StackCBPred: A Stacking based Prediction of Protein-Carbohydrate Binding Sites from Sequence

Suraj Gattani[1],[§], Avdesh Mishra[1],[§] and Md Tamjidul Hoque[1],[*]

[1]Department of Computer Science, University of New Orleans, New Orleans, LA, 70148, USA.

[§] These authors contributed equally to this work as first authors.

*To whom correspondence should be addressed, thoque@uno.edu

Abstract

Motivation: Carbohydrate-binding proteins play vital roles in many biological processes such as intercellular signaling, protein folding, subcellular localization, ligand recognition and developmental process in connections to drug targeting and gene expression. Analysis and study of these proteins are useful in treating many diseases including cancer. To understand the nature of protein-carbohydrate interactions, examination at the individual residue level, is necessary. Analyzing the local sequential environments of the binding and non-binding regions to predict the protein-carbohydrate binding sites is one of the challenging problems in molecular and computational biology. Existing experimental methods for identifying protein-carbohydrate binding sites are laborious and expensive. Thus, prediction of such binding sites, directly from sequences, using computational methods, can be useful to quickly annotate the binding sites and guide the experimental process. Because the number of carbohydrate-binding residues is significantly lower than the number of non-carbohydrate-binding residues, most of the methods developed for the prediction of protein-carbohydrate binding sites are biased towards over predicting the negative class (or non-carbohydrate-binding). Here, we propose a balanced predictor, called StackCBPred, which utilizes features, extracted from evolution-driven sequence profile, called the position-specific scoring matrix (PSSM) and several predicted structural properties of amino acids to effectively train a stacking-based machine learning method for the accurate prediction of protein-carbohydrate binding sites.

Results: Based on the balanced benchmark dataset (1028 binding and equal number of non-binding residues) created from 100 protein sequences, the StackCBPred achieved 10-fold cross-validation accuracy (ACC) of 86%, sensitivity (SENS) of 86.09%, specificity (SPEC) of 85.89% and Matthew's correlation coefficient (MCC) of 0.7198. Moreover, when tested on the existing as well as new independent test datasets designed in this study, StackCBPred provide expected accuracy. This indicates that the proposed approach outperforms existing protein-carbohydrate binding prediction approach and can be used for effective prediction of protein-carbohydrate binding sites from sequence alone.

Availability and implementation: The code-data is available at
http://cs.uno.edu/~tamjid/Software/StackCBPred/code_data.zip

1. Introduction

Organisms need four types of molecules: nucleic acids, proteins, carbohydrates (or polysaccharides) and lipids for life, which are usually referred to as the molecules of life. Carbohydrates are often considered as the third important molecule of life, after DNA and proteins [1]. Carbohydrate interacts with many different protein families which include lectins, antibodies, sugar transporters and enzymes [2]. Protein-carbohydrate interactions are responsible for various biological processes, including intercellular signaling, cellular adhesion, cellular recognition, protein folding, subcellular localization, ligand recognition and developmental process [3-5]. In fact, carbohydrate of one or the other type generally covers living cells in all organisms [6]. These carbohydrates play important roles in defense for human cell against pathogens [7]. Moreover, some pathogens such as influenza use these carbohydrates on the outside of the human cell to gain entry [8]. The proteins, which recognize and bind to the cell-surface carbohydrates, are useful as biomarkers or drug targets [9-12]. The study of protein-carbohydrate interactions is usually carried out by experimental techniques including X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy study, molecular modeling, fluorescence spectrometry, and dual polarization interferometry. However, protein-carbohydrate interactions are challenging to study experimentally because of the weak binding affinity and synthetic complexity of individual carbohydrates [6]. Therefore, the prediction of protein-carbohydrate interactions through a computational approach becomes essential. This motivated us to develop an effective computational predictor for effective identification and characterization of protein-carbohydrate binding sites.

Study of protein-carbohydrate interactions using computational methods mainly focuses on locating the sites of proteins that bind to carbohydrates. One of the promising computational techniques is docking. Docking methods are often used to predict the orientation of the carbohydrate in the binding site [2]. Docking algorithms such as Autodock [13], GLIDE [14], Dock [15], etc. consider the orientation of the dangling groups, such as, hydroxyl groups and hydrogen bond network that stabilizes the complex, and the conformational behavior of the glycosidic bonds for oligosaccharides [2]. On the other hand, Taroni *et al*. [16] proposed a first bioinformatics approach for predicting protein-carbohydrate binding sites from a known protein structure. In their work, six parameters of amino acids were evaluated, which includes solvation potential, residue propensity, hydrophobicity, planarity, protrusion, and relative accessible surface area. A simple combination of three of the parameters (residue propensity, protrusion and relative accessible surface area) out of six were found to distinguish the observed binding sites with an overall accuracy of 65% for a set of 40 protein-carbohydrate complexes. Likewise, Sujatha and Balaji developed a method called COTRAN for predicting protein-galactose binding sites [17]. They utilized the combination of geometrical and structural characteristics such as solvent accessibility and secondary structure type that allowed proper detection of potential galactose-binding sites. Kulharia *et al*. developed a program called *InCa-SiteFinder* for predicting non-covalent inositol and carbohydrate binding sites on the surface of the protein structure [18].

They employed van der Waals interaction energy between protein and a probe and amino acid propensities as the parameters to locate and predict carbohydrate-binding sites. A continuous surface pocket interacting with protein-probes was considered binding sites. Nassif *et al*. proposed a glucose-binding site classifier which considers the sugar-binding pocket as a spherical spatio-chemical environment and represents it as a vector of geometric and chemical features which includes charges, hydrophobicity, hydrogen bonding and more [19]. They employed random decision forest for feature selection and used selected geometric and chemical features to train the support vector machines (SVM) for predicting protein-glucose binding sites. Tsai *et al*. predicted binding sites by employing three-dimensional probability density distribution of interacting atoms in protein surfaces as input to the neural networks and SVM [20]. In the recent past, an energy-based approach for the identification and analysis of binding sites residues in protein-carbohydrate complexes has been proposed [21]. Through this study, it was found that 3.3% of residues are identified as binding sites in protein-carbohydrate complexes whereas the binding site residues in protein-protein, protein-RNA, and protein-DNA complexes are 10.8%, 7.6%, and 8.7% respectively. Furthermore, the binding propensity analysis performed in this study indicates the dominance of Tryptophan (TRP) amino acid to interact with the carbohydrates through aromatic-aromatic interactions. More recently, Shanmugam *et al.* proposed a method to identify and analyze the residues, which are involved in both the folding and binding of protein-carbohydrate complexes [22]. Stabilizing residues were identified by using the knowledge of hydrophobicity, long-range interactions, and conservations, as well as binding site residues, were identified using a distance cutoff of 3.5Å between heavy atoms in protein and ligand. Residues, which were common in stabilizing and binding, were termed as key residues. Some of the interesting findings of the work indicate that most of the key residues are present in β-strands and polar and charged residues have a high tendency to serve as key residues.

The structure-based methods discussed above, rely on protein structures that are often not available, which makes the sequence-based method inevitable. The first sequence-based method for protein-carbohydrate binding sites prediction was developed by Malik and Ahmad in 2007 [8]. In their work, Malik and Ahmad used only the evolutionary attributes called PSSM as input to the neural network to create a predictive model. Their method achieved the average of 87% sensitivity and 23% specificity while tested by leave-one-out technique on a dataset of 40 protein-carbohydrate complexes. After a year less than a decade, Taherzadeh *et al*. proposed a method, called SPRINT-CBH, which used PSSM profiles with additional information on sequence and predicted solvent accessible surface area as features to develop an SVM based predictor in 2016 [6]. As reported, SPRINT-CBH achieved the average of 18.8% sensitivity and 99.6% specificity while tested using 10-fold cross-validation (CV) on a dataset of 102 protein-carbohydrate complexes and 22.3% sensitivity and 98.8% specificity while tested using independent test set of 50 protein-carbohydrate complexes. Both aforementioned methods suffer from the problem of imbalanced prediction accuracies. These methods either yield a high sensitivity and low specificity or vice versa. Thus, both the proposed methods are limited in their ability to effectively predict binding sites and explain how protein-carbohydrate interaction occurs. Therefore, it

becomes essential to identify new features and effective machine learning techniques that can help in improved binding site prediction as well as help interpret protein-carbohydrate interactions.

