An ensemble method approach to investigate kinase-specific phosphorylation sites

Sutapa Datta
Subhasis Mukhopadhyay
Department of Biophysics, Molecular Biology and Bioinformatics and Distributed Information Centre for Bioinformatics, University of Calcutta, West Bengal, India

Abstract: Protein phosphorylation is one of the most significant and well-studied post-translational modifications, and it plays an important role in various cellular processes. It has made a considerable impact in understanding the protein functions which are involved in revealing signal transductions and various diseases. The identification of kinase-specific phosphorylation sites has an important role in elucidating the mechanism of phosphorylation; however, experimental techniques for identifying phosphorylation sites are labor intensive and expensive. An exponentially increasing number of protein sequences generated by various laboratories across the globe require computer-aided procedures for reliably and quickly identifying the phosphorylation sites, opening a new horizon for in silico analysis. In this regard, we have introduced a novel ensemble method where we have selected three classifiers (least square support vector machine, multilayer perceptron, and k-Nearest Neighbor) and three different feature encoding parameters (dipeptide composition, physicochemical properties of amino acids, and protein–protein similarity score). Each of these classifiers is trained on each of the three different parameter systems. The final results of the ensemble method are obtained by fusing the results of all the classifiers by a weighted voting algorithm. Extensive experiments reveal that our proposed method can successfully predict phosphorylation sites in a kinase-specific manner and performs significantly better when compared with other existing phosphorylation site prediction methods.

Keywords: post-translational modification, cell signaling, phosphate

Introduction
Protein phosphorylation, an important reversible post-translational modification, occurs due to the addition of a covalently bound phosphate group into certain acceptor residues (serine, threonine, and tyrosine) in the substrate sequence by a group of enzymes called kinases. It is one of the most ubiquitous post-transitional modifications, found across the phylogeny from prokaryotes to eukaryotes, and plays a major role in the broad range of critical cellular phenomenon such as metabolism, cell signalling, apoptosis, and cellular proliferation. It has been shown that almost 30%–50% of the eukaryotic proteins undergo phosphorylation.

Kinases, by which the phosphorylation takes place, constitute one of the largest known protein superfamilies. About 1.7% of all the human genes encode as many as 518 different types of kinases, and they are classified into a hierarchical fashion with ten groups, 134 families, and 201 subfamilies primarily based on the homology of their catalytic domains. Therefore, the accurate recognition of phosphorylation sites along with the relevant kinase of a eukaryotic protein of interest is necessary to decipher the intracellular phenomenon.
Although mass spectrometry techniques are being used to detect the phosphorylation sites in a high-throughput manner, the ever-increasing number of protein sequences has rendered these methods to be prohibitively labor and cost intensive. As a result, the development of an accurate and automated in silico method for predicting phosphorylation sites based on the protein primary sequence information is desirable. A number of in silico methods have been proposed to predict phosphorylation sites; these methods can be roughly categorized into two groups. Some methods can only predict the phosphorylation sites, without providing any information about the specific kinase most likely to interact with the protein of interest; these methods include DISPHOS, CKSAAP_PhSite, PPRED, Netphos, PHOSIDA, and AMS. Other methods predict phosphorylation sites in a kinase-specific manner; in addition to predicting whether a candidate site is a phosphorylation site or not, they provide the kinase information of the target proteins. Such methods include Scansite, KinasePhos, NetphosK, PPSP, GPS, Postmod, BAE, AMS 4.0 Server, and Metapred. The details of these methods are discussed in a recent review by Trost et al. Most of the phosphorylation prediction methods use a single classifier; these methods sometimes do not produce satisfactory results due to their inherent limitations. In recent years, ensemble methods have received considerable attention in the machine learning community for increasing the effectiveness of a single classifier. A number of ensemble methods have been successfully applied in studying several biological problems such as predicting subcellular locations in proteins, gene expression analysis, protein–protein interaction site prediction, and prediction of siRNA efficacy. Some recent methods, such as Baggng-AdaBoost ensemble (BAE) proposed by Yu et al., PHOSphytation Site FindER (PHOSFER) by Trost et al., and recently a method proposed by Gao et al., have also used ensemble methods to predict phosphorylation sites. BAE is based on the Bagging and Adaboost ensemble approach, whereas PHOSFER uses random forest algorithms to predict phosphorylation sites in an organism-specific way. The method proposed by Gao et al uses a multitask learning framework to predict the phosphorylation site of four kinase groups. Although much progress has been made in the prediction of phosphorylation sites using various ensemble methods, there still exists a wide scope for improvement. None of these methods uses more than one feature scheme to train the classifiers, but a specific type of feature encoding scheme for precisely predicting phosphorylation sites of a protein sequence is not fully exploited. Therefore, no single feature encoding scheme can absolutely differentiate the phosphorylation sites from nonphosphorylation sites for all the kinases. In our proposed method, we have used three types of features: physiochemical features of amino acids, dipeptide composition, and protein–protein scoring for encoding a protein sequence. In the next step, we have used a feature selection method to remove the redundant features without deteriorating the performance of the method. Also, in order to construct a good ensemble method, the ensemble needs to construct accurate and diverse classifiers and to combine outputs from the classifiers effectively. Selecting a set of diverse classifiers that perform individually well is a nontrivial problem. In our proposed algorithm, we have adopted k-Nearest Neighbor (K-NN), multilayer perceptron (MLP), and least square support vector machine (LSSVM) as three diverse classifiers. The cardinal problem in predicting phosphorylation sites is the very small size of the dataset, because the number of known phosphorylation sites is very limited as compared to the nonphosphorylation sites. To overcome this problem, we have incorporated bootstrap resampling techniques to construct a number of datasets. Each of the resampled datasets generated from three feature encoding parameters is used as input in each of these three classifiers separately. The final result is obtained by fusing all the results obtained from the three classifiers through weighted voting. It is empirically demonstrated that an ensemble method can improve the prediction accuracy of the single classifiers performing independently.

