Hindawi Publishing Corporation
BioMed Research International
Volume 2014, Article ID 438341, 10 pages
http://dx.doi.org/10.1155/2014/438341

Research Article

Prediction of S-Nitrosylation Modification Sites Based on Kernel Sparse Representation Classification and mRMR Algorithm

Guohua Huang,[1,2] Lin Lu,[3] Kaiyan Feng,[4] Jun Zhao,[3] Yuchao Zhang,[5,6] Yaochen Xu,[7] Ning Zhang,[8] Bi-Qing Li,[9] Weiping Huang,[2] and Yu-Dong Cai[1]

[1] *Institute of Systems Biology, Shanghai University, Shanghai 200444, China*
[2] *Department of Mathematics, Shaoyang University, Shaoyang, Hunan 422000, China*
[3] *School of Biomedical Engineering, Shanghai Jiaotong University, Shanghai 200240, China*
[4] *Shanghai Center for Bioinformation Technology, Shanghai 200235, China*
[5] *Graduate School of the Chinese Academy of Sciences, Beijing 100049, China*
[6] *State Key Laboratory of Medical Genomics, Institute of Health Sciences, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai Jiao Tong University School of Medicine, Shanghai 200025, China*
[7] *East China Normal University Software Engineering Institute, Shanghai 200062, China*
[8] *Department of Biomedical Engineering, Tianjin University, Tianjin Key Lab of BME Measurement, Tianjin 300072, China*
[9] *Key Laboratory of Systems Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, China*

Correspondence should be addressed to Yu-Dong Cai; cai_yud@126.com

Received 19 June 2014; Accepted 23 July 2014; Published 12 August 2014

Academic Editor: Tao Huang

Protein S-nitrosylation plays a very important role in a wide variety of cellular biological activities. Hitherto, accurate prediction of S-nitrosylation sites is still of great challenge. In this paper, we presented a framework to computationally predict S-nitrosylation sites based on kernel sparse representation classification and minimum Redundancy Maximum Relevance algorithm. As much as 666 features derived from five categories of amino acid properties and one protein structure feature are used for numerical representation of proteins. A total of 529 protein sequences collected from the open-access databases and published literatures are used to train and test our predictor. Computational results show that our predictor achieves Matthews' correlation coefficients of 0.1634 and 0.2919 for the training set and the testing set, respectively, which are better than those of k-nearest neighbor algorithm, random forest algorithm, and sparse representation classification algorithm. The experimental results also indicate that 134 optimal features can better represent the peptides of protein S-nitrosylation than the original 666 redundant features. Furthermore, we constructed an independent testing set of 113 protein sequences to evaluate the robustness of our predictor. Experimental result showed that our predictor also yielded good performance on the independent testing set with Matthews' correlation coefficients of 0.2239.

1. Introduction

Nitric oxide (NO) has been reported to be an important signaling molecule which involves physiological and pathophysiological regulations of some cellular processes, such as cardiovascular, respiratory, gastrointestinal, reproductive, and host defense [1–4]. Protein S-nitrosylation which is covalently modified by NO has recently been discovered to play important roles in regulating diverse pathways [5–7] and other biological activities [8], such as chromatin remodeling [9], transcriptional regulation [10], cellular trafficking [11], and apoptosis [12]. Also, it has been reported that aberrant S-nitrosylation might contribute to some diseases such as neurodegenerative disorders [1, 13] and cancers [14]. Several biochemical approaches have been developed to identify S-nitrosylation sites; for example, Forrester et al. [15] used RAC

(resin-associated capture) method to isolate SNO protein, and Foster et al. [16] utilized an approach based on protein microarray to screen S-nitrosylation sites.

In contrast to time-consuming and labor-intensive experiments, computational approach is fast and cost-effective. It is reported that there have been at least 170 databases and computational tools concerned with posttranslational modification including protein S-nitrosylation modification [17]. With regard to predicting S-nitrosylation modification sites, Xue et al. [17] developed a software tool named GPS-SNO 1.0; Hao et al. [18] applied support vector machine (SVM), Lee et al. [19] used the maximal dependence decomposition- (MDD-) clustered SVMs, and Li et al. [20] utilized k-nearest neighbor algorithm to deal with the problem. Although computational approach is becoming more and more attractive, prediction of S-nitrosylation sites still remains a great challenge due to the complications of effectively protein encoding.

In the paper, we presented a new computational framework based on kernel sparse representation theory to predict S-nitrosylation sites. The framework consists of two steps: feature extraction and feature selection. Firstly, 666 features were extracted from five categories of amino acid properties, that is, sequence conservation, amino acid factor, secondary structure, solvent accessibility, and amino acid occurrence frequency, and one protein structure feature, the residual disorder. Then, a two-stage feature selection procedure was applied to select an optimal subset from the 666 redundant features. Finally, a webserver for the prediction of S-nitrosylation sites based on kernel sparse representation classification and minimum Redundancy Maximum Relevance algorithm is available at http://www.zhni.net/snopred/index.html.

