Chapter 5

StackSSSPred: A Stacking-Based Prediction of Supersecondary Structure from Sequence

Michael Flot, Avdesh Mishra, Aditi Sharma Kuchi, and Md Tamjidul Hoque

Abstract

Supersecondary structure (SSS) refers to specific geometric arrangements of several secondary structure (SS) elements that are connected by loops. The SSS can provide useful information about the spatial structure and function of a protein. As such, the SSS is a bridge between the secondary structure and tertiary structure. In this chapter, we propose a stacking-based machine learning method for the prediction of two types of SSSs, namely, β-hairpins and β-α-β, from the protein sequence based on comprehensive feature encoding. To encode protein residues, we utilize key features such as solvent accessibility, conservation profile, half surface exposure, torsion angle fluctuation, disorder probabilities, and more. The usefulness of the proposed approach is assessed using a widely used threefold cross-validation technique. The obtained empirical shows that the proposed approach is useful and prediction can be improved further.

Key words Supersecondary structure prediction, Beta-hairpins, Beta-alpha-beta, Stacking, Machine learning, Sequence-based prediction

1 Introduction

A protein macromolecule is a linear chain of amino acid residues linked together by peptide bond. Protein structure can be described in terms of four different hierarchies of structural and folding patterns-based complexities: *primary structure* is the sequence of amino acid chain only that makes up a polypeptide chain without any structural information; *secondary structure* concerns regular, repeated local three-dimensional (3D) segments of proteins including α-helix and β-strand; *tertiary structure* is the global 3D structure of a protein molecule; and *quaternary structure* describes the way in which the different tertiary subunits are packed together to form the structure of a protein complex [1]. The *supersecondary structures* bridge the secondary structure and the tertiary structure of a protein. Secondary structure elements connected by a

Michael Flot and Avdesh Mishra contributed equally to this work.

Alexander E. Kister (ed.), *Protein Supersecondary Structures: Methods and Protocols*, Methods in Molecular Biology, vol. 1958, https://doi.org/10.1007/978-1-4939-9161-7_5,

polypeptide (loop) in specific geometric arrangements are called motifs or supersecondary structures [2]. These supersecondary structures (SSSs) can provide information about the spatial structure of a protein. Some of the most commonly occurring SSSs include α-helix hairpins, β-hairpins, β-α-β, coiled coils, Greek key, α-loop-α, α-turn-α, and Rossmann motifs. Accurate knowledge of 3D structure of a protein provides insight on a protein's function, which is crucial in effective design and development of drug.

The classic work of Anfinsen, in the 1950s, on the enzyme ribonuclease revealed the relation between the amino acid sequence of a protein and its conformation. Through his experiments, Anfinsen showed that the information needed for a protein to obtain its 3D structure is contained in its amino acid sequence. Nevertheless, prediction of 3D structure of protein from sequence remains as one of the greatest challenge for the scientific community [3, 4]. Investigators are exploring two fundamentally different approaches of predicting the 3D structure from amino acid sequence. The first is ab initio or de novo protein structure prediction (*aiPSP*), which attempts to build the structure from the sequence of amino acid residues without prior knowledge about similar sequences in known protein structure database [5–12]. Computational methods are employed that attempt to minimize the free energy of a structure with a given sequence or to simulate the folding process. Molecular dynamics (MD) is an example of ab initio method that performs simulation of the protein folding process. MD has been successfully applied for the prediction of small proteins and peptides as well as for the refinement of the structures (both small and large proteins) by minimizing the energy, to some extent [13, 14]. The second approach is dependent on the availability of similar templates in the protein database and is commonly known as homology modelling [15–19]. Amino acid sequence of a known structure or fragments is scanned for sequence similarity with the sequence of the target protein with unknown structures, and if a significant match is detected, the known structural knowledge is applied to construct the final model. Moreover, the prediction of tertiary structure of a protein can also be achieved by proceeding in a hierarchal fashion. First, the secondary structure of the protein is predicted from the amino acid sequence, then the supersecondary structures are derived from the secondary structure elements, and finally the information about the secondary and SSSs is used to computationally determine the 3D shape of the protein molecule [19–24].

The past decade has witnessed tremendous progress in the development of accurate predictors of secondary structure. Some of the recent and successful predictors of secondary structures include SSpro [25], Spider 3 [26], Spider 2 [27], and SPINE X [28]. As reported, SSpro achieved highest accuracy of 92.9% for secondary structure prediction by combining sequence similarity

and sequence-based structural similarity. In addition to being useful for the prediction of the tertiary structure, the secondary structure predicted from the sequence is widely applied for the analysis and prediction of numerous structural and functional properties of proteins. These properties include prediction of RNA-binding proteins [29], DNA-binding protein and their binding sites [30], protein-peptide interactions [31], protein-carbohydrate interaction [32], residue contacts [8, 33], disorder region [34, 35], accessible surface of amino acids [36], target selection for structural genomics [37, 38], and more.

In the past, many attempts have been made in predicting individual SSSs types, and several effective computational prediction methods have been proposed in literature for analyzing them, such as β hairpins [39, 40], β-α-β [2, 41], coiled coils [42, 43], and helix-turn-helix motifs [44–46]. Many of the SSS prediction methods capitalize on the fact that the prediction of secondary structure provides useful information for the prediction of SSS [2, 47]. Predicted SSSs are useful features for various applications, such as simulation of protein folding [48], analysis of relation between coiled coils and disorder regions [49], study and identification of many functional and active sites [2], analysis of amyloids [50], genome-wide studies of protein structure [51, 52], and prediction of protein domains [53].

In this chapter, we present a machine learning (ML) approach for the prediction of SSSs directly from the sequence of amino acids instead of following the traditional hierarchical approach of first predicting the secondary structures and then utilizing the predicted SS types (labels) to predict the SSSs. We implement several ML methods along with a recently studied [31, 54] stacking-based ML predictor for two different types of supersecondary structures β-hairpins and β-α-β. The stacking-based ML approach combines the information of several different ML algorithms to generate a new prediction model. It provides a scheme for minimizing the generalization error rate of one or more predictive models. The utility of the proposed approach is fast assessed by threefold cross-validation approach. The results obtained from extensive examination shows that the proposed approach is time-consuming, yet very promising. Along with detailed methodology and explanation of required tools, techniques, and resources, we provide useful notes to assist readers with the process of improving the prediction accuracy of the proposed method.

2 Materials

In this section, we describe the procedure for benchmark dataset preparation, tools necessary for class label assignment, aggregation and encoding of input features, machine learning algorithms, and the criteria to evaluate them.

2.1 Dataset

We built up a benchmark dataset [55] of protein sequences collected from the protein data bank [56] using an *Advanced Search* interface with the following specifications: (1) experimental method, X-ray; (2) molecule type, protein; (3) X-ray resolution, < 1.5 Å; (4) chain length, ≥ 40; and (5) sequence identity (cutoff), 30%. This resulted in 3474 proteins. The chains of these proteins were split into separate structures, and the sequence from these single-chain structures were extracted resulting in 5388 different sequences. To reduce bias from too many similar sequences, BLASTCLUST [57] was used to reduce sequence similarity to 25%. Keeping just the first of each cluster reduced the number of sequences to 3349.