**Materials and method**

**Data preparation**

In order to evaluate the performance of our proposed method and to compare it with other existing methods, we have extracted the phosphorylation sites from Phospho.ELM database (version 9.0 [available from http://phospho.elm.eu.org]). Experimentally validated phosphorylation sites of eukaryotic cells for 299 types of different kinases are curated in Phospho.ELM database. Version 9.0 of this database contains 8,718 proteins from different vertebrate species covering 31,754 serine, 7,449 threonine, and 3,370 tyrosine instances. Each entry in the database provides information about the substrate proteins along with the exact positions of the residues phosphorylated by a given kinase. In our study, we have considered the kinase families having at least 100 known and experimentally validated phosphorylation sites. Nine kinase families, including eight serine/threonine kinase families (protein kinase A [PKA], protein kinase B [PKB], protein kinase C [PKC], casein kinase 2 [CK2],...
cycalin-dependent kinase 1 (CDK1), mitogen-activated protein kinase [MAPK], calmodulin-dependent protein kinase II [CAM KII], glycogen synthase kinase 3 [GSK-3]) and one tyrosine kinase family (SRC) were chosen. In this study, we have considered the phosphorylation sites for the vertebrates only. We extracted the 21-mer sequence including 10 bp upstream to 10 bp downstream from the phosphorylation site with the phosphorylated residue at the central position; i.e., at position 11. If the phosphorylated residue (serine, S; threonine, T; tyrosine, Y) appears in the first ten residues or last ten residues, that is, if the distance between the phosphorylated residue and N or C terminal is less than 10-mer, an extra residue X (where X denotes any residue) is added before the first residue or after the last residue of the sequence in order to keep the protein sequence size at 21-mer and phosphorylated residue centered.

A negative dataset is prepared by taking the 21-mer peptides and centering all the nonphosphorylated S/T/Y residues of the substrate proteins corresponding to the nine kinase families mentioned above. We have taken nonphosphorylated S/T residues as negative control for the serine/threonine kinases in the PKA, PKB, PKC, CK2, CDK1, MAPK, CAM KII, GSK-3 kinase families and nonphosphorylated Y residues for tyrosine kinase family in SRC. In order to avoid the overestimation on accuracy while cross-validating, we have discarded the highly homologous sequences (ie, sequences having more than 60% similarity) from positive and negative datasets by using the CD-HIT clustering program.34

Furthermore, in context of the multiple kinase families the positive phosphorylation sites are outnumbered by the negative sites, resulting in an imbalanced dataset. As there are far more nonphosphorylated S/T/Y residues than phosphorylated residues, it is impractical to adopt whole nonphosphorylated sites as negative instances; it is preferable to select almost similar numbers of positive instances and negative instances. So in our study, the ratio of positive samples to negative samples is kept at 1:1.5 to avoid biased prediction. The number of positive phosphorylation sites and corresponding negative sites for the nine protein kinases are given in Table 1.

**Table 1** Number of positive and negative phosphorylation sites for nine protein kinases after removing the redundant data obtained from Phospho.ELM database version 9.0

<table>
<thead>
<tr>
<th>Kinase</th>
<th>Positive sites</th>
<th>Negative sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKA</td>
<td>288</td>
<td>440</td>
</tr>
<tr>
<td>PKB</td>
<td>70</td>
<td>105</td>
</tr>
<tr>
<td>PKC</td>
<td>279</td>
<td>420</td>
</tr>
<tr>
<td>MAPK</td>
<td>247</td>
<td>375</td>
</tr>
<tr>
<td>GSK-3</td>
<td>53</td>
<td>80</td>
</tr>
<tr>
<td>CDK1</td>
<td>133</td>
<td>200</td>
</tr>
<tr>
<td>CK2</td>
<td>187</td>
<td>281</td>
</tr>
<tr>
<td>CAM KII</td>
<td>67</td>
<td>105</td>
</tr>
<tr>
<td>SRC</td>
<td>120</td>
<td>180</td>
</tr>
</tbody>
</table>

**Note:** Phospho.ELM database (version 9.0) available from [http://phospho.elm.eu.org](http://phospho.elm.eu.org). Abbreviations: CAM KII, calmodulin-dependent protein kinase II; CDK1, cyclin-dependent kinase 1; CK2, casein kinase 2; GSK-3, glycogen synthase kinase 3; MAPK, mitogen-activated protein kinase; PKA, protein kinase A; PKB, protein kinase B; PKC, protein kinase C.

