

Enhancement of conformational B-cell epitope prediction using CluSMOTE

Binti Solihah^{1,2}, Azhari Azhari¹ and Aina Musdholifah¹

¹Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia

²Department of Informatics Engineering, Universitas Trisakti, Grogol, Jakarta Barat, Indonesia

ABSTRACT

Background. A conformational B-cell epitope is one of the main components of vaccine design. It contains separate segments in its sequence, which are spatially close in the antigen chain. The availability of Ag-Ab complex data on the Protein Data Bank allows for the development predictive methods. Several epitope prediction models also have been developed, including learning-based methods. However, the performance of the model is still not optimum. The main problem in learning-based prediction models is class imbalance.

Methods. This study proposes CluSMOTE, which is a combination of a cluster-based undersampling method and Synthetic Minority Oversampling Technique. The approach is used to generate other sample data to ensure that the dataset of the conformational epitope is balanced. The Hierarchical DBSCAN algorithm is performed to identify the cluster in the majority class. Some of the randomly selected data is taken from each cluster, considering the oversampling degree, and combined with the minority class data. The balance data is utilized as the training dataset to develop a conformational epitope prediction. Furthermore, two binary classification methods, Support Vector Machine and Decision Tree, are separately used to develop model prediction and to evaluate the performance of CluSMOTE in predicting conformational B-cell epitope. The experiment is focused on determining the best parameter for optimal CluSMOTE. Two independent datasets are used to compare the proposed prediction model with state of the art methods. The first and the second datasets represent the general protein and the glycoprotein antigens respectively.

Result. The experimental result shows that CluSMOTE Decision Tree outperformed the Support Vector Machine in terms of AUC and Gmean as performance measurements. The mean AUC of CluSMOTE Decision Tree in the Kringelum and the SEPPA 3 test sets are 0.83 and 0.766, respectively. This shows that CluSMOTE Decision Tree is better than other methods in the general protein antigen, though comparable with SEPPA 3 in the glycoprotein antigen.

Subjects Bioinformatics, Data Mining and Machine Learning

Keywords Cluster-based undersampling, SMOTE, Class imbalance, Hybrid sampling, Hierarchical DBSCAN, Vaccine design

INTRODUCTION

A B-cell epitope is among the main components of peptide-based vaccines (*Andersen, Nielsen & Lund, 2006; Zhang et al., 2011; Ren et al., 2015*). It can be utilized in

Submitted 12 May 2019

Accepted 15 April 2020

Published 1 June 2020

Corresponding author

Binti Solihah,

binti.solihah@mail.ugm.ac.id

Academic editor

Sebastian Ventura

Additional Information and
Declarations can be found on
page 12

DOI 10.7717/peerj-cs.275

© Copyright

2020 Solihah et al.

Distributed under

Creative Commons CC-BY 4.0

OPEN ACCESS

immunodetection or immunotherapy to induce an immune response (*Rubinstein et al., 2008*). Many B-cell epitopes are conformational and originate from separate segments of an antigen sequence, forming a spatial neighborhood in the antigen-antibody (Ag–Ab) complex. Identifying epitopes through experiments is tedious and expensive work, and therefore, there is a high risk of failure. Current progress in bioinformatics makes it possible to create vaccine designs through 3D visualization of protein antigen. Many characteristics, including composition, cooperativeness, hydrophobicity, and secondary structure, are considered in identifying potential substances for an epitope (*Kringelum et al., 2013*). Since no dominant characteristic helps experts to easily distinguish epitopes from other parts of the antigen, the risk of failure is quite high.

The availability of the 3D structure of the Ag–Ab complex in the public domain and computational resources eases the development of predictive models using various methods, including the structure and sequence-based approaches. However, the conformational epitope prediction is still challenging. The structure-based approach can be divided into three, including dominant-characteristic-based, graph-based, and learning-based categories.

There are several characteristic-based approaches, including (1) CEP, which uses solvent-accessibility properties, (2) Discotope using both solvent-accessibility-based properties and epitope log odds ratio of amino acid, (3) PEPITO that adds half-sphere exposure (HSE) to log odds ratio of amino acid in Discotope and (4) Discotope 2.0, which is an improved version of Discotope. It defines the log odd ratios in spatial contexts and adds half-sphere exposure (HSE) as a feature, and (5) SEPPA, which utilizes exposed and adjacent residual characteristics to form a triangle unit patch (*Kulkarni-kale, Bhosle & Kolaskar, 2005; Andersen, Nielsen & Lund, 2006; Kringelum et al., 2012; Sun et al., 2009*). The dominant-characteristic-based approach is limited by the number of features and the linear relationships between them.

The graph-based method is yet another critical method, although only two from the same study were found during the literature review. *Zhao et al. (2012)* developed a subgraph that could represent the planar nature of the epitope. Although the model is designed to identify a single epitope, it can also detect multiples. *Zhao et al. (2014)* used features extracted from both antigens and the Ag–Ab interaction, which is expressed by a coupling graph and later transformed into a general graph.

The learning-based approach utilizes machine-learning to work with a large number of features. It also uses nonlinear relationships between features to optimize model performance. *Rubinstein, Mayrose & Pupko (2009)* used two Naïve Bayesian classifiers to develop structure-based and sequence-based approaches. SEPPA 2.0 combines amino acid index (AAindex) characteristics in the SEPPA algorithm in the calculation of cluster coefficients (*Qi et al., 2014; Kawashima et al., 2008*). Aaindex in SEPPA 2.0 is consolidated via Artificial Neural Networks (ANN). However, SEPPA 3.0 adds the glycosylation triangles and glycosylation-related AAindex to SEPPA 2.0 (*Zhou et al., 2019*). Glycosylation-related AAindex is consolidated to SEPPA 3.0 via ANN. Several researchers utilized the advantages of random forest (*Dalkas & Rومان, 2017; Jespersen et al., 2017; Ren et al., 2014; Zhang et al., 2011*). The main challenge in developing a conformational B-cell epitope prediction

model is the class imbalance problem. This is a condition where the sample of the target or epitope class is less than that of the nontarget or the non-epitope classes.

Several methods have been proposed to handle the class imbalance problem. However, studies that focus on handling this issue in epitope prediction models are still limited. [Ren et al. \(2014\)](#) and [Zhang et al. \(2011\)](#) used simple random undersampling to handle the class imbalance problem. [Dalkas & Rooman \(2017\)](#) used the Support Vector Machine (SVM) Synthetic Minority Over-sampling Technique (SMOTE) method, which is a variant of SMOTE. Another common approach used is weighted SVM, which is included in the cost-sensitive algorithm level category ([Ren et al., 2015](#)). Additionally, [Zhang et al. \(2014\)](#) used a cost-sensitive ensemble approach and proved that the method was superior to Easy Ensemble, Balance Cascade and SMOTEBoost ([Liu, Wu & Zhou, 2009](#); [Chawla et al., 2003](#)).

