Computational and Structural Biotechnology Journal 20 (2022) 165–174

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

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

SortPred: The first machine learning based predictor to identify bacterial sortases and their classes using sequence-derived information

Adeel Malik [a], Sathiyamoorthy Subramaniyam [b], Chang-Bae Kim [c,*], Balachandran Manavalan [d,*]

[a] Institute of Intelligence Informatics Technology, Sangmyung University, Seoul 03016, Republic of Korea
[b] Research and Development Center, Insilicogen Inc., Yongin-si 16954, Gyeonggi-do, Republic of Korea
[c] Department of Biotechnology, Sangmyung University, Seoul 03016, Republic of Korea
[d] Department of Physiology, Ajou University School of Medicine, Suwon, Republic of Korea

ARTICLE INFO

Article history:
Received 1 September 2021
Received in revised form 8 December 2021
Accepted 9 December 2021
Available online 14 December 2021

Keywords:
Sortase
Machine learning
Random forest
Cysteine transpeptidase
Hybrid features
Bioinformatics

ABSTRACT

Sortase enzymes are cysteine transpeptidases that embellish the surface of Gram-positive bacteria with various proteins thereby allowing these microorganisms to interact with their neighboring environment. It is known that several of their substrates can cause pathological implications, so researchers have focused on the development of sortase inhibitors. Currently, six different classes of sortases (A-F) are recognized. However, with the extensive application of bacterial genome sequencing projects, the number of potential sortases in the public databases has exploded, presenting considerable challenges in annotating these sequences. It is very laborious and time-consuming to characterize these sortase classes experimentally. Therefore, this study developed the first machine-learning-based two-layer predictor called SortPred, where the first layer predicts the sortase from the given sequence and the second layer predicts their class from the predicted sortase. To develop SortPred, we constructed an original benchmarking dataset and investigated 31 feature descriptors, primarily on five feature encoding algorithms. Afterward, each of these descriptors were trained using a random forest classifier and their robustness was evaluated with an independent dataset. Finally, we selected the final model independently for both layers depending on the performance consistency between cross-validation and independent evaluation. SortPred is expected to be an effective tool for identifying bacterial sortases, which in turn may aid in designing sortase inhibitors and exploring their functions. The SortPred webserver and a standalone version are freely accessible at: https://procarb.org/sortpred.

1. Introduction

The cysteine transpeptidase enzymes, commonly known as "sortases" are present in all Gram-positive bacteria, some Gram-negative bacteria, and a few species of archaea [1,2]. These sortase enzymes covalently link proteins with their cell walls, thus regulating surface architecture [3]. Sortases encoded by pathogenic bacteria are known to recognize various proteins with pathological roles such as adhesion, immune response, and acquisition of required nutrients [4–6]. Therefore, sortases are one of the key factors for virulence, and may thus be used as a target for fighting bacterial infections [6]. In light of these factors, several studies have focused on identifying sortase inhibitors against methicillin-resistant bacteria [7] or bacteria that infect animals with significance in aquaculture [8]. The sortase inhibitors are expected to block the function of these sortases and have been extensively reviewed elsewhere [9,10]. Furthermore, sortase enzymes are known to be involved in a variety of industrial applications. One such important application is 'Sortagging' (sortase-mediated transpeptidation), a versatile chemoenzymatic ligation strategy for site-specific labelling of proteins with small probes [11], a strategy widely used to immobilize proteins on surfaces, labelling proteins, protein cyclization and protein dimerization [12].

Proteins that are recognized by sortase enzymes have cell wall sorting signal (CWSS) towards the C-terminal end, which includes a LPXTG (L: leucine, P: proline, X: any amino acid, T: threonine and G: glycine) recognition motif, a hydrophobic region of about 20 residues and a tail region with positively charged amino acids (lysine/arginine) [9]. *Staphylococcus aureus* sortase A (SaSrtA) is one of the most well-studied sortase enzyme. A two-step reaction

* Corresponding authors.
E-mail addresses: evodevo@smu.ac.kr (C.-B. Kim), bala@ajou.ac.kr (B. Manavalan).

https://doi.org/10.1016/j.csbj.2021.12.014

mechanism is involved in anchoring SaSrtA to the cell envelope. First, the SaSrtA enzyme cleaves the LPXTG motif between T and G residues, producing a thioester acyl-enzyme intermediate that is resolved and then transferred to the cell wall via a SrtA-mediated transpeptidation reaction [9].

Sortases are grouped into different families based on their amino acid sequences (class A to F enzymes) [4]. Nevertheless, irrespective of their classification status, all these sortase enzymes share a few common features: a highly conserved catalytic triad consisting of amino acids HIS, CYS, and ARG [13,14]. On the other hand, in class F enzymes from Actinobacteria, a highly conserved ASN was present instead of an ARG [15]. Sortase enzymes bind a variety of substrates and thereby regulate different functions, including sporulation, pilus assembly, ion acquisition, and other general housekeeping roles of the cell [12]. Particularly, class A sortases present in Firmicutes perform cellular housekeeping functions. The class B sortases are also predominant in Firmicutes and have a wide variety of roles, including the attachment of haem-receptors to the peptidoglycan and pilus assembly. Class C-type sortases are found in both Firmicutes and Actinobacteria, with the exception of *Streptomyces* family [15], which act as pilin polymerases to help pili formation. Sortases belonging to class D are predominant in Bacilli and are known to facilitate sporulation. Contrarily, class E and F enzymes are found in Actinobacteria, and their functional roles remain unknown [4]. However, studies have indicated that class E enzymes may play a role in developing aerial hyphae in *Streptomyces coelicolor* [16].

Despite the fact that the classification of these sortases has been updated regularly, majority of them have not been assigned to any sortase class. Due to the widespread sequencing of bacterial genomes, the number of potential sortase sequences has increased rapidly in the public databases, posing a greater challenge to annotate these sequences. Furthermore, experimental identification and classification of sortases are time consuming and expensive. Hence, computational approaches offer a robust means of accurately identifying sortase enzymes from their primary sequences. Currently, the only methods available to identify and classify sortases are based on the sequence-similarity approaches such as BLAST [17,18] and HMMER [19]. A major disadvantage of such methods is that they only work if the given sequence shares some degree of sequence similarity with the existing sortase sequences. As a result, these approaches are not efficient in detecting novel sortases. Therefore, machine learning (ML) based methods provide promising alternatives to develop prediction models for sortase classification.

In this study, we developed the first two-layer predictor called SortPred. The first layer identifies whether a given sequence belongs to sortase or not, and the second layer identifies one of the six classes (A-F) of the predicted sortase. An overall framework for SortPred is shown in Fig. 1. To develop the SortPred, we employed five different sequence-based encodings, including amino acid composition (AAC), composition/transition/distribution (CTD), conjoint triad (CTriad), dipeptide composition (DPC), and quasi-sequence-order (QSO), and their possible combinations (hybrid features). Afterward, these features are trained using an RF binary classifier for the first layer prediction and an RF multi-label classifier for the second layer prediction. Finally, we independently selected the best model for two layers based on the consistent cross-validation and independent evaluation results. To our knowledge, this is the first time a ML-based method has been used

