Enhancement of conformational B-cell epitope prediction using CluSMOTE

Binti Solihah [Corresp., 1, 2] , **Azhari Azhari** [1] , **Aina Musdholifah** [1]

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

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

Corresponding Author: Binti Solihah
Email address: binti.solihah@mail.ugm.ac.id

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 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, precisely 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.

1 # Enhancement of Conformational B-cell Epitope
2 # Prediction using CluSMOTE

3 Binti Solihah[1,2], Azhari Azhari[1], Aina Musdholifah[1]

5 [1]Department of Computer Science and Electronics, Faculty of Mathematics and Natural
6 Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia
7 [2]Department of Informatics Engineering, Universitas Trisakti, Jakarta Barat, Indonesia

9 Corresponding Author:
10 Binti Solihah[1]
11 Jl Kyai Tapa No 1, Grogol, Jakarta Barat, 11440, Indonesia
12 Email address: binti.solihah@mail.ugm.ac.id/binti@trisakti.ac.id

Abstract

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

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

34 **Result.** The experimental result shows that CluSMOTE Decision Tree outperformed the
35 Support Vector Machine in terms of AUC and Gmean as performance measurements. The
36 mean AUC of CluSMOTE Decision Tree in the Kringelum and the SEPPA 3 test sets are 0.83
37 and 0.766, respectively. This shows that CluSMOTE Decision Tree is better than other

38 methods in the general protein antigen, though comparable with SEPPA 3 in the glycoprotein
39 antigen.
40

Introduction

42 A B cell epitope is among the main components of peptide-based vaccines (Andersen et al.,
43 2006; Zhang et al., 2011; Ren et al., 2015). It can be utilized in immunodetection or
44 immunotherapy to induce an immune response (Rubinstein et al., 2008). Many B cell
45 epitopes are conformational and originate from separate segments of an antigen sequence,
46 forming a spatial neighborhood in the antigen-antibody (Ag–Ab) complex. Identifying
47 epitopes through experiments is tedious and expensive work, and therefore, there is a high
48 risk of failure. Current progress in bioinformatics makes it possible to create vaccine designs
49 through 3D visualization of protein antigen. Many characteristics, including composition,
50 cooperativeness, hydrophobicity, and secondary structure, are considered in identifying
51 potential substances for an epitope (Kringelum et al., 2013). Since no dominant
52 characteristic helps experts to easily distinguish epitopes from other parts of the antigen, the
53 risk of failure is quite high.

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

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

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

75   The learning-based approach utilizes machine-learning to work with a large number of
76   features. It also uses nonlinear relationships between features to optimize model
77   performance. Rubinstein et al. (2009) used two Naïve Bayesian classifiers to develop
78   structure-based and sequence-based approaches. SEPPA 2.0 combines amino acid index
79   (AAindex) characteristics in the SEPPA algorithm in the calculation of cluster coefficients (Qi
80   et al., 2014; Kawashima et al., 2007). Aaindex in SEPPA 2.0 is consolidated via Artificial
81   Neural Networks (ANN). However, SEPPA 3.0 adds the glycosylation triangles and
82   glycosylation-related AAindex to SEPPA 2.0 (Chou et al., 2019). Glycosylation-related
83   AAindex is consolidated to SEPPA 3.0 via ANN. Several researchers utilized the advantages
84   of random forest (Dalkas and Rooman, 2017; Jespersen et al., 2017; Ren et al., 2014; Zhang
85   et al., 2011). The main challenge in developing a conformational B cell epitope prediction
86   model is the class imbalance problem. This is a condition where the sample of the target or
87   epitope class is less than that of the nontarget or the non-epitope classes.

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

97   Currently, several studies focus on Class imbalance using various approaches that are mainly
98   divided into four, including data and algorithm levels, cost-sensitive, and ensemble (Galar et
99   al., 2012). In the data level approach, the resampling method is used to ensure a balanced
100   distribution of data (Gary, 2013). The approaches under this category include
101   undersampling, oversampling and a combination of both (Estrabrooks et al., 2004; Chawla
102   et al., 2002; Chawla et al., 2008). In the algorithm level, the minority class is specifically
103   considered. Most algorithms are equipped with a search system to identify rare patterns
104   (Gary, 2013). The learning process of classifiers usually ignores the minority class. Specific
105   recognition algorithms are used to detect rare patterns, providing different misclassification
106   weights between minority and majority classes or different weights (Elkan, 2001; Batuwita
107   and Palade, 2012; Japcowicz et al., 1995; Raskutti and Kowalczyk, 2003). In general, adding
108   cost to an instance is categorized as cost-sensitive in the data level (Galar et al., 2012). The
109   approach is also applied in the ensemble method (Blaszczynski and Stefanowski, 2014).
110   However, the determination of the weight is carried out through trial and error.

111   The most common ensemble methods used to handle the class imbalance problem include
112   bagging and boosting. In bagging, a balanced class sample is generated using the

113   bootstrapping mechanism. The sampling methods used in this case include random
114   undersampling and oversampling, as well as SMOTE (Blaszczynski and Stefanowski, 2014;
115   Galar et al., 2012). In boosting, samples are selected iteratively and their weight calculated
116   based on the misclassification costs. Many boosting variations have been proposed, though
117   the most influential is the AdaBoost (Freund and Schapire, 1996).