Currently, several studies focus on class imbalance using various approaches that are mainly divided into four, including data and algorithm levels, cost-sensitive, and ensemble ([Galar et al., 2012](#)). In the data level approach, the resampling method is used to ensure a balanced distribution of data ([Gary, 2012](#)). The approaches under this category include undersampling, oversampling and a combination of both ([Drummond & Holte, 2003](#); [Estabrooks, Jo & Japcowick, 2004](#); [Chawla et al., 2002](#); [Chawla et al., 2008](#)). In the algorithm level, the minority class is specifically considered. Most algorithms are equipped with a search system to identify rare patterns ([Gary, 2012](#)). The learning process of classifiers usually ignores the minority class. Specific recognition algorithms are used to detect rare patterns, providing different misclassification weights between minority and majority classes or different weights ([Elkan, 2001](#); [Batuwita & Palade, 2012](#); [Japkowicz, Myers & Gluck, 1995](#); [Raskutti & Kowalczyk, 2003](#)). In general, adding cost to an instance is categorized as cost-sensitive in the data level ([Galar et al., 2012](#)). The approach is also applied in the ensemble method ([Blaszczynski & Stefanowski, 2014](#)). However, the determination of the weight is carried out through trial and error.

The most common ensemble methods used to handle the class imbalance problem include bagging and boosting. In bagging, a balanced class sample is generated using the bootstrapping mechanism. The sampling methods used in this case include random undersampling and oversampling, as well as SMOTE ([Blaszczynski & Stefanowski, 2014](#); [Galar et al., 2012](#)). In boosting, samples are selected iteratively and their weight calculated based on the misclassification costs. Many boosting variations have been proposed, though the most influential is the AdaBoost ([Freund & Schapire, 1996](#)).

Random oversampling and undersampling are the simplest sampling methods used in balancing data distribution. Handling class imbalance in the preprocessing data is versatile since it does not depend on the classifier used. Similarly, the random oversampling method is versatile because it does not rely on the classifier used. However, its main drawback is overfitting because new sample data are not added. The SMOTE technique avoids overfitting by interpolating adjacent members of the minority class to create new sample data ([Chawla et al., 2002](#)). Furthermore, oversampling that considers certain conditions, such as the density distribution and the position of the sample point to the majority class, improves the performance of the classifier ([He & Garcia, 2009](#); [Han et al., 2005](#)). Random

undersampling is a concern in the sense that loss of information from the dataset might occur. This is because of pruning may considerably affect and reduce its performance. To reduce the loss of information, several cluster-based methods have been used in resampling (Yen & Lee, 2009; Das, Krishnan & Cook, 2013; Sowah et al., 2016; Tsai et al., 2019).

Cluster-based undersampling can be conducted by omitting class labels (Yen & Lee, 2009; Das, Krishnan & Cook, 2013). Alternatively, it can be performed only on the negative class (Sowah et al., 2016; Lin et al., 2017; Tsai et al., 2019). Das, Krishnan & Cook (2013) discarded the negative class data that overlap the positive in a specific cluster based on the degree of overlapping. According to Yen & Lee (2009), the samples from the negative class are proportional to the ones in the positive class in a particular cluster. Also, Sowah et al. (2016) randomly selected several sample data from each cluster. In Tsai et al. (2019), the cluster members were selected using optimization algorithms. Clustering samples of the negative to positive class may lead to a suboptimal cluster number of data in the negative class (Lin et al., 2017).

In this research, the cluster-based undersampling method is combined with SMOTE to obtain a balanced dataset. The parameter r is defined to determine the proportion of the majority class data sampled and compared with the minority. A classifier model is built with the decision tree (DT) and SVM algorithms to assess the performance of the proposed method.

MATERIAL AND METHODS

Dataset

This research uses Rubinstein's dataset as training (Rubinstein, Mayrose & Pupko, 2009). The formation criteria of the training dataset are explained by Rubinstein et al. (2008). The study shows the following, (1) The Ag–Ab complex structure should contain antibodies with both heavy and light chains, (2) Contact between antigens and antibodies must occur in the complementarity-determining regions, (3) The amount of antigen residues binds to antibodies is large, and (4) The complex used cannot be similar to other complexes, as stated in the Structural Classification of Proteins criteria (Murzin et al., 1995). The training dataset consists of 76 antigen chains derived from 62 3D structure Ag–Ab complexes. The chain list is shown in Table S1. The complexes are downloaded from the Protein Data Bank (PDB) (Berman et al., 2000).

Two independent test sets are used, including Kringelum and SEPPA 3.0 (Kringelum et al., 2012; Chou et al., 2019). Kringelum's test set consists of 39 antigen chains. Data were filtered from 52 antigen chains and thirteen antigens were excluded from the list because they were used as training data with the compared method. The data released include 1AFV, 1BGX, 1RVF, 2XTJ, 3FMG, 3G6J, 3GRW, 3H42, 3MJ9, 3RHW, 3RI5, 3RIA, and 3RIF. The details of Zhang's test set are presented in Table S2A. The test set represents protein antigen in the general category. It is used to compare the CluSMOTE DT with the Discotope 1.2, Ellipro, Epitepia, EPCES, PEPITO and Discotope 2 (Andersen, Nielsen & Lund, 2006; Ponomarenko et al., 2008; Rubinstein, Mayrose & Pupko, 2009; Liang et al., 2009; Sweredoski & Baldi, 2008; Kringelum et al., 2012). The SEPPA 3.0 test set is a glycoprotein category

test set. This dataset consists of 98 antigen chains and eight were excluded because they were multiple epitopes, including 5KEM A1, 5KEM A2, 5T3X G1, 5T3X G2, 5TLJ X1, 5TLJ X2, 5TLK X1, and 5TLK X2. The test set was used to compare the CluSMOTE DT with the SEPPA 2.0 SEPPA 3.0, PEPITO, Epitopia, Discotope 2, CBTOPE and BepiPred 2.0 methods (Qi *et al.*, 2014; Ansari & Raghava, 2010; Jespersen *et al.*, 2017). The antigen list for the test set is presented in Table S2B.

Conformational B-cell epitope prediction method

Conformational epitopes are residues of exposed antigens that are spatially close, though they form separate segments when viewed from sequences (Andersen, Nielsen & Lund, 2006). To build a conformational epitope prediction model, the steps needed, as shown in Fig. 1 are include (1) preparing the dataset, (2) balancing the dataset, and (3) creating a classification model for the prediction of residual potential epitopes. The preparation step aims to build the training and testing datasets. The number of exposed residues considered as epitopes is less than the exposed residues that are not-epitopes. Balancing the dataset is meant to overcome the class imbalance found in Step 1, while the classification model categorizes residues as members of the epitope or non-epitope class.

