Artificial Intelligence In Medicine 94 (2019) 28–41

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Antigenic: An improved prediction model of protective antigens

M. Saifur Rahman[a], Md. Khaledur Rahman[b], Sanjay Saha[c], M. Kaykobad[a], M. Sohel Rahman[a,*]

[a] Department of CSE, BUET, ECE Building, West Palasi, Dhaka 1205, Bangladesh
[b] Indiana University, Bloomington, USA
[c] Department of Computer Science and Engineering, University of Asia Pacific, 74/A Green Road, Dhaka 1215, Bangladesh

ARTICLE INFO

Keywords:
Antigens
Non-antigens
Vaccine
Reverse vaccinology
Classification
Prediction
Support vector machine
Random forest
PseAAC

ABSTRACT

An antigen is a protein capable of triggering an effective immune system response. Protective antigens are the ones that can invoke specific and enhanced adaptive immune response to subsequent exposure to the specific pathogen or related organisms. Such proteins are therefore of immense importance in vaccine preparation and drug design. However, the laboratory experiments to isolate and identify antigens from a microbial pathogen are expensive, time consuming and often unsuccessful. This is why Reverse Vaccinology has become the modern trend of vaccine search, where computational methods are first applied to predict protective antigens or their determinants, known as epitopes. In this paper, we propose a novel, accurate computational model to identify protective antigens efficiently. Our model extracts features directly from the protein sequences, without any dependence on functional domain or structural information. After relevant features are extracted, we have used Random Forest algorithm to rank the features. Then Recursive Feature Elimination (RFE) and minimum redundancy maximum relevance (mRMR) criterion were applied to extract an optimal set of features. The learning model was trained using Random Forest algorithm. Named as *Antigenic*, our proposed model demonstrates superior performance compared to the state-of-the-art predictors on a benchmark dataset. Antigenic achieves accuracy, sensitivity and specificity values of 78.04%, 78.99% and 77.08% in 10-fold cross-validation testing respectively. In jackknife cross-validation, the corresponding scores are 80.03%, 80.90% and 79.16% respectively. The source code of Antigenic, along with relevant dataset and detailed experimental results, can be found at https://github.com/srautonu/AntigenPredictor. A publicly accessible web interface has also been established at: http://antigenic.research.buet.ac.bd.

1. Introduction

An antigen is a protein that is capable of triggering a measurable immune system response [1]. Antigens can be subdivided into overlapping subclasses such as serodiagnostic, crossreactive and protective antigens [2]. Serodiagnostic antigens are associated with a differential humoral antibody response between naive and exposed individuals. Such antigens are important for diagnostics purposes. Cross-reactive antigens are associated with a strong humoral antibody response in both naive and exposed individuals. Protective antigens, on the other hand, are the ones that can stimulate protective immunity against pathogens. That is, these antigens can invoke specific and enhanced adaptive immune response to subsequent exposure to the specific pathogen or related organisms. Protective antigens are of immense importance in vaccine preparation and drug design [3–5].

Vaccines are molecular or supramolecular agents that can stimulate protective immunity against microbial pathogens. They can prevent, or at least improve, the effects of infection [6]. Vaccination has been the most effective method of preventing infectious diseases such as influenza, smallpox, varicella, diphtheria, tetanus, polio, hepatitis, rotavirus and more [7–12]. However, the battle against many infectious diseases is far from complete. It is still difficult to develop safe and effective vaccines against tuberculosis, HIV, malaria and so on [13].

Vaccines are prepared from killed or attenuated microorganisms, or subunits purified from them [14,6]. While vaccines based on attenuated pathogens can be highly effective, this technique is seldom used in modern vaccinology due to safety concerns and technical reasons [15]. Subunit vaccines, on the other hand, use only the protective antigens, instead of the entire microorganism. This reduces the chance of any adverse reaction to the vaccine [16]. The hepatitis B vaccine, containing the surface antigen HbsAg, is an example of one of the most successful subunit vaccines [17,18]. The advent of recombinant DNA

* Corresponding author.
E-mail addresses: mrahman@cse.buet.ac.bd (M.S. Rahman), morahma@iu.edu (Md. K. Rahman), sanjay@uap-bd.edu (S. Saha), kaykobad@cse.buet.ac.bd (M. Kaykobad), msrahman@cse.buet.ac.bd (M.S. Rahman).

https://doi.org/10.1016/j.artmed.2018.12.010
Received 4 May 2018; Received in revised form 31 October 2018; Accepted 28 December 2018

technology (rDNA) has conceived the idea of *multiepitopic* vaccines [19]. In this technique, several protective epitopes (parts of an antigen that is recognized by the immune system) are included in a single molecule, immunodominant but non-protective epitopes are discarded. Epitopes exerting adjuvant effects can also be included to enhance the protective response. This opens up the possibility of designing highly efficient, multi-target vaccines [20].

The modern trend in vaccine preparation has therefore been towards creating subunit vaccines or epitope vaccines containing only full or partial protective antigens. As a result, identification of protective antigens or their determinants is a key step in any vaccine development project [21]. The microbiological approach for antigen identification comprises several steps. At first, the target pathogen is cultivated under laboratory conditions. It is then purified and dissected into the constituent proteins. The proteins are then assayed in cascades of *in vitro* and *in vivo* assays. Finally, the proteins which display requisite protective immunity are identified [22]. While this process requires many hours of expensive and laborious tasks, it does not always yield fruitful results. For example, it is not always possible to cultivate a particular pathogen outside of the host organism. Also, as many proteins are only expressed transiently during the course of an infection, the antigens expressed *in vivo* may not always express during *in vitro* cultivation [1]. These limitations of the laboratory experiments, coupled with wide availability of whole genome sequences of pathogens, have led researchers explore techniques that are based on computational genomics and thus a new paradigm known as Reverse Vaccinology has emerged.

Reverse vaccinology (RV) [16,23] is a computational pipeline for identification of protective antigens or epitopes against microorganisms from their genome sequences. In this approach, all proteins of a pathogen proteome are first screened *computationally* for their vaccine potential. Computationally predicted protective antigens are then tested *in vivo* and *in vitro* for their immunogenicity. This approach dramatically cuts down the cost and increases the speed of progress in vaccine discovery. RV was first applied to the development of a vaccine against serogroup B Neisseria meningitidis (MenB), which causes sepsis and meningitis in children and young adults [23]. This has eventually led to the approval of the first MenB vaccine, BEXSERO®, for use in Europe [24], and United States [25]. This is a milestone for rational vaccine design using RV. This principle for vaccine development has successfully been applied against many other pathogens, including *Helicobacter pylori* [26], *Streptococcus pneumoniae* [27], *Porphyromonas gingivalis* [28], *Chlamydia pneumoniae* [29], *Bacillus anthracis* [30] and *Mycobacterium tuberculosis* [31].

Over the years, researchers have developed many computational techniques for protective antigen prediction. Some of these techniques are focused on specific pathogen models, while some are more generic. Some techniques use concepts of sequence alignment, while other ones leverage statistical tools or machine learning methods. In this paper, we propose a protective antigen predictor that is based on the latter approach. Based on features extracted from the primary sequence of the protein, our method provides a fast and simple prediction model that outperforms the existing predictors. But before we jump into the details of our predictor, we briefly review the literature of protective antigen prediction here.

For a sequence-alignment based approach to be useful, sequences of many extant antigens must be available in a database. Sequence searching programs such as BLAST [32], FASTA [33], PSORT [34], etc., can then be applied to identify similar sequences in the target genome. However, such an approach will fail to discover truly novel protective antigens which lack any sequence similarity with the repository of known protective antigens.

Another criterion, that has frequently been used to screen for potential antigens, is the likelihood of a protein containing a signal sequence. SignalP [35] has widely been used in this regard. It originally employed neural networks to predict the presence and location of signal peptide cleavage site [36]. Subsequently a hidden Markov model (HMM) was implemented which is able to discriminate uncleaved signal anchors from cleaved signal peptides [37]. Several updates to this predictor have been made in recent years [38,39]. One of the limitations of SignalP, however, is overprediction, as it cannot reliably discriminate between several very similar yet distinct signal sequences [1].

Vivona et al. [40] developed a system for antigen discovery, called NERVE, that works in several stages as follows. Firstly, the target protein's subcellular localization is predicted. Then whether the protein is an adhesin is determined. This is followed by the identification of transmembrane domains. The protein is then compared against human and pathogen proteomes. Finally it is assigned a suggestive score. However, the system requires software download and database setup and does not include precomputed data of vaccine target prediction, which makes its use inconvenient and time consuming [41].

Doytchinova et al. [42] proposed the first alignment-free approach for antigen prediction. They trained the predictor for three different models: bacteria, virus and tumor. Each model was trained with a balanced dataset of 100 known protective antigens and 100 non-antigens. The principal amino acid properties were represented by z descriptors, originally derived by Hellberg et al. [43]. A transformation using auto cross covariance (ACC) [44] was then applied to produce a uniform vector of 45 terms for each protein sequence. Then a two-class discriminant analysis was performed using the partial least squares technique (DA-PLS). The cross-validation accuracy of their predictor was 82% for the bacterial model, 87% for the viral model and 85% for the tumor model. The models were implemented in a server called VaxiJen [45], which has since been widely used. However, the dataset used to create VaxiJen was rather small. Additionally, several of the sequences in the non-antigen set were subsequently predicted as antigens by other methods [46]; some were also experimentally discovered as such [47,48].

In a subsequent work [49], Doytchinova et al. added parasite and fungal models to the VaxiJen predictor. For this purpose, 117 parasitic and 33 fungal antigens were identified from the literature. For each antigen, a non-antigen protein was randomly selected from the same species. The same features and learning algorithms were used as before. The parasite model achieved an accuracy of 78% while the fungal model obtained 97% accuracy.

Ansari et al. developed AntigenDB [50], a database compiling more than 500 antigens, from 44 important pathogenic species. This database maintains information regarding the sequence, structure, origin, etc. of antigens. B and T-cell epitopes, MHC binding, function, gene-expression and post translational modifications are also available for some antigens. He et al. [41] introduced Vaxign, another web-based vaccine design system that can predict protein subcellular location, transmembrane helices, adhesin probability, conservation to human and/or mouse proteins etc. The precomputed Vaxign database contains prediction of vaccine targets for more than 70 genomes.

Magnan et al. [2] developed another predictor for protective antigens, called ANTIGENpro. Unlike VaxiJen's approach of pathogen specific prediction models, they created a generic classifier of antigens from any pathogen. To train their classifier, they first collected known protective antigens from literature. They then augmented this set using human immunoglobulin reactivity data obtained from protein microarray analyses. ANTIGENpro achieved 76% accuracy in 10-fold cross-validation experiments. Unfortunately, ANTIGENpro server [51] restricts queries to only one protein sequence per submission. This makes its use on a genome-wide scale quite impractical [52].

El-Manzalawy et al. [52] proposed another predictor called BacGen which can classify antigens for bacteria model only. They used amino acid moment descriptors (AAMD) [53] as features. After applying Haar wavelet transform (HWT) [54], they used Random Forest [55] as the classifier. Finally they combined the prediction of Random Forest algorithm with SignalP [35] prediction. Their approach produced results that are competitive with ANTIGENpro. However, while BacGen was

implemented as a web server (http://ailab.cs.iastate.edu/bacgen/), it does not seem to be in service anymore.