118   Random oversampling and undersampling are the simplest sampling methods used in
119   balancing data distribution. Handling class imbalance in the preprocessing data is versatile
120   since it does not depend on the classifier used. Similarly, the random oversampling method
121   is versatile because it does not rely on the classifier used. However, its main drawback is
122   overfitting because new sample data are not added. The SMOTE technique avoids overfitting
123   by interpolating adjacent members of the minority class to create new sample data (Chawla
124   et al., 2002). Furthermore, oversampling that considers certain conditions, such as the
125   density distribution and the position of the sample point to the majority class, improves the
126   performance of the classifier (He et al., 2008; Han et al., 2005). Random undersampling is a
127   concern in the sense that loss of information from the dataset might occur. This is because
128   of pruning may considerably affect and reduce its performance. To reduce the loss of
129   information, several cluster-based methods have been used in resampling (Yen and Lee,
130   2009; Das et al., 2013; Sowah et al., 2016; Tsai et al., 2019).

131   Cluster-based undersampling can be conducted by omitting class labels (Yen and Lee, 2009;
132   Das et al., 2013). Alternatively, it can be performed only on the negative class (Sowah et al.,
133   2016; Lin et al., 2017; Tsai et al., 2019). Das et al. (2013) discarded the negative class data
134   that overlap the positive in a specific cluster based on the degree of overlapping. According
135   to Yen and Lee (2009), the samples from the negative class are proportional to the ones in
136   the positive class in a particular cluster. Also, Sowah et al. (2016) randomly selected several
137   sample data from each cluster. In Tsai et al. (2019), the cluster members were selected using
138   optimization algorithms. Clustering samples of the negative to positive class may lead to a
139   suboptimal cluster number of data in the negative class (Lin et al. 2017).

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

144   ## Materials and Methods

145   ### Dataset
146   This research uses Rubinstein's dataset as training (Rubinstein et al., 2009). The formation
147   criteria of the training dataset are explained by Rubinstein et al. (2008). The study shows the
148   following, (1) The Ag–Ab complex structure should contain antibodies with both heavy and
149   light chains, (2) Contact between antigens and antibodies must occur in the
150   complementarity-determining regions, (3) The amount of antigen residues binds to

151 antibodies is large, and (4) The complex used cannot be similar to other complexes, as stated
152 in the Structural Classification of Proteins criteria (Murzin et al., 1995). The training dataset
153 consists of 76 antigen chains derived from 62 3D structure Ag–Ab complexes. The chain list
154 is shown in Supplement Table 1. The complexes are downloaded from the Protein Data Bank
155 (PDB) (Berman et al., 2000).

156 Two independent test sets are used, including Kringelum and SEPPA 3.0 (Kringelum et al.,
157 2012; Chou et al., 2019). Kringelum's test set consists of 39 antigen chains. Data were filtered
158 from 52 antigen chains and thirteen antigens were excluded from the list because they were
159 used as training data with the compared method. The data released include 1AFV, 1BGX,
160 1RVF, 2XTJ, 3FMG, 3G6J, 3GRW, 3H42, 3MJ9, 3RHW, 3RI5, 3RIA, and 3RIF. The details of
161 Zhang's test set are presented in Supplement Table 2a. The test set represents protein
162 antigen in the general category. It is used to compare the CluSMOTE DT with the Discotope
163 1.2, Ellipro, Epitopia, EPCES, PEPITO and Discotope 2 (Andersen et al., 2006; Ponomarenko
164 et al., 2009; Rubinstein et al., 2009; Liang et al., 2009; Sweredoski and Baldi, 2008; Kringelum
165 et al., 2012). The SEPPA 3.0 test set is a glycoprotein category test set. This dataset consists
166 of 98 antigen chains and eight were excluded because they were multiple epitopes, including
167 5KEM A1, 5KEM A2, 5T3X G1, 5T3X G2, 5TLJ X1, 5TLJ X2, 5TLK X1, and 5TLK X2. The test set
168 was used to compare the CluSMOTE DT with the SEPPA 2.0 SEPPA 3.0, PEPITO, Epitopia,
169 Discotope 2, CBTOPE and BepiPred 2.0 methods (Qi et al., 2014; Anshari and Raghava, 2010;
170 Jespersen et al., 2017). The antigen list for the test set is presented in Supplement Table 2b.

Conformational B Cell Epitope Prediction Method

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

Data Preprocessing

182 The creation of feature vectors and epitope annotations for the training and testing data is
183 conducted on surface residues only. Relatively accessible surface area (RSA) is used as a
184 parameter to distinguish surface and buried residues. Different values were used as limits,
185 including the 0.05, 0.01, 0.1, and 0.25 thresholds (Rubinstein et al., 2009, Zhang et al., 2011;
186 Kringelum et al., 2012; Ren et al., 2014; Dalkas et al., 2017). This variation affects the
187 imbalance ratio between the data epitope and non-epitope classes. Although the standard
188 burial and non-burial threshold are 0.05, the value of 0.01 is used as the limit. This is because
189 of the larger the surface exposure threshold, the smaller the predictive performance (Basu
190 et al., 2011; Kringelum et al., 2012). Choosing 0.01 as the limit is relevant to the finding of

191 Zheng et al. (2015), where all RSA values of epitopes are positive, though slightly larger than
192 zero.

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

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

208 RSA is the result of the ASA value with the maximum figure calculated based on the GXG
209 tripeptide theory, where G is glycine and X is the residual sought (Lee and Richard, 1971).
210 The maximum value of ASA is obtained from Tien et al. (2013), which is an improvement of
211 Rose and Sander (1994) and Miller et al. (1987). There was no difference in the list of
212 datasets obtained using the three methods, as presented in Appendix 1. The RSA attribute
213 values used include the total RSA of all atoms, backbone atoms, side-chain atoms, polar
214 atoms (including oxygen, nitrogen, and phosphorus), and nonpolar atoms (carbon atoms).