2. Materials

The training and testing sets adopted in the paper were constructed as follows. A total of 645 protein sequences (see Supplementary Material S1 available online at http://dx.doi.org/10.1155/2014/438341) containing S-nitrosylation sites (see Supplementary Material S2) were first collected from open-access databases and the published literatures. Among the 645 protein sequences, 25 were from Uniprot database (version 2011_7) [21], 327 were from a research done by Xue et al. [17], and the other 293 protein sequences were from three recent reviews [22–24] on S-nitrosylation identification. The S-nitrosylation sites on the 645 protein sequences are all verified by experiments. Then, the sequence-clustering program CD-HIT [25] was applied to screen the 645 protein sequences. The cutoff value of CD-HIT was 0.4, meaning that the protein sequences having pairwise sequence identity greater than 40% to one another were removed. Finally, 529 protein sequences were left for analysis. Samples were then collected by taking peptides composed of 21 continuous residues with the central residue as cysteine; that is, peptides including a central cysteine and with each 10 residues in the upstream and downstream of the cysteine were picked out. For peptides with cysteine but which were less than 21 residuals, labels "X" were appended to end of the peptides. Thus, there were totally 2516 peptides obtained from the 529 proteins. 827 peptides with S-nitrosylation modification sites were labeled as positive samples and the remaining 1689 peptides were labeled as negative ones. More detailed information about collecting data can be found in our previous work [20]. The 2516 samples were grouped into training dataset and testing dataset at the ratio of 4 : 1; that is, we used 80% of the samples as the training samples, because sufficient samples were needed to train the predictor. Meanwhile, to evaluate the robustness, 20% of the samples were left for the testing. During sample grouping, positive samples and negative samples are distributed in a way so that the ratios of positive-to-negative samples in the training and testing datasets remained the same as that of the whole dataset which is about 1 : 2 (positive-to-negative ratio was 827 : 1689 in the whole date set). Consequently, the training set was composed of 662 positive and 1351 negative samples, and the testing set was composed of 165 positive and 338 negative samples (see Supplementary Materials S3 and S4).

Besides the training and testing sets mainly collected from published literatures, we also constructed an independent testing set with the Uniprot database of the latest version (version 2014_05). We searched the Uniprot database for those protein sequences with S-nitrosylation identification. Then, by deleting the proteins which had been used in the training and testing sets, totally 113 sequences containing S-nitrosylation sites were obtained. The 113 sequences were used as the independent testing set (see Supplementary Material S6). Thus, we could do comparison between different methods based on the independent testing set.

3. Methods

3.1. Feature Extraction. All features were derived from five categories of amino acid properties and one protein structure feature: (1) evolutionary conservation, (2) physicochemical or biochemical properties, (3) solvent accessibilities, (4) frequency around nitrosylated cysteine, (5) secondary structural properties, and (6) disorder status.

The evolutionary conservation of amino acid is very important, which is generally represented as the probability that it would mutate into other 20 kinds of amino acid. By using PSI-BLAST program [26], a $21 \times 20 = 420$ dimensional vector describing conservation of each peptide was obtained.

Physicochemical or biochemical properties of amino acid were characterized quantitatively as a 5-dimensional vector using amino acid index database [27], whose elements represent properties of polarity, secondary structure, molecular volume, codon diversity, and electrostatic charge, respectively. Except the cysteine, 20 amino acids in a peptide were represented as a 100-dimensional vector.

Disorder status of amino acid was quantified as a disorder score by the predictor of protein disorder [28], and thus, for a peptide, its disorder status was represented by a 21-dimensional vector.

Secondary structural properties, that is, "helix," "strand," and "others," and the solvent accessibility, that is, "buried" and "exposed," of an amino acid were calculated by the predicting software of protein structure and structural feature [29], resulting in a 5-dimensional encoding vector consisting of

0 or 1. A $21 \times 5 = 105$ dimensional vector represented the secondary structural and solvent accessibility properties of a peptide.

Frequency of the twenty amino acids around nitrosylated cysteine (nitrosylation site was excluded) was also taken into consideration.

Hence, each sample could be represented as a numerical vector containing as many as 666 (420 + 100 + 21 + 105 + 20) features. Table 1 shows the distribution of features. Details of feature construction could be found in our previous work [20].

3.2. Feature Selection. A two-stage feature selection procedure is used to select optimal feature subset from the feature space. The predictor constructed by the optimal feature subset is our final S-nitrosylation sites predictor. The procedure is described as follows.

Stage 1. All features are evaluated by the minimum Redundancy Maximum Relevance (mRMR) algorithm [30] and then ranked according to their mRMR scores.

Stage 2. Based on the mRMR evaluation, incremental feature selection procedure [31, 32] is adopted to search for the optimal feature subset with the help of kernel sparse representation classification (KSRC) algorithm.

3.2.1. mRMR Algorithm. The mRMR algorithm proposed by Peng et al. [30] is a feature evaluation method based on mutual information. Mutual information is able to quantify the dependency between two variables. The larger the mutual information is, the more the dependency between the two variables is. Mutual information between two random variables X and Y is defined as follows:

$$\mathrm{MI}(X,Y) = \iint p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)} dx\,dy, \tag{1}$$

where function p denotes probabilistic or joint probabilistic density.

Mutual information between the feature space $\Omega = (X_1, X_2, \ldots, X_k)$ and the target variable Y is defined as follows:

$$\begin{aligned}
&\mathrm{MI}(\Omega, Y) \\
&= \iint p(\Omega, y) \log \frac{p(\Omega, y)}{p(\Omega)\,p(y)} ds\,dy \\
&= \int_{k+1} \cdots \iint p(x_1, x_2, \ldots, x_k, y) \\
&\qquad \times \log \frac{p(x_1, x_2, \ldots, x_k, y)}{p(x_1, x_2, \ldots, x_k)\,p(y)} dx_1\,dx_2 \cdots dx_k\,dy.
\end{aligned} \tag{2}$$

The mRMR algorithm aims to evaluate feature subsets S and then selects the optimal feature subset that meets the minimal redundancy and maximal relevance criteria, that is, the minimal dependency to the entire feature space and the maximal dependency to the target variable Y. Minimal redundancy to the entire feature space can be calculated by the following equation:

$$\min_{S \subseteq \Omega} \frac{1}{|S|^2} \sum_{X_i, X_j \in S} \mathrm{MI}\left(X_j, X_i\right). \tag{3}$$