Furthermore, we discarded the protein sequences with unknown amino acid, labelled with "X" character, because of the unavailability of the corresponding features. Structures with unknown coordinates of amino acids were removed as well, because the corresponding supersecondary structure of the amino acids could not be obtained. Moreover, to train the ML algorithms, several tools were used to generate features from sequence (*see* Subheading 2.3). For some of the sequences, these tools failed to generate useful information. Such sequences were discarded from further consideration. We also discarded sequences where we found that the length of a sequence given by the tool's output and the length of a FASTA sequence provided by the collected PDB files differed. Finally, this reduced the number of sequences to 3203.

In addition, if none of the amino acid residues in a protein sequence were labelled as either β-hairpin or β-α-β, such sequences were discarded from their respective benchmark dataset. As a result, the β-hairpin dataset contains 2520 proteins, and the β-α-β dataset contains 1208 proteins.

2.2 Assignment of Supersecondary Structures

The SSS is composed of two secondary structure units connected by a polypeptide (loop) with a specific arrangement of geometry. Among more than a dozen types of the SSSs, the β-hairpins, coiled coils, α-turn-α, and β-α-β motifs received more attention due to the fact that they are present in a large number of protein structures and play an important role in many biological activities. In this study, we focus on the study of β-hairpins and β-α-β motifs. The second largest group of protein domains is the β-hairpins. They are found in diverse protein families, including enzymes, transporter proteins, antibodies, and viral coats [47]. The β-hairpin motif consists of two strands that are adjacent in primary structure, oriented in an antiparallel direction, and linked by a short loop of two to five amino acids. On the other hand, β-α-β is a complex supersecondary structure in proteins and often appears in *Bacillus subtilis* proteases [58]. The study of β-α-β motifs is important because many functional as well as active sites often occur in the polypeptide of β-α-β motifs, including ADP-binding sites, FAD-binding sites,

NAD-binding sites, and more [59]. In this work, we used the PROMOTIF [60] program to generate annotations (labels) for two types of supersecondary structures β-hairpin and β-α-β predictors. PROMOTIF is a program like DSSP [61] as it uses the distances and hydrogen bonding between residues to assign supersecondary structures. The single-chain protein structures are passed to the PROMOTIF program to obtain the information about the residues which belong to β-hairpin or β-α-β motifs. Based on the outcome of the PROMOTIF program, if the residue belongs to the β-hairpin or β-α-β motif, the residue is labelled as "1" else "0," respectively.

2.3 Feature Extraction

Feature extraction and encoding is an important step in the development of machine learning-based predictors. To create an effective machine learning-based method to predict β-hairpin and β-α-β motifs from sequence alone, we use various sequence and structure-based features. These features provide information about the chemical, structural, and flexibility profiles of the proteins. A set of features used in this study are listed in Table 1 and are briefly discussed below.

1. *Amino acid (AA)*: Twenty different standard amino acids were encoded using 20 different integers ranging from 1 to

Table 1
List of features used in SSS prediction

Feature category	Features count
Amino acid (AA)	1
Physiochemical properties (PP)	7
Position-specific scoring matrix	20
Secondary structure probabilities	6
Accessible surface area	1
Torsion angle (φ, ψ) fluctuation	2
Monogram	1
Bigram	20
Position-specific estimated energy	1
Terminal indicator (TI)	1
Disorder probability	1
Phi and psi torsion angles	2
Half sphere exposures	2
Total	65

20, which is a useful feature to capture the amino acid composition.

2. *Physicochemical properties (PP)*: Seven different physiochemical properties per amino acid, namely, steric parameter, normalized van der Waals volume, hydrophobicity, isoelectric point, and helix and sheet probabilities, were collected from DisPredict2 [35] program. These features were originally reported in [62].
3. *Position-specific scoring matrix (PSSM)*: PSSM captures the conservation pattern using multiple sequence alignments and stores this pattern as a matrix of scores for each position in the alignment. High scores in this matrix represent more conserved positions, and scores close to zero or negative represent weakly conserved position. Thus, PSSM provides the evolutionary information in proteins. Evolutionary information is one of the most important kinds of information for protein functionality prediction in biological analysis and is widely used in such studies [34, 36, 63–66]. We executed three iterations of PSI-BLAST [67] against NCBI's nonredundant database to generate PSSM of size sequence length × 20, which gave us 20 features per residue.
4. *Monogram (MG) and bigram (BG)*: The monogram (single feature) and bigram (20 features) were computed from PSSM by further extending the PSSM values to higher dimension. Both of these features were collected from DisPredict2 program. These features are found to be useful in protein fold recognition [68, 69] and various other applications such as disordered prediction [35] and protein-peptide binding [31].
5. *Local structural properties*: We collected a total of eleven predicted local structural features, which include three secondary structures probabilities for helix (H), beta (B), and coil (C) obtained from MetaSSpred [64] and three additional SS probabilities obtained from Spider 3 [26]; two torsion angles, phi (Φ) and psi (Ψ); one accessible surface area (ASA); and two half sphere exposure (HSE), namely, HSE-up and HSE-down. The torsion angles and HSE features were predicted using Spider 3 program. ASA was predicted using DisPredict2 which generates this feature from Spine X [28].
6. *Flexibility properties*: We include multiple flexibility properties of amino acids, which include two torsion angle fluctuations, dphi (∇Φ) and dpsi (∇Ψ), and one disorder probability. The torsion angle features can be originally predicted using DAVAR [70]; however, all the above features were extracted from DisPredict2.
7. *Energy features*: Since many functional sites and active sites often occur in the polypeptide of β-α-β motifs, they play a

significant role in binding. The binding of protein and ligand involves formation and dissolution of atomic interactions that require change in free energy [71]. Thus, to capture the state of free energy contribution of residues, we include position-specific estimated energy (PSEE) which was also predicted using DisPredict2.

8. *Terminal region:* Often terminal residues of a protein show higher flexibility. Thus, to distinguish the terminal residues from others, we included terminal indicator feature by encoding five residues of N-terminal as $[-1.0, -0.8, -0.6, -0.4, -0.2]$ and C-terminal as $[+1.0, +0.8, +0.6, +0.4, +0.2]$, respectively, whereas the rest of the residues were labelled as 0.0.

Before using the features mentioned above into the classifier, different sized sliding windows were evaluated. This technique is used to incorporate neighboring information for each residue. Sliding windows work by aggregating information on both sides of the target residue. For example, if window size 11 is chosen, the corresponding features for 5 neighboring residues are gathered on either side of the target residue which generates (11×65) or, 715 features per residue.

2.4 Machine Learning Algorithms

In this study, we explored five different potential machine learning algorithms for the prediction of two types of SSSs: β-hairpin and β-α-β. The implemented algorithms are briefly discussed below. All of the classifiers used in our study are built and tuned using scikit-learn [72].