**Data encoding**

Data encoding is one of the most crucial factors affecting the performance of the classifier as well as the ensemble architecture. This is the process through which a sequence is converted to its numerical form and is presented to the classifier. Hence, it is necessary to choose a high-quality data encoding method that possesses the salient features of the amino acid sequences, keeping the generated code compact in dimensionality. Instead of using a simple binary representation, we have adopted a variety of data encoding schemes that include three types of amino acids features: physicochemical features, dipeptide composition, and protein–protein similarity score.

**Physicochemical features**

The Amino Acid Index35 (AAindex) is a database composed of numerical indices of various physicochemical and biochemical properties of amino acids and amino acid pairs. The latest version (version 9.1) of the AAindex database is divided into three sections: AAindex1, AAindex2 and AAindex3. AAindex1 contains 544 amino acid indices, AAindex2 contains 94 amino acid mutation matrices, and AAindex3 contains 47 statistical protein contact potential matrices. Here we have considered only the 544 indices of AAindex1 for the protein representation; AAindex2 and AAindex3 are matrices, and are not suitable for the protein sequence representation. Among the 544 amino acid indices, 13 had incomplete data or an overrepresentation of zeroes and were removed, leaving 531 properties as potential features for protein sequence representation.

Each amino acid residue in a sequence of length L is represented by the corresponding numerical value of the 531 amino acid indices in the AAindex1 database. Hence,
a single sequence is represented by a 531*L dimensional matrix. Although only a moderate correlation exists among various AAindex properties, a considerable correlation exists between the physicochemical property values corresponding to different amino acids.

In order to correlate the physicochemical property values of amino acids along a sequence, physicochemical distance transformation is carried out to represent the sequence order information as proposed by Liu et al** in 2012. For a given protein sequence of length L,

\[ A_1 A_2 A_3 A_4 \ldots A_L \]

\( A_1 \) is the amino acid at position 1, \( A_2 \) at position 2, and so on. For an amino acid index \( j \) in AAindex, the sequence order information associated with the \( j \)th physicochemical property can be given by the following equation:

\[
\delta_{ij} = -\sum_{i=1}^{L-\lambda} D_j(A_i, A_{i+\lambda})
\]

where \( \lambda \) is the distance between two amino acids along a sequence. \( D_j(A_i, A_{i+\lambda}) \) is given by the following sequence:

\[
D_j(A_i, A_{i+\lambda}) = (I_j(A_i) - I_j(A_{i+\lambda}))^2
\]

where \( I_j(A_i) \) and \( I_j(A_{i+\lambda}) \) are the normalized physicochemical property value of the amino acid \( A_i \) for index \( j \), \( I_j \) is the mean of the \( j \)th index of 20 amino acids, and \( \sigma \) is the variance. \( I_j \) and \( \sigma \) are given by:

\[
I_j = \frac{\sum_{i=1}^{20} I_j(A_i)}{20}
\]

\[
\sigma = \frac{\sum_{i=1}^{20} (I_j(A_i) - \bar{I}_j)^2}{20}
\]

Using the above information, we have calculated the sequence order information \( \delta \) for each physicochemical property in AAindex, taking \( \lambda=1, 2, \) and 3. We then averaged the three values of \( \delta \) corresponding to \( \lambda=1, \lambda=2, \) and \( \lambda=3 \) to obtain \( \delta_{\text{avg}} \), which is then used as a feature vector for classification.

**Dipeptide composition**

Dipeptide composition is a simplistic descriptor of protein sequence features and a well-known technique for amino acid feature encoding. There are 441 combinations of dipeptides considering 21 amino acids (including one dummy amino acid). Dipeptide composition is defined as \( fr(r,s) = N_{rs}/(N-1) \),

\[
fr(r,s) = \frac{N_{rs}}{N(N-1)}
\]
where \( r,s = 1,2,3, \ldots 20 \) and \( N_{rs} \) is the number of dipeptides of amino acid type \( r \) and \( s \). The value of each component of the dipeptide composition gives the fraction of the corresponding amino acid pairs in the sequence fragment. Here we have used dipeptide composition to represent the protein sequences in numerical form and have used as an input feature vector for further classification.

**Sequence similarity scoring scheme**

Another very popular sequence encoding method used in this study is the protein–protein sequence similarity scores. This scoring scheme is based on the hypothesis that the peptides having high similarity provide high scores; ie, if a peptide sequence receives a high score with another peptide, they are likely to be phosphorylated by the same kinase. It is assumed that the sequences having sequence similarity also bear similar structural and physicochemical properties. In our proposed method, we have used BLOSUM62 substitution matrix to calculate the similarity between two peptides.

The similarity score between two peptides \( A \) and \( B \) of length \( L \) is defined as

\[
S(A, B) = \sum_{i=1}^{L} \text{Score}(A_i, B_i)
\]

where \( \text{Score}(A_i, B_j) \) is the substitution score of amino acids \( A_i \) and \( B_j \) in the BLOSUM62 matrix. If \( S(A, B) < 0 \), we have redefined it as \( S(A, B) = 0 \). In this study, we have taken 10 bp upstream to 10 bp downstream of the phosphorylation site; ie, the length of each peptide sequence is \( L = 21 \).

