Protein Engineering, Design & Selection vol. 20 no. 8 pp. 405–412, 2007
Published online July 24, 2007 doi:10.1093/protein/gzm035

GANNPhos: a new phosphorylation site predictor based on a genetic algorithm integrated neural network

Yu-Rong Tang[1], Yong-Zi Chen[1], Carlos A. Canchaya[2] and Ziding Zhang[1,3]

[1]Bioinformatics Center, College of Biological Sciences, China Agricultural University, Beijing 100094, China and [2]Department of Microbiology, National University of Ireland, Cork Western Road, Cork, Ireland

[3]To whom correspondence should be addressed.
E-mail: zidingzhang@cau.edu.cn

With the advance of modern molecular biology it has become increasingly clear that few cellular processes are unaffected by protein phosphorylation. Therefore, computational identification of phosphorylation sites is very helpful to accelerate the functional understanding of huge available protein sequences obtained from genomic and proteomic studies. Using a genetic algorithm integrated neural network (GANN), a new bioinformatics method named GANNPhos has been developed to predict phosphorylation sites in proteins. Aided by a genetic algorithm to optimize the weight values within the network, GANNPhos has demonstrated a high accuracy of 81.1, 76.7 and 73.3% in predicting phosphorylated S, T and Y sites, respectively. When benchmarked against Back-Propagation neural network and Support Vector Machine algorithms, GANNPhos gives better performance, suggesting the GANN program can be used for other prediction tasks in the field of protein bioinformatics.

Keywords: genetic algorithm/neural network/phosphorylation site/prediction/protein bioinformatics

Introduction

As a ubiquitous mechanism for cellular regulation, protein post-translational phosphorylation plays a crucial role in a variety of biological processes such as signal transduction, transcription, translation, transport, cytoskeletal regulation and metabolism (Kobe *et al.*, 2005; Groban *et al.*, 2006). Serine (S), threonine (T) and tyrosine (Y) are the most frequently observed amino acids of phosphorylation in eukaryotes. As a very large family of enzymes, protein kinases catalyze phosphorylation, and many kinases themselves are regulated by phosphorylation, resulting in complicated signaling and regulatory networks (Groban *et al.*, 2006). Defects in protein kinase function cause a variety of diseases and kinases are major targets for drug design aimed at treating cancer, diabetes, etc. (Kobe *et al.*, 2005). More than 30% of proteins in the eukaryotic cell were estimated to be phosphorylated (Hubbard and Cohen, 1993). It was also estimated that the majority of human proteins might be phosphorylated at multiple sites (totally $>100\,000$ sites) (Zhang *et al.*, 2002; Blom *et al.*, 2004). Therefore, faster and more efficient screening tools for detecting phosphorylation sites in protein sequences are highly needed.

Classical experimental approaches such as radioactive labeling are commonly used in detecting phosphorylated proteins, and usually they are labor-intensive (Hjerrild and Gammeltoft, 2006). Recently, refinements of several affinity-based strategies such as immunoaffinity, immobilized metal affinity chromatography (IMAC) and strong cation exchange (SCX) chromatography, coupled with the tandem mass spectrometry have dramatically speeded up the identification of phosphorylation sites (Schwartz and Gygi, 2005). So far, several thousand sites of post-translational phosphorylation are known and have been compiled into some searchable phosphorylation site datasets such as Phospho.ELM (http://phospho.elm.eu.org) (Diella *et al.*, 2004). Although the number of experimentally characterized protein kinases and their phosphorylation sites will continue to grow quickly, there is still a serious need for bioinformatics methods to predict possible phosphorylation sites in proteins, which may be helpful to accelerate the functional understanding of massively available protein sequences obtained from genomic and proteomic studies.

Two classes of prediction tasks have been designed to address the computational identification of potential phosphorylation sites in proteins of interest. In the first category, prediction is only focused on forecasting whether a query S, T or Y residue is a phosphorylation site or not. Two commonly used methods are NetPhos (Blom *et al.*, 1999) and DISPHOS (Iakoucheva *et al.*, 2004). In the second category, prediction is switched to the identification of protein kinase-specific phosphorylation sites, such as NetPhosK (Blom *et al.*, 2004; Hjerrild *et al.*, 2004), PSSP (Xue *et al.*, 2006), PredPhospho (Kim *et al.*, 2004) and KinasePhos (Huang *et al.*, 2005a, 2005b), etc. By considering kinase classification in predicting phosphorylation sites, the predictive accuracy is increased and the corresponding kinase information of every possible phosphorylation site can be obtained (Kim *et al.*, 2004; Huang *et al.*, 2005a, 2005b). Due to the nature of statistical learning-based algorithms, this type of prediction may only be suitable for those kinase families in which the number of known phosphorylation sites is large enough. Therefore, these kinase-specific prediction systems are inevitably limited to only a few kinase families (Kim *et al.*, 2004).

To construct a statistical learning-based prediction algorithm, generally the input for a phosphorylation site predictor is presented by a $2n + 1$ residue long sequence with S, T or Y in the central position (i.e. the window size is equal to $2n + 1$) (Iakoucheva *et al.*, 2004; Xue *et al.*, 2006). The feature construction for a potential site (i.e. a sequence fragment of $2n + 1$ residues) is further required for the processing of a prediction algorithm. The common position-specific features have been constructed using the standard orthogonal representation (Blom *et al.*, 1999), which was previously used in predicting protein secondary structure (Qian and Sejnowski, 1988). It has also been well accepted that the

phosphorylation sites are preferred to be positioned on the surface of proteins, generally in disordered regions that are flexible enough to be accessed by the catalytic residues in the protein kinase (Kobe *et al.*, 2005). Fragments with rigid helical structure or buried in the hydrophobic core of the protein have low probability to contain phosphorylation sites (Kobe *et al.*, 2005). Since some structural information can be derived from sequence data, the incorporation of predicted structural information [e.g. predicted secondary structure (Rost and Sander, 1993; Jones, 1999) and protein disorder information (Ward *et al.*, 2004; Han *et al.*, 2006; Radivojac *et al.*, 2007)] can reasonably improve the prediction accuracy (Iakoucheva *et al.*, 2004).

