Computers in Biology and Medicine 112 (2019) 103374

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

Cluster Quality based Non-Reductional (CQNR) oversampling technique and effector protein predictor based on 3D structure (EPP3D) of proteins

Rishika Sen[a], Somnath Tagore[b], Rajat K. De[a,*]

[a] Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India
[b] The Azrieli Faculty of Medicine, Bar-Ilan University, 8 Henrietta Szoldt St, Safed, Israel

ARTICLE INFO

Keywords:
Host-pathogen interactions
Oversampling
Imbalanced dataset
Convex hull
Gram-negative bacteria
Effector proteins

ABSTRACT

Background: Effector proteins of bacteria infect their hosts by specific dedicated machinery identified as secretion systems. Currently, no mechanism to identify the effector proteins based on their 3D structure has been reported in the literature. In order to identify effector proteins, extraction of features from their 3D structure is crucial. However, effector protein datasets are highly imbalanced. State-of-the-art oversampling algorithms are incapable of dealing with such datasets. They usually eliminate samples as noise. They do not ensure generation of synthetic samples strictly in the vicinity of the minority class samples. In effector protein datasets, deletion of any samples as noise would lead to loss of crucial information. Furthermore, generation of synthetic samples of the minority class in the vicinity of majority class samples would lead to an inept classifier.
Method: In this paper, we introduce an algorithm called Cluster Quality based Non-Reductional (CQNR) oversampling technique. Its novelty lies in generating new samples proportional to the distribution of samples of the minority classes, without eliminating any sample as noise. Utilizing CQNR, we develop a novel Effector Protein Predictor based on the 3D (EPP3D) structure of proteins. EPP3D is trained on a feature set, balanced by CQNR, comprising 3D structure-based features, namely, convex hull layer count, surface atom composition, radius of gyration, packing density and compactness, derived from the 3D structure of the experimentally verified effector proteins.
Result: Fscore and *Gmean* demonstrate that CQNR has outperformed some well-established oversampling methods by approximately 3–5%, with respect to classification accuracy, on five benchmark datasets and three other highly imbalanced synthetically generated datasets. Likewise, for classification of pathogenic effector proteins, a significant improvement of 7–9% in accuracy has been noticed, on the application of CQNR followed by EPP3D. Moreover, EPP3D has exhibited an improvement of 2–4% in classifying effector proteins based on their 3D structure compared to the classification of effector proteins based on their amino acid sequences. The software for CQNR and EPP3D are available at http://projectphd.droppages.com/CQNR.html.

1. Introduction

Pathogens are external agents that adversely alter cellular functions in a host. They adopt numerous ways of transporting proteins in the host via various kinds of secretion apparatus, known as secretion systems (SS) [1]. The proteins transferred by these secretion systems and any other bacterial proteins, like flagellar proteins assisting in infecting a host, are known as “effector proteins” [2]. Bacterial secretion systems are of different classes, with respect to their structures, functions, and specificity. Seven types of secretion systems (Type 1 Secretion System (T1SS) through Type 7 Secretion System (T7SS)) have been identified so far in pathogenic bacteria [1,3–6]. Bacterial pathogens use these secretion systems to manipulate the cellular activity of their hosts and ascertain a replicative niche. These secretion systems facilitate the invasion of pathogens into the hosts.

Effector proteins are translocated into host cells predominantly by

[*] Corresponding author.
E-mail addresses: rishikasen_r@isical.ac.in (R. Sen), somnath.tagore@biu.ac.il (S. Tagore), rajat@isical.ac.in (R.K. De).

https://doi.org/10.1016/j.compbiomed.2019.103374
Received 23 January 2019; Received in revised form 26 July 2019; Accepted 26 July 2019
0010-4825/

T3SS, T4SS and T6SS [7–10]. T3SS has extensively been studied [5,6]. Pathogens with T3SS effectors can infect both plants and animals [5,6,10–12]. T4SS, discovered during 1990s [13], is considered as one of the most functionally diverse bacterial secretion systems, both in terms of transported substrates and targeted recipients [14]. The working mechanism of T6SS has been discovered in 2006 [15]. Several aspects of T6SS working mechanism are still unknown. T3SS, T4SS, and T6SS associated effector proteins in many gram-negative bacteria are yet to be discovered.

Several methods have been developed to classify/predict/identify T3 and T4 effector proteins based on their amino acid sequences [16–25]. However, no 3D structural feature based classification mechanism for differentiating T3, T4, T6 effectors (pathogenic proteins), other secretion system effectors, and non-pathogenic proteins have been reported so far in the literature. Moreover, 3D structure based effector protein datasets are generally highly imbalanced.

A dataset is said to be imbalanced if the number of samples belonging to each of the classes is unequal [26]. Imbalanced datasets pose a severe problem for decision making [27]. A classifier is biased towards the class having larger number of samples, and thereby yielding unsatisfactory performance [27]. Training a classifier with an imbalanced dataset leads to overfitting [28]. Inadequate predictions using biased and overfitted classifiers have led to the notion of balancing an imbalanced dataset. A well-balanced dataset is essential for designing a reliable classification and prediction model. For a 2-class classification problem, the class having the higher cardinality is called the majority class while the other class is referred to as the minority class. Data imbalance can be tackled either by eliminating the samples from the majority class (undersampling) or increasing the number of samples in the minority class (oversampling) [26]. No matter how varied the strategies for sampling are, both the sampling techniques aim at making the cardinalities of the classes equal.

One of the simplest algorithms for oversampling is the random oversampling technique [29]. Several other algorithms have been developed over time. They include SMOTE [26], borderline-SMOTE [30], C-SMOTE [31] and Safe-Level-SMOTE [32] among others. Zhang et al. [33,34] have explored the task of balancing imbalanced image datasets for pathological brain detection. However, none of the oversampling techniques have taken any measure to regulate the generation of synthetic samples of the minority class, which may fall in the vicinity of majority samples. Moreover, some of the techniques discard samples as noise. However, discarding samples as noise may lead to loss of information embedded in the dataset.

In order to overcome the shortcomings of the aforementioned oversampling algorithms and to identify effector proteins based on their 3D structure, we develop two algorithms in this article. The article primarily has five parts. The first part describes extraction of numerous features based on 3D coordinates of each atom of the experimentally verified effector proteins. The effector dataset is imbalanced, and the state-of-the-art algorithms do not generate satisfactory prediction results. In the second part of the article, we introduce a novel oversampling algorithm, called Cluster Quality based Non-Reductional (CQNR) oversampling. CQNR has resulted in a significant improvement in performance over some existing oversampling techniques on benchmark datasets. In the third part, we develop a supervised learning based tool, called Effector Protein Predictor based on 3D structure (EPP3D) of proteins. EPP3D predicts the class of an unknown protein, after being trained with a balanced dataset obtained as the output of CQNR taking the imbalanced effector dataset as its input. EPP3D classifies unknown proteins into five classes, namely, individual classes of T3, T4 and T6 effector proteins, a composite class of T1, T2, T5, T7 effector proteins, and a class of non-effector proteins. The effectiveness of 3D structure-based classification of effector proteins has been exhibited using five classifiers individually as well as EPP3D. The performance comparison CQNR and EPP3D with respect to state-of-the-art algorithms form the fourth part of the article. Qualitative discussion regarding the comparison constitutes the fifth part of the article. The article concludes with the future scope of the current investigation.

2. Methodology

This section presents the proposed methodology of CQNR and EPP3D. The first part of this section presents collection of experimentally verified effector proteins along with the extraction of feature set from their 3D structure. A thorough discussion on the dataset created for effector classification has been provided. This is followed by a thorough explanation of CQNR, for balancing imbalanced datasets. Following CQNR, is the description of the design of EPP3D.

As stated before, we consider the following classes of effector proteins.

- Class 1 - T3 effector proteins
- Class 2 - T4 effector proteins
- Class 3 - T6 effector proteins
- Class 4 - Other (T1, T2, T7) effector proteins
- Class 5 - Non-effector proteins

The effector proteins list consists of proteins from 35 different species. The non-effector list consists of the proteins of two non-pathogenic organisms, namely, *Bacteroides vulgatus* [35] and *Listeria innocua* [36], depending on the availability of their 3D structures in the form of PDB files.

2.1. Data collection

We have accumulated experimentally verified data associated with effector proteins of T3, T4, and T6 secretion systems in 35 different species (provided in Table S1 in Supplementary Information) from different repositories/literature. These species are pathogenic to various living organisms, such as fish, amphibians, reptiles, birds, human, other animals, and various plants. We have accumulated information on 3D structures of T3, T4 and T6 effector proteins from databases, such as SecretEPDB [37], SecReT4 [38], SecReT6 [39] and Protein Data Bank [40].

Out of 1230 T3 effector proteins reported by SecretEPDB and 56 T3 effector proteins listed by Yang et al. [16], PDB structures of 36 effector proteins have been collected. Among 731 T4 effector proteins published in SecretEPDB [37] and 186 in SecReT4 database [38], PDB structures of 80 proteins have been found. Likewise, out of 107 T6 effector proteins summarized in SecReT6 database [39] and 181 in SecretEPDB, 31 PDB structures of T6 effector proteins have been obtained. Consolidating these data and removing redundancy, the summary of the ultimate list obtained has been provided in Table 1. No database containing information regarding T1, T2, T7, and non-effectors have been reported so far.

For the other groups of secreted proteins, i.e., T1, T2, T5, and T7, we have searched PDB to retrieve secreted proteins of different secretion systems.[1] The “Others” class consisted of 2 T1SS, 19 T2SS, and 3 T7SS proteins. We could not find any T5 effector protein. Due to the inadequacy of T1, T2 and T7 effectors, these effector proteins have been grouped into a single class, i.e., class 4, consisting of 24 proteins.

For the non-effectors, we have chosen two organisms, namely, *Bacteroides vulgatus* [35] and *Listeria innocua* [36], which are non-pathogenic. It may be mentioned here that there is no protein in

[1] PDB has effector protein names with “T*SS” in them, where “*” denotes the secretion system type 1, 2, 3, 4, 5, 6 or 7.

Table 1
Summary of the data comprising T1, T2, T3, T4, T6, T7 effector proteins, and non-effector proteins considered here. The column "Databases" contains the number of a particular class of effector or non-effector proteins obtained from a particular database whose reference has been given within " ()". The column "PDB" indicates the number of effector or non-effector proteins of a particular class, obtained from Protein Data Bank. The structures of proteins from the respective databases, whose 3D structures are found in PDB, have been used to create the final set of experimentally verified proteins for training the classifiers of EPP3D.