Maximal dependency to the target variable Y can be calculated by the following equation:

$$\max_{S \subseteq \Omega} \frac{1}{|S|} \sum_{X_j \in S} \mathrm{MI}\left(X_j, Y\right). \tag{4}$$

Thus, the mRMR evaluation can be quantified as score by integrating (3) and (4) into the following equation:

$$\max_{S \subseteq \Omega} \left\{ \frac{1}{|S|} \sum_{X_j \in S} \mathrm{MI}\left(X_j, Y\right) - \frac{1}{|S|^2} \sum_{X_i, X_j \in S} \mathrm{MI}\left(X_j, X_i\right) \right\}. \tag{5}$$

3.2.2. Incremental Feature Selection. In the implementation, the mRMR criterion is hard to satisfy, especially when the feature space is large. Hence, to attain an optimal feature subset of minimal redundancy and maximal relevance, a heuristic strategy named incremental feature selection [31, 32] is adopted for the search of feature subset.

Firstly, all the features are scored by (5), by shrinking feature subset S to contain only one feature. Secondly, arrange all the features according to their mRMR scores. Thirdly, search for optimal feature subset by an increment means as follows.

Suppose all the features in the feature space Ω have been arranged in the order from high mRMR score to low mRMR score. Beginning from the feature of the highest mRMR score, move features from the scored feature space to the selected feature subset sequentially. When one feature is added, evaluate the classification performance of the feature subset by predictors which are constructed by the KSRC algorithm (see Section 3.2.4 for details). Finally, the feature subset of the highest classification performance is selected as the optimal feature subset and the predictor constructed by the optimal feature subset is the final predictor. In this study, the method used to evaluate the classification performance is presented in Section 3.2.3.

Table 1: Distribution of feature type for a sample.

Feature category	Number of features from each category
Evolutionary conservation	21×20
Amino acid factor	20×5
Secondary structure	21×3
Solvent accessibility	21×2
Amino acid frequency	20×1
Disorder	21×1
Number of features of a sample	666

3.2.3. Evaluation Metrics. Four indicators, sensitivity (SN), specificity (SP), accuracy (ACC), and Matthews' correlation coefficient (MCC), are used to evaluate the performance of predictors when new features are added. Consider the following:

$$\begin{aligned} \mathrm{SN} &= \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})}, \\ \mathrm{SP} &= \frac{\mathrm{TN}}{(\mathrm{TN}+\mathrm{FP})}, \\ \mathrm{ACC} &= \frac{(\mathrm{TP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{FN}+\mathrm{FP}+\mathrm{TN})}, \\ \mathrm{MCC} &= \frac{(\mathrm{TP}\times\mathrm{TN}-\mathrm{FN}\times\mathrm{FP})}{\sqrt{(\mathrm{TP}+\mathrm{FN})\,(\mathrm{TP}+\mathrm{FP})\,(\mathrm{TN}+\mathrm{FP})\,(\mathrm{TN}+\mathrm{FN})}}. \end{aligned} \tag{6}$$

TP and TN represent the numbers of true positive and true negative, respectively. FP and FN represent the numbers of false positive and false negative, respectively. Among the four indicators, MCC is the most significant indicator, which is used to optimize the procedure of feature selection in this study.

3.2.4. KSRC Algorithm. In this paper, KSRC algorithm is applied to construct predictor. The KSRC algorithm integrates the sparse representation classification (SRC) algorithm and the kernel function technique to fulfill classification task [33, 34]. In the following section, we will introduce the SRC algorithm and the kernel function technique, respectively, and then illustrate how to integrate the two techniques.

In the recent years, the SRC algorithm has been successfully applied in these fields of signal recovery, signal encoding, and signal classification [33–41]. The principle underlying the SRC algorithm is that testing samples can be represented as linear combination of training samples if the testing and training samples belong to the same category so that the representation coefficient of a testing sample under all training samples might supply sufficient information to determine the category of the testing samples.

Suppose there are c distinct classes, each with n_k samples, $k = 1, 2, \ldots, c$. And $X^k = (x_1^k, x_2^k, \ldots, x_{n_k}^k)$ is a matrix consisting of samples from the kth class, where x_j^k $(1 \le j \le n_k)$ is a column vector, representing the jth sample in the class k. All training samples are concatenated to form a matrix $X = [X^1, X^2, \ldots, X^c]$. Computing the sparsest coefficient vector α of a test sample y under the matrix X is modeled as follows:

$$\min \|\alpha\|_0, \quad \text{subject to } y = X\alpha \tag{7}$$

or

$$\min \|\alpha\|_0, \quad \text{subject to } \|y - X\alpha\|_2 \le \varepsilon, \tag{8}$$

where operator $\|\bullet\|_0$ denotes the l_0 norm, which counts nonzero entries, and operator $\|\bullet\|_2$ denotes the l_2 norm of a vector, respectively.

Since the pursuit of exact solution of (7) and (8) is an NP-hard problem [42], the orthogonal matching pursuit (OMP) [43, 44] algorithm is used to seek an approximate solution to (7) and (8) in our works. The OMP is an iterative greedy method. Each step of iteration in OMP algorithm contains three operations: (1) computing residual referring to difference between original signal and recovery one, (2) selecting the column with the highest correlation to the current residual, and (3) projecting original signal into the linear subspace spanned by these already selected columns. For convenient description, the following symbols were used. The symbol X specified a matrix, X_t referred to the column t in the matrix, and X_Θ consisted of columns of the matrix X with the indices Θ. The OMP algorithm is described in Algorithm 1.