1. *K Nearest Neighbor (KNN) Classifier:* The KNN algorithm compares an input to the *K* closest training examples [73]. A majority vote coming from the most similar neighbors in the training set decides the classification. We used Euclidean distance as a metric for finding the nearest neighbors. As the idea of learning a model using KNN is simple, this method is computationally cheap. For all our experiments with KNN method, the value of *K* was set to 7 and all the neighbors were weighted uniformly.
2. *Extra Tree (ET) Classifier:* The extremely randomized tree or ET [74] is one of the ensemble methods, which constructs randomized decision trees from the original learning sample and uses above-average decision to improve the predictive accuracy and control over-fitting. We constructed the ET model with 1000 trees, and the quality of a split was measured by Gini impurity index.
3. Gradient Boosting Classifier (GBC): The GBC works by combining weak learners into a single learner in an iterative fashion [75]. We applied 1000 boosting stages where a regression tree was fit on the negative gradient of the deviance loss function. In

our implementation, the learning rate was set to 0.1 and the maximum depth of each regression tree was set to 3. GBC overcomes over-fitting with higher number of boosting stages, and we observed that 1000 stages were giving competitive performance for this application.

4. *Logistic Regression (LogReg):* We used LogReg [76] with L2 regularization for the prediction of SSSs. LogReg measures the relationship between the dependent variable, which is categorical (in our case: whether an amino acid belongs to SSSs type or not), and one or more independent variables by generating an estimation probability using logistic regression. It utilizes the sigmoid function to predict the output [77].
5. *Random Decision Forest (RDF):* The RDF [77, 78] operates by constructing a multitude of decision trees on various sub-samples of the dataset and results the mean prediction of the decision trees to improve the prediction accuracy and control over-fitting. We used bootstrap samples to construct 1000 trees in the forest.

2.5 Performance Metrics

To build the proof-of-concept of stacking versus non-stacking approach fast, we used threefold cross validation (FCV) [76, 79, 80] to compare and evaluate the performance of each predictor. FCV is performed in folds, where the data is divided into m parts, which are each of about equal size. While a fold is set aside for testing, the other $(m - 1)$ folds are used to train the classifier. This process is repeated until each fold has been set aside once for testing and then the m estimates of error are combined to find the average. We employed various performance measures listed in Table 2 to test the predictive ability of various predictors. The majority of the metrics listed in the table are computed from the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) metrics. TP refers to the number of instances that are correctly predicted as positive. FP refers to the number of instances that are incorrectly predicted as positive. TN refers to the number of instances that are correctly labelled as negative. FN refers to the number of instances that are incorrectly labelled as negative. Recall is defined as proportion of real positive cases that are correctly predicted positive. Similarly, precision is defined as proportion of predicted positive cases that are correctly real positives. Likewise, F1 score is defined as the harmonic mean of recall and precision. The miss rate and fallout rate measure two complementary types of incorrect predictions. The miss rate is defined as proportion of real positive cases that occur as predicted negative. Similarly, fallout rate is defined as proportion of real negative cases that are correctly predicted positive. Furthermore, *Matthews correlation coefficient* (MCC) measures the degree of overlap between the predicted

Table 2
Name and definition of the evaluation metric

Name of metric	Definition
True positive (TP)	Correctly predicted supersecondary structures
True negative (TN)	Correctly predicted non-supersecondary structures
False positive (FP)	Incorrectly predicted supersecondary structures
False negative (FN)	Incorrectly predicted non- supersecondary structures
Recall/sensitivity	$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP+FN}$
Specificity	$True\ Negative\ Rate\ (TNR) = \frac{TN}{FP+TN}$
Fallout (or overprediction) rate	$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP+TN}$
Miss rate	$False\ Negative\ Rate\ (FNR) = \frac{FN}{FN+TP}$
Accuracy (ACC)	$\frac{TP+TN}{FP+FP+TN+FN}$
Balanced accuracy (mean of specificity and recall)	$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$
Precision	$\frac{TP}{TP+FP}$
F1 score (harmonic mean of precision and recall)	$\frac{2TP}{2TP+FP+FN}$
Matthews correlation coefficient (MCC)	$\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

labels and true labels of all the samples in the benchmark dataset. Lastly, the balanced accuracy is defined as mean of recall and specificity.

3 Methods

In this section, we discuss the implementation of stacking-based machine learning approach for the prediction of two types of supersecondary structures: β-hairpin and β-α-β only from sequence. We first discuss the results of individual classifiers and, subsequently, report the performance of various stacked predictors based on the benchmark dataset.

3.1 Stacking Framework

We applied the stacking technique [81] to deal with the sequence-based supersecondary structure prediction problem. Stacking is an ensemble technique which minimizes the generalization error by combining information from multiple predictive models to generate a new model. Because stacking minimizes the generalization error rate of one or more predictive models, it has been successfully applied in several ML tasks [82–86] and recently has been shown to

work well with the prediction of protein-peptide binding sites [31] and prediction of DNA-binding proteins [54].

The stacking method uses a two-tier learning framework. The first (i.e., base) tier consists of a collection of classifiers called base-learners. In the second (i.e., meta) tier, the outputs of the base-level learners are combined with the original input vector and fed to another classifier called a meta-learner. This method considers the fact that different base-learners can react to certain regions of the feature space poorly due to the no-free-lunch theorem [87]. Thus, using meta-learner in the second tier, the outputs of the base classifiers are combined with the aim of reducing the generalization error. For a better performance, it is desirable to choose classifiers that are highly uncorrelated to each other [31] or are different from each other based on their underlying operating principle as the base classifiers [54]. As stacking combines the outputs from the first tier in the second tier, this makes the stacking technique different from other ensemble methods like bagging and boosting as these techniques apply weighted average or majority vote to form a final prediction.

The base and meta-classifiers used in the stacking framework for this experiment include (a) Logistic Regression (LogReg) [23, 76], (b) Extra Trees (ET) [74], (c) Random Decision Forest (RDF) [78], (d) K Nearest Neighbor (KNN) [73], and (e) Gradient Boosting Classifier (GBC) [75]. These algorithms and their configurations are briefly discussed in Subheading 2.4. For each algorithm, feature window size which results in best accuracy was identified, and then, the classifiers with their respective best window sizes were used in the stacking framework.

In our implementation of stacking framework, we explored four different classifiers KNN, ET, GBC, and RDF as both meta- and base classifiers. While one of the four methods was used as the meta-learner, the rest of the methods were used as the base-learners. We dropped LogReg classifier out from stacking because it took longer to train this classifier on our benchmark dataset. The combinations of stacked model (SM) separately assessed for both β-hairpin and β-α-β in this study are:

1. **SM1:** includes KNN, GBC, and RDF as base-learners and ET as meta-learner.
2. **SM2:** includes ET, GBC, and RDF as base-learners and KNN as meta-learner.
3. **SM3:** includes ET, KNN, and RDF as base-learners and GBC as meta-learner.
4. **SM4:** includes ET, KNN, and GBC as base-learners and RDF as meta-learner.

