International Journal of Statistical Sciences *ISSN 1683-5603*
Vol. 18, 2019, pp 45-64

Computational Prediction of Protein S-nitrosylation Sites Mapping on Mus Musculus

Fee Faysal Ahmed[1], Md. Mehedi Hassan[2], Md. Hadiul Kabir[3], Mst. Shamima Khatun[2], Md. Parvez Mosharaf[3], Mohammad Ali Moni[4] and Md. Nurul Haque Mollah[3]*

[1]Department of Mathematics, Jashore University of Science and Technology, Jashore-7408, Bangladesh

[2]Department of Bioscience and Bioinformatics, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan

[3]Bioinformatics Lab., Department of Statistics, Rajshahi University, Rajshahi-6205, Bangladesh

[4]Discipline of Biomedical Science, School of Medical Sciences, University of Sydney, Sydney, New South Wales, Australia

*Correspondence should be addressed to Md. Nurul Haque Mollah (mollah.stat.bio@ru.ac.bd)

[Received June 23, 2019; Revised August 1, 2019; Accepted November 15, 2019]

Abstract

Prediction of the S-nitrosylation site is pivotal for understanding the mechanism of protein. The detection of S-nitrosylation site is a challenging task, since the experimental methods are laborious, time-wasting and expensive. There are some *in silico* existing approaches to identify S-nitrosylation sites. However, their performances are not so satisfactory. Therefore, our proposed method essential to improve or develop a good predictor for identifying S-nytrosylation sites. In this study, we have proposed a novel *in silico* predictor based on the random forest for identifying S-nitrosylation sites of *Mus Musculus* species. To develop a novel S-nitrosylation site predictor using the experimentally generated S-nitrosylation protein sequences of *Mus Musculus,* we have considered six different popular classifiers for a comparative study on the prediction of S-nitrosylation sites. We have observed that the random forest based predictor (proposed) is improved the performance over the other five classifiers based predictors. The

performance was measured by ROC curves, AUC and pAUC scores for both training and independent test datasets. We observed that our proposed predictor performs (AUC 0.759, Accuracy 75% and MCC 47%) much better than the other existing predictors under the 10-fold cross-validation. The proposed predictor also achieved an average performance of the AUC score of 0.788 and the accuracy score of 72.8% for test datasets. The output of the proposed method may also helpful to explore the SNO-related cellular functions in Human since *Mus Musculus* is evolutionary related to Human.

Keywords: Protein sequence, S-nitrosylation site, composition of k-spaced amino acid pairs (CKSAAP) encoding, Feature selection, Random forest (RF) and Building predictor.

AMS Classification: 92C40.

1. Introduction

S-nitrosylation (SNO) is the utmost ubiquitous post translational modification (PTM) of protein that contributes in regulating many cellular plasticity and dynamics (Foster et al., 2009). Under both normal and pathological conditions, SNO is the main chemical-mechanism in which nitric oxide regulates protein functions and exposed to change protein functions, subcellular localization and protein-protein interactions (Hess et al., 2005; Whalen et al., 2007). Many studies have shown that SNO proteins retrospectively and irregularly increases or decreases in a range of diseases (Lugovskoy et al.,1999; Foster et al., 2003; Nakamura et al., 2006; Schonhoff et al., 2006; Lipton et al.,1993). It can associate with stroke, Alzheimer, cancer and a number of chronic diseases. The use of traditional mass spectrometry-based proteomics has been suffering due to the essential chemical uncertainty of the SNO bond (Nakamura et al., 2006). So, detection of the SNO site is important for understanding the mechanism of SNO related biological functions. Generally, the SNO sites are experimentally identified by mass spectrometry using a biotin switch method to label the oxidized cysteines (Foster et al., 2003; Nakamura et al., 2006). The avobe detection process of SNO site is a challenging task, since the experimental methods are laborious, time-wasting and expensive. Therefore, a good in silico method which can reduce time, labor and cost for identifying SNO sites is required. There are some in silico approaches to identify SNO sites for different species including Mus Musculus (Li et al., 2011; Li et al., 2012; Xue et al., 2010; Xu et al., 2013). However, it has

been observed that their performances are not so satisfactory yet. Therefore, in this study, an attempt is made to propose a novel in silico predictor based on random forest (RF) for identifying SNO sites of Mus Musculus species. It should be noted here that the demo version of this new procedure was published in a conference proceeding (Ahmed et al., 2017).