The performance of a predictor is also strongly related to the adopted prediction algorithm. Artificial neural networks (ANN) (Blom *et al.*, 1999; Berry *et al.*, 2004; Hjerrild *et al.*, 2004) and support vector machines (SVM) (Kim *et al.*, 2004; Plewczynski *et al.*, 2005a, 2005b) have been frequently used in the prediction of phosphorylation sites. Some other algorithms were also employed. For example, DISPHOS relied on a logistic regression-based linear predictor (Iakoucheva *et al.*, 2004). Huang *et al.* (2005b) used a profile hidden Markov model in their KinasePhos predictor. Very recently, a PPSP predictor was constructed to predict PK-specific phosphorylation sites with Bayesian decision theory (Xue *et al.*, 2006). In addition to the above statistical learning-based algorithms, the consensus sequences (motifs) have also been used to search the phosphorylation sites in proteins of interest such as Scansite (Obenauer *et al.*, 2003).

Although a series of phosphorylation site predictors are publicly available, there is still a room to improve the predictive accuracy. With the recent increase in protein phosphorylation sites identified by mass spectrometry, a unique opportunity has arisen to develop new statistical learning-based algorithms for the prediction of phosphorylation sites. Although the ANN trained with the standard back-propagation algorithm (i.e. BPNN) has been well implemented in phosphorylation site prediction (Blom *et al.*, 1999; Hjerrild *et al.*, 2004), the BPNN has its limitations such as its inability to escape local optima (Sexton and Dorsey, 2000). To overcome the weakness of BPNN, genetic algorithm (GA) has been well employed for optimizing the connection weights within ANNs to achieve a better performance (Salomon, 1998; Sexton *et al.*, 1999; Sexton and Dorsey, 2000). To our best knowledge, such GA-integrated neural network (GANN) has not been used in predicting the phosphorylation sites of proteins yet.

This study is focused on developing a new phosphorylation site prediction system GANNPhos by using a GANN. The overall performance of this newly developed GANNPhos is characterized and compared with the BPNN- and SVM-based algorithms. GANNPhos is also benchmarked against one existing phosphorylation site predictor (i.e. DISPHOS) and its future development is discussed in the final part of this paper.

Methods

Datasets

Phosphorylation sites were obtained form Phospho.ELM database (Version 5.0) (http://phospho.elm.eu.org/), which contains 2540 substrate proteins from different species covering 4799 S, 974 T and 1433 Y sites. Since these residues are experimentally verified phosphorylation sites, they are regarded as the positive sites (P_S, P_T and P_Y) and they are compiled into the positive datasets (DB_P_S, DB_P_T and DB_P_Y). Each site within the dataset is represented by a sequence fragment of 25 amino acids, where the S, T or Y is in the central position. To remove redundant fragments within the datasets, the initial datasets were filtered using a 30% sequence identity. Since each site is represented by a sequence fragment with fixed length, the sequence identity is based merely on the matching between two fragments (i.e. no-gap alignment). As the middle residue in each site is always the same (S, T or Y), the sequence identity is defined as the percentage of identically matched residues out of 24 positions (i.e. the central position is not included). Thus, for any pair of fragments within one dataset, a 30% sequence identity cut-off means only seven residues are maximally allowed to be identically matched in the generated no-gap alignment. The numbers of P_S, P_T and P_Y were then reduced to 2546, 643 and 944, respectively (Table I), and the detailed data (Positive_S.txt, Positive_T.txt and Positive_Y.txt) is available in the supplemental material.

Similar to the methods used in the literature (Blom *et al.*, 1999; Iakoucheva *et al., 2*004), the negative sites (i.e. non-phosphorylation sites, N_S, N_T and N_Y) were obtained from these 2540 protein sequences and represented all S, T and Y residues that were not reported as being phosphorylated in Phospho.ELM. Since the number of the available negative sites is much larger than the phosphorylation sites, the negative sites were not exhaustively extracted. In the present study, 94 172 Ns, 57 867 N_T and 26 915 N_Y were initially selected. The 30% cut-off for sequence identity was also used to remove the redundancy within the negative datasets (i.e. DB_N_S, DB_N_T and DB_N_Y). Furthermore, the negative sites with $>30\%$ identity with any of the positive sites were also discarded. Thus, the final negative sites contained 22 597 Ns, 22 292 N_T and 13 505 N_Y (Table I), and the detailed data (Negative_S.txt, Negative_T.txt and Negative_Y.txt) is also available in the supplemental material.

To benchmark the proposed method against DISPHOS1.3, the datasets used in DISPHOS1.3 were also used to train and

Table I. Phosphorylation and non-phoshorylation sites used in the current study

Datasets[a]		Positive sites		Negative sites	
		No. of initial sites	No. of final sites	No. of final sites	No. of final sites
I	S	4799	2546	94172	22597
	T	974	643	57867	22292
	Y	1433	944	26915	13505
II	S	NA[b]	1079	NA	30310
	T	NA	270	NA	35085
	Y	NA	375	NA	15514

[a]The dataset I was based on the Phospho.ELM database (Version 5.0). The dataset II was initially used in training DISPHOS1.3.
[b]The corresponding data was not available.

test the proposed methods. In their datasets, the sizes for P_S, P_T and P_Y were 1079, 270 and 375, while the numbers of N_S, N_T and N_Y were 30 310, 35 085 and 15 514, respectively (Table I). The selection of these datasets were initially described in the original paper of DISPHOS (Iakoucheva *et al., 2*004), and more detailed information is available at http://www.ist.temple.edu/DISPHOS.