Jaiswal et al. [46] also developed a web-based predictor, for protein vaccine candidates (PVCs) for bacterial pathogens. Called Jenner-Predict, the predictor targets host-pathogen interactions by considering known functional domains from various protein classes. Altindis et al. [15] examined the structural and functional features recurring in known bacterial protective antigens to define "protective signatures" which can be used for protective antigen discovery. They applied their approach to Staphylococcus aureus and Group B Streptococcus and were able to identify two new protective antigens, in addition to rediscovering the already known protective antigens. Ong et al. [56] in a recent publication verified the critical role of adhesins, subcellular localization, peptide signaling, in predicting secreted extracellular or surface-exposed protective antigens. They also found a significant negative correlation of transmembrane α-helix to antigen protectiveness in Gram-positive and Gram-negative pathogens. Their findings can be used to extract relevant features from the protein secondary structure to discriminate between protective antigens and non-antigens.

While significant amount of work has been done in protective antigen prediction, the performance of the current predictive tools has left a lot of room for improvement. Also, some of the state-of-the-art predictors use feature extraction techniques that are time consuming, some use sophisticated prediction models which are susceptible to the overfitting problem. In this paper, we therefore propose a protective antigen predictor that extracts features from the protein sequence alone, that has a fast and simple prediction model and that outperforms the existing predictors. We have followed Chou's 5-step procedure [57] for establishing our predictor. The steps include dataset preparation, extracting relevant features from protein sequences, learning the classification model using a powerful algorithm, objectively evaluating the predictor and finally making the predictor available through a web server for wide adoption. We have collected a benchmark dataset from literature and then applied a fixed length vector representation of the protein. In addition to amino acid composition (AAC), we have used three different sequence based feature construction techniques to create the feature vector. Each of these features provides some sequence-order information. As we created a large feature vector, feature selection became necessary. Random Forest [55] algorithm was then applied to rank the features. We have then applied Support Vector Machine (SVM) [58] in combination with Recursive Feature Elimination (RFE) to identify an optimal subset of features. In this step we have also experimented with the minimum redundancy maximum relevance (mRMR) [59,60] criterion for feature selection. Finally Random Forest was used again, but this time to train the classifier. Named as *Antigenic*, our predictor has been evaluated based on several well-established performance metrics. Antigenic convincingly demonstrated superior predictive performance compared to its predecessors. Therefore, it has been made available publicly as an web interface for wide adoption.

2. Material and methods

There are five steps in establishing a predictor for any protein attribute prediction problem [57]. These steps can be summarized as follows:

1. Preparation of a stringent benchmark dataset.
2. Protein sample representation. The representation scheme should be able to extract and utilize intrinsic information relevant to the attribute to be predicted.
3. Development of a powerful algorithm for the prediction process.
4. Predictor evaluation.
5. Making the predictor publicly available for wide adoption.

In what follows, we describe our methodology in accordance with this 5-step rule.

2.1. Benchmark dataset

In order to create a robust predictor, there needs to be a reliable training dataset of relatively large size. For our study we have collected the benchmark dataset from [2]. This dataset, prepared by Magnan et al., was not available publicly. However, they kindly provided us with the dataset upon request through private communication. Below, we provide a brief description of the dataset and how it was prepared.

Magnan et al. [2] argued that mere literature review did not generate a satisfactory collection of protective antigens. Therefore, they prepared the benchmark dataset based on protein microarray data analysis for training and testing their predictor. They leveraged a high-throughput technology [61] to study the humoral immune response to pathogen infection using protein microarrays. In this approach, proteins of a pathogen genome are expressed by a proprietary *in vitro* expression system. These expressed proteins can then be probed with sera from naive, exposed and vaccinated individuals. The resulting reactivity data gives a reliable estimate of the humoral immune response. The protein microarray data can thus be used to prepare a dataset of antigens and non-antigens to train a predictor. Although the microarray data does not directly provide information about whether or not a particular antigen is protective, Magnan et al. [2] hypothesized that the actual protective antigens are significantly overrepresented among the set of antigens for which the protected individuals elicit a significant antibody response, and the unprotected individuals do not. They have validated this hypothesis in their work.

The benchmark dataset contains a training set as well as a testing set. The training set consisted of six subsets. Of these, five subsets were curated from protein microarray data analysis for pathogens *Candida albicans*, *Plasmodium falciparum*, *Brucella melitensis*, *Burkholderia pseudomallei* and *Mycobacterium tuberculosis*. Each of these subsets contained some antigens as well as non-antigens. The other (6th) subset, on the other hand, contained only protective antigens collected from literature and public databases. This subset is referred to as *PAntigens*.

Any redundancy or considerable pairwise sequence similarity in the training dataset may hamper the quality of the model being trained. The cross-validation results may also get overestimated. To mitigate this concern, BLASTCLUST [62] was run with a 30% similarity threshold after combining the data from the five pathogens in the training set and redundant sequences were removed. The PAntigens set was similarly processed. It is possible, however, that some antigens in the PAntigen set may have redundancy with the proteins in the pathogens set. As such, proteins in the merged pathogen set with more than 30% sequence similarity with any protein in PAntigens were also removed. The composition of the training set, after all processing, is shown in Table 1.

It is noteworthy here that earlier works used much smaller datasets and did not have validated non-antigens. Instead, proteins selected at random and having very little sequence similarity with known protective antigens were tagged as non-antigens. In the benchmark dataset of [2], however, the non-antigens are curated by selecting proteins with low seroreactivity according to the protein mircoarray experiments.

The testing set was constructed from protein microarray data analysis for the pathogen *Bartonella henselae*. This dataset consists of 1463

Table 1
Size and composition of the six protein sets used as the training set.

Protein set	Size	Antigenic	Non-antigenic
PAntigens (PAN)	213	213	0
Brucella (BRU)	206	70	136
Burkholderia (BUR)	17	5	12
Candida (CAN)	13	3	10
Malaria (MAL)	333	114	219
Tuberculosis (TUB)	542	171	371
Total	1324	576	748

proteins of which 73 were antigenic. The remaining 1390 were non-antigens.

For details of the microarray data analysis and protocols followed to prepare the benchmark dataset, the reader is referred to [2].

2.2. Protein sample representation

Let a protein sequence P of length L be written as:

$$P = R_1R_2R_3R_4R_5\cdots R_L$$

Here R_1 represents the first amino acid residue, R_2 the second residue and so on. We have chosen a discrete model for its representation. In a discrete model, each protein is represented by a fixed length feature vector that is independent of the protein sequence length. We have done so because the learning algorithms that we have applied cannot work with a sequential representation model. The discrete model that we have chosen is Chou's General Formulation of Pseudo Amino Acid Composition (PseAAC) [57,63].

The key idea in PseAAC is to use a discrete model to represent a protein yet somehow incorporate sequence-order information into the feature vector. Mathematically, the generalized PseAAC of a protein can be represented as follows:

$$P = [\psi_1 \ \ \psi_2 \ \ \ldots \ \ \psi_u \ \ \ldots \ \ \psi_\Omega]^T$$

where the classical amino acid composition (AAC) is represented for subscripts $1 \le u \le 20$ and the subsequent features express sequence order information through one or more different schemes. This simple yet powerful concept of PseAAC has successfully been used in almost all the areas of computational proteomics. (See a long list of references cited in a 2014 article [64] as well as the 2009 review [65].)

In this paper, too, we have applied the concept of PseAAC. To capture the sequence order information, we have extracted position independent as well as position specific features. Among the position independent features are dipeptides (Dip), tripeptides and n-gapped-dipeptides (nGDip). These features do not depend on any specific position in the amino acid sequence. These features have widely been used in the literature of protein attribute prediction. The position specific features, on the other hand, were used only in [66,67]. We describe each of these feature construction techniques briefly below. In describing the feature types, we have followed the nomenclature from [68].

2.2.1. Amino acid composition (AAC)

Amino acid composition (AAC) of a protein sequence refers to the normalized frequencies of the 20 native amino acids. The frequencies are normalized by dividing each of these by the protein sequence length.

2.2.2. Dipeptides (Dip)

Dipeptides (Dip) or dipeptide composition (DPC) features are the normalized frequency of adjacent amino acids within the protein sequence. These features provide some sequence-order information. Dip (DPC) contributes 400 features to PseAAC.

2.2.3. Tripeptides

Similarly, the normalized frequency of three consecutive amino acids can be used as features. This is called tripeptides composition feature type. This feature space consists of 8000 features. AAC, dipeptides and tripeptides – all these feature types can be generalized under the umbrella of *n-grams* feature type, where frequencies of n-length peptides are used as feature vectors. Such features have also been referred to as n-peptide descriptors in literature [69]. In our study, we extract a total of 8420 n-grams features, for $n = 1, 2$ and 3.

2.2.4. n-gapped-dipeptides (nGDip)

The n-gapped-dipeptides (nGDip) features are extracted by counting the frequency of amino acid dipeptides such that the amino acids are separated by n positions. The frequency is normalized, dividing it by the total number of nGDip (i.e., $L - n - 1$ for a sequence of length L). For each specific gap, 400 features can be generated.

The nGDip feature extraction technique is motivated by the belief that the gap between any two amino acids may carry significant information about the protein [70]. In our work, we have considered up to 25 position gaps. Thus we get a total of $25 \times 400 = 10,000$ n-gapped-dipeptides features.

For some features, all the samples of the training set may produce 0 frequency. Such features will naturally have no effect on the learning model. We have carefully removed these features from the feature vector. Subsequently the n-grams feature count reduced to 8409.

2.2.5. Position specific n-grams (PSN)

The position specific n-grams (PSN) represent whether specific n-grams occur in specific positions in the protein sequence. The value of each such feature in any sequence will therefore be either 0 or 1 (*on* or *off*). Like in the case of position independent features, we have considered n-grams for $n = 1, 2$ and 3 in case of PSN as well. To avoid feature space explosion, we followed the approach used in [66] and considered only the first 10 positions of the N-terminus part of the protein sequence to extract PSN features. The number of PSN features for our training set was 14,058.

Counting all types of features, we have thus extracted a total of $8409 + 10,000 + 14,058 = 32,467$ features. These features can be represented in Chou's PseAAC as follows: For $1 \le u \le 20$, we have the amino acid composition in the feature vector. From $21 \le u \le 8409$, the dipeptide and tripeptide compositions are represented. From $8410 \le u \le 18,409$, the features in this vector comes from the nGDip feature space. Finally, the PSN features construct the remaining portion of the PseAAC, from $18,410 \le u \le 32,467 = \Omega$.

Notably, a comprehensive list of protein feature extraction approaches, within the framework of PseAAC, can be found in [69]. These include, in addition to the n-peptides, Quasiresidue Couple (QRC), Autocovariance (AC), AAIndexLoc, Global Encoding (GE), Discrete Wavelet (DW) and so on. We plan to explore different combinations of these for protein sample representation in our future work.