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

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

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

228   **AAindex:** The AAindex consists of 544 indices representing the amino acid physicochemical
229   and biochemical properties (Kawashima et al., 2008). The AAindex value of each residue was
230   extracted from component I of the HDRATJCI constituents in the aaindex1.txt file. The detail
231   of component I of aaindex file is attached as a Supplement in Table 3.

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

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

243   Epitope annotation on the antigen residue is carried out by analyzing the interaction in the
244   PSAIA software (Mihel, 2008) using contact criterion, threshold, and Van der Waals radii.
245   The maximum distance of 4 was derived from the chotia.radii file. The ASA change
246   parameters include the Delta ASA, Z_Slice Size, and Probe Radius with values 1.0, 0.25, and
247   1.4, respectively. The interaction analyzer output is a list of all adjacent residual pairs within
248   the allowable distance range. A procedure for selecting antigen residues that bind to
249   antibodies is created to obtain a list of epitopes.

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

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

263   The proposed CluSMOTE method uses the following steps,
264   1.   Separate the positive and the negative class data.

265   2. Cluster the negative class (−) using the HDBScan algorithm.

266   3. Take a certain number of data items from each cluster. Consider the ratio of the
267   number of clusters to the overall members of the negative class. The samples from
268   the Ci cluster is defined in Eq. (1), according to Sowah et al. (2016).

269         $Size\_Ci = r \times MI \times M\_ci/MA,$ *(1)*

270   where MI is the number of minority class samples, MA is the total number of majority
271   classes, M_ci is the number of Ci cluster members, and r is the negative class dataset
272   ratio from the cluster. In case r = 2, the number of negative class datasets to be formed
273   is twice the positive class datasets. The samples are taken from each cluster
274   randomly.

275   4. Combine positive classes with all datasets taken in Step 3.

276   5. Carry out SMOTE on the results obtained in Step 4.

277

278 Program implementation was conducted in the Java programming environment with
279 NetBeans IDE 8.2. A new class for implementing the CluSMOTE method was implemented in
280 the Java Language Programming, supported by the JSAT statistics library version 0.09 (Raff,
281 2017).

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

287 This study uses the JSAT (Raff, 2017) software package, utilizing the Pegasos SVM with a
288 mini-batch linear kernel (Shalev-Shwartz et al., 2000). Pegasos SVM works fast since the
289 primal update process is carried out directly, and no support vector is stored. The default
290 values used for the epoch, regularization, and batch size parameters include 5, 1e-4, and 1,
291 respectively. The decision tree is formed by nodes that are built on the principle of decision
292 stump (Iba and Langley, 1992). Also, the study used a bottom-up pessimistic pruning with
293 error-based pruning from the C4.5 algorithm (Quinlan, 1993). The proportion of the data set
294 used for pruning is 0.1.

295 **Performance Measurement of the Conformational Epitope Prediction Model**
296 A dataset used for conformational epitope prediction contains the class imbalance problem.
297 The area used is mainly under the ROC curve (AUC) as a performance parameter. In class
298 imbalance, the AUC is a better measure than accuracy, which is biased to the majority class.
299 Another performance parameter used is F-measure, as expressed in Eq. (2):

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

301

302   where PPV = TP/(TP + FP) and SE denote sensitivity or TPR. The F-measure is not affected
303   by imbalance conditions provided the training data used are balanced (Batuwita and Palade,
304   2009). Other metrics that can be used to assess performance include Gmean and adjusted
305   Gmean (AGm). The Gmean is expressed in Eq. (3) below,

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

307   where SP denotes specificity/TPR and SE denotes sensitivity/FPR. AGm is expressed in Eq.
308   (4):

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

310   where Gm is Gmean, SP specificity, SE sensitivity, and Nn the proportion of negative samples
311   in the dataset. AGm is suitable for application in case an increase in TPR is achieved with
312   minimal reduction in TNR. Generally, this criterion is suitable for bioinformatics cases,
313   where errors in the identification of negative classes are unexpected (Batuwita and Palade,
314   2009). In the case of epitope prediction, the false negative is not expected to be high. The
315   selection of the wrong residue leads to the failure of the subsequent process.

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

321   **Effect of the Selection of the *r* Value on Model Performance**
322   In the original dataset, the ratio of imbalance between negative and positive classes is 10:1.
323   To assess the effectiveness of sampling, this study utilized several *r* values derived using the
324   ratio of negative to positive class data. The value r = 1 indicates that only the clustering and
325   undersampling steps are applied. The value r = 2 indicates that the number of negative class
326   datasets is twice the number of positive ones. The test results obtained without a balancing
327   mechanism show the effectiveness of the proposed resampling method. The results of the
328   assessment of the performance of the classification model expressed by the TPR, TNR,
329   precision, AUC, Gmean, AGm, and F-measure parameters are shown in Table 2.