Secretion system	Number of proteins from	
	Databases	PDB Structure
T1	–	2
T2	–	19
T3	56 (Yang et al.), 1230 (SecretEPDB)	36
T4	186 (SecReT4), 731 (SecretEPDB)	80
T5	No data found	No data found
T6	107 (SecReT6), 181 (SecretEPDB)	31
T7	–	3
Non-effector	–	120

pathogenic organisms, which has been experimentally verified to be non-effectors. Here, an argument may arise that the housekeeping proteins of the same pathogenic species might have been considered as the non-effector proteins. However, in prokaryotes, genes are often found to be multi-functional [41–43]. Furthermore, a housekeeping protein of the same species may have a direct or indirect association with an effector protein [44]. Therefore, we have considered the proteins of two non-pathogenic organisms as non-effectors. We have collected 120 proteins of experimentally verified non-pathogenic organisms to constitute the non-effector set.

2.2. Features

Effector proteins T3, T4, and T6 bind with host proteins. The binding alters the working mechanism of these host proteins, which eventually disrupts the regular function of the host [45]. Thus, understanding 3D structural characteristics of pathogenic effector proteins is indeed crucial for exploring the mechanism of protein-protein interactions between host proteins and effector proteins [46]. Three-dimensional structural characteristics of pathogenic effectors (T3, T4, T6), effectors from other secretion systems (T1, T2, T7) and proteins from non-pathogenic organisms have been characterized in terms of eight features. A detailed description of these features based on 3D structures of the proteins along with their significance, is furnished below.

A protein structure can be perceived as a point cloud, where a point corresponds to an atom, a constituent element of the protein in the 3D coordinate system. The concept of a point cloud has been utilized to generate the values of eight features, namely, radius of gyration (r_g), compactness (f), convex hull layer count (h), surface atom composition (c_N - percentage composition of nitrogen atom, c_O - percentage composition of oxygen atom, c_S - percentage composition of sulfur atom, c_C - percentage composition of carbon atom) and packing density (d). These features provide essential information pertaining to the overall surface, packing pattern of atoms as well as hydrophilicity/hydrophobicity of surface atoms of a protein. These features additionally determine the binding potential of a protein molecule with other protein molecules [46]. Among them, h, f, c_N, c_O, c_S and c_C have been calculated using the notion of convex hull [47].

Radius of gyration (r_g). The radius of gyration of a protein is defined as the average distance between the center of a protein and each of its atoms [48]. The center of a protein is given by

$$(\bar{\alpha}, \bar{\beta}, \bar{\gamma}) \equiv \left(\frac{\sum_{i=1}^{n} \alpha_i}{n}, \frac{\sum_{i=1}^{n} \beta_i}{n}, \frac{\sum_{i=1}^{n} \gamma_i}{n} \right), \tag{1}$$

where (α_i, β_i, γ_i) is the coordinates of the ith atom of a protein containing n atoms. Now, the radius of gyration of a protein is defined as

$$r_g = \frac{1}{n} \sum_{i=1}^{n} [(\alpha_i - \bar{\alpha})^2 + (\beta_i - \bar{\beta})^2 + (\gamma_i - \bar{\gamma})^2]^{\frac{1}{2}} \tag{2}$$

The term r_g provides a quantitative estimate of the size of a protein. Larger the value of r_g, larger is the size of the protein. A protein molecule with a large surface area is exposed to many binding sites of another protein molecule [49]. Hence, larger the size of the protein molecule, higher is the chance of binding.

Compactness (f). Compactness of a protein has been defined as a measure of molecular surface area [50]. Mathematically, it can be represented by the ratio of its accessible surface area to the surface area of a sphere having radius r_g [50]. In order to obtain the accessible area of a protein, convex hull of the protein is determined, which is formed by using the points corresponding to the atoms of the protein.

A convex hull of a set S of points is the smallest convex polyhedron that incorporates these points. The polyhedron is such that some of the points lie on the bounding surface while the others are inside it. The convex hull of a protein has been obtained by using Delaunay triangulation [51]. Let $Conv(S)$ be the set of points on the bounding surface of the convex polyhedron. A triangulation of a finite point set $S \subset \mathbb{R}^3$ is a set $\mathscr{T}$ of triangles such that:

- $Conv(S) = \cup_j V(T_j)$ where $V(T_j)$ is the set of vertices forming j^{th} triangle T_j.
- For every distinct pair T_j, $T_{j'}$, $\in \mathscr{T}$, T_j and $T_{j'}$ have either a common vertex, a common edge or none.

Here $T = <\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3>$, such that $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ are the points forming the triangle T. The vector $\mathbf{t}_i$ is represented by the coordinates (α_i, β_i, γ_i). Thus compactness (f) of a protein is computed as

$$f = \frac{\sum_{j=1}^{p} \Delta(T_j)}{4\pi r_g^2} \tag{3}$$

where $\Delta(T_j)$ denotes the area of jth triangle T_j forming a part of the convex hull, p is the number of such triangles formed, and $\sum_{j=1}^{p} \Delta(T_j)$ denotes an estimate of the accessible surface area of the protein. The utility of this measure is to identify protein domains. Protein domain regions, compared to other regions of the protein, are more compact. The functionality of a protein depends on its domain. Due to protein folding, reduction in the surface area is linearly dependent on hydrophobicity [52]. Lower the surface area, lower is the hydrophobicity of a protein.

Convex hull layer count (h). The convex hull of a protein has been determined by using the concept of Delaunay triangulation [51] as defined above. The convex hull of a protein corresponds to its solvent accessible surface area [53]. Initially, a convex hull of all the atoms of a protein is formed. Then the points on the convex hull are eliminated. The next convex hull is obtained afresh using the remaining set of

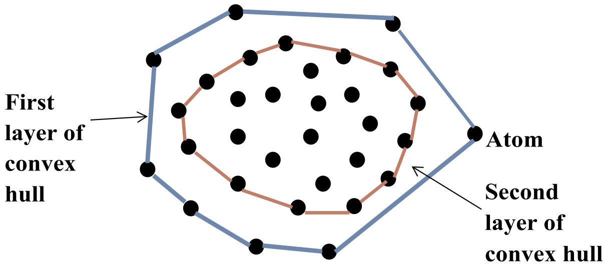


Fig. 1. Schematic diagram depicting the formation of convex hull layers. A point cloud has been taken into consideration. The atoms of a protein are denoted by these points (black) in the point cloud. The outer boundary (depicted in blue) is the first convex hull layer created with the surface atoms of an effector protein. The inner boundary (depicted in red) is the second convex hull layer created in a similar manner with the surface atoms after removing the atoms on the first convex hull layer.

atoms, and the points on the convex hull are removed. This process continues till there is no more point left [54]. If S is the set of all the points representing atoms in a protein, the next set of vertices on which convex hull will be formed, is

$$\begin{aligned} &S_1 = S - Conv(S),\\ &S_2 = S_1 - Conv(S_1),\\ &\vdots\\ &S_h = S_{h-1} - Conv(S_{h-1})\\ &S_{h+1} = \varnothing \end{aligned} \tag{4}$$

where h is the number of convex hull layers, and $Conv(S)$ is the function that returns the set of points lying on the convex hull obtained from S as shown in Fig. 1. The term h is called the convex hull layer count. As mentioned above, the convex hull layer provides information concerning the solvent accessibility analysis for proteins [47]. Convex hull layer count h provides a physical distance through which a solvent molecule would have to travel to reach the core of the protein from its surface, thus giving us an insight into how deep a protein molecule is. This feature determines the binding surface of a protein with another protein [55]. Convex hull gives a measure of the exposed surface area of a protein, hence giving an insight into the available binding area provided by a protein.

Surface atom composition. Investigation of the surface atoms of a protein presents an insight into the estimation of hydrophobic forces and their subsequent effect on protein structure [56,57]. We have extracted the percentage composition of nitrogen (c_N), carbon (c_C), oxygen (c_O) and sulfur (c_S) atoms present on the first convex hull layer of the experimentally verified effectors and non-effectors. Thus we have got four features.

Packing Density (d). It measures the packing pattern of atoms in a protein [58]. Packing Density (d) is defined as the ratio of the total volume of all the atoms to the volume of the protein, and is given by,

$$d = \frac{\sum_{i=1}^{n} v_i}{\frac{4}{3}\pi r_g^3} \tag{5}$$

where v_i is the volume of ith atom [50] and r_g is the radius of gyration of the protein. The volume of each atom has been estimated using the radius of nitrogen (N), oxygen (O), carbon (C) and sulfur (S) atoms available from the database. Packing density has a pronounced effect on the binding property of the protein [59].

The effector protein dataset that we consider in this article, comprises experimentally verified 36 T3, 80 T4, 31 T6, 24 effectors of T1, T2 and T7, and 120 non-effector proteins from two non-pathogenic bacteria - *Bacteriodes vulgatus* and *Listeria innocua*. Each of these proteins is characterized by the above eight features extracted from its 3D structure. The dataset is clearly an imbalanced one. In order to balance the same, we develop CQNR for oversampling the dataset. The working mechanism of CQNR has been furnished below.

2.3. Cluster Quality based Non-Reductional (CQNR) oversampling technique

The fundamental objectiveof algorithm CQNR is as follows: finding the best number of clusters using Davies-Bouldin index [60] for the samples of the minority class (es) to be clustered using K-means clustering algorithm, followed by generating well-spaced points within a cluster in proportion to the size of each cluster. We have applied K-means as it is widely used and produces tighter clusters than other clustering methods [61]. It has been found that the performance of Davies-Bouldin index in identifying the appropriate number of clusters is better compared to many other cluster validity indices [62]. The comparative study regarding clustering algorithms and cluster validity indices has been reported in Table S2 in Supplementary Information. A summary of the variables used in the algorithm has been given in Table 2.

Table 2
Summary of the variables used in this article.