2.3. Prediction algorithm

The first step in our prediction algorithm is *feature selection*. In this step we apply several techniques to reduce the size of the feature vector. As our protein samples are represented by a total of 32,467 features, it would be computationally infeasible to train a classifier with the amount of computing power and memory we have at our disposal. The motivation behind the feature selection step obviously is to reduce the cardinality of the feature vector to a manageable size. Besides, a set of relevant features must be selected that is able to express the intrinsic difference between antigens and non-antigens. To achieve this goal we have first applied Random Forest algorithm to rank the features based on their importance. Then multiple rounds of SVM-RFE [71] were applied to select a suitable subset of the ranked features. We have also experimented with the minimal redundancy maximal relevance (mRMR) [59,60] criterion for feature selection.

Feature ranking based on Random Forest. The importance of different features can be computed by permuting out-of-bag (OOB) data of Random Forest algorithm. First, the prediction error on the OOB portion of the data is recorded for each tree. Afterwards, each predictor variable is permuted and the error is recalculated. The difference between the two errors is then averaged over all trees and then normalized by the standard deviation of the differences. This is called *mean decrease in accuracy*. The larger this value is for a feature, the more important that feature is in the context of the prediction task. We would like to run Random Forest algorithm on the full set of features and take the mean decrease in accuracy as the ranking score of each feature.

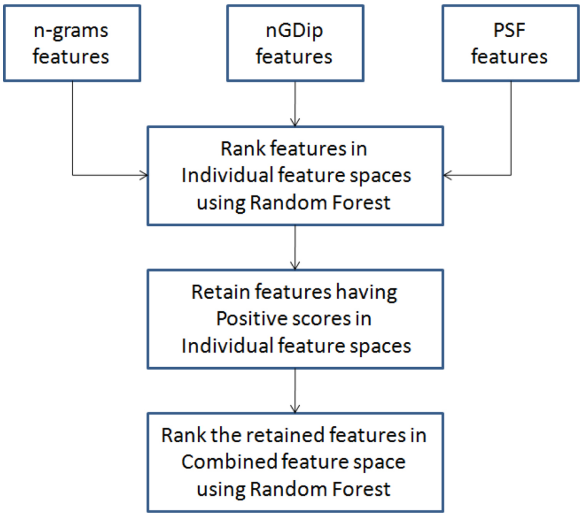


Fig. 1. Steps in feature ranking based on Random Forest.

Random Forest algorithm is not used as the classifier at this point, rather it has been used as a means to generating the feature ranking. However, as our feature space is quite large, our attempt to get the ranking scores with the best server machine in our computing laboratories did not finish even after one month of execution. To overcome this bottleneck, we followed the steps shown in Fig. 1 as follows.

- In the first step, we ranked the features in each individual feature space by applying the Random Forest algorithm in that respective feature space in separation.This was manageable, since the size of the largest feature space was around 14,000. Generation of each of the three random forests (and therefore the respective ranking scores) took less than a day.
- In the second step, features with positive score in each feature space were retained, while the remaining features were eliminated. We thus retained 3625 n-grams features, 5811 nGDip features and 1363 PSN features which totals to 10,799 features.
- In the third step, Random Forest algorithm was applied on the combined feature space to rank these 10,799 features. In this ranking, 5196 features had positive mean decrease accuracy scores, 2283 features had 0 scores and remaining features had negative scores. This is why the cumulative scores of the features are almost identical for top 5500 and 7500 features, demonstrated in Fig. 2; and they are superior compared to the cumulative scores when all features are considered, as observed in the same figure.

Feature ranking based on SVM-RFE. After getting the feature ranking as above, the top-ranked 10,000 features were re-ranked using SVM based Recursive Feature Elimination (SVM-RFE) [71] as follows. SVM was first run on the entire feature set and the technique described in [71] was applied to rank the features. In the recursive step, we eliminated 25 least ranked features and ran SVM again to recompute the ranking in the reduced feature space. The recursion was repeated until all the features are eliminated. Thus a new feature ranking is obtained. We call it the SVM-RFE (coarse) ranking.

Once the feature ranking was done, different prediction models were constructed using the Random Forest algorithm. As we have an imbalanced dataset, we balanced it by undersampling the larger (negative) class randomly. After this step of random under sampling, we generated several prediction models by varying the number of features and compared their performances. We found the top 600 features to be most promising. (The reader is referred to Section 3 for details of how the number of features were varied). A second round of SVM-RFE was applied in this feature space, but this time with steps of 1 feature elimination (instead of 25 features). This gives a more reliable ranking of the top 600 features. We have referred to this final ranking as SVM-RFE (fine) ranking. Using this ranking, we again explored several models of different feature count and found the model with 490 features to be the best model.

Feature ranking based on mRMR. In some experiments, we have applied mRMR criterion for feature ranking, instead of SVM-RFE. Relevant features can be chosen according to the maximal statistical dependency criterion based on mutual information. However, this criterion is difficult to implement. As such, Ding et al. [59,60] proposed an equivalent form, which is known as minimal redundancy maximal relevance (mRMR) criterion. After all the features were ranked using Random Forest algorithm, the top 600 features were re-ranked using the mRMR criterion. Using this ranking, several models with different number of features were explored in both the balanced and unbalanced setup.

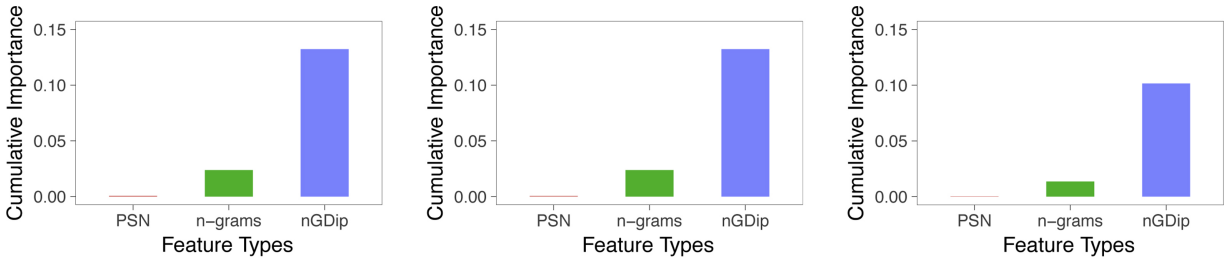
2.4. Predictor evaluation

To evaluate the performance of antigenic, we have applied the following testing methods and performance metrics.

2.4.1. Testing methods

A prediction model can be tested using different well established testing methodologies. These include jackknife cross-validation, 10-fold cross-validation test, independent test etc. We briefly describe these techniques below.

Jackknife cross validation. In jackknife cross-validation, the prediction model is trained with the training dataset after leaving one sample out. Then the sample that was left out is used to test the model. This is repeated N times, where N is the size of the training set. In each iteration, the testing sample is always different from previous testing samples, so that all samples are considered once as the testing sample.



(a) Top 5500 features. (b) Top 7500 features. (c) All features.

Fig. 2. Categorized feature importance based on Random Forest ranking. The aggregate ranking score is better for subsets of top-ranked features, compared to all features. PSN: position specific n-grams, n-grams: combination of AAC, dipeptide and tripeptide composition features, nGDip: n-gapped dipeptides.

Since a specific sample can be left out in only one way, no more than one partitioning is possible in each iteration. As such, it always produces unique result, which is a key advantage of this technique. However, it executes slowly compared to other testing techniques. Since one sample is left out in each iteration, this technique is also widely known as the *leave-one-out* cross-validation technique.

10-fold cross validation. As jackknife cross validation is slow, a popular alternative is the 10-fold cross validation. In this approach, the training dataset is divided into 10 equal parts. Among these 10 parts, one part is used for testing and other 9 parts are used to train the model. This is repeated 10 times so that each part gets to be used for testing exactly once. We have employed this technique in this paper. Since a dataset can be divided into 10 subgroups in exponentially many ways, the result of 10-fold cross validation may vary depending on how the data has been partitioned. To mitigate this problem, in this paper, we have averaged the results of 5 runs of 10-fold cross validations. We have also recorded the standard deviation of the results obtained in these runs.

Independent testing. In independent testing, the testing dataset is completely different from the training dataset. After constructing the model using the training set, independent testing is performed using the testing dataset. The distribution of the testing dataset should be similar to that of the training dataset. Otherwise, output of this testing strategy may be misleading [72]. In this paper, we have conducted independent testing using the protein microarray dataset of the pathogen *B. henselae*, as obtained from [2].

2.4.2. Performance metrics

As performance metrics, in this paper we have used accuracy, sensitivity, specificity and Matthew's correlation coefficient (MCC). These are well-established performance metrics in literature [73,74]. We have also analyzed the area under receiver operating characteristic curve (ROC-Curve) [75] and area under precision-recall curve (PR-Curve) [76].

In a binary classification problem, we can label the samples in the dataset into two classes: the positive class and the negative class. When the true class of a test sample is positive (negative) and the predicted class is also positive (negative), it is called a true positive (true negative). When true class of a testing sample is positive (negative) but predicted class is negative (positive) it is called a false negative (false positive).

Let *P*, *N*, *TP*, *TN*, *FP*, *FN* respectively denote the number of positives, negatives, true positives, true negatives, false positives and false negatives. Then we can define the relevant performance metrics by the following equations:

$$\text{Accuracy(Acc)} = \frac{\text{TP} + \text{TN}}{P + N}$$

$$\text{Sensitivity(Sn)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity(Sp)} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FN})(\text{TP} + \text{FP})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

The receiver operating characteristic curve (ROC-Curve), is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. TPR is Sensitivity, while (1 − *Specificity*) gives FPR. When ROC-Curve gets close to the left upper corner in the graph, it indicates better performance [75]. In this case, we get higher value for area under ROC-Curve (auROC). The precision recall curve or the PR-curve, on the other hand, is created by plotting the precision against the recall at various threshold settings. The closer the PR-curve is to the top right corner of the graph, the better is the performance of the predictor [76]. In this case, we get higher value for area under PR-Curve (auPR). We have analyzed the ROC-Curve and the PR-Curve in evaluating Antigenic. These two measures combined can accurately reflect the performance of a system considering both balanced and imbalanced datasets.

2.4.3. Experimental setup and packages

We have conducted all our experiments using R language (version 3.2.3 or above). We used three different machines with the following configurations:

- A Desktop computer with Intel Core i5 CPU @ 3.00GHz × 4, Windows 7, 64-bit OS and 4 GB RAM.
- A Desktop computer with Intel Core i5 CPU @ 3.20GHz × 4, Ubuntu 16.04, 64-bit OS and 8 GB RAM.
- A server machine with Intel Xeon CPU E5-4617 0 @ 2.90GHz × 6, Ubuntu 13.04 64-bit OS, 15 MB L3 cache and 64 GB RAM.

Random Forest and support vector machine (SVM) algorithms were used from the R packages *randomForest* and *e1071* respectively. The number of trees (*ntree*) in the Random Forest algorithm was restricted to 500, while the number of variables tried at each split (*mtry*) was set to the square root of the total number of features. The mRMR criterion for feature selection was used from the *mRMRe* package. In addition to pre-installed packages in R, we have also used *ROCR* and *pracma* packages for performance analysis of our model. For plotting different graphs, we have leveraged *ggplot2* package. All of our source code, experimental results, cross-validation and independent datasets are available at: https://github.com/srautonu/AntigenPredictor.

