- and *C*-termini to preserve the central 20 residues. Conversely, when the length of an LE was shorter than 20 residues, an equal number of neighboring residues were added to both the *N*- and *C*-termini according to its original sequences until the epitope comprised 20 residues. Both epitopes and non-epitopes with fixed length were then used to analyze their corresponding features and trained to produce an SVM model for future prediction.

3.4 Statistical Analysis of Amino Acid Segments and Corresponding Epitope Indexes

For constructing an SVM model in this example, we simply considered three statistical features by calculating the occurrence frequencies of combined residues in different lengths for both epitopes and non-epitopes. For the first feature of amino acid segment with two residues (AAS^2), 400 possible combinations of residue pairs should be analyzed for their corresponding occurrence frequencies in both the collected epitope and non-epitope segments. The epitope index Epidex$_i^2$ of the *i*th pattern (AAS$_i^2$) is defined by taking logarithmic value of the ratio of the number of AAS$_i^2$ among all epitopes AASs^2 compared to the same ratio in the non-epitope AASs^2 group. It can be formulated as the following equation:

$$\mathrm{Epidex}_i^2 = \log\left(\frac{f_i^{2^+} / \sum_i f_i^{2^+}}{f_i^{2^-} / \sum_i f_i^{2^-}}\right), \qquad i = 1, 2, \ldots, 400$$

where f_i^{2+} and f_i^{2-} are the numbers of AAS$_i^2$ in the epitope and non-epitope datasets, respectively, and $\sum_i f_i^{2+}$ and $\sum_i f_i^{2-}$ denote the total number of AAS$_i^2$ in the corresponding dataset. Finally, the values of Epidex$_i^2$ are normalized to the range of [0, 1] to avoid dominance of any individual Epidex$_i^2$ in the classifier learning processes. For the next two features of amino acid segments with three and four residues (AAS^3 and AAS^4), there are a total of 8,000 and 160,000 possible combinations, respectively. In this case, a large portion of AAS^3 or AAS^4 do not appear in the non-epitope dataset and it would cause a problem of dividing by zero. Hence, the definitions of Epidex$_i^3$ and Epidex$_i^4$ are modified from the definition of Epidex$_i^2$, and the corresponding epitope indices for AAS^3 and AAS^4 are defined as the following formula. Both obtained Epidex$_i^3$ and Epidex$_i^4$ will be normalized to the range of [0, 1] as well:

$$\mathrm{Epidex}_i^l = \frac{f_i^{l^+}}{\sum_i f_i^{l^+}}, \qquad l = 3 \text{ or } 4.$$

3.5 SVM Features and Kernel Selection

There are a variety of choices of open-source SVM software for feature training, model selection, and cross validation [56]. Users are able to select a suitable SVM tool based on their own requirements. Here one of the most popular open-source toolboxes, LIBSVM (Library for Support Vector Machines) developed by Chang and Lin [42], is adopted to demonstrate the application on LE prediction. In LIBSVM, each instance in the training set possessed one target value (class label) and several features (attributes). In the testing set, only the features are required for each instance. The objective of SVM is to generate a model from the training set that facilitated the prediction of the target value of each instance in the testing set. A peptide corresponded to an instance and the target value (1 or −1) represents whether that peptide is an epitope. Each peptide contains three feature values including Epidex$_i^2$, Epidex$_i^3$, and Epidex$_i^4$. For example, a 20-mer peptide is decomposed into 19 AAS$_i^2$ subsegments, and the corresponding epitope index of this peptide is obtained by taking the average of 19 Epidex$_i^2$ from the corresponding AAS$_i^2$. Similarly, the feature values of Epidex$_i^3$ and Epidex$_i^4$ can be obtained by calculating the averages of 18 Epidex$_i^3$ and 17 Epidex$_i^4$ subsegments, respectively.