Name	Description
h	convex hull layer count
c_N, c_C, c_O, c_S	count of nitrogen, carbon, oxygen and sulfur atoms on the surface of the molecule
r_g	radius of gyration of a protein molecule
n	number of atoms in a protein molecule
α_i, β_i, γ_i	the x,y,z coordinates of the ith atoms in a protein molecule, $1 \le i \le n$
p	number of triangles forming the convex hull
T_j	set of points forming the jth triangle of convex hull, $1 \le j \le p$
$\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{t}_3$	the three points forming T_j
$\mathscr{L}$	majority class
$\mathscr{S}$	minority class
μ	number of features in the dataset
f_e	eth feature of the dataset, $1 \le e \le \mu$
ζ_1	number of samples in the majority class
ζ_2	number of samples in the minority class
δ	difference in the sample size of the classes
m	number of clusters to be considered
C_l	lth cluster on k-mean clustering, $1 \le l \le m$
$\mathbf{x}_{lk}$	kth sample of the lth cluster of minority class, $1 \le l \le m$, $1 \le k \le \lvert C_l \rvert$
$\bar{\mathbf{x}}_l$	center of the lth cluster, $1 \le l \le m$
d	distance of $\mathbf{x}_{lk}$ with the center $\bar{\mathbf{x}}_l$ of the lth cluster, $1 \le l \le m$, $1 \le k \le \lvert C_l \rvert$
θ_l	number of synthetic samples to be generated in C_l, $1 \le l \le m$
DBI_g	Davies-Bouldin index for g clusters, $2 \le g \le 20$
$\mathbf{z}_{lq}$	a newly generated qth synthetic sample of lth cluster, $1 \le l \le m$, $1 \le q \le \theta_l$
r_l	radius the lth cluster, $1 \le l \le m$
w_1, w_2	random weight values to be generated, $0 < w_1, w_2 < 1$
$\mathbf{a}_1$, $\mathbf{a}_2$	random samples selected from a cluster of samples
$\mathscr{N}$	final balanced minority class dataset

Algorithm 1. CQNR

```
Algorithm 1 CQNR
Procedure $CQNR(\mathscr{C}_1, \mathscr{C}_2)$
Check the cardinalities of the classes $\mathscr{C}_1, \mathscr{C}_2$
Assign to $\mathscr{L}$ the class with the larger cardinality and to $\mathscr{S}$ the other class
Find $DBI_g$ for $g$ clusters obtained from $\mathscr{S}$ for $2 \leq g \leq 20$
Assign to $m$ the value of $g$ for which minimum $DBI$ score is obtained
Find the difference between the cardinalities of $\mathscr{L}$ and $\mathscr{S}$
Calculate $\theta_l$ using equation 7
$\mathscr{N} = \varnothing$
for $l = 1 : m$
  $q = 1$
  while $q \leq \theta_l$
    Select 2 random data samples $(\mathbf{a}_1, \mathbf{a}_2)$ from $l$th cluster $C_l$
    Generate a random $w_1$ in $(0, 1)$
    $w_2 = 1 - w_1$
    Generate a synthetic sample $\mathbf{z}_{lq}$ such that
    $\mathbf{z}_{lq} \leftarrow w_1\mathbf{a}_1 + w_2\mathbf{a}_2$
    Calculate $d, r_l$ using equations 10 and 12
    if $d \leq r_l$ then
      Put $\mathbf{z}_{lq}$ to $\mathscr{N}$
      $q = q + 1$
$\mathscr{N} = \mathscr{S} \cup \mathscr{N}$
return $\mathscr{N}$
```

Let us consider a two-class classification problem where the associated dataset is imbalanced. Let $\mathscr{S}$ be the minority class and $\mathscr{L}$ be the majority class. Each sample of majority or minority class is defined as $\mathbf{x} = [f_1, f_2, ..., f_\mu]^T$ where f_e is the feature value and μ is the number of features and $1 \leq e \leq \mu$. The required number of synthetic data samples to be generated is

$$\delta = \zeta_1 - \zeta_2, \tag{6}$$

where ζ_1, ζ_2 are the cardinalities of the classes $\mathscr{L}$ and $\mathscr{S}$ respectively. CQNR is applied to the minority class $\mathscr{S}$. Initially, all samples in the minority class have been taken into consideration. K-means clustering algorithm has been applied to the samples in the minority class. The number of clusters generated ranges from 2 to 20. The most suitable number (m) of clusters in the minority class is obtained by Davis-Bouldin index. CQNR takes this m value for further operations, as it has turned out to be the best with respect to Davies-Bouldin Index. Further, the cardinalities of these clusters sum up to the cardinality of the minority dataset. We now aim at generating synthetic samples in such a way that the percentage contribution of each cluster to the cardinality of the entire minority class is sustained.

Let θ_l be the number of synthetic data samples that need to be generated in l^{th} cluster C_l of the minority class. Then,

$$\theta_l = \lfloor \frac{|C_l|}{\zeta_2} \times \delta \ + 0.5 \rfloor, 1 \leq l \leq m \tag{7}$$

where

$$\delta \approx \sum_{l=1}^{m} \theta_l \tag{8}$$

To balance the dataset following the original distribution of cluster C_l, CQNR generates a new synthetic sample as a weighted sum of the randomly selected minority class samples. It might so happen that among two random samples selected from the minority class, one of the samples may have feature values close to that of samples belonging to the majority class. For such a case, the weighted sum of these two randomly chosen minority class samples subsequently may fall in the region of the majority class. In order to avoid the generation of synthetic samples in the region of majority class, the radius of a cluster has been considered. The center of the cluster is calculated to obtain the radius of the cluster. The distance of a newly generated synthetic point for a particular cluster from the cluster center is calculated and reviewed if the distance is less than the radius. If the distance is less than the radius, the synthetic sample is saved, else it is discarded.

For generation of θ_l synthetic samples, weighted sum of two randomly selected samples from C_l is considered. The cluster center $\bar{\mathbf{x}}_l$ and the radius of the cluster r_l are given by,

$$\bar{\mathbf{x}}_l = \frac{1}{|C_l|} \sum_{k=1}^{|C_l|} \mathbf{x}_{lk} \tag{9}$$

$$r_l = \frac{1}{|C_l|} \sum_{k=1}^{|C_l|} ||\mathbf{x}_{lk} - \bar{\mathbf{x}}_l|| \tag{10}$$

Each synthetic sample $\mathbf{z}_{lq}$ in C_l to be generated is given by

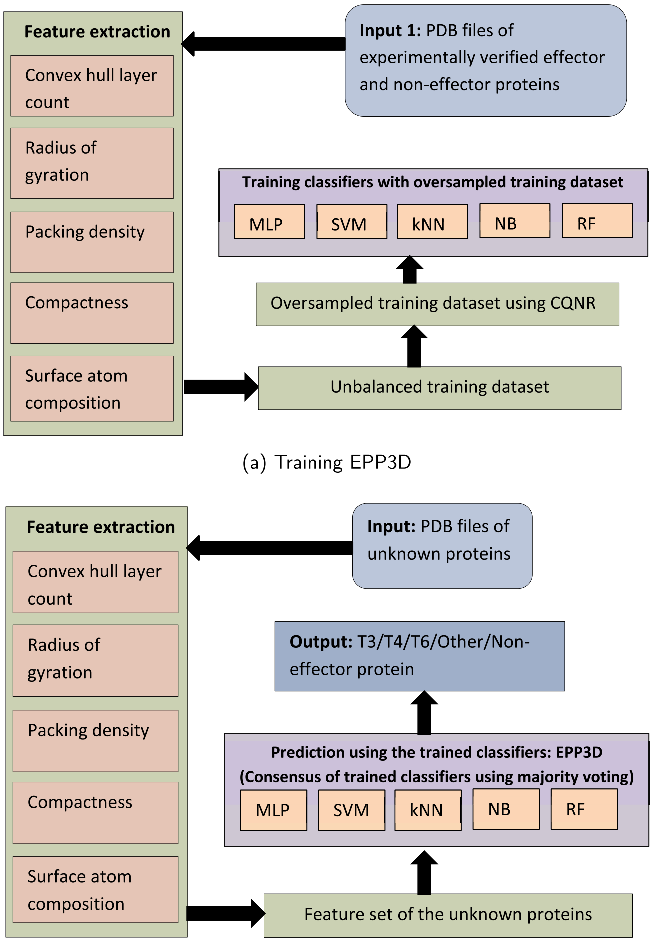
$$\mathbf{z}_{lq} = w_1\mathbf{a}_1 + w_2\mathbf{a}_2, q = 1,2, .., \theta_l; 0 < w_1, w_2 < 1, w_2 = 1 - w_1 \tag{11}$$

where $\mathbf{a}_1$, $\mathbf{a}_2$ are the two random samples selected from C_l, and w_1 is a random number generated in (0,1). The distance (d) of the newly generated sample $\mathbf{z}_{lq}$ from the center $\bar{\mathbf{x}}_l$ is calculated. If $d \leq r_l$, $\mathbf{z}_{lq}$ is selected as a synthetic sample, else it is discarded, and another $\mathbf{z}_{lq}$ is generated and checked for the criterion mentioned above. CQNR keeps generating distinct synthetic samples for which the criterion $d \leq r_l$ is satisfied. The synthetically generated samples $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_m$ corresponding to the clusters $C_1, C_2, ...C_m$ of the minority class are then merged together with the initial minority class $\mathscr{S}$ to form the final set $\mathscr{N}$. Henceforth, $\mathscr{N}$ is the new oversampled minority class, and the cardinality of $\mathscr{N}$ is approximately equal to the cardinality of the majority class.

CQNR can also be applied to imbalanced datasets consisting of samples in more than two classes. For b-class classification problem, the class with the maximum number of samples is the majority class while all the other $(b-1)$ classes form the minority classes. CQNR separately processes these $(b-1)$ minority classes for making their cardinalities approximately equal to the cardinality of the majority class. Algorithm 1 describes the working principle of CQNR. As observed, CQNR retains the original dataset and generates the minimum number of synthetic samples required for balancing majority and minority classes, consequently sustaining the distribution of the original dataset. It can handle the oversampling of disjoint clusters of data points of the minority class. It does not eliminate any data point as noise.

One may argue the biological validity of the synthetic samples. It may be noted that the discovery of effector proteins in several pathogenic species is currently being actively researched, with new effector proteins being discovered frequently. There is no guarantee that these new effector proteins will not resemble the synthetically generated samples. Also, for any sort of class imbalance, even for biological datasets, oversampling has been carried out in multiple investigations of Hu et al. [63], Santos et al. [64], Zhang et al. [65] among others. To ensure maximum resemblance to the experimentally validated data, we have generated synthetic samples in the vicinity of the experimentally verified samples without replicating the samples themselves.

Effector proteins in pathogenic bacteria are very less in number, compared to the whole protein set of the pathogen. In such a case, none of the samples can be termed as noise, since every sample of a dataset represents a unique feature. In such a small but potent dataset of effector proteins, undersampling would lead to loss of information. Keeping that in mind, we have implemented algorithm CQNR to balance the effector protein dataset created previously and utilize the balanced dataset to build the effector protein predictor EPP3D.