Data preprocessing

The creation of feature vectors and epitope annotations for the training and testing data is conducted on surface residues only. Relatively accessible surface area (RSA) is used as a parameter to distinguish surface and buried residues. Different values were used as limits, including the 0.05, 0.01, 0.1, and 0.25 thresholds (Rubinstein, Mayrose & Pupko, 2009; Zhang *et al.*, 2011; Kringelum *et al.*, 2012; Ren *et al.*, 2014; Dalkas & Rومان, 2017). This variation affects the imbalance ratio between the data epitope and non-epitope classes. Although the standard burial and non-burial threshold are 0.05, the value of 0.01 is used as the limit. This is because of the larger the surface exposure threshold, the smaller the predictive performance (Basu, Bhattacharyya & Banerjee, 2011; Kringelum *et al.*, 2012). Choosing 0.01 as the limit is relevant to the finding of Zheng *et al.* (2015), where all RSA values of epitopes are positive, though slightly larger than zero.

The feature vectors used include accessible surface area (ASA), RSA, depth index (DI), protrusion index (PI), contact number (CN), HSE, quadrant sphere exposure (QSE), AAindex, B factor, and log odds ratio, as shown in Table 1.

ASA and RSA are the key features in determining if a residue is likely to bind to other molecules for accessibility reasons. Although several programs can be used to calculate ASA, the most commonly used include NACCESS and DSSP (Hubbard & Thornton, 1993; Kabsch & Sander, 1983). DSSP only calculates the total ASA per residue, while NACCESS computes the backbone, side chain, polar, and nonpolar ASA. However, NACCESS can only count one molecular structure at a time. These users need to create additional scripts to count several molecular structures at a time (Mihel *et al.*, 2008). This study used the PSAIA application was used (Mihel *et al.*, 2008). The PSAIA is not only limited to counting one molecular structure but can be used to calculate other features, including RSA, PI, and DI. No significant difference is observed between the ASA calculation results using

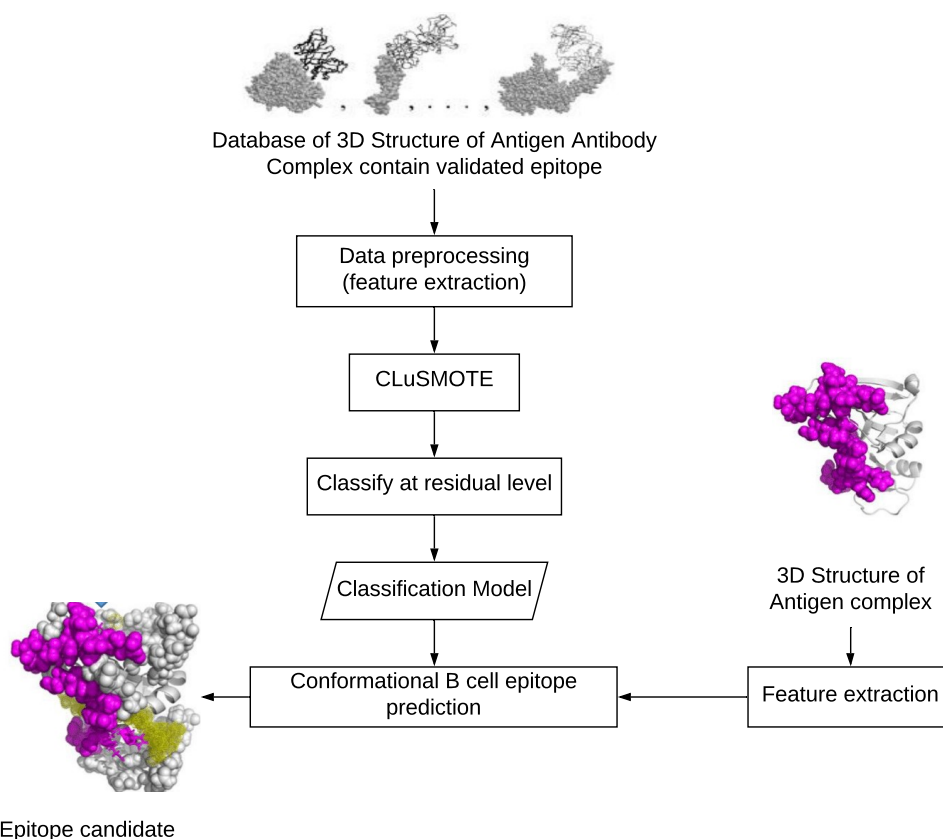


Figure 1 Development stage of conformational B-cell epitopes prediction. .

Full-size DOI: 10.7717/peerjcs.275/fig-1

Table 1 Features for antigenic determinant and the methods to compute.

Category	Feature	Data Source/Method
Structural	ASA	PSAIA <i>Mihel et al. (2008)</i>
	RSA	PSAIA <i>Mihel et al. (2008)</i>
	Protrusion Index	PSAIA <i>Mihel et al. (2008)</i>
	CN	<i>Nishikawa & Ooi (1980)</i>
	HSE	<i>Hamelryck (2005)</i>
	QSE	<i>Li et al. (2011)</i>
Physicochemical	AAIndex	<i>Kawashima et al. (2008)</i>
	B factor	<i>Ren et al. (2014)</i> and <i>Ren et al. (2015)</i>
Statistic	Log-odd ratio	<i>Andersen, Nielsen & Lund (2006)</i>

Notes.

ASA, solvent-accessible surface area; RSA, Relative solvent-accessible Surface Area; CN, Contact Number; HSE, Half-Sphere Exposure; QSE, Quadrant Sphere Exposure; AAIndex, Amino Acid Index.

NACCESS and PSAIA. The ASA attribute values used include the backbone, side chain, polar (including oxygen, nitrogen, and phosphorus), and nonpolar atoms (carbon atoms).

RSA is the result of the ASA value with the maximum figure calculated based on the GXG tripeptide theory, where G is glycine and X is the residual sought (*Lee & Richards, 1971*).

The maximum value of ASA is obtained from *Tien et al. (2013)*, which is an improvement of *Rost & Sender (1994)* and *Millerl et al. (1987)*. There was no difference in the list of datasets obtained using the three methods, as presented in Appendix 1. The RSA attribute values used include the total RSA of all atoms, backbone atoms, side-chain atoms, polar atoms (including oxygen, nitrogen, and phosphorus), and nonpolar atoms (carbon atoms).

DI: The DI of the i th atom refers to its minimum distance to the exposed atoms. The DI attribute values used include the average and standard deviation of all atoms, average of side-chain atoms, maximum, and minimum.