As previously described, sometimes the sample data points are crossly distributed in a feature space and cannot be separated by a linear hyperplane. In that case, a kernel function transformation might be able to provide a solution by adding an additional dimension for sample points. However, there is no straightforward decision or theoretical methods to decide what kind of kernel functions provides the best results for a given dataset; trial and error on experimenting with different kernel functions is the only way to find the best function. In this example, the experimental dataset was used to construct an SVM model based on three feature values and the target values of each epitope and non-epitope. Four common kernel functions including linear, polynomial, RBF, and sigmoid were provided by LIBSVM. We examined all these four kernel functions with a fivefold cross-validation (*see* **Note 2**). The training dataset was equally divided into five different subsets; four of the subsets were used for training the model and the last one was used for testing the model. These processes were repeated five times with each individual subset used as the testing subset. Based on the cross-validation results in this case, the RBF kernel function provided the best performance regarding the collected samples and it was selected as the default kernel function. Subsequently, the RBF kernel function was applied to train the whole collected positive and negative datasets again and construct the final SVM classifier for future LE prediction.

3.6 Performance Measurement

To evaluate the performance of an epitope prediction system, either peptide- or residue-level evaluation could be applied according to the characteristics of prediction system and testing databases. For example, several epitope/non-epitope datasets provided by LBtope only contain fragments of antigen proteins and are required to be verified as LEs or not. Definitely, the peptide-level evaluation will be an appropriate selection in this application. However, if a whole-antigen protein sequence was considered as the query data and the prediction system could provide flexible length LE candidates, then a residue-level evaluation method is more suitable. Therefore, residue-level evaluation method was applied to the LEPS prediction system. There are five commonly used indicators for measuring effectiveness of a prediction system, which include (1) *sensitivity* (*SEN*), defined as the percentage of epitopes that are correctly predicted as epitopes; (2) *specificity* (*SPE*), defined as the percentage of non-epitopes that are correctly predicted as non-epitopes; (3) *positive predictive value* (*PPV*), defined as the probability that a predicted epitope is an epitope; (4) *accuracy* (*ACC*), defined as the proportion of correctly predicted peptides; and (5) *Matthews correlation coefficient* (*MCC*), which is a measure of the predictive performance incorporating both SEN and SPE into a single value between −1 and +1. A merged and non-redundant testing dataset called AHP dataset was created by Wang et al. from AntiJen, HIV, and PC datasets, which contained 193 proteins with 843 non-overlapping epitopes [30]. These three datasets were selected to balance the variations in each dataset including variations in epitope length and the physicochemical properties of antigens. It should be noticed that all antigen proteins selected in testing dataset must be different from the training dataset and all repeated proteins should be removed in advance. In this example the SVM-based learning system could achieve a performance of SEN of 27.0 %, SPE of 84.2 %, ACC of 72.5 %, PPV of 32.1 %, and MCC of 10.4 %. One point should be mentioned here: The PPV indicated the rate of identifying real epitopes among all positive predicted candidates, and it is one of the most important factors for immunologists in conducting vaccine development. Reduction of the false-positive candidates can significantly improve the effectiveness and efficiency of identifying the real epitopes. Compared to other systems, the LEPS also showed its excellent performance for all different testing datasets. All the comparison details can be referred to Wang et al. [30].

4 Conclusion

In silico linear B-cell epitope prediction is definitely an important procedure for designing peptide-based vaccine, diagnostic test, disease prevention, treatment, antibody production, and other related applications. Successful prediction facilitates biomedical researchers and immunologists in reduction of experimental time and overall costs. In this chapter, current linear B-cell epitope prediction methods, collected databases, and available online systems based on machine learning techniques are comprehensively reviewed. Especially, one of the most applied machine learning methods, the support vector machine classifier, is also briefly introduced for non-computer-background readers. To demonstrate the usage of combining popularly used propensity scales and machine learning techniques, an LE prediction system proposed by Wang et al. was also introduced through step-by-step description. We hope that the details can help beginners to find some important materials, to clarify some fundamental questions, and to gain a better understanding of applying machine learning approaches on epitope prediction. Though research on epitope analysis and related prediction systems were booming in the last two decades, the performance on B-cell epitope prediction is yet to be satisfied comparing to T-cell epitope prediction. This is mainly due to the complexity of B-cell epitope binding mechanisms, variable lengths of B-cell epitopes, and limited availability of resolved antigen and antibody–antigen complex structures. In addition, still several other existing problems are waiting for solutions to improve recent prediction tools. The first problem is the deficiency of a comprehensive learning dataset containing both verified epitope and verified non-epitope peptides. Especially the verified non-epitope dataset plays an important role to improve prediction accuracy. In general, many trained non-epitope samples were generated by artificially random approaches which might lead to a wrong and biased learning model. Except the verified epitopes on both positive and negative classes, the total number of non-redundant epitopes should be large enough for reliable training and similar sequences should be avoided for appearing in both training and testing datasets. It must be very careful to remove redundant and high similarity sequences from a testing dataset comparing to the training dataset. Overly optimistic performance may be obtained if similar protein sequences appeared in both datasets. Most importantly, if more three-dimensional antigen structures or antigen–antibody complexes could be crystallized and determined, the binding mechanisms between antibodies and antigens could be understood in more details. Hence, it might be possible to categorize B-cell epitopes into several different interaction mechanisms and various levels of immunogenic potency. The machine learning approaches could also be advanced from a two-class to a multiple-class classifier. With all these considerations, the integrated prediction system based on sequence and structural features could become more reliable and practical.