While there are still very few methods for predicting protein-carbohydrate binding sites, many other methods have been established for binding site and binding proteins prediction in the area of protein-protein [23-26], protein-peptide [27-30], protein-DNA [31-34], protein-RNA [35-38] and protein-ligand [39-42] interactions. Several of the aforementioned sequence-based methods have shown that the use of evolution-derived and predicted sequence and structure-based features can significantly improve the overall performance of the binding sites prediction.

In this study, we investigated different descriptors, which include information extracted from the evolutionary profile as well as predicted sequence and structural properties. Moreover, we examined various machine learning approaches to develop sequence-based unbiased and balanced predictor of non-covalent protein-carbohydrate binding sites. Useful feature groups were selected to build a stacking-based classifier called StackCBPred. The StackCBPred was trained and cross-validated by 100 carbohydrate-binding proteins and independently tested by two different test sets containing 50 and 88 proteins with known high-resolution protein-carbohydrate complex structures, respectively. As the dataset contain significantly more non-binding residues than binding residues, StackCBPred was trained with a balanced dataset obtained by employing the undersampling technique with an aim to design a more balanced predictor. The development of StackCBPred offered a significant improvement in sensitivity and balanced accuracy based on the benchmark and independent test data when compared to the existing sequence-based binding predictor. We believe that the superior performance of StackCBPred will motivate the researchers to use this method to identify protein-carbohydrate binding sites directly from sequence and utilize the outcomes for drug targeting. In addition, the stacking-based machine learning technique and features proposed in this work could be applied to solve various other biologically important problems.

2. Methods

In this section, we describe the approach taken to prepare benchmark and independent test data sets, feature extraction, feature selection, performance evaluation, and machine learning framework development.

2.1 Dataset

We collected the benchmark dataset [43] that contains a total of 102 high-resolution carbohydrate-binding protein sequences. However, in our implementation, we only used 100 high-resolution carbohydrate-binding protein sequences for training and cross-validation as two

of the sequences contain non-standard amino acid and the physicochemical properties of the non-standard amino acids could not be obtained. Furthermore, we collected an independent test dataset [43] to compare the performance of StackCBPred with the existing predictor. This dataset consists of 50 high-resolution carbohydrate-binding protein sequences of which only 49 were used in our implementation. We found that 1 out of 50 proteins contain non-standard amino acid which, we discarded because the physicochemical properties of the nonstandard amino acids could not be obtained. Here and after we represent this test dataset as TS49.

From the benchmark dataset of 100 sequences, we obtained a total of 26,986 residues, of which, 1028 residues are binding and the rest are non-binding. To avoid bias caused by a large number of non-binding residues, a balanced dataset was prepared following an undersampling approach [45] by randomly selecting a number of non-binding residues equal to the number of binding residues. This resulted in a benchmark dataset, which consists of 1028 binding and an equal number of non-binding residues. From TS49 sequences, we obtained a total of 13,738 residues of which 508 residues are binding and the rest are non-binding. Using similar undersampling approach as above, a balanced independent test set was prepared which consist of 508 binding and an equal number of non-binding residues.

To further test the performance of our predictor, we collected an additional dataset PROCARB604 from PROCARB [46] database. The proteins whose ID's matched to the protein ID's that were present in either the benchmark or the independent test dataset mentioned above, were removed from this new dataset. Next, the redundant proteins with a sequence identity cutoff of ≥ 30% according to BLAST-CLUST [47] were removed. Then, the sequences for which the features could not be extracted using DisPredict2 program were discarded. Finally, the dataset, which consists of 88 protein-carbohydrate complexes was obtained. Here and after we represent this dataset as TS88. This new TS88 dataset consists of 688 binding residues. Using an undersampling approach, we prepared a balanced dataset which contains 688 binding and an equal number of non-binding residues.

2.2 Feature Extraction

We collected various useful features which include information extracted from evolutionary profile as well as predicted sequence and structural properties of proteins, which we described in this section.

2.2.1 Position Specific Scoring Matrix (PSSM) and Monogram (MG)

PSSM captures the evolution derived information in proteins. Evolutionary information is very impactful for protein function annotation in biological analysis and is widely used in many studies [27, 34, 44, 48-51]. Furthermore, evolutionarily conserved residues are found to play crucial functional roles such as binding [52]. For this study, we obtained the normalized PSSM values for every residue in protein sequence from DisPredict2 [44] program. DisPredict2 internally executes

three iterations of PSI-BLAST [53] against NCBI's non-redundant database to generate a PSSM profile and subsequently converts it to normalized PSSM by dividing each value by a value of 9. PSSM is a matrix of L×20 dimensions, where L is the length of the protein. The rows in PSSM represent the position of amino acid in the sequence and the columns represent the 20 standard amino acid types. Hence, every residue in the protein sequence is encoded by a 20-dimensional feature vector. In addition, the PSSM score was further extended to compute monogram feature [54], which is obtained by taking the sum of the scores over the length of the protein for 20 standard amino acid types. This resulted in 1 feature for every amino acid.

2.2.2 Accessible Surface Area (ASA) and Secondary Structure (SS)

ASA and SS are predicted structural features that are found to be highly effective for binding sites prediction. We used DisPredict2 program to obtain predicted ASA and SS probabilities for helix, coil, and beta-sheet at the residue level. DisPredict2 internally uses a program called SPINE-X [55] to predict ASA and SS probabilities directly from the protein sequence.

2.2.3 Half Sphere Exposure (HSE) and Torsion angles

HSE is a measure of protein solvent exposure that was first introduced in [56]. HSE measures how buried amino acid residues are in protein conformation. The calculation of HSE is obtained by dividing a contact number (CN) sphere into two halves by the plane perpendicular to the $C\beta$-$C\alpha$ vector. This simple division of the CN sphere produces two different measures, called HSE-up and HSE-down. In this study, we used these two measures as features which were extracted from the SPIDER3 program [57-59]. Additionally, protein backbone structure can be described by torsion angles Phi (ϕ) and Psi (ψ). This local structure descriptor is important for understanding and predicting protein structure, function, and interactions. In our study, we employed predicted ϕ and ψ angles as features which were also extracted from SPIDER3 program.

2.2.4 Physiochemical Properties

We obtained seven representative physiochemical attributes of the amino acids, which include steric parameters, hydrophobicity, volume, polarizability, isoelectric point, helix probability, and sheet probability [60]. As these features are inherently encoded within DisPredict2, we directly extracted these features from the DisPredict2 [44].

2.2.5 Molecular Recognition Features (MoRFs)

Post-translational modifications (PTMs) can induce disorder-to-order transitions of intrinsically disordered proteins (IDPs). IDPs can transition from disorder to order due to binding to other proteins, nucleic acids, lipids, carbohydrates and other small molecules [61, 62]. MoRFs are key to the biological function of IDPs located within long disordered protein sequences [63]. Thus, to inherently capture functional properties of IDPs which may bind to carbohydrates, we employed a single predicted MoRFs score as a feature in this work. We obtain the MoRFs feature from OPAL [63].