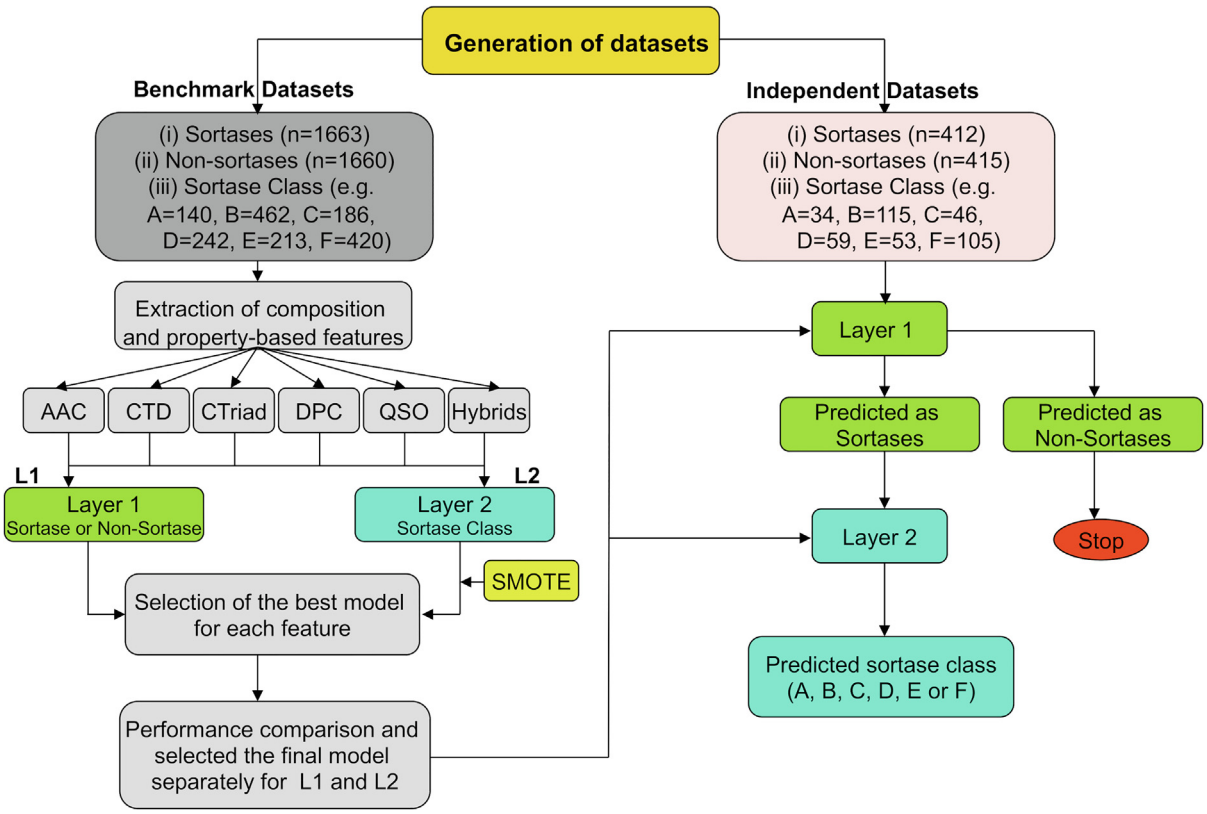

Fig. 1. An overview of the proposed methodology for predicting sortase enzymes. The benchmark and independent datasets for Layer 1 consist of sortases and non-sortases, whereas Layer 2 consists of sequences representing the individual sortase classes. Both layers use five composition-based and property-based features (AAC, CTD, CTriad, DPC & QSO) and their hybrids in a 10-fold cross validation using RF to identify the best models from each layer. During cross-validation, the SMOTE algorithm is used to handle the imbalance data for layer 2. The performance of each of the selected models is evaluated separately on the independent dataset for each layer. At last, if the sequence is predicted to be as a sortase enzyme, the sequence information is passed to Layer 2 for the prediction of the sortase class.

for predicting bacterial sortases and their classes. Therefore, we anticipate our method will be an effective tool for identifying bacterial sortases, which will be useful to design sortase inhibitors and to investigate their functions in various industrial applications.

2. Materials and methods

2.1. Dataset construction

Positive dataset: We used the keyword "sortase" to search against the NCBI's protein database to construct the positive samples. All bacterial sequences with a length ranging from 100 to 500 were retained and excluded other sequences, even those containing non-standard amino acids (B|J|O|U|X|Z). To annotate sortase sequences, position-specific scoring matrix (PSSM) searches against pre-formatted conserved domain database (CDD) [20], "little_endian" (Downloaded: November 2020) were carried out by using a standalone RPS-BLAST v2.10.0+ [18] algorithm with an e-value threshold of 1e-5. For each input sequence, RPS-BLAST lists the conserved domain models that score above a certain cut-off and includes the PSSMID of the conserved domain, scores (e.g., e-value and bit score) and the actual alignment between the input sequence and the conserved domain. The output of the RPS-BLAST was further processed by running another command line utility "rpsbproc" available from the CDD website (https://ftp.ncbi.nih.gov/pub/mmdb/cdd/rpsbproc/). The rpsbproc utility converts the raw alignments into domain or site annotations on the input sequence and presents the annotation data as tab-delimited files. From the rpsbproc utility output, sequences assigned to one of the six sortase classes (Classes A, B, C, D, E and F) were selected. Using these sortase sequences, a redundancy reduced dataset was generated by applying CD-HIT v4.8.1 [21] with the 40% sequence identity cut-off. Sequences annotated as sortases without being assigned to a particular class, as well as only a limited number of marine sortases (from proteobacteria) identified in the preceding steps, were also excluded from the positive dataset. Furthermore, redundancy reduction was applied to excluded sortase sequences as well, so that they could be used for additional validation later.

Negative dataset: We constructed negative dataset as follows: (i) retrieved all the reviewed bacterial sequences having a length between 100 and 500 amino acids from the UniProt database and discarded the sequences that contained non-standard amino acids. (ii) RPS-BLAST and the rpsbproc utility (described above) were used to identify the potential sortase sequences and excluded them from the negative dataset. (iii) We further filtered the negative dataset by removing any sequence that showed a greater than 30% sequence identity to sequences from the positive dataset. In the same way as the positive dataset, we also generated a negative dataset with a CD-HIT cut-off of 40% sequence identity. A prediction model developed using a balanced dataset is generally more reliable and robust than a model developed using an imbalanced dataset [22,23]. In an imbalanced dataset, the model is overfitted to favor the sample belonging to the large class. Therefore, we randomly selected negative samples that are equivalent in number to positive samples. The combined positive and negative datasets were divided into training and independent validation sets by using the createDataPartition function of the CARET (short for Classification And REgression Training) package [24] available in R (https://www.r-project.org/). In layer 1, we used 1663 sortases and 1660 non-sortases to develop the model, followed by 412 sortases and 415 non-sortases for independent validation. For layer 2, classes A, B, C, D, E, and F each contains 140, 462, 186, 242, 213, and 420 samples for multi-class training. Those classes corresponding to independent validation are 34, 115, 46, 59, 53, and 105. A statistical summary of the dataset is provided in Table S1.

2.2. Feature generation

This work aimed to train an RF classifier that can accurately map input features extracted from primary protein sequences in order to predict if a sequence is a sortase or non-sortase, and subsequently its class (A, B, C, D, E, or F). In particular, the training dataset contain sequences of diverse length that should be converted into fixed length feature vectors using feature encoding algorithms, which is essential for RF training. In our study, we employed five different features that have been extensively used in previous works [25–27], that cover major compositional and physicochemical aspects of sequence information and are described below:

1. Amino acid composition (AAC)
   In protein sequence, the AAC consists of the fraction of each naturally occurring 20 amino acid residues, and can be calculated by using the following formula:

$$AAC(i) = \frac{AA_i}{K} \tag{1}$$

where AA_i is the number of amino acids of type i and K is the length of the protein sequence. The AAC has a fixed length of 20 features.

2. Composition (C), Transition (T), and Distribution (D) (CTD)
   The CTD descriptors have been proposed by Dubchak et al. [28,29] for predicting protein folding classes, which have several applications, such as the prediction of protein/peptide functions. A total of twenty naturally occurring standard amino acids have been grouped into three groups (polar, neutral, and hydrophobicity) according to seven different types of physicochemical properties (Table S2), including hydrophobicity, polarizability, normalized van der Waals volume, secondary structure, polarity, charge, and solvent accessibility.

In CTD, C represents the percentage composition of polar, neutral, and hydrophobic residues of a given protein. The composition descriptor can be expressed as:

$$C(a) = \frac{Za}{K}, a \in \{neutral, polar, hydrophobic\} \tag{2}$$

where Z_a is the number of amino acid of type a in the given sequence.

In CTD, T consists of three values (polar, neutral, and hydrophobic). A transition from a neutral group to a hydrophobic group is the frequency with which a neutral residue is followed by a hydrophobic residue or vice versa. The transitions between polar and neutral groups, and hydrophobic and polar groups, are also defined in the same way. T can be calculated as follows:

$$T(ab) = \frac{Z_{ab} + Z_{ba}}{K-1}, a, b \in \{(polar, neutral), (neutral, hydrophobic), (hydrophobic, polar)\} \tag{3}$$

where Z_{ab} and Z_{ba} respectively represent the numbers of dipeptide encoded as ab and ba in the sequences.

In CTD, D consists of five values for each of the three classes, and it measures the percentage of a target sequence length within which amino acids belonging to a specific property are found within 25, 50, 75, and 100% of their position. Overall, CTD generates 147-dimensional features (21 × 7), and each PCP is characterized by a 21-dimensional feature vector.