Once a coefficient vector α was gained by the OMP algorithm, the category of the corresponding testing sample was determined by the following rule:

$$K = \arg\min_{k=1,2,\ldots,c} \|y - X\alpha_k\|_2, \tag{9}$$

where $\alpha_k = (0, 0, \ldots, 0, \alpha_1^k, \alpha_2^k, \ldots, \alpha_{n_k}^k, \ldots, 0)$ was a coefficient whose entries were all zero except α_i^k $(1 \le i \le n_k)$ which corresponds to the samples from the class k and is equal to the corresponding element from α. The details of the SRC algorithm were shown in Algorithm 2.

Nevertheless, the performance of the SRC algorithm might be limited, if the testing samples are not linearly representable in the space of training sample [34]. Therefore, in our work, kernel function technique is applied to project testing sample into higher-dimensional space so as to alter the distributed structures of the samples.

Kernel function technique is a widely used technique that is able to map data from low-dimensional space to higher-dimensional space [34]. A well-chosen kernel function enables original linearly inseparable samples to become linearly separable in the high-dimensional feature space. In our work, the Laplacian kernel function $\Psi(x, y) = e^{-|x-y|/\delta}$ was employed.

Assume that the training samples with c classes $X = [X^1, X^2, \ldots, X^c] = [x_1, x_2, \ldots, x_n]$ as previously shown and the testing sample y are mapped to high-dimensional data $\Psi(X) = [\Psi(X^1), \Psi(X^2), \ldots, \Psi(X^c)] = [\Psi(x_1), \Psi(x_2), \ldots, \Psi(x_n)]$ and $\Psi(y)$, respectively. Similar to (7), the problem with the sparest coefficient representation of $\Psi(y)$ under $\Psi(X)$ was formulated as follows:

$$\min \|\alpha\|_0, \quad \text{subject to } \Psi(y) = \Psi(X)\,\alpha. \tag{10}$$

Let $\Pi = [\Psi(x_1), \Psi(x_2), \ldots, \Psi(x_n)]^T$ be a column vector. Equation $\Psi(y) = \Psi(X)\alpha$ left multiplied by Π was rewritten as

$$\begin{bmatrix} \Psi(y)\,\Psi(x_1) \\ \vdots \\ \Psi(y)\,\Psi(x_n) \end{bmatrix} = \begin{bmatrix} \Psi(x_1)\,\Psi(x_1) & \cdots & \Psi(x_1)\,\Psi(x_n) \\ \vdots & \cdots & \vdots \\ \Psi(x_n)\,\Psi(x_1) & \cdots & \Psi(x_n)\,\Psi(x_n) \end{bmatrix} \alpha. \tag{11}$$

Input: the matrix X, the sparsity k, the testing sample y
Output: the coefficient vector α
(1) initialize residual $e = y$, $\Theta = \Phi$, $i = 0$
(2) normalize columns of the matrix X with the l_2 norm
(3) while $i < k$
$j = \arg\max_{t \notin \Theta} \{X_t' e\}$
$\Theta = \Theta \cup \{j\}$
$P = X_\Theta (X_\Theta' X_\Theta)^{-1} X_\Theta'$ //compute the projection
$e = (y - Py)$ //update the residual
$i = i + 1$ //update the loop index
(4) $\alpha = Py$

Algorithm 1: OMP algorithm.

Input: the training set with c distinct classes, the test sample y
Output: the category of the testing sample y
(1) Concatenate all training samples to construct the matrix X
(2) normalize columns of the matrix X with the l_2 norm
(3) solve (7) or (8) using the OMP in Algorithm 1, and obtain the coefficient vector α
(4) determine the category of the testing sample according to (9)

Algorithm 2: SRC algorithm.

According to the properties of kernel function, (11) is further expressed as

$$\begin{bmatrix} \Psi(y, x_1) \\ \vdots \\ \Psi(y, x_n) \end{bmatrix} = \begin{bmatrix} \Psi(x_1, x_1) & \cdots & \Psi(x_1, x_n) \\ \vdots & \cdots & \vdots \\ \Psi(x_n, x_1) & \cdots & \Psi(x_n, x_n) \end{bmatrix} \alpha. \quad (12)$$

Therefore, minimum equation (10) is equivalent to

$$\min \|\alpha\|_0, \quad \text{subject to (8)}. \quad (13)$$

Equation (13) has the same solution as (10). The KSRC was shown in Algorithm 3.

4. Results and Discussion

4.1. Optimal Feature Subset Selection. First, the mRMR algorithm [30] was applied to the training set, producing a sequence of 666 scored features. Details of the results can be found in Supplementary Material S5.

Second, apply incremental feature selection procedure to search optimal feature subset. Figure 1 shows MCC values of each candidate feature subset by using 10-fold cross validation on the training set. The best MCC value is 0.1634, corresponding to the combination of the first 134 features. Therefore, this candidate feature subset was regarded as the optimal subset.

In the implementation, the factor δ of the Laplacian kernel function in the KSRC algorithm is 100. The sparsity k in OMP algorithm was 50. The used OMP algorithm codes are available at the following site: http://www.cs.technion.ac.il/~ronrubin/software.html [45]. The used mRMR codes are available at http://penglab.janelia.org/proj/mRMR/ [30].

4.2. Comparison with Other Algorithms. As was mentioned in Section 1, quite a few methods have been developed to predict the S-nitrosylation sites in recent years. However, it was difficult to make direct comparisons between them due to the following two reasons. First, different methods usually employed different datasets. It was biased to compare their overall performances based on different datasets. Secondly, we did not know what parameters they used to optimize the predictors. So, it was difficult for us to compare other methods with ours based on the same training and testing datasets.