2.3 Performance Evaluation

Performance of the StackCBPred was evaluated by 10-fold CV as well as using the independent test. In 10-fold CV, the dataset is segmented into 10 parts, which are each of about equal size. When a fold is set aside for testing, the other 9 folds are used to train the classifier. This process is repeated until each fold has been set aside once for testing and then the test accuracies of each fold are combined to find the average [64]. On the other hand, to perform the independent test, the classifier is trained with the validation dataset and then tested using the independent test dataset. We used various performance evaluation metrics listed in Table 1 to test the accuracy of our proposed method as well as to compare it with the existing method.

In addition, we used AUC and ROC performance evaluation metrics. AUC is the area under the receiver operating characteristics (ROC) curve and is used to evaluate a predictor to see how well it separates two classes of information, which is, in this case, carbohydrate binding and non-binding residues.

Table 1: Name and definition of the evaluation metric.

Name of Metric	Definition
True Positive (TP)	Correctly predicted carbohydrate-binding residues
True Negative (TN)	Correctly predicted non-carbohydrate-binding residues
False Positive (FP)	Incorrectly predicted carbohydrate-binding residues
False Negative (FN)	Incorrectly predicted non-carbohydrate-binding residues
Recall/Sensitivity/True Positive Rate (TPR)	$\dfrac{TP}{TP+FN}$
Specificity/True Negative Rate (TNR)	$\dfrac{TN}{TN+FP}$
Fall-out Rate/False Positive Rate (FPR)	$\dfrac{FP}{FP+TN}$
Miss Rate/False Negative Rate (FNR)	$\dfrac{FN}{FN+TP}$
Accuracy (ACC)	$\dfrac{TP+TN}{FP+FP+TN+FN}$
Balanced Accuracy (BACC)	$\dfrac{1}{2}\left(\dfrac{TP}{TP+FN}+\dfrac{TN}{TN+FP}\right)$
Precision	$\dfrac{TP}{TP+FP}$
F1 score (Harmonic mean of precision and recall)	$\dfrac{2TP}{2TP+FP+FN}$

| Mathews Correlation Coefficient (MCC) | $$\frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FN) * (TP + FP) * (TN + FP) * (TN + FN)}}$$ |

2.4 Framework of StackCBPred

The idea of stacking based machine learning technique [65] which has recently been successfully applied to solve some interesting bioinformatics problems [27, 34, 66-68] is utilized in this work to develop the StackCBPred predictor for carbohydrate-binding sites prediction. Stacking is an ensemble approach, which obtains the information from multiple models and aggregates them to form a new model. In stacking, the information gained from more than one predictive models minimize the generalization error rate and yields more accurate results.

Stacking framework includes two-stages of learners. The classifiers of the first-stage are called base-classifiers. More than one base-classifiers are employed in the first-stage. Likewise, the classifiers of the second-stage are called meta-classifiers. Using meta-classifier, the prediction probabilities from the base-classifiers are combined to reduce the generalization error. To supply the meta-classifier with significant information on the problem space, the classifiers that are different from one another based on their underlying operating principle are used as the base-classifiers.

To find the base-classifiers and meta-classifiers to use in the first and second-stage of stacking framework, we examined eight different machine learning algorithms: (a) Support Vector Machines (SVM) [69], (b) Gradient Boosting Classifier (GBC) [70], (c) Bagging Classifier (BAG) [71], (d) Extra Tree Classifier (ETC) [72], (e) Random Decision Forest (RDF) [73], (f) K-Nearest Neighbor (KNN) [74], (g) Logistic Regression (LOGREG) [64, 75] and (h) XGBoost (XGB) [76]. The algorithms in addition to their configuration details are briefly discussed below.

i) *SVM:* We employed SVM [69] with the radial basis function (RBF) kernel as one of the classifiers to be used in stacking framework. SVM classifies by maximizing the separating hyperplane between two classes and penalizes the instances on the wrong side of the decision boundary. The performance of SVM with the RBF kernel relies on two parameters C, and γ. The RBF kernel parameter γ and the cost parameter C are optimized to achieve the best 10-fold CV balanced accuracy using a grid search [77] technique. The optimal values of the parameters of the SVM were found to be C = $2^{1.24}$ and γ = $2^{-8.75}$.

ii) *LOGREG:* We implemented LOGREG [64, 75] with *L2* regularization as another classifier to be used in staking framework. It measures the relationship between the dependent categorical variable (in our case: a carbohydrate-binding or non-carbohydrate-binding) and one or more independent variables by generating an estimation probability using logistic regression. The parameter, *C* which controls the regularization strength is optimized to achieve the best 10-fold CV balanced accuracy using grid search [77]. In our implementation, the optimal value of the parameter, C was found to be 2.3784.

iii) *ETC:* We employed extremely randomized tree or ETC [72] as another classifier to be used in stacking framework. ETC fits several randomized decision trees from the data sample and uses averaging to improve the prediction accuracy and control over-fitting. We constructed the ETC model with 1,000 trees and the quality of a split was assessed by the Gini impurity index.

iv) *RDF:* RDF [73] constructs a multitude of decision trees on various sub-samples of the dataset and outputs the mean prediction of the decision trees to improve the predictive accuracy and control over-fitting. In our implementation of the RDF, we used bootstrap samples to construct 1,000 trees in the forest.

v) *KNN:* KNN [74] operates by learning from the *K* number of training samples closest in distance to the target point in the feature space. The classification decision is computed from the majority votes coming from the neighbors. In this work, the value of *K* was set to 9 and all the neighbors were weighted uniformly.

vi) *BAG:* BAG [71] method forms a class of algorithms which builds several instances of a base classifier/estimator on random subsets of the training samples and subsequently aggregates their individual predictions to yield a final prediction. It is useful for reducing variance in the prediction. In our study, BAG classifier was fit on multiple subsets of data with the repetitions using 1,000 decision trees, and the outputs were combined by weighted averaging.

vii) *GBC*: GBC [70] involves three elements: (a) a loss function to be optimized, (b) a weak learner to make predictions and (c) an additive model to add weak learners to minimize the loss function. The objective of GBC is to minimize the loss of the model by adding weak learners in a stage-wise fashion using a procedure similar to gradient descent. The existing weak learners in the model are remained unchanged while adding new weak learner. The output from the new learner is added to the output of the existing sequence of learners in an effort to correct or improve the final output of the model. Here, we used 1,000 bosting stages where a regression tree was fit on the negative gradient of the deviance loss function. The learning rate and the maximum depth of each regression tree were set to 0.1 and 3, respectively.

viii) XGB: As GBC, XGB [76] also follows the principle of gradient boosting. However, XGB uses a more regularized model formalization to control over-fitting, which results in better performance. In addition to the better performance, XGB is designed to provide higher computational speed. In our implementation of the XGB, we used 100 bosting stages with a soft prob learning objective, where the number of classes was set to 2 as we are dealing with a binary classification problem of carbohydrate-binding and non-carbohydrate-binding residues. The values of the additional parameters: learning rate, maximum depth, minimum child weight, and subsample ratio were set to 0.1, 3, 5 and 0.9, respectively.

Algorithms mentioned above are built and optimized using Scikit-learn [78]. To select the algorithms to be used as the base-classifiers for the stacked model, we evaluate four different combinations of base-classifier which are:

a. Model-1: includes SVM, LOGREG, KNN and ETC.
b. Model-2: includes SVM, LOGREG, KNN and RDF.
c. Model-3: includes SVM, LOGREG, KNN and BAG.
d. Model-4: includes GBC, LOGREG and KNN.