To investigate the performance of the proposed RF based predictor using the previous experimentally generated SNO protein sequences of Mus Musculus, we have considered five other different popular classifiers (k-nearest neighbor algorithm (KNN), support vector machines (SVM), naive bayes (NB), adaBoost (ADA) & logistic regression (LOGI)) for a comparative study on the prediction of SNO sites. We have considered two popular encoding schemes (CKSAAP & Binary) for comparative study on the prediction of SNO sites. The CKSAAP has been selected as a better encoding scheme through a comparative study to develop the proposed predictor. The proposed RF based predictor has shown better performance measured by ROC curves and AUC scores for both training and independent test datasets. We provide the whole development procedure in details for the newly proposed method in the next section.

2. Materials and Methods

2.1 Data Source

To construct SNO site predictor we have downloaded 1356 amino acid sequences from the Mus Musculus database of cysteine SNO (http://140.138.144.145/~dbSNO/download.php). In this dataset, 2641 SNO sites were experimentally detected in 1356 proteins.

2.2 Data preparation

In this study, the experimentally detected SNO sites (Cysteine residues) were considered as positive samples and all the other cysteine residues as negative samples (i.e. non-SNO sites). The negative samples were considered based on an instinctive assumption (Hasan et al., 2015), even though there was no clear evidence of which negative was present or not. An individual site was denoted as a fragment of sequence with cysteine in the midpoint. The length of the fragment sequence is known as the window size. In 1,356 SNO proteins, there were 2,641

positive and 1,0394 negative samples of sites. Since the amounts of positive and negative samples were unbalanced in the original dataset, we constructed three types of training datasets to develop the best classifier (class predictor). But we can't take similar sequences of positive and negative samples. Because the similar sequences cannot classify the SNO sites and non-SNO sites. Those three types of training datasets are as follows:

The training dataset-1 with a 1:1 ratio was constructed by randomly selecting n11=2300 positive samples out of 2641 and n12=2300 negative samples out of 10394, the training dataset-2 with a 1:2 ratio was constructed by randomly selecting n21=2300 positive samples out of 2641 and n22=4600 negative samples out of 10394 and the training dataset-3 with a 1:3 ratio was constructed by randomly selecting n31=2300 positive samples out of 2641 and n32=6900 negative samples out of 10394 without replacement. For each classifier, to investigate the performance of 3 predictors developed based on the 3 training datasets as early mentioned and we constructed a test (independent) dataset by the rest of the positive and negative samples of the original dataset.

To develop the classifier based on a training dataset, we converted the sequence dataset to the numeric dataset by the encoding approaches. A large number of features (e.g. 2646) were created in the encoded dataset. To reduce the computational complexity of predictors, we considered the feature selection strategy also. Finally, we optimized the ratio, encoding, window size, number of important features and classifiers to find the best SNO site predictor.

2.3 Data Encoding

There are several encoding approaches in the literature to covert the sequence data into numeric data. Here we considered two popular encoding approaches as discussed below:

2.3.1 CKSAAP encoding

The CKSAAP encoding method was firstly introduced by Chen et al. (Chen *et al.*, 2007). It has been broadly using in the numerous bioinformatics work (Xu *et al.*, 2013; Hasan *et al.*, 2019a; Hasan and Kurata, 2018; Hasan *et al.*, 2018b; Hasan *et*

al., 2019b; Khatun *et al., 2*019). If window size of a fragment is 2r + 1 (where r = 1, 2, 3 ,…) and 21 types of amino acids (including the gap (O)), it may create (21× 21) = 441 types of amino acid pairs (i.e. AA, AC, AD,. . ., OO) for every single k (k denotes the space between two amino acids). For the optimal kmax =5, there are 21× (kmax + 1) × 21 = 2646 different amino acid pairs are created for each sequence. Then the feature vectors are calculated using the following equation:

$$\left(\frac{N_{AA}}{N_{total}},\frac{N_{AC}}{N_{total}},\frac{N_{AD}}{N_{total}},\ \cdots\ ,\frac{N_{OO}}{N_{total}}\right)_{441} \quad (1)$$

where N_{total} is the length of the total composition residues (for example, if the fragment length H is 29 and k = 0, 1, 2, 3, 4, 5 then N_{total} = H - k -1 will be 28, 27, 26, 25, 24 and 23, respectively). N_{AA}, N_{AC},...,N_{OO} are the frequency of the amino acid pair within the fragment. More details are available somewhere (Hasan *et al.,* 2018b; Hasan *et al.,* 2019b; Khatun *et al., 2*019).

2.3.2 Binary encoding

According to the binary encoding approach, 21 amino acids (including gap (O)) are converted to numeric vectors. The 21 types of residues are organized as ACDEFGHIKLMNPQRSTVWYO. In the query proteins, A is denoted as 10000000000000000000 and C as 01000000000000000000, and so on for binary vector. The center position is always C in each window of SNO sites for the query protein. If we select a window of size 29, the feature vectors with a dimensionality (21×29) = 609 are obtained from the binary encoding.

2.4 Feature selection

As discussed previously, each encoding approach produces high dimensional features with each window. However, high dimensional features create the computational complexity in some popular classifiers during the development of class predictor. Therefore, we considered the filtering approach to select the important features, since the equally expressed features (The equally expressed features mean that features that show similar frequencies between the positive and negative data set) among two or more conditions do not have any significant contribution to the class prediction. There are several statistical approaches for

filtering the feature variables. However, most of the parametric tests are depended on the normality of the dataset. The encoded dataset generated in this study usually not satisfies the normality assumption. Therefore, we have used the non-parametric Wilcoxon Sign Rank Test for filtering the feature variables (Whitley E and Ball J, 2002).

2.5 Learning classifiers

To build a better predictor for protein SNO site prediction, we considered six popular classifiers k-nearest neighbor algorithm (KNN), Support Vector Machines (SVM), Naive Bayes (NB), AdaBoost (ADA), Random Forest (RF) & Logistic Regression (LOGI)) for a comparison based on the encoded protein sequences. For the convenience of the readers, let us introduce those classifiers as follows:

KNN: The KNN is a non-parametric method used for classification. The KNN is classified by a plurality vote of its neighbors. According to the KNN algorithm (Keller *et al.*, 1985; Zhang Z, 2016), the query sample is predicted to the subset represented by its k-nearest neighbors. In this study, if the majority of the k-nearest neighbors of the query sample is being assigned positive sample, this means that it is an SNO site. Otherwise, the query sample is a negative one. Here, some distances are used to measure the nearest neighbors for the KNN algorithm, such as the Hamming distance, Euclidean distance and the Mahalanobis distance. In this study, the KNN algorithm was implemented using the R package 'kknn' as kknn(training_data$class ~ ., traindata, testdata, k = 50, distance = 2).

SVM: The SVM was developed based on the structural risk minimization principle (Cristianini N and Schölkopf B, 2002; Hasan *et al.*, 2015). Supervised learning method mainly applied to classification. The main idea of an SVM is to predict classes with a surface that maximizes the boundaries between them. There are two concepts of boundaries, first is the concept of an optimum linear margin classifier and second is the concept of a kernel. In this study, the SVM algorithm was implemented using the R package 'e1071' as svm(training_data$class ~ ., trainingdata , kernel = "radial").

NB: The NB is a probabilistic classifier established on relating Bayes' theorem with individuality assumptions. NB classifiers are based on the conditional probability of features belonging to a class, which the features are selected by feature selection methods (Zhanga W and Gao F, 2011). In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Depending on the precise nature of the probability model, naïveBayes classifiers can be trained very efficiently in a supervised learning setting. The Naive Bayes classifier was implemented using the R package 'naivebayes' as naiveBayes(training_data$class ~ ., trainingdata).