**Feature selection**

After encoding the protein sequences into its corresponding numerical form using three feature encoding schemes, we have applied a feature selection method in order to improve the performance of the classifiers because there may be some redundant and irrelevant features that may reduce the performance of the classifiers and increase the complexity. A feature is proved to be high quality if it not only can differentiate classes by itself or in combination with other features but if there remains no redundancy among the features. Here, we have adopted a method proposed by Mitra et al\(^ \text{38} \) called Feature Selection using Feature Similarity (FSFS) for feature selection. The method uses similarity or correlation among features to remove the redundant one; it does not require any knowledge of class labels as FSFS is an unsupervised feature selection algorithm. FSFS selects features by clustering them into subsets and then chooses a representative feature from each cluster. Then a maximum information compression index is used for enumerating feature similarity measures. The method significantly reduces the dimensionality of the features. In our proposed method, the numbers of features are varied from 100% to 2% and the number of features that yields the best result is opted. We have shown variation of accuracy with the

![Accuracy versus number of features](image)
number of features in Figure 2. Based on the graph shown in Figure 2, we have selected 20% of features because for 20% and above, there is no significant change in the accuracy for all the kinases. The selected features are used to represent the protein sequences.

Data resampling

The three feature datasets generated by three data encoding methods are then divided into a set of new datasets using a bootstrap resampling method. In bootstrap resampling, a number of resampled subsets of the original dataset are generated by random sampling with replacement (so individual instances may appear in the subsets more than once) such that the size of each resampled subset is equal to the size of the original dataset. From the original dataset \( D \) of size \( n \), a set of new datasets \( \{D_1, D_2, D_3, D_4, \ldots, D_n\} \) is generated each of size \( n' \) such that \( n' = n \).

In the next step, each of the datasets generated through bootstrap resampling is used as input to a set of classifiers \( \{C_1, C_2, \ldots, C_r\} \). In this paper, we have used three supervised classifiers, LSSVM, MLP, and K-NN, for classification.

LSSVM

We have used LSSVM as a classifier. To avoid the high computational complexity of support vector machine for high-dimensional data, we have adopted the least square version of SVM. LSSVM simplifies the training procedure by avoiding the solving of quadratic programming problem.\(^{(39,40)}\)

Let us consider a linearly separable binary classification problem:

\[
(x_i, y_i)_{i=1}^n \text{ and } y_i = \{+1, -1\}
\]

where \( x_i \) is an \( n \)-dimensional feature vector and \( y_i \) is the label of this vector. LSSVM can be formulated as the optimization problem:

\[
\min_{w, b, \alpha} \tau(w, b, \epsilon) = \frac{1}{2}w'w + \frac{1}{2}C \sum_{i=1}^{n} \epsilon_i^2 \tag{8}
\]

subject to the quality constraints

\[
y_i [w' \varphi(x_i) + b] = 1 - \epsilon_i \tag{9}
\]

where \( C > 0 \) is a regularization factor, \( b \) is a bias term, \( w \) is the weight vector, \( \epsilon_i \) is the difference between the desired output and the actual output, and \( \varphi(x_i) \) is a mapping function.

The Lagrangian for problem of Eq 8 is defined as follows:

\[
L(w, \epsilon_i, b, \alpha_i) = \min_{w, b, \epsilon} \tau(w, b, \epsilon) - \sum_{i=1}^{n} \alpha_i [y_i [w' \varphi(x_i) + b] - 1 + \epsilon_i] \tag{10}
\]

where \( \alpha_i \) are Lagrange multipliers. The Karush–Kuhn–Tucker (KKT) conditions for optimality,

\[
\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i \varphi(x_i),
\]

\[
\frac{\partial L}{\partial \epsilon_i} = 0 \rightarrow \alpha_i = C \epsilon_i,
\]

\[
\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0, \text{ and }
\]

\[
\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow y_i [w' \varphi(x_i) + b] - 1 + \epsilon_i = 0,
\]

are the solution to the following linear system:

\[
\begin{bmatrix}
0 & -Y & b
\end{bmatrix}
\begin{bmatrix}
\alpha
\end{bmatrix}
= \begin{bmatrix}
0
\end{bmatrix} \tag{11}
\]

where \( \varphi = [\varphi(x_1), y_1, \ldots, \varphi(x_n), y_n] \), \( Y = [y_1, \ldots, y_n] \), \( \bar{I} = [1, \ldots, 1] \), and \( \alpha = [\alpha_1, \ldots, \alpha_n] \).

For a given RBF kernel function \( K(x, x_i) \) and a new test sample point \( x \), the LSSVM classifier is given by

\[
f(x) = \text{sgn} \left[ \sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b \right]. \tag{12}
\]