The output probabilities (probability p belonging to β-hairpin or β-α-β and probability $(1 - p)$ not belonging to β-hairpin or

β-α-β) generated by the respective base classifiers are combined with the original windowed feature vector to train a new meta-classifier. In our implementation, we found that a window size of 11 gives the highest performance for each of the classifier. Thus, in stacking all the base-learners were trained using best window size feature vector of (65 × 11) or 715 and the meta-learners were trained using best window size features plus six additional probability features, resulting into feature vector of (65 × 11 + 6) or 721. The general framework of our stacking-based predictor is shown in Fig. 1.

3.2 Results

In this section, we first present the results obtained from best window size selection experiment and then provide the comparative results of individual classifiers obtained on best window size and, subsequently, report the performance of the stacked predictors on the benchmark dataset.

3.2.1 Window Selection

In this experiment, we searched for a suitable size of the sliding window (W) that determines the number of residues around a target residue, which could belong to SSS types of either β-hairpin or β-α-β. To select the optimal window size, we designed five different models using GBC classifier with five different window sizes: 1, 3, 7, 11, and 13. The models were trained and validated using threefold cross validation on the benchmark dataset. Figure 1 illustrates the overall accuracy obtained for all the window sizes for both β-α-β and β-hairpin SSS types while using GBC classifier.

From Fig. 2, we observed that the overall accuracy of the model increased drastically from window size 1 to 11, whereas, after window size 11, the increment is only after two decimal places. Thus, we selected window size of 11 as the best window size. All the methods used in the stacking were trained on window size 11 feature vector.

3.2.2 Analysis and Evaluation of Individual Machine Learning Algorithms

Here, we analyze the performance of five individual classifiers, LogReg, ET, KNN, RDF, and GBC. The performance metrics of the classifiers were obtained by performing threefold cross validation. The predicted annotations of every residue were compared against the actual annotations obtained from the PROMOTIF program.

From Table 3, we observe that the GBC gives an outstanding balanced as well as overall accuracy compared to other methods for the prediction of beta-alpha-beta SSS. The ET method resulted in the best recall or sensitivity of 0.705 and FNR or miss rate of 0.295. However, based on the rest of the performance measures, ET performed less accurately than the GBC. In addition, based on specificity, balanced accuracy, overall accuracy, FPR, precision, F1 score, and MCC, the GBC outperformed other methods. It is also evident that the RDF is the second-best method based on balanced

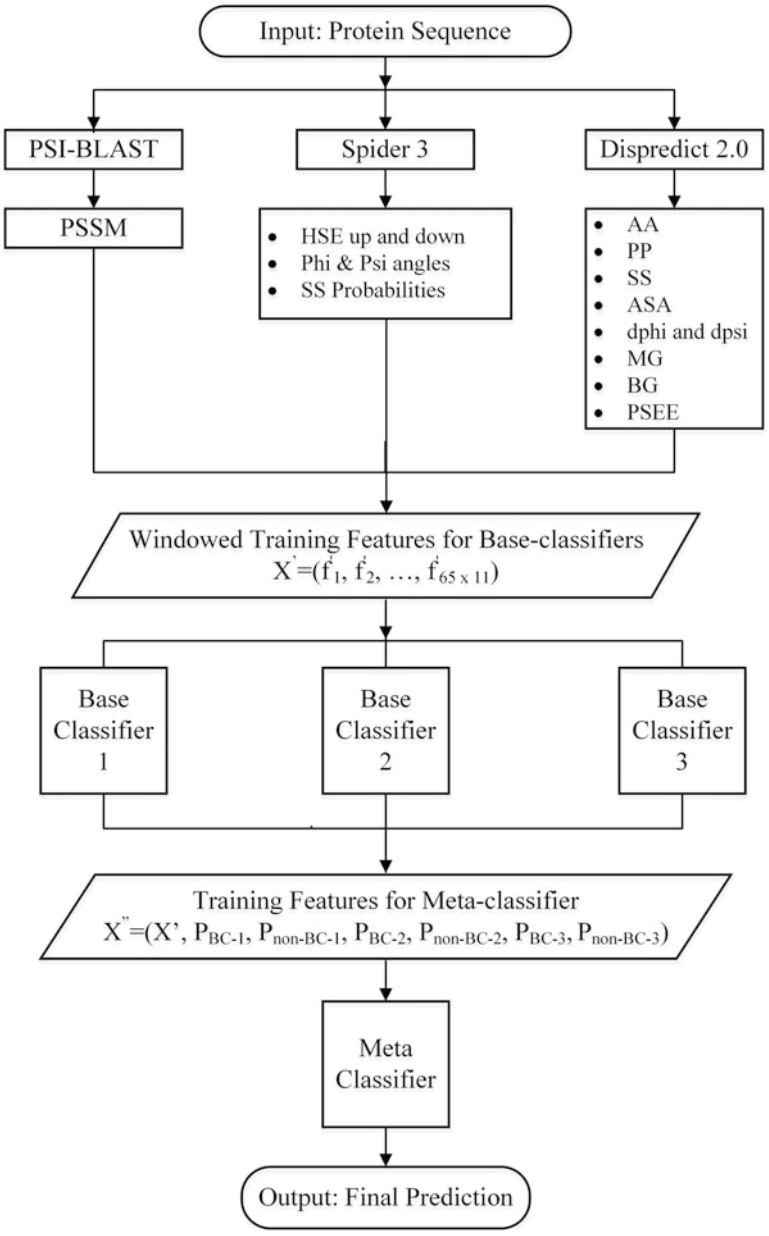

Fig. 1 Flowchart describing the stacking prediction framework. This framework was applied for both β-hairpin and β-α-β separately

and overall accuracy. Likewise, ET stands third, LogReg stands fourth, and KNN is the least performing method based on the balanced and overall accuracies.

From Table 4, GBC provides an outstanding balanced and overall accuracy compared to other methods for the prediction of β-hairpin SSS. The ET method gave best recall or sensitivity of 0.991 and FNR of 0.010. However, based on the rest of the