Feature construction

A new feature construction was developed in this study. The detailed procedures were exemplified in encoding the phosphorylated S residues. Since a sequence fragment of 25 amino acids was deployed to define a candidate phosphorylation site, the whole DB_P_S dataset can be presented as a profile with a sequence length of 25. Firstly, the position-specific amino acid composition within this profile was calculated. For instance, the amino acid composition for the j-th position could be denoted as the following vector.

$$(a_{P,1,j}, a_{P,2,j}, \ldots, a_{P,20,j})^{\mathrm{T}} \tag{1}$$

As usual, the 20 amino acids are ordered alphabetically according to their single-letter codes. For example, $a_{P,1,j}$ represents the composition of alanine (A) in the j-th position, and so forth. The position j can be ranged from 1 to 24. The 1–12 and 13–24 stands for 12 upstream and 12 downstream positions surrounding the candidate phosphorylation site, respectively. Since the residue in the center position is always S, the corresponding amino acid composition was not considered. Concerning the DB_N_S dataset, in the second step 10 subsets were randomly constructed and the size of each subset was controlled to be equal to the DB_P_S dataset. Then, the average amino acid composition and the standard deviation in each position were also calculated. They were expressed as:

$$(\bar{a}_{N,1,j}, \bar{a}_{N,2,j}, \ldots, \bar{a}_{N,20,j})^{\mathrm{T}}, \quad j = 1, 2, \ldots, 24 \tag{2}$$

and

$$(\sigma_{N,1,j}, \sigma_{N,2,j}, \ldots, \sigma_{N,20,j})^{\mathrm{T}}, \quad j = 1, 2, \ldots, 24 \tag{3}$$

In the third step, a parameter $z_{i,j}$ was established to indicate the propensity of the i-th amino acid appearing at the j-th position in the phosphorylated S sites.

$$z_{i,j} = \frac{(a_{P,i,j} - \bar{a}_{N,i,j})}{\sigma_{N,i,j}} \tag{4}$$

If $z_{i,j}$ is >0.0, the i-th amino acid in the j-th position is enriched around phosphorylation sites, higher value of $z_{i,j}$ corresponding to more significant enrichment. In contrast, if $z_{i,j}$ is <0.0, the i-th amino acid in the j-th position is depleted. In case $\sigma_{N,i,j} = 0.0$, $z_{i,j}$ was set as 0.0. Because of the limited number of the sites in the current datasets, this situation (i.e.$\sigma_{N,i,j} = 0.0$) may occur in few cases. When the calculation was performed on each position, the following matrix $Z_{20\times24}$ was obtained.

$$Z = \begin{pmatrix} z_{1,1} & z_{1,2} & z_{1,3} & \cdots & z_{1,24} \\ z_{2,1} & z_{2,2} & z_{2,3} & \cdots & z_{2,24} \\ z_{3,1} & z_{3,2} & z_{3,3} & \cdots & z_{3,24} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{20,1} & z_{20,2} & z_{20,3} & \cdots & z_{20,24} \end{pmatrix}_{20\times24} \tag{5}$$

To encode one potential site (i.e. a fragment of 25 amino acids), a 24-dimensional feature vector (X) was constructed by looking up the corresponding parameters from the above matrix, which was further explained in the following example.

Given that a phosphorylated S residue was presented by the following 25 residue long fragment:

$$\overbrace{\mathrm{G\ E\ S\ \cdots\ V}}^{12\ \text{Downstream}}\ \mathrm{S}\ \overbrace{\mathrm{P\ \cdots\ P\ P\ D}}^{12\ \text{Upstream}} \tag{6}$$

Then, the corresponding feature vector (X) was derived as below:

$$X = (x_1, x_2, \ldots, x_{24}) = (\overbrace{z_{6,1},\ z_{4,2},\ z_{16,3},\ \ldots,\ z_{18,12}}^{12\ \text{Downstream}}, \times \overbrace{z_{13,13},\ \ldots,\ z_{13,22},\ z_{13,23},\ z_{3,24}}^{12\ \text{Upstream}}) \tag{7}$$

In the above equation, x_1 is encoded by glycine (G) in the first position of 12 downstream residues. Since G is alphabetically ranked as the sixth position among the 20 amino acids, the corresponding value of x_1 is $z_{6,1}$. Analogous to x_1, the values of x_2, x_3, …, x_{24} can also be obtained as described in Eq. (7). Since the matrix Z reflects the position-specific amino acid propensity surrounding the phosphorylation sites, this encoding system is called PSAAP feature. The same feature construction procedures were used in encoding the phosphorylated T and Y sites.

Genetic algorithm-integrated neural network

The proposed GANN uses GA to optimize the connection weights of the ANN over the training dataset. With a basic architecture similar to the BPNN, the current GANN contains one input layer, one hidden layer and one output layer (Fig. 1a). The number of input nodes is equal to the dimensionality of the PSAAP feature vector, which is 24 in the present study. According to our preliminary optimization, the number of hidden nodes is set as 15 in this work. Only one output node (y) is needed, and the corresponding value is '1' or '0', representing a phosphorylation site or a non-phosphorylation site. w_{ij} denotes the weight from an input node to a hidden node and w_{jk} the weight from a hidden node to the only output node. The neural network uses a sigmoid function to provide a continuous activation function.