Feature extraction
Convex hull layer count
Radius of gyration
Packing density
Compactness
Surface atom composition
Input 1: PDB files of experimentally verified effector and non-effector proteins
Training classifiers with oversampled training dataset
MLP
SVM
kNN
NB
RF
Oversampled training dataset using CQNR
Unbalanced training dataset
Feature extraction
Convex hull layer count
Radius of gyration
Packing density
Compactness
Surface atom composition
Input: PDB files of unknown proteins
Output: T3/T4/T6/Other/Non-effector protein
Prediction using the trained classifiers: EPP3D
(Consensus of trained classifiers using majority voting)
MLP
SVM
kNN
NB
RF
Feature set of the unknown proteins

(a) Training EPP3D

(b) Prediction using EPP3D

Fig. 2. Flowcharts depicting the internal architecture of EPP3D. Figure (a) depicts the stage of training EPP3D, while Figure (b) shows the role of trained EPP3D as a predictor. The feature extraction module extracts feature set from PDB files of the experimentally verified effector/non-effector proteins in training phase (Figure (a)) along with the extraction of feature set from PDB files of unknown proteins, which needs to be classified in prediction phase (Figure (b)). In Figure (a), the feature set of the experimentally verified effector and non-effector proteins, after having been balanced by CQNR, is used to train EPP3D. In Figure (b), the trained EPP3D has been used for prediction of the class label of an unknown protein. The Output module accumulates the outcome of the five classifiers, and determines the class an unknown protein belongs to, based on majority voting.

2.4. EPP3D: Effector protein predictor based on the 3D structure of proteins

Due to the lack of any method for classification and prediction of effector proteins based on 3D structure, we have developed a tool for classification of various effector and non-effector proteins, based on their 3D structure. EPP3D is a python based tool that predicts, based on eight features, the class of an unknown protein. Due to the imbalance in the dataset, it has been oversampled by CQNR. A consensus of five classifiers based on majority voting system predicts the class of an unknown protein. It is a more suitable alternative in classification over single classifiers [66]. A consensus of classifiers average out biases, reduce variance, and are unlikely to overfit. The tool uses the training dataset set, which involves the features extracted from the PDB files of the experimentally verified effector and non-effector proteins. EPP3D extracts features from the PDB files of the unknown proteins. EPP3D applies CQNR to balance the training dataset. As an output, it predicts whether a protein belongs to class 1 (T3), class 2 (T4), class 3 (T6), class 4 (T1, T2, T7) or class 5 (non-effectors). The flow of EPP3D has been depicted in Fig. 2.

3. Results

The effectiveness of CQNR has been demonstrated on some of the profoundly referenced benchmark datasets, namely, Pima Indians Diabetes, Haberman, Spambase, Hill-Valley, and Blood Transfusion datasets as given in Table 3. These datasets have been downloaded from UCI machine learning repository [67]. Besides, we have also generated three highly imbalanced synthetic datasets. The superior performance of CQNR has been exhibited on the datasets mentioned above over some existing oversampling algorithms, namely, random oversampling, SMOTE, Borderline-SMOTE, C-SMOTE, and Safe-level-SMOTE.

We have also worked on various effector protein datasets. For this

Table 3
Summary of the imbalanced datasets (2-class) used to compare the performance of various oversampling techniques.

Dataset	Number of Features	Majority Class cardinality	Minority Class cardinality
Pima Diabetes	8	500	268
Haberman	3	225	81
Spambase	57	2788	1813
Hill-Valley	100	612	329
Blood transfusion	4	570	178
Synthetic Dataset 1	2	400	100
Synthetic Dataset 2	2	500	100
Synthetic Dataset 3	2	600	100

purpose, we have, first of all, extracted and analyzed features from various experimentally verified effector proteins. The datasets have been balanced by CQNR. Finally, the effector proteins have been classified using five popular classification algorithms.

3.1. Performance evaluation parameters

The classification performance has been depicted in terms of accuracy, sensitivity, specificity, and precision, which are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{12}$$

$$Sensitivity = \frac{TP}{TP + FN}, \tag{13}$$

$$Specificity = \frac{TN}{FP + TN}, \tag{14}$$

$$Precision = \frac{TP}{TP + FP}. \tag{15}$$

Here TP (FP) denotes the number of positive (negative) class samples identified correctly; FP (FN) stands for the number of positive (negative) class samples identified incorrectly. For a more robust performance measure for classification of imbalanced datasets, we have considered *Fscore* and *Gmean*. *Fscore* [68] combines *sensitivity* and *precision*, and is given by

$$\begin{aligned} Fscore &= \frac{2}{\frac{1}{precision} + \frac{1}{sensitivity}} \\ &= \frac{2TP}{2TP + FP + FN} \end{aligned} \tag{16}$$

Gmean[69] attempts to maximize the accuracy across the two classes with a good balance, and is defined as

$$\begin{aligned} Gmean &= \sqrt{Sensitivity \times Specificity} \\ &= \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{FP + TN}} \end{aligned} \tag{17}$$

Larger the values of *Fscore* and *Gmean*, better is the classifier.

Let M_{ij} be the number of samples being in *i*th class, having been predicted to belong to *j*th class. For a *b*-class classification problem, c, s, t_j, p_j are defined below.

- $c = \sum_{i=1}^{b} M_{ii}$ is the total number of samples correctly predicted.
- $s = \sum_{i=1}^{b} \sum_{j=1}^{b} M_{ij}$ is the total number of samples.
- $t_j = \sum_{i=1}^{b} M_{ji}$ is the total number of samples in class *j*.
- $p_j = \sum_{i=1}^{b} M_{ij}$ is the number of samples predicted to be in class *j*.

For multiclass classification, we have used accuracy, Matthew's correlation coefficient (MCC) and Cohen's kappa (κ) score. Cohen's kappa score, for multiclass classification problem, is given by Ref. [70].

$$\kappa = \frac{c \times s - \sum_{j=1}^{b} p_j t_j}{\left(s^2 - \sum_{j=1}^{b} p_j t_j\right)} \tag{18}$$

MCC is defined as [71].

$$MCC = \frac{c \times s - \sum_{j=1}^{b} p_j t_j}{\sqrt{\left(s^2 - \sum_{j=1}^{b} p_j^2\right)\left(s^2 - \sum_{j=1}^{b} t_j^2\right)}} \tag{19}$$

Cohen's kappa score is a simple and widely used metric for measuring the performance of a classifier dealing with more than two classes [72,73]. The value of κ is always less than or equal to 1, where score less than 0 indicates a random prediction. Closer the value of κ to 1, better is the prediction. MCC takes true/false positives/negatives into account, and is generally regarded as a balanced measure [71]. MCC values lie between -1 and $+1$, where the value of $+1$ represents a perfect prediction, 0 an average random prediction, and the value of -1 indicates an inverse prediction.

3.2. Application of CQNR for balancing various benchmark datasets along with comparison

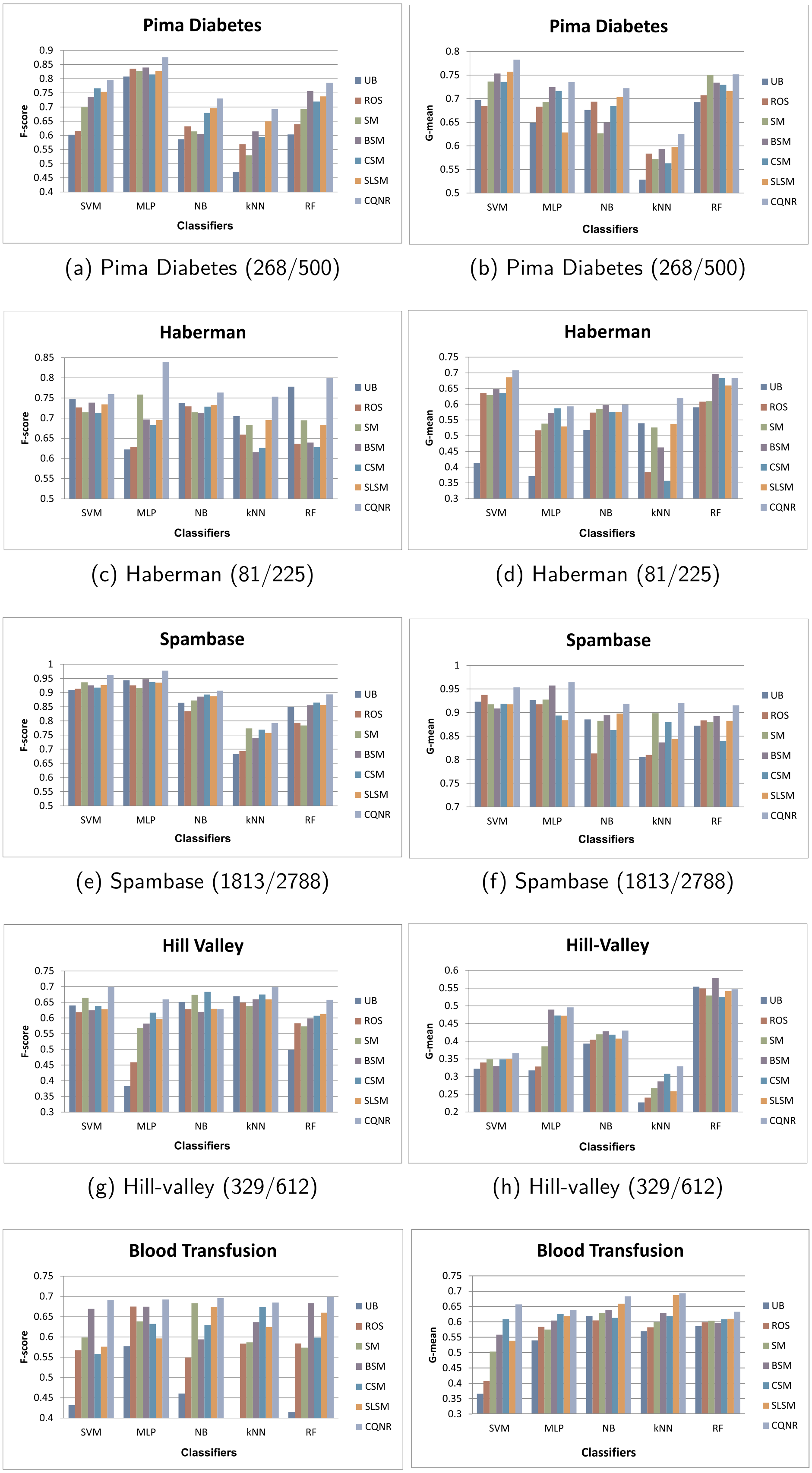
The performance of CQNR has been demonstrated on the five benchmark datasets, namely, Pima Diabetes, Haberman, Spambase, Hill-valley, and Blood Transfusion. Besides, we have designed three synthetic datasets which are highly imbalanced. To assess the performance of the oversampling methods, we have considered five classification algorithms, namely, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Naive Bayes (NB) classifier, k-Nearest Neighbor (kNN) classifier, and Random Forest (RF) classifier. For SVM, the decision function used is one-over-one ('ovo') with RBF kernel. For MLP, we have considered two hidden layers, apart from the input and output layers. For kNN, we have considered k = 3. We have first split the entire dataset into two sets - training set and test set. The training set comprises 70% of the entire dataset while remaining 30% samples form the test set. We have kept the test set (30%) aside for testing purpose. The training set (70%) has been oversampled. A 10-fold cross validation has been carried out on the oversampled training set. That is, the oversampled training set has been divided into ten non-overlapping subsets. The classifiers have then been trained using nine such subsets while the remaining subset has been used for validation. Comparative performance of CQNR with respect to some other oversampling algorithms has been depicted in terms of *Fscore* and *Gmean* (Fig. 3).