Model-1, Model-2, and Model-3 are constructed to include classifiers that are different from each other based on the underlying principles of learning. Here, the tree-based classifiers ETC, RDF and BAG are individually combined with the other three classifiers, SVM, LOGREG and KNN to learn different information from the problem-space. On the other hand, Model-4 is formed by the pair-wise correlation analysis of the residue-wise probabilities given by the individual classifiers. Three of the classifiers, with the least Pearson correlation coefficient, are selected as base-classifiers. For all the above combinations, SVM is used as a meta-classifier. The 10-fold CVs of the above four combinations indicate that the Model-1, when combined with SVM gives the best performance. Therefore, we employ four classifiers SVM, LOGREG, KNN and ETC as base classifiers and SVM as meta-classifier in the StackCBPred framework. In StackCBPred, the binding and non-binding probabilities generated by the four base-classifiers are combined with original 33 features and given as input features to the meta-classifier which eventually predict binding and non-binding residues. Figure 1. Illustrates the StackCBPred prediction framework.
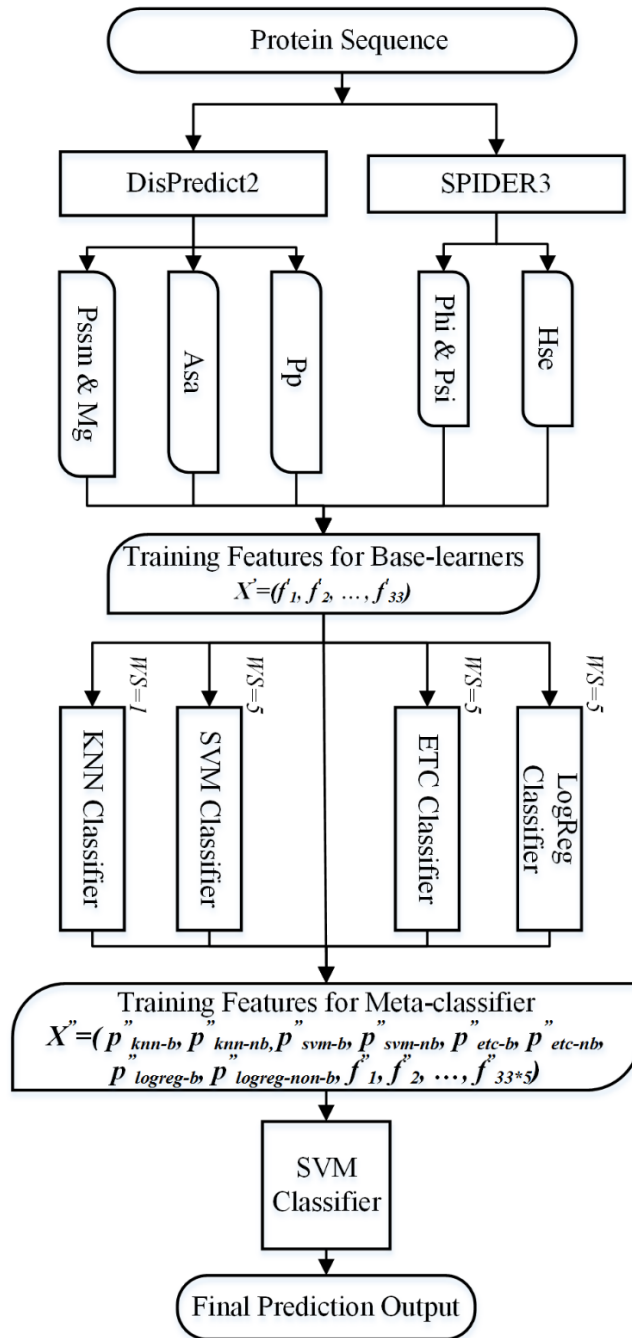
Fig. 1. Illustration of the framework of the final predictor, StackCBPred, which is principally the Model-1 with another version of SVM in the meta layer.

3. Results

In this section, we first demonstrate the results of the feature selection and optimal window selection. Then, we present the performance comparison of potential base-classifiers and stacked models. Finally, we report the performance of StackCBPred on the benchmark dataset

and the independent test datasets and subsequently compare it with the existing method. We would like to emphasize that balanced datasets are utilized for training, cross-validation, and independent test unless otherwise indicated.

3.1 Feature Selection

To identify the features that support the performance of the classifier, we applied a simple incremental feature selection (IFS) approach. IFS begins with the empty feature set and a feature group is added to the feature set if the addition of the feature group improves the performance of the predictor. In case, the accuracy of the predictor is reduced by adding the new feature group, this feature group is discarded, and a new feature group is tested in an iterative fashion. For IFS we used the benchmark dataset to train and TS49 dataset to test the GBC predictor. We initially collected thirty-nine features, of which, we discarded six features based on IFS. The three secondary structure features and three MoRFs features were removed as these features did not help improve the performance of the predictor.

3.2 Window Selection

An optimal size of the sliding window (W) was searched to determine the number of residues around a target residue, which can moderate the interaction between protein and carbohydrate. We designed 8 different models of every machine learning classifiers with 8 different window sizes (1, 3, 5, 7, 9, 11, 13 and 15). Window size for which the classifier yields the highest 10-fold CV balanced accuracy on benchmark dataset was selected as the optimal window size for that classifier. We found that the optimal window size for different classifiers are different. For example, the optimal window size for the SVM was found to be 5 (see Figure 2) whereas, for the KNN it was 1. In this study, the optimal window size for every classifier was separately identified to design an accurate and effective predictor.
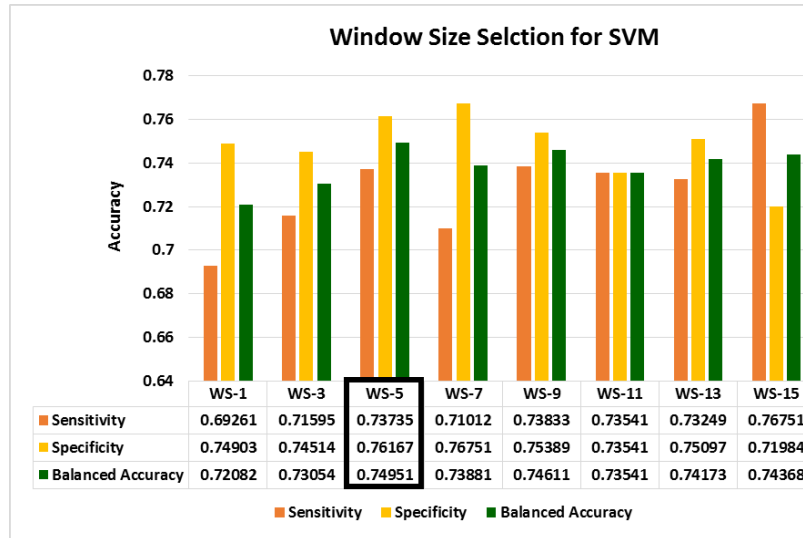
Fig. 2. Performance comparison of SVM based models created from different sliding window sizes. The sensitivity, specificity and balanced accuracy are reported. The optimal size of the window and the corresponding performance scores are marked by a black rectangle. Optimal window size was selected based on the highest 10-fold CV balanced accuracy.

3.3 Selection of the Base and Meta-classifiers

To select the methods for base and meta-classifiers, we examined the performance of eight different machine learning methods, BAG, ETC, LOGREG, KNN, RDF, GBC, XGB and SVM on the benchmark dataset using 10-fold CV. The performance comparison of the classifiers is shown in Table 2.

Table 2. Comparisons of various machine learning algorithms on the benchmark dataset using 10-fold CV.