2.5. Predictor availability

Antigenic is freely available as an R script at https://github.com/srautonu/AntigenPredictor. Additionally, we have established a publicly accessible web server at http://antigenic.research.buet.ac.bd to facilitate wide adoption. We hope our predictor will be beneficial to researchers working in the field of reverse vaccinology.

3. Results and discussion

We have conducted several experiments to assess the quality of our proposed predictor. We have experimented with varied number and type of features, feature selection technique etc. We also run experiments to compare Antigenic with VaxiJen and ANTIGENpro, the two most widely used alignment-free predictors of protective antigens. We describe these experiments and analyze their results in this section.

As the benchmark dataset is imbalanced, using it directly to learn the classifier may create a bias towards the majority class. Therefore we have balanced the dataset by random undersampling of the majority class, following [2]. Dittman et al. [77] has recently shown, however, when Random Forest is used as the learning algorithm, the increase in performance due to balancing the training set using random undersampling is not statistically significant. Hence, in many of our experiments we have used two different models – *Antigenic**, a model that was trained directly on the entire training set, and *Antigenic*, a model that was trained with a balanced (reduced by random undersampling) set. In all the experiments, we have applied Random Forest based ranking, followed by SVM-RFE, for feature selection. In some cases, we have also experimented with the mRMR criterion. In such cases, the model names have been annotated with the feature selection method. When mRMR has not been used and there is no chance of ambiguity, such annotation is removed. In other words, the default method of feature selection has been SVM-RFE, after the initial feature ranking is obtained from Random Forest algorithm.

3.1. Impact of number of features

To find the ideal number of features, we ran several experiments

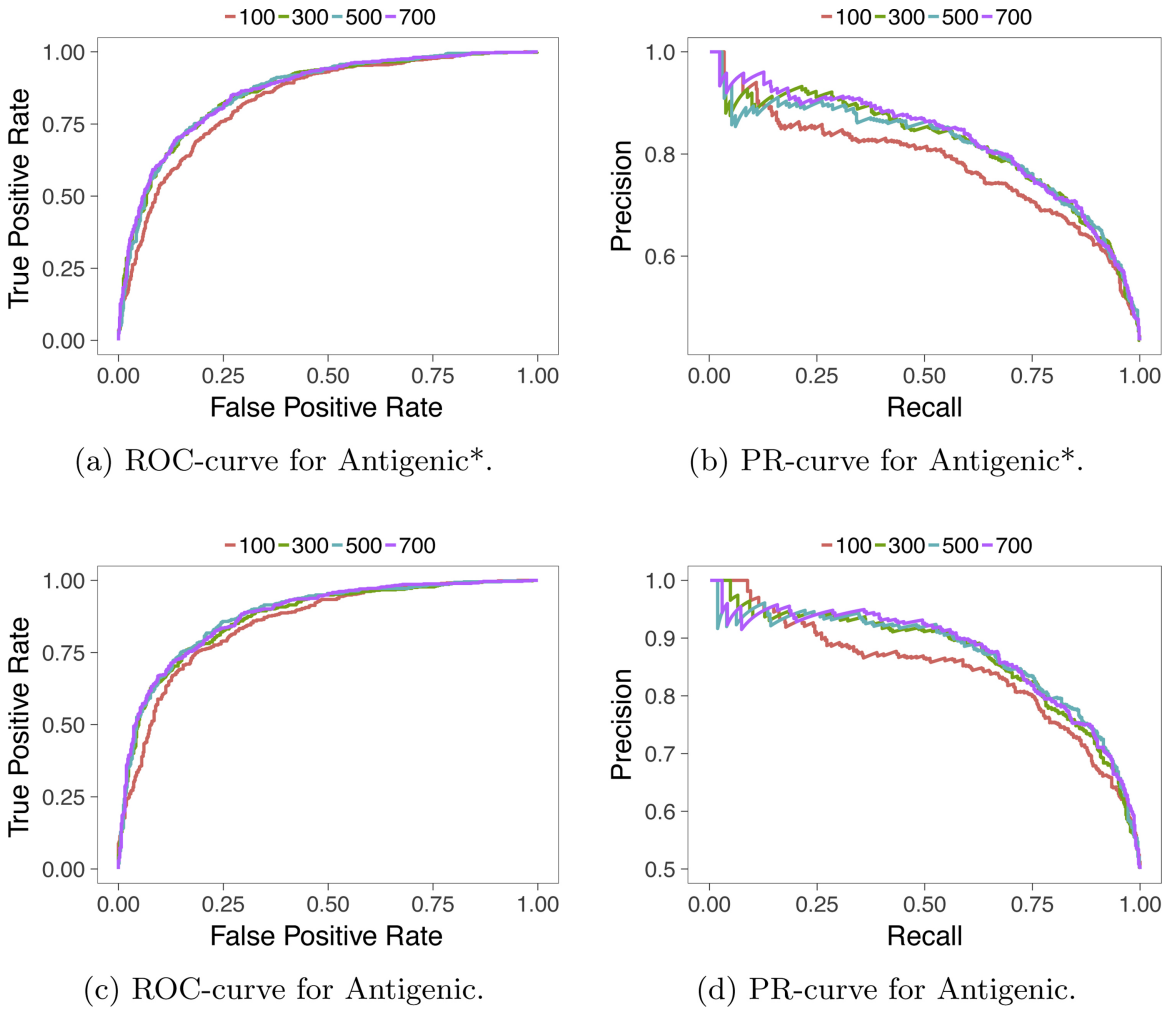


Fig. 3. ROC and PR curves of prediction models with varying number of features, generated by 10-fold cross validation on the training dataset. (The reader is referred to the web version of this article for interpretation of the references to color in this figure legend.)

varying the number of features and analyzed the impact on the classification model. The analysis was based on the various performance metrics discussed in the earlier section. In Fig. 3, ROC and PR curves have been plotted for Antigenic and Antigenic*. In each case, four different curves are generated for models trained with best 100, 300, 500 and 700 features respectively. The closer a ROC curve is to the top-left corner of the graph, the better is the performance of the corresponding model. Therefore, it is clear that as the number of features is increased beyond 100, the performance continues to improve at a good rate. The curves for 300, 500 and 700 features, on the other hand, lie very close to each other.

When the dataset is not balanced, ROC curve alone is not able to identify the relevance of selected features. Precision recall (PR) curve is of more significance in this case [76]. We have plotted the PR curves for Antigenic* and Antigenic in Fig. 3b and d respectively. The closer a PR curve is to the top-right corner of the graph, the better is the performance of the corresponding model. From this analysis too, we observe that the performance increases with the increased number of features up to a certain point. As the number of features increases, the return on the performance gradually diminishes.

In Fig. 4 we plot the auROC, auPR, accuracy, sensitivity, specificity and MCC of models that are created with varying number of top-ranked features. We have considered the unbalanced training set (Fig. 4a–c) as well as training set balanced using random undersampling (Fig. 4d–f). As both sets of experiments yielded similar pattern of results, we only describe the experiments with the balanced training set. At first we explored a large feature space, but we varied the number of features with coarse granularity. That means, the number of features that are added (removed) between experiments was large. As an example, Fig. 4d is generated by starting with a model with 500 top-ranked features. The SVM-RFE (coarse) feature ranking was used in this case. Then 1000 next ranked features were added in each iteration. Based on the curves, the feature space range [100, 1500] seemed promising. Therefore, more models were generated in this space, however the change of features in each step was made finer: 100 features. We thus examine 15 models, whose performances are recorded in Fig. 4e. Moving along, we kept zooming in the interesting terrain of the feature space and increased our thoroughness in investigating the narrowed-down spaces. Fig. 4 f examines 60 models, with 10 feature increase steps. Based on the performance comparison of these models, the model built with 490 features was chosen to be our final classifier (*Antigenic*). Among the features, there were 181 n-grams, 170 nGDip and 139 PSN features. On the other hand, the final model trained with the unbalanced dataset, *Antigenic**, consisted of 500 features.

Another important observation comes out of these experiments. That is, balancing the training dataset helps make the classifier behave in a more balanced fashion. The model trained this way does not show any bias towards a particular class. Antigenic* models, on the other hand, are clearly biased towards the negative class. The specificity is much higher compared to the sensitivity in these models. The overall

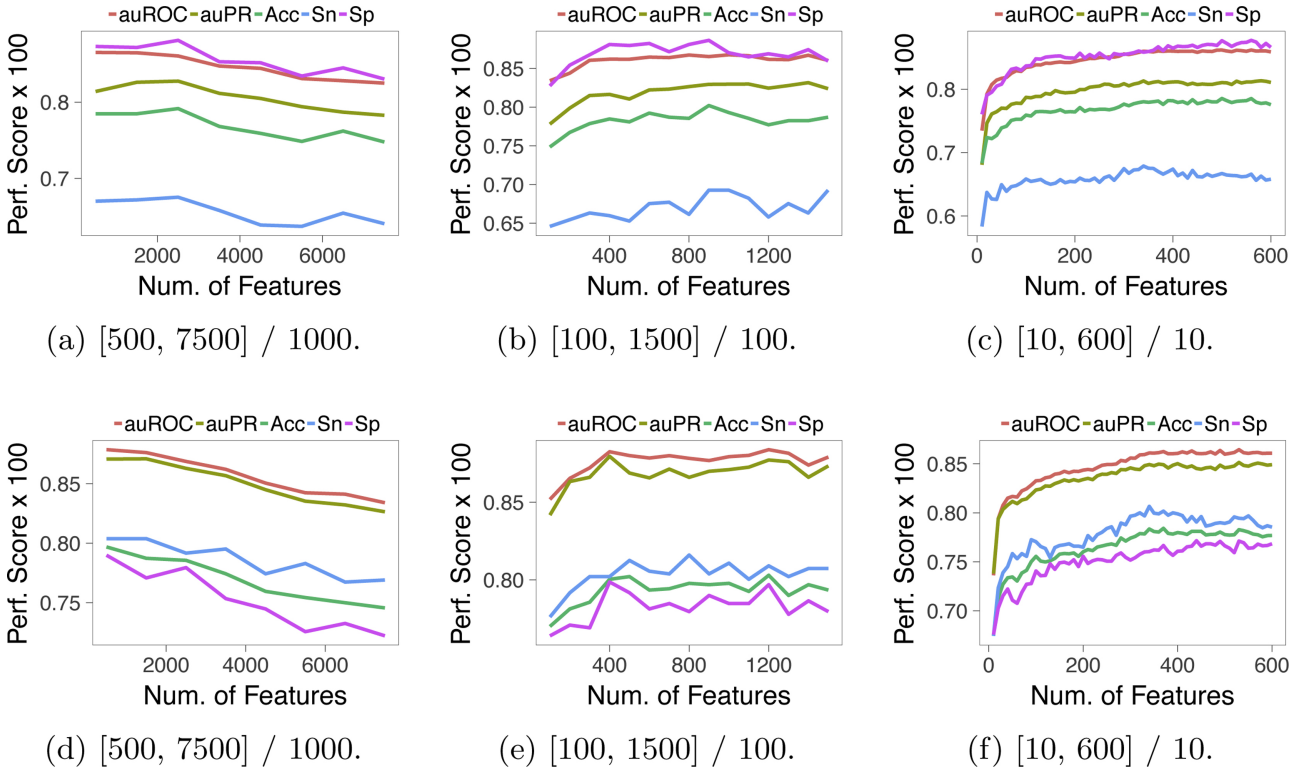


Fig. 4. Area under ROC-curve (auROC), Area under PR-curve (auPR), accuracy (Acc), sensitivity (Sn), and specificity (Sp) of models with varying number of features. The models were generated by 10-fold cross validation. For the top curves, the entire (unbalanced) training set was used (*Antigenic** models). For the bottom curves, training set was balanced with random undersampling (*Antigenic* models). The [*x*, *y*]/*z* style annotation of each sub-figure means that, the experiment started with *x* top-ranked features. Then a model was trained with *z* more features and the performance scores were recomputed. This process continued until the feature count became *y*. (The reader is referred to the web version of this article for interpretation of the references to color in this figure legend.)

accuracy is naturally dictated by the specificity and is somewhat misleading. Therefore, it is reasonable to claim that balancing the training dataset has a clear positive impact on the overall performance of our predictor. While [77] claims that the data balancing results in an improvement that is not statistically significant, the authors there analyzed performance solely based on auROC. However, when dealing with data imbalance, analyzing the PR curve is more important [76]. From Fig. 4b and e, the minimum, average and maximum increase in auPR, due to balancing the dataset, were respectively 4%, 6% and 8%.