The basic idea of GA is that a population of potential solutions (individuals) is refined iteratively to get the 'fittest' individual. The individuals in a population are also called 'chromosomes', consisting of 'genes' that represent the properties of the individual. The function to optimize is called a 'fitness' function. Each iteration is frequently called a

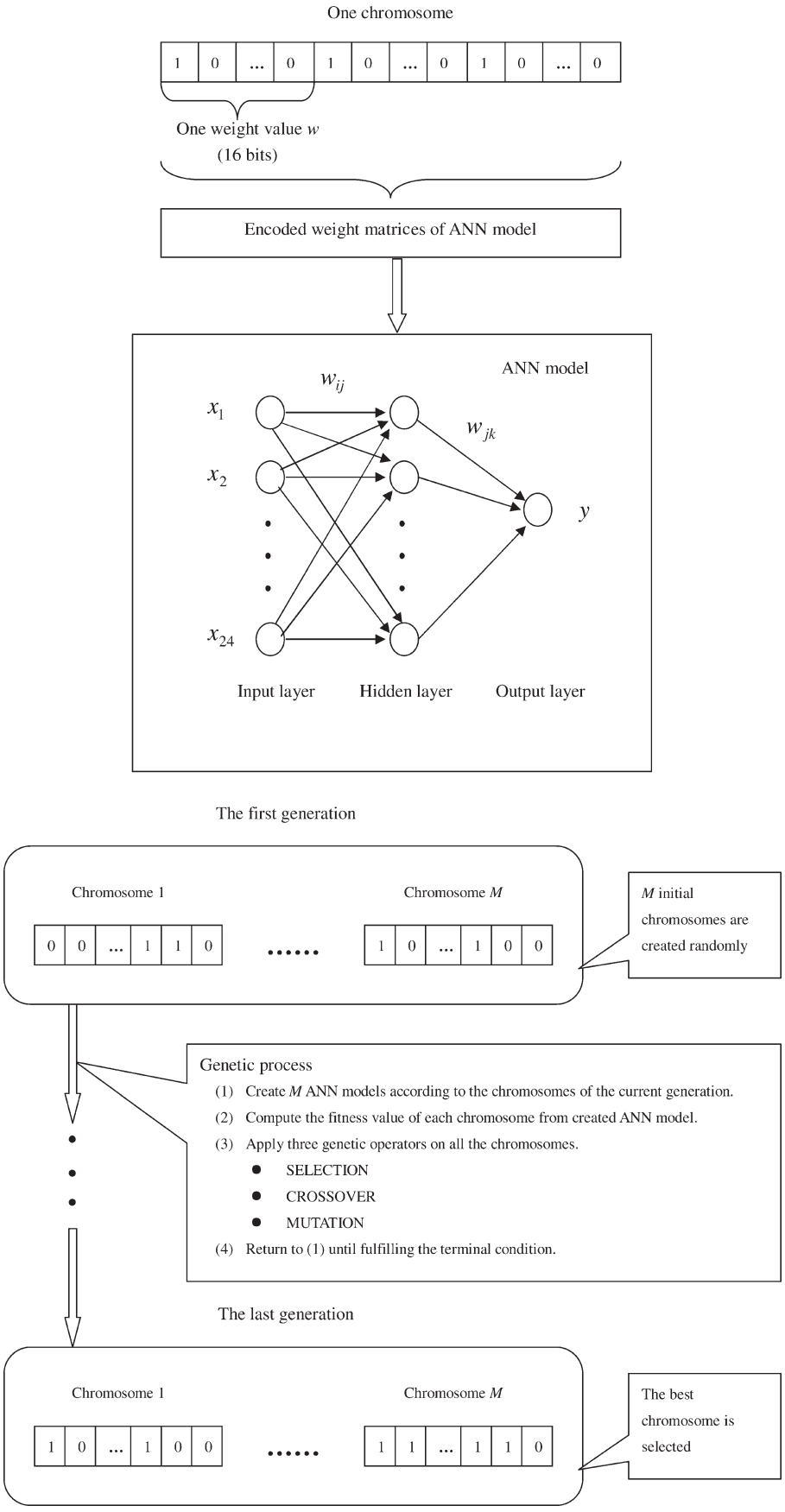

Fig. 1. The genetic algorithm-integrated neural network (GANN) used in developing GANNPhos predictor. (a) The architecture of GANN. (b) The training procedures of GANN.

'generation'. In the current study, the chromosome consists of a combination of '0' and '1' characters (bits) (Fig. 1a). Sixteen characters (i.e. 16 'genes') represent one weight value and one chromosome encodes all the connection weights (i.e. weight matrixes) of an ANN model.

To optimize the connection weights, an initial population of M chromosomes is randomly generated in the first generation (Fig. 1b). Genetic process is then applied to the initial chromosomes, which can be divided into four steps. In the first step, M ANN models are created according to chromosomes of the current generation. In the second step, the fitness value of each chromosome is computed to evaluate the corresponding ANN model. The Matthews correlation coefficient (MCC) (Matthews, 1975) is defined as the fitness function f.

$$f = \mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FN} \times \mathrm{FP})}{\sqrt{(\mathrm{TP}+\mathrm{FN}) \times (\mathrm{TN}+\mathrm{FP}) \times (\mathrm{TP}+\mathrm{FP}) \times (\mathrm{TN}+\mathrm{FN})}} \quad (8)$$

Where TP, FP, FN and TN denote true positives, false positives, false negatives and true negatives. In the third step, three operators (SELECTION, CROSSOVER and MUTATION) are applied to the chromosomes of the current generation to obtain the new chromosomes of the next generation.