3. Conjoint triad (CTriad)
   The CTriad encodings were initially proposed by Shen et al. [30] to model protein–protein interactions. Using this encoding, any given protein sequence is represented as a vector space containing descriptors of amino acids. Subsequently, the vector space is reduced by clustering the 20 amino acids based on their dipoles and side chains volumes. As a result, the CTriad encoding generates a 343-dimensional feature vector for a given protein sequence.
4. Dipeptide composition (DPC)
   DPC gives a fixed length of 400 (20 × 20) features, which is defined as:

$$DPC(ab) = \frac{Z_{ab}}{K-1} \tag{4}$$

5. Quasi-Sequence-Order (QSO)
   QSO encoding of each protein sequence results in a fixed length of a 100-dimensional feature vector by measuring the physicochemical distance between the amino acids. A set of equations and details regarding the QSO feature encoding have been presented in previous studies [31,32].

2.3. Machine learning classifier and parameter optimization

In this study, we employed an RF classifier. Using the widely used open-source R package CARET [24], we generated several RF models based on the five main features described above and all possible combinations. In developing each feature-based model, a grid-based search was applied and parameters 'mtry' (number of variables randomly selected at each node split) and 'ntree' (number of trees to grow) were optimized. Here, mtry search space is set to 1 to 10, with a step size of 1, and ntree search space is set to 100 to 700 with a step size of 20.

Using the 10-fold cross-validation (CV) approach, we assessed the performances of a given set of feature encodings and parameters. Subsequently, selected the optimal parameter that eventually achieved the best performance. In the 10-fold CV, the training data was randomly divided into 10 subsets of which one was used as a test set and the remaining nine subsets were used for training [33,34]. Ten times this procedure was repeated in order to make sure each subset was used as a test set at least once. The performance of the 10 corresponding outcomes is averaged, with the result implying classifier's overall performance.

2.4. Performance evaluation metrics

Six commonly used metrics were used [35–37] to evaluate the performance of constructed models, including sensitivity (Sn), specificity (Sp), accuracy (ACC), balanced accuracy (BACC), F1-score and Matthews correlation coefficient (MCC). These performance metrics are calculated as follows:

$$\begin{cases} Sn = \frac{TP}{TP+FN} \\ Sp = \frac{TN}{TN+FP} \\ ACC = \frac{TP+TN}{TP+FN+TN+FP} \\ BACC = \frac{Sn+Sp}{2} \\ F1 = \frac{2\times precision\times recall}{recall+precision} \\ MCC = \frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \end{cases} \tag{5}$$

where *TP*, *TN*, *FP*, and *FN* represent the true positive, true negative, false positive, and false negative, respectively. In all cases, the higher the value, the better.

2.5. Handling imbalanced dataset by SMOTE algorithm

As explained in the above section (dataset construction) the number of samples in each specific sortase class differed considerably. Consequently, the number of sequences in the respective classes are highly imbalanced. Generally, developing an ML-based model from an imbalanced dataset can be challenging because the performance skews in the majority's favor. Therefore, to address this issue, we applied SMOTE (Synthetic minority oversampling technique) algorithm [38] on the training data by using the SmoteClassif function available within the UBL (v0.0.7) package. The SMOTE algorithm uses a combination of oversampling the minority class and undersampling the majority class for better classification performance. The method has been used successfully in various studies to eliminate the class imbalance [39–41]. Finally, each sortase class consisted of 277 sequences in the balanced training dataset, except for class E, which contained 276 sequences.

3. Results and discussion

3.1. Overall framework of SortPred

A two-step approach is more effective than a single predictor for the identification of sortase enzymes and their classes. In this work, we developed SortPred, a two-layer predictor (Fig. 1), where the first layer predicts whether a given sequence belongs to sortase enzyme or not. Using the predicted sortase sequence, the second layer predicts its class (A, B, C, D, E, and F). The two-layer framework was developed by exploring five different sequence encodings (AAC, DPC, CTD, CTriad, and QSO), along with 26 possible feature combinations. After that, each of the 31 descriptors was trained with a RF classifier using 10-fold CV and their performance was assessed. Notably, we employed a binary RF classifier for the first layer and a multi-class RF classifier for the second layer. The following section discusses various descriptors' performances in the first- and second-layer prediction.

3.2. Performance of 31 descriptors in identifying sortases on the Layer 1 training dataset

Fig. 2A shows the performance of various feature descriptors by employing the Layer 1 training dataset. Results demonstrate that DPC is the best performing feature descriptor among the five feature encodings, with an ACC of 94.9%, which is 2.9–8.8% higher than the four other features (AAC, CTD, CTriad, and QSO). Next, we examined the performance of hybrid features. In general, hybrid features have better prediction performance than their individual feature encoding contained within them. Interestingly, seven hybrid features (AAC_CTD_CTriad_DPC_QSO, AAC_CTD__DPC_QSO, AAC_CTriad_DPC_QSO, CTD_CTriad_DPC_QSO, CTD_DPC_QSO, CTriad_DPC_QSO, and DPC_QSO) achieved an ACC in the range of 95.8 to 96.4%, which is ∼ 1 to 1.5% higher than the DPC encoding. It is surprising that all seven encodings encompass DPC, indicating that DPC plays a major role whereas other encodings play a supporting role in classifying sortases from non-sortases. Generally, cross-validation performance alone is not enough to select the best model. There is a possibility that the excellent performance during cross-validation may be a result of overoptimization of the ML parameters [42–44]. As a result, we tested each model with an independent validation set and compared their performance consistency or robustness.

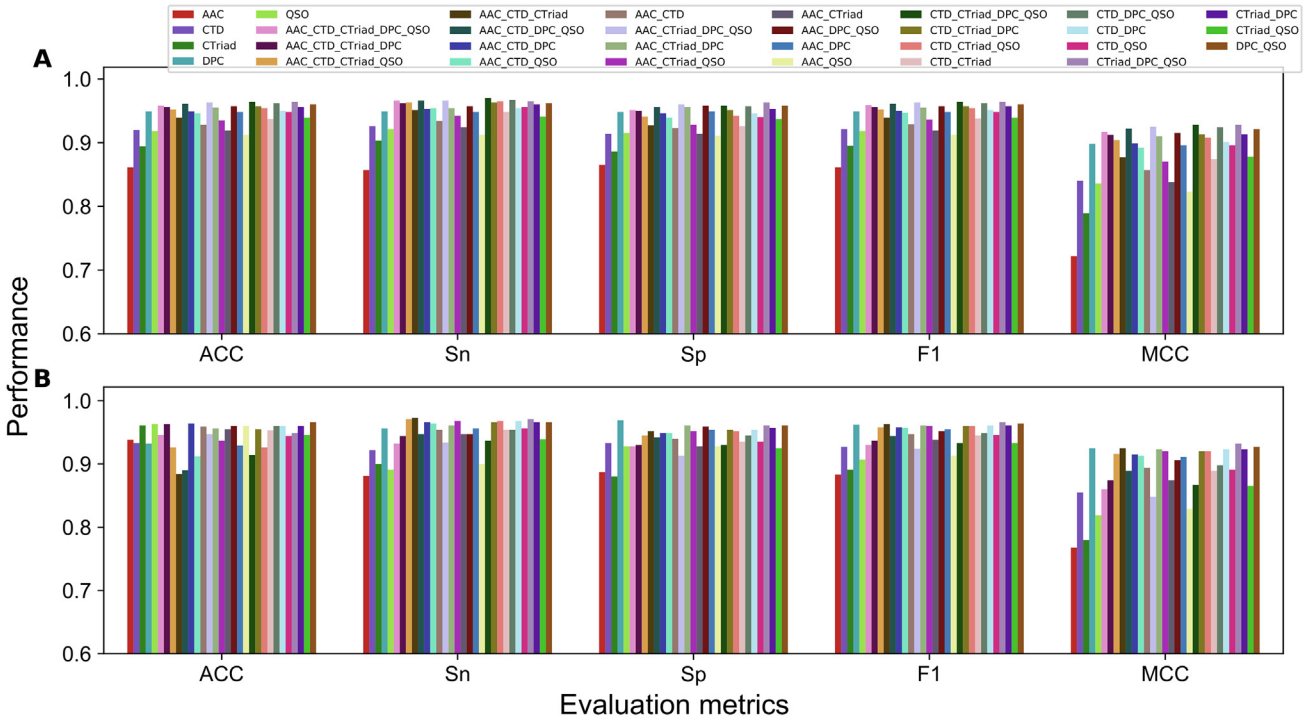

Fig. 2. An analysis of 31 feature descriptors based on random forest models on Layer 1 training dataset (A) and Layer 1 independent dataset (B). The 31 feature descriptors are represented by different colors. AAC: amino acid composition, DPC: dipeptide composition, CTD: composition transition and distribution, QSO: quasi sequence order, and CTriad: conjoint triad descriptors.