Notably, in the above experiments, we have reported scores that are averaged over 5 different runs. As 10-fold cross validation results may vary based on how the data is partitioned, 5 runs with different data partitioning were conducted and the average score was taken to have more confidence on the result.

3.2. Impact of feature extraction techniques

To analyze the contribution of the different feature extraction techniques in building the model, we ran some experiments with the top 500 features. We trained a model with these features after balancing the training set with random undersampling. Using the same balanced data, we trained 3 more models using top 500 features of the 3 individual feature extraction techniques separately. In Fig. 5a, the accuracy, sensitivity, specificity and MCC values from these four models are compared. The nGDip feature extraction technique is a clear winner over the other two, while the combination of all performs slightly better than that.

In yet another experiment, we used combination of two feature spaces, leaving the other feature space out. Like before, we chose the top 500 features to construct the model. We compared the performance of the three generated models with that of the model created using the combination of all 3 feature spaces. The results are shown in Fig. 5b. The composition of each combination is tabulated below:

Id.	Feature spaces	n-grams	nGDip	PSN
COMB1	n-grams, nGDip	258	242	–
COMB2	n-grams, PSN	283	–	217
COMB3	nGDip, PSN	–	269	231
COMB	n-grams, nGDip, PSN	186	181	133

It is clear from Fig. 5b that among the 2 feature space combinations, the combination of n-grams and nGDip is the best. Nonetheless, adding the PSN feature space clearly adds value – the model constructed with combination of all 3 feature spaces is superior to models built with 2 feature space combinations in terms of each performance metric we have used.

3.3. Feature importance visualization

As discussed earlier, we have used *mean decrease in accuracy* to estimate the importance of each feature. The larger this value is for a feature, the more important that feature is in the context of the prediction task. The importance score of top 25 features of Antigenic is demonstrated in Fig. 6. The feature names are encoded as follows:

- A feature starting with the prefix "G_" is an nGDip feature. The integer that follows is the particular gap being considered. The dipeptide in question is given as the suffix. Therefore, the feature "G_22_SS" represents the normalized frequency of dipeptide "SS", such that the residues are separated from each other by 22 residues.
- A feature starting with the prefix "C_0_" is an n-grams feature. The suffix represents the particular n-gram. Therefore, the feature

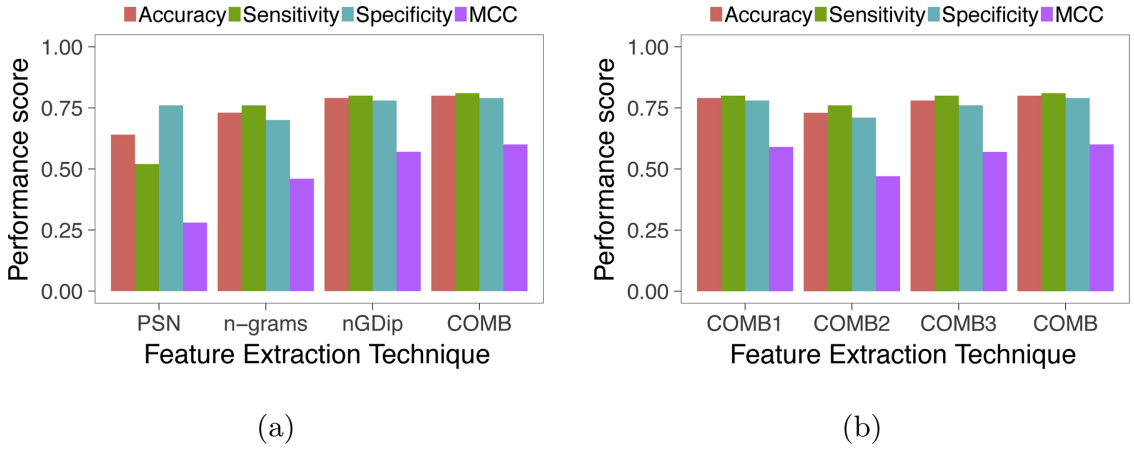


Fig. 5. Performance of different feature extraction techniques. The results are obtained from 10-fold cross validation after balancing the training dataset with random undersampling. PSN: position specific n-grams; n-grams: combination of AAC, dipeptide and tripeptide composition; nGDip: n-gapped-dipeptides; COM: combination of all the feature extraction techniques; COM1: combination of n-grams and nGDip; COM2: Combination of n-grams and PSN; COM3: combination of nGDip and PSN. (The reader is referred to the web version of this article for interpretation of the references to color in this figure legend.)

"C_0_TT" represents the normalized frequency of the dipeptide "TT".

- A feature starting with the prefix "P_" is a PSN feature. The integer that follows is the particular position. The n-gram in question is given as the suffix. Therefore, the feature "P_2_LV" represents whether the dipeptide "LV" occurs in the second position of the protein sequence.

3.4. 10-fold cross validation results

Table 2 records the performance of variations of our proposed models in 10-fold cross validation tests. In all cases, we used a decision threshold of 0.5. The results are averaged over 5 runs. The standard deviation (SD) in the different performance scores are shown after the ± sign. The fact that the SD in each metric is very small vouches for the reliability of the average score. The performance of VaxiJen and ANTIGENpro, as obtained from [2] are also recorded in the same table. The best values are highlighted in bold faced font. It is clear from this tabulated data that our models outperform the state-of-the-art predictors. Also, we observe that the RFE based versions of Antigenic and Antigenic* are superior to the mRMR based versions. Antigenic (RFE) has superior accuracy, sensitivity, specificity, MCC and auROC compared to ANTIGENpro. It also performs better than VaxiJen in all metrics except for sensitivity. VaxiJen has a commendable sensitivity of almost 90%. However, it has poor specificity (26%), thus rendering itself as a predictor very much biased towards the positive class. This also means that it has poor precision. Antigenic (RFE), on the other hand, provides a balanced performance (79% sensitivity vs. 77% specificity). This is why Antigenic (RFE) is better than Antigenic* (RFE) as well, albeit both demonstrate similar performance in terms of MCC and auROC. But Antigenic* (RFE) is biased towards the negative class (67% sensitivity vs. 88% specificity). Though it has the best accuracy among the lot, the accuracy is overestimated due to its bias towards the majority (negative) class, as it was trained with unbalanced dataset. For models trained with unbalanced datasets, the auPR is a good metric for performance comparison [76]. But, neither VaxiJen nor ANTIGENpro reports this metric. We nonetheless compare our own models in terms of auPR - while Antigenic* (RFE) has an auPR of 0.81 ± 0.004, Antigenic (RFE)'s auPR is a superior 0.85 ± 0.009. Therefore, based on the performance scores reported in Table 2 and the qualitative arguments given above, we can conclude that Antigenic (RFE) is the best predictor.

3.5. Leave one protein set out cross-validation results

Magnan et al. [2] conducted another interesting cross-validation to

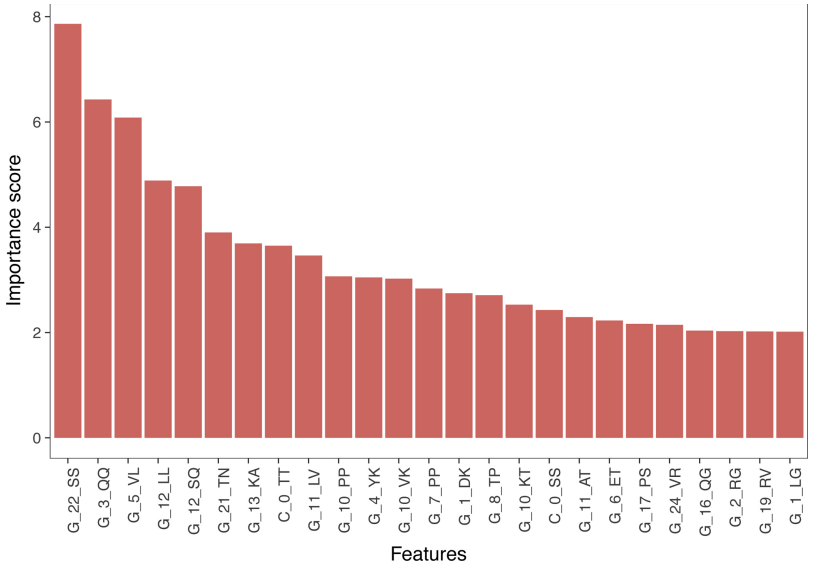


Fig. 6. The importance score of top 25 features.

Table 2
Comparison of variations of Antigenic with VaxiJen and ANTIGENpro based on 10-fold cross-validation. In terms of accuracy and specificity, Antigenic* (RFE) beats all the methods, including both versions of Antigenic. However, the lack of balance between the class-wise performance of the former is evident from the huge difference between the second and third column. The same lack of balance is present in Vaxijen, albeit more strikingly.