Metric/Method	BAG	ETC	LOGREG	KNN	RDF	GBC	XGB	SVM
Sensitivity	0.743	0.743	0.718	0.636	0.751	0.733	**0.763**	0.737
Specificity	0.728	0.744	0.679	0.585	0.745	0.715	0.706	**0.762**
Fall Out Rate	0.272	0.256	0.321	0.415	0.255	0.285	0.294	**0.238**
Miss Rate	0.247	0.257	0.282	0.364	0.249	0.268	**0.237**	0.263
Bal. Accuracy	0.740	0.744	0.698	0.610	0.748	0.724	0.734	**0.750**
Accuracy	0.740	0.744	0.698	0.610	0.748	0.724	0.734	**0.750**
Precision	0.734	0.744	0.691	0.605	0.747	0.720	0.722	**0.756**
F1 score	0.744	0.744	0.704	0.620	**0.749**	0.726	0.742	0.746
MCC	0.481	0.487	0.397	0.221	0.496	0.448	0.470	**0.499**

Best score values are bold faced.

Table 2 shows that the optimized SVM with RBF-kernel provides the highest performance in terms of specificity, fall out rate, balanced accuracy, accuracy, precision and MCC, among all the classifiers examined in this application. Moreover, the sensitivity and miss rate is highest for the

XGB and F1 score is highest for RDF. Similarly, it is evident that the performance of tree-based ensemble methods, BAG, ETC, RDF, GBC, and XGB are close to SVM. The balanced accuracy of these tree-based methods differs from each other and SVM only by about 1 to 4%. However, the balanced accuracy of LOGREG and KNN are 7.31% and 22.79% lower than the SVM, respectively. Moreover, the learning principle of LOGREG, KNN, and SVM are different from each other. Following the guidelines of base-classifier selection based on different underlying principles, we initially selected SVM, LOGREG and KNN as three of the base classifiers. Then, we added one tree-based ensemble method out of five methods, BAG, ETC, RDF, GBC, and XGB, at a time as the fourth base-classifier and formulated five different combinations. For all of the combinations, the meta-classifier is SVM. Out of five combinations, we present the performance of the top three combinations namely Model-1, Model-2 and Model-3 in Table 4.

Moreover, we created an additional stacked model following the guidelines of base-classifier selection based on low Pearson correlation coefficient. We computed the Pearson correlation coefficient (ρ) between the two sets of probabilities given by two classifiers using equation (1).

$$\rho = \frac{\sum XY}{\sqrt{\sum X^2 \sum Y^2}} \qquad (1)$$

The principle of stacking states that it is preferable to use learners that are thinly correlated in the first-stage to obtain better performance at the second-stage [27]. To select thinly correlated methods, we performed a pair-wise correlation analysis of the residue-wise probabilities between the classifiers. To obtain residue-wise probabilities, the classifiers were trained on benchmark dataset and the probabilities for each residue in the TS49 test set was obtained through an independent test. The results of these correlations are shown in Table 3.

Table 3. Pair-wise correlation analysis of the probability distribution given by the base-classifiers on TS49.

Classifiers	ETC	GBC	KNN	LOG REG	RDF	BAG	SVM	XGB
ETC	–	0.918	0.881	0.946	0.997	0.989	0.875	0.868
GBC	–	–	**0.762**	0.922	0.924	0.936	**0.782**	0.727
KNN	–	–	–	**0.816**	0.879	0.861	0.831	0.804
LOG REG	–	–	–	–	0.951	0.953	0.824	0.774
RDF	–	–	–	–	–	0.994	0.875	0.878
BAG	–	–	–	–	–	–	0.863	0.855
SVM	–	–	–	–	–	–	–	0.809

Identified least pair-wise correlation scores are bold faced.

Since the SVM was found to be the top performing method from the above comparison between different classifiers, it was selected as a meta-classifier to create the fourth combination

(i.e., Model-4). Next, the method which is least correlated with SVM was identified. From Table 3, we can see that the SVM is least correlated with the GBC with a correlation coefficient of 0.782. Thus, GBC was selected as the first base-classifier. Consequently, the next method which is least correlated with GBC was identified. Again, from Table 3, we found that GBC is least correlated with XGB. However, as both XGB and GBC are based on boosting principle, instead of selecting XGB, next least correlated method was identified. The next least correlated method to GBC was found to be KNN with a correlation coefficient of 0.762. Thus, the KNN was selected as the second base-classifier. Successively, the least correlated method to KNN was identified. Table 3 shows that the least correlated method to KNN excluding GBC and XGB is LOGREG with a correlation coefficient of 0.816. The GBC and XGB were excluded because GBC was already selected as one of the base-classifier and XGB follows the same principle of boosting as GBC. Finally, with the above approach GBC, KNN and LOGREG were selected as base-classifiers to create Model-4. The performance of Model-4 and its comparison with other models is shown in Table 4.

Table 4. Comparisons of stacked models with different set of base-classifiers on benchmark dataset through 10-fold CV.

Metric/Method	Model-1	Model-2	Model-3	Model-4
Sensitivity	**0.861**	0.855	0.857	0.859
Specificity	**0.859**	0.859	0.859	0.859
Fall Out Rate	**0.141**	0.141	0.141	0.141
Miss Rate	**0.139**	0.145	0.143	0.141
Bal. Accuracy	**0.860**	0.857	0.858	0.859
Accuracy	**0.860**	0.857	0.858	0.859
Precision	**0.859**	0.858	0.859	0.859
F1 score	**0.860**	0.857	0.858	0.859
MCC	**0.720**	0.714	0.716	0.718

Best score values are bold faced.

Table 4 shows that the Model-1, which includes SVM, LOGREG, KNN and ETC as base-classifier and another version of SVM as meta-classifier, provides the highest performance. Thus, we select Model-1 as our final stacking model.

3.4 Performance Comparison on Benchmark Dataset

In this section, we compute the performance of StackCBPred using 10-fold CV on the benchmark dataset. While performing 10-fold CV the training of the StackCBPred was done using a balanced number of samples whereas, the testing was performed using a balanced as well as an imbalanced number of samples, respectively. Testing using the imbalanced number of samples in 10-fold CV was performed so that the results could be directly compared to SPRINT-CBH. Table 5 shows the performance comparison of StackCBPred with SPRINT-CBH. The quantities for all the evaluation metrics for SPRINT-CBH are obtained from [6].

From Table 5, we observed that the performance of SPRINT-CBH is biased more towards the negative class (non-carbohydrate binding) because of which the specificity (98.9%) is extremely high and the sensitivity (18%) is extremely low. When the test data is highly imbalanced, it is easy to achieve high overall accuracy (ACC) simply by predicting every test data point as the majority class which is what we can see from the result of SPRINT-CBH in Table 5. Balanced accuracy, which avoids inflated performance estimates on imbalanced datasets would be a proper measure of accuracy. However, the balanced accuracy of SPRINT-CBH was not reported in the literature. We computed the balanced accuracy of SPRINT-CBH by utilizing the expression of balanced accuracy provided in Table 1. Moreover, the main goal of the carbohydrate-binding sites prediction is to predict the binding sites accurately. However, due to the low sensitivity of 18%, the SPRINT-CBH bears the risk of not identifying the binding sites accurately. On the other hand, StackCBPred is able to predict the binding sites more accurately than the SPRINT-CBH based on the sensitivity and balanced accuracy scores as shown in Table 5. The sensitivity of the StackCBPred is 66.5% and 86.1% for the imbalanced and balanced number of samples used in testing through 10-fold CV. Additionally, the balanced accuracy of the StackCBPred is 66.5% and 86% for the imbalanced and balanced number of samples used in testing through 10-fold CV. The StackCBPred attains 13.68% improvement in balanced accuracy over SPRINT-CBH while, tested using imbalanced test set. These results indicate that StackCBPred is able to predict the binding sites more accurately compared to the SPRINT-CBH.