Multilayer perceptron

Feedforward MLP has also been used to predict the phosphorylation sites specific to a kinase label. Multilayer perceptron is the most commonly used feedforward neural network. MLP is the extended form of perceptron having one or more hidden layer. The nodes in the input layer are connected to the nodes of the hidden layer, which in turn are connected to the nodes of the output layer. Each connection is associated with a weight. In the input layer, a number of real valued inputs are given and MLP generates a single real values output according to an activation function applied to the weighted sum of the outputs of the units in the preceding layer. The feedforward neural network is trained with a back-propagation learning algorithm to optimize the classification accuracy. The final error \( E \) at the output layer is the sum of squared differences of the desired outputs \( d_i \) and the actually calculated outputs \( o_i \) of each output unit \( i \), and can be expressed as:

\[
E = \sum_i (d_i - o_i)^2. \tag{13}
\]
In this study, we have used Log sigmoid activation function \( \log \text{sig}(n) = \frac{1}{1 + \exp(-n)} \) in the hidden layer and linear activation function \( \text{a}=\text{purelin}(n) \) in the output layer. The network was initialized with random weights and biases and was trained using the Levenberg–Marquardt algorithm. We have tested the network by varying the number of iterations from 500 to 3,000 and the learning rate 0.1 to 0.5, and averaged all outcomes to obtain the best possible result. A momentum term of 0.95 was added to increase the learning rate with stability. The performance of the network was measured in terms of mean square error. The lower the mean square error, the better the network’s performance.

**K-NN classifier**

K-NN is one of the most fundamental and simple nonparametric methods used for classifying objects based on the closest training examples in the feature space. The K-NN algorithm works in two steps. First, for each query \( Q \), the K-NNs from the training data are identified based on distance (Euclidean, Manhattan, etc). In our work we have used Euclidean distance to find the K-NNs of a given query. If \( X \) is a particular training sample with \( i \)th feature \( x_i \), and \( Q \) is a query with \( i \)th feature \( q_i \), then the Euclidean distance between the training sample and query is given by

\[
\text{dist}(X,Q) = \sqrt{\sum_{i=1}^{n} (x_i - q_i)^2} \tag{14}
\]

where \( n \) is the total number of features.

In the second step, the query is assigned to the class most common among its K-NNs by a majority voting. Once we have obtained K-NNs of query \( Q \) using Euclidean distance, it is time for the neighbors to vote in order to predict \( Q \)'s class. For each class \( cl \), we count how many of the \( K \) neighbors have that class \( cl \). If \( R(Q) \) denotes the class of the query \( Q \), then

\[
R(Q) = \arg \max S(F,cl) \tag{15}
\]

where \( S(F,cl) \) denotes the number of samples \( x_i \ (x_i \in F) \), where \( F \) is the set of nearest neighbor of the sample \( Q \) with label \( l_k = cl \).

**Weighted voting**

The set of newly generated \( m \) resampled datasets \( \{D_1, D_2, D_3, \ldots D_m\} \) for three types of features are given as input into the three individual classifiers. Since there are three different types of feature spaces generated by three sequence encoding methods and each feature space is divided into \( m \) resampled datasets, there are \( 3 \times m \) numbers of input datasets given to each of the classifiers, resulting in \( 9 \times m \) numbers of classifiers and \( 9 \times m \) numbers of outputs.

The final output of the ensemble method is generated by fusing the outputs produced by \( 9 \times m \) numbers of individual classifiers through weighted voting.

Suppose that the classification results corresponding to the \( 9 \times m \) numbers of classifiers for a query protein sequence \( P \) are \( R_1, R_2, R_3, \ldots, R_{9m} \) such that

\[
R_a \in \{C_1, C_2, C_3, \ldots, C_9\} (n=1,2,3, \ldots, 9m), \tag{16}
\]

where \( C_1, C_2, C_3, \ldots, C_9 \) are the class levels corresponding to the protein sequences.

For a given query sequence \( P \), the weighted sum of the base classifiers for each class is calculated by

\[
V_j = \sum_{n=1}^{9m} \partial w_n, \partial = \begin{cases} 1, & R_a = i \\ 0, & R_a \neq i \end{cases} \tag{17}
\]

where \( 9m \) is the number of classifiers, \( i = C_1', C_2', C_3', \ldots, C_9 \) is the class levels, \( R_a \) is the predicted class level by the \( n \)th classifier, and \( w_n \) is the weight associated with the \( n \)th classifier. Each classifier has been assigned a weight that would denote the contribution of the classifier to the prediction system. In this study, we have considered the overall accuracy of each classifier as the weight corresponding to that classifier.

The final classification result, the class associated with the sample \( P \), is determined by the following equation:

\[
C_{final} = \arg \max \sum_{j=1}^{9} V_j \tag{18}
\]

where, \( V_j \) denotes the number of occurrence of each class level and is obtained from Eq 17.

**Performance assessment**

To evaluate the performance of the proposed method, a tenfold cross-validation is done. In tenfold cross-validation, each dataset is divided into ten equal subsets; nine subsets are used for training and one part is used for testing. This process was repeated for all ten parts.