Method	Accuracy	Sensitivity	Specificity	MCC	auROC
Vaxijen	59.48 ± 0.140	**89.69** ± 0.000	25.85 ± 0.742	0.20 ± 0.008	0.67 ± 0.006
ANTIGEN-pro	75.51 ± 0.992	75.88 ± 1.937	75.14 ± 1.480	0.51 ± 0.020	0.81 ± 0.012
Antigenic* (RFE)	**78.55** ± 0.005	66.70 ± 0.010	**87.67** ± 0.005	**0.56** ± 0.011	**0.86** ± 0.002
Antigenic* (mRMR)	75.66 ± 0.005	67.15 ± 0.009	82.22 ± 0.009	0.50 ± 0.010	0.83 ± 0.002
Antigenic (RFE)	78.04 ± 0.008	78.99 ± 0.004	77.08 ± 0.018	**0.56** ± 0.017	**0.86** ± 0.003
Antigenic (mRMR)	75.26 ± 0.006	77.81 ± 0.005	72.71 ± 0.005	0.51 ± 0.012	0.83 ± 0.006

assess the performance of ANTIGENpro. Recall that the training dataset consisted of antigens and non-antigens from 5 different pathogens and another subset of antigens obtained from literature. In this cross-validation approach, they left one subset out, trained the predictor with the remaining samples, then tested its performance using the subset that was left out. Thus each of the 6 subsets were used as testing set in 6 different iterations. The training set (i.e. the combination of remaining 5 subsets) was balanced using random undersampling before the training step. In addition, the testing set was also balanced using random undersampling. This was done to ensure a fair estimation of the predictor performance. An exception however was made when PAntigens subset was used as the testing set. Since this set does not have any non-antigens at all, undersampling cannot be done. Therefore, this set was used unaltered during testing. We have followed the same approach to measure the performance of our models and have a comparison with ANTIGENpro. We refer to this as *Leave one protein set out* cross-validation. In case of Antigenic, the training data was balanced using random undersampling. For the Antigenic* model, the training data was not balanced. In all the experiments, we kept the decision threshold at 0.5.

The results of this cross-validation approach are recorded in Table 3. For our models, the accuracy scores are averaged over 5 runs, with the standard deviation recorded after the ± sign. Once again, the small standard deviations gives confidence on the average scores. For ANTIGENpro, the average scores were obtained from [2], but the corresponding standard deviations were unavailable. When PAntigens is used as the testing set, ANTIGENpro has an formidable average accuracy of 81.60%. Antigenic (mRMR) surpasses that, achieving 86.29% accuracy on average. Antigenic (RFE) is not far behind, logging an average accuracy score of 80.28%. In fact, in 2 runs the accuracy scores were 82.63% and 81.69%, which are better than the reported value of ANTIGENpro. The performance on the PAntigens test set demonstrates that our featuring scheme, combined with the prediction algorithm, is able to predict protective antigens by learning solely from protein microarray data.

When the Brucella subset was used, Antigenic (RFE) produced the best performance. For Burkholderia and Candida test sets, Antigenic* (RFE) and ANTIGENpro respectively demonstrated the best performance. However, since these testing sets were very small, no conclusions should be made based on these results. For the Malaria test set, ANTIGENpro performed significantly better than our models. For the Tuberculosis test set, Antigenic (RFE) is the winner with 70% accuracy, but ANTIGENpro is not far away (68.3% accuracy).

Another observation that we can make from these experiments is regarding the benefit of balancing the training dataset. In case of the PAntigens test set, Antigenic* has a poor accuracy – around 30% and 42% respectively in the RFE and mRMR based versions. As argued earlier, these models are quite biased towards the negative class. And since the testing set did not have any negative instances at all, the accuracy merely reflects the sensitivity, which is poor. In this setting the negative class is twice as large as the positive class. On the other hand, when the full training dataset was used during 10-fold cross-validation (Table 2), the imbalance ratio was 1:1.3, yielding 67% sensitivity.

3.6. Jackknife cross-validation results

Neither VaxiJen nor ANTIGENpro has reported jackknife cross-validation results. However, for completeness and to enable comparison with future predictors, we report the jackknife cross-validation performance of our models in Table 4. Like before, the best scores are highlighted in bold-face. Among all the models, Antigenic (RFE) prevails as superior.

As discussed in Section 2.4.1, jackknife cross-validation always produces unique result, which is a key advantage of this technique. 10-fold cross validation results, on the other hand, may vary depending on how folds are constructed. In the 10-fold cross validation, Antigenic (RFE) had the best sensitivity among all our models. However, Antigenic* (RFE) recorded superior accuracy and both method logged the same MCC and auROC. In jackknife testing, on the contrary, Antigenic (RFE) demonstrated superior accuracy, sensitivity, MCC, auROC and auPR. Since the jackknife test cannot be biased by any particular way of splitting the data for cross-validation, the performance results obtained in this testing should therefore be given preference over the 10-fold cross-validation results. Also, since the training dataset in our case guarantees that pairwise sequence similarity is no more than 30%, any concerns of overestimation in jackknife approach is reasonably mitigated [57].

Now that we have established the superiority of RFE based versions of Antigenic over the mRMR based versions, the experiments that follow have been conducted with the RFE based versions only.

Table 3
Comparison of accuracy between Antigenic and ANTIGENpro based on leave one protein set out cross-validation.

Test set	Test set	ANTIGEN	Antigenic*		Antigenic	
	Size	pro	(RFE)	(mRMR)	(RFE)	(mRMR)
PAN	213	81.60	29.77 ± 0.016	41.78 ± 0.006	80.28 ± 0.019	**86.29** ± 0.015
BRU	140	70.00	72.71 ± 0.034	66.57 ± 0.033	**73.00** ± 0.014	67.00 ± 0.018
BUR	10	66.00	**70.00** ± 0.071	62.00 ± 0.084	64.00 ± 0.055	62.00 ± 0.110
CAN	6	**66.67**	43.33 ± 0.253	30.00 ± 0.139	46.67 ± 0.075	43.33 ± 0.091
MAL	228	**59.96**	52.72 ± 0.008	51.75 ± 0.010	51.93 ± 0.007	51.93 ± 0.006
TUB	342	68.30	69.94 ± 0.019	65.96 ± 0.008	**70.00** ± 0.009	66.96 ± 0.007

Table 4
Jackknife cross-validation performance of Antigenic* and Antigenic.

↓ prediction method	Accuracy	Sensitivity	Specificity	Precision	MCC	auROC	auPR
Antigenic*							
(RFE)	79.15	67.71	**87.97**	**81.25**	0.57	0.87	0.82
(mRMR)	75.98	68.75	81.55	74.16	0.51	0.83	0.79
Antigenic							
(RFE)	**80.03**	**80.90**	79.16	79.52	**0.60**	**0.88**	**0.87**
(mRMR)	76.04	79.86	72.22	74.19	0.52	0.83	0.81

Table 5
Comparison of Antigenic with VaxiJen and ANTIGENpro based on independent testing.

Method	Accuracy	Sensitivity	Specificity
VaxiJen	39.71	72.60	37.99
ANTIGENpro	56.94	65.75	56.47
Antigenic*	**61.18**	61.64	**61.15**
Antigenic	46.27	**76.71**	44.68

Table 6
Area under ROC and PR curves for different predictors on the Bartonella dataset.

Method	auROC	auPR
VaxiJen	0.603	0.074
ANTIGENpro	0.671	**0.143**
Antigenic*	**0.674**	0.136
Antigenic	0.662	0.125

3.7. Independent test results

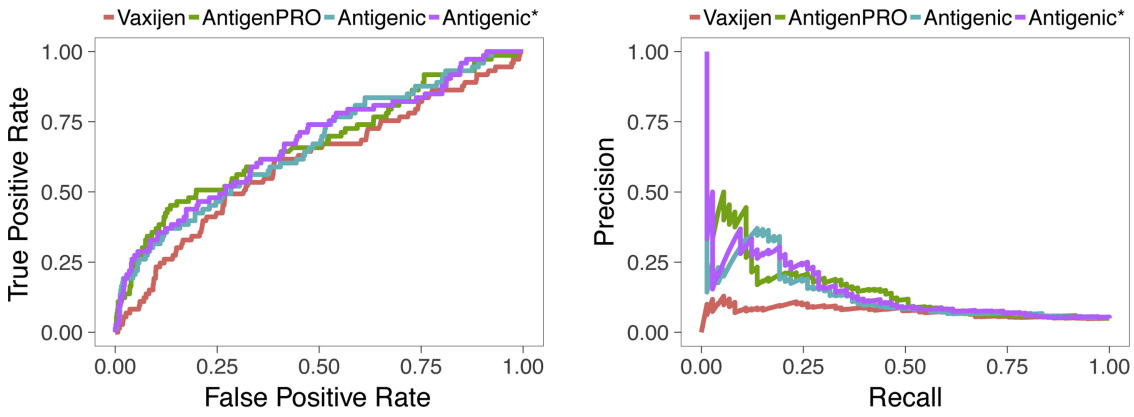
In Table 5, independent testing performance of different predictors are recorded. The entire proteome of *B. henselae* pathogen has been used for independent testing. As mentioned earlier, it contains 1463 proteins, of which only 73 are protective antigens. The FASTA file containing the proteome was easily uploaded to the VaxiJen server [45]. The prediction results were obtained in a response webpage within minutes of the query. The publicly available ANTIGENpro web tool [51] is less friendly for bulk queries. Single protein sequence can be pasted in a form and submitted. After some time the prediction results are provided via an email response. We wrote a simple Java program to automatically query the tool for each sequence of the proteome. Between queries, one minute waiting time was added so that the server does not get flooded with a lot of query in a short period of time. The response emails were also processed through code written in Java. Getting the ANTIGENpro predictions for the Bartonella proteome this way took approximately 2 days. The results were obtained in a few minutes in case of Antigenic* and Antigenic.

In each case, we have considered the default class discriminating threshold. For VaxiJen, it is 0.4, for all others it is 0.5. As seen from Table 5, Antigenic is more sensitive than all other tools (even VaxiJen) at the default threshold. Its specificity is better than that of VaxiJen, but worse than that of ANTIGENpro and Antigenic*. Since the proteome is extremely imbalanced, the inferior specificity also impacts the overall accuracy. Surprisingly, Antigenic* demonstrates the most balanced performance in the independent testing.

Fig. 7 shows the ROC curve for VaxiJen, ANTIGENpro, Antigenic* and Antigenic. Since Antigenic* is built using an imbalanced dataset, the PR curves of all the tools are also drawn in the same figure for better comparative assessment. Performance of ANTIGENpro is the best, while VaxiJen is the least performing. The area under ROC curve for ANTIGENpro, as recorded in Table 6, is marginally less than that of Antigenic* and slightly larger than that of Antigenic. It has the best auPR score (0.143). Antigenic* and Antigenic are not too far behind, however. VaxiJen, on the other hand, has a modest 0.074 unit area under PR curve.

In 10-fold cross-validation testing, Antigenic demonstrated superiority over the state-of-the-art predictors. Its superior performance was also corroborated by the jackknife cross-validation testing. While it did not produce the best results in the independent testing, we have, as argued by Chou [57], preferred the cross-validation results over the independent test results and concluded Antigenic to be the superior predictor.

The success of reverse vaccinology relies heavily on how efficiently and precisely the predictors can find protective antigens [78,79]. This is why the predictors are assessed in terms of yet another metric, known as *enrichment*. It is the ratio of number of protective antigens among a



(a) ROC-curve. (b) PR-curve.

Fig. 7. ROC and PR curves on the independent test for different prediction tools. (The reader is referred to the web version of this article for interpretation of the references to color in this figure legend.)

Table 7
Enrichment among top ranked proteins of Bartonella dataset, ranked by different predictors.