PI: The PI is the ratio of the space of a sphere with a radius of 10 Å centered on C α divided by the area occupied by the heavy atoms constituting proteins (*Pintar, Carugo & Pongor, 2002*). In this study, PI was calculated using the PSAIA software (*Mihel et al., 2008*). The PI attribute values used include the average and standard deviation of all atoms, average of side-chain atoms, maximum, and minimum.

CN, HSE, QSE: The CN is the total number of C α adjacent to the residue measured under the microsphere environment. It is limited by a ball with radius r centered on C α (*Nishikawa & Ooi, 1980*). In HSE, the number of C α was distributed in two areas, including the upper hemisphere and the lower hemisphere balls (*Hamelryck, 2005*). In QSE, the number of C α was specifically distributed in eight regions in the microsphere environment (*Li et al., 2011*).

AAindex: The AAindex consists of 544 indices representing the amino acid physicochemical and biochemical properties (*Kawashima et al., 2008*). The AAindex value of each residue was extracted from component I of the HDRATJCI constituents in the aaindex1.txt file. The detail of component I of aaindex file is attached as a [Table S3](#).

B factor: The B factor indicates the flexibility of an atom or a residue. An exposed residue has a larger B factor than a latent residue. The B factor for each atom is derived from the PDB data. The attribute values used include the B factor of C α and the average of all atoms or residues (*Ren et al., 2014*).

Log odds ratio: This feature is extracted based on the primary protein structure and calculated based on *Andersen, Nielsen & Lund (2006)*. A sliding window of size 9 residues was run in each sequence of antigens in the dataset to form overlapping segments that can be used in the calculation of the appearance of the individual residues. Each segment was grouped as an epitope or non-epitope depending on its center. The log odds ratio was calculated at the fifth position residue based on *Nielsen, Lundegaard & Worning (2004)*. In this study, a segment would be included in the calculation in case the fifth position residue is exposed.

Epitope annotation on the antigen residue is carried out by analyzing the interaction in the PSAIA software (*Mihel et al., 2008*) using contact criterion, threshold, and Van der Waals radii. The maximum distance of 4 was derived from the chotia.radii file. The ASA change parameters include the Delta ASA, Z_Slice Size, and Probe Radius with values 1.0, 0.25, and 1.4, respectively. The interaction analyzer output is a list of all adjacent residual pairs within the allowable distance range. A procedure for selecting antigen residues that bind to antibodies is created to obtain a list of epitopes.

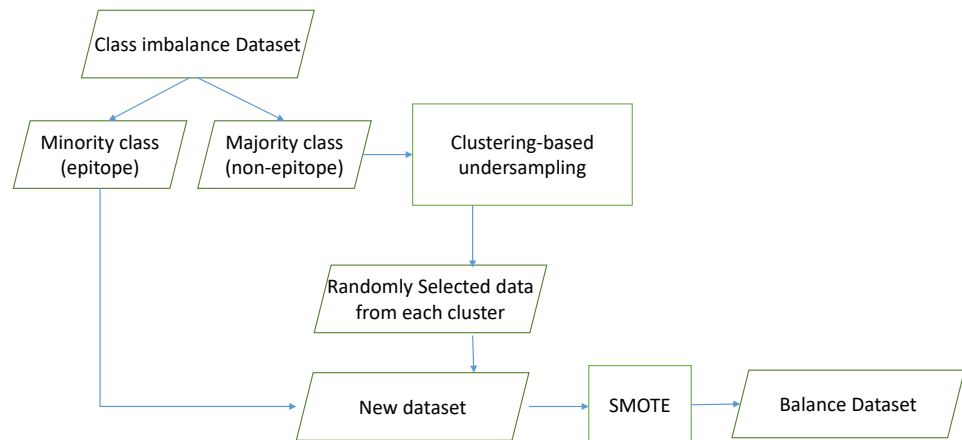


Figure 2 CluSMOTE sampling.

Full-size DOI: [10.7717/peerjcs.275/fig-2](https://doi.org/10.7717/peerjcs.275/fig-2)

Handling class imbalance with CluSMOTE

Resampling with undersampling and oversampling has advantages and disadvantages. Therefore, cluster-based sampling was conducted to minimize the loss of information caused by the pruning effect of undersampling. Oversampling with SMOTE has often proven to be reliable. Merging the two increases classifier performance. A parameter stating the degree of oversampling is used to identify the optimal combination.

This study proposed CluSMOTE, a cluster-based undersampling mechanism combined with SMOTE as shown in Fig. 2. Negative class data are clustered using the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm. This is meant to identify the optimal clusters based on stability (Campello, Moulavi & Sander, 2013). The number of clusters is less than the positive class data. This means each cluster contains several data. The simplest sampling mechanism is random selection. To select data, the cluster size and degree of oversampling should be considered.

The proposed CluSMOTE method uses the following steps,

1. Separate the positive and the negative class data.
2. Cluster the negative class (-) using the HDBScan algorithm.
3. Take a certain number of data items from each cluster. Consider the ratio of the number of clusters to the overall members of the negative class. The samples from the C_i cluster is defined in (1), according to Sowah et al. (2016). where MI is the number of minority class samples, MA is the total number of majority classes, M_{ci} is the number of C_i cluster members, and r is the negative class dataset ratio from the cluster. In case $r = 2$, the number of negative class datasets to be formed is twice the positive class datasets. The samples are taken from each cluster randomly.

$$Size_{Ci} = r \times MI \times M_{ci} / MA \quad (1)$$

4. Combine positive classes with all datasets taken in Step 3.
5. Carry out SMOTE on the results obtained in Step 4.

Program implementation was conducted in the Java programming environment with NetBeans IDE 8.2. A new class for implementing the CluSMOTE method was implemented

in the Java Language Programming, supported by the JSAT statistics library version 0.09 (Raff, 2017).

Classification algorithm

Two classification algorithms, SVM and DT, were used to evaluate the performance of CluSMOTE. Generally, SVM is a popular learning algorithm used in previous studies of conformational epitope prediction. DT is often used to handle the class imbalance problem and classified as one of the top 10 data mining algorithms (Galar et al., 2012).

This study uses the JSAT (Raff, 2017) software package, utilizing the Pegasos SVM with a mini-batch linear kernel (Shalev-shwartz, Singer & Srebro, 2007). Pegasos SVM works fast since the primal update process is carried out directly, and no support vector is stored. The default values used for the epoch, regularization, and batch size parameters include 5, $1e-4$, and 1, respectively. The decision tree is formed by nodes that are built on the principle of decision stump (Iba & Langley, 1992). Also, the study used a bottom-up pessimistic pruning with error-based pruning from the C4.5 algorithm (Quinland, 1993). The proportion of the data set used for pruning is 0.1.