The performance of the proposed method is measured by means of four parameters: recall, precision, F-measure, and accuracy. The measures are given by the following equations:

\[
\text{Recall} = \frac{TP}{TP + FN}, \tag{19}
\]

\[
\text{Precision} = \frac{TP}{TP + FP}, \tag{20}
\]
Results and discussion

Positive and negative phosphorylation sites for each kinase family were downloaded from Phospho.ELM database and filtered using the CD-HIT clustering program to get a nonhomologous dataset. The number of positive phosphorylation sites and negative sites corresponding to the nine protein kinases after removing the homologous sequences and data imbalance correction are given in Table 1. As discussed in the previous section, three feature encoding schemes were used to convert the sequences into their numerical form, and each of the encoded datasets was resampled using the bootstrap resampling method. We varied the number of resampled datasets, taking B = 10, B = 15, and B = 20, where B is the number of datasets obtained through bootstrap resampling. The resampled datasets were used as input to the three classifiers. The whole method was repeated by varying the window size from 5 to 21, taking all the odd numbers within the range as the window size with the phosphorylated residue at the center position. The variation of accuracy with various window sizes for all nine kinases is shown in Figure 3A–I, varying the number of resampled datasets (B = 10, B = 15, and B = 20). Figure 3A–I clearly indicates that window size 11 yielded the best result for almost all the kinases. Therefore, an optimal window size of 11 with the phosphorylated residue S/T/Y at center position was taken for further examination. The performance of each individual classifier is annotated in Tables 2–4 in terms of accuracy for three different encoding schemes while varying the number of resampled datasets from 10, 15, and 20. In order to enhance the performance of individual classifiers, we undertook a weighted voting, took the result of each classifier as input, and obtained the final result of our proposed method by using Eq 18. Table 5 shows the result of the ensemble method corresponding to the number of resampled datasets (B = 10, B = 15, and B = 20).

Evaluating different feature schemes on different classifiers

From Tables 2–4, we find that for the three different feature encoding schemes, none of the three classifiers individually performed extensively well in predicting the phosphorylation

\[
AC = \frac{TP + TN}{TP + FP + TN + FN} \quad (21)
\]

\[
F\text{-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (22)
\]

where TP, FP, TN, FN represent the number of true positives, false positives, true negatives and false negatives respectively.

Figure 3 Accuracy with various window sizes for all the nine kinases while varying the number of resample datasets.

Abbreviations: CAM KII, calmodulin-dependent protein kinase II; CDK1, cyclin-dependent kinase 1; CK2, casein kinase 2; GSK-3, glycogen synthase kinase 3; MAPK, mitogen-activated protein kinase; PKA, protein kinase A; PKB, protein kinase B; PKC, protein kinase C; SRC, tyrosin kinase SRC.
sites. On average, all three features performed similarly for most of the kinases. For some kinases the physicochemical features performed well, whereas for other kinases the dipeptide composition performed better. Performance of Protein–protein scoring was average for all the kinases. Tables 2–4 do not show any significant trend of dominance of any one feature in predicting phosphorylation sites. Although the performance of the individual classifiers varied a lot, none of the single classifiers showed any satisfactory result. LSSVM as an independent classifier yielded the highest accuracy, 83.8%, for kinase family CDK1 for B=10 when physicochemical properties were used for data encoding; for all other kinases, LSSVM gave an accuracy in the range of 65%–80%. K-NN gave an accuracy between 55%–75% for all the kinases. Among the three classifiers, MLP performed worst, with an accuracy ranging between 50%–65%. In order to circumvent this insufficiency and to improve our results, we applied the weighted voting method as discussed in the previous section.

**Performance of the ensemble method with weighted voting**

In weighted voting, the outputs from various classifiers are fused to obtain a final output for each query sequence. Here, instead of relying on the outputs of the classifier, we assigned a weight to each classifier, which implies that the greater the weight of a classifier; the greater its contribution towards predicting the right sequence.

Table 5 shows the result of the ensemble method, and a remarkable improvement can be seen for all the kinase families. For the kinase family CDK1, the ensemble method yielded 85.3%, 83.4%, and 83.1% accuracy for B=10, B=15, and B=20, respectively. For MAPK kinase, the proposed method yielded a quite impressive accuracy: 84%, 85.3%, and 84.4% for B=10, B=15, and B=20, respectively. Our method gives the lowest accuracy for the kinase family PKB. Table 5 also shows the change in precision, recall, accuracy, and F-measure for the different kinases when the number of resampled datasets was varied from B=10,

---

### Table 2 Performance of individual classifiers with different resampled data sizes in terms of accuracy (%)

<table>
<thead>
<tr>
<th>Kinase</th>
<th>SVM</th>
<th>MLP</th>
<th>K-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B=10</td>
<td>B=15</td>
<td>B=20</td>
</tr>
<tr>
<td>PKA</td>
<td>65.49</td>
<td>62.81</td>
<td>64.16</td>
</tr>
<tr>
<td>PKB</td>
<td>66.06</td>
<td>65.02</td>
<td>62.5</td>
</tr>
<tr>
<td>PKC</td>
<td>62.24</td>
<td>64.56</td>
<td>63.61</td>
</tr>
<tr>
<td>MAPK</td>
<td>64.36</td>
<td>66.95</td>
<td>63.3</td>
</tr>
<tr>
<td>GSK-3</td>
<td>73.01</td>
<td>65.51</td>
<td>77.68</td>
</tr>
<tr>
<td>CDK1</td>
<td>68.71</td>
<td>69.55</td>
<td>63.58</td>
</tr>
<tr>
<td>CK2</td>
<td>68.75</td>
<td>62.41</td>
<td>63.7</td>
</tr>
<tr>
<td>CAM KII</td>
<td>66.02</td>
<td>68.93</td>
<td>61.7</td>
</tr>
<tr>
<td>SRC</td>
<td>67.95</td>
<td>66.28</td>
<td>64.73</td>
</tr>
</tbody>
</table>

**Notes:** Dipeptide composition is used for sequence encoding.

**Abbreviations:** CAM KII, calmodulin-dependent protein kinase II; CDK1, cyclin-dependent kinase 1; CK2, casein kinase 2; GSK-3, glycogen synthase kinase 3; K-NN, K-nearest neighbor; MAPK, mitogen-activated protein kinase; MLP, multilayer perceptron; PKA, protein kinase A; PKB, protein kinase B; PKC, protein kinase C; SRC, tyrosin kinase SRC; SVM, support vector machine.