5 Notes

1. Since the original propensity scores are in different scales, a normalization procedure needs to be performed before combining each antigenicity score. The final antigenicity scores for each reside therefore appear within a range of [0, 1].
2. In addition to the selection of kernel functions, several parameters for each kernel function are required to be identified. LIBSVM provides a simple parameter selection tool based on grid-search approach to try different combinations of parameters heuristically.

Acknowledgements

This work is supported by the Center of Excellence for the Oceans, National Taiwan Ocean University, and National Science Council, Taiwan, R.O.C. (NSC 102-2321-B-019-001 and NSC 102-2221-E-019-059 to T.-W. Pai).

References

1. Davies DR, Cohen GH (1996) Interactions of protein antigens with antibodies. Proc Natl Acad Sci U S A 93(1):7–12
2. Korber B, LaBute M, Yusim K (2006) Immunoinformatics comes of age. PLoS Comput Biol 2(6):e71. doi:10.1371/journal.pcbi.0020071
3. Greenbaum JA, Andersen PH, Blythe M, Bui HH, Cachau RE, Crowe J, Davies M, Kolaskar AS, Lund O, Morrison S, Mumey B, Ofran Y, Pellequer JL, Pinilla C, Ponomarenko JV, Raghava GP, van Regenmortel MH, Roggen EL, Sette A, Schlessinger A, Sollner J, Zand M, Peters B (2007) Towards a consensus on datasets and evaluation metrics for developing B-cell epitope prediction tools. J Mol Recognit 20(2):75–82. doi:10.1002/jmr.815
4. Yang X, Yu X (2009) An introduction to epitope prediction methods and software. Rev Med Virol 19(2):77–96. doi:10.1002/rmv.602
5. Salimi N, Fleri W, Peters B, Sette A (2010) Design and utilization of epitope-based databases and predictive tools. Immunogenetics 62(4):185–196. doi:10.1007/s00251-010-0435-2
6. El-Manzalawy Y, Honavar V (2010) Recent advances in B-cell epitope prediction methods. Immunome Res 6(Suppl 2):S2. doi:10.1186/1745-7580-6-S2-S2
7. Caoili SE (2010) Benchmarking B-cell epitope prediction for the design of peptide-based vaccines problems and prospects. J Biomed Biotechnol, vol. 2010, Article ID 910524:1–14, doi:10.1155/2010/910524
8. Yao B, Zheng D, Liang S, Zhang C (2013) Conformational B-cell epitope prediction on antigen protein structures: a review of current algorithms and comparison with common binding site prediction methods. PLoS One 8(4):e62249. doi:10.1371/journal.pone.0062249
9. Jardetzky TS, Brown JH, Gorga JC, Stern LJ, Urban RG, Strominger JL, Wiley DC (1996) Crystallographic analysis of endogenous peptides associated with HLA-DR1 suggests a common, polyproline II-like conformation for bound peptides. Proc Natl Acad Sci U S A 93(2):734–738