Performance measurement of the conformational epitope prediction model

A dataset used for conformational epitope prediction contains the class imbalance problem. The area used is mainly under the ROC curve (AUC) as a performance parameter. In class imbalance, the AUC is a better measure than accuracy, which is biased to the majority class. Another performance parameter used is F-measure, as expressed in Eq. (2):

$$F_m = \frac{2 * PPV * SE}{PPV + SE} = 2 * TP / (2 * TP + FN + FP) \quad (2)$$

where $PPV = TP / (TP + FP)$ and SE denote sensitivity or TPR. The F-measure is not affected by imbalance conditions provided the training data used are balanced (Batuwita & Palade, 2009). Other metrics that can be used to assess performance include Gmean and adjusted Gmean (AGm). The Gmean is expressed in Eq. (3) below,

$$Gmean = \sqrt{SP * SE} \quad (3)$$

where SP denotes specificity/TPR and SE denotes sensitivity/FPR. AGm is expressed in Eq. (4):

$$AGm = \begin{cases} (Gm + SP * N) / (1 + N_n) & \text{if } SE > 0 \\ 0 & \text{if } SE = 0 \end{cases} \quad (4)$$

where Gm is Gmean, SP specificity, SE sensitivity, and N_n the proportion of negative samples in the dataset. AGm is suitable for application in case an increase in TPR is achieved with minimal reduction in TNR. Generally, this criterion is suitable for bioinformatics cases, where errors in the identification of negative classes are unexpected (Batuwita & Palade, 2009). In the case of epitope prediction, the false negative is not expected to be high. The selection of the wrong residue leads to the failure of the subsequent process.

RESULTS AND DISCUSSIONS

The complex-based leave-one-out cross-validation method is used to test the reliability of the classifier model. Each training set is built from the $n - 1$ complexes and tested with a test set from the n -th complex. Model performance was measured using seven parameters, including TPR, TNR, precision, AUC, Gmean, AGm, and F-measure.

Effect of the selection of the r value on model performance

In the original dataset, the ratio of imbalance between negative and positive classes is 10:1. To assess the effectiveness of sampling, this study utilized several r values derived using the ratio of negative to positive class data. The value $r = 1$ indicates that only the clustering and undersampling steps are applied. The value $r = 2$ indicates that the number of negative class datasets is twice the number of positive ones. The test results obtained without a balancing mechanism show the effectiveness of the proposed resampling method. The results of the assessment of the performance of the classification model expressed by the TPR, TNR, precision, AUC, Gmean, AGm, and F-measure parameters are shown in [Table 2](#).

The results of internal model validation on several variations of the r -value are also shown in [Table 2](#). Where the r -value varies from $r = 1$ to $r = 5$, both in CluSMOTE DT and CluSMOTE SVM, the TPR and the FPR value tends to decrease with the increase in the r -value. The larger the degree of oversampling, the smaller the TPR. The TNR value, as well as precision, also tends to increase with the increase in the r -value. The increase in TNR values means more negative classes are recognized. This can also be interpreted as TNR value increases means less information loss of the negative class. These two conditions indicate a trade-off between the degrees of oversampling and undersampling. Oversampling without undersampling yields TNR and precision values greater than undersampling without oversampling. Similarly, undersampling without oversampling yields TPR and FPR values greater than oversampling without undersampling. This finding indicates the undersampling mechanism is more effective in increasing positive class recognition than the oversampling, which is consistent with previous studies. Also, the resampling mechanism increases the TPR and FPR values compared to no resampling. However, the overall performance improvement indicated by the AUC, Gmean, AGm, and F-measure is not significant.

In CluSMOTE DT, AUC and Gmean have the same tendency. The best AUC and Gmean are 0.815 and 0.811 at $r = 2$ respectively. The AGm and F-measure values also have the same tendency, though the values are different. In DT, the best AGm and F-measure are obtained using the SMOTE oversampling method. In the SVM classifier, the best AGm is obtained using the SMOTE oversampling mechanism. However, the best F-measure is obtained using CluSMOTE at $r = 5$.

Previous studies on class imbalance stated that the hybrid resampling method could significantly improve performance. However, this was not the case in epitope prediction using the CluSMOTE DT method. No r value significantly influenced the overall performance improvement expressed by the AUC, Gmean, AGm, and F-measure. In case the TPR and TNR values are considered together, the selection of $r = 2$ is quite good as shown by the AUC and Gmean values. The selection of r values based on the experiment

Table 2 Performance of classification model with variations in the r -value.

No	Resampling method	r	classifier	TPR (recall)	TNR	Precision (PPV)	FPR	AUC	Gmean	Adjusted Gmean	Fmeasure
1	Cluster-based only	1	DT	0.855 ^a	0.769	0.454	0.231 ^a	0.812	0.806	0.791	0.581
2	CluSMOTE	2	DT	0.797	0.834	0.526	0.163	0.815 ^a	0.811 ^a	0.823	0.622
3	CluSMOTE	3	DT	0.764	0.862	0.558	0.138	0.813	0.807	0.833	0.634
4	CluSMOTE	4	DT	0.730	0.881	0.575	0.119	0.806	0.796	0.835	0.631
5	CluSMOTE	5	DT	0.724	0.880	0.591	0.120	0.802	0.794	0.834	0.641
6	SMOTE only	–	DT	0.644	0.939 ^a	0.732 ^a	0.061	0.791	0.771	0.848 ^a	0.675 ^a
7	No Resampling	–	DT	0.637	0.939	0.730	0.061	0.788	0.767	0.846	0.669
8	Cluster-based only	1	SVM	0.591 ^b	0.668	0.393	0.328 ^b	0.629	0.579	0.620	0.388
9	CluSMOTE	2	SVM	0.577	0.746	0.441	0.254	0.661 ^b	0.60 ^b	0.666	0.400
10	CluSMOTE	3	SVM	0.498	0.790	0.486	0.210	0.644	0.580	0.675	0.396
11	CluSMOTE	4	SVM	0.475	0.801	0.508	0.199	0.638	0.566	0.672	0.387
12	CluSMOTE	5	SVM	0.468	0.819	0.529	0.178	0.643	0.572	0.683	0.401 ^b
13	SMOTE only	–	SVM	0.384	0.881 ^b	0.606 ^b	0.119	0.632	0.532	0.688	0.368
14	No Resampling	–	SVM	0.409	0.874	0.569	0.126	0.641	0.557	0.699 ^b	0.392

Notes.

TPR, True Positive Rate; TNR, True Negative Rate; AUC, Area Under ROC Curve; Gmean, Geometric mean.

^aThe best parameter value in DT model.