The SELECTION operator determines if a chromosome in the current generation should be selected in the next generation or not. The chromosome with the highest fitness value in the current generation is automatically selected in the next generation. Subsequently, $M-1$ chromosomes are selected from all the chromosomes in the current generation using Roulette wheel selection (Wong *et al.*, 2000) and then moved into an intermediate pool. Note that the same chromosome may be selected more than once. This process consists in computing the selection probability of each chromosome according to its fitness value and the chromosomes with high probability are randomly chosen into the intermediate pool.

The CROSSOVER operator exchanges the structure between two chromosomes. For each pair of chromosomes from the intermediate pool, a uniform crossover (Beasly *et al.*, 1993) with a probability P_c is carried out to randomly distribute the genes from the two original chromosomes to obtain two new chromosomes.

The MUTATION operator arbitrarily alters one or more genes of chromosomes in the intermediate pool. Since high mutation probability can transform the GA into a purely random searching algorithm and too low probability can often result in the premature convergence of GA, generally the mutation probability is constrained to be in the range of 0.001–0.1. In this study, a self-adaptive mutation method is employed to identify the optimal mutation probability with the key idea that the adaptation of mutation probability is based on the standard deviation of the fitness values of all chromosomes in each generation. The mutation probability in the i-th generation P_m^i is computed as:

$$P_m^i = \begin{cases} P_m \times [100.00 \times (0.05 - s) + 1.00], & s \le 0.05 \\ P_m, & s > 0.05 \end{cases} \quad (9)$$

Where P_m is the initial mutation probability set as 0.01. The parameter s is the standard deviation of the fitness values. Using Eq. (9), the enough diversity among all the chromosomes in each generation can be well maintained. With the calculated mutation probability of the current generation, the mutation probability of each gene (i.e. each bit) in each weight value is further assigned with the principle that the leftmost gene (i.e. the highest bit) should have the lowest probability for mutation to guarantee that the real weight value encoded by the 16 bits is reasonably changed after mutation. The mutation probability of the j-th gene in each weight value $P_m^i(j)$ is defined as below:

$$P_m^i(j) = P_m^i \times f(j), \quad j = 1, 2, \ldots, 16 \quad (10)$$

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad (11)$$

Here, $f(x)$ is a standard normal distribution function that distributes diverse probability scale to each bit. Then, the mutation operation is performed on the $M-1$ chromosomes in the intermediate pool by changing the value of the j-th character of each weight from 0 to 1 or from 1 to 0 with a probability of $P_m^i(j)$.

After the accomplishment of the above mutation operator, all the $M-1$ chromosomes in the intermediate pool are merged into the new generation. In the fourth step, the above iterative training procedures are carried out to obtain the newer generation until fulfilling a terminal condition. The terminal condition is that the maximum generation numbers or the threshold of the fitness value is reached. At the end of the training, the best chromosome with the highest fitness from the last generation is selected to create an ANN prediction model that can be used to perform a feed-forward computation to obtain the prediction output over a test dataset.

The proposed GANN program with C++ source code was developed in our laboratory and it was implemented in a Windows XP system. Using the PSAAP encoding as input, GANN was used to construct a phosphorylation site predictor named GANNPhos with the following configuration: (1) the maximum generation number–1000; (2) the threshold of the fitness value–0.7; (3) population size (M)–100; (4) crossover probability (P_c)–0.9.

Back-propagation neural network (BPNN)

For the purpose of comparison, the BPNN with a single hidden layer was also used to construct a phosphorylation site predictor with the PSAAP vectors as input. The same number of units (15 units) in the hidden layer and the same activation function (sigmoid function) used in GANNPhos were adopted.

Support vector machine

The SVM is a machine-learning algorithm for two classes of classification with the goal to find a rule that best maps each member of training set to the correct classification (Cai *et al.*, 2003; Dobson and Doig, 2003). In linearly separable cases, SVM constructs a hyperplane that separates two different groups of feature vectors in the training set with a maximum margin. The orientation of a test sample relative to the hyperplane gives the predicted score, and hence the predicted class can be derived. For the purpose of

comparison, the SVM algorithm was also used to predict the phosphorylation sites. Apart from the PSAAP feature vectors used as input, the orthogonal encoding was also employed. The implementation of the SVM algorithm was SVM-Light (http://svmlight.joachimms.org). The applied kernel functions were the linear, polynomial and radial basis functions. Other than changing the kernel functions, the algorithm was run with the default settings in a Linux Platform.

Training and testing

In this study, three subsets ($DB\_N'_S$, $DB\_N'_T$ and $DB\_N'_Y$) were randomly constructed from DB_N_S, DB_N_T and DB_N_Y to have the same size as DB_P_S, DB_P_T and DB_P_Y, respectively. Each set of DB_P_S, DB_P_T and DB_P_Y with the corresponding negative sets of $DB\_N'_S$, $DB\_N'_T$ and $DB\_N'_Y$ to construct predictors for S, T and Y. A 10-fold cross-validation was performed. The whole dataset (e.g. $DB\_P_S + DB\_N'_S$) was randomly divided into 10 subgroups of roughly equal size. The sizes of the positive and negative sites in each subgroup were set as the same. In each evaluation step, one subgroup was selected for testing, while the rest nine subgroups were used as the training dataset. Concerning the PSAAP encoding, it should be emphasized that the testing set was always excluded in deriving the Z matrix. For each of phosphorylated sites (S, T or Y), 10 models were constructed to assess the performance (i.e. 10-fold cross-validation). To scrutinize the difference of predictive accuracy caused by the different choices of negative datasets, the above 10-fold cross-validation was repeated 30 times by randomly changing the negative datasets (i.e. $DB\_N'_S$, $DB\_N'_T$ and $DB\_N'_Y$). Finally, the overall performance was averaged over these 30 times of 10-fold cross-validation tests. It was found that the average accuracy almost remained unchanged when the 10-fold cross-validation test was carried out more than 10 times. Thus, the current cross-validation reliably reflected the overall performance of the proposed method over the selected datasets. The same training and testing procedures were used in assessing the BPNN- and SVM-based predictors.