3.3. Performance of 31 descriptors on layer 1 independent validation dataset

An independent validation dataset was used to evaluate the performance of 31 models and the results are shown in Fig. 2B. Rather than solely focusing on independent performance, we compared the consistency of cross-validation and independent validation performance, particularly ACC. We observed inconsistencies in ACC between training and independent datasets for the five feature encodings (AAC, DPC, CTriad, CTD, and QSO) as shown in Fig. 2A and B. For instance, DPC was the best performer in training, but it ranked last. Similarly, QSO ranked third in training, but earned the best performance in the independent dataset. However, unlike five feature encodings, consistent performance was observed with seven hybrid features (AAC_CTD_CTriad_DPC, AAC_CTriad_DPC, AAC_DPC_QSO, CTD_CTriad_DPC, CTD_DPC_QSO, CTriad_DPC, and DPC_QSO), which achieved a ~ 96.0% ACC on both datasets. Finally, we selected CTD_DPC_QSO as the final model for SortPred (the first layer prediction) because it contained three feature encodings that achieved a consistent ACC on the training dataset (96.2%) and the independent dataset (96.0%).

3.4. Performance of various feature descriptors in classifying sortase classes based on layer 2 training and independent datasets

We assessed the performance of various feature descriptors for sortase classes prediction using an imbalanced training dataset. Interestingly, when predicting the individual classes based on imbalanced data, the model based on QSO performed the best among all the five descriptors with an ACC and MCC scores of 92.2% and 0.682 (Table 1), respectively. An analysis of model performances based on ACC or BACC would not be straightforward because of the imbalance in the dataset. Chicco et al. [45] have recently demonstrated the importance of the MCC metrics by using the datasets that are imbalanced. Hence, we adopted MCC for model comparison. Note that QSO model achieved MCC that was 5.6–13.1% higher as compared to the other four encodings. Compared to layer 1, most of the hybrid feature-based performance deteriorated, containing redundant or irrelevant features that may not be suitable for class prediction. Only three of the 26 hybrid features containing QSO features (AAC_DPC_QSO, CTD_DPC_QSO, and DPC_QSO) exhibited similar performances with MCC between 0.691 and 0.703. Specifically, whose MCC is 0.9 to 2.1% higher than the QSO model, indicating QSO plays the central role in sortase class prediction, while other encodings play the supporting role. To ensure the robustness of the models, we evaluated all of them independently.

The independent validation assessment of 31 models is presented in Table 1. We examined only three hybrid models that demonstrated superior performance during the training. Out of three models, two (DPC_QSO and AAC_DPC_QSO) achieved the MCC in the range of 0.739–0.748. Finally, we selected DPC_QSO model as it exhibited the consistent cross-validation and independent performance. Next, we examined how DPC_QSO performed for each class on training and independent datasets. Results demonstrate that classes A, B, C, and F achieved excellent performance on the training dataset with MCCs ranging from 0.715 to 0.776, while class D achieved above-average performance with MCC of 0.633, and class E achieved moderate performance (Fig. 3A). Furthermore, we observed similar performance with the same ranking for each class in the independent dataset (Fig. 3B).

3.5. SMOTE improves the layer 2 prediction performance on training and independent datasets

The sortase classes (A, B, C, D, E, and F) used in this study are highly imbalanced. Therefore, to balance the sample, we applied the SMOTE resampling technique and obtained an equal number of samples for each class. Table 2 shows that three hybrid features

Table 1
Performance comparison of different feature descriptors on Layer 2 imbalanced training and independent validation datasets.

	Training					Validation				
Features	ACC	BACC	Sn	Sp	MCC	ACC	BACC	Sn	Sp	MCC
AAC_DPC_QSO	0.924	0.824	0.699	0.950	0.703	0.933	0.853	0.749	0.956	0.739
DPC_QSO	0.922	0.820	0.692	0.949	0.696	0.935	0.858	0.758	0.958	0.748
CTD_DPC_QSO	0.922	0.822	0.695	0.950	0.691	0.928	0.840	0.725	0.954	0.710
CTD_CTriad_DPC_QSO	0.922	0.818	0.688	0.949	0.689	0.927	0.837	0.721	0.953	0.720
AAC_CTD_CTriad_DPC_QSO	0.921	0.817	0.685	0.948	0.685	0.922	0.829	0.708	0.950	0.698
QSO	0.922	0.825	0.699	0.950	0.682	0.926	0.840	0.727	0.953	0.693
AAC_CTriad_DPC_QSO	0.917	0.807	0.668	0.945	0.680	0.932	0.850	0.746	0.955	0.750
AAC_CTriad_QSO	0.919	0.813	0.68	0.947	0.680	0.926	0.839	0.727	0.952	0.719
CTriad_DPC_QSO	0.917	0.805	0.666	0.945	0.678	0.924	0.832	0.714	0.950	0.717
AAC_CTD_CTriad_QSO	0.920	0.815	0.683	0.948	0.677	0.928	0.842	0.731	0.954	0.721
AAC_CTD_DPC_QSO	0.919	0.815	0.682	0.948	0.672	0.929	0.842	0.729	0.955	0.717
AAC_QSO	0.920	0.820	0.691	0.949	0.672	0.926	0.841	0.728	0.953	0.702
CTD_CTriad_QSO	0.918	0.811	0.675	0.946	0.671	0.922	0.825	0.701	0.949	0.693
CTriad_QSO	0.916	0.804	0.664	0.945	0.668	0.923	0.829	0.709	0.949	0.707
CTD_QSO	0.919	0.818	0.688	0.948	0.667	0.921	0.825	0.700	0.950	0.675
AAC_CTD_QSO	0.916	0.812	0.677	0.947	0.657	0.926	0.838	0.723	0.954	0.698
AAC_CTD_CTriad_DPC	0.912	0.794	0.645	0.942	0.648	0.916	0.814	0.683	0.946	0.675
AAC_DPC	0.910	0.790	0.640	0.941	0.646	0.916	0.813	0.682	0.945	0.679
AAC_CTriad_DPC	0.909	0.785	0.630	0.939	0.644	0.914	0.808	0.673	0.943	0.677
CTriad_DPC	0.907	0.778	0.619	0.938	0.636	0.913	0.796	0.65	0.942	0.660
CTD_CTriad_DPC	0.909	0.787	0.635	0.94	0.636	0.914	0.807	0.670	0.945	0.658
AAC_CTD_CTriad	0.909	0.792	0.642	0.941	0.635	0.913	0.810	0.676	0.945	0.654
CTD_DPC	0.909	0.791	0.641	0.941	0.634	0.918	0.820	0.692	0.947	0.679
AAC_CTD_DPC	0.909	0.793	0.646	0.941	0.632	0.915	0.812	0.678	0.946	0.657
DPC	0.905	0.776	0.616	0.937	0.626	0.922	0.821	0.694	0.948	0.701
AAC_CTD	0.906	0.792	0.642	0.941	0.612	0.911	0.808	0.671	0.945	0.635
AAC_CTriad	0.901	0.769	0.603	0.934	0.611	0.901	0.776	0.618	0.935	0.614
CTD_CTriad	0.901	0.773	0.610	0.936	0.594	0.909	0.797	0.652	0.941	0.637
CTD	0.900	0.779	0.623	0.936	0.590	0.902	0.788	0.637	0.939	0.596
AAC	0.893	0.766	0.600	0.933	0.551	0.896	0.772	0.610	0.935	0.555
CTriad	0.883	0.726	0.531	0.922	0.533	0.885	0.733	0.543	0.923	0.542

Feature descriptors are listed in the first column. Columns 2–6 represent the ACC, BACC, Sn, Sp, and ACC obtained from the training dataset. Columns 7–11 list a metric corresponding to an independent dataset. The table is sorted based on the training data MCC scores. ACC = Accuracy, BACC = Balanced Accuracy, Sn = Sensitivity, Sp = Specificity, and MCC = Matthews Correlation Coefficient.