330   The results of internal model validation on several variations of the *r*-value are also shown
331   in Table 2. Where the *r*-value varies from r = 1 to r = 5, both in CluSMOTE DT and CluSMOTE
332   SVM, the TPR and the FPR value tends to decrease with the increase in the *r*-value. The larger
333   the degree of oversampling, the smaller the TPR. The TNR value, as well as precision, also
334   tends to increase with the increase in the *r*-value. The increase in TNR values means more

335 negative classes are recognized. This can also be interpreted as TNR value increases means
336 less information loss of the negative class. These two conditions indicate a trade-off between
337 the degrees of oversampling and undersampling. Oversampling without undersampling
338 yields TNR and precision values greater than undersampling without oversampling.
339 Similarly, undersampling without oversampling yields TPR and FPR values greater than
340 oversampling without undersampling. This finding indicates the undersampling mechanism
341 is more effective in increasing positive class recognition than the oversampling, which is
342 consistent with previous studies. Also, the resampling mechanism increases the TPR and FPR
343 values compared to no resampling. However, the overall performance improvement
344 indicated by the AUC, Gmean, AGm, and F-measure is not significant.

345 In CluSMOTE DT, AUC and Gmean have the same tendency. The best AUC and Gmean are
346 0.815 and 0.811 at r = 2 respectively. The AGm and F-measure values also have the same
347 tendency, though the values are different. In DT, the best AGm and F-measure are obtained
348 using the SMOTE oversampling method. In the SVM classifier, the best AGm is obtained using
349 the SMOTE oversampling mechanism. However, the best F-measure is obtained using
350 CluSMOTE at r = 5.

351 Previous studies on class imbalance stated that the hybrid resampling method could
352 significantly improve performance. However, this was not the case in epitope prediction
353 using the CluSMOTE DT method. No *r* value significantly influenced the overall performance
354 improvement expressed by the AUC, Gmean, AGm, and F-measure. In case the TPR and TNR
355 values are considered together, the selection of r = 2 is quite good as shown by the AUC and
356 Gmean values. The selection of r values based on the experiment shows opposing conditions
357 between the TPR and TNR. From Table 2, the best performance using AUC and Gmean is
358 fairer compared to Agm and F-score. In the best AUC and AGm, a balanced proportion was
359 obtained between the TPR and TNR. The best AGm and F-score resulted from the lowest TPR
360 value. Generally, the performance models built with DT exhibit better performance than
361 those from SVM. The performance of SVM is likely to be affected by kernel selection
362 problems. Linear kernels are cannot separate the classes in polynomial cases. Other
363 configurations or models may be explored for future work.

364 **Comparison of the Proposed Method with Previous Methods**
365 CluSMOTE DT was evaluated on an independent test set from Kringelum et al. (2012) by
366 filtering the dataset from the details used in the training process of the method being
367 compared. A total of 39 antigen data were used in the comparison, as listed in Table 4. The
368 final results of the test show that CluSMOTE with r=2 is superior to the other methods with
369 an average AUC value of 0.83. The average AUC values of Discotope, Ellipro, Epitopia, EPCES,
370 PEPITO, and Discotope 2.0 were 0.727, 0.721, 0.673, 0.697, 0.746, and 0.744, respectively.

371 CluSMOTE DT with r=2 was evaluated on the independent test set of glycoprotein antigen by
372 Zhou et al. (2019). Testing with glycoprotein antigen showed that the performance of

373   CluSMOTE DT was similar to that of SEPPA 3.0, with the AUC values of 0.766 and 0.739,
374   respectively. Both CluSMOTE DT and SEPPA 3.0 were superior to Epitopia, Discotope 2.0,
375   PEPITO, CBTOPE, SEPPA 2.0, and BepiPred 2.0. The detailed performance of the eight
376   methods compared is shown in Table 5. The AUC achieved by CluSMOTE DT is comparable
377   to the one from SEPPA 3.0, showing that the proposed method might handle epitope cases
378   with glycoprotein well. The model developed with CluSMOTE uses the dataset presented by
379   Andersen et al. (2006), which consists of 76 antigen structures. The number of complex
380   structures used in the CluSMOTE model is less than that used in SEPPA 3.0, which consists
381   of 767 antigen structures. The small number of antigen structures speeds up the training
382   time for model development.

Conclusions

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

393

Acknowledgments

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

References

398   **Andersen PH, Nielsen M, Lund OLE**. Prediction of residues in discontinuous B-cell epitopes using
399   protein 3D structures. *Protein Sci*. 2006;15:2558-2567. doi:10.1110/ps.062405906.2558
400   **Ansari HR, Raghava GPS**. Identification of conformational B-cell Epitopes in an antigen from its
401   primary sequence. *Immunome Res*. 2010;6(1):6. doi:10.1186/1745-7580-6-6
402   **Basu S, Bhattacharyya D, Banerjee R**. Mapping the distribution of packing topologies within
403   protein interiors shows predominant preference for specific packing motifs. BMC Bioinformatics.
404   2011;12(195).
405   **Batuwita R, Palade V**. Class Imbalance Learning Methods For Support Vector. In: He H, Ma Y, eds.
406   *Imbalanced Learning: Foundations, Algorithms, and Applications*. John Wiley & Sons, Inc; 2012:1-20.
407   **Berman HM, Westbrook J, Feng Z,** Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The
408   Protein Data Bank. *Nucleic Acids Res*. 2000;28(1):235-242.
409   **Blaszczynski J, Stefanowski J.** Neighbourhood sampling in bagging for imbalanced data.
410   *Neurocomputing*. 2014. doi:10.1016/j.neucom.2014.07.064