Results and Discussion

Prediction performance of GANNPhos

A novel phosphorylation site prediction system (i.e. GANNPhos) has been developed by using our in-house GANN program. Here, four measurements, i.e. Accuracy (Ac), Sensitivity (Sn), Specificity (Sp) and MCC, were jointly used to assess the performance of GANNPhos (Table I). The overall prediction accuracy (Ac) of GANNPhos reached 81.1% for S (Sn = 80.0%, Sp = 82.1%, MCC = 0.621), 76.7% for T (Sn = 72.1%, Sp = 81.3%, MCC = 0.536) and 73.3% for Y (Sn = 70.9%, Sp = 75.8%, MCC = 0.467).

To avoid the overweighing of negative sites, the similar size of positive sites and negative sites in training a phosphorylation site predictor was used in the literature (Kim *et al.*, 2004; Iakoucheva *et al.*, 2004). For instance, Kim *et al.* (2004) reported that a maximum accuracy was achieved when the ratio of positive and negative sites were set to 2:3 in their PredPhospho predictor. A balanced number of the positive and negative sites (1:1) was used in training DISPHOS (Iakoucheva *et al.*, 2004). In this work, the ratio of positive sites and negative sites was also set as 1:1, although it may be further optimized to achieve a higher accuracy.

To encode a site, the window size was set as 25 (i.e. $n = 12$) in the present study. In our preliminary testing of the algorithm, it was found that the results were similar when the window size ranged from 17 to 25 (data not shown). These results are in good agreement with the general consensus that the kinases physically contact a region of 7–12 residues surrounding the phosphorylated residue (Songyang *et al.*, 1994).

Although the reported accuracy of GANNPhos is relatively high, the unreliable negative training set may still impact its proteome-wide application. In this work, some phosphorylation sites which have not been experimentally discovered are likely to be presented in the negative training set and inevitably cause the 'noise' of our algorithm. To decrease such type of 'noise', two strategies were reported in the literature (Blom *et al.*, 1999; Iakoucheva *et al.*, 2004). In the first strategy adopted by Blom *et al.* (1999), all potential acceptor residues in the entire set of protein sequences not reported as being phosphorylated, were initially assigned as negative sites. Subsequently, during initial neural network training sessions, all negative sites predicted as positive sites were excluded. The resulting dataset was used for the final neural network training session. Compared with the random selection of the negative sites, the predictive accuracy was significantly increased in this so-called 'augmented method'. In the second strategy adopted in DISPHOS, the training procedure was repeated for $I = 30$ random selections of negative examples, and the final jury-based prediction on the test set was made by averaging raw outputs from all I models (Iakoucheva *et al.*, 2004). It was estimated that if the I was set as 1, the accuracy is decreased by 2–3% in each case (Iakoucheva *et al.,* 2004). Since more attention was paid on investigating the performance based on the GANN algorithm, note that in this work the negative dataset was randomly selected without any constraint. If the above strategies were used in GANNPhos system, a higher performance would be expected.

Comparison of GANNPhos with BPNN- and SVM-based algorithms

Based on the same feature construction (i.e. PSAAP), the conventional BPNN and SVM algorithms were also used to build phosphorylation site predictors. Since the same dataset and the same cross-validation method were chosen, it allowed for a reliable comparison of performance resulted from different algorithms. As clearly shown in Table II, the Ac, Sn, Sp and MCC values reported in GANNPhos were considerably higher than those of BPNN and SVM. Compared with the BPNN-based predictor, the accuracy of S, T and Y predictors in GANNPhos was increased by 2.8, 4.7 and 4.5%, respectively. Compared with the SVM-based predictor, the increased accuracy was in the range of 1.0% for S to 4.0% for T. The overall performance of BPNN and SVM were comparable, although a higher accuracy was observed in the SVM algorithm. Currently, BPNN and SVM have been served as two major machine-learning tools in constructing diverse prediction tasks within the field of protein bioinformatics. The higher accuracy obtained by GANNPhos as a case study in the phosphorylation site

Table II. The results of phosphorylation site prediction for GANNPhos and other algorithms[a]