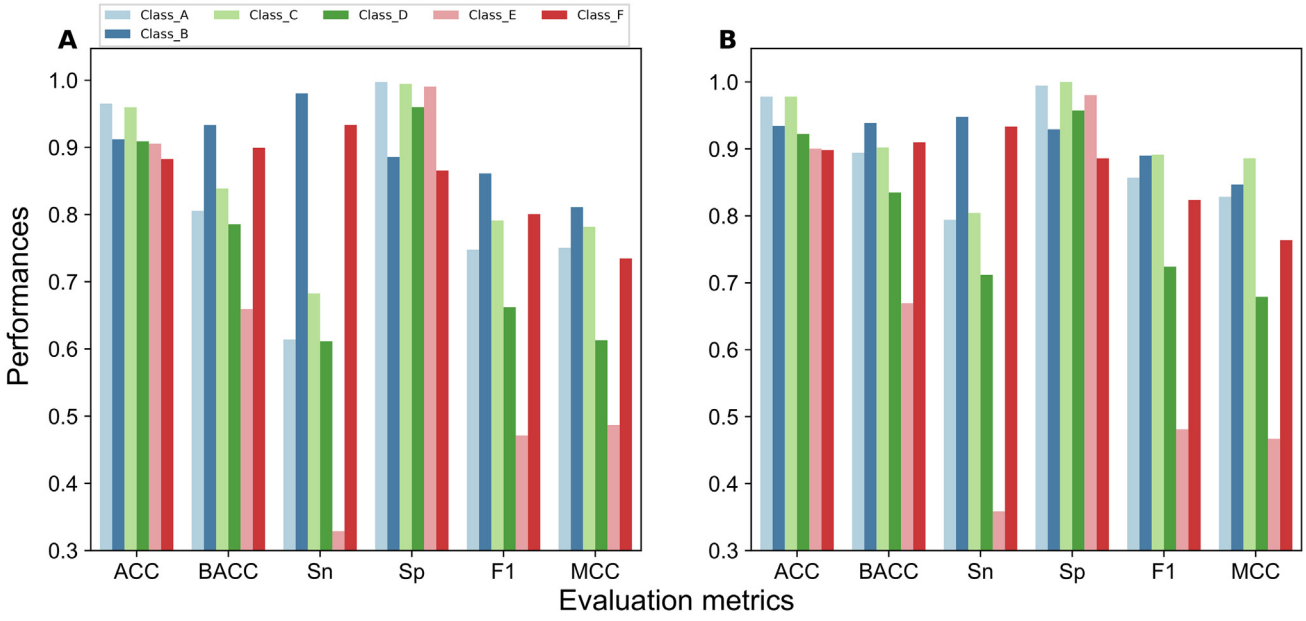

Fig. 3. The performance of each class prediction by DPC_QSO model using the Layer 2 imbalanced dataset. (A) Cross-validation results using the training dataset. (B) Independent dataset performance.

(AAC_DPC, CTriad_DPC_QSO, and DPC_QSO) achieved MCC in the range of ~ 0.85 on the training dataset and ~ 0.80 on the independent assessment, which is significantly better than that of the other 28 models. Furthermore, among these three hybrid features, we have selected CTriad_DPC_QSO, which has the best MCC of 0.860 and 0.798, respectively, during cross-validation and independent validation. To demonstrate the superiority of the SMOTE algorithm, we compared the CTriad_DPC_QSO model performance with the

Table 2
Performance comparison of different feature descriptors on Layer 2 SMOTE training and independent validation datasets.

	Training					Validation				
Features	ACC	BACC	Sn	Sp	MCC	ACC	BACC	Sn	Sp	MCC
CTriad_DPC_QSO	0.961	0.930	0.883	0.977	0.860	0.948	0.900	0.832	0.968	0.798
AAC_CTriad_DPC_QSO	0.960	0.928	0.881	0.976	0.857	0.946	0.893	0.820	0.967	0.784
AAC_CTriad_DPC	0.959	0.926	0.876	0.975	0.852	0.941	0.883	0.803	0.964	0.767
DPC_QSO	0.959	0.927	0.878	0.976	0.854	0.947	0.898	0.829	0.968	0.796
AAC_DPC_QSO	0.958	0.925	0.875	0.975	0.849	0.943	0.888	0.812	0.965	0.778
AAC_DPC	0.956	0.922	0.869	0.974	0.844	0.946	0.897	0.827	0.967	0.793
CTD_CTriad_DPC_QSO	0.955	0.919	0.864	0.973	0.838	0.936	0.876	0.792	0.961	0.752
CTriad_DPC	0.955	0.918	0.864	0.973	0.837	0.944	0.887	0.809	0.965	0.788
DPC	0.954	0.917	0.861	0.972	0.834	0.937	0.873	0.785	0.961	0.754
AAC_CTD_CTriad_DPC_QSO	0.953	0.916	0.860	0.972	0.833	0.936	0.876	0.792	0.961	0.754
AAC_CTriad_QSO	0.953	0.915	0.859	0.972	0.831	0.939	0.880	0.796	0.963	0.760
CTD_DPC_QSO	0.952	0.913	0.855	0.971	0.826	0.938	0.883	0.804	0.962	0.761
CTriad_QSO	0.952	0.913	0.855	0.971	0.825	0.936	0.874	0.788	0.961	0.752
AAC_CTD_CTriad_DPC	0.951	0.911	0.852	0.970	0.823	0.938	0.878	0.793	0.962	0.757
AAC_CTD_CTriad_QSO	0.950	0.909	0.849	0.970	0.819	0.934	0.874	0.788	0.960	0.748
AAC_CTD_DPC_QSO	0.950	0.911	0.851	0.970	0.821	0.938	0.882	0.801	0.963	0.756
CTD_CTriad_DPC	0.949	0.908	0.846	0.969	0.816	0.934	0.873	0.786	0.960	0.746
CTD_CTriad_QSO	0.948	0.906	0.844	0.969	0.813	0.926	0.855	0.755	0.955	0.711
QSO	0.947	0.904	0.840	0.968	0.807	0.922	0.844	0.734	0.953	0.680
AAC_QSO	0.947	0.905	0.842	0.968	0.810	0.920	0.842	0.732	0.952	0.673
CTD_DPC	0.947	0.905	0.842	0.968	0.810	0.932	0.868	0.776	0.959	0.734
AAC_CTD_DPC	0.946	0.903	0.839	0.968	0.806	0.930	0.866	0.773	0.958	0.730
AAC_CTriad	0.944	0.900	0.833	0.967	0.800	0.924	0.847	0.740	0.953	0.697
CTD_QSO	0.943	0.898	0.830	0.966	0.796	0.923	0.852	0.750	0.954	0.699
AAC_CTD_QSO	0.942	0.896	0.827	0.965	0.792	0.927	0.861	0.766	0.956	0.716
AAC_CTD_CTriad	0.941	0.894	0.824	0.965	0.790	0.926	0.856	0.757	0.955	0.715
CTD_CTriad	0.940	0.892	0.820	0.964	0.784	0.921	0.847	0.742	0.952	0.694
CTriad	0.935	0.883	0.804	0.961	0.766	0.915	0.830	0.713	0.947	0.669
AAC_CTD	0.931	0.876	0.793	0.959	0.752	0.911	0.823	0.698	0.947	0.639
CTD	0.924	0.863	0.772	0.954	0.726	0.905	0.810	0.676	0.943	0.613
AAC	0.915	0.847	0.745	0.949	0.691	0.892	0.783	0.630	0.936	0.555

Feature descriptors are listed in the first column. Columns 2–6 represent the ACC, BACC, Sn, Sp, and ACC obtained from the training dataset. Columns 7–11 list a metric corresponding to an independent dataset. The table is sorted based on the training data MCC scores. ACC = Accuracy, BACC = Balanced Accuracy, Sn = Sensitivity, Sp = Specificity, and MCC = Matthews Correlation Coefficient.

DPC_QSO model based on the imbalanced dataset. Fig. 4 indicates that the balanced sample generated by the SMOTE algorithm consistently improved the performance in all five metrics, not only in the training dataset but also in the independent dataset.