### Table 3 Performance of individual classifiers with different resampled data sizes in terms of accuracy (%)

<table>
<thead>
<tr>
<th>Kinase</th>
<th>SVM</th>
<th>MLP</th>
<th>K-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B=10</td>
<td>B=15</td>
<td>B=20</td>
</tr>
<tr>
<td>PKA</td>
<td>66.25</td>
<td>65.43</td>
<td>63.44</td>
</tr>
<tr>
<td>PKB</td>
<td>71.69</td>
<td>78.61</td>
<td>66.98</td>
</tr>
<tr>
<td>PKC</td>
<td>68.85</td>
<td>61.72</td>
<td>60.76</td>
</tr>
<tr>
<td>MAPK</td>
<td>61.24</td>
<td>67.67</td>
<td>62.02</td>
</tr>
<tr>
<td>GSK-3</td>
<td>67</td>
<td>73.16</td>
<td>66.5</td>
</tr>
<tr>
<td>CDK1</td>
<td>83.8</td>
<td>65.53</td>
<td>65.9</td>
</tr>
<tr>
<td>CK2</td>
<td>67.35</td>
<td>64.9</td>
<td>67.42</td>
</tr>
<tr>
<td>CAM KII</td>
<td>78.2</td>
<td>71.2</td>
<td>73.6</td>
</tr>
<tr>
<td>SRC</td>
<td>66.64</td>
<td>65.48</td>
<td>68.55</td>
</tr>
</tbody>
</table>

**Note:** Physicochemical properties of amino acids are used for sequence encoding.

**Abbreviations:** CAM KII, calmodulin-dependent protein kinase II; CDK1, cyclin-dependent kinase 1; CK2, casein kinase 2; GSK-3, glycogen synthase kinase 3; K-NN, K-nearest neighbor; MAPK, mitogen-activated protein kinase; MLP, multilayer perceptron; PKA, protein kinase A; PKB, protein kinase B; PKC, protein kinase C; SRC, tyrosin kinase SRC; SVM, support vector machine.
In order to evaluate the performance of our method, we compared our method with five other open access kinase-specific phosphorylation site prediction methods. In most previous studies, these five methods were used because of their high performance. The five methods are PPSP,\(^{18}\) KinasePhos 2.0,\(^{16}\) GPS2.0,\(^{19}\) Scansite,\(^{15}\) and NetPhosK 1.0.\(^{17}\) Bayesian decision theory was used to develop PPSP. KinasePhos employs the hidden Markov model to predict kinase-specific phosphorylation sites. Scansite searches for motifs within proteins that are likely to be phosphorylated by specific protein kinases, using the scores calculated from position-specific score matrices. NetPhosK 1.0 uses an artificial neural network to predict 17 kinase-specific phosphorylation sites. The GPS2.0 server uses a modified version of a group-based scoring algorithm\(^{41,42}\) to predict PK-specific phosphorylation sites in the hierarchy. In this evaluation, we selected all kinase groups and the balance performance option for PPSP. In the case of NetPhosK, prediction without filtering and a threshold value of 0.5 was selected to predict phosphorylation sites. KinasePhos 2.0 was run with the option of default specificity for a specific kinase. In this work, Scansite 2.0 was run by searching all motifs, with the “high stringency level” setting selected. For GPS2.0, a medium threshold was selected for a particular kinase family.

To avoid biased prediction, we have considered a candidate site to be true positive only when the site is predicted correctly. The comparison is done on the basis of the parameter’s precision, recall, accuracy, and F-measure. Figures 4–7 show the performance comparison of our proposed method with other methods. Figures 6 and 7 illustrate that the proposed method achieves the best performance in terms of accuracy and F-measure for all nine kinase families. PPSP performance was the second best with respect to accuracy and F-measure. Upon evaluating the results of this comparison, we determined that GPS2.0 performed almost the same as PPSP except in the case of CAM KII, where its performance was the worst among all methods in terms of precision, recall, accuracy, and F-measure. Scansite performed similarly to GPS2.0 for all the kinases except CAM KII. Scansite performance was the second best in terms of precision, accuracy, and F-measure for SRC, while for
other kinase families it gave an average performance. PPSP provided the highest precision value for CAM KII, followed by our method. But for kinases like PKA, PKB, PKC, MAPK, GSK-3, CDK1, CK2, and SRC, our proposed method outperformed the other methods in terms of precision. KinasePhos demonstrated the lowest precision value for all the kinase families except CAM KII. NetPhosK demonstrated better accuracy than KinasePhos, except for PKA and SRC, with an almost equal precision value to that of KinasePhos for PKA and a better precision value for PKB, PKC, MAPK, GSK-3, CDK1, CK2, CAM KII, and SRC. PPSP yielded the best recall values among all the methods, with the exception of CK2 and CAM KII. The proposed method comes next with respect to recall, which is almost equal to the prediction performance of PPSP; NetphosK demonstrated the worst performance.