411   **Campello RJGB, Moulavi D, Sander J**. Density-Based Clustering Based on Hierarchical Density
412   Estimates. In: Pei J, Tseng VS, Cao L, Motoda H, Xu G, eds. *Advances in Knowledge Discovery and Data*
413   *Mining PAKDD Part II LNAI*. Springer, Berlin, Heidelberg; 2013:160-172.
414   **Chawla N V, Cieslak DA, Hall LO, Joshi A**. Automatically countering imbalance and its empirical
415   relationship to cost. *Data Min Knowl Discov*. 2008;17(2):225-252. doi:10.1007/s10618-008-0087-0
416   **Chawla N V, Bowyer KW, Hall LO, Kegelmeyer WP**. SMOTE: Synthetic Minority Over-sampling
417   Technique. *J Artif Intell Res*. 2002;16:321-357.
418   **Chawla N V, Lazarevic A, Hall LO, Bowyer KW.** SMOTEBoost: Improving Prediction of the Minority
419   Class in Boosting 7th European Conference on Principles and Practice of Knowledge Discovery
420   SMOTEBoost : Improving Prediction of the Minority Class in Boosting. In: *Lecture Notes in Computer*
421   *Science*; 2003. doi:10.1007/978-3-540-39804-2
422   **Dalkas GA, Rooman M**. SEPIa, a knowledge-driven algorithm for predicting conformational B-cell
423   epitopes from the amino acid sequence. BMC Bioinformatics. 2017;18(95):1-12.
424   doi:10.1186/s12859-017-1528-9
425   **Das B, Krishnan NC, Cook DJ**. Handling Class Overlap and Imbalance to Detect Prompt Situations in
426   Smart Homes. In: *IEEE 13th International Conference on Data Mining Workshops*. IEEE Computer
427   Society; 2013:266-273. doi:DOI 10.1109/ICDMW.2013.18
428   **Drummond C, Holte RC.** C4. 5 , Class Imbalance, and Cost Sensitivity : Why Under-Sampling beats
429   Over-Sampling. In: *ICML Workshop on Learning from Imbalanced Data Sets II*. Washington DC; 2003.
430   **Elkan C.** The Foundations of Cost-Sensitive Learning. In: *Proceedings of the Seventeenth International*
431   *Joint Conference on Artificial Intelligence*; 2001:973-978.
432   **Estabrooks A, Jo T, Japcowick N**. A Multiple Resampling Method for Learning from Imbalanced Data
433   Sets. *Comput Intell*. 2004;20(1):18-36.
434   **Freund Y, Schapire RE**. Experiments with a New Boosting Algorithm. In: *Machine Learning:*
435   *Proceedings Of the Thirteenth International Conference*; 1996.
436   **Galar M, Fern A, Barrenechea E, Bustince H**. Hybrid-Based Approaches. *IEEE Trans Syst Cybern -*
437   *PART C Appl Rev*. 2012;42(4):463-484.
438   **Gary MW**. Foundation of Imbalanced Learning. In: He H, Ma Y, eds. *Imbalanced Learning:*
439   *Foundations, Algorithms, and Applications*. John Wiley & Sons, Inc; 2012:13-42
440   **Hamelryck T.** An Amino Acid Has Two Sides : A New 2D Measure Provides a Different View of Solvent
441   Exposure. *ProteinsStructure, Funct Bioinforma*. 2005;59(September 2004):38-48.
442   doi:10.1002/prot.20379
443   **Hubbard SJ, Thornton JM.** NACCESS. Department of Biochemistry and Molecular Biology, University
444   College London; 1993.
445   Iba, W and Langley, P. Induction of One-Level Decision Trees, in ML92: Proceedings of the Ninth
446   International Conference on Machine Learning, Aberdeen, Scotland, 1–3 July 1992, San Francisco, CA:
447   Morgan Kaufmann, 1992(July): pp. 233–240.
448   **Japkowicz N, Myers C, Gluck M.** A Novelt y Detectio n Approac h t o Classificatio. In: *The Fourteenth*
449   *Joint Conference on Artificial Intelligence*; 1989:518-523.
450   **Jespersen MC, Peters B, Nielsen M, Marcatili P**. epitope prediction using conformational epitopes.
451   *Nucleic Acids Res*. 2017;45(May):24-29. doi:10.1093/nar/gkx346
452   **Kabsch W, Sander C**. Dictionary of Protein Secondary Structure:Pattern Recognition of Hydrogen-
453   Bonded and Geometrical Features. *Biopolymers*. 1983;22:2577-2637.

454 **Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M**. AAindex :
455 amino acid index database , progress report 2008. *Nucleid Acids Res.* 2008;36(November 2007):202-
456 205. doi:10.1093/nar/gkm998

457 **Kringelum JV, Lundegaard C, Lund O, Nielsen M**. Reliable B Cell Epitope Predictions : Impacts of
458 Method Development and Improved Benchmarking. *PLOS Comput Biol.* 2012;8(12).