Table 5. Comparisons of StackCBPred with SPRINT-CBH on benchmark dataset.

Methods		Sensitivity	Specificity	Balanced Accuracy	Overall Accuracy	MCC
SPRINT-CBH	Imbalanced	0.180	0.989	0.585	0.950	0.250
StackCBPred	Imbalanced	0.665	0.664	0.665	0.664	0.134
	Balanced	0.861	0.859	0.860	0.860	0.720

3.5 Performance Comparison using Independent Test Datasets

In this section, we further examine the performance of StackCBPred by performing an independent test on two independent test datasets, TS49 and TS88. The TS49 dataset was recently constructed by Taherzadeh *et al*. [6] to test the performance of carbohydrate-binding site predictor, called SPRINT-CBH. However, the TS88 dataset was collected in this study to further test the robustness of StackCBPred. To test using TS49 and TS88, StackCBPred was first trained on balanced benchmark dataset and simultaneously tested on both the independent test datasets. Table 6 lists the predictive results of StackCBPred and SPRINT-CBH on the TS49 test set.

Table 6 indicates that the StackCBPred outperforms SPRINT-CBH by 42.16% and 80.72% based on sensitivity while, tested on the imbalanced and balanced TS49 test set, respectively. Similarly, StackCBPred attains 2.59% and 22.53% improvement in balanced accuracy over SPRINT-CBH while, tested using imbalanced and balanced TS49 test set, respectively. It is to be noted that the main goal here is to predict carbohydrate-binding sites thus, higher sensitivity is preferable. The

results in Table 6 also indicate that the sensitivity of StackCBPred improves from 55.3% to 70.3% and the balanced accuracy improves from 67.4% to 80.5% while, the number of carbohydrate-binding and non-binding residues are balanced in the TS49 test set. The improved sensitivity of carbohydrate binding sites prediction by StackCBPred on TS49 test set also indicates that StackCBPred predicts binding sites more accurately compared to the SPRINT-CBH predictor. Furthermore, the balanced accuracy measure indicates that the StackCBPred is not biased more towards the majority class rather it provides a balanced performance compared to SPRINT-CBH method.

Table 6. Comparisons of StackCBPred with SPRINT-CBH on balanced and imbalanced independent test dataset, TS49.

Methods		Sensitivity	Specificity	Balanced Accuracy	Overall Accuracy	MCC
SPRINT-CBH	Imbalanced	0.389	0.925	0.657	0.906	0.195
StackCBPred	Imbalanced	0.553	0.795	0.674	0.786	0.159
	Balanced	0.703	0.907	0.805	0.805	0.623

Additionally, the performance of StackCBPred and SPRINT-CBH was further evaluated on the TS88 test set and their prediction results are listed in Table 7. Table 7, shows that the sensitivity of StackCBPred is 334.62% better than SPRINT-CBH. Besides, the miss rate of SPRINT-CBH is 0.870 which is very close to 1. Therefore, the specificity of the SPRINT-CBH is very high i.e., it most of the time predicts the sample point as the majority class (non-carbohydrate-binding) which results into low sensitivity. Additionally, the balanced accuracy of the SPRINT-CBH is 20.74% lower compared to StackCBPred. Thus, these results indicate that StackCBPred predicts a greater number of carbohydrate-binding and non-binding residues correctly and therefore is also a balanced predictor of carbohydrate-binding sites.

Table 7. Comparisons of StackCBPred with SPRINT-CBH on imbalanced independent test dataset, TS88.

Methods	Sensitivity	Specificity	Fall Out Rate	Miss Rate	Bal. Accuracy	MCC
SPRINT-CBH	0.130	0.997	0.003	0.870	0.564	0.257
StackCBPred	0.565	0.797	0.203	0.435	0.681	0.139

Moreover, Figure 3 presents the ROC curves generated by StackCBPred and SPRINT-CBH, while the predictions are evaluated on the imbalanced TS88 test set. The ROC curves show the TPR (sensitivity)/FPR (1-specificity) pairs at different classification thresholds. It is evident from the ROC curves that the StackCBPred provides higher TPR compared to SPRINT-CBH at different classification thresholds. Moreover, the AUC score given by StackCBPred is about 1.18% higher than that of SPRINT-CBH.
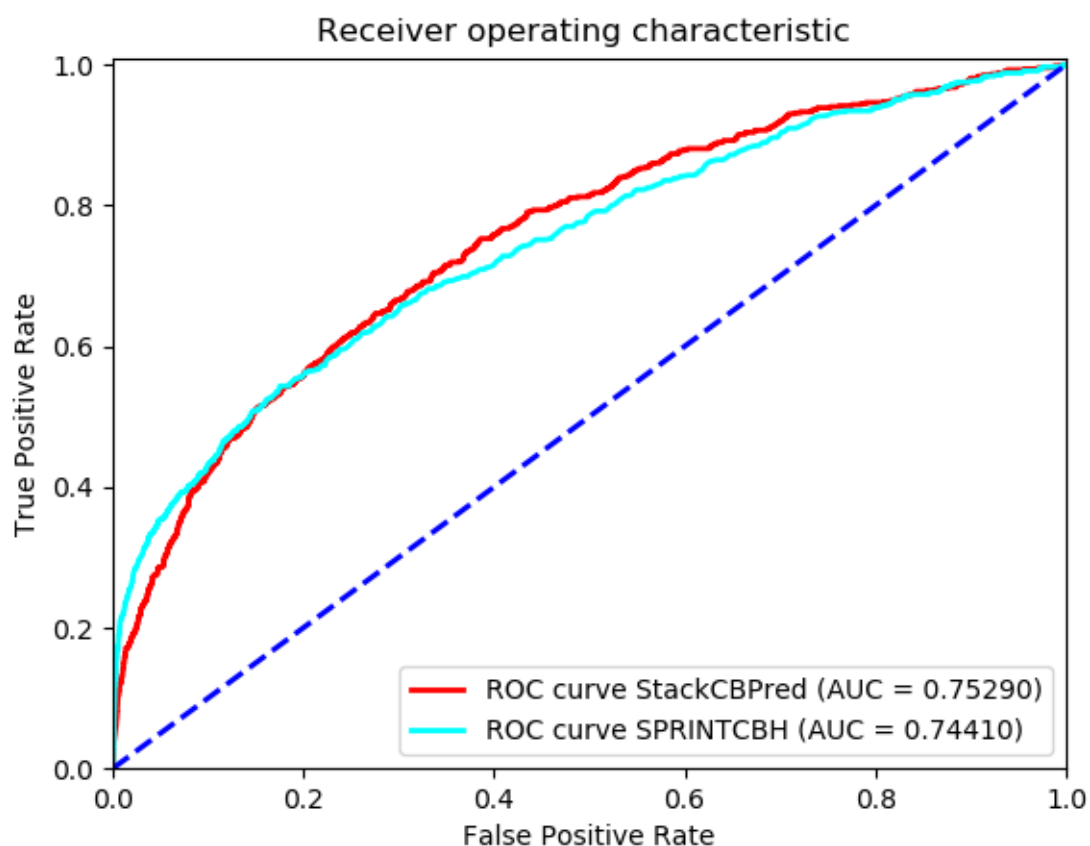
Fig 3. Comparison of ROC and AUC scores given by StackCBPred and SPRINT-CBH on imbalanced independent test dataset, TS88.