Method	SignalP	VaxiJen	ANTIGENpro	Antigenic*	Antigenic
Top ranked 2%	1.7	2.1	5.5	6.2	**6.9**
Top ranked 5%	2.7	1.6	4.1	**4.9**	3.8
Top ranked 10%	2.9	1.9	**3.4**	3.3	3.0
Top ranked 25%	**2.2**	1.6	2.0	**2.2**	1.8

top ranked subset to the number of protective antigens in the entire proteome. The expected enrichment of a random predictor is thus 1.0. For any good predictor, enrichment would be much higher than 1.0 in any top ranked subsets. Table 7 compares the enrichment of our predictors in independent testing (i.e. on the Bartonella dataset) with that of VaxiJen, ANTIGENpro and SignalP in top ranked 2%, 5%, 10% and 25% subsets. The data for SignalP was obtained from [2]. The data for VaxiJen and ANTIGENpro were computed based on the prediction scores, as obtained from their respective servers [45,51]. While the scores for VaxiJen matched what was reported in [2], there was slight variation in the scores for ANTIGENpro. The enrichment for top ranked 5% turned out to be 4.1 instead of 4.4; and for top ranked 25% turned out to be 2.0 instead of 2.1. Both Antigenic* and Antigenic had superior enrichment in the top ranked 2% subset. In particular, Antigenic scored 6.9 which is much higher than ANTIGENpro's 5.5. This means, if a practitioner ranks all the proteins of a new pathogen using Antigenic and selects only one protein at random from the top 2% for wet lab testing, his chance of identifying a protective antigen is almost 7 times higher than if he were to select one protein at random from the entire proteome. Therefore, our predictor seems quite suitable for wide adoption in reverse vaccinology based projects.

3.8. Discussion

Antigenic demonstrated superior performance compared to other predictors in 10-fold cross-validation. It showed good performance in the leave one protein set out cross-validation as well. Like ANTIGENpro, it too demonstrated the ability to recognize protective antigens by learning the classifier from a training dataset that is prepared solely from the protein microarray data. In case of independent testing, it showed superior sensitivity. It also showed better enrichment in the top ranked 2% subset. It did not hold the best auROC or auPR in the independent test. The best auPR was obtained by ANTIGENpro, which also demonstrated good sensitivity. However, preference should be given on the cross-validation results over the independent test results in comparing different predictors [57]. Since Antigenic outperforms ANTIGENpro in the 10-fold cross-validation on the same training dataset, we consider Antigenic to be the superior predictor.

While VaxiJen has widely been adopted in various reverse vaccinology projects, it was able to correctly classify only 59.48% of the bacterial and viral proteins in the benchmark training dataset. The other antigens (around 25% of the antigens in the training set) could not be tested since no prediction model is available for these pathogens. This is a clear shortcoming of Vaxijen. Antigenic, on the other hand, provides a generic classification model for any pathogens and is able to demonstrate superior prediction performance.

Additionally, Antigenic has a fairly simple prediction model. The features it rely on are extracted from the protein's primary sequence directly. On the other hand ANTIGENpro has a relatively complex model. It uses eight different feature sets, six of which are frequencies of amino acid monomers and dimers using three different amino acid alphabets. The remaining two feature sets are computed and predicted features. The computed features include sequence length, turn-forming residues fraction, absolute charge per residue, molecular weight, GRAVY Index [80], Aliphatic index [81]. Predicted features are obtained from external predictors like SSpro [82], DOMpro [83], ACCPro [82], and TMHMM [84]. Using these eight feature sets, forty distinct primary classifiers are then trained using one algorithm from Naive Bayes, C4.5, k-nearest neighbors, neural networks and SVMs. The 40 probability estimates thus obtained are then fed as input to a second stage SVM classifier. Perhaps it is because of the external dependency that ANTIGENpro limits query submissions to a single protein at a time. Antigenic, on the other hand, can handle large files with multiple protein sequences, making it convenient to use for whole proteome analysis.

The novelty of Antigenic lies in the addition of gapped dipeptides, tripeptide composition and PSN features into Chou's general PseAAC. The combination of Random Forest followed by SVM-RFE for feature selection is also a new approach in this prediction problem. Another distinguishing factor is that we explored a large feature space, comprising 32467 features and then selected 490 features for training the model. In contrast, VaxiJen used only 45 features, while ANTIGENpro used a total of 768 features.

4. Conclusion

In this paper, we have presented Antigenic, a machine learning based predictor for protective antigens. We applied three different feature extraction techniques on a benchmark dataset that was primarily prepared from protein microarray data. Represented in a discrete model known as Chou's general PseAAC, the proteins were then subjected to Random Forest algorithm, followed by either SVM-RFE method or mRMR criterion, to obtain a reliable ranking of the features. Finally, Random Forest was employed to learn a prediction model using a top-ranked feature subset. As the training dataset was not balanced, random undersampling was performed to balance the data. We trained models with both the unbalanced and balanced dataset and found the latter to be superior. We compared the SVM-RFE based model to the mRMR based model and the former was the winner. Finally, Our approach outperformed state-of-the-art techniques according to different performance metrics in 10-fold cross-validation. The independent test results were also found to be satisfactory.

Our predictor is available as an R script that can readily be applied to target protein sequences, without dependency on any other services or pre-processing. Antigenic is also available as a publicly accessible web based predictor. We hope the simple to use web interface, combined with the good performance, will lead to wide adoption of Antigenic. At the same time, we hope that our simple and lightweight framework will trigger further research using this in similar other domains. In future, we plan to augment Antigenic by utilizing other sequence based features, such as amino acid physicochemical properties, polarity, grouped amino acid frequencies, etc. Such features have successfully been used in other prediction tasks recently [85–87]. In particular, web services such as *Pse-in-One* [88] and its updated version *Pse-in-One 2.0* [89] can be used to generate any desired feature vectors conveniently. We also plan to apply more recent methods for feature selection, such as Max-Relevance-Max-Distance (MRMD) [90], in our model construction pipeline. Also, other novel classifiers, such as XGBoost [91], LibD3C [92] etc., should be experimented within this context.

Conflict of interest

None declared.

Acknowledgement

We thank Dr. Christophe N. Magnan for providing us with the dataset used in [2]. This dataset was used to train and test Antigenic. We are grateful to Dr. Muhammad Sougat Islam and Mr. Arif Khan for an earlier discussion that lead us to this research. We also thank the

anonymous reviewers and the editor for their constructive comments and for providing us with a number of relevant references.