459 **Kulkarni-kale U, Bhosle S, Kolaskar AS**. CEP : a conformational epitope prediction server. Nucleic
460 Acids Res. 2005;33(Web Server issue):168-171. doi:10.1093/nar/gki460

461 **Lee B, Richards FM**. The Interpretation of Protein Structures : Estimation of Static Accessibility. *J*
462 *Mol Biol.* 1971;55:379-400.

463 **Li P, Pok G, Ã KSJ, Shon HS, Ryu KH**. QSE : A new 3-D solvent exposure measure for the analysis of
464 protein structure. *Proteomics.* 2011;11:3793-3801. doi:10.1002/pmic.201100189

465 **Liang S, Zheng D, Zhang C, Zacharias M**. consensus scoring. *BMC Bioinformatics.* 2009;10(302):1-
466 10. doi:10.1186/1471-2105-10-302

467 **Lin W, Tsai C, Hu Y, Jhang J.** Clustering-based undersampling in class-imbalanced data. *Inf Sci (Ny).*
468 2017;409-410:17-26. doi:10.1016/j.ins.2017.05.008

469 **Liu X, Wu J, Zhou Z**. Exploratory Undersampling for. *IEEE Trans Syst Cybern -PART B CYBERNETICS.*
470 2009;39(2):539-550.

471 **Mihel J, Šiki M, Tomi S, Jeren B, Vlahovi K**. PSAIA–Protein Structure and Interaction Analyzer. *BMC*
472 *Struct Biol.* 2008;11:1-11. doi:10.1186/1472-6807-8-21

473 **Millerl S, Janin J, Leskv AM, Chothial C**, Laboratories CI, Physicochimique DB. Interior and Surface
474 of Monomeric Proteins t. *J Mol Biol.* 1987;196:641-656.

475 **Murzin AG, Brenner SE, Hubbard T, Chothia C**. SCOP : A Structural Classification of Proteins
476 Database for the Investigation of Sequences and Structures. *J Mol Biol.* 1995;247:536-540.

477 **Nielsen M, Lundegaard C, Worning P**. Improved prediction of MHC class I and class II epitopes using
478 a   novel   Gibbs   sampling   approach.   *Bioinformatics.*   2004;20(9):1388-1397.
479 doi:10.1093/bioinformatics/bth100

480 **Nishikawa K, Ooi T**. Prediction of the surface-interior diagram of globular proteins by an empirical
481 method.pdf. *Int J Pept Protein Res.* 1980;16:19-32.

482 **Pintar A, Carugo O, Pongor S**. CX, an algorithm that identifies protruding atoms in proteins.
483 *Bioinformatics.* 2002;18(7):980-984.

484 **Ponomarenko J, Bui H, Li W, Fusseder N, Bourne PE, Sette A, Peters B.** ElliPro : a new structure-
485 based tool for the prediction of antibody epitopes. BMC Bioinformatics. 2008;9(514):1-8.
486 doi:10.1186/1471-2105-9-514

487 **Qi T, Qiu T, Zhang Q, Tang K, Fan Y, Qiu J, Wu D, Zhang W, Chen Y, Gao J, Zhu R, Cao Z**. SEPPA 2.
488 0 –– more refined server to predict spatial epitope considering species of immune host and
489 subcellular   localization   of   protein   antigen.   *Nucleid   Acids   Res.*   2014;42(May):59-63.
490 doi:10.1093/nar/gku395

491 Quinland JR, C4.5 programs for machine learning. San Mateo, CA: Morgan Kaufmann, 1993.

492 **Raff E.** JSAT : Java Statistical Analysis Tool, a Library for Machine Learning. *J Mach Learn Res.*
493 2017;18:1-5.

494 **Raskutti B, Kowalczyk A**. Extreme Re-balancing for SVMs: a case study. In: *Workshop on Learning*
495 *from Imbalanced Datasets II.* Washington DC; 2003.