^bThe best parameter value in SVM model.

shows opposing conditions between the TPR and TNR. From [Table 2](#), the best performance using AUC and Gmean is fairer compared to Agm and F-score. In the best AUC and AGm, a balanced proportion was obtained between the TPR and TNR. The best AGm and F-score resulted from the lowest TPR value. Generally, the performance models built with DT exhibit better performance than those from SVM. The performance of SVM is likely to be affected by kernel selection problems. Linear kernels are cannot separate the classes in polynomial cases. Other configurations or models may be explored for future work.

Comparison of the proposed method with previous methods

CluSMOTE DT was evaluated on an independent test set from [Kringelum et al. \(2012\)](#) by filtering the dataset from the details used in the training process of the method being compared. A total of 39 antigen data were used in the comparison, as listed in [tab4](#). The final results of the test show that CluSMOTE with $r = 2$ is superior to the other methods with an average AUC value of 0.83. The average AUC values of Discotope, Ellipro, Epitopia, EPCES, PEPITO, and Discotope 2.0 were 0.727, 0.721, 0.673, 0.697, 0.746, and 0.744, respectively.

CluSMOTE DT with $r = 2$ was evaluated on the independent test set of glycoprotein antigen by [Zhou et al. \(2019\)](#). Testing with glycoprotein antigen showed that the performance of CluSMOTE DT was similar to that of SEPPA 3.0, with the AUC values of 0.766 and 0.739, respectively. Both CluSMOTE DT and SEPPA 3.0 were superior to Epitopia, Discotope 2.0, PEPITO, CBTOPE, SEPPA 2.0, and BepiPred 2.0. The detailed performance of the eight methods compared is shown in [tab5](#). The AUC achieved by CluSMOTE DT is comparable to the one from SEPPA 3.0, showing that the proposed

method might handle epitope cases with glycoprotein well. The model developed with CluSMOTE uses the dataset presented by *Andersen, Nielsen & Lund (2006)*, which consists of 76 antigen structures. The number of complex structures used in the CluSMOTE model is less than that used in SEPPA 3.0, which consists of 767 antigen structures. The small number of antigen structures speeds up the training time for model development.

CONCLUSIONS

An epitope is a small part of the exposed antigen that creates class imbalance problems in the prediction of learning-based conformational epitopes. In this study, the CluSMOTE method was proposed to overcome the class imbalance problem in the prediction of the conformational epitope. The study shows that CluSMOTE considerably increases the TPR compared to SMOTE only. The comparison of the proposed model with state-of-the-art methods in the two datasets shows that CluSMOTE DT is comparable to or better than other methods. Its mean AUC values in Kringelum and the SEPPA 3.0 test sets are 0.83 and 0.766, respectively. This result shows that CluSMOTE DT is better than other methods in classifying the general protein antigen, though it is comparable to SEPPA 3.0 in the glycoprotein antigen.

ACKNOWLEDGEMENTS

The authors thank the Publishing and Publication Agency of Universitas Gadjah Mada for the English proof-reading of this manuscript.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by Universitas Trisakti (Doctoral scholarship). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Universitas Trisakti (Doctoral scholarship).

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Binti Solihah conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Azhari Azhari and Aina Musdholifah conceived and designed the experiments, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The data and source code are available at GitHub:

- <https://github.com/BSolihah/LIBFROMJSAT>

- <https://github.com/BSolihah/conformational-epitope-predictor>.

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj-cs.275#supplemental-information>.

REFERENCES

- Andersen PH, Nielsen M, Lund OLE. 2006.** Prediction of residues in discontinuous B-cell epitopes using protein 3D structures. *Protein Science* **15**:2558–2567 DOI [10.1110/ps.062405906.2558](https://doi.org/10.1110/ps.062405906.2558).
- Ansari HR, Raghava GPS. 2010.** Identification of conformational B-cell Epitopes in an antigen from its primary sequence. *Immunome Research* **6**(1):1–26 DOI [10.1186/1745-7580-6-6](https://doi.org/10.1186/1745-7580-6-6).
- Basu S, Bhattacharyya D, Banerjee R. 2011.** Mapping the distribution of packing topologies within protein interiors shows predominant preference for specific packing motifs. *BMC Bioinformatics* **12**(195)1–26 DOI [10.1186/1471-2105-12-195](https://doi.org/10.1186/1471-2105-12-195).
- Batuwita R, Palade V. 2009.** A new performance measure for class imbalance learning. Application to bioinformatics problems. In: *International conference on machine learning and applications. Miami Beach, Florida*. Florida: IEEE Computer Society, 545–550 DOI [10.1109/ICMLA.2009.126](https://doi.org/10.1109/ICMLA.2009.126).
- Batuwita R, Palade V. 2012.** Class imbalance learning methods for support vector. In: He H, Ma Y, eds. *Imbalanced learning: foundations, algorithms, and applications*. Hoboken: John Wiley & Sons, Inc, 83–99.
- Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. 2000.** The protein data bank. *Nucleic Acids Research* **28**(1):235–242 DOI [10.1093/nar/28.1.235](https://doi.org/10.1093/nar/28.1.235).
- Blaszczynski J, Stefanowski J. 2014.** Neighbourhood sampling in bagging for imbalanced data. *Neurocomputing* **150**:529–542 DOI [10.1016/j.neucom.2014.07.064](https://doi.org/10.1016/j.neucom.2014.07.064).
- Campello RJGB, Moulavi D, Sander J. 2013.** Density-based clustering based on hierarchical density estimates. In: Pei J, Tseng VS, Cao L, Motoda H, Xu G, eds. *Advances in knowledge discovery and data mining PAKDD Part II LNAI*. Berlin: Springer, 160–172.
- Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. 2002.** SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* **16**:321–357 DOI [10.1613/jair.953](https://doi.org/10.1613/jair.953).
- Chawla NV, Cieslak DA, Hall LO, Joshi A. 2008.** Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery* **17**(2):225–252 DOI [10.1007/s10618-008-0087-0](https://doi.org/10.1007/s10618-008-0087-0).