References

[1] Flower DR, Macdonald IK, Ramakrishnan K, Davies MN, Doytchinova IA. Computer aided selection of candidate vaccine antigens. Immunome Res 2010;6(2):S1.
[2] Magnan CN, Zeller M, Kayala MA, Vigil A, Randall A, Felgner PL, Baldi P. High-throughput prediction of protein antigenicity using protein microarray data. Bioinformatics 2010;26(23):2936–43.
[3] Rana A, Akhter Y. A multi-subunit based, thermodynamically stable model vaccine using combined immunoinformatics and protein structure based approach. Immunobiology 2016;221(4):544–57.
[4] Gilchuk P, Knight FC, Wilson JT, Joyce S. Eliciting epitope-specific cd8+ t cell response by immunization with microbial protein antigens formulated with α-galactosylceramide: theory, practice, and protocols. Vaccine Adjuvants. Springer; 2017. p. 321–52.
[5] Longley RJ, Halbroth BR, Salman AM, Ewer KJ, Hodgson SH, Janse CJ, Khan SM, Hill AV, Spencer AJ. Assessment of the plasmodium falciparum preerythrocytic antigen UIS3 as a potential candidate for a malaria vaccine. Infect Immun 2017;85(3):e00641–16.
[6] Patronov A, Doytchinova I. T-cell epitope vaccine design by immunoinformatics. Open Biol 2013;3(1):120139.
[7] Liesegang TJ. Varicella zoster virus vaccines: effective, but concerns linger. Can J Ophthalmol 2009;44(4):379–84.
[8] Fiore AE, Bridges CB, Cox NJ. Seasonal influenza vaccines. Vaccines for Pandemic Influenza. Springer; 2009. p. 43–82.
[9] Chang Y, Brewer NT, Rinas AC, Schmitt K, Smith JS. Evaluating the impact of human papillomavirus vaccines. Vaccine 2009;27(32):4355–62.
[10] WHO, UNICEF, World Bank. State of the world's vaccines and immunization. 3rd ed. Geneva: WHO; 2009.
[11] Arinaminpathy N, Ratmann O, Koelle K, Epstein SL, Price GE, Viboud C, Miller MA, Grenfell BT. Impact of cross-protective vaccines on epidemiological and evolutionary dynamics of influenza. Proc Natl Acad Sci U S A 2012;109(8):3173–7.
[12] Rappuoli R, Pizza M, Del Giudice G, De Gregorio E. Vaccines, new opportunities for a new society. Proc Natl Acad Sci U S A 2014;111(34):12288–93.
[13] WHO. MDG 6: combat HIV/AIDS, malaria and other diseases. Geneva: WHO; 2014.
[14] Ada G. The traditional vaccines: an overview. New Gen Vac 1997:12–23.
[15] Altindis E, Cozzi R, Di Palo B, Necchi F, Mishra RP, Fontana MR, Soriani M, Bagnoli F, Maione D, Grandi G, et al. Protectome analysis: a new selective bioinformatics tool for bacterial vaccine candidate discovery. Mol Cell Proteom 2015;14(2):418–29.
[16] Rappuoli R. Reverse vaccinology. Curr Opin Microbiol 2000;3(5):445–50.
[17] Szmuness W, Oleszko W, Stevens C, Goodman A. Passive. active immunisation against hepatitis B: immunogenicity studies in adult Americans. Lancet 1981;317(8220):575–7.
[18] Szmuness W, Stevens CE, Harley EJ, Zang EA, Taylor PE, Alter HJ. The immune response of healthy adults to a reduced dose of hepatitis B vaccine. J Med Virol 1981;8(2):123–9.
[19] Jackwood MW, Hickle L, Kapil S, Silva R, Osterrieder K, Prideaux C, Schultz R, Bell A. Vaccine development using recombinant DNA technology. 2008.
[20] Soria-Guerra RE, Nieto-Gomez R, Govea-Alonso DO, Rosales-Mendoza S. An overview of bioinformatics tools for epitope prediction: implications on vaccine development. J Biomed Inform 2015;53:405–14.
[21] Doytchinova IA, Flower DR. Identifying candidate subunit vaccines using an alignment-independent method based on principal amino acid properties. Vaccine 2007;25(5):856–66.
[22] Woodrow G. An overview of biotechnology as applied to vaccine development. New Gen Vac 1997;25.
[23] Pizza M, Scarlato V, Masignani V, Giuliani MM, Aricò B, Comanducci M, Jennings GT, Baldi L, Bartolini E, Capecchi B, et al. Identification of vaccine candidates against serogroup B meningococcus by whole-genome sequencing. Science 2000;287(5459):1816–20.
[24] Vernikos G, Medini D. Bexsero® chronicle. Pathog Global Health 2014;108(7):305–16.
[25] Folaranmi T, Rubin L, Martin SW, Patel M, MacNeil JR. Use of serogroup B meningococcal vaccines in persons aged = 10 years at increased risk for serogroup B meningococcal disease: recommendations of the advisory committee on immunization practices, 2015. MMWR Morb Mortal Wkly Rep 2015;64(22):608–12.
[26] Chakravarti DN, Fiske MJ, Fletcher LD, Zagursky RJ. Application of genomics and proteomics for identification of bacterial gene products as potential vaccine candidates. Vaccine 2000;19(6):601–12.
[27] Wizemann TM, Heinrichs JH, Adamou JE, Erwin AL, Kunsch C, Choi GH, Barash SC, Rosen CA, Masure HR, Tuomanen E, et al. Use of a whole genome approach to identify vaccine molecules affording protection against streptococcus pneumoniae infection. Infect Immun 2001;69(3):1593–8.
[28] Ross BC, Czajkowski L, Hocking D, Margetts M, Webb E, Rothel L, Patterson M, Agius C, Camuglia S, Reynolds E, et al. Identification of vaccine candidate antigens from a genomic analysis of *Porphyromonas gingivalis*. Vaccine 2001;19(30):4135–42.
[29] Montigiani S, Falugi F, Scarselli M, Finco O, Petracca R, Galli G, Mariani M, Manetti R, Agnusdei M, Cevenini R, et al. Genomic approach for analysis of surface proteins in *Chlamydia pneumoniae*. Infect Immun 2002;70(1):368–79.
[30] Ariel N, Zvi A, Grosfeld H, Gat O, Inbar Y, Velan B, Cohen S, Shafferman A. Search for potential vaccine candidate open reading frames in the *Bacillus anthracis* virulence plasmid pxo1: in silico and in vitro screening. Infect Immun 2002;70(12):6817–27.
[31] Baldwin SL, Reese VA, Po-wei DH, Beebe EA, Podell BK, Reed SG, Coler RN. Protection and long-lived immunity induced by the id93/gla-se vaccine candidate against a clinical *Mycobacterium tuberculosis* isolate. Clin Vac Immunol 2016;23(2):137–47.
[32] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990;215(3):403–10.
[33] Pearson WR. Rapid and sensitive sequence comparison with FASTP and FASTA. 1990.
[34] Nakai K, Horton P. Psort: a program for detecting sorting signals in proteins and predicting their subcellular localization. 1999.
[35] Nielsen H. Predicting secretory proteins with Signalp. Protein Funct Predict: Methods Protocols 2017:59–73.
[36] Nielsen H, Engelbrecht J, Brunak S, von Heijne G. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Protein Eng 1997;10(1):1–6.
[37] Nielsen H, Krogh A. Prediction of signal peptides and signal anchors by a hidden Markov model. ISMB, vol. 6 1998:122–30.
[38] Bendtsen JD, Nielsen H, von Heijne G, Brunak S. Improved prediction of signal peptides: Signalp 3.0. J Mol Biol 2004;340(4):783–95.
[39] Petersen TN, Brunak S, von Heijne G, Nielsen H. Signalp 4.0: discriminating signal peptides from transmembrane regions. Nat Methods 2011;8(10):785.
[40] Vivona S, Bernante F, Filippini F. Nerve: new enhanced reverse vaccinology environment. BMC Biotechnol 2006;6(1):35.
[41] He Y, Xiang Z, Mobley HL. Vaxign: the first web-based vaccine design program for reverse vaccinology and applications for vaccine development. Biomed Res Int 2010.
[42] Doytchinova IA, Flower DR. Vaxijen: a server for prediction of protective antigens, tumour antigens and subunit vaccines. BMC Bioinf 2007;8(1):4.
[43] Hellberg S, Sjoestroem M, Skagerberg B, Wold S. Peptide quantitative structure–-activity relationships, a multivariate approach. J Med Chem 1987;30(7):1126–35.
[44] Wold S, Jonsson J, Sjörström M, Sandberg M, Rännar S. Dna and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures. Anal Chim Acta 1993;277(2):239–53.
[45] VaxiJen v2.0, http://www.ddg-pharmfac.net/vaxijen/VaxiJen/VaxiJen.html [last accessed on 25.03.2018].
[46] Jaiswal V, Chanumolu SK, Gupta A, Chauhan RS, Rout C. Jenner-predict server: prediction of protein vaccine candidates (PVCS) in bacteria based on host–pathogen interactions. BMC Bioinf 2013;14(1):211.
[47] Lee JS, Shin SJ, Collins MT, Jung ID, Jeong Y-I, Lee C-M, Shin YK, Kim D, Park Y-M. *Mycobacterium avium* subsp. *paratuberculosis* fibronectin attachment protein activates dendritic cells and induces a th1 polarization. Infect Immun 2009;77(7):2979–88.
[48] Loosmore SM, Yang Y-p, Oomen R, Shortreed JM, Coleman DC, Klein MH. The haemophilus influenzae HTRA protein is a protective antigen. Infect Immun 1998;66(3):899–906.
[49] Doytchinova IA, Flower DR. Bioinformatic approach for identifying parasite and fungal candidate subunit vaccines. Open Vac J 2008;1(1):4.
[50] Ansari HR, Flower DR, Raghava G. Antigendb: an immunoinformatics database of pathogen antigens. Nucleic Acids Res 2009;38(suppl_1):D847–53.
[51] SCRATCH Protein Predictor, http://scratch.proteomics.ics.uci.edu/cgi-bin/new_server/sql_predict.cgi [last accessed on 25.03.2018].
[52] El-Manzalawy Y, Dobbs D, Honavar V. Predicting protective bacterial antigens using random forest classifiers. Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine 2012:426–33. ACM.
[53] Shi J, Zhang S, Liang Y, Pan Q. Prediction of protein subcellular localizations using moment descriptors and support vector machine. International Workshop on Pattern Recognition in Bioinformatics 2006:105–14. Springer.
[54] Haar A. Zur theorie der orthogonalen funktionensysteme. Math Ann 1910;69(3):331–71.
[55] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.
[56] Ong E, Wong MU, He Y. Identification of new features from known bacterial protective vaccine antigens enhances rational vaccine design. Front Immunol 2017;8.
[57] Chou K-C. Some remarks on protein attribute prediction and pseudo amino acid composition. J Theor Biol 2011;273(1):236–47.
[58] Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory 1992:144–52. ACM.
[59] Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. J Bioinf Comput Biol 2005;3(02):185–205.
[60] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 2005;27(8):1226–38.
[61] Davies DH, Liang X, Hernandez JE, Randall A, Hirst S, Mu Y, Romero KM, Nguyen TT, Kalantari-Dehaghi M, Crotty S, et al. Profiling the humoral immune response to infection by using proteome microarrays: high-throughput vaccine and diagnostic antigen discovery. Proc Natl Acad Sci U S A 2005;102(3):547–52.
[62] Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res 1997;25(17):3389–402.
[63] Chou K-C. Prediction of protein cellular attributes using pseudo-amino acid composition. Protein: Struct Funct Bioinf 2001;43(3):246–55.
[64] Du P, Gu S, Jiao Y. Pseaac-general: fast building various modes of general form of Chou's pseudo-amino acid composition for large-scale protein datasets. Int J Mol Sci

2014;15(3):3495–506.
[65] Chou K-C. Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. Curr Proteom 2009;6(4):262–74.
[66] Rahman MS, Rahman MK, Kaykobad M, Rahman MS. isgpt. An optimized model to identify sub-Golgi protein types using SVM and random forest based feature selection. Artif Intell Med 2018;84:90–100.
[67] Rahman MS, Shatabda S, Saha S, Kaykobad M, Rahman MS. DPP-PseAAC: a DNA-binding protein prediction model using Chou's general PseAAC. J Theor Biol 2018;452:22–34.
[68] Bernardes JS. A review of protein function prediction under machine learning perspective. Recent Pat Biotechnol 2013;7(2):122–41.
[69] Nanni L, Lumini A, Brahnam S. An empirical study of different approaches for protein classification. Sci World J 2014.
[70] Chang J-M, Su EC-Y, Lo A, Chiu H-S, Sung T-Y, Hsu W-L. Psldoc: Protein subcellular localization prediction based on gapped-dipeptides and probabilistic latent semantic analysis. Protein: Struct Funct Bioinf 2008;72(2):693–710.
[71] Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Mach Learn 2002;46(1-3):389–422.
[72] Kohavi R, Sommerfield D, Dougherty J. Data mining using/spl mscr//spl lscr//spl cscr/++ a machine learning library in C++. Proceedings Eighth IEEE International Conference on Tools with Artificial Intelligence 1996:234–45. IEEE.
[73] Powers DM. Evaluation: from precision, recall and f-measure to ROC, informedness, markedness and correlation. 2011.
[74] Altman DG, Bland JM. Statistics notes-diagnostic-tests-1-sensitivity and specificity. 1994. p. 3.
[75] Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett 2006;27(8):861–74.
[76] Davis J, Goadrich M. The relationship between precision-recall and ROC curves. Proceedings of the 23rd international conference on machine learning 2006:233–40. ACM.
[77] Dittman DJ, Khoshgoftaar TM, Napolitano A. The effect of data sampling when using random forest on imbalanced bioinformatics data. 2015 IEEE International Conference on Information Reuse and Integration (IRI) 2015:457–63. IEEE.
[78] Rappuoli R, Aderem A. A 2020 vision for vaccines against HIV, tuberculosis and malaria. Nature 2011;473(7348):463.
[79] Jones D. Reverse vaccinology on the cusp. 2012.
[80] Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol 1982;157(1):105–32.
[81] Ikai A. Thermostability and aliphatic index of globular proteins. J Biochem 1980;88(6):1895–8.
[82] Cheng J, Randall AZ, Sweredoski MJ, Baldi P. Scratch: a protein structure and structural feature prediction server. Nucleic Acids Res 2005;33(suppl_2):W72–6.
[83] Cheng J, Sweredoski MJ, Baldi P. Dompro: protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks. Data Mining Knowled Discovery 2006;13(1):1–10.
[84] Krogh A, Larsson B, Von Heijne G, Sonnhammer EL. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes1. J Mol Biol 2001;305(3):567–80.
[85] Wan S, Duan Y, Zou Q. Hpslpred: an ensemble multi-label classifier for human protein subcellular location prediction with imbalanced source. Proteomics 2017;17(17-18):1700262.
[86] Zou Q, Wan S, Ju Y, Tang J, Zeng X. Pretata: predicting tata binding proteins with novel features and dimensionality reduction strategy. BMC Syst Biol 2016;10(4):114.
[87] Song L, Li D, Zeng X, Wu Y, Guo L, Zou Q. ndna-prot: identification of DNA-binding proteins based on unbalanced classification. BMC Bioinf 2014;15(1):298.
[88] Liu B, Liu F, Wang X, Chen J, Fang L, Chou K-C. Pse-in-one: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nucleic Acids Res 2015;43(W1):W65–71.
[89] Liu B, Wu H, Chou K-C. Pse-in-one 2.0: an improved package of web servers for generating various modes of pseudo components of dna, rna, and protein sequences. Nat Sci 2017;9(04):67.
[90] Zou Q, Zeng J, Cao L, Ji R. A novel features ranking metric with application to scalable visual and bioinformatics data classification. Neurocomputing 2016;173:346–54.
[91] Chen T, Guestrin C. Xgboost: a scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining 2016:785–94. ACM.
[92] Lin C, Chen W, Qiu C, Wu Y, Krishnan S, Zou Q. Libd3c: ensemble classifiers with a clustering and dynamic selection strategy. Neurocomputing 2014;123:424–35.