496   **Ren J, Liu Q, Ellis J, Li J.** Tertiary structure-based prediction of conformational B-cell epitopes
497   through B factors. *Bioinformatics*. 2014;30:264-273. doi:10.1093/bioinformatics/btu281
498   **Ren J, Liu Q, Ellis J, Li J**. Positive-unlabeled learning for the prediction of conformational B-cell
499   epitopes. *BMC Bioinformatics*. 2015;16(Suppl 18):1-15.
500   **Rost B, Sender C**. Conservation and Prediction of Solvent Accesibility in Protein Families.
501   *ProteinsStructure, Funct Genet*. 1994;20(November):216-226. doi:10.1002/prot.340200303
502   **Rubinstein ND, Mayrose I, Halperin D, Yekutieli D, Gershoni JM, Pupko T**. Computational
503   characterization   of   B-cell   epitopes.   *Mol   Immunol*.   2008;45:3477-3489.
504   doi:10.1016/j.molimm.2007.10.016
505   **Rubinstein ND, Mayrose I, Pupko T.** A machine-learning approach for predicting B-cell epitopes.
506   *Mol Immunol*. 2009;46:840-847. doi:10.1016/j.molimm.2008.09.009
507   **Sowah RA, Agebure MA, Mills GA, Koumadi KM, Fiawoo SY**. New cluster undersampling technique
508   for   class   imbalance   learning.   Int   J   Mach   Learn   Comput.   2016;6(3):205-214.
509   doi:10.18178/ijmlc.2016.6.3.599
510   **Sun J, Wu D, Xu T, Wang X, Xu X, Tao L, Li, YX, Cao ZW.** SEPPA : a computational server for spatial
511   epitope   prediction   of   protein   antigens.   Nucleic   Acids   Res.   2009;37:612-616.
512   doi:10.1093/nar/gkp417
513   **Sweredoski MJ, Baldi P.** PEPITO : improved discontinuous B-cell epitope prediction using multiple
514   distance   thresholds   and   half sphere   exposure.   Bioinforma   Appl   Note.   2008;24(12):1459-1460.
515   doi:10.1093/bioinformatics/btn199
516   **Tien MZ, Meyer AG, Sydykova DK, Spielman SJ, Wilke CO**. Maximum allowed solvent accessibilites
517   of residues in proteins. *PLoS One*. 2013;8(11). doi:10.1371/journal.pone.0080635
518   **Tsai C, Lin W, Hu Y, Yao G**. Under-sampling class imbalanced datasets by combining clustering
519   analysis and instance selection. Inf Sci (Ny). 2019;477:47-54.
520   **Yen S, Lee Y**. Expert Systems with Applications Cluster-based under-sampling approaches for
521   imbalanced   data   distributions.   *Expert   Syst   Appl*.   2009;36:5718-5727.
522   doi:10.1016/j.eswa.2008.06.108
523   **Zhang W, Xiong Y, Zhao M, Zou H, Ye X, Liu J**. Prediction of conformational B-cell epitopes from 3D
524   structures by random forests with a distance-based feature. *BMC Bioinformatics*. 2011;12(341).
525   **Zhang J, Zhao X, Sun P, Gao B, Ma Z**. Conformational B-cell epitopes prediction from sequences using
526   cost-sensitive ensemble classifiers and spatial clustering. *Biomed Res Int*. 2014;2014.
527   **Zheng W, Ruan J, Hu G, Wang K, Hanlon M, Gao J**. Analysis of Conformational B-Cell Epitopes in the
528   Antibody-Antigen Complex Using the Depth Function and the Convex Hull. PLoS One. 2015;10(8):1-
529   16. doi:10.1371/journal.pone.0134835
530   **Zhao L, Wong L, Lu L, Hoi SCH, Li J.** B-cell epitope prediction through a graph model. BMC
531   Bioinformatics. 2012;13(Sup 17)(S20):1:12.
532   **Zhao L, Hoi SCH, Li Z, Wong L, Nguyen H, Li J**. Coupling graphs , efficient algorithms and B-cell
533   epitope prediction. IEEE/ACM Trans Comput Biol Bioinforma. 2014;11(1):7-16.
534   **Zhou C, Chen Z, Zhang L, Zhang L, Yan D, Mao T, Tang K, Qiu T,Cao Z,.** SEPPA 3 . 0 –– enhanced
535   spatial epitope prediction enabling glycoprotein antigens. *Nucleic Acids Res*. 2019;47(May):388-394.
536   doi:10.1093/nar/gkz413