Conclusions

In this work, we have developed a stacking-based machine learning predictor, named StackCBPred, for the prediction of protein-carbohydrate binding sites directly from the protein sequence. We collected a benchmark dataset and two independent test datasets of high-resolution carbohydrate binding proteins to train, validate and independently test StackCBPred. Several important evolution-derived, sequence-based and structural features were extracted and chosen in an incremental fashion to find the trained best performing model. In addition, an advanced machine learning technique called stacking was implemented to ensure robust performance. We used incrementally chosen features to train the ensemble of predictors at the first-stage (i.e., base-layer). Then, we combined the output from the base-learners with the original features and used it as an input to the predictor at second-stage (i.e., meta-layer). Eventually, the meta-layer predictor of the StackCBPred achieves a 10-fold CV balanced accuracy and sensitivity of 86.00% and 86.09% respectively, on a balanced benchmark dataset. For the

balanced independent test dataset, TS49, StackCBPred attains a balanced accuracy and sensitivity of 80.51% and 70.28%, respectively. Furthermore, for the new imbalanced independent test dataset TS88 introduced in this work, StackCBPred attains a balanced accuracy and sensitivity of 68.46% and 56.39%, respectively. These results allow us to conclude that the stacking technique helps improve the accuracy significantly by reducing the generalization error. Moreover, comparative results highlight that the proposed method, StackCBPred, outperforms the existing method based on both benchmark and independent test datasets. These outcomes help us surmise that the StackCBPred can be effectively used for the rapid annotation of carbohydrate-binding sites directly from the sequence and can provide insight in treating critical diseases.

References

1. Shionyu-Mitsuyama, C., et al., *An empirical approach for structure-based prediction of carbohydrate-binding sites on proteins.* Protein Engineering, 2003. **16**(7): p. 467-478.
2. Fernandez-Alonso, M.d.C., et al., *Protein-carbohydrate interactions studied by NMR: from molecular recognition to drug design.* Current Protein and Peptide Science, 2012. **13**: p. 816-830.
3. Shin, I., S. Park, and M.r. Lee, *Carbohydrate Microarrays: An Advanced Technology for Functional Studies of Glycans.* Chemistry - European Journal, 2005. **11**: p. 2894-2901.
4. Sharon, N. and H. Lis, *Lectins*. 2 ed. 2003, The Netherlands: Springer. 454.
5. Wimmerová, M., et al., *Stacking interactions between carbohydrate and protein quantified by combination of theoretical and experimental methods.* PLoS ONE, 2012. **7**(10): p. e46032.
6. Taherzadeh, G., et al., *Sequence-based prediction of protein–carbohydrate binding sites using support vector machines.* Journal of Chemical Information and Modeling, 2016. **56**(10): p. 2115-2122.
7. McKinley, M.P., et al., *Human anatomy.* Fourth edition ed. 2015, New York, NY: McGraw-Hill Education.
8. Malik, A. and S. Ahmad, *Sequence and structural features of carbohydrate binding in proteins and assessment of predictability using a neural network.* BMC Structural Biology, 2007. **7**(1).
9. Malik, A. and S. Ahmad, *Sequence and structural features of carbohydrate binding in proteins and assessment of predictability using a neural network.* BMC Structural Biology, 2007. **7**: p. 1.
10. Brown, A. and M.K. Higgins, *Carbohydrate binding molecules in malaria pathology.* Current Opinion in Structural Biology, 2010. **20**(5): p. 560-566.
11. François, K. and J. Balzarini, *Potential of carbohydrate-binding agents as therapeutics against enveloped viruses.* Medicinal Research Reviews, 2012. **32**: p. 349-387.
12. Raz, A. and S. Nakahara, *Biological modulation by lectins and their ligands in tumor progression and metastasis.* Anti-Cancer Agents in Medicinal Chemistry, 2008. **8**(1): p. 22-36.
13. Morris, G.M., et al., *Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function.* Journal of Computational Chemistry, 1998. **19**(14): p. 1639-1662.
14. Friesner, R.A., et al., *Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy.* Journal of Medical Chemistry, 2004. **47**(7): p. 1739-1749.
15. Moustakas, D.T., et al., *Development and validation of a modular, extensible docking program: DOCK 5.* Journal of Computer-Aided Molecular Design, 2006. **20**(10-11): p. 601-619.
16. Taroni, C., S. Jones, and J.M. Thornton, *Analysis and prediction of carbohydrate binding sites.* Protein Engineering, 2000. **13**(2): p. 89-98.

17. Sujatha, M.S. and P.V. Balaji, *Identification of common structural features of binding sites in galactose-specific proteins.* Proteins: Structure. Function. Bioinformatics, 2004. **55**(1): p. 44-65.
18. Kulharia, M., et al., *InCa-SiteFinder: A method for structure-based prediction of inositol and carbohydrate binding sites on proteins.* Journal of Molecular Graphics and Modelling, 2009. **28**(3): p. 297-303.
19. Nassif, H., et al., *Prediction of protein-glucose binding sites using support vector machines.* Proteins: Structure, Function, Bioinformatics, 2009. **77**(1): p. 121-132.
20. Tsai, K.-C., et al., *Prediction of carbohydrate binding sites on protein surfaces with 3-dimensional probability density distributions of interacting atoms.* PLoS ONE, 2012. **7**(7): p. e40846.
21. Gromiha, M.M., K. Veluraja, and K. Fukui, *Identification and analysis of binding site residues in proteincarbohydrate complexes using energy based approach.* Protein and Peptide Letters, 2014. **21**(8): p. 799-807.
22. Shanmugam, N.R.S., et al., *Identification and analysis of key residues involved in folding and binding of protein-carbohydrate complexes.* Protein and Peptide Letters, 2018. **25**(4): p. 379-389.
23. Deng, L., et al., *Boosting prediction performance of protein–protein interaction hot spots by using structural neighborhood properties.* Journal of Computational Biology, 2013. **20**(11): p. 878-891.
24. Lei, C. and J. Ruan, *A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity.* Bioinformatics, 2013. **29**(3): p. 355-364.
25. Rao, V.S., et al., *Protein-protein interaction detection: methods and analysis.* International Journal of Proteomics, 2014. **2014**.
26. Liang, S., et al., *Protein binding site prediction using an empirical scoring function.* Nucleic Acids Research, 2006. **34**(13): p. 3698-3707.
27. Iqbal, S. and M.T. Hoque, *PBRpredict-Suite: a suite of models to predict peptide-recognition domain residues from protein sequence.* Bioinformatics, 2018: p. bty352-bty352.
28. Lavi, A., et al., *Detection of peptide-binding sites on protein surfaces: The first step towards the modeling and targeting of peptide-mediated interactions.* Proteins: Structure, Function, Bioinformatics, 2013. **81**(12): p. 2096-2105.
29. Petsalaki, E., et al., *Accurate prediction of peptide binding sites on protein surfaces.* PLoS Computational Biology, 2009. **5**(3).
30. Taherzadeh, G., et al., *Sequence-based prediction of protein–peptide binding sites using support vector machine.* Journal of Computational Chemistry, 2016. **37**(13): p. 1223-1229.
31. Lin, C.-K. and C.-Y. Chen, *PiDNA: predicting protein–DNA interactions with structural models.* Nucleic Acids Research, 2013. **41**: p. W523-W530.
32. Si, J., et al., *MetaDBSite: a meta approach to improve protein DNA-binding sites prediction.* BMC System Biology, 2011. **5**.
33. Zhao, H., et al., *Predicting DNA-binding proteins and binding residues by complex structure prediction and application to human proteome.* PLoS ONE, 2014. **9**.
34. Mishra, A., P. Pokhrel, and M.T. Hoque, *StackDPPred: a stacking based prediction of DNA-binding protein from sequence.* Bioinformatics, 2018: p. bty653.
35. Murakami, Y., et al., *PiRaNhA: a server for the computational prediction of RNA-binding residues in protein sequences.* Nucleic Acids Research, 2010. **38**: p. W412-W416.
36. Zhang, T., et al., *Analysis and prediction of RNA-binding residues using sequence, evolutionary conservation, and predicted secondary structure and solvent accessibility.* Current Protein and Peptide Science, 2010. **11**(7): p. 609-628.
37. Zhao, H., Y. Yang, and Y. Zhou, *Highly accurate and high-resolution function prediction of RNA binding proteins by fold recognition and binding affinity prediction.* RNA Biology, 2011. **8**(6): p. 988-996.