Notwithstanding this, we attempted to compare our methods with other data mining methods based on our training and testing datasets. Hence, the KSRC algorithm proposed in this paper was compared to five other data mining algorithms: SRC [38], k-nearest neighbor algorithm (KNN) [46], random forest (RF) [47], sequential minimal optimization (SMO) [48], and Dagging [49]. KNN is an instance-based learning algorithm, which is widely used due to its simplicity and efficiency in training. RF is an integration method by combining many tree predictors together. Each tree predictor performs computation based on the values of a random vector sampled independently and with the same distribution for all trees in the forest. SMO is an algorithm that trains the support vector machine. Dagging is an algorithm that ensembles weak classifiers. In terms of implementation, KSRC and SRC were coded in Matlab language by virtue of the OMP package [45]. The computation of KNN, RF, SMO, and Dagging algorithms was performed by Weka (version 3-6-1) [50], which is a collection of learning machine algorithms and is available at http://www.cs.waikato.ac.nz/ml/weka/. In this work, the number of the nearest neighbors in the KNN is 3. The RF, SMO, and Dagging use the default parameters in the Weka. The sparsity of the OMP in the SRC is 50, the

Input: the training set with c distinct classes $\{x_i\}_{i=1}^{n}$, the testing sample y, and kernel function Ψ
Output: the category of the test sample
(1) compute the matrix D and the test sample W such that $W = D\alpha$ by using (12)
(2) normalize columns of the matrix D with the l_2 norm
(3) solve (13) using the OMP in Algorithm 1, and obtain the coefficient vector α
(4) compute $K = \arg\min_{k=1,2,\ldots,c} \|W - D\alpha_k\|_2$
(5) assign the testing sample to the class K

Algorithm 3: KSRC algorithm.

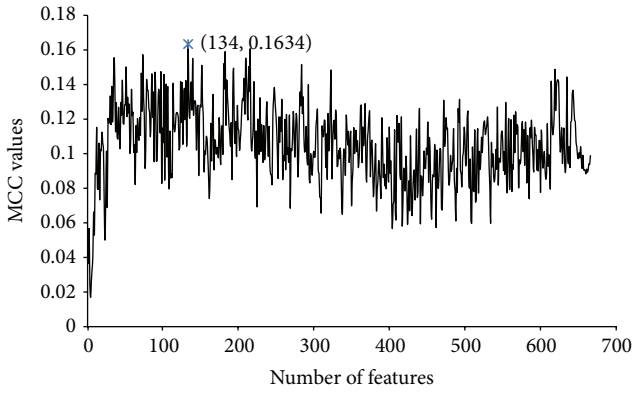

Figure 1: MCC value of 10-fold cross validation of the KSRC on the training set in the incremental feature selection procedure.

same as that of the KSRC. All the computer programs were executed on the Operation System platform Fedora 17.

The four indicators, SN, SP, ACC, and MCC, mentioned in Section 3.2.3, were also used for the comparison of different algorithms. The MCC curves of SRC, KNN, RF, SMO, and Dagging on the training set were plotted in Figure 2. The five algorithms attained optimal feature subsets containing 76, 52, 38, 127, and 103 features, respectively. All six algorithms were compared both on the training set and on the testing set with optimal feature subsets of their own. Tables 2 and 3 showed their performances on the training and testing datasets, respectively. As indicated by Table 2 and Figure 2, KSRC could achieve MCC that exceeded 0.16 on the training set. Although SMO and Dagging performed better in terms of the MCC, KSRC showed better SN than that of SMO and Dagging. Table 3 presented the performances of the six algorithms on the testing dataset, which were not previously used in the training. As shown in Table 3, KSRC yielded the highest MCC and SN among all of the six algorithms, while SMO and Dagging showed poor MCC on the testing set. The high MCC and SN of KSRC on both the training and testing datasets indicated that KSRC was more effective and robust than the other five data mining algorithms.

To compare the predictive performances of the 134 optimal features with that of the original 666 features, the 10-fold cross validation and independent tests were also conducted on the training and testing sets by the 666 original features, respectively. Table 4 shows the performance of using original

Table 2: Performances of six algorithms on the training set with the respective optimal features using 10-fold cross validation.

	SN	SP	ACC	MCC
KSRC	0.4048	0.7543	0.6393	0.1634
SRC	0.3489	0.7876	0.6433	0.1467
KNN	0.3852	0.7469	0.6279	0.1358
RF	0.3399	0.7957	0.6458	0.1473
SMO	0.2840	0.8705	0.6776	0.1887
Dagging	0.3610	0.8320	0.6771	0.2150

KSRC: kernel sparse representation classification; SRC: sparse representation classification; KNN: k-nearest neighbor algorithm; RF: random forest method; SMO: sequential minimal optimization; Dagging refers to the use of majority vote to combine multiple models derived from a single learning algorithm using disjoint samples.

Table 3: Performances of six algorithms on the testing set with the respective optimal features.

	SN	SP	ACC	MCC
KSRC	0.4727	0.8077	0.6978	0.2919
SRC	0.2909	0.7988	0.6322	0.1000
KNN	0.4061	0.7899	0.6649	0.2062
RF	0.3636	0.8343	0.6799	0.2206
SMO	0.2364	0.8669	0.6600	0.1299
Dagging	0.2848	0.8343	0.6541	0.1386

Table 4: Performances of KSRC on the training and testing sets with the original 666 features.

	SN	SP	ACC	MCC
The training set	0.2749	0.8120	0.6354	0.0991
The testing set	0.2909	0.8462	0.6640	0.1612

666 features on the training and testing sets, respectively. It can be seen in Table 4 that SN and MCC with the 134 optimal features were much better than those of the original features, though SP is a bit worse. Since the MCC is the most important criterion among the adopted metrics, we conclude that the 134 optimal features performed better than the original 666 features.