As observed from the plots in Fig. 3, CQNR has achieved the highest *Fscore* of 0.87 for Pima Diabetes dataset, 0.83 for Haberman, 0.97 for Spambase, 0.69 for Hill-Valley and 0.69 for Blood Transfusion datasets. CQNR has achieved the highest *Gmean* score of 0.78 for Pima Diabetes dataset, 0.71 for Haberman dataset, 0.96 for Spambase dataset, 0.54 for Hill-Valley dataset, 0.69 for Blood Transfusion dataset. CQNR has outperformed, in terms of *Fscore* and *Gmean*, the other oversampling algorithms for almost all the datasets using all the five classification algorithms. The performance of numerous oversampling algorithms, including that of CQNR on three synthetic datasets, has been given in Tables S3–S5 in Supplementary Information.

CQNR has led to more reliable performance for different datasets and classification techniques, with some exceptions (Fig. 3). For Haberman and Hill-valley datasets, borderline-SMOTE has shown the best performance among all the other oversampling algorithms for random forest classifier with respect to *Gmean*. For almost all the datasets, CQNR shows a stable performance in terms of *Fscore* and *Gmean*, indicating an unbiased prediction of samples. For Pima dataset, the variation of *Fscore* is low, which suggests that nearly all the oversampling algorithms have a negligible difference in performance. On the other hand, Blood transfusion dataset has shown a drastic difference for performance metric *Fscore* for various oversampling algorithms. Hill-valley dataset, in terms of *Gmean*, has shown a stable performance over all the oversampling algorithms with a negligible difference. For Haberman dataset, on the other hand, the oversampling algorithms have resulted in a drastic difference in performance.

3.3. Comparative performance of EPP3D on various effector protein datasets balanced by some existing oversampling methods including CQNR

Several classification algorithms have been used to classify the experimentally verified T3, T4, and T6 effector proteins against other effector and non-effector proteins. The dataset consists of 36 T3 effector proteins, 80 T4 effector proteins, 31 T6 effector proteins, 24 effectors of T1, T2, T5 and T7, and 120 non-effector proteins from the non-pathogenic bacteria *Bacteriodes vulgatus* and *Listeria innocua*. Thus, the dataset



(a) Pima Diabetes (268/500)

(b) Pima Diabetes (268/500)

(c) Haberman (81/225)

(d) Haberman (81/225)

(e) Spambase (1813/2788)

(f) Spambase (1813/2788)

(g) Hill-valley (329/612)

(h) Hill-valley (329/612)

(i) Blood Transfusion (178/570)

(j) Blood Transfusion (178/570)

Fig. 3. Comparison of the classification performances, in terms of *Fscore* and *Gmean*, of different classifiers on datasets oversampled by different oversampling algorithms. The numbers within brackets indicate the ratio of the cardinalities of minority class to the majority class. The abbreviations for the methods are UB - imbalanced, ROS - Random Oversampling, SM - SMOTE, BSM - borderline SMOTE, CSM - C Smote, SLSM - Safe level SMOTE, CQNR - Cluster Quality based Non-Reductional Oversampling. As observed from the figures, CQNR has performed the best over the other oversampling algorithms considered here.

is visibly imbalanced. In such a small but potent dataset of effector proteins, undersampling would lead to loss of information. Keeping this fact in mind, we have applied algorithm CQNR to balance the dataset by oversampling.

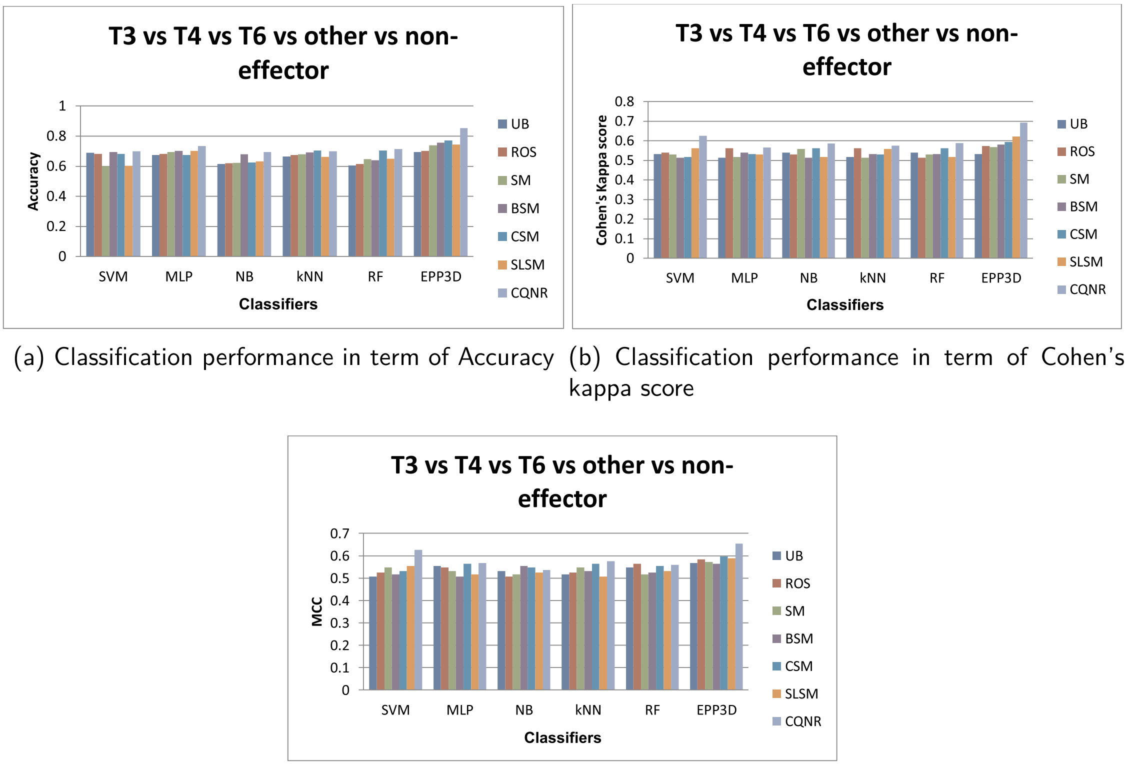
After balancing, the feature set has been normalized. We have subjected the dataset to variance threshold, which removes features with low variance. None of the features in the effector protein dataset has ultimately been removed. We have reported accuracy, MCC, and κ score of the five commonly used techniques along with EPP3D for classification of T3, T4, and T6 effector proteins. These five techniques are Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Naive Bayes (NV), K-Nearest Neighbor (KNN) and Random Forest (RF). It has been found that EPP3D has resulted in a remarkable improvement in the performance of predicting unknown proteins into different classes.

In Fig. 4, we have assessed the performance of EPP3D, in terms of accuracy, κ score and MCC, to know how diverse the 3D structural characteristics of the pathogenic effectors pertaining to different secretion systems are. Classification of effectors by EPP3D, where CQNR has balanced the effector dataset, has resulted in the best performance in terms of accuracy (85.43%), MCC (0.6536), and κ score (0.6821). κ score ranging from 0.61 to 0.80 indicates a substantially well performing predictor [74]. As observed from Fig. 4, CQNR has resulted in performances having an accuracy ranging from 69.94% for SVM to 85.43% for EPP3D, while the imbalanced dataset has produced an approximate performance ranging from 68.97% for SVM to 69.43% for EPP3D. CQNR has led to an average classification performance of 73.29%, the highest among all the other oversampling techniques. EPP3D, along with CQNR, has obtained an average accuracy of 75.18%, the highest among the performances of individual classifiers and oversampling algorithms. However, Naive-Bayes classifier has given better performance for borderline-SMOTE compared to the other oversampling algorithms. A visible improvement has been noticed in classification accuracy using balanced data compared to that using imbalanced data for most of the classifiers. The performance of EPP3D with CQNR has provided an overall better accuracy, MCC value, and κ score than that of individual classifiers.

Several subsets of proteins oversampled by CQNR have been classified using EPP3D. As observed from the tables (Tables S6–S11 in Supplementary Information), the dataset, where any one of T3, T4 and T6 has been considered as a single class versus the “Others”, shows the best classification performance with respect to accuracy (96.24%), MCC (0.8834) and κ score (0.8624). Similar values across all these three measurements indicate a stable performance of CQNR. On the other hand, the subsets of the 3-class dataset (T4, T3, and T6) combined show an unsatisfactory performance, in terms of accuracy (64.32%), MCC (0.5432) and κ score (0.5932). Such a poor performance indicates that the features of the 3-class dataset (T3, T4, and T6) have considerably low variance. CQNR, together with EPP3D, has shown better performance for majority of the effector protein datasets. A detailed tabulated representation of the performance of various oversampling algorithms has been given in Tables S6–S11 in Supplementary Information.

3.4. Comparative performance of EPP3D with existing effector protein prediction algorithms

Several methods have been developed to classify effector proteins based on their peptide sequences. These include ones using machine learning techniques [16–25]. So far, no work has been reported, which predicts the classes of effector proteins based on 3D structural characteristics of experimentally verified effectors. Absence of a 3D structure-based effector protein predictor has led to the designing of EPP3D.