Next, we compared the class-based performance between the best models derived from balanced and imbalanced datasets (Fig. 5). Cross-validation analysis, showed that sortase classes A, B, C, D, E, and F improved by 19.629, 10.582, 15.696, 13.933, 29.94, and 8.42%, respectively. However, in the case of independent validation, A and D exhibit similar performance. The remaining classes B, C, E, and F improved by 5.113, 5.166, 13.714, and 5.289%, respectively. As the SMOTE-based CTriad_DPC_QSO model achieved superior performance in identifying sortase classes, we chose it for layer 2 prediction. Usually, the developed predictor is

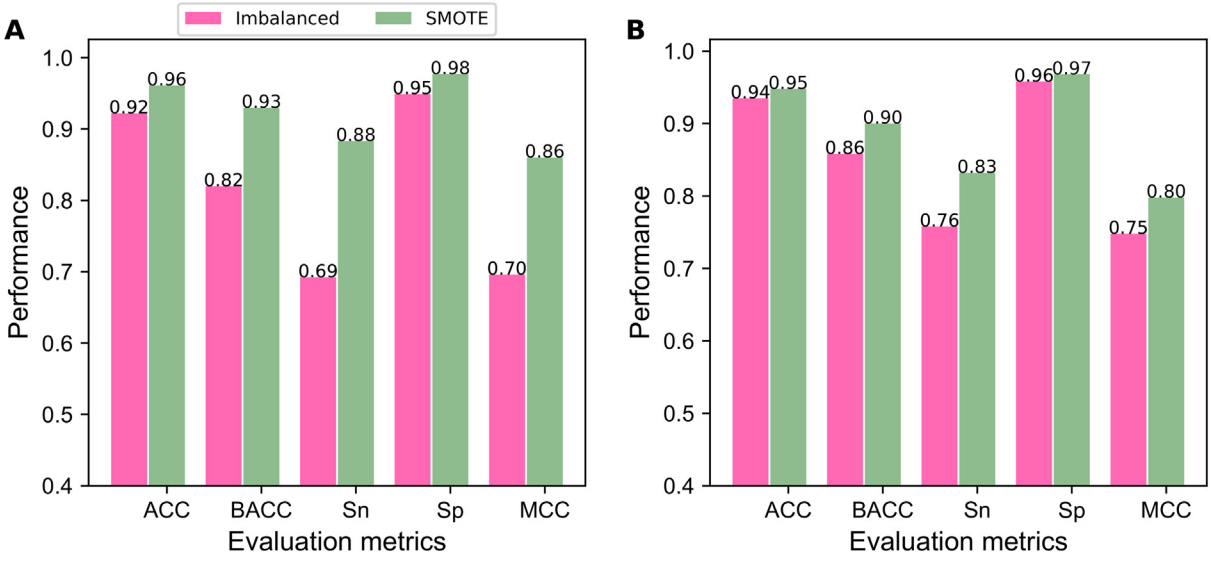

Fig. 4. Performance comparison between imbalanced and balanced Layer 2 datasets. (A) cross-validation performance (B) Independent test performance. Note that SMOTE was used to convert an imbalanced dataset to a balanced one. The DPC_QSO and CTriad_DPC_QSO descriptors achieved the highest performance on imbalanced and balanced datasets, respectively.

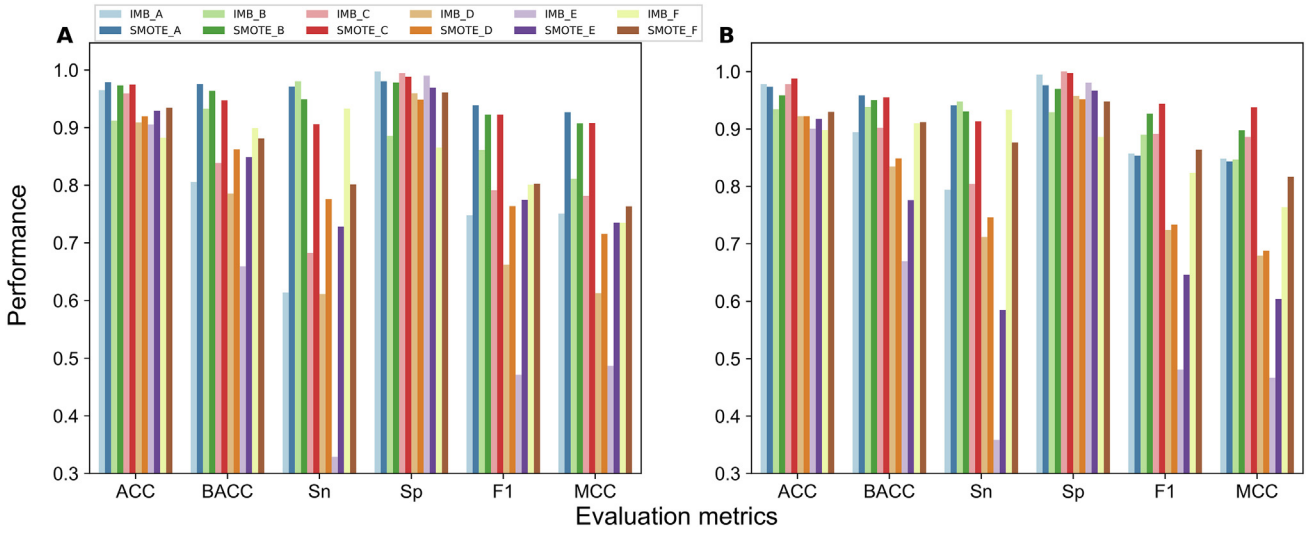

Fig. 5. Performance comparison of each class between SMOTE and imbalanced dataset. (A) cross-validation performance based on training dataset. (B) Independent performance. Performance of DPC_QSO model based on imbalanced dataset represents IMB, while performance of CTriad_DPC_QSO model based on SMOTE represents the same.

compared with the existing predictors to demonstrate the advantages of the proposed approach. However, given that this is the first proposed predictor, we must exclude a comparison.

3.6. Performance comparison of RF with different classifier on both layers

To demonstrate the superiority of the RF algorithm, we employed three different commonly used classifiers, namely Support Vector Machines (SVM), Naive Bayes (NB), and K Nearest Neighbors (KNN), whose optimal models for 31 different descriptors independently were developed for both layers using the same training datasets and 10-fold CV. Tables S3 and S4 provide a performance comparison of RF and other classifiers on training datasets for layers 1 and layer 2. Based on ACC, the RF consistently outperforms the other classifiers regardless of the encodings on both layers, suggesting that RF is the most suitable classification algorithm for discriminating between sortase and non-sortase, and their classes. Thus, we chose RF as the final classifier. In the future, when large-scale training datasets become available, additional algorithms can be applied to determine if they improve the performance.

3.7. Case studies

We examined the performance of SortPred on a variety of datasets in order to demonstrate the potential applications of this method. The performance results are presented below according to the dataset.

1. We then used another independent dataset, which consisted of non-redundant sequences of 736 (including 10 proteobacterial sortases) sortase sequences not included in the training dataset. Sortases in this dataset were either not assigned to any specific class or represent very few proteobacterial sortases. SortPred was able to correctly predict 547 (74.32%) of the 736 sequences as sortases with an average probability score of 0.703 (±0.10). The majority of these predicted sortases were assigned to class D (228), followed by F (111), E (75), B (69), A (49), and C (14) classes. Additionally, SortPred successfully predicted 8/10 of the 10 marine sortases (proteobacterial sortases) that were not part of either the training or validation sets (Table 3).
2. As an additional evaluation of SortPred's ability to predict various sortase enzymes from well-known bacteria for which the genome data are available, we retrieved the proteomes of eight different bacterial strains. Specifically, there were two proteomes each from *Corynebacterium diphtheriae* and *Streptococcus pneumoniae* strains followed by one proteome each from *Staphylococcus aureus, Streptomyces coelicolor, Syntrophothermus lipocalidus,* and *Lactobacillus plantarum,* respectively. Some of these organisms are model organisms and the experimental characterization of sortases in their genomes have been established [46,47]. As described in the methods section, we first excluded the sequences that did not meet the selection criteria.

Table 3
Prediction results for 10 Gram negative proteobacterial sortase sequences. SortPred correctly identified 8/10 sortases and attempted to assign class (A-F) to each of the input sequences with probability scores. X indicates that the given sequence was not identified as a sortase.