4.3. Comparison of Algorithms on Independent Testing Set. Since the training and testing sets were mainly collected from published literatures, we constructed an independent

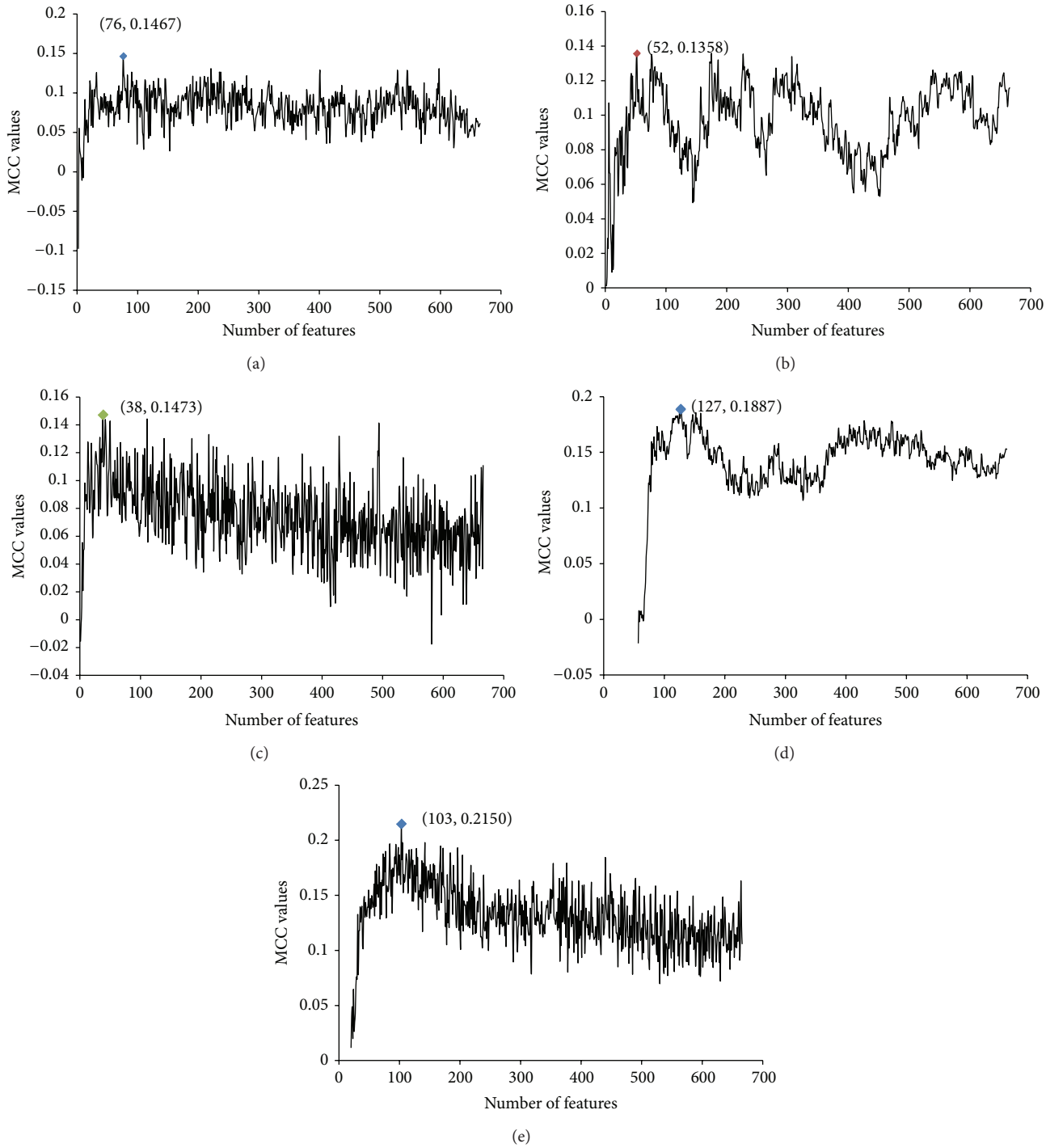

Figure 2: MCC curves of 10-fold cross validation on the training set of (a) SRC, (b) KNN, (c) RF, (d) SMO, and (e) Dagging in the incremental feature selection procedure.

testing set for the comparison between our method and other methods. The independent testing set contained 113 protein sequences from the latest version of Uniprot database (version 2014_05) (see Section 2 for details). Two existing S-nitrosylation predictors, iSNO-AAPair [51] and iSNO-PseAAC [52], were used for comparison. The comparison results of our predictor, iSNO-AAPair, iSNO-PseAAC, and other five data mining algorithms on the independent testing set were presented in Table 5. As shown in Table 5, the SRC algorithm achieved the highest MCC of 0.2617, and our proposed KSRC algorithm was the second with MCC of 0.2239. The iSNO-AAPair and iSNO-PseAAC predictors attained MCC of 0.1125 and 0.1190, respectively, both of which were only approximately half of the KSRC algorithm.

Table 5: Performances of eight algorithms on the independent testing set with the respective optimal features.

	SN	SP	ACC	MCC
KSRC	0.5196	0.7368	0.6915	0.2239
SRC	0.5588	0.7419	0.7038	0.2617
KNN	0.4069	0.7419	0.6721	0.1333
RF	0.4657	0.7535	0.6936	0.1958
SMO	0.1765	0.8645	0.7211	0.0474
Dagging	0.2745	0.7884	0.6813	0.0612
iSNO-AAPair	0.4020	0.7252	0.6578	0.1125
iSNO-PseAAC	0.5343	0.6103	0.5945	0.1190

Although the MCC of KSRC algorithm was a little lower than that of SRC algorithm, the KSRC algorithm was the one algorithm that could achieve high and stable performance in both of the testing set and the independent set (as shown in Tables 3 and 5), demonstrating the robustness of the KSRC algorithm among different datasets.