Figure 1

Development stage of conformational B-cell epitopes prediction

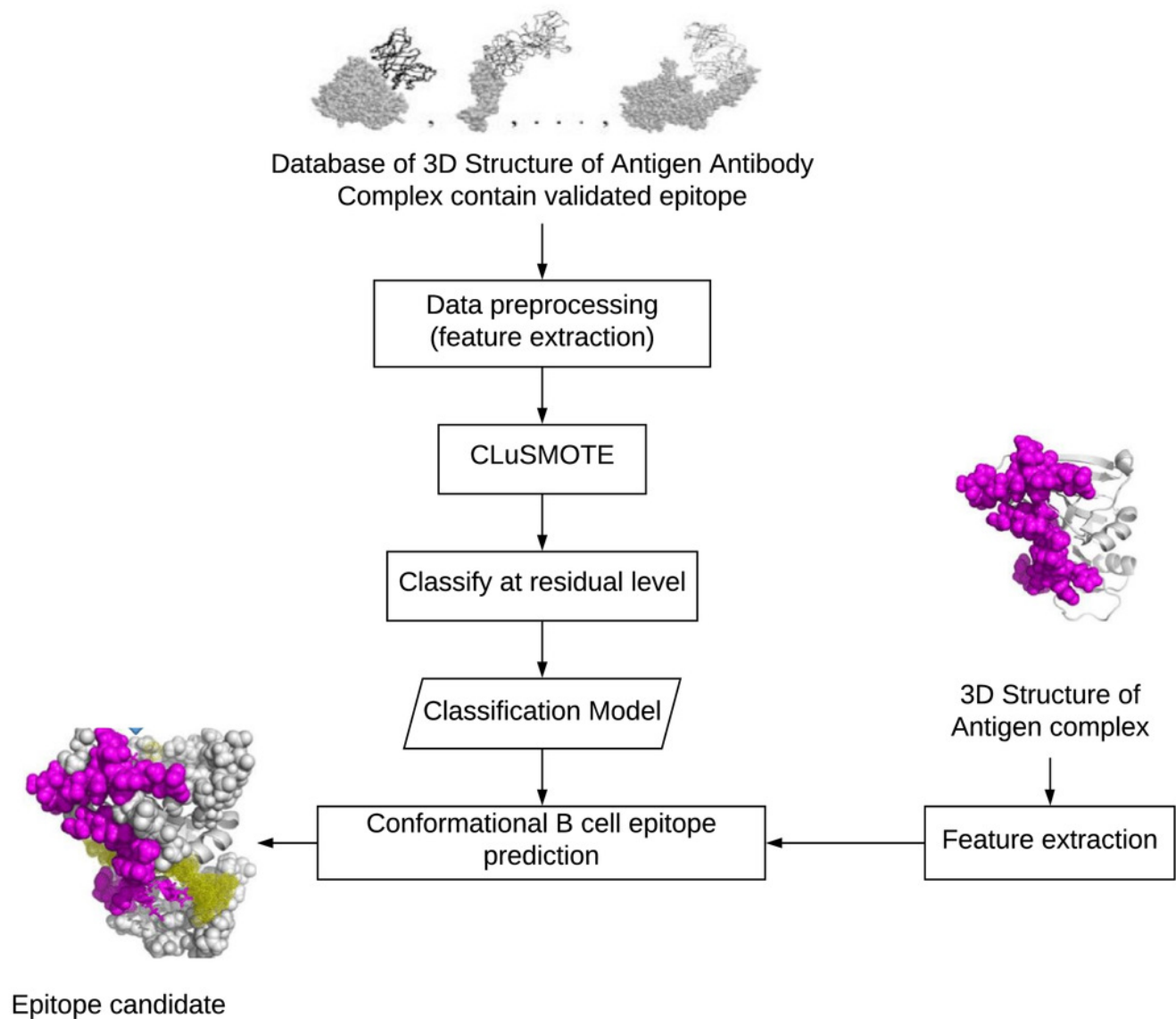
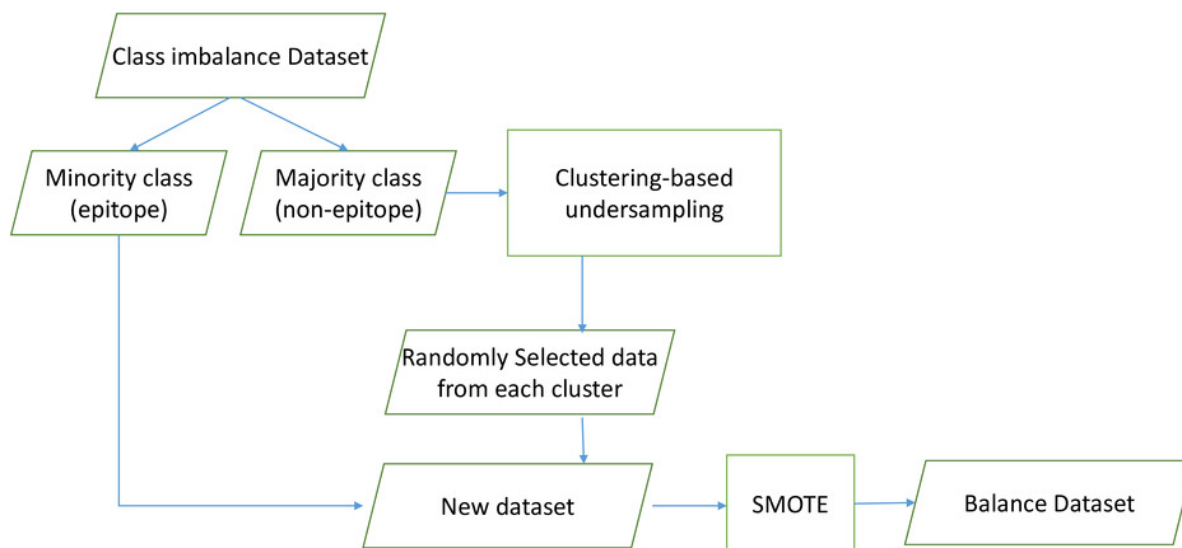



Figure 2

CluSMOTE sampling

Features for antigenic determinant and the methods to compute

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

Category	Feature	Data Source/Method
Structural	ASA	PSAIA (Mihel et al, 2008)
	RSA	PSAIA (Mihel et al, 2008)
	Protrusion Index	PSAIA (Mihel et al, 2008)
	CN	Nishikawa and Ooi (1979)
	HSE	Hamelryck, 2005
	QSE	Li et al,2011
Physicochemical	AAIndex	Kawashima, 2008
	B factor	Ren et al, 2014; Ren et al, 2015
Statistic	Log-odd ratio	Andersen et al, 2006

3   ASA: solvent-accessible surface area

4   RSA: Relative solvent-accesible Surface Area

5   CN: Contact Number

6   HSE: Half-Sphere Exposure

7   QSE: Quadrant Sphere Exposure

8   AAIndex: Amino Acid Index

Performance of classification model with variations in the r-value

1   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***	0,769	0,454	0,231*	0,812	0,806	0,791	0,581
2	CluSMOTE	2	DT	0,797	0,834	0,526	0,163	**0,815***	**0,811***	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***	**0,732***	**0,061**	0,791	0,771	**0,848***	**0,675***
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****	0,668	0,393	0,328**	0,629	0,579	0,620	0,388
9	CluSMOTE	2	SVM	0,577	0,746	0,441	0,254	**0,661****	**0,600****	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****
13	SMOTE only	-	SVM	0,384	**0,881****	**0,606****	**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****	0,392

3   TPR: True Positive Rate

4   TNR: True Negaitive Rate

5   AUC: Area Under ROC Curve

6   Gmean: Geometric mean

7   *: the best parameter value in DT model

8   **:the best parameter vaue in SVM model