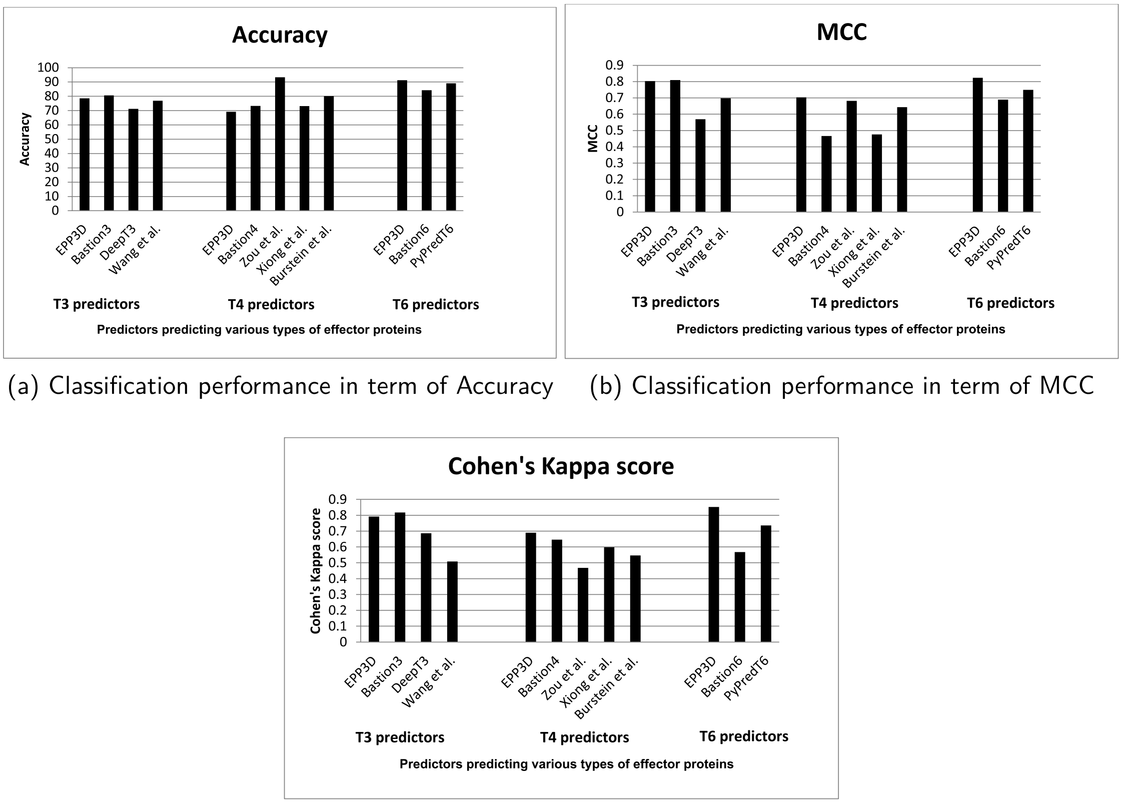


(a) Classification performance in term of Accuracy

(b) Classification performance in term of Cohen's kappa score

(c) Classification performance in term of MCC

Fig. 4. Performance comparison of various classification algorithms on effector and non-effector proteins, after balancing the dataset by different oversampling methods. The abbreviations for the methods are UB - imbalanced, ROS - Random Oversampling, SM - SMOTE, BSM - borderline SMOTE, CSM - C Smote, SLSM - Safe level SMOTE, CQNR - Cluster Quality based Non-Reductional Oversampling. As observed from the graphs, CQNR with consensus-based classifier (EPP3D) has provided superior performance over the other oversampling algorithms while classifying the effectors.



(a) Classification performance in term of Accuracy

(b) Classification performance in term of MCC

(c) Classification performance in term of Cohen's Kappa (κ) score

Fig. 5. Comparison of classification performance of EPP3D with different effector predictors.

The primary data and the feature set of the aforesaid existing methods are completely different from that used in EPP3D. In order to compare the performance of these existing methods with EPP3D, we have collected the peptide sequences of the corresponding PDB structures of T3, T4 and T6 effectors. A summary of the comparison has been depicted in Fig. 5. A detailed tabulated representation of the same has been provided in Supplementary Information Table S12.

Classification of T3 effector proteins using EPP3D has resulted in an accuracy of 78.65%, MCC of 0.8026 and κ score of 0.7913; whereas Bastion3 has resulted in an accuracy of 92.7%, MCC of 0.809 and κ score of 0.8174. DeepT3 has resulted in an accuracy of 81.2%, MCC of 0.569 and κ score of 0.6864. The technique developed by Wang et al. has obtained an accuracy of 86.88%, MCC of 0.6979 and κ score of 0.5079. Here Bastion3 has shown the highest accuracy, MCC value and κ score.

EPP3D has classified T4 effector and non-effector proteins with an accuracy of 69.24%, MCC of 0.7038 and κ score of 0.6893. The algorithm developed by Burstein et al. [20] has achieved an accuracy of 80.2%, MCC of 0.643 and κ score of 0.546 for prediction of T4 effectors, while the method of Zou et al. [19] has reported an accuracy of 93.3%, MCC of 0.682 and κ score of 0.4679. Bastion4 predictor has rendered an accuracy of 73.3%, a low MCC of 0.466 and κ score of 0.6457. The method of Xiong et al. has resulted in an accuracy of 73.2%, MCC of 0.476 and κ score of 0.5975. Here, EPP3D has shown the best performance with respect to MCC and κ score, while Zou et al. has shown the best performance with respect to accuracy.

T6 effector proteins have been classified by EPP3D with an accuracy of 91.23%, MCC of 0.8233 and κ score of 0.8523. Bastion6 predictor has provided an accuracy of 84.3%, MCC of 0.689 and κ score of 0.568. PyPredT6 has reported an accuracy of 89.12%, MCC of 0.7492, and κ score of 0.736. EPP3D has provided much better accuracy in classifying T6 effector proteins based on their 3D structures.

In order to assess and compare the performance of the aforesaid existing methods with EPP3D, we have considered three individual lists for each of T3, T4 and T6 effector proteins. Each list contains 20 independent non-overlapping experimentally verified effectors proteins.

Among 20 T3 effector proteins, EPP3D has been able to predict 15 proteins correctly, whereas Bastion3 [17] has been able to predict 17 proteins. DeepT3 [18] has predicted 12 proteins correctly. The method of Wang et al. [75] has been able to predict 11 T3 proteins. Among 20 T4 effector proteins, EPP3D has been able to predict 18 proteins correctly. Bastion4 [22], however, was unable to generate any predictive result. The algorithm developed by Zou et al. [19] has been able to predict 12 proteins correctly, while Xiong et al. [21] have predicted 13 proteins correctly, and Burstein et al. [20] have predicted 11 out of 20 proteins correctly. Among 20 T6 effector proteins, EPP3D has been able to predict 17 proteins correctly. Bastion6, however, has been unable to generate any predictive result. PyPredT6 [24] has been able to predict 14 T6 effector proteins correctly. The summary of the results has been provided in Supplementary Information Tables S13–S15.

4. Discussion

In this section, we discuss the qualitative comparison of CQNR and EPP3D with the current state-of-the-art investigations.

4.1. Comparison of CQNR with other oversampling algorithms

CQNR has been compared with five existing oversampling methods, namely, Random oversampling [29], SMOTE [26], Borderline-SMOTE [30], C-SMOTE [31], and Safe-level-SMOTE [32]. In the random oversampling algorithm, minority class samples are duplicated at random, such that the majority and the minority classes become balanced, thereby leading to a severe drawback of overfitting.

Generation of overfitted classifiers due to random oversampling has led to the development of SMOTE [26]. In SMOTE, an entirely new synthetic dataset is conceived from the original minority dataset to form a new set containing the original samples and new synthetic samples. However, a high SMOTE rate may lead to overfitting and adversely influence the prediction performance of the minority class. SMOTE has randomly generated synthetic points, and many of them have been generated in the region where minority class samples do not exist. Borderline-SMOTE [30], another oversampling method, is a tweak of SMOTE, designed to do away with the ambiguities of SMOTE. Borderline-SMOTE is exclusively applicable to datasets, where the number of borderline samples is low. Borderline-SMOTE [30] has divided the points into three categories - noise, danger, and safe. Noise samples of a minority class are those that have a maximum number of majority class samples as their nearest neighbors. Danger samples are the borderline samples having a mixture of minority and majority class samples as their nearest neighbors. Safe samples have the maximum number of minority class samples as their nearest neighbors. It oversamples only the borderline samples of the minority class. A limitation of borderline-SMOTE is the following. If the number of danger samples is low compared to the others, the synthetic samples generated by borderline-SMOTE may not balance the final dataset. In such a scenario, the number of danger samples will have to be large enough, which would lead to clustering of synthetic data around the limited boundary samples.

C-SMOTE [31] comprises the same procedure as SMOTE [26], except that it has generated the best SMOTE rate such that the classification results in maximum accuracy. The method uses a classifier ensemble to attain an optimal SMOTE rate and implement oversampling based on this SMOTE rate. For the present datasets, the number of synthetic samples generated is more than the number of majority class samples.

Safe-Level-SMOTE [32], another variation of SMOTE, splits the initial minority class samples into three categories, safe, borderline, and noise, and discards the noise samples. Only the safe synthetic samples have been considered for oversampling. Safe-Level-SMOTE generates a synthetic sample in the space densely populated by the original samples, which may lead to overfitting. Hence, the generation of synthetic samples has been restricted to the center of the dataset.

A significant drawback of all the above algorithms is that if a minority class is clustered, these algorithms may generate synthetic samples between these clusters. None of these algorithms ensures generation of synthetic samples in the vicinity of the minority class samples and not near majority class samples. The area outside the clusters of minority class samples may belong to the majority class. CQNR checks whether the distance between the cluster center and a new synthetic sample generated in that particular cluster is less than the radius of the cluster. If yes, the synthetic sample is added to the minority class; if no, the sample is discarded, and a new sample is generated. Another major drawback of some of these algorithms is that they eliminate samples which are noise. In biological datasets, deletion of any samples as noise would lead to loss of crucial information. CQNR does not eliminate any samples as noise, thus keeping the original dataset intact.

For different oversampling algorithms, reasonable sets of parameter values have been experimentally determined and used. For borderline-SMOTE [30] and safe-level-SMOTE [32], $k = 5$, and the number of random samples to be selected from k neighbors has been taken as three. The threshold value used by these algorithms, for deciding whether a minority class sample is noise, danger, or safe has been set to six. In other words, if the number of majority class samples in k nearest neighbors of a minority sample is less than the threshold value, the sample is said to be safe. If the number of majority class samples in k nearest neighbors of a minority sample is equal to the threshold value, the samples are said to be borderline. On the other hand, if the number of majority class samples in k nearest neighbors of a minority sample is more than the threshold value, the minority sample is classified as noise. The number of clusters, predefined in C-SMOTE [31], has been set to six.

4.2. Comparison of EPP3D with other effector protein predictors

As mentioned in Section 3.4, a few attempts have been made towards classification of effector proteins based on their peptide sequences [16–24]. Prediction of T3 effector proteins in genomes of gram-negative bacteria has been done by Yang et al. The authors have used Support Vector Machine (SVM) on N-terminal of amino acid sequences to predict novel T3 effector proteins [16]. A two-layered ensemble predictor, called Bastion3 [17], has predicted T3 effector proteins. Bastion3 is based on the features obtained from N-terminal of the proteins. Another investigation of Wang et al. [75] has used SVM to predict effector proteins based on the features obtained from N and C terminals of the proteins. Xue et al. [18] have used deep learning framework, called DeepT3, to predict T3 effector proteins taking only the first 100 residues for prediction. Bastion3 has shown the maximum accuracy, MCC values, and κ score.