5. Conclusions

In the paper, we proposed a framework based on the KSRC to computationally identify S-nitrosylation modification sites. Our experimental results show that KSRC outperforms other state-of-the-art algorithms in terms of the key prediction metrics. The KSRC is an application of kernel function technique to the SRC. Kernel approach can project linearly inseparable samples into high-dimensional feature space with the use of kernel functions. If an appropriate kernel function is selected, the original linearly inseparable samples could become linearly separable in the high-dimensional feature space. Kernelizing of the sparse representation by Laplacian function could improve the separability of the samples and yields higher MCC than those linear classification algorithms, such as KNN and SRC. We believe that the proposed KSRC based framework could become a helpful tool for the prediction and analyses of protein S-nitrosylation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

Guohua Huang and Lin Lu contributed equally to this paper.

Acknowledgments

This work was supported by Grants from National Basic Research Program of China (2011CB510101 and 2011CB510102), National Natural Science Foundation of China (31371335), Innovation Program of Shanghai Municipal Education Commission (12ZZ087), the Grant of "The First-Class Discipline of Universities in Shanghai," Scientific Research Fund of Hunan Provincial Science and Technology Department (2014FJ3013 and 2011FJ3197), Hunan National Science Foundation (11JJ5001), Scientific Research Fund of Hunan Provincial Education Department (11C1125), Tianjin Research Program of Application Foundation and Advanced Technology (14JCQNJC09500), the Seed Foundation of Tianjin University (60302069), and the National Research Foundation for the Doctoral Program of Higher Education of China (20130032120070).

References

[1] M. W. Foster, T. J. McMahon, and J. S. Stamler, "S-nitrosylation in health and disease," *Trends in Molecular Medicine*, vol. 9, no. 4, pp. 160–168, 2003.

[2] B. Derakhshan, P. C. Wille, and S. S. Gross, "Unbiased identification of cysteine S-nitrosylation sites on proteins," *Nature Protocols*, vol. 2, no. 7, pp. 1685–1691, 2007.

[3] D. T. Hess, A. Matsumoto, S. Kim, H. E. Marshall, and J. S. Stamler, "Protein S-nitrosylation: purview and parameters," *Nature Reviews Molecular Cell Biology*, vol. 6, no. 2, pp. 150–166, 2005.

[4] M. Liu, J. E. Talmadge, and S.-J. Ding, "Development and application of site-specific proteomic approach for study protein S-nitrosylation," *Amino Acids*, vol. 42, no. 5, pp. 1541–1551, 2012.

[5] E. J. Whalen, M. W. Foster, A. Matsumoto et al., "Regulation of β-adrenergic receptor signaling by S-nitrosylation of G-protein-coupled receptor kinase 2," *Cell*, vol. 129, no. 3, pp. 511–522, 2007.

[6] E. Nozik-Grayck, E. J. Whalen, J. S. Stamler, T. J. McMahon, P. Chitano, and C. A. Piantadosi, "S-nitrosoglutathione inhibits α1-adrenergic receptor-mediated vasoconstriction and ligand binding in pulmonary artery," *The American Journal of Physiology—Lung Cellular and Molecular Physiology*, vol. 290, no. 1, pp. L136–L143, 2006.

[7] T. Kokkola, J. R. Savinainen, K. S. Mönkkönen, M. D. Retamal, and J. T. Laitinen, "S-nitrosothiols modulate G protein-coupled receptor signaling in a reversible and highly receptor-specific manner," *BMC Cell Biology*, vol. 6, no. 1, article 21, 2005.

[8] M. T. Forrester, M. W. Foster, M. Benhar, and J. S. Stamler, "Detection of protein S-nitrosylation with the biotin-switch technique," *Free Radical Biology and Medicine*, vol. 46, no. 2, pp. 119–126, 2009.

[9] A. Nott, P. M. Watson, J. D. Robinson, L. Crepaldi, and A. Riccio, "S-nitrosylation of histone deacetylase 2 induces chromatin remodelling in neurons," *Nature*, vol. 455, no. 7211, pp. 411–415, 2008.

[10] F. Li, P. Sonveaux, Z. N. Rabbani et al., "Regulation of HIF-1α stability through S-Nitrosylation," *Molecular Cell*, vol. 26, no. 1, pp. 63–74, 2007.

[11] K. Ozawa, E. J. Whalen, C. D. Nelson et al., "S-nitrosylation of β-arrestin regulates β-adrenergic receptor trafficking," *Molecular Cell*, vol. 31, no. 3, pp. 395–405, 2008.

[12] M. Benhar, M. T. Forrester, D. T. Hess, and J. S. Stamler, "Regulated protein denitrosylation by cytosolic and mitochondrial thioredoxins," *Science*, vol. 320, no. 5879, pp. 1050–1054, 2008.

[13] M. W. Foster, D. T. Hess, and J. S. Stamler, "Protein S-nitrosylation in health and disease: a current perspective," *Trends in Molecular Medicine*, vol. 15, no. 9, pp. 391–404, 2009.

[14] K. Lim, B. B. Ancrile, D. F. Kashatus, and C. M. Counter, "Tumour maintenance is mediated by eNOS," *Nature*, vol. 452, no. 7187, pp. 646–649, 2008.

[15] M. T. Forrester, J. W. Thompson, M. W. Foster, L. Nogueira, M. A. Moseley, and J. S. Stamler, "Proteomic analysis of S-nitrosylation and denitrosylation by resin-assisted capture," *Nature Biotechnology*, vol. 27, no. 6, pp. 557–559, 2009.

[16] M. W. Foster, M. T. Forrester, and J. S. Stamler, "A protein microarray-based analysis of S-nitrosylation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 45, pp. 18948–18953, 2009.