RF: Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest (Breiman L, 2001a). RF classifier is a group of decision tree classifiers. It is broadly used in protein bioinformatics (Hasan *et al.,* 2018b; Hasan *et al.,* 2019b; Khatun *et al., 2*019; Hasan *et al.,* 2018a; Hasan *et al.,* 2017; Hasan *et al.,*2018; Breiman, 2001b; Hasan *et al.,* 2016). The RF classifier is predicted class by the voting among the number of trees, which covers two classes, either positive samples (SNO sites) or negative samples (non-SNO sites). The RF classifier was implemented using the R package 'randomForest' as randomForest(training_data$class ~ ., trainingdata, ntree = 400, mtry = 10).

ADA: AdaBoost is a machine learning developed by Yoav Freund and Robert Schapire. The boosted classifier represents the final output by combining into a weighted sum of the output of the other learning algorithms. The main ideas of ADA algorithm is to maintain a distribution or set of weights over the training set (Wang R, 2012). The classifier was implemented using the R package 'ada' as ada (training_data$class~ ., trainingdata, iter=20, nu=1, type = "discrete").

LOGI: Logistic regression is used to obtain odds ratio in the presence of more than one explanatory variable. The procedure is quite similar to multiple linear regression, with the exception that the response variable is binomial. By fitting data to a logistic curve, logistic regression is used for identification of the probability of incident of an event (Sperandei S, 2014). It is broadly used for predictor variables which are either numerical or categorical. In LOGI, we set the function as glm (training_data$class~ ., family = 'binomial', data(train)).

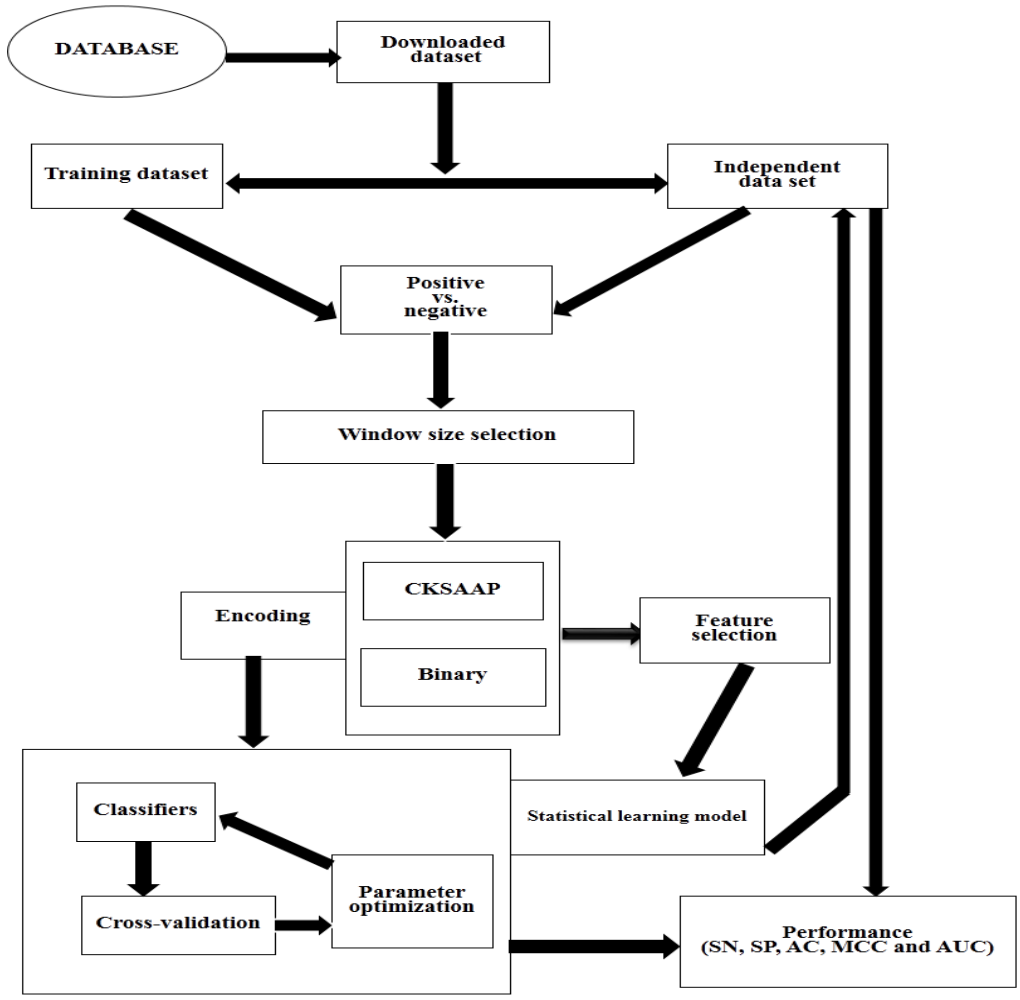

Figure 1: The SNO predictor pipeline. The best prediction model or classifier was built after parameter optimization and performance evaluation.

3. Performance Assessment

To investigate the performance of the proposed predictor, we have used true positive rate (TPR), true negative rate (TNR), false discovery rate (FDR), accuracy (AC), misclassification rate (MCR), mathew correlation coefficient (MCC) as six measurements defined as below:

$$TPR = \frac{nTP}{nTP+nFN} \times 100 \quad (2)$$

$$TNR = \frac{nTN}{nTN+nFP} \times 100 \quad (3)$$

$$AC = \frac{nTP+nTN}{nTP+nFP+nTN+nFN} \times 100 \quad (4)$$

$$MCR = \frac{nFP+nFN}{(nTP+nFP+nTN+nFN)} \times 100 \quad (5)$$

$$FDR = \frac{nFP}{nFP+nTP} \times 100 \quad (6)$$

$$MCC = \frac{((nTP \times nTN)-(nFP \times nFN)) \times 100}{\sqrt{(nTP+nFN) \times (nTN+nFP) \times (nTP+nFP) \times (nTN+nFN)}} \times 100 \quad (7)$$

where, number of true positive = nTP, number of false positive = nFP, number of true negative = nTN, number of false negative = nFN. Where, a true positive is a predicted positive class where the predictor correctly predicts the positive class, a true negative is a predicted negative class where the model correctly predicts the negative class, a false positive is a predicted positive class where the model incorrectly predicts the positive class and a false negative is a predicted negative class where the model incorrectly predicts the negative class. We also usedreceiver operating characteristic (ROC) curve, area under the ROC curve (AUC) score and partial AUC (pAUC) score to investigate the performance. The ROC curve is made from the plot of true positive rate against the false positive rate at different thresholds. The partial area under the ROC curve (pAUC) over the range (0,e) is defined as an integral of the ROC function over the given range, i.e. $\text{pAUC(e)} = \int_0^e \text{ROC(f)df}$. When e=1 the partial area represents the conventional area under the entire ROC curve (AUC). The values of TPR, TNR and AC lies between 0 to 100; MCC lies between -100 to 100, AUC lie between 0 to 1 and pAUC lies between 0 to e. And a higher value represents a better prediction. The values of FDR and MCR lies between 0 to 100, and a lower value represents a better prediction.

4. Results and Discussion

4.1 Selection of Encoding Method

To compare better encoding method from two popular encoding methods (CKSAAP and binary), we trained AdaBoost, RF, SVM, KNN and NB classifiers by the two encoded training datasets. Then the performance indexes (TPR, TNR, AC, MCR, FDR, MCC, ROC, AUC and pAUC) were computed. The comparative

performance results under two encoding systems with different classifiers are given in Table 1.

Table 1: Comparison of CKSAAP encoding with the binary encoding with different classifiers

Classifiers	Encoding	TPR (%)	TNR (%)	FDR (%)	MCR (%)	AC (%)	MCC (%)	AUC	pAUC (at FPR 0.3)
KNN	CKASSP	94	95	5	5	95	89	0.95	0.26
	BINARY	70	62	35	34	66	32	0.69	0.09
SVM	CKASSP	94	92	8	7	93	86	0.95	0.26
	BINARY	86	85	14	14	86	71	0.92	0.24
NB	CKASSP	87	84	16	14	86	71	0.89	0.22
	BINARY	85	77	20	18	82	64	0.86	0.18
ADA	CKASSP	90	98	3	6	94	87	0.98	0.29
	BINARY	94	92	7	7	93	86	0.93	0.25
RF	CKASSP	93	98	2	4	96	91	0.99	0.29
	BINARY	90	98	3	6	94	87	0.93	0.23

Table 2: Comparison of CKSAAP encoding with the binary encoding for RF on test data

Classifier	Encoding	TPR (%)	TNR (%)	FDR (%)	MCR (%)	AC (%)	MCC (%)	AUC	pAUC (at FPR 0.3)
RF	CKASSP	70	87.7	18.6	20	80	59.1	0.854	0.178
	BINARY	52	83.1	29.7	30.4	69.6	37.4	0.648	0.113

From Table 1, we see that all classifiers give better scores with CKSAAP encoding than binary encoding. However, RF gives better performance among the others with the CKSAAP encoded training dataset. So, in this case, we checked the performance of RF classifier only based on the test dataset under these two encodings to select the better encoding system. Table 2 indicates that CKSAAP encoded test dataset also shows better performance with the RF classifier.

4.2 Selection of Better Classifier under the CKSAAP Encoded Datasets with Different Ratios

Naturally, the SNO and non-SNO datasets are incredibly imbalanced. It has been proven that because of the nature of the imbalanced datasets, the accuracy of statistical learning algorithms powerfully affected and computationally inflexible. For this reason, three datasets were prepared by 1:1 (i.e. 2300 positive SNO sites

and 2300 negative SNO sites in the dataset), 1:2 and 1:3 ratios of the positive and negative sample. In each dataset, 21× (5+1) ×21=2646 feature variables were created. The dimensions of 3 training datasets were 4600×2646, 6900×2646 and 9200×2646 corresponding to 1:1, 1:2 and 1:3 ratios, respectively. However, most of classifiers suffer from the computational complexity due to the high-dimensional feature variables. To overcome this problem, we selected top 1500 significant features by non-parametric Wilcoxon signed rank test as a filtering approach, because insignificant features have no significant contribution in both supervised and unsupervised learning. The detailed performance measurements of different classifiers (RF, SVM, KNN, NB, LOGI and ADA) based on 1:1, 1:2 and 1:3 ratio of positive and negative samples with 1500 features in the training datasets are shown in Table 3.

Table 3: Comparison of different class predictors with 1:1, 1:2 and 1:3 ratios of the positive and negative sample for training dataset based on 29 window size and 1500 features.

Ratio of training dataset	Classifiers	TPR (%)	TNR (%)	FDR (%)	MCR (%)	AC (%)	MCC (%)	AUC	pAUC (at FPR 0.3)
1:1	KNN	94	95	5	7	93	89	0.92	0.21
	SVM	94	92	8	7	93	86	0.92	0.22
	NB	87	84	16	14	86	71	0.89	0.20
	ADA	90	98	3	6	94	87	0.93	0.22
	LOGI	84	95	6	11	89	79	0.86	0.23
	RF	**95**	**98**	**2**	**3**	**97**	**94**	**0.99**	**0.29**
1:2	KNN	33	95	25	26	74	37	0.81	0.09
	SVM	88	90	19	11	89	76	0.93	0.23
	NB	67	88	26	19	81	56	0.86	0.18
	ADA	79	89	21	14	86	68	0.94	0.24
	LOGI	94	95	10	6	93	88	0.92	0.24
	RF	**93**	**95**	**9**	**5**	**95**	**92**	**0.97**	**0.27**
1:3	KNN	9	97	46	25	75	14	0.77	0.12
	SVM	92	92	21	8	92	80	0.92	0.22
	NB	61	87	.38	19	81	49	0.85	0.17
	ADA	.64	96	14	12	88	66	0.93	0.23
	LOGI	94	93	14	6	94	86	0.91	0.24
	RF	**92**	**93**	**10**	**6**	**94**	**89**	**0.96**	**0.26**

Table 3 shows that RF classifier with 1:1 ratio based training dataset gives better performance (TPR = 95%, TNR = 98%, FDR=2%, MCC = 94%, MCR=3%, AC = 97%, AUC=0. 99, pAUC=0.29) than any other combination of the classifiers and ratios. Figure 2 also shows that the RF classifier with 1:1 ratio based predictor perform better for both the training and independent test dataset than other predictors.