Identification of T4 effector proteins has been made based on amino acid composition. Zou et al. [19] have used SVM to predict T4 effector proteins. In the investigation of Burstein et al., the ORFs of the proteins in *Legionella pneumophila* have been classified either as effector or non-effector proteins using a machine learning approach [20]. Xiong et al. [21] and Wang et al. [22] have predicted T4 effectors using ensemble classifiers based only on C-terminal features. The latter group has developed Bastion4 to predict T4 effectors [22]. EPP3D has shown the best performance with respect to MCC and κ score, while Zou et al. has shown the best performance with respect to accuracy. For identification of T6 effector proteins, Bastion6, an SVM based T6 effector protein predictor [23], and PyPredT6 [24], an ensemble learning-based predictor [24] are the two currently available tools. EPP3D has provided much better prediction accuracy for T6 effector proteins.

EPP3D, based on their 3D structural features, has reported stable performance in terms of the performance measures. However, such a trend is not noticed for the other classifiers, except for Bastion3, the classifier that classifies T3 effectors and non-effectors. Another issue has been noticed regarding consideration of the non-effector dataset. For example, Bastion6 has considered the non-effector set of Zou et al. The method of Zou et al. classifies T4 effectors and non-effectors, where the non-effectors are those that are not T4 effectors. Due to the multi-functional nature of prokaryotic genes [42], this may not be a reliable approach. Proteins that are not T4 effectors may have an association with T6SS machinery. Likewise, Yang et al. have extracted the effectors from *P. syringae*, and the remaining proteins from the entire genome have been treated as non-effectors. In contrast to these, we have taken the non-effector dataset from an experimentally verified non-pathogenic organism.

5. Conclusions

In this article, we have developed a novel oversampling technique, called CQNR, and an effector protein predictor, termed as EPP3D, based on 3D structure of proteins. We have depicted how the application of the oversampling technique has helped in better classification of the effectors and the development of EPP3D. The experiments show that CQNR effectively has resolved the shortcomings of some existing algorithms. CQNR has resulted in superior performance over some existing algorithms as well as sustained the essence of the original dataset.

In order to demonstrate the effectiveness of the present method, we have considered a dataset derived from 3D structures of experimentally verified T3, T4, and T6 effector proteins. Only 3D structural patterns have been considered here as earlier investigations have already reported the inconclusive nature of 1D (amino acid sequence) and 2D (alpha helices, beta sheets, and random coils among others) features in differentiating T3 and T4 effectors. These feature patterns can be used for distinguishing the known effector proteins from non-effectors as well as discovering the novel effector proteins. The sample size here is considerably limited since only a few known resources are available for effector proteins.

We have also developed EPP3D for classification of unknown proteins into T3, T4, and T6 effector proteins against other secreted proteins (T1, T2, T5, T7) and non-effector proteins. Since the original training dataset is imbalanced, we have used CQNR to balance the dataset. A considerable increase in the performance in classification of effector proteins after applying CQNR and a consensus of classifiers has been reported. As a future scope, we intend to incorporate more features based on 3D structure of effector proteins along with the existing ones to develop a more robust classifier. With the discovery of new secretion systems, we intend to include more types of effector proteins for a more versatile classifier.

Funding

This work has no funding.

Author's contribution

RS has conceptualized the idea, conducted the experiments, and prepared the first draft of the article. ST and RKD gave theoretical input. RKD guided the work. ST and RKD read and corrected the article.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Ethical approval

Ethical approval is not required as the data are de-identified.

Informed consent

Informed consent is not needed for this work as no human subject is involved and data are de-identified.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2019.103374.

References

[1] T.R. Costa, C. Felisberto-Rodrigues, A. Meir, M.S. Prevost, A. Redzej, M. Trokter, G. Waksman, Secretion systems in gram-negative bacteria: structural and mechanistic insights, Nat. Rev. Microbiol. 13 (6) (2015) 343.

[2] R.G. Gerlach, M. Hensel, Protein secretion systems and adhesins: the molecular armory of gram-negative pathogens, International Journal of Medical Microbiology 297 (6) (2007) 401–415.

[3] E. Durand, C. Cambillau, E. Cascales, L. Journet, Vgrg, tae, tle, and beyond: the versatile arsenal of type VI secretion effectors, Trends Microbiol. 22 (9) (2014) 498–507.

[4] A. Economou, P.J. Christie, R.C. Fernandez, T. Palmer, G.V. Plano, A.P. Pugsley, Secretion by numbers: protein traffic in prokaryotes, Mol. Microbiol. 62 (2) (2006) 308–319.

[5] J.E. Galán, M. Lara-Tejero, T.C. Marlovits, S. Wagner, Bacterial type III secretion systems: specialized nanomachines for protein delivery into target cells, Annu. Rev. Microbiol. 68 (2014) 415–438.

[6] J.S. Pearson, Y. Zhang, H.J. Newton, E.L. Hartland, Post-modern pathogens: surprising activities of translocated effectors from E. coli and Legionella, Curr. Opin. Microbiol. 23 (2015) 73–79.

[7] A.B. Russell, S.B. Peterson, J.D. Mougous, Type VI secretion system effectors: poisons with a purpose, Nat. Rev. Microbiol. 12 (2) (2014) 137.

[8] E.L. Zechner, S. Lang, J.F. Schildbach, Assembly and mechanisms of bacterial type IV secretion machines, Philosophical Transactions of the Royal Society B 367 (1592) (2012) 1073–1087.

[9] M. Basler, Type VI secretion system: secretion by a contractile nanomachine, Philosophical Transactions of the Royal Society B 370 (1679) (2015) 20150021.

[10] X. Yang, Y. Guo, J. Luo, X. Pu, M. Li, Effective identification of gram-negative bacterial type III secreted effectors using position-specific residue conservation profiles, PLoS One 8 (12) (2013) e84439.

[11] J.H. Chang, D. Desveaux, A.L. Creason, The abcs and 123s of bacterial secretion systems in plant pathogenesis, Annu. Rev. Phytopathol. 52 (2014) 317–345.

[12] J.E. Galan, R. Curtiss, Cloning and molecular characterization of genes whose products allow salmonella typhimurium to penetrate tissue culture cells, Proc. Natl. Acad. Sci. 86 (16) (1989) 6383–6387.

[13] G.A. Kuldau, G. De Vos, J. Owen, G. McCaffrey, P. Zambryski, The virb operon of agrobacterium tumefaciens ptic58 encodes 11 open reading frames, Mol. Gen. Genet. 221 (2) (1990) 256–266.

[14] E. Cascales, P.J. Christie, The versatile bacterial type IV secretion systems, Nat. Rev. Microbiol. 1 (2) (2003) 137.

[15] S. Pukatzki, A.T. Ma, D. Sturtevant, B. Krastins, D. Sarracino, W.C. Nelson, J.F. Heidelberg, J.J. Mekalanos, Identification of a conserved bacterial protein secretion system in vibrio cholerae using the dictyostelium host model system, Proc. Natl. Acad. Sci. 103 (5) (2006) 1528–1533.

[16] Y. Yang, J. Zhao, R.L. Morgan, W. Ma, T. Jiang, Computational prediction of type III secreted proteins from gram-negative bacteria, BMC Bioinf. 11 (1) (2010) S47.

[17] J. Wang, J. Li, B. Yang, R. Xie, T.T. Marquez-Lago, A. Leier, M. Hayashida, T. Akutsu, Y. Zhang, K.-C. Chou, et al., Bastion3: a two-layer ensemble predictor of type III secreted effectors, Bioinformatics 35 (12) (2018) 2017–2028.

[18] L. Xue, B. Tang, W. Chen, J. Luo, Deept3: deep convolutional neural networks accurately identify gram-negative bacterial type III secreted effectors using the n-terminal sequence, Bioinformatics 35 (12) (2018) 2051–2057.

[19] L. Zou, C. Nan, F. Hu, Accurate prediction of bacterial Type IV secreted effectors using amino acid composition and PSSM profiles, Bioinformatics 29 (24) (2013) 3135–3142.

[20] D. Burstein, T. Zusman, E. Degtyar, R. Viner, G. Segal, T. Pupko, Genome-scale identification of Legionella pneumophila effectors using a machine learning approach, PLoS Pathog. 5 (7) (2009) e1000508.

[21] Y. Xiong, Q. Wang, J. Yang, X. Zhu, D. Wei, Predt4se-stack: prediction of bacterial type IV secreted effectors from protein sequences using a stacked ensemble method, Front. Microbiol. 9 (2018) 2571.

[22] J. Wang, B. Yang, Y. An, T. Marquez-Lago, A. Leier, J. Wilksch, Q. Hong, Y. Zhang, M. Hayashida, T. Akutsu, et al., Systematic analysis and prediction of type iv secreted effector proteins by machine learning approaches, Briefings Bioinf. 20 (3) (2017) 931–951.

[23] J. Wang, B. Yang, A. Leier, T.T. Marquez-Lago, M. Hayashida, A. Rocker, Y. Zhang, T. Akutsu, K.-C. Chou, R.A. Strugnell, et al., Bastion6: a bioinformatics approach for accurate prediction of type vi secreted effectors, Bioinformatics 34 (15) (2018) 2546–2555.

[24] R. Sen, L. Nayak, R. K. De, Pypredt6: a python based prediction tool for identification of type vi effector proteins, J. Bioinform. Comput. Biol. 17 (3) (n.d).

[25] R. Sen, L. Nayak, R.K. De, A review on host-pathogen interactions: classification and prediction, Eur. J. Clin. Microbiol. Infect. Dis. 35 (10) (2016) 1581–1599.

[26] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[27] M.M. Rahman, D. Davis, Addressing the class imbalance problem in medical datasets, International Journal of Machine Learning and Computing 3 (2) (2013) 224.

[28] C.-Y. Chang, M.-T. Hsu, E.X. Esposito, Y.J. Tseng, Oversampling to overcome overfitting: exploring the relationship between data set composition, molecular descriptors, and predictive modeling methods, J. Chem. Inf. Model. 53 (4) (2013) 958–971.

[29] M. Kubat, S. Matwin, et al., Addressing the curse of imbalanced training sets: one-sided selection, ICML 97 (1997) 179–186 Nashville, USA.

[30] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE, A new over-sampling method in imbalanced data sets learning, Advances in Intelligent Computing (2005) 878–887.