ID	Organism	A	B	C	D	E	F	Predicted Sortase Class
CAI85716.1	*Pseudoalteromonas translucida*	0.152	0.144	0.086	0.372	0.09	0.156	D
ABO23660.1	*Shewanella loihica* PV-4	0.142	0.15	0.184	0.248	0.156	0.12	D
KKU10892.1	Parcubacteria group bacterium GW2011_GWF1_45_5	0.2	0.152	0.148	0.224	0.168	0.108	D
KKZ86298.1	*Rhizobium phaseoli* Ch24-10	X	X	X	X	X	X	X
OEE61991.1	*Enterovibrio norvegicus*	0.168	0.18	0.094	0.304	0.15	0.104	D
WP_083763095.1	*Saccharophagus degradans*	0.124	0.076	0.162	0.228	0.188	0.222	D
ARU28296.1	*Cellvibrio* sp. PSBB006	0.06	0.1	0.168	0.19	0.304	0.178	E
OUT41711.1	*Micavibrio* sp. TMED2	0.064	0.094	0.102	0.26	0.264	0.216	E
OYX47620.1	Alphaproteobacteria bacterium 32–64-14	0.04	0.076	0.072	0.178	0.326	0.308	E
PVV08381.1	Gamma proteobacterium symbiont of *Ctena orbiculata*	X	X	X	X	X	X	X

Then, RPS-BLAST was used to annotate the remaining sequences from each genome, which assigned 26 sequences as sortase enzymes. One sequence from *Corynebacterium diphtheriae* strain ATCC 700971/NCTC 13129/Biotype gravis (UniProt ID: Q6NG63) was not assigned a class. SortPred also successfully identified and assigned classes to each of these sequences. Moreover, SortPred predicted the above sequence (UniProt ID: Q6NG63) as a class F sortase (Table S5). It is important to note that only four of the 26 sequences are highly similar to the training dataset with sequence similarity greater than 70%, while the remaining 22 sequences have sequence identities ranging from 39 to 67%. Overall, SortPred performed well when applied to low sequence similarity sequences, indicating that the method can identify putative sortases when applied to different bacterial genomes. Among these 26 sequences, six have been experimentally characterized and have their three-dimensional structures already available in Protein Data Bank.

3. We created an additional non-redundant independent dataset consisting of 464 sortase sequences that were submitted to the NCBI protein (https://www.ncbi.nlm.nih.gov/protein) database between June 2021 and October 2021. According to RPS-BLAST analysis, the majority of these sequences belongs to class F (1 0 1) sortases, followed by B (97), C (69), D (55), A (51), and E (41) sortases. Also, no specific class was assigned to 47 sequences, whereas three sequences were classified as marine sortases (proteobacterial). On testing these annotated sequences using SortPred, we observed that SortPred correctly identified 437 of the 464 sortase sequences. Moreover, 365/464 (78.66%) annotations were identical between the RPS-BLAST annotations and SortPred predictions. Discrepancies were found between only 72 (15.15%) sequences, including 40 sequences for which RPS-BLAST was unable to determine a class. SortPred, on the other hand, predicted and attempted to classify each of these sequences, including those associating with proteobacterial (assigned to class D) origin. Generally, SortPred classified proteobacterial sortases as class D enzymes (Table S6).

In summary, the results of our study suggest that our proposed approach (SortPred) using sequence derived features may yield an effective method for predicting bacterial sortases, especially for the newly released sequences, and demonstrate that our method can also be successfully applied to identify sortase sequences from gram-negative bacteria (proteobacteria).

4. Conclusions

In recent years, a great deal of success has been achieved with ML models in learning complex patterns that enable them to predict the data that has not yet been seen [48]. ML algorithms parse the known data and learn from it and make predictions regarding any new datasets [49,50]. An early application of ML algorithms in protein science was reported about two decades ago, where a logic based approach was used to predict the secondary structure of the proteins [51]. Since then, various aspects of protein science have been addressed with the aid of ML methods [52–54]. Considering the power of ML to deal with a wide variety of features simultaneously, as well as its ability to capture the hidden relationships [55–59], we used one of the common ML algorithms known as RF for the prediction of sortase enzymes. This is the first time a ML-based method has been applied for the prediction of sortase enzymes and their classes.

The two-layer predictor is quite famous in the field of bioinformatics for identifying different information about predicted positive sample [60,61]. This multiple information will help experimentalists while selecting the putative candidates. In this regard, we developed a two-layer novel predictor called SortPred, which allow us to identify the sortase and their classes based on the sequence information. Firstly, we constructed a novel dataset and partitioned it separately for the first and the second layer model development. At the first layer, a balanced dataset and binary classifier are used, while at the second layer, the SMOTE algorithm is used to generate the balanced dataset and multi-label classifier. To develop SortPred, we explored five different feature encoding algorithms and possible combinations, with the corresponding prediction model developed based on RF. Then, we used an independent validation set to assess the robustness of each model. In the end, the final model for the first layer and the second layer was selected based on the robustness. Our prediction model is publicly available at: https://procarb.org/sortpred/. Further improvements to the proposed approach can be achieved by exploring other ML algorithms such as decision tree-based [62], neural network-based algorithms [63–65], incorporating novel features and classical computational approaches used in other studies. Furthermore, we expect that our work will spark interest in predicting sortase enzymes using ML methods, and the performance will improve even further as more balanced data becomes available.

Funding

This work was fully supported by the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) (2021R1I1A1A01056363 and 2021R1A2C1014338).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2021.12.014.

References

[1] Comfort D, Clubb RT. A comparative genome analysis identifies distinct sorting pathways in gram-positive bacteria. Infect Immun 2004;72(5):2710–22.
[2] Jacobitz AW, Kattke MD, Wereszczynski J, Clubb RT. Sortase transpeptidases: structural biology and catalytic mechanism. Adv Protein Chem Struct Biol 2017;109:223–64.
[3] Hendrickx APA, Budzik JM, Oh S-Y, Schneewind O. Architects at the bacterial surface—sortases and the assembly of pili with isopeptide bonds. Nat Rev Microbiol 2011;9(3):166–76.
[4] Spirig T, Weiner EM, Clubb RT. Sortase enzymes in Gram-positive bacteria. Mol Microbiol 2011;82:1044–59.
[5] Schneewind O, Missiakas DM. Protein secretion and surface display in Gram-positive bacteria. Philos Trans R Soc Lond B Biol Sci 2012;367(1592):1123–39.
[6] Cascioferro S, Totsika M, Schillaci D. Sortase A: an ideal target for anti-virulence drug development. Microb Pathog 2014;77:105–12.
[7] Suree N, Yi SW, Thieu W, Marohn M, Damoiseaux R, Chan A, et al. Discovery and structure-activity relationship analysis of Staphylococcus aureus sortase A inhibitors. Bioorg Med Chem 2009;17(20):7174–85.
[8] Dong J, Zhang L, Xu N, Zhou S, Song Yi, Yang Q, et al. Rutin reduces the pathogenicity of Streptococcus agalactiae to tilapia by inhibiting the activity of sortase A. Aquaculture 2021;530:735743.
[9] Cascioferro S, Raffa D, Maggio B, Raimondi MV, Schillaci D, Daidone G. Sortase A inhibitors: recent advances and future perspectives. J Med Chem 2015;58(23):9108–23.
[10] Ha MW, Yi SW, Paek S-M. Design and synthesis of small molecules as potent staphylococcus aureus sortase a inhibitors. Antibiotics 2020;9(10):706.
[11] Popp MW, Antos JM, Grotenbreg GM, Spooner E, Ploegh HL. Sortagging: a versatile method for protein labeling. Nat Chem Biol 2007;3(11):707–8.