10. Patronov A, Doytchinova I (2013) T-cell epitope vaccine design by immunoinformatics. Open Biol 3(1):120139. doi:10.1098/rsob.120139
11. Barlow DJ, Edwards MS, Thornton JM (1986) Continuous and discontinuous protein antigenic determinants. Nature 322(6081):747–748
12. Van Regenmortel MH (2006) Immunoinformatics may lead to a reappraisal of the nature of B cell epitopes and of the feasibility of synthetic peptide vaccines. J Mol Recognit 19(3):183–187
13. Salimi N, Fleri W, Peters B, Sette A (2012) The immune epitope database: a historical retrospective of the first decade. Immunology 137(2):117–123. doi:10.1111/j.1365-2567.2012.03611.x
14. Kringelum JV, Nielsen M, Padkjaer SB, Lund O (2013) Structural analysis of B-cell epitopes in antibody:protein complexes. Mol Immunol 53(1–2):24–34. doi:10.1016/j.molimm.2012.06.001
15. Kolaskar AS, Tongaonkar PC (1990) A semi-empirical method for prediction of antigenic determinants on protein antigens. FEBS Lett 276(1–2):172–174
16. Alix AJ (1999) Predictive estimation of protein linear epitopes by using the program PEOPLE. Vaccine 18(3–4):311–314
17. Odorico M, Pellequer JL (2003) BEPITOPE: predicting the location of continuous epitopes and patterns in proteins. J Mol Recognit 16(1):20–22
18. Saha S, Raghhava GPS (2004) BcePred: prediction of continuous B-cell epitopes in antigenic sequences using physico-chemical properties. LNCS 3239:197–204
19. Saha S, Raghava GP (2006) Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. Proteins 65(1):40–48
20. Larsen JE, Lund O, Nielsen M (2006) Improved method for predicting linear B-cell epitopes. Immunome Res 2:2
21. Sollner J, Mayer B (2006) Machine learning approaches for prediction of linear B-cell epitopes on proteins. J Mol Recognit 19(3):200–208
22. Chen J, Liu H, Yang J, Chou KC (2007) Prediction of linear B-cell epitopes using amino acid pair antigenicity scale. Amino Acids 33(3):423–428. doi:10.1007/s00726-006-0485-9
23. El-Manzalawy Y, Dobbs D, Honavar V (2008) Predicting linear B-cell epitopes using string kernels. J Mol Recognit 21(4):243–255. doi:10.1002/jmr.893
24. El-Manzalawy Y, Dobbs D, Honavar V (2008) Predicting flexible length linear B-cell epitopes. Comput Syst Bioinformatics Conf 7:121–132
25. Chang HT, Liu CH, Pai TW (2008) Estimation and extraction of B-cell linear epitopes predicted by mathematical morphology approaches. J Mol Recognit 21(6):431–441. doi:10.1002/jmr.910
26. Sweredoski MJ, Baldi P (2009) COBEpro: a novel system for predicting continuous B-cell

