DF-Phos: Prediction of Protein phosphorylation Sites by Deep Forest

Zeynab Zahiri
University of Birjand
Nasser Mehrshad
nmehrshad@birjand.ac.ir
University of Birjand
Maliheh Mehrshad
Swedish University of Agricultural Sciences

Research Article

Keywords: protein phosphorylation, prediction, Deep Forest, machine learning, feature extraction

Posted Date: February 16th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2462218/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at The Journal of Biochemistry on December 28th, 2023. See the published version at https://doi.org/10.1093/jb/mvad116.

Abstract

Background

Phosphorylation is the most important and studied post-translational modification (PTM), which plays a crucial role in protein function studies and experimental design. Many significant studies have been performed to predict phosphorylation sites using various machine-learning methods. Recently, several studies have claimed that deep learning-based methods are the best way to predict the phosphorylation sites because deep learning as an advanced machine learning method can automatically detect complex representations of phosphorylation patterns from raw sequences and thus offers a powerful tool to improve phosphorylation site prediction.

Results

In this study, we report DF-Phos, a new phosphosite predictor based on the deep forest to predict phosphorylation sites. In DF-Phos, the feature vector taken from the CkSAApair method is as input for a deep forest framework for predicting phosphorylation sites. The results of 10-fold cross-validation show that the deep forest method has the highest performance among other available methods.

Conclusions

We implemented a python program of DF-Phos, which is freely available for non-commercial use at https://github.com/zahiriz/DF-Phos Moreover, users can use it for various PTM predictions.

Background

Phosphorylation is the most important post-translational modification[1] and it is a key mechanism in many biological processes, including DNA repair, transcriptional regulation, environmental stress response, apoptosis, metabolism, immune responses, signal transmission, cellular differentiation[2]. In eukaryotes, phosphorylation occurs in serine(S), threonine (T), and tyrosine (Y) residues, like eukaryotes in Prokaryotes, phosphorylation mainly occurs on S, T, and Y; but in prokaryotic, phosphorylation also occurs on additional types of amino acids, including arginine (R), histidine (H), cysteine(C) and aspartic acid (D) residues.

In the last few decades, phosphorylation site prediction research has attracted much attention, and the development of accurate phosphorylation site prediction methods has become very important. Existing methods can be generally divided into two categories: biological experimental methods, which are expensive and time-consuming, and computational methods, which are fast speed and low cost. moreover, identification based on experimental methods is labor-intensive and requires specialized equipment and technical knowledge. In this regard, phosphosite prediction algorithms are becoming popular and used to predict the list of possible phosphorylation sites in a protein of interest, then experimental methods are applied in verifying the phosphorylation sites that were predicted. So far, many predictors have been introduced to predict PTM sites (such as phosphorylation and methylation, etc.) [3], [4]. But it seems that the specific phosphorylation predictors provide more accurate results.

Computational Phosphorylation site prediction tools are divided into three categories, general (non-kinase specific) site prediction, kinase-specific site prediction, and global prediction, while general tools predict sites that can be phosphorylated and kinase-specific tools predict sites that can be phosphorylated by a specific kinase and also a global Prediction predict a General and Kinase-specific Phosphorylation Sites [5].

Non-kinase-specific tools may be able to predict phosphosites for which the associated kinase is unknown or the number of new substrate sequences of the associated kinase is few [2]. Moreover, by the recent advances in sequencing technology, many genomes of non-model organisms have been sequenced, and more kinases in those reconstructed genomes have been discovered, some of which have no sufficient substrate information to train the kinase-specific prediction algorithms. Thus, there is an increased interest in developing non-kinase-specific tools for a wider variety of species and high specificity for whole-genome annotation [6].

Until now, a few general phosphorylation site prediction models have been proposed, most of the existing methods are different in choosing the machine learning algorithms and feature engineering extraction, which have been used to capture the complex and definite patterns surrounding the phosphorylated residues for phosphorylation site prediction. The most widely used machine learning methods in general prediction tools include artificial neural networks (ANNs), support vector machines (SVMs), linear regression (LR), and random forest (RF). For instance, NetPhos uses neural networks to identify phosphorylation sites [7], while DISPHOS uses the amino acid frequency and disorder information to train an LR model for predicting the phosphorylation sites [8], Biswas et al. in PPRED combine the evolutionary information of the proteins with the SVMs to predict phosphorylation sites [9]. Musite, integrates three sets of parameters, including K nearest neighbor scores, protein disorder scorers, and amino acid frequencies, as features to train an SVM [4] and Phospho- SVM, which is one the most recent prediction tools based on SVMs, combines eight different sequence-level scoring functions using SVMs[6]. RF algorithms can provide insights into the relative importance of each feature; thus, RF classifiers have been applied to Various bioinformatics problems [10]. For example, in RF-Phos the random forest with sequence and structural features has been used to predict the general phosphorylation site [11].

A major recent advance in machine learning is introducing deep artificial neural networks. Deep learning is now one of the most effective fields in machine learning and has made breakthroughs in image and speech recognition, natural language processing, and most recently, computational biology [12]. Compared to traditional machine learning techniques, a deep neural network takes the raw data at the lowest (input) layer and automatically discovers the complex representations, and captures the high-level abstraction adaptively from the training data for classification. Thus, the application of deep learning for biological sequence analysis is growing. For example, DeepBind uses a convolutional neural network (CNN) for predicting sequence specificities of DNA- and RNA-binding proteins [13], MusiteDeep chooses a CNN with a two-dimensional attention mechanism for site prediction [14], DeepNitro uses a multi-layer deep neural network to predict nitration and nitrosylation sites [15], DeepPhos improved upon the performance of MusiteDeep, utilizing a multi-layer CNN architecture [16], DeepPSP extracts both local and global features from protein sequences with two parallel modules [17].

In addition to the deep learning methods that have been introduced based on neural networks, deep networks based on other learning methods have also been introduced, including deep forests [18]. Deep Forest is a classification method that consists of several layers and each layer contains several random forests. This method has fewer parameters than conventional deep learning methods, and the complexity of the model can be automatically identified through the data. Another advantage of this method is that it can produce good results without using backpropagation [19].

In this study, we focused on developing a new phosphorylation site predictor by seeking a more informative encoding scheme and the best machine-learning method. After our preliminary assessment of 37 different encoding schemes for training one of the 9 machine learning methods (The architecture of our phosphorylation predictor is shown in Fig. 1), we found that the composition of k-spaced amino acid pairs (CKSAAP) and the deep forest is suitable for phosphorylation prediction. Then we present DF-Phos, a general protein phosphorylation site predictor that uses a deep Forest and CkSAApair feature extraction method to predict the phosphorylation sites using protein sequence information. We collect human and muse data from two databases the dbpaf [20], and the P.ELM[21]. To avoid a biased classifier, the training set was created with a positive-to-negative ratio of 1:1. The optimal window length was determined using 10-fold cross-validation and independent test methods. Then we evaluated our predictor using a 10-fold cross-validation procedure and compared this method with several phosphosite predictors.

Results And Discussion

To evaluate the performance of phosphorylation site prediction, several well-known performance measures were used, including sensitivity (SN), specificity (SP), accuracy (ACC), Precision (PR), F1 measure, Mattews correlation coefficient (MCC), area under the receiver operating characteristic curve (AUC), and they are defined as follows:

$$SN = \frac{TP}{TP + FN}$$

1

$$SP = \frac{TN}{TN + FP}$$

2

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

3

$$PR = \frac{TP}{TP + FP}$$

4

$$F1 - measure = 2 \times \frac{Pre \times Sn}{Pre + Sn}$$

5

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$$

6

where, TP and TN are the numbers of positive and negative phosphorylation sites that were correctly predicted by the model, respectively. FN and FP indicate the numbers of positive and negative phosphorylation sites that were wrongly classified as negative and positive, respectively. Therefore, SN refers to the percentage of positive phosphorylation sites correctly classified by a predictor. PR represents the ratio of true positive samples produced by the predictor. MCC and F1-measure are two types of combined classification performance measures that take all four basic parameters (TP, TN, FP, and FN) into account.

37 feature vectors and 9 classification methods were studied in this research which contained a total of 333 different results. To achieve the best result, 10-fold cross-validation was used and the assessment of the results was done using ACC and AUC. As can be seen in Figs. 2 and 3, CkSAApair as a feature extraction method and Deep Forest as a

machine learning method have obtained the highest accuracy. The CKSAAP encoding strategy calculates the composition of k-spaced amino acid pairs. In other words, it computes all amino acid pairs frequency with k spaces, which has been successfully employed for the prediction of ubiquitination sites[22] and phosphorylation sites [23]–[25] and Deep Forest has been successfully employed for Detecting Blood Methylation Signatures [18] and prediction of RNA velocity [26].

It should be noted that the data selected in the test part and the training part are considered the same for all the above methods. More details about these two methods are given below.

This section first evaluates our method and finds the best window length. The training process of this model was done by two databases P.ELM and dbPAF separately and with Different window lengths 21, 25, 29, 33, 37, 41 and its results include the ACC, PR, SN and MCC, and SP of the DF-Phos using 10-fold cross-validation and the independent test is shown in Table 1. The obtained results show that the best results occurred on window length 37 and the independent test method.

Table 2 shows the accuracy results of the introduced model on human and mouse species. As can be seen, Mus-muscles data have better results than Homo-sapiens data. In addition, by calculating the performance for each of the S, T, and Y residues, it can be seen that the prediction accuracy of S and T residues is higher than Y, and this issue can be seen in both Homo-sapiens and Mus-muscles species.

Table 1
The results obtained on the P.ELM and dbPAF databases with different window lengths (WL).

		Cross Validation						**Independent Test**				
	P.ELM	WL	ACC	PR	SN	MCC	SP	ACC	PR	SN	MCC	SP
database		21	0.73	0.73	0.73	0.46	0.62	0.73	0.73	0.73	0.46	0.63
		25	0.75	0.75	0.75	0.49	0.67	0.75	0.75	0.75	0.50	0.64
		29	0.770	0.770	0.770	0.52	0.70	0.79	0.79	0.79	0.57	0.720
		33	0.774	0.774	0.774	0.53	0.71	0.776	0.776	0.776	0.54	0.729
		37	**0.783**	**0.783**	**0.783**	**0.55**	**0.724**	**0.80**	**0.80**	**0.80**	**0.59**	**0.73**
		41	0.782	0.782	0.782	0.58	0.73	0.79	0.79	0.79	0.62	0.75
	dbPAF	21	0.72	0.673	0.673	0.45	0.71	0.673	0.673	0.673	0.45	0.71
		25	0.69	0.69	0.69	0.5	0.74	0.69	0.69	0.65	0.51	0.741
		29	0.74	0.740	0.740	0.47	0.78	0.74	0.74	0.74	0.56	0.71
		33	0.742	0.742	0.742	0.48	0.79	0.746	0.746	0.746	0.48	0.79
		37	**0.75**	**0.75**	**0.75**	**0.51**	**0.79**	**0.755**	**0.755**	**0.755**	**0.53**	**0.80**
		41	0.75	0.75	0.75	0.52	0.8	0.74	0.74	0.74	0.55	0.81

In other articles, it has been mentioned that the prediction accuracy of S, and T is higher than that of Y [27], [4], [14], [17].

Table 2
ACC results obtained by separating amino acids in the ELM database.

Species	S	T	Y	Total
Mus-muscles	0.86	0.84	0.78	0.83
Homo-sapiens	0.78	0.82	0.73	0.75
All data	0.81	0.81	0.74	0.79

The performance of the DF-Phos method was compared with eight of the best methods available for predicting the location of other phosphorylation sites, including two traditional machine-learning methods (Netphos3.0, Musite) and six deep-learning methods (PPSP, MusiteDeep, DeepPhos, DeepPhos71, PhosIDNseq, PhosIDN). Table 3 gives the value of SN, ACC, MCC, PR, and F1 that were reported for these methods and the result of DF-Phos with our database. As can be seen, DF-Phos has better results in SN, ACC, MCC, and F1 indices than other methods, while PR does not have good results. However, DeepPhos, which has good results in PR, does not have good Sensitivity.

Table 3
Comparing the results of previous methods and the current method.

Model	SN	ACC	MCC	PR	F1
PPSP	27.8	58.9	22.55	73.15	40.15
Netphos3.0	19.85	54.9	13.4	65.6	30.3
Musite	16	55.7	18.4	76.7	26.35
MusiteDeep	34.7	62.35	29.2	76.5	47.25
DeepPhos	47.8	64.5	41.4	**87.7**	48.77
DeepPhos-71	52.1	71.5	33.3	83.5	64.3
PhosIDNSeq	40.25	65.15	34.65	79.5	53.05
PhosIDN	52.1	71.1	45.5	83.9	64.3
DF-Phos	**78**	**78**	**51**	76	**74**

Note: best-performing method in bold

To more accurately compare the performance of the proposed predictor, we divided the sequences extracted from the P.Elm database with a window length of 37 into two training and testing groups, and then the DeepPhos and DF-Phos methods, were trained using the training data and the evaluation results of models using test data is given in Table 4. The results show that DF-Phos performs is better than the DeepPhos method in all parameters.

Table 4
The value of evaluation results for the DeepPhos and DeepForest methods.

Model	MCC	ACC	PR	SP	SN
DF-Phos	**0.74**	**0.74**	**0.742**	**0.742**	**0.743**
DeepPhos	0.653	0.69	0.682	0.632	0.673

According to the results, in all cases, DF-Phos exhibit the highest performance among all the methods evaluated. For instance, DF-Phos compared to DeepPhos was able to improve by 5–10% in parameter evaluation.

Conclusions

However, it's very easy to predict the protein phosphorylation sites, but creating a highly accurate prediction is difficult. Thus, many computational methods have been used to predict phosphorylation sites with higher accuracy, so far. In this study, 9 classification methods and 37 different feature vectors were used to predict phosphorylation sites. Results show that the deep forest method with the CkSAApair feature extraction method, had a much better performance (assessing by Accuracy, and AUC) compared to other available methods. Then, a new phosphorylation site predictor named DF-Phos was developed to predict protein phosphorylation sites using only the primary sequence information. The highlight of DF-Phos was to utilize the CkSAApair method as the encoding scheme, and then the deep forest architecture was used as the predictor. The performance of DF-Phos was measured with a sensitivity of 78%, a precision of 76%, and an accuracy of 78% for all data. Experimental results obtained from 10-fold cross-validation suggested that DF-Phos is a powerful tool to predict the phosphorylation site for both Homo sapiens and Mus-muscles species.

The approach presented in this paper provides an efficient way to identify phosphorylation sites in a given protein primary sequence and deep forest approach that would be a piece of valuable information for the molecular biologists working on protein phosphorylation sites and for bioinformaticians developing generalized prediction systems for the post-translational modifications like glycosylation, nitration, and phosphorylation.

Methods

To collect the training data set, we used two databases dbPAF [20] and Phospho.ELM [21]. dbPAF contains known phosphorylation sites in Homo sapiens, Rattus norvegicus, Mus musculus, Drosophila melanogaster, Caenorhabditis elegans, Schizosaccharomyces pombe, and Saccharomyces cerevisiae[20] that are integrated from nine public databases in eukaryotes, including dbPTM, PHOSIDA, Phospho.ELM, PhosphositePlus, PhosphoPep, PhosphoGRID, SysPTM, HPRD, and Uniprot, with manual curation of the literature. Phospho.ELM contains experimentally verified phosphorylation sites manually curated from the literature and is developed as part of the ELM (Eukaryotic Linear Motif) resource [28]. In this study, we focused on human and mouse phosphorylation site prediction and, consequently, extracted all Homo sapiens and Mus musculus phosphorylation datasets from the above datasets then randomly select the same number of phosphorylation sites from each species and combined all phosphorylation on S, T, and Y as the positive instances. The same amino acids without annotated phosphorylation sites from the same proteins were considered negative instances. These sites have been used to construct a sequence of length 33 in which the positive or negative site is located at the central position. Due to the larger number of negative sites than positive sites, it is very likely to train a biased classifier that would lead to predicting most of the unknown sites as negative [9]. Thus, to overcome the problem, we randomly selected negative sites to match the number of positive examples [29],[16]. Finally, the data obtained from human and mouse species were combined, and to reduce sequence redundancy in the extracted datasets and avoid potential bias in model training, the redundant sequences were removed using the CD-HIT tool [30] with a similarity threshold of 90%. Table 5 shows the number of final data after applying CD-HIT for different species. These data are used as the final data for the training phase of the classification.

Table 5
phosphorylation data collected in this study.

		Homo sapiens				Mus musculus			
		Total	S	Y	T	Total	S	Y	T
Database	Phospho.ELM	7482	2505	2343	2630	5518	2395	1526	1597
	dbpaf	20476	6855	6412	7209	15856	5232	4915	5709

Feature extraction

The brief name of each feature extraction method for sequences and its description are listed in Table 6. All of these 37 feature vectors were extracted by the ftrCool library [31].

Table 6
Feature extraction methods.

Row	Brief Feature name	Feature Description
1	AAutoCor	Amino Acid Autocorrelation-Autocovariance
2	CkSAApair	Composition of k-spaced Amino Acids pairs
3	CkSGAApair	Composition of k-Spaced Grouped Amino Acids pairs
4	CTD	Composition Transition Distribution
5	CTDC	Composition Transition Distribution
6	CTDD	CTD Distribution
7	DDE	Dipeptide Deviation from Expected Mean value
8	EAAComposition	Enhanced Amino Acid Composition
9	EGAAComposition	Enhanced Grouped Amino Acid Composition
10	PseKRAAC_T13	Pseudo K_tuple Reduced Amino Acid Composition Type_13
11	PseKRAAC_T14	Pseudo K_tuple Reduced Amino Acid Composition Type_14
12	PseKRAAC_T15	Pseudo K_tuple Reduced Amino Acid Composition Type_15
13	PseKRAAC_T16	Pseudo K_tuple Reduced Amino Acid Composition Type_16
14	PseKRAAC_T3A	Pseudo K_tuple Reduced Amino Acid Composition Type_3A
15	PseKRAAC_T3B	Pseudo K_tuple Reduced Amino Acid Composition Type_3B
16	PseKRAAC_T4	Pseudo K_tuple Reduced Amino Acid Composition Type_4
17	PseKRAAC_T5	Pseudo K_tuple Reduced Amino Acid Composition Type_5
18	PseKRAAC_T6A	Pseudo K_tuple Reduced Amino Acid Composition Type_6A
19	GrpDDE	Group Dipeptide Deviation from Expected Mean
20	kGAAComposition	k Grouped Amino Acid Composition
21	LocalPoSpKaaF	Local Position Specific k Amino Acids Frequency
22	PseKRAAC_T1	Pseudo K_tuple Reduced Amino Acid Composition Type_1
23	PseKRAAC_T10	Pseudo K_tuple Reduced Amino Acid Composition Type_10
24	PseKRAAC_T11	Pseudo K_tuple Reduced Amino Acid Composition Type_11
25	QSOrder	Quasi Sequence Order
26	SAAC	Split Amino Acid Composition
27	PseKRAAC_T6B	Pseudo K_tuple Reduced Amino Acid Composition Type_6B
28	SGAAC	Split Group Amino Acid Composition
29	SOCNumber	Sequence Order Coupling Number
30	PseKRAAC_T12	Pseudo K_tuple Reduced Amino Acid Composition Type_12

Row	Brief Feature name	Feature Description
31	PseKRAAC_T9	Pseudo K_tuple Reduced Amino Acid Composition Type_9
32	ExpectedValueGKmerAA	Expected Value for Grouped K-mer Amino Acid
33	ExpectedValueKmerAA	Expected Value for K-mer Amino Acid
34	PseKRAAC_T7	Pseudo K_tuple Reduced Amino Acid Composition Type_7
35	PseKRAAC_T8	Pseudo K_tuple Reduced Amino Acid Composition Type_8
36	ExpectedValueGAA	Expected Value for each Amino Acid
37	ExpectedValueAA	Expected Value for each Amino Acid

Machine Learning Methods

In this study as shown in Fig. 1, for each feature extraction method, the following classification methods were used and Best Accuracy and AUC were determined; Ada Boost (ADA) [32], K nearest neighbor (KNN) [33], Naive Bayes (NB), Support Vector Machine (SVM) [34], Random Forest (RF)[35], multi-Layer perceptron (MLP) [36], Logistic Regression (LR) [37], Decision Tree (DT)[38] and Deep Forest [18].

ADA

Ada boost is one of the learning methods that used a mixture of classifiers, for better and more accurate prediction. Each learner method creates an output (a class) for each sample. Then the linear sum of these learners is selected to minimize the classifier error.

KNN

KNN is one of the simplest learning algorithms. The basic idea of this algorithm is to calculate the distance of an object to the k nearest neighbors and then find the first k-nearest samples and determine the category of the new instance.

NB

This classifier is based on Bayes' theorem and independence assumptions between the data for a given class. This assumption can highly reduce the computational cost. NB method has been used for PTM prediction [39].

SVM

SVM is one of the most applicable machine learning methods for binary problems and has high accuracy and also high performance. This method utilizes an optimized hyperplane to distinguish classes and it is widely used for the prediction of PTM [25], [40] and phosphorylation [41].

MLP

Multi-layer perceptron (MLP) is a type of artificial neural network that consists of three types of layers, an input to receive the input signal to be processed, an output layer that the result of prediction and classification that could be extracted from this layer, and hidden layers that are placed in between the input and output layer are the true computational engine of the MLP. This method can solve problems that are not linearly separable. One of the important applications of MLP is pattern classification and prediction [42].

LR

Logistic Regression is a “Supervised machine learning” algorithm that can be used to model the probability of a specified class. It is usually used for Binary classification problems. That means Logistic regression is usually used when the outcome is binary. LR has shown good results in phosphorylation predictors [43], [44].

DT

DT is a model that gives interpretable decision rules. It creates a classification model based on the IF-THEN structure to discover the relation between feature vectors and classes [45].

RF

RF is a collection of decision trees. Each decision tree is trained by some randomly selected features and samples from the original dataset. For a test sample, the majority of votes are used, to calculate the predicted value.

DF

The deep forest is a deep model based on decision trees. Compared with deep neural networks, the training process of deep forest does not depend on backpropagation and gradient adjustment and it has fewer hyper-parameters.

Various architectures have been introduced for the deep forest, one of the newest architectures is called gcForest. In this method, the layers are placed next to each other in a cascade manner. The layered structure created in gcForest gives the network the ability to learn more difficult patterns. In other words, in this method, each layer receives feature info from its previous layer, similar to conventional deep learning methods. Figure 4 shows this issue.

Abbreviations

PTM	Post-Translational Modification
S	Serine
T	Threonine
Y	Tyrosine
CkSAApair	Composition of k-spaced Amino Acids pairs
ADA	Ada boost
KNN	K-Nearest Neighbors
NB	Naive Bayes
SVM	Support Vector Machine
MLP	Multi-Layer Perceptron
LR	Logistic Regression
DT	Decision Tree
RF	Random Forest
DF	Deep Forest

SN	Sensitivity
SP	Specificity
ACC	Accuracy
PR	Precision
MCC	Mattews Correlation Coefficient
AUC	Area Under the receiver operating Characteristic curve
F1	F1 measure

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Data Availability statement

DF-Phos with benchmark dataset is available at https://github.com/zahiriz/DF-Phos and is free for non-commercial academic use.

Project name: DF-Phos

Project home page: https://github.com/zahiriz/DF-Phos

Operating system(s): macOS , Windows

Programming language: python

Other requirements: python 3 or higher, R (programming language)

Conflict of interest

The authors declare that they have no competing interests.

Funding

Not applicable.

Acknowledgments

Not applicable.

Authors' contributions

Zeynab Zahiri designed and implemented the algorithm and wrote the manuscript text. Nasser Mehrshad supervised the work and prepared the manuscript tables. Maliheh Mehrshad advised the work and helped with data analysis and drawing the manuscript figures. All authors have reviewed and approved the submitted manuscript.

References

1. Trost B, Kusalik A: **Computational phosphorylation site prediction in plants using random forests and organism-specific instance weights**. Bioinformatics, 2013, 29: 686–694.
2. Trost B, Kusalik A: **Computational prediction of eukaryotic phosphorylation sites**. Bioinformatics, 2011, 27, 2927–2935.
3. Basu S, Plewczynski D: **AMS 3.0: prediction of post-translational modifications**. BMC Bioinformatics, 2010, 11, 210.
4. Gao J, Thelen J J, Dunker A K, Xu D: **Musite, a tool for global prediction of general and kinase-specific phosphorylation sites**. Mol Cell Proteomics, 2010, 9, 2586–2600.
5. Jamal S, Ali W, Nagpal P, Grover A, Grover S: **Predicting phosphorylation sites using machine learning by integrating the sequence, structure, and functional information of proteins**. J Transl Med, 2021, 19, 218.
6. Dou Y, Yao B, Zhang C: **PhosphoSVM: Prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine**. Amino Acids, 2014, 46, 1459–1469.
7. Blom N, Gammeltoft S, Brunak S: **Sequence and structure-based prediction of eukaryotic protein phosphorylation sites**. J Mol Biol, 1999, 294, 1351–62.
8. Iakoucheva LM, Radivojac P, Brown C J, Connor T R O, Sikes J G, Obradovic Z, Dunker A K: **The importance of intrinsic disorder for protein phosphorylation**. *Nucleic Acids Res*, 2004, 32, 1037–1049.
9. Biswas A K, Noman N, Sikder A R: **Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information**. BMC Bioinformatics, 2010, 11, 273.
10. Breiman L: **Random Forests**. Mach Learn, 2001, 455–32.
11. Jones A, Ismail H, Kim J H, Newman R, Dukka B K: **RF-Phos**: **Random forest-based prediction of phosphorylation sites**. in 2015 *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Nov. 2015, IEEE; 2015: 135–140.
12. Angermueller C, Pärnamaa T, Parts L, Stegle O: **Deep learning for computational biology**. 2016: 1–16.
13. Alipanahi B, Delong A, Weirauch M T, Frey B J, **Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning**. Nat Biotechnol, 2015, 1–9.
14. Wang D, Zeng S, Xu C, Qiu W, Liang Y, Joshi T, Xu D: **MusiteDeep: A deep-learning framework for general and kinase-specific phosphorylation site prediction**. Bioinformatics, 2017, 33, 3909–3916.
15. Xie Y, Luo X, Li Y, Chen L, Ma W, Huang J, Cui J, Zhao Y, Xue Y, Zuo Z, Ren J.: **DeepNitro: Prediction of Protein Nitration and Nitrosylation Sites by Deep Learning**. Genomics Proteomics Bioinformatics, 2018, 16, 294–306.
16. Luo F, Wang M, Liu Y, Zhao X M, Li A, Hancock J: **DeepPhos: Prediction of protein phosphorylation sites with deep learning**. Bioinformatics, 2019, 35, 2766–277.
17. Guo L et al: **DeepPSP: A Global-Local Information-Based Deep Neural Network for the Prediction of Protein Phosphorylation Sites**. J Proteome Res, 2021, 20, 346–356.
18. Zhou Z H, Feng J: **Deep Forest**. Natl Sci Rev, 2019, 6, 74–86.
19. Li Z *et al.*: **Detecting Blood Methylation Signatures in Response to Childhood Cancer Radiotherapy via Machine Learning Methods**. *Biology (Basel)*, 2022, 11.

20. Ullah S *et al*<bi>.</bi>: **DbPAF: An integrative database of protein phosphorylation in animals and fungi**. Sci Rep, 2016, 6, 1–9. https://doi.org/10.1038/srep23534.

21. Dinkel H *et al.*: **Phospho.ELM: A database of phosphorylation sites-update 2011**. Nucleic Acids Res, 2011, 39, 261–267. https://doi.org/10.1093/nar/gkq1104.

22. Chen Z, Chen Y Z, Wang X F, Wang C, Yan R X, Zhang Z: **Prediction of Ubiquitination Sites by Using the Composition of k-Spaced Amino Acid Pairs**. PLoS One, 2011, 6.

23. Ahmed S, Kabir M, Arif M, Khan Z U, Yu D J: **DeepPPSite**: **A deep learning-based model for analysis and prediction of phosphorylation sites using efficient sequence information**. *Anal Biochem*, 2021, 612,113955.

24. Chen Z, Zhao P, Li F, Leier A: **PROSPECT: A web server for predicting protein histidine phosphorylation sites**. J Bioinform Comput Biol, 2020, 18, 1–17.

25. Lin S, Song Qi, Tao H, Wang W, Wan W, Huang J, Xu C, Chebii V, Kitony J, Que S, Harrison A, He H: **Rice-Phospho 1.0: A new rice-specific SVM predictor for protein phosphorylation sites**. Sci Rep, 2015, 5.

26. Zeng Z, Zhao S, Peng Y, Hu X, Yin Z: **Cascade Forest-Based Model for Prediction of RNA Velocity**. Molecules, 2022, 27, 7873.

27. Wang D, Liang Y, Xu D: **Capsule network for protein post-translational modification site prediction**. Bioinformatics, 2019, 35, 2386–2394.

28. Diella F, *et al*: **Phospho.ELM: A database of experimentally verified phosphorylation sites in eukaryotic proteins**. 2004.

29. Chen C W, Huang L Y, Liao C F, Chang K P, Chu Y W: **GasPhos: Protein phosphorylation site prediction using a new feature selection approach with a GA-aided ant colony system**. Int J Mol Sci, 2020, 21, 1–16.

30. Li W, Godzik A: **Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences**. Bioinformatics, 2006, 22, 1658–1659.

31. Amerifar S, Zahiri J: **ftrCOOL**: **An R Package for Feature Extraction from Biological Sequences**. 2020.

32. Bartlett P, Traskin M: **AdaBoost is Consistent**. in Advances in Neural Information Processing Systems, 2006, 19.

33. Cover T, Hart P: **Nearest neighbor pattern classification**. IEEE Trans Inf Theory, 1967, 13,21–27.

34. Cortes C, Vapnik V: **Support-vector networks**. Mach Learn, 1995, 20, 273–297.

35. Breiman L: **Random Forests**. Mach Learn, 2001, 45, 5–32.

36. Yao X: **Evolving artificial neural networks**. *Proceedings of the IEEE*, 1999, 87, 1423–1447.

37. Abraham A et al: **Machine learning for neuroimaging with scikit-learn**. Front Neuroinform, 2014, 8.

38. Safavian S R, Landgrebe D: **A survey of decision tree classifier methodology**. IEEE Trans Syst Man Cybern, 1991, 21, 660–674.

39. Ahmed Md S, Shahjaman Md, Kabir E, Kamruzzaman Md: **Prediction of Protein Acetylation Sites using Kernel Naive Bayes Classifier Based on Protein Sequences Profiling**. Bioinformation, 2018, 14, 213–218.

40. Zhao J, Zhuang M, Liu J, Zhang M, Zeng C, Jiang B, Wu J, and Song: **pHisPred: a tool for the identification of histidine phosphorylation sites by integrating amino acid patterns and properties**. BMC Bioinformatics, 2022, 23, 399.

41. Kim J H, Lee J, Oh B, Kimm K, Koh I: **Prediction of phosphorylation sites using SVMs**. Bioinformatics, 2004, 20, 3179–3184.

42. Banerjee S, Ghosh D, Basu S, Nasipuri M: **JUPred_MLP**: **Prediction of Phosphorylation Sites Using a Consensus of MLP Classifiers**. in *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015*, Springer,404. 2016, 23, 35–42.

43. Li F *et al*: **Quokka: a comprehensive tool for rapid and accurate prediction of kinase family-specific phosphorylation sites in the human proteome**. Bioinformatics, 2018, 34, 4223–4231.

44. Wang C *et al*<bi>:</bi> **GPS 5.0: An Update on the Prediction of Kinase-specific Phosphorylation Sites in Proteins**. Genomics Proteomics Bioinformatics, 2020, 18, 72–80.

45. Quinlan J R: **Induction of decision trees**. Mach Learn, 1986, 1, 81–106.

Figures

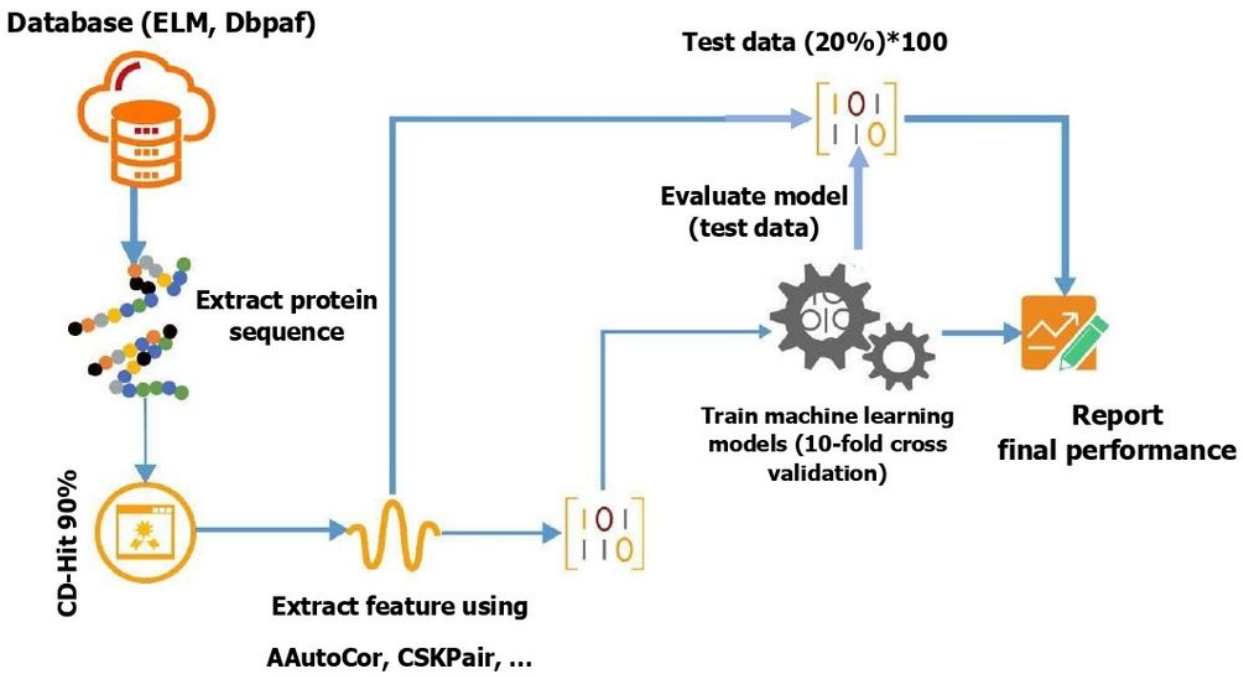

Figure 1

The architecture of our phosphorylation predictor. The predictor input consists of N sequences of length L from our database. Then using the feature extraction methods, the feature vector of these sequences was extracted and the resulting matrix was fed into a machine learning block and trained, we used the 10-fold cross-validation methods to evaluate the model.

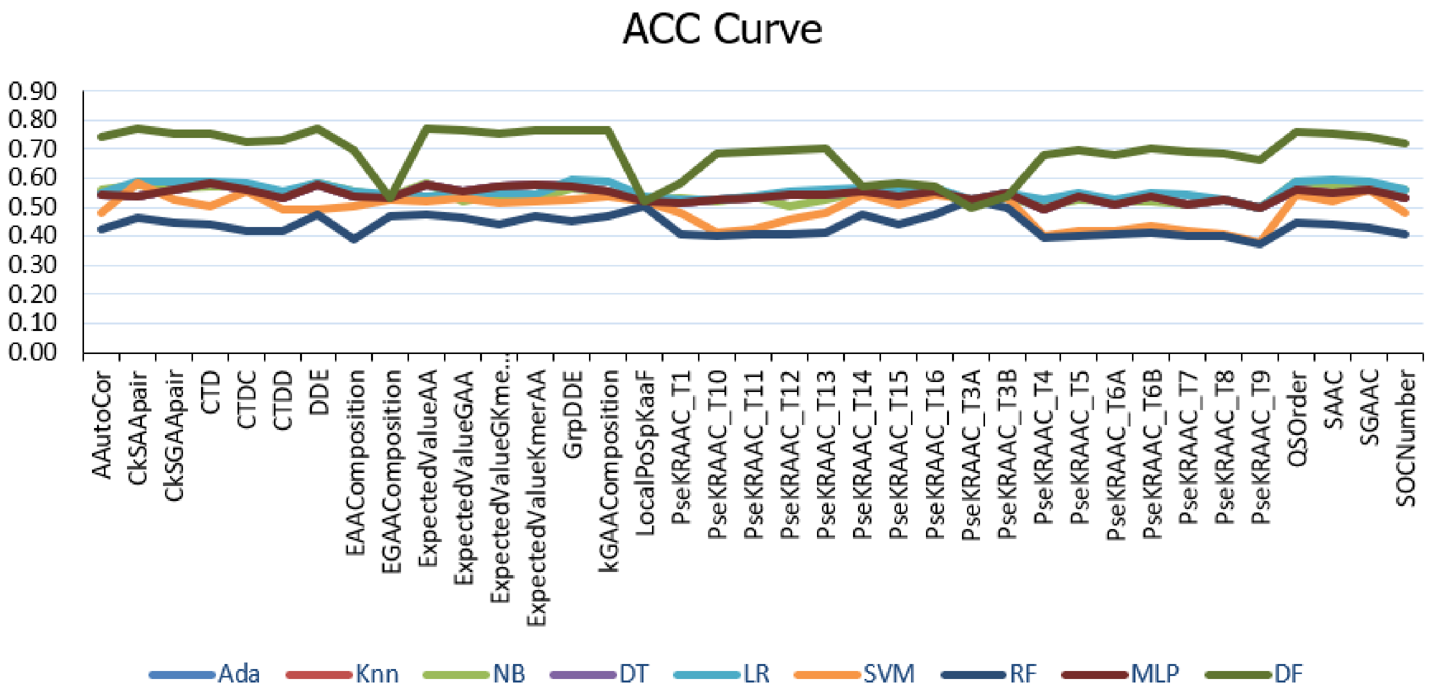

Figure 2

ACC of different feature extraction and machine learning methods.

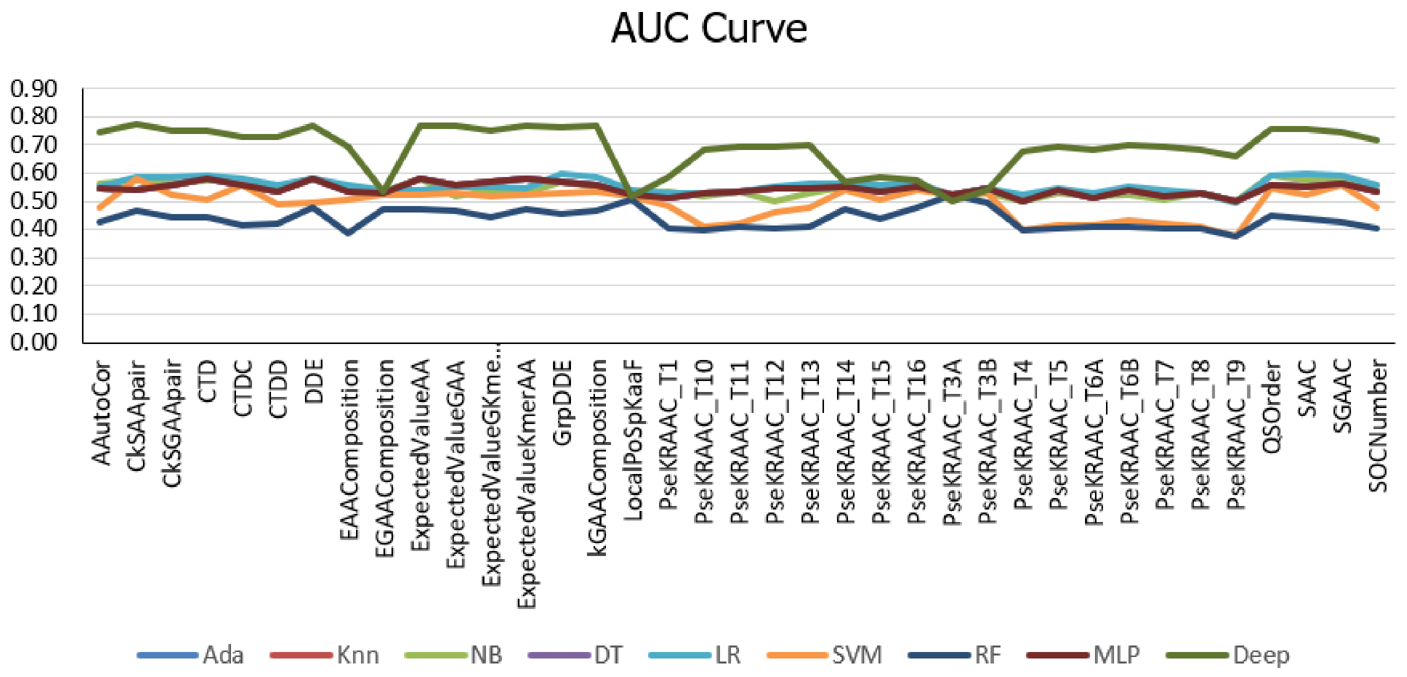

Figure 3

AUC of different feature extraction and machine learning methods.

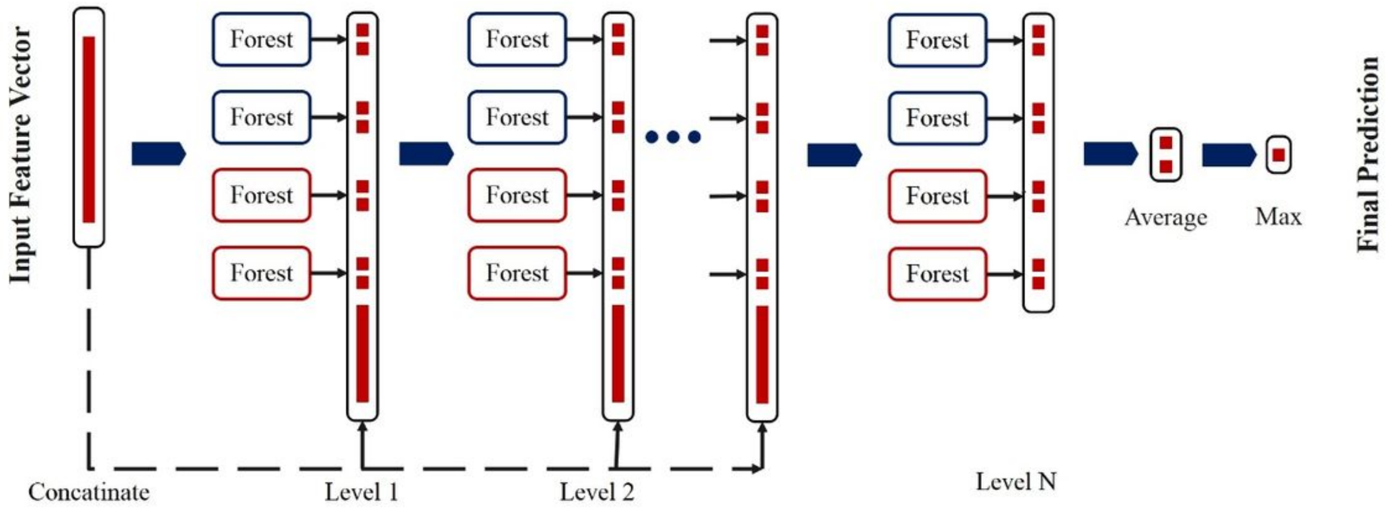

Figure 4

shows the gcforest structure. assume that each level of the cascade consists of two random forests (blue) and two completely random forests (red). Suppose that there are two classes to predict; therefore, each forest will output a 2D class vector, which is then concatenated for re-representation of the input. In the last layer, mediation is done between all the previous layers and finally, its max is introduced as the prediction result.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryFile.docx