38. Zhang, X. and S. Liu, *RBPPred: predicting RNA-binding proteins from sequence using SVM.* Bioinformatics, 2017. **33**(6): p. 854-862.

39. Bolia, A., Z.N. Gerek, and S.B. Ozkan, *BP-Dock: a flexible docking scheme for exploring protein–ligand interactions based on unbound structures.* Journal of Chemical Information and Modeling, 2014. **54**(3): p. 913-925.

40. Komiyama, Y., et al., *Automatic generation of bioinformatics tools for predicting protein–ligand binding sites.* Bioinformatics, 2016. **32**(6): p. 901-907.

41. Yang, Y., J. Zhan, and Y. Zhou, *SPOT-Ligand: fast and effective structure-based virtual screening by binding homology search according to ligand and receptor similarity.* Journal of Computational Chemistry, 2016. **37**(18): p. 1734-1739.

42. Litfin, T., Y. Zhou, and Y. Yang, *SPOT-ligand 2: improving structure-based virtual screening by binding-homology search on an expanded structural template library.* Bioinformatics, 2017. **33**(8): p. 1238-1240.

43. Taherzadeh, G., et al., *Sequence-based prediction of protein–carbohydrate binding sites using support vector machines.* Journal of Chemical Information and Modeling, 2016. **56**.

44. Iqbal, S. and M.T. Hoque, *Estimation of position specific energy as a feature of protein residues from sequence alone for structural classification.* PLOS ONE, 2016. **11**(9): p. e0161452.

45. Yen, S.-J. and Y.-S. Lee, *Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset.* Springer Berlin Heidelberg, 2006: p. 731-740.

46. Malik, A., et al., *PROCARB: a database of known and modelled carbohydrate-binding protein structures with sequence-based prediction tools.* Advances in Bioinformatics, 2010. **2010**: p. 436036.

47. Johnson, M., et al., *NCBI BLAST: A better web interface.* Nucleic Acids Research, 2008. **36**: p. W5-W9.

48. Iqbal, S., A. Mishra, and T. Hoque, *Improved Prediction of Accessible Surface Area Results in Efficient Energy Function Application.* Journal of Theoretical Biology, 2015. **380**: p. 380-391.

49. Islam, N., et al., *A Balanced Secondary Structure Predictor.* Journal of Theoretical Biology, 2016. **389**: p. 60-71.

50. Biswas, A.K., N. Noman, and A.R. Sikder, *Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information.* BMC Bioinformatics, 2010. **11**(273).

51. Verma, R., G.C. Varshney, and G.P.S. Raghava, *Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile.* Amino Acids, 2010. **39**(1): p. 101-110.

52. Glaser, F., et al., *ConSurf: identification of functional regions in proteins by surface-mapping of phylogenetic information.* Bioinformatics, 2003. **19**(1): p. 163-164.

53. Altschul, S.F., et al., *Basic Local Alignment Search Tool.* J. Mol. Biol., 1990. **215**: p. 403-410.

54. Sharma, A., et al., *A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition.* Journal of Theoretical Biology, 2013. **320**: p. 41-46.

55. Faraggi, E., et al., *SPINE X: Improving protein secondary structure prediction by multi-step learning coupled with prediction of solvent accessible surface area and backbone torsion angles.* Journal of Computational Chemistry, 2012. **33**(3): p. 259-67.

56. Hamelryck, T., *An amino acid has two sides: A new 2D measure provides a different view of solvent exposure.* Proteins: Structure, Function, Bioinformatics, 2005. **59**(1): p. 38-48.

57. Heffernan, R., et al., *Highly accurate sequence-based prediction of half-sphere exposures of amino acid residues in proteins.* Bioinformatics, 2016. **32**(6): p. 843-849.

58.   Heffernan, R., et al., *Single-sequence-based prediction of protein secondary structures and solvent accessibility by deep whole-sequence learning.* Journal of Computational Chemistry, 2018 (in press).

59.   Heffernan, R., et al., *Capturing non-local interactions by long short term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers, and solvent accessibility.* Bioinformatics, 2017. **33**(18): p. 2842-2849.

60.   Meiler, J., et al., *Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks.* Journal of Molecular Modeling 2001. **7**: p. 360-369.

61.   Bah, A. and J.D. Forman-Kay, *Modulation of intrinsically disordered protein function by post-translational modifications.* Journal of Biological Chemistry, 2016. **291**: p. 6696-6705.

62.   Lina, Y.-H., et al., *The intrinsically disordered N-terminal domain of galectin-3 dynamically mediates multisite self-association of the protein through fuzzy interactions.* Journal of Biological Chemistry, 2017. **292**: p. 17845-17856.

63.   Sharma, R., et al., *OPAL: prediction of MoRF regions in intrinsically disordered protein sequences.* Bioinformatics, 2018. **34**(11): p. 1850-1858.

64.   Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. 2 ed. Springer Series in Statics. 2009: Springer-Verlag New York.

65.   Wolpert, D.H., *Stacked generalization.* Neural Networks, 1992. **5**(2): p. 241-259.

66.   Hu, Q., et al., *A Stacking-Based Approach to Identify Translated Upstream Open Reading Frames in Arabidopsis Thaliana*, in *International Symposium on Bioinformatics Research and Applications*. 2015, Bioinformatics Research and Applications. p. 138-149.

67.   Nagi, S. and D.K. Bhattacharyya, *Classification of microarray cancer data using ensemble approach.* Network Modeling Analysis in Health Informatics and Bioinformatics, 2013. **2**(3): p. 159-173.

68.   Verma, A. and S. Mehta, *A comparative study of ensemble learning methods for classification in bioinformatics*, in *7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*. 2017, IEEE: Noida, India.

69.   Vapnik, V.N., *An Overview of Statistical Learning Theory.* IEEE TRANSACTIONS ON NEURAL NETWORKS, 1999. **10**(5).

70.   Friedman, J.H., *Greedy function approximation: a gradient boosting machine.* The Annals of Statistics, 2001. **29**(5): p. 1189-1232.

71.   Breiman, L., *Bagging predictors.* Machine Learning, 1996. **24**(2): p. 123-140.

72.   Geurts, P., D. Ernst, and L. Wehenkel, *Extremely randomized trees.* Machine Learning, 2006. **63**(1): p. 3-42.

73.   Ho, T.K., *Random decision forests*, in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*. 1995, IEEE: Montreal, Que., Canada. p. 278-282.

74.   Altman, N.S., *An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression.* The American Statistician, 1992. **46**: p. 175-185.

75.   Szilágyi, A. and J. Skolnick, *Efficient prediction of nucleic acid binding function from low-resolution protein structures.* Journal of Molecular Biology, 2006. **358**(3): p. 922-933.

76.   Chen, T. and C. Guestrin, *XGBoost: a scalable tree boosting system.*, in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, ACM: New York, NY, USA. p. 785-794.

77.   Bergstra, J. and Y. Bengio, *Random Search for Hyper-Parameter Optimization.* Journal of Machine Learning Research, 2012. **13**.

78.   Pedregosa, F., et al., *Scikit-learn: Machine learning in python.* Journal of Machine Learning Research, 2012. **12**.