[17] Y. Xue, Z. Liu, X. Gao et al., "GPS-SNO: computational prediction of protein s-nitrosylation sites with a modified GPS algorithm," *PLoS ONE*, vol. 5, no. 6, Article ID e11290, 2010.

[18] G. Hao, B. Derakhshan, L. Shi, F. Campagne, and S. S. Gross, "SNOSID, a proteomic method for identification of cysteine S-nitrosylation sites in complex protein mixtures," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 4, pp. 1012–1017, 2006.

[19] T. Lee, Y. Chen, T. Lu, and H. Huang, "Snosite: exploiting maximal dependence decomposition to identify cysteine S-Nitrosylation with substrate site specificity," *PLoS ONE*, vol. 6, no. 7, Article ID e21849, 2011.

[20] B. Li, L. Hu, S. Niu, Y. Cai, and K. Chou, "Predict and analyze S-nitrosylation modification sites with the mRMR and IFS approaches," *Journal of Proteomics*, vol. 75, no. 5, pp. 1654–1665, 2012.

[21] Consortium TU, "The Universal Protein Resource (UniProt) in 2010," *Nucleic Acids Research*, vol. 38, pp. D142–D148, 2010.

[22] Y. W. Lam, Y. Yuan, J. Isaac, C. V. S. Babu, J. Meller, and S. Ho, "Comprehensive identification and modified-site mapping of S-nitrosylated targets in prostate epithelial cells," *PLoS ONE*, vol. 5, no. 2, Article ID e9075, 2010.

[23] P. Doulias, J. L. Greene, T. M. Greco et al., "Structural profiling of endogenous S-nitrosocysteine residues reveals unique features that accommodate diverse mechanisms for protein S-nitrosylation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 39, pp. 16958–16963, 2010.

[24] M. Liu, J. Hou, L. Huang et al., "Site-specific proteomics approach for study protein S-nitrosylation," *Analytical Chemistry*, vol. 82, no. 17, pp. 7160–7168, 2010.

[25] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[26] S. F. Altschul, T. L. Madden, A. A. Schäffer et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[27] S. Kawashima and M. Kanehisa, "AAindex: amino acid index database," *Nucleic Acids Research*, vol. 28, no. 1, p. 374, 2000.

[28] K. Peng, P. Radivojac, S. Vucetic, A. K. Dunker, and Z. Obradovic, "Length-dependent prediction of protein in intrinsic disorder," *BMC Bioinformatics*, vol. 7, article 208, 2006.

[29] J. Cheng, A. Z. Randall, M. J. Sweredoski, and P. Baldi, "SCRATCH: a protein structure and structural feature prediction server," *Nucleic Acids Research*, vol. 33, no. 2, pp. W72–W76, 2005.

[30] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[31] Z. He, J. Zhang, X. Shi et al., "Predicting drug-target interaction networks based on functional groups and biological features," *PLoS ONE*, vol. 5, no. 3, Article ID e9603, 2010.

[32] T. Huang, W. Cui, L. Hu, K. Feng, Y. Li, and Y. Cai, "Prediction of pharmacological and xenobiotic responses to drugs based on time course gene expression profiles," *PLoS ONE*, vol. 4, no. 12, Article ID e8126, 2009.

[33] L. Zhang, W. Zhou, P. Chang et al., "Kernel sparse representation-based classifier," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1684–1695, 2012.

[34] J. Yin, Z. Liu, Z. Jin, and W. Yang, "Kernel sparse representation based classification," *Neurocomputing*, vol. 77, no. 1, pp. 120–128, 2012.

[35] J. Wright, Y. Ma, S. Member, J. Mairal, and G. Sapiro, "Sparse representation for computer vision and pattern recognition," *Proceedings of IEEE*, vol. 98, no. 10, pp. 1031–1044, 2009.

[36] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, and S. Ji, "Discriminant sparse neighborhood preserving embedding for face recognition," *Pattern Recognition*, vol. 45, no. 8, pp. 2884–2893, 2012.

[37] Y. Hu, A. S. Mian, and R. Owens, "Face recognition using sparse approximated nearest points between image sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1992–2004, 2012.

[38] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[39] Q. Sami Ul Haq, L. Tao, F. Sun, and S. Yang, "A fast and robust sparse approach for hyperspectral data classification using a few labeled samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 6, pp. 2287–2302, 2012.

[40] H. F. Huang, G. S. Hu, and L. Zhu, "Sparse representation-based heartbeat classification using independent component analysis," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1235–1247, 2012.

[41] Y. Xu, W. Zuo, and Z. Fan, "Supervised sparse representation method with a heuristic strategy and face recognition experiments," *Neurocomputing*, vol. 79, pp. 125–131, 2012.

[42] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[43] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, 2011.

[44] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27th Asilomar Conference on Signals, Systems & Computers*, vol. 1, pp. 40–44, November 1993.

[45] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Tech. Rep., CS Technion, 2008.

[46] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.

[47] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[48] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.

[49] K. M. Ting and I. H. Witten, "Stacking bagged abd dagged models," in *Proceedings of the 14th international Conference on Machine Learning*, pp. 367–375, San Francisco, Calif, USA, 1997.

[50] M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, and I. H. Witten, "The WEKA data mining software?: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[51] Y. Xu, J. Ding, L. Wu, and K. Chou, "iSNO-PseAAC: predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition," *PLoS ONE*, vol. 8, no. 2, Article ID e55844, 2013.

[52] Y. Xu, J. Ding, L. Wu, and K. Chou, " iSNO-PseAAC: predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition," *PLoS ONE*, vol. 8, no. 2, Article ID e55844, 2013.