Site	Method	Datasets[e]	Ac (%)	Sn (%)	Sp (%)	MCC
S	GANNPhos	I	81.1 ± 0.2	80.0 ± 0.5	82.1 ± 0.6	0.621 ± 0.005
	BPNN	I	78.3 ± 1.3	78.4 ± 1.7	78.3 ± 2.4	0.567 ± 0.025
	SVM_1[b]	I	80.1 ± 0.7	76.4 ± 0.5	83.8 ± 0.6	0.605 ± 0.008
	SVM_2[c]	I	80.3 ± 0.6	72.4 ± 0.4	88.2 ± 0.5	0.614 ± 0.007
	GANNPhos	II	81.2 ± 0.6	80.0 ± 0.9	82.4 ± 1.1	0.624 ± 0.011
	DISPHOS1.3[d]	II	81.3	NA[f]	NA	NA
T	GANNPhos	I	76.7 ± 0.8	72.1 ± 2.2	81.3 ± 1.8	0.536 ± 0.016
	BPNN	I	72.0 ± 1.1	70.2 ± 2.6	73.7 ± 2.4	0.439 ± 0.021
	SVM_1[b]	I	72.6 ± 1.7	63.5 ± 1.1	82.0 ± 1.4	0.461 ± 0.020
	SVM_2[c]	I	72.3 ± 1.4	57.7 ± 1.0	87.0 ± 1.3	0.469 ± 0.017
	GANNPhos	II	79.8 ± 1.4	75.3 ± 2.3	84.3 ± 2.1	0.598 ± 0.028
	DISPHOS1.3[d]	II	74.9	NA	NA	NA
Y	GANNPhos	I	73.3 ± 0.7	70.9 ± 1.6	75.8 ± 1.5	0.467 ± 0.014
	BPNN	I	68.8 ± 0.8	65.1 ± 2.7	72.5 ± 2.3	0.377 ± 0.015
	SVM_1[b]	I	70.5 ± 1.3	65.9 ± 0.8	75.1 ± 1.0	0.413 ± 0.015
	SVM_2[c]	I	69.6 ± 1.4	64.2 ± 1.0	74.9 ± 1.1	0.395 ± 0.015
	GANNPhos	II	79.4 ± 0.8	74.7 ± 1.9	84.0 ± 1.6	0.589 ± 0.016
	DISPHOS1.3[d]	II	79.5	NA	NA	NA

[a]The corresponding measurement in this work was represented as the average value ± standard deviation.
[b]The PSAAP vectors were used as input. The optimal results were obtained using the linear kernel function.
[c]The orthogonal feature vectors were used as input. The optimal results were obtained using the polynomial kernel function.
[d]The corresponding values were cited from http://www.ist.temple.edu/DISPHOS.
[e]The dataset I was based on the Phospho.ELM database. The dataset II was the training data set used in DISPHOS1.3. More details about these two datasets can be found in Table I.
[f]The corresponding value was not available.

prediction suggests that the GANN program can be further applied in other protein bioinformatics-related topics.

Comparison of different feature vectors

In this work, a new encoding named PSAAP with a relatively low dimension to present a potential phosphorylation site was developed. Using the SVM algorithm, we benchmarked the accuracy resulted from the orthogonal and PSAAP encodings. Since both encodings are position-specific and the embedded information is close, a similar performance was expected. The comparison confirmed that the overall accuracy based on these two encodings were almost identical, although the PSAAP encoding was found to have a higher accuracy (+1.0%) in predicting phosphorylated Y sites (Table II). Compared with BPNN and SVM, the learning procedure in GANN is more computational time-consuming. Therefore, the newly constructed PSAAP encoding is particularly suitable for the GANNPhos predictor.

In addition to the sequence-based encodings, structural information has been considered in some predictors such as NetPhos and DISPHOS. Compared with the available sequence information, the experimentally determined structural data of proteins is still limited. Fortunately, many structural properties can be predicted with reasonable accuracy. Therefore, how to effectively combine predicted structural information with sequence information represents an important strategy to improve the accuracy of protein phosphorylation site prediction.

Comparison of GANNPhos with DISPHOS1.3

Since several kinase-non-specific phosphorylation site prediction systems have been publicly available, it is important and interesting to compare GANNPhos with some existing predictors. As reported by Iakoucheva *et al.* (2004), DISPHOS demonstrated higher accuracy when benchmarked against NetPhos and Scansite. Therefore, DISPHOS remains one of the best non-kinase-specific phosphorylation site predictors. This study is only focused on benchmarking our GANNphos against DISPHOS1.3, the newest version of DISPHOS.

The major differences among different predictors originated from many aspects such as the collection of dataset, the choice of mathematical models, the selection of feature vectors and cross-validation processes. Therefore, a comparison based merely on the accuracy reported by the original papers is somehow subjective. Here, the training datasets used in DISPHOS1.3 was further employed for training GANNPhos, which allowed for a fair comparison of the performance between these two predictors. As also shown in Table II, the results of GANNPhos are fully comparable to DISPHOS1.3, revealing almost identical accuracy in predicting S and Y sites, a considerably higher accuracy in predicting T sites (about +5.0%).

Future development

The proposed GANNPhos has been benchmarked to have a good performance, suggesting that it can serve as a competitive predictor to be practically applied to detect potential phosphorylation sites in proteins of interest. Concerning the future development, the following two strategies should be helpful to obtain a better prediction system. (1) In addition to optimize the connection weights within an ANN, GA has also been successfully employed to optimize the architecture of an ANN (Heckerling *et al.*, 2004). Therefore, integrating the GA-based network architecture optimization into the current GANNphos can result in a higher predictive accuracy; (2) Using some dimensionality reducing methods

reported in the literature (Dobson and Doig, 2003; Iakoucheva *et al.*, 2004; Zhang *et al.*, 2005) to effectively combine and refine the available features (e.g. predicted structure information) is promising to construct a better predictor. Finally, it should be emphasized that the GANN algorithm developed in this work can be used in many diverse prediction tasks and it may be expected to serve as an important algorithm in computational function prediction of proteins in the post-genomic era (Ofran *et al.*, 2005).

Acknowledgement

We thank Dr. Francesca Diella (EMBL) for providing the dataset of Phospho.ELM (Version 5.0) for this study. We are also grateful to Dr. Predrag Radivojac (School of Informatics, Indiana University) for distributing the training datasets of DISPHOS1.3 and valuable suggestions regarding the comparison of two phosphorylation site predictors. This research was supported by Program for New Century Excellent Talents in University (NCET-06-0116).

References

Beasly,D., Bull,D.R. and Martin,R.R. (1993) *University Computing*, **15**, 170–181.

Berry,E.A., Dalby,A.R. and Yang,Z.R. (2004) *Comput. Biol. Chem.*, **28**, 75–85.