[31] G. He, H. Han, W. Wang, An over-sampling expert system for learing from imbalanced data sets, Neural Networks and Brain vol. 1, ICNN&B'05, 2005, pp. 537–541 IEEE, 2005.

[32] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-smote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, Advances in Knowledge Discovery and Data Mining (2009) 475–482.

[33] Y.-D. Zhang, Y. Zhang, P. Phillips, Z. Dong, S. Wang, Synthetic minority oversampling technique and fractal dimension for identifying multiple sclerosis, Fractals 25 (04) (2017) 1740010.

[34] Y.-D. Zhang, G. Zhao, J. Sun, X. Wu, Z.-H. Wang, H.-M. Liu, V.V. Govindaraj, T. Zhan, J. Li, Smart Pathological Brain Detection by Synthetic Minority Oversampling Technique, Extreme Learning Machine, and Jaya Algorithm, Multimedia Tools and Applications, 2017, pp. 1–20.

[35] D. Haller, L. Holt, S.C. Kim, R.F. Schwabe, R.B. Sartor, C. Jobin, Transforming growth factor-β1 inhibits non-pathogenic gramnegative bacteria-induced NF-κb recruitment to the interleukin-6 gene promoter in intestinal epithelial cells through modulation of histone acetylation, J. Biol. Chem. 278 (26) (2003) 23851–23860.

[36] O. Wurtzel, N. Sesto, J.R. Mellin, I. Karunker, S. Edelheit, C. Bécavin, C. Archambaud, P. Cossart, R. Sorek, Comparative transcriptomics of pathogenic and non-pathogenic listeria species, Mol. Syst. Biol. 8 (1) (2012) 583.

[37] Y. An, J. Wang, C. Li, J. Revote, Y. Zhang, T. Naderer, M. Hayashida, T. Akutsu, G.I. Webb, T. Lithgow, et al., Secretepdb: a comprehensive web-based resource for secreted effector proteins of the bacterial types III, IV and VI secretion systems, Sci. Rep. 7 (2017) 41031.

[38] D. Bi, L. Liu, C. Tai, Z. Deng, K. Rajakumar, H.-Y. Ou, Secret4: a web-based bacterial type IV secretion system resource, Nucleic Acids Res. 41 (D1) (2012) D660–D665.

[39] J. Li, Y. Yao, H.H. Xu, L. Hao, Z. Deng, K. Rajakumar, H.-Y. Ou, Secret6: a web-based resource for type VI secretion systems found in bacteria, Environ. Microbiol. 17 (7) (2015) 2196–2202.
[40] H. Berman, K. Henrick, H. Nakamura, J.L. Markley, The worldwide protein data bank (wwpdb): ensuring a single, uniform archive of pdb data, Nucleic Acids Res. 35 (suppl_1) (2006) D301–D303.
[41] P.R. Pereira, L.G. Fernandes, G.O. de Souza, S.A. Vasconcellos, M.B. Heinemann, E.C. Romero, A.L. Nascimento, Multifunctional and redundant roles of leptospira interrogans proteins in bacterial-adhesion and fibrin clotting inhibition, International Journal of Medical Microbiology 307 (6) (2017) 297–310.
[42] C. Kaimer, P.L. Graumann, Players between the worlds: multifunctional DNA translocases, Curr. Opin. Microbiol. 14 (6) (2011) 719–725.
[43] B. Xayarath, H. Marquis, G.C. Port, N.E. Freitag, Listeria monocytogenes ctap is a multifunctional cysteine transport-associated protein required for bacterial pathogenesis, Mol. Microbiol. 74 (4) (2009) 956–973.
[44] Y.G. Santin, E. Cascales, Domestication of a housekeeping transglycosylase for assembly of a type VI secretion system, EMBO Rep. 18 (1) (2017) 138–149.
[45] R.A. Günster, S.A. Matthews, D.W. Holden, T.L. Thurston, Ssek1 and ssek3 type III secretion system effectors inhibit nf-κb signaling and necroptotic cell death in salmonella-infected macrophages, Infect. Immun. 85 (3) (2017) e00010–17.
[46] T. Bose, K. Venkatesh, S.S. Mande, Computational analysis of host–pathogen protein interactions between humans and different strains of enterohemorrhagic escherichia coli, Frontiers in cellular and infection microbiology 7 (2017) 128.
[47] M. Stout, J. Bacardit, J.D. Hirst, N. Krasnogor, Prediction of recursive convex hull class assignments for protein residues, Bioinformatics 24 (7) (2008) 916–923.
[48] M. DATT, Geometric analysis of the conformational features of protein structures, BIOMAT 2015: Proceedings of the International Symposium on Mathematical and Computational Biology, World Scientific, 2016, p. 166.
[49] K.C. Dee, D.A. Puleo, R. Bizios, An Introduction to Tissue-Biomaterial Interactions, John Wiley & Sons, 2003.
[50] M.H. Zehfus, G.D. Rose, Compact units in proteins, Biochemistry 25 (19) (1986) 5759–5765.
[51] D.-T. Lee, B.J. Schachter, Two algorithms for constructing a delaunay triangulation, Int. J. Comput. Inf. Sci. 9 (3) (1980) 219–242.
[52] N.F. Goodacre, D.L. Gerloff, P. Uetz, Protein domains of unknown function are essential in bacteria, mBio 5 (1) (2014) e00744–13.
[53] M.L. Connolly, Solvent-accessible surfaces of proteins and nucleic acids, Science 221 (4612) (1983) 709–713.
[54] K. Dalal, Counting the onion, Random Struct. Algorithm 24 (2) (2004) 155–165.
[55] J. Chen, N. Sawyer, L. Regan, Protein–protein interactions: general trends in the relationship between binding affinity and interfacial buried surface area, Protein Sci. 22 (4) (2013) 510–515.
[56] H. Hegyi, M. Gerstein, The relationship between protein structure and function: a comprehensive survey with application to the yeast genome 1, J. Mol. Biol. 288 (1) (1999) 147–164.
[57] F.M. Richards, Packing defects, cavities, volume fluctuations, and access to the interior of proteins. including some general comments on surface area and protein structure, Carlsberg Res. Commun. 44 (2) (1979) 47.
[58] A. Shahmoradi, C.O. Wilke, Dissecting the roles of local packing density and longer-range effects in protein sequence evolution, Proteins: Structure, Function, and Bioinformatics 84 (6) (2016) 841–854.
[59] N. Leo, J. Liu, I. Archbold, Y. Tang, X. Zeng, Ionic strength, surface charge, and packing density effects on the properties of peptide self-assembled monolayers, Langmuir 33 (8) (2017) 2050–2058.
[60] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 2 (1979) 224–227.
[61] P. Vora, B. Oza, et al., A survey on k-mean clustering and particle swarm optimization, Int. J. Sci. Math. Educ. 1 (3) (2013) 1–14.
[62] A. Ghosh, B.C. Dhara, R.K. De, Comparative analysis of cluster validity indices in identifying some possible genes mediating certain cancers, Molecular Informatics 32 (4) (2013) 347–354.
[63] J. Hu, X. He, D.-J. Yu, X.-B. Yang, J.-Y. Yang, H.-B. Shen, A new supervised over-sampling algorithm with application to protein-nucleotide binding residue prediction, PLoS One 9 (9) (2014) e107676.
[64] M.S. Santos, P.H. Abreu, P.J. García-Laencina, A. Simão, A. Carvalho, A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients, J. Biomed. Inform. 58 (2015) 49–59.
[65] S. Zhang, X. Duan, Prediction of protein subcellular localization with oversampling approach and chou's general PseAAC, J. Theor. Biol. 437 (2018) 239–250.
[66] L. Lam, S. Suen, Application of majority voting to pattern recognition: an analysis of its behavior and performance, IEEE Trans. Syst. Man Cybern. A Syst. Hum. 27 (5) (1997) 553–568.
[67] D. Dua, C. Graff, UCI machine learning repository, URL http://archive.ics.uci.edu/ml.
[68] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[69] R. Espíndola, N. Ebecken, On extending f-measure and g-mean metrics to multi-class problems, WIT Transactions on Information and Communication Technologies 35, URL https://doi.org/10.2495/DATA050031.
[70] A.J. Tallón-Ballesteros, J.C. Riquelme, Data mining methods applied to a digital forensics task for supervised machine learning, Computational Intelligence in Digital Forensics: Forensic Investigation and Applications, Springer, 2014, pp. 413–428.
[71] G. Jurman, S. Riccadonna, C. Furlanello, A comparison of MCC and CEN error measures in multi-class prediction, PLoS One 7 (8) (2012) e41882.
[72] A. Ben-David, About the relationship between ROC curves and cohen's kappa, Eng. Appl. Artif. Intell. 21 (6) (2008) 874–882.
[73] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, Pattern Recognit. 44 (8) (2011) 1761–1776.
[74] J.L. Fleiss, J. Cohen, The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability, Educ. Psychol. Meas. 33 (3) (1973) 613–619.
[75] Y. Wang, Q. Zhang, M.-a. Sun, D. Guo, High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles, Bioinformatics 27 (6) (2011) 777–784.

Rishika Sen received her BSc and MSc degrees in Computer Science from University of Calcutta, India, in 2012 and 2014 respectively. She is currently working towards the Doctorate Degree at Machine Intelligence Unit in Indian Statistical Institute. Her current research interest includes bioinformatics, computational biology and machine learning.

Somnath Tagore received the BSc degree from the University of Calcutta, in 2003, the MSc degree from Manipal University, in 2005, the MTech degree from the West Bengal University of Technology, in 2007, and the PhD (Engg) degree in engineering from ISI (Jadavpur University), in 2014. He was a post-doctoral fellow with the Cancer Genomics and BioComputing Lab, Faculty of Medicine, Bar-Ilan University, Safed, 676 Israel. Currently, he is working as a research scientist with the Califano Laboratory of Systems Biology, Columbia University Medical Center, Herbert Irving Cancer Research Center, New York. His current research interests include systems biology, network medicine, infectious disease modeling, cancer metabolomics, data mining, in-silico metabolic engineering, algorithms, graphs, and optimization.

Rajat K. De is a Professor of Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. He obtained his Ph.D. degree from the same Institute in the year 2000. He was a Distinguished Postdoctoral Fellow at the Whitaker Biomedical Engineering Institute, Johns Hopkins University, USA, during 2002–2003. Professor De visited the Department of Medicine, University of California, San Diego, in 2017 and 2018, with a Fulbright-Nehru Academic and Professional Excellence Fellowship. He has published about 100 research papers in international journals, conference proceedings and in edited books, and coedited three books to his credit. His research interest include machine learning, bio-engineering, computational biology, bioinformatics, systems biology and big data analytics.