- Chawla NV, Lazarevic A, Hall LO, Bowyer KW. 2003.** In: *SMOTEBoost: Improving prediction of the minority class in boosting 7th european conference on principles and practice of knowledge discovery SMOTEBoost: improving prediction of the minority class in boosting*. In: *Lecture Notes in Computer Science*. DOI 10.1007/978-3-540-39804-2.
- Dalkas GA, Rooman M. 2017.** SEPIa, a knowledge-driven algorithm for predicting conformational B-cell epitopes from the amino acid sequence. *BMC Bioinformatics* 18(95):1–12 DOI 10.1186/s12859-017-1528-9.
- Das B, Krishnan NC, Cook DJ. 2013.** Handling class overlap and imbalance to detect prompt situations in smart homes. In: *IEEE 13th international conference on data mining workshops*. IEEE Computer Society. 266–273 DOI 10.1109/ICDMW.2013.18.
- Drummond C, Holte RC. 2003.** C4. 5, Class imbalance, and cost sensitivity : Why under-sampling beats over-sampling. In: *ICML workshop on learning from imbalanced data sets II*. Washington, D.C.
- Elkan C. 2001.** The foundations of cost-sensitive learning. In: *Proceedings of the seventeenth international joint conference on artificial intelligence*. 973–978.
- Estabrooks A, Jo T, Japcowick N. 2004.** A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence* 20(1):18–36 DOI 10.1111/j.0824-7935.2004.t01-1-00228.x.
- Freund Y, Schapire RE. 1996.** Experiments with a new boosting algorithm. In: *Machine learning: proceedings of the thirteenth international conference*.
- Galar M, Fern A, Barrenechea E, Bustince H. 2012.** Hybrid-based approaches. *IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews)* 42(4):463–484 DOI 10.1109/TSMCC.2011.2161285.
- Gary MW. 2012.** Foundation of imbalanced learning. In: He H, Ma Y, eds. *Imbalanced learning: foundations, algorithms, and applications*. Hoboken: John Wiley & Sons, Inc, 13–42.
- Hamelryck T. 2005.** An amino acid has two sides : a new 2D measure provides a different view of solvent exposure. 2005. *Proteins Structure, Funct Bioinforma* 59:38–48 DOI 10.1002/prot.20379.
- Han H, Wang W, Mao B. 2009.** Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang DS, Zhang XP, Huang GB, eds. *Advances in Intelligent Computing*. ICIC 2005. *Lecture notes in computer science, vol. 3644*, Berlin, Heidelberg: Springer, 878–887.
- He H, Garcia EA. 2009.** Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* 21(9):1263–1284.
- Hubbard SJ, Thornton JM. 1993.** NACCESS. Computer Program Version 2.1.1. London: Department of Biochemistry and Molecular Biology, University College London.
- Iba W, Langley P. 1992.** In: *Induction of one-level decision trees, in ML92: proceedings of the ninth international conference on machine learning*. Aberdeen, Scotland: 233–2401–3 1992, San Francisco, CA: Morgan Kaufmann, 1992(July).

- Japkowicz N, Myers C, Gluck M. 1995.** A novelty detection approach to classification. In: *The fourteenth joint conference on artificial intelligence*. New York: ACM, 518–523.
- Jespersen MC, Peters B, Nielsen M, Marcatili P. 2017.** epitope prediction using conformational epitopes. *Nucleic Acids Research* **45**:24–29 DOI [10.1093/nar/gkx346](https://doi.org/10.1093/nar/gkx346).
- Kabsch W, Sander C. 1983.** Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**:2577–2637 DOI [10.1002/bip.360221211](https://doi.org/10.1002/bip.360221211).
- Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M. 2008.** AAindex: amino acid index database, progress report 2008. *Nucleic Acids Research* **36**(2007):202–205 DOI [10.1093/nar/gkm998](https://doi.org/10.1093/nar/gkm998).
- Kringelum JV, Lundegaard C, Lund O, Nielsen M. 2012.** Reliable B cell epitope predictions: impacts of method development and improved benchmarking. *PLOS Computational Biology* **8**(12):e1002829 DOI [10.1371/journal.pcbi.1002829](https://doi.org/10.1371/journal.pcbi.1002829).
- Kringelum JV, Nielsen M, Padkjaer S, Lund O. 2013.** Structural analysis of B-cell epitopes in antibody: protein complexes. *Molecular Immunology* **53**(1–2):24–34 DOI [10.1016/j.molimm.2012.06.001](https://doi.org/10.1016/j.molimm.2012.06.001).
- Kulkarni-kale U, Bhosle S, Kolaskar AS. 2005.** CEP : a conformational epitope prediction server. *Nucleic Acids Research* **33**(Web Server issue):168–171 DOI [10.1093/nar/gki460](https://doi.org/10.1093/nar/gki460).
- Lee B, Richards FM. 1971.** The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology* **55**:379–400 DOI [10.1016/0022-2836\(71\)90324-X](https://doi.org/10.1016/0022-2836(71)90324-X).
- Li P, Pok G, KSJ A, Shon HS, Ryu KH. 2011.** QSE: a new 3-D solvent exposure measure for the analysis of protein structure. *Proteomics* **11**:3793–3801 DOI [10.1002/pmic.201100189](https://doi.org/10.1002/pmic.201100189).
- Liang S, Zheng D, Zhang C, Zacharias M. 2009.** consensus scoring. *BMC Bioinformatics* **10**(302):1–10 DOI [10.1186/1471-2105-10-302](https://doi.org/10.1186/1471-2105-10-302).
- Lin W, Tsai C, Hu Y, Jhang J. 2017.** Clustering-based undersampling in class-imbalanced data. *Information Sciences* **409–410**:17–26 DOI [10.1016/j.ins.2017.05.008](https://doi.org/10.1016/j.ins.2017.05.008).
- Liu X, Wu J, Zhou Z. 2009.** Exploratory Undersampling for. *IEEE Transaction on Cybernetics* **39**(2):539–550 DOI [10.1109/TSMCB.2008.2007853](https://doi.org/10.1109/TSMCB.2008.2007853).
- Mihel J, Šiki M, Tomi S, Jeren B, Vlahovi K. 2008.** PSAIA—protein structure and interaction analyzer. *BMC Structural Biology* **11**:1–11 DOI [10.1186/1472-6807-8-21](https://doi.org/10.1186/1472-6807-8-21).
- Millerl S, Janin J, Leskv AM, Chothial C, Laboratories CI, Physicochimique DB. 1987.** Interior and surface of monomeric proteins t. *Journal of Molecular Biology* **196**:641–656 DOI [10.1016/0022-2836\(87\)90038-6](https://doi.org/10.1016/0022-2836(87)90038-6).
- Murzin AG, Brenner SE, Hubbard T, Chothia C. 1995.** SCOP : a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology* **247**:536–540.
- Nielsen M, Lundegaard C, Worning P. 2004.** Improved prediction of MHC class I and class II epitopes using a novel Gibbs sampling approach. *Bioinformatics* **20**(9):1388–1397 DOI [10.1093/bioinformatics/bth100](https://doi.org/10.1093/bioinformatics/bth100).