epitopes. Protein Eng Des Sel 22(3):113–120. doi:10.1093/protein/gzn075

27. Rubinstein ND, Mayrose I, Martz E, Pupko T (2009) Epitopia: a web-server for predicting B-cell epitopes. BMC Bioinformatics 10:287. doi:10.1186/1471-2105-10-287
28. Rubinstein ND, Mayrose I, Pupko T (2009) A machine-learning approach for predicting B-cell epitopes. Mol Immunol 46(5):840–847. doi:10.1016/j.molimm.2008.09.009
29. Wee LJ, Simarmata D, Kam YW, Ng LF, Tong JC (2010) SVM-based prediction of linear B-cell epitopes using Bayes feature extraction. BMC Genomics 11(Suppl 4):S21. doi:10.1186/1471-2164-11-S4-S21
30. Wang HW, Lin YC, Pai TW, Chang HT (2011) Prediction of B-cell linear epitopes with a combination of support vector machine classification and amino acid propensity identification. J Biomed Biotechnol 2011:432830. doi:10.1155/2011/432830
31. Gao J, Faraggi E, Zhou Y, Ruan J, Kurgan L (2012) BEST: improved prediction of B-cell epitopes from antigen sequences. PLoS One 7(6):e40104. doi:10.1371/journal.pone.0040104
32. Yao B, Zhang L, Liang S, Zhang C (2012) SVMTriP: a method to predict antigenic epitopes using support vector machine to integrate tri-peptide similarity and propensity. PLoS One 7(9):e45152. doi:10.1371/journal.pone.0045152
33. Lin SY, Cheng CW, Su EC (2013) Prediction of B-cell epitopes using evolutionary information and propensity scales. BMC Bioinformatics 14(Suppl 2):S10
34. Singh H, Ansari HR, Raghava GP (2013) Improved method for linear B-cell epitope prediction using antigen's primary sequence. PLoS One 8(5):e62216. doi:10.1371/journal.pone.0062216
35. Emini EA, Hughes JV, Perlow DS, Boger J (1985) Induction of hepatitis A virus-neutralizing antibody by a virus-specific synthetic peptide. J Virol 55(3):836–839
36. Parker JM, Guo D, Hodges RS (1986) New hydrophilicity scale derived from high-performance liquid chromatography peptide retention data: correlation of predicted surface residues with antigenicity and X-ray-derived accessible sites. Biochemistry 25(19):5425–5432
37. Vihinen M, Torkkila E, Riikonen P (1994) Accuracy of protein flexibility predictions. Proteins 19(2):141–149
38. Debelle L, Wei SM, Jacob MP, Hornebeck W, Alix AJ (1992) Predictions of the secondary structure and antigenicity of human and bovine tropoelastins. Eur Biophys J 21(5):321–329
39. Blythe MJ, Flower DR (2005) Benchmarking B cell epitope prediction: underperformance of existing methods. Protein Sci 14(1):246–248
40. Noble WS (2006) What is a support vector machine? Nat Biotechnol 24(12):1565–1567. doi:10.1038/nbt1206-1565
41. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
42. Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27. doi:10.1145/1961189.1961199
43. Joachims T (1999) Making large-scale support vector machine learning practical. Advances in kernel methods. MIT Press, Cambridge, MA, pp 169–184
44. Gasteiger E, Hoogland C, Gattiker A, Duvaud S, Wilkins MR, Appel RD, Bairoch A (2005) Protein identification and analysis tools on the ExPASy server. In: Walker JM (ed) The proteomics protocols handbook. Humana, Totowa, NJ, pp 571–607
45. Deleage G, Roux B (1987) An algorithm for protein secondary structure prediction based on class prediction. Protein Eng 1(4):289–294
46. Kyte J, Doolittle RF (1982) A simple method for displaying the hydropathic character of a protein. J Mol Biol 157(1):105–132
47. Karplus PA, Schulz GE (1987) Refined structure of glutathione reductase at 1.54 A resolution. J Mol Biol 195(3):701–729
48. Alix AP (1997) Molecular modeling of globular proteins: strategy 1D ⇒ 3D: secondary structures and epitopes. In: Vergoten G, Theophanides T (eds) Biomolecular structure and dynamics, vol. 342. NATO ASI series. Springer, Netherlands, pp 121–150. doi:10.1007/978-94-011-5484-0_6
49. Giardina CR, Dougherty ER (1988) Morphological methods in image and signal processing. Prentice-Hall, Inc., Upper Saddle River, NJ
50. Maragos P, Schafer RW (1987) "Morphological Filters" part I and II. IEEE Trans Signal Process 35(8):1153–1184
51. Serra J (1982) Image analysis and mathematical morphology, vol 1. Academic, New York
52. Serra J (1988) Image analysis and mathematical morphology, vol 2. Academic, New York

53. Liu C-H (2007) Mathematical morphology based biochemical property filters for linear epitope prediction. National Taiwan Ocean University, Keelung, Taiwan

54. Yousef M, Jung S, Showe LC, Showe MK (2008) Learning from positive examples when the negative class is undetermined–microRNA gene identification. Algorithms Mol Biol 3:2. doi:10.1186/1748-7188-3-2

55. Saha S, Bhasin M, Raghava GP (2005) Bcipep: a database of B-cell epitopes. BMC Genomics 6:79. doi:10.1186/1471-2164-6-79

56. Ivanciuc O (2007) Applications of support vector machines in chemistry. Reviews in computational chemistry. Wiley, Hoboken, NJ, pp 291–400. doi:10.1002/9780470116449.ch6