Blom,N., Gammeltoft,S. and Brunak,S. (1999) *J. Mol. Biol.*, **294**, 1351–1362.

Blom,N., Sicheritz-Ponten,T., Gupta,R., Gammeltoft,S. and Brunak,S. (2004) *Proteomics*, **4**, 1633–1649.

Cai,C.Z., Wang,W.L., Sun,L.Z. and Chen,Y.Z. (2003) *Math. Biosci.*, **185**, 111–122.

Diella,F., Cameron,S., Gemund,C., Linding,R., Via,A., Kuster,B., Sicheritz-Ponten,T., Blom,N. and Gibson,T.J. (2004) *BMC Bioinf.*, **5**, 79.

Dobson,P.D. and Doig,A.J. (2003) *J. Mol. Biol.*, **330**, 771–783.

Groban,E.S., Narayanan,A. and Jacobson,M.P. (2006) *PLoS Comput. Biol.*, **2**, e32.

Han,P., Zhang,X., Norton,R.S. and Feng,Z.P. (2006) *J. Comput. Biol.*, **13**, 1579–1590.

Heckerling,P.S., Gerber,B.S., Tape,T.G. and Wigton,R.S. (2004) *Artif. Intell. Med.*, **30**, 71–84.

Hjerrild,M. and Gammeltoft,S. (2006) *FEBS Lett.*, **580**, 4764–4770.

Hjerrild,M., Stensballe,A., Rasmussen,T.E., Kofoed,C.B., Blom,N., Sicheritz-Ponten,T., Larsen,M.R., Brunak,S., Jensen,O.N. and Gammeltoft,S. (2004). *J. Proteome Res.*, **3**, 426–433.

Hubbard,M.J. and Cohen,P. (1993) *Trends Biochem. Sci.*, **18**, 172–177.

Huang,H.D., Lee,T.Y., Tzeng,S.W. and Horng,J.T. (2005a) *Nucleic Acids Res.*, **33**, W226–W229.

Huang,H.D., Lee,T.Y., Tzeng,S.W., Wu,L.C., Horng,J.T., Tsou,A.P. and Huang,K.T. (2005b) *J. Comput. Chem.*, **26**, 1032–1041.

Iakoucheva,L.M., Radivojac,P., Brown,C.J., O'Connor,T.R., Sikes,J.G., Obradovic,Z. and Dunker,A.K. (2004) *Nucleic Acids Res.*, **32**, 1037–1049.

Jones,D.T. (1999) *J. Mol. Biol.*, **292**, 195–202.

Kim,J.H., Lee,J., Oh,B., Kimm,K. and Koh,I. (2004) *Bioinformatics*, **20**, 3179–3184.

Kobe,B., Kampmann,T., Forwood,J.K., Listwan,P. and Brinkworth,R.I. (2005) *Biochim. Biophys. Acta*, **1754**, 200–209.

Matthews,B.W. (1975) *Biochim. Biophys. Acta*, **405**, 442–451.

Obenauer,J.C., Cantley,L.C. and Yaffe,M.B. (2003) *Nucleic Acids Res.*, **31**, 3635–3641.

Ofran,Y., Punta,M., Schneider,R. and Rost,B. (2005) *Drug Discov. Today*, **10**, 1475–1482.

Plewczynski,D., Tkacz,A., Godzik,A. and Rychlewski,L. (2005) *Cell Mol. Biol. Lett.*, **10**, 73–89.

Plewczynski,D., Tkacz,A., Wyrwicz,L.S. and Rychlewski,L. (2005) *Bioinformatics*, **21**, 2525–2527.

Qian,N. and Sejnowski,T.J. (1988) *J. Mol. Biol.*, **202**, 865–884.

Radivojac,P., Iakoucheva,L.M., Oldfield,C.J., Obradovic,Z., Uversky,V.N. and Dunker,A.K. (2007) *Biophys. J.*, **92**, 1439–1456.

Rost,B. and Sander,C. (1993) *J. Mol. Biol.*, **232**, 584–599.

Salomon,R. (1998) *IEEE Trans. Evolution. Comput.*, **2**, 45–55.

Schwartz,D. and Gygi,S.P. (2005) *Nat. Biotechnol.*, **23**, 1391–1398.

Sexton,R.S. and Dorsey,R.E. (2000) *Decision Support Syst.*, **30**, 11–22.

Sexton,R.S., Dorsey,R.E. and Johnson,J.D. (1999) *Eur. J. Oper. Res.*, **114**, 589–601.

Songyang,Z., Blechner,S., Hoagland,N., Hoekstra,M.F., Piwnica-Worms,H. and Cantley,L.C. (1994) *Curr. Biol.*, **4**, 973–982.

Ward,J.J., Sodhi,J.S., McGuffin,L.J., Buxton,B.F. and Jones,D.T. (2004) *J. Mol. Biol.*, **337**, 635–645.

Wong,M.H., Chung,T.S. and Wong,Y.K. (2000) *Microprocess. Microsyst.*, **24**, 251–262.

Xue,Y., Li,A., Wang,L., Feng,H. and Yao,X. (2006) *BMC Bioinf.*, **7**, 163.

Zhang,H., Zha,X., Tan,Y., Hornbeck,P.V., Mastrangelo,A.J., Alessi,D.R., Polakiewicz,R.D. and Comb,M.J. (2002) *J. Biol. Chem.*, **277**, 39379–39387.

Zhang,Z., Kochhar,S. and Grigorov,M.G. (2005) *Protein Sci.*, **14**, 431–444.

Revised April 3, 2007; accepted June 21, 2007

Edited by Andrej Sali