- Nishikawa K, Ooi T. 1980.** Prediction of the surface-interior diagram of globular proteins by an empirical method.pdf. *International Journal of Peptide and Protein Research* **16**:19–32.
- Pintar A, Carugo O, Pongor S. 2002.** CX, an algorithm that identifies protruding atoms in proteins. *Bioinformatics* **18**(7):980–984 DOI [10.1093/bioinformatics/18.7.980](https://doi.org/10.1093/bioinformatics/18.7.980).
- Ponomarenko J, Bui H, Li W, Fusseder N, Bourne PE, Sette A, Peters B. 2008.** ElliPro : a new structure-based tool for the prediction of antibody epitopes. *BMC Bioinformatics* **9**(514):1–8 DOI [10.1186/1471-2105-9-514](https://doi.org/10.1186/1471-2105-9-514).
- Qi T, Qiu T, Zhang Q, Tang K, Fan Y, Qiu J, Wu D, Zhang W, Chen Y, Gao J, Zhu R, Cao Z. 2014.** SEPPA 2.0—more refined server to predict spatial epitope considering species of immune host and subcellular localization of protein antigen. *Nucleic Acids Research* **42**(May):59–63 DOI [10.1093/nar/gku395](https://doi.org/10.1093/nar/gku395).
- Quinland JR. 1993.** *C4.5 programs for machine learning*. San Mateo: Morgan Kaufmann.
- Raff E. 2017.** JSAT: java statistical analysis tool, a library for machine learning. *Journal of Machine Learning Research* **18**:1–5.
- Raskutti B, Kowalczyk A. 2003.** Extreme Re-balancing for SVMs: a case study. In: *Workshop on learning from imbalanced datasets II*. Washington, D.C.
- Ren J, Liu Q, Ellis J, Li J. 2014.** Tertiary structure-based prediction of conformational B-cell epitopes through B factors. *Bioinformatics* **30**:264–273 DOI [10.1093/bioinformatics/btu281](https://doi.org/10.1093/bioinformatics/btu281).
- Ren J, Liu Q, Ellis J, Li J. 2015.** Positive-unlabeled learning for the prediction of conformational B-cell epitopes. *BMC Bioinformatics* **16**(Suppl 18):1–15.
- Rost B, Sender C. 1994.** Conservation and prediction of solvent accessibility in protein families. *Proteins Structure, Function Genetics* **20**(November):216–226 DOI [10.1002/prot.340200303](https://doi.org/10.1002/prot.340200303).
- Rubinstein ND, Mayrose I, Halperin D, Yekutieli D, Gershoni JM, Pupko T. 2008.** Computational characterization of B-cell epitopes. *Molecular Immunology* **45**:3477–3489 DOI [10.1016/j.molimm.2007.10.016](https://doi.org/10.1016/j.molimm.2007.10.016).
- Rubinstein ND, Mayrose I, Pupko T. 2009.** A machine-learning approach for predicting B-cell epitopes. *Molecular Immunology* **46**:840–847 DOI [10.1016/j.molimm.2008.09.009](https://doi.org/10.1016/j.molimm.2008.09.009).
- Shalev-shwartz S, Singer Y, Srebro N. 2007.** Pegasos: Primal Estimated sub-Gradient Solver for SVM. In: *International Conference on Machine Learning (ICML)*. New York. 807–814.
- Sowah RA, Agebure MA, Mills GA, Koumadi KM, Fiawoo SY. 2016.** New cluster undersampling technique for class imbalance learning. *International Journal of Machine Learning and Computing* **6**(3):205–214 DOI [10.18178/ijmlc.2016.6.3.599](https://doi.org/10.18178/ijmlc.2016.6.3.599).
- Sun J, Wu D, Xu T, Wang X, Xu X, Tao L, Li YX, Cao ZW. 2009.** SEPPA: a computational server for spatial epitope prediction of protein antigens. *Nucleic Acids Research* **37**:612–616 DOI [10.1093/nar/gkp417](https://doi.org/10.1093/nar/gkp417).
- Sweredoski MJ, Baldi P. 2008.** PEPITO: improved discontinuous B-cell epitope prediction using multiple distance thresholds and half sphere exposure. *Bioinformatics* **24**(12):1459–1460 DOI [10.1093/bioinformatics/btn199](https://doi.org/10.1093/bioinformatics/btn199).

- Tien MZ, Meyer AG, Sydykova DK, Spielman SJ, Wilke CO. 2013.** Maximum allowed solvent accessibilities of residues in proteins. *PLOS ONE* **8(11)**:e80720 DOI [10.1371/journal.pone.0080635](https://doi.org/10.1371/journal.pone.0080635).
- Tsai C, Lin W, Hu Y, Yao G. 2019.** Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences* **477**:47–54 DOI [10.1016/j.ins.2018.10.029](https://doi.org/10.1016/j.ins.2018.10.029).
- Yen S, Lee Y. 2009.** Expert systems with applications cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications* **36**:5718–5727 DOI [10.1016/j.eswa.2008.06.108](https://doi.org/10.1016/j.eswa.2008.06.108).
- Zhang J, Zhao X, Sun P, Gao B, Ma Z. 2014.** Conformational B-cell epitopes prediction from sequences using cost-sensitive ensemble classifiers and spatial clustering. *BioMed Research International* **2014**:1–12 DOI [10.1155/2014/689219](https://doi.org/10.1155/2014/689219).
- Zhang W, Xiong Y, Zhao M, Zou H, Ye X, Liu J. 2011.** Prediction of conformational B-cell epitopes from 3D structures by random forests with a distance-based feature. *BMC Bioinformatics* **12(341)**:1–10 DOI [10.1186/1471-2105-12-341](https://doi.org/10.1186/1471-2105-12-341).
- Zhao L, Hoi SCH, Li Z, Wong L, Nguyen H, Li J. 2014.** Coupling graphs, efficient algorithms and B-cell epitope prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **11(1)**:7–16 DOI [10.1109/TCBB.2013.136](https://doi.org/10.1109/TCBB.2013.136).
- Zhao L, Wong L, Lu L, Hoi SCH, Li J. 2012.** B-cell epitope prediction through a graph model. *BMC Bioinformatics* **13((Sup 17)(S20))**:1–12.
- Zheng W, Ruan J, Hu G, Wang K, Hanlon M, Gao J. 2015.** Analysis of conformational B-Cell epitopes in the antibody-antigen complex using the depth function and the convex hull. *PLOS ONE* **10(8)**:1–16 DOI [10.1371/journal.pone.0134835](https://doi.org/10.1371/journal.pone.0134835).
- Zhou C, Chen Z, Zhang L, Zhang L, Yan D, Mao T, Tang K, Qiu T, Cao Z. 2019.** SEPPA 3.0—enhanced spatial epitope prediction enabling glycoprotein antigens. *Nucleic Acids Research* **47(May)**:388–394 DOI [10.1093/nar/gkz413](https://doi.org/10.1093/nar/gkz413).