[12] Bradshaw WJ, Davies AH, Chambers CJ, Roberts AK, Shone CC, Acharya KR. Molecular features of the sortase enzyme family. FEBS J 2015;282(11):2097–114.
[13] Perry AM, Ton-That H, Mazmanian SK, Schneewind O. Anchoring of surface proteins to the cell wall of Staphylococcus aureus. III. Lipid II is an in vivo peptidoglycan substrate for sortase-catalyzed surface protein anchoring. J Biol Chem 2002;277(18):16241–8.
[14] Ton-That H, Mazmanian SK, Alksne L, Schneewind O. Anchoring of surface proteins to the cell wall of Staphylococcus aureus. Cysteine 184 and histidine 120 of sortase form a thiolate-imidazolium ion pair for catalysis. J Biol Chem 2002;277(9):7447–52.
[15] Malik A, Kim SB. A comprehensive in silico analysis of sortase superfamily. J Microbiol 2019;57(6):431–43.
[16] Duong A, Capstick DS, Di Berardo C, Findlay KC, Hesketh A, Hong H-J, et al. Aerial development in Streptomyces coelicolor requires sortase activity. Mol Microbiol 2012;83(5):992–1005.
[17] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403–10.
[18] Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. BMC Bioinf 2009;10(1):421.
[19] Potter SC, Luciani A, Eddy SR, Park Y, Lopez R, Finn RD. HMMER web server: 2018 update. *Nucleic Acids Res*. 2018;46:W200-W4.
[20] Marchler-Bauer A, Bo Yu, Han L, He J, Lanczycki CJ, Lu S, et al. CDD/SPARCLE: functional classification of proteins via subfamily domain architectures. Nucleic Acids Res. 2017;45(D1):D200–3.
[21] Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 2006;22(13):1658–9.
[22] Manavalan B, Govindaraj RG, Shin TH, Kim MO, Lee G. iBCE-EL: A new ensemble learning framework for improved linear B-cell epitope prediction. Front Immunol 2018;9:1695.
[23] Manavalan B, Shin TH, Kim MO, Lee G. PIP-EL: a new ensemble learning method for improved proinflammatory peptide predictions. Front Immunol. 2018;9:1783.
[24] Kuhn M. Building predictive models in R using the caret package. J Stat Softw 2008;28:1–26.
[25] Boopathi V, Subramaniyam S, Malik A, Lee G, Manavalan B, Yang D-C. mACPpred: a support vector machine-based meta-predictor for identification of anticancer peptides. Int J Mol Sci 2019;20(8).
[26] Charoenkwan P, Chiangjong W, Nantasenamat C, Hasan MM, Manavalan B, Shoombuatong W. StackIL6: a stacking ensemble model for improving the prediction of IL-6 inducing peptides. *Brief Bioinform*. 2021.
[27] Hasan MM, Alam MA, Shoombuatong W, Deng HW, Manavalan B, Kurata H. NeuroPred-FRL: an interpretable prediction model for identifying neuropeptide using feature representation learning. *Brief Bioinform*. 2021.
[28] Dubchak I, Muchnik I, Holbrook SR, Kim SH. Prediction of protein folding class using global description of amino acid sequence. Proc Natl Acad Sci 1995;92(19):8700–4.
[29] Dubchak I, Muchnik I, Mayor C, Dralyuk I, Kim SH. Recognition of a protein fold in the context of the Structural Classification of Proteins (SCOP) classification. Proteins 1999;35:401–7.
[30] Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, et al. Predicting protein-protein interactions based only on sequences information. Proc Natl Acad Sci USA 2007;104(11):4337–41.
[31] Chou K-C. Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. Biochem Biophys Res Commun 2000;278(2):477–83.
[32] Wang J, Li J, Yang B, Xie R, Marquez-Lago TT, Leier A, et al. Bastion3: a two-layer ensemble predictor of type III secreted effectors. *Bioinformatics*. 2019;35:2017-28.
[33] Liu M-L, Su W, Wang J-S, Yang Y-H, Yang H, Lin H. Predicting preference of transcription factors for methylated DNA using sequence information. Mol Ther-Nucl Acids 2020;22:1043–50.
[34] Zhang D, Xu ZC, Su W, Yang YH, Lv H, Yang H, et al. iCarPS: a computational tool for identifying protein carbonylation sites by novel encoded features. *Bioinformatics*. 2021;37:171-7.
[35] Dao FY, Lv H, Zulfiqar H, Yang H, Su W, Gao H, et al. A computational platform to identify origins of replication sites in eukaryotes. *Brief Bioinform*. 2021;22:1940-50.
[36] Lv H, Dao FY, Guan ZX, Yang H, Li YW, Lin H. Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method. *Brief Bioinform*. 2021;22.
[37] Wang D, Zhang Z, Jiang Y, Mao Z, Wang D, Lin H, et al. DM3Loc: multi-label mRNA subcellular localization prediction and analysis based on multi-head self-attention mechanism. *Nucleic Acids Res*. 2021;49:e46.
[38] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artificial Intelligence Res 2002;16:321–57.
[39] Taherzadeh G, Zhou Y, Liew A-C, Yang Y. Sequence-based prediction of protein–carbohydrate binding sites using support vector machines. J Chem Inf Model 2016;56(10):2115–22.
[40] Jing X-Y, Li F-M. Predicting Cell Wall Lytic Enzymes Using Combined Features. Front Bioeng Biotechnol 2020;8.
[41] Pan X, Chen L, Liu I, Niu Z, Huang T, Cai YD. Identifying protein subcellular locations with embeddings-based node2loc. IEEE/ACM Trans Comput Biol Bioinform 2021.
[42] Hasan MM, Shoombuatong W, Kurata H, Manavalan B. Critical evaluation of web-based DNA N6-methyladenine site prediction tools. *Brief Funct Genomics*. 2021;20:258-72.
[43] Su R, Hu J, Zou Q, Manavalan B, Wei L. Empirical comparison and analysis of web-based cell-penetrating peptide prediction tools. *Brief Bioinform*. 2020;21:408-20.
[44] Manavalan B, Hasan MM, Basith S, Gosu V, Shin T-H, Lee G. Empirical comparison and analysis of web-based DNA N4-methylcytosine site prediction tools. Mol Thera-Nucl Acids. 2020;22:406–20.
[45] Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 2020;21:6.
[46] Mazmanian SK, Liu G, Ton-That H, Schneewind O. Staphylococcus aureus sortase, an enzyme that anchors surface proteins to the cell wall. Science 1999;285(5428):760–3.
[47] Kattke MD, Chan AH, Duong A, Sexton DL, Sawaya MR, Cascio D, et al. Crystal structure of the Streptomyces coelicolor sortase E1 transpeptidase provides insight into the binding mode of the novel class E sorting signal. PLoS ONE 2016;11(12):e0167763.
[48] Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B. Definitions, methods, and applications in interpretable machine learning. Proc Natl Acad Sci U S A. 2019;116(44):22071–80.
[49] Vamathevan J, Clark D, Czodrowski P, Dunham I, Ferran E, Lee G, et al. Applications of machine learning in drug discovery and development. Nat Rev Drug Discov. 2019;18(6):463–77.
[50] Basith S, Manavalan B, Hwan Shin T, Lee G. Machine intelligence in peptide therapeutics: A next-generation tool for rapid disease screening. Med Res Rev. 2020;40(4):1276–314.
[51] Muggleton S, King RD, Stenberg MJE. Protein secondary structure prediction using logic-based machine learning. Protein Eng. 1992;5(7):647–57.
[52] Malik A, Ahmad S. Sequence and structural features of carbohydrate binding in proteins and assessment of predictability using a neural network. BMC Struct Biol. 2007;7:1.
[53] Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee GR, et al. Accurate prediction of protein structures and interactions using a three-track neural network. Science 2021;373(6557):871–6.
[54] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature 2021;596(7873):583–9.
[55] Wei L, He W, Malik A, Su R, Cui L, Manavalan B. Computational prediction and interpretation of cell-specific replication origin sites from multiple eukaryotes by exploiting stacking framework. *Brief Bioinform*. 2021;22.
[56] Hasan MM, Manavalan B, Shoombuatong W, Khatun MS, Kurata H. i6mA-Fuse: improved and robust prediction of DNA 6 mA sites in the Rosaceae genome by fusing multiple feature representation. Plant Mol Biol 2020;103:225–34.
[57] Hasan MM, Manavalan B, Shoombuatong W, Khatun MS, Kurata H. i4mC-Mouse: improved identification of DNA N4-methylcytosine sites in the mouse genome using multiple encoding schemes. Comput Struct Biotechnol J 2020;18:906–12.
[58] Hasan MM, Manavalan B, Khatun MS, Kurata H. i4mC-ROSE, a bioinformatics tool for the identification of DNA N4-methylcytosine sites in the Rosaceae genome. Int J Biol Macromol 2020;157:752–8.
[59] Govindaraj RG, Subramaniyam S, Manavalan B. Extremely-randomized-tree-based Prediction of N(6)-Methyladenosine Sites in Saccharomyces cerevisiae. Curr Genomics. 2020;21(1):26–33.
[60] Liu B, Fang L, Long R, Lan X, Chou K-C. iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. Bioinformatics 2016;32(3):362–9.
[61] Manavalan B, Subramaniyam S, Shin TH, Kim MO, Lee G. Machine-learning-based prediction of cell-penetrating peptides and their uptake efficiency with improved accuracy. J Proteome Res 2018;17(8):2715–26.
[62] Firoz A, Malik A, Joplin KH, Ahmad Z, Jha V, Ahmad S. Residue propensities, discrimination and binding site prediction of adenine and guanine phosphates. BMC Biochem. 2011;12:20.
[63] Nguyen QH, Nguyen-Vo TH, Le NQK, Do TTT, Rahardja S, Nguyen BP. iEnhancer-ECNN: identifying enhancers and their strength using ensembles of convolutional neural networks. BMC Genomics 2019;20:951.
[64] Ho QT, Le NQK, Ou YY. mCNN-ETC: identifying electron transporters and their functional families by using multiple windows scanning techniques in convolutional neural networks with evolutionary information of protein sequences. Brief Bioinform 2021.
[65] Le NQK, Ho QT, Nguyen TT, Ou YY. A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information. *Brief Bioinform*. 2021;22.