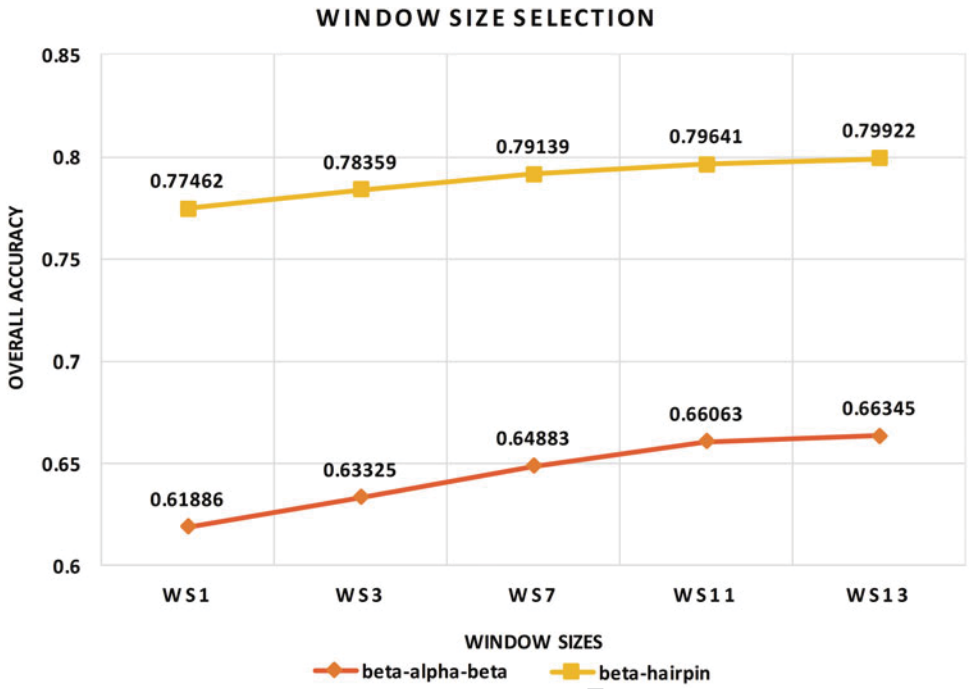

Fig. 2 Performance comparison of different window sizes using GBC method for both beta-alpha-beta and beta-hairpin SSS types. The accuracies of different window sizes are compared and used to decide the best window size

Table 3 t.1
Comparison of individual ML methods on predicting β-α-β SSS on the benchmark dataset using threefold cross validation and feature vector with window size 11

Metric/method	LogReg	ET	KNN	RDF	GBC	t.2
Sensitivity	0.679	**0.705**	0.603	0.704	0.695	t.3
Specificity	0.576	0.595	0.550	0.608	**0.626**	t.4
Balanced accuracy	0.627	0.650	0.577	0.656	**0.661**	t.5
Overall accuracy	0.627	0.650	0.577	0.656	**0.661**	t.6
FPR/fallout rate	0.424	0.405	0.450	0.392	**0.374**	t.7
FNR/miss rate	0.321	**0.295**	0.397	0.296	0.305	t.8
Precision	0.615	0.635	0.573	0.642	**0.650**	t.9
F1 score	0.645	0.668	0.588	0.672	**0.672**	t.10
MCC	0.256	0.302	0.154	0.313	**0.322**	t.11

Bold indicates best score t.12

t.1 **Table 4**
Comparison of individual ML methods on predicting β-hairpin SSS on the benchmark dataset using threefold cross validation and feature vector with window size 11

t.2 Metric/method	LogReg	ET	KNN	RDF	GBC
t.3 Sensitivity	0.990	**0.992**	0.947	0.989	0.972
t.4 Specificity	0.141	0.179	0.174	0.196	**0.259**
t.5 Balanced accuracy	0.566	0.585	0.560	0.592	**0.616**
t.6 Overall accuracy	0.781	0.791	0.756	0.793	**0.796**
t.7 FPR/fallout rate	0.859	0.821	0.826	0.804	**0.741**
t.8 FNR/miss rate	0.010	**0.008**	0.053	0.011	0.028
t.9 Precision	0.779	0.787	0.778	0.790	**0.800**
t.10 F1 score	0.872	0.877	0.854	**0.878**	0.878
t.11 MCC	0.282	0.335	0.190	0.344	**0.358**

t.12 **Bold** indicates best score

t.1 **Table 5**
Results of various stacked models for the prediction of β-α-β SSS

t.2 Method/metric	Sensitivity	Specificity	Balanced accuracy	Overall accuracy	Fallout rate	Miss rate	Precision	F1 score	MCC
t.3 SM1	0.709	0.609	0.659	**0.659**	0.391	0.291	0.644	0.675	0.319
t.4 SM2	0.641	0.583	0.612	0.612	0.417	0.359	0.606	0.623	0.224
t.5 SM3	0.693	0.620	0.657	0.657	0.380	0.307	0.646	0.669	0.314
t.6 SM4	0.707	0.610	0.658	0.658	0.391	0.293	0.644	0.674	0.318

t.7 **Bold** indicates best overall accuracy

performance metrics, ET performed less accurately than the GBC. In addition, except the performance metrics sensitivity, FNR, and F1 score, the GBC showed better performance than other methods based on the rest of the measurements. Furthermore, we can see that RDF is the second-best method based on balanced and overall accuracy. Likewise, ET stands third, LogReg stands fourth, and KNN is the least performing method based on the balanced and overall accuracies.

Next, we selected four of the ML techniques including GBC, RDF, KNN, and ET to use in our stacking approach. While one method was selected as the meta-learner, the others were selected as the base-learners. Table 5 shows the performance comparison of the combination of stacked models SM1, SM2, SM3, and SM4 used for the prediction of β-α-β SSSs using threefold cross validation on benchmark dataset. It can be seen from the table that the

Table 6 t.1
Results of various stacked models for the prediction of β-hairpin SSS

Method/ metric	Sensitivity	Specificity	Balanced accuracy	Overall accuracy	Fallout rate	Miss rate	Precision	F1 score	MCC	t.2
SM1	0.984	0.221	0.603	0.796	0.779	0.016	0.794	0.879	0.355	t.3
SM2	0.953	0.243	0.597	0.777	0.758	0.047	0.793	0.866	0.285	t.4
SM3	0.965	0.269	0.617	0.793	0.731	0.035	0.801	0.876	0.348	t.5
SM4	0.977	0.242	0.610	**0.796**	0.758	0.023	0.797	0.878	0.355	t.6

Bold indicates best overall accuracy t.7

Table 7 t.1
Comparison of overall accuracies obtained by stacked models and the individual methods for the prediction of β-α-β SSSs

Machine learning method	Individual	Meta-learner	t.2
ET	0.650	**0.659**	t.3
KNN	0.577	**0.612**	t.4
GBC	**0.661**	0.657	t.5
RDF	0.656	**0.658**	t.6

SM1, which consists of ET as meta-learner, provides the highest overall accuracy of 0.659 followed by SM4, SM3, and SM2, respectively. Furthermore, except SM2 the overall accuracies of all other stacked models are close to each other with differences after two decimal places.

Similarly, Table 6 shows the performance comparison of the combination of stacked models SM1, SM2, SM3, and SM4 used for the prediction of β-hairpin SSSs using threefold cross validation on benchmark dataset. It can be observed from Table 6 that the SM4, which consists of RDF as meta-learner, provides the highest overall accuracy of 0.796 followed by SM1, SM3, and SM2, respectively. Furthermore, except SM2 the overall accuracies of all other stacked models are close to each other with differences after two decimal places.

Next, in Table 7, we show the comparison of overall accuracies achieved by individual methods with the accuracies obtained while the respective methods were used as the meta-learner in the stacking framework for the prediction of β-α-β SSSs. It is evident from the table that, while the methods are used as the meta-learner in stacking, they yield better accuracy compared to while the respective methods are used independently. For example, while KNN was separately used for the prediction of β-α-β SSSs, we achieved an

t.1 **Table 8**
Comparison of overall accuracies obtained by stacked models and the individual methods for the prediction of β-hairpin SSSs

t.2 Machine learning method	Individual	Meta-learner
t.3 ET	0.791	**0.796**
t.4 KNN	0.756	**0.777**
t.5 GBC	**0.796**	0.793
t.6 RDF	0.793	**0.796**

overall accuracy of 0.577 or 57.67%. However, while KNN was used as the meta-learner, we achieved an overall accuracy of 0.612 or 61.2%, which is 6% higher than while KNN was used separately. This indicates that the stacking-based methods can be useful in predicting the supersecondary structures.

Similarly, in Table 8, we show the comparison of overall accuracies achieved by individual methods with the accuracies obtained while the respective methods were used as the meta-learner in the stacking framework for the prediction of β-hairpin SSSs. Table 8 also shows that stacking yielded better accuracy compared to individual methods for the prediction of β-hairpins except for stacking model in which GBC was used as a meta-learner. In case of GBC, the accuracy seems to slightly decrease. This decrease in accuracy is negligible however and occurs after the second decimal place. Moreover, the results for all other cases indicate that stacking resulted in better accuracy compared to the individual methods in this study.

4 Notes

1. The stacking-based machine learning predictors have been utilized in various bioinformatics applications [31, 54, 84, 86, 88]. Among others, Iqbal et al. recently proposed a stacking framework, called PBRpredict-Suite, to predict peptide-binding residues of receptor proteins from sequence [31]. They first compared six predictors to find the best predictor (SVM) and the two predictors least correlated with it (GBC and KNN) to use as base-learners. These base-learners' probability outputs were then used to train a logistic regression-based meta-learner. As reported, PBRpredict-Suite provides the best accuracy of 80.4% for the prediction of protein-peptide binding residues. Moreover, very recently, Mishra et al. applied stacking to develop a predictor, called StackDPPred, to predict DNA-binding proteins from sequence [54]. They combined

machine learning methods SVM, Logistic Regression, KNN, and RDF which are different from each other based on their underlying operating principle at the base layer. Next, to enrich the meta-learner (SVM), the original feature vector was combined with the base predictor probabilities. As reported, StackDPPred provided an accuracy of over 89% on benchmark dataset and an accuracy of 86.5 and 85.95% on two different independent test datasets, respectively.

2. Stacking-based predictors developed recently [31, 54] use SVM either as a base-learner or as the meta-learner, whereas in this application of supersecondary structure prediction, we were unable to use SVM because of the time constraint as it took longer to train. The SVM has been proven to be a very useful machine learning algorithm for various bioinformatics applications [32, 35, 36, 89]. Therefore, using SVM as either a base-learner or meta-learner could significantly improve the accuracy of supersecondary structure prediction. We propose to use SVM as a learner in our stacking application in our future work on supersecondary structure prediction. SVM is a fast learner; however, its optimization using grid search for RBF kernel parameters is often found impractically slow, especially for larger dataset.
3. Feature ranking and selection techniques have also been successfully applied in numerous applications including bioinformatics to improve the accuracy of the machine learning-based predictors [29, 32, 90]. Identifying relevant features and removing unimportant or redundant features can reduce computation time and improve results. In our future work, we will implement various feature ranking and selection techniques to improve the accuracy of our current supersecondary structure prediction approach.
4. Here, we present a review of some recently developed supersecondary structure prediction methods. In one of the recent work, Sun et al. developed a predictor which uses statistical approach and SVM to predict β-α-β motifs [2] from sequence but uses predicted secondary structure labels to predict β-α-β motifs. Similarly, Jia et al. proposed a predictor which also uses statistical approach and RDF to predict β-α-β motifs [41]. In this work, authors used DSSP and PROMOTIF software to obtain the secondary structure and supersecondary structure labels. Additionally, they performed a statistical analysis on β-α-β and non-β-α-β motifs and only selected the motifs that contain loop-helix-loop length from 10 to 26 amino acids. One major difference between these approaches and the method proposed in this study is that we predict any length of β-α-β motifs, while other methods only select the motifs that contain loop-helix-loop length from 10 to 26.

5. For intermediate steps, a more reliable SS prediction could improve the accuracy of SSS prediction: instead of utilizing single SS assignment method such as DSSP [61], the consensus-based SS label assignment would generate better SS assignment. For example, utilizing consensus of DSSP [61], STRIDE [91], and KASKI [92] methods, where final class label is generated based on majority vote from the assignment of these programs, could improve the SS assignment which could subsequently lead to better SSS prediction. This leverages the fact that each assignment software approaches this problem in different ways. Combining multiple methods should theoretically correct inaccuracies that can come from using single label generation method.
6. In real-world applications, computing resources and computing time are important factors in deciding which machine learning algorithm to use. It is common to weigh predictive accuracy against computational time to decide which method to use. In our application, we found that the SVM and LogReg were the two methods which took longer to run. Thus, due to time constraints, we discarded these methods for the prediction of supersecondary structures in this work. We look forward to implementing these methods in our future work on SSS prediction.
7. For the readers' convenience, links to the software necessary for feature and annotation collection are provided below:
   (a) Dispredict2: http://cs.uno.edu/~tamjid/Software.html
   (b) PSI-BLAST: https://blast.ncbi.nlm.nih.gov/Blast.cgi
   (c) Spider 3: http://sparks-lab.org/server/SPIDER3/
   (d) PROMOTIF: http://www.img.bio.uni-goettingen.de/ms-www/internal/manuals/promotif/promotif.html
8. While predictors for a few major class of SSSs have been proposed and tested, many have not been approached due to the limited scope of this article. Some interesting structures for further research are listed below:
   (a) α-Helix hairpins
   (b) Psi loop
   (c) Greek key
   (d) Rossmann motifs
9. For our future work, we intend to improve the accuracy of stacking-based prediction for supersecondary structures using various feature ranking and selection technique as well as including SVM, LogReg, and other useful machine learning methods in our stacking framework. We also intend to develop stacking-based machine learning predictors for coiled coil, Psi

loop, and other SSS types. Furthermore, we plan to develop a stacking-based software suit (tool) to predict multiple types of SSSs through a single complex framework. Finally, we will explore deep learning-based techniques for the prediction of supersecondary structures.

Acknowledgment

The authors gratefully acknowledge the Louisiana Board of Regents through the Board of Regents Support Fund LEQSF (2016-19)-RD-B-07.

References

1. Chen K, Kurgan L (2012) Computational prediction of secondary and supersecondary structures. In: Kister A (ed) Protein supersecondary structures, vol 932. Humana Press, Totowa, NJ
2. Sun L, Hu X, Li S, Jiang Z, Li K (2016) Prediction of complex super-secondary structure βαβ motifs based on combined features. Saudi J Biol Sci 23(1):66–71
3. Baker D, Sali A (2001) Protein structure prediction and structural genomics. Science 294 (5540):93–96
4. Skolnick J, Fetrow JS, Kolinski A (2000) Structural genomics and its importance for gene function analysis. Nat Biotechnol 18:283–287
5. Bhattacharya D, Cao R, Cheng J (2016) UniCon3D: de novo protein structure prediction using united-residue conformational search via stepwise, probabilistic sampling. Bioinformatics 32(18):2791–2799
6. Bhattacharya D, Cheng J (2013) i3Drefine software for protein 3D structure refinement and its assessment in CASP10. PLoS One 8(7): e69648
7. Bradley P, Misura KMS, Baker D (2005) Toward high-resolution de novo structure prediction for small proteins. Science 309 (5742):1868–1871
8. Cao R, Bhattacharya D, Adhikari B, Li J, Cheng J (2015) Large-scale model quality assessment for improving protein tertiary structure prediction. Bioinformatics 31(12): i116–i123
9. Jauch R, Yeo HC, Kolatkar PR, Clarke ND (2007) Assessment of CASP7 structure predictions for template free targets. Proteins 69 (S8):57–67
10. Klepeis JL, Wei Y, Hecht MH, Floudas CA (2005) Ab initio prediction of the three-dimensional structure of a de novo designed protein: a double-blind case study. Proteins 58(3):560–570
11. Liwo A, Khalili M, Scheraga HA (2005) Ab initio simulations of protein-folding pathways by molecular dynamics with the united-residue model of polypeptide chains. Proc Natl Acad Sci U S A 102(7):2362–2367
12. Wu S, Skolnick J, Zhang Y (2007) Ab initio modeling of small proteins by iterative TASSER simulations. BMC Biol 5:17
13. He X, Zhu Y, Epstein A, Mo Y (2018) Statistical variances of diffusional properties from ab initio molecular dynamics simulations. npj Comput Mater 4(1):18. https://doi.org/10.1038/s41524-018-0074-y
14. Magnan CN, Baldi P (2015) Molecular dynamics simulations advances and applications. Adv Appl Bioinforma Chem 8:37–47
15. Ginalski K, Pas J, Wyrwicz LS, Mv G, Bujnicki JM, Rychlewskia L (2003) ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. Nucleic Acids Res 31(13):3804–3807
16. Jones DT (1999) GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. J Mol Biol 287(4):797–815
17. Karplus K, Barrett C, Hughey R (1998) Hidden Markov models for detecting remote protein homologies. Bioinformatics 14 (10):846–856
18. Skolnick J, Kihara D, Zhang Y (2004) Development and large scale benchmark testing of the PROSPECTOR 3.0 threading algorithm. Proteins 56:502–518
19. Wu S, Zhang Y (2008) MUSTER: improving protein sequence profile-profile alignments by

using multiple sources of structure information. Proteins 72(2):547–556

20. Yang Y, Faraggi E, Zhao H, Zhou Y (2011) Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. Bioinformatics 27 (15):2076–2082
21. Yang J, Yan R, Roy A, Xu D, Poisson J, Zhang Y (2015) The I-TASSER Suite: protein structure and function prediction. Nat Methods 12:7–8
22. Faraggi E, Yang Y, Zhang S, Zhou Y (2010) Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. Structure 17(11):1515–1527
23. Szilágyi A, Skolnick J (2006) Efficient prediction of nucleic acid binding function from low-resolution protein Structures. J Mol Biol 358(3):922–933
24. Zhou H, Skolnick J (2007) Ab initio protein structure prediction using chunk-TASSER. Biophys J 93(5):1510–1518
25. Magnan CN, Baldi P (2014) SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. Bioinformatics 30(18):2592–2597
26. Heffernan R, Yang Y, Paliwal K, Zhou Y (2017) 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 33(18):2842–2849
27. Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, Wang J, Sattar A, Yang Y, Zhou Y (2015) Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. Sci Rep 5:11476
28. Faraggi E, Zhang T, Yang Y, Kurgan L, Zhou Y (2012) SPINE X: improving protein secondary structure prediction by multi-step learning coupled with prediction of solvent accessible surface area and backbone torsion angles. J Comput Chem 33(3):259–267
29. Zhang X, Liu S (2017) RBPPred: predicting RNA-binding proteins from sequence using SVM. Bioinformatics 33(6):854–862
30. Chowdhury SY, Shatabda S, Dehzangi A (2017) iDNAProt-ES: identification of DNA-binding proteins using evolutionary and structural features. Sci Rep 7:14938
31. Iqbal S, Hoque MT (2018) PBRpredict-Suite: a suite of models to predict peptide-recognition domain residues from protein sequence. Bioinformatics 34(19):3289–3299
32. Taherzadeh G, Zhou Y, Liew AW-C, Yang Y (2016) Sequence-based prediction of protein-carbohydrate binding sites using support vector machines. J Chem Inf Model 56 (10):2115–2122
33. Eickholt J, Cheng J (2012) Predicting protein residue–residue contacts using deep networks and boosting. Bioinformatics 28 (23):3066–3072
34. Iqbal S, Hoque MT (2015) DisPredict: a predictor of disordered protein using optimized RBF kernel. PLoS One 10(10):e0141551
35. Iqbal S, Hoque MT (2016) Estimation of position specific energy as a feature of protein residues from sequence alone for structural classification. PLoS One 11(9):e0161452
36. Iqbal S, Mishra A, Hoque T (2015) Improved prediction of accessible surface area results in efficient energy function application. J Theor Biol 380:380–391
37. Mizianty MJ, Kurgan L (2011) Sequence-based prediction of protein crystallization, purification and production propensity. Bioinformatics 27(13):i24–i33
38. Slabinski L, Jaroszewski L, Rychlewski L, Wilson IA, Lesley SA, Godzik A (2007) XtalPred: a web server for prediction of protein crystallizability. Bioinformatics 23(24):3403–3405
39. Jia S-C, Hu X-Z (2011) Using random forest algorithm to predict β-hairpin motifs. Protein Pept Lett 18(6):609–617
40. Hu X-Z, Li Q-Z, Wang C-L (2010) Recognition of β-hairpin motifs in proteins by using the composite vector. Amino Acids 38 (3):915–921
41. Sun L, Hu X (2013) Recognition of beta-alpha-beta motifs in proteins by using Random Forest algorithm. Paper presented at the sixth International Conference on Biomedical Engineering and Informatics, Hangzhou, China
42. Mahrenholz CC, Abfalter IG, Bodenhofer U, Volkmer R, Hochreiter S (2011) Complex networks govern coiled-coil oligomerization—predicting and profiling by means of a machine learning approach. Mol Cell Proteomics 10(5): M110.004994
43. Bartoli L, Fariselli P, Krogh A, Casadio R (2009) CCHMM_PROF: a HMM-based coiled-coil predictor with evolutionary information. Bioinformatics 25(21):2757–2763
44. Pellegrini-Calace M, Thornton JM (2005) Detecting DNA-binding helix-turn-helix structural motifs using sequence and structure

information. Nucleic Acids Res 33 (7):2129–2140

45. Dodd IB, Egan JB (1990) Improved detection of helix-turn-helix DNA-binding motifs in protein sequences. Nucleic Acids Res 18 (17):5019–5026
46. Ferrer-Costa C, Shanahan HP, Jones S, Thornton JM (2005) HTHquery: a method for detecting DNA-binding proteins with a helix-turn-helix structural motif. Bioinformatics 21 (18):3679–3680
47. Kumar M, Bhasin M, Natt NK, Raghava GPS (2005) BhairPred: prediction of β-hairpins in a protein from multiple alignment information using ANN and SVM techniques. Nucleic Acids Res 33(Web Server issue):W154–W159
48. Sun ZR, Cui Y, Ling LJ, Guo Q, Chen RS (1998) Molecular dynamics simulation of protein folding with supersecondary structure constraints. J Protein Chem 17(8):765–769
49. Szappanos B, Süveges D, Nyitray L, Perczel A, Gáspári Z (2010) Folded-unfolded cross-predictions and protein evolution: the case study of coiled-coils. FEBS Lett 584(8):1623–1627
50. O'Donnell CW, Waldispühl J, Lis M, Halfmann R, Devadas S, Lindquist S, Berger B (2011) A method for probing the mutational landscape of amyloid structure. Bioinformatics 27(13):i34–i42
51. Rackham OJL, Madera M, Armstrong CT, Vincent TL, Woolfson DN, Gough J (2010) The evolution and structure prediction of coiled coils across all genomes. J Mol Biol 403(3):480–493
52. Gerstein M, Hegyi H (1998) Comparing genomes in terms of protein structure: surveys of a finite parts list. FEMS Microbiol Rev 22 (4):277–304
53. Reddy CC, Shameer K, Offmann BO, Sowdhamini R (2008) PURE: a webserver for the prediction of domains in unassigned regions in proteins. BMC Bioinformatics 9:281
54. Mishra A, Pokhrel P, Hoque MT (2018) StackDPPred: a stacking based prediction of DNA-binding protein from sequence. http://cs.uno.edu/~tamjid/TechReport/StackDPPred_TR2018_2.pdf
55. Flot M, Mishra A, Kuchi AS, Hoque MT (2018) Benchmark data for supersecondary structure prediction only from sequence. University of New Orleans. http://cs.uno.edu/~tamjid/Software/StackSSSPred/code_data.zip. Accessed June 2018
56. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. Nucleic Acids Res 28(1):235–242
57. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421
58. Blundell TL, Sibanda BL, Sternberg MJE, Thornton JM (1987) Knowledge-based prediction of protein structures and the design of novel molecules. Nature 326:347–352
59. Wierenga RK, Terpstra P, Hol WG (1986) Prediction of the occurrence of the ADP-binding βαβ-fold in proteins, using an amino acid sequence fingerprint. J Mol Biol 187 (1):101–107
60. Hutchinson EG, Thornton JM (1996) PROMOTIF—a program to identify and analyze structural motifs in proteins. Protein Sci 5 (2):212–220
61. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22:2577–2637
62. Meiler J, Müller M, Zeidler A, Schmäschke F (2001) Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. J Mol Model 7:360–369
63. Biswas AK, Noman N, Sikder AR (2010) Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information. BMC Bioinformatics 11:273
64. Islam N, Iqbal S, Katebi AR, Hoque MT (2016) A balanced secondary structure predictor. J Theor Biol 389:60–71
65. Kumar M, Gromiha MM, Raghava GP (2007) Identification of DNA-binding proteins using support vector machines and evolutionary profiles. BMC Bioinformatics 8:463
66. Verma R, Varshney GC, Raghava GPS (2010) Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile. Amino Acids 39(1):101–110
67. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215:403–410
68. Paliwal KK, Sharma A, Lyons J, Dehzangi A (2014) A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition. IEEE Trans Nanobioscience 13(1):44–50
69. Sharma A, Lyons J, Dehzangi A, Paliwal KK (2013) A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. J Theor Biol 320:41–46
70. Zhang T, Faraggi E, Zhou Y (2010) Fluctuations of backbone torsion angles obtained from

NMR-determined structures and their prediction. Proteins 78:3353–3362

71. London N, Movshovitz-Attias D, Schueler-Furman O (2010) The structural basis of peptide-protein binding strategies. Structure 18(2):188–199
72. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
73. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46:175–185
74. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63 (1):3–42
75. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38 (4):367–378. https://doi.org/10.1016/S0167-9473(01)00065-2
76. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, Springer series in statics, 2nd edn. Springer, New York
77. Freedma DA (2009) Statistical models: theory and practice. Cambridge University Press, Cambridge
78. Ho TK (1995) Random decision forests. Paper presented at the Document Analysis and Recognition, 1995. Proceedings of the Third International Conference, Montreal, Quebec, Canada
79. Duda RO, Hart PE, Stork DG (2000) Pattern classification. Wiley, Hoboken, NJ
80. Bishop C (2009) Pattern recognition and machine learning. Information science and statistics. Springer, New York
81. Wolpert DH (1992) Stacked generalization. Neural Netw 5(2):241–259
82. Frank E, Hall M, Trigg L, Holmes G, Witten IH (2004) Data mining in bioinformatics using Weka. Bioinformatics 20(15):2479–2481
83. Ginsburga GS, McCarthyb JJ (2001) Personalized medicine: revolutionizing drug discovery and patient care. Trends Biotechnol 19 (12):491–496
84. Nagi S, Bhattacharyya DK (2013) Classification of microarray cancer data using ensemble approach. Netw Model Anal Health Inform Bioinform 2(3):159–173
85. Hu Q, Merchante C, Stepanova AN, Alonso JM, Heber S (2015) A stacking-based approach to identify translated upstream open reading frames in Arabidopsis Thaliana. Paper presented at the International Symposium on Bioinformatics Research and Applications
86. Verma A, Mehta S (2017) A comparative study of ensemble learning methods for classification in bioinformatics. Paper presented at the seventh International Conference on Cloud Computing, Data Science & Engineering—Confluence, Noida, India
87. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evolut Comput 1(1):67–82
88. Frank E, Hall M, Trigg L, Holmes G, Written IH (2004) Data mining in bioinformatics using Weka. Bioinformatics 20:2479–2481
89. Guruge I, Taherzadeh G, Zhan J, Zhou Y, Yang Y (2018) B-factor profile prediction for RNA flexibility using support vector machines. J Comput Chem 39:407–411
90. Anne C, Mishra A, Hoque MT, Tu S (2018) Multiclass patent document classification. Artif Intell Res 7(1):1
91. Heinig M, Frishman D (2004) STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. Nucleic Acids Res 32(Web Server issue):W500–W502
92. Martin J, Letellier G, Marin A, Taly J-F, AGD B, Gibrat J-F (2005) Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. BMC Struct Biol 5:17