We also applied our proposed method to predict the phosphorylation sites corresponding to serine, threonine, and tyrosine phospho residues in a nonkinase-specific manner. For this we prepared a new dataset comprising all the annotated S/T/Y sites as positive data in the database Phospho. ELM version 9.0. Negative data was prepared taking all the nonannotated S/T/Y sites from the same database. We took a 21-mer sequence centering at S/T/Y residues for both positive and negative data. In the dataset, positive to negative sequence ratios were kept at 1:1.5 to avoid any unnecessary biased predictions. To evaluate the performance of the proposed method for nonkinase-specific phosphorylation site prediction, we compared our method with three other well-known phosphorylation site prediction methods: DISPHOS, PPRED, and NetPhos. DISPHOS uses a position-specific amino acid composition and structural disorder information to predict phosphorylation sites. NetPhos uses an artificial neural network for prediction, whereas PPRED uses position-specific scoring matrices and support vector machines for distinguishing between phosphorylation and nonphosphorylation sites. The prediction accuracy comparison is shown in Figure 8. This figure shows that our proposed method outperforms the other methods in terms of accuracy to predict S/T/Y phosphorylation sites.

Taking into consideration measures such as accuracy, precision and F-measure, our proposed method yielded the best performance. The reasonably high and balanced values of precision and recall indicate that our method can predict true positives as well as true negatives to a high extent, and yielded the best F-measure among all the methods. Our method outperformed the well-known existing methods.
Figure 4 Comparison of various kinase-specific phosphorylation site prediction methods (PPSP, KinasePhos, GPS2.0, Scansite, and NetphosK) with our proposed methods in terms of precision.

Abbreviations: CAM KII, calmodulin-dependent protein kinase II; CDK1, cyclin-dependent kinase 1; CK2, casein kinase 2; GSK-3, glycogen synthase kinase 3; MAPK, mitogen-activated protein kinase; PKA, protein kinase A; PKB, protein kinase B; PKC, protein kinase C; SRC, tyrosin kinase SRC.

Figure 5 Comparison of various kinase-specific phosphorylation site prediction methods (PPSP, KinasePhos, GPS2.0, Scansite, and NetphosK) with our proposed methods in terms of recall.

Abbreviations: CAM KII, calmodulin-dependent protein kinase II; CDK1, cyclin-dependent kinase 1; CK2, casein kinase 2; GSK-3, glycogen synthase kinase 3; MAPK, mitogen-activated protein kinase; PKA, protein kinase A; PKB, protein kinase B; PKC, protein kinase C; SRC, tyrosin kinase SRC.
Figure 6 Comparison of various kinase-specific phosphorylation site prediction methods (PPSP, KinasePhos, GPS2.0, Scansite, and NetphosK) with our proposed methods in terms of accuracy.

Abbreviations: CAM KII, calmodulin-dependent protein kinase II; CDK1, cyclin-dependent kinase 1; CK2, casein kinase 2; GSK-3, glycogen synthase kinase 3; MAPK, mitogen-activated protein kinase; PKA, protein kinase A; PKB, protein kinase B; PKC, protein kinase C; SRC, tyrosin kinase SRC.

Figure 7 Comparison of various kinase-specific phosphorylation site prediction methods (PPSP, KinasePhos, GPS2.0, Scansite, and NetphosK) with our proposed methods in terms of F-measure.

Abbreviations: CAM KII, calmodulin-dependent protein kinase II; CDK1, cyclin-dependent kinase 1; CK2, casein kinase 2; GSK-3, glycogen synthase kinase 3; MAPK, mitogen-activated protein kinase; PKA, protein kinase A; PKB, protein kinase B; PKC, protein kinase C; SRC, tyrosin kinase SRC.
and effectively distinguished the phosphorylation sites from nonphosphorylation sites in a kinase-specific manner compared to existing kinase-specific phosphorylation site prediction methods.

**Conclusion**

In this study, we used an ensemble method to predict kinase-specific phosphorylation sites in vertebrates only. Three types of strategies, physicochemical features, dipeptide composition, and protein–protein scoring, were taken to represent the protein sequences in their numerical forms, and three popular classifiers, LSSVM, K-NN, and MLP, were used in the proposed ensemble method. The three datasets composed of three different parameter systems were given as input to each classifier, and the final result was obtained by fusing the outputs of the above classifiers through weighted voting. Because of the relatively small data size, we incorporated data resampling so that vigorous experimentation could be performed. The ensemble method yields a better result than individual classifiers. While LSSVM, K-NN, and MLP were used in this work, other classifiers can also be used to form different ensemble methods. In summary, the results of the predictions through the proposed ensemble method indicate that our method is very promising in detecting protein phosphorylation sites and may serve as an important complement to existing methods.

**Acknowledgments**

The authors wish to thank the anonymous reviewers for their valuable remarks. The authors also acknowledge the computational facility of DIC and Department of Biophysics, Molecular Biology and Bioinformatics, University of Calcutta.

**Disclosure**

The authors report no conflicts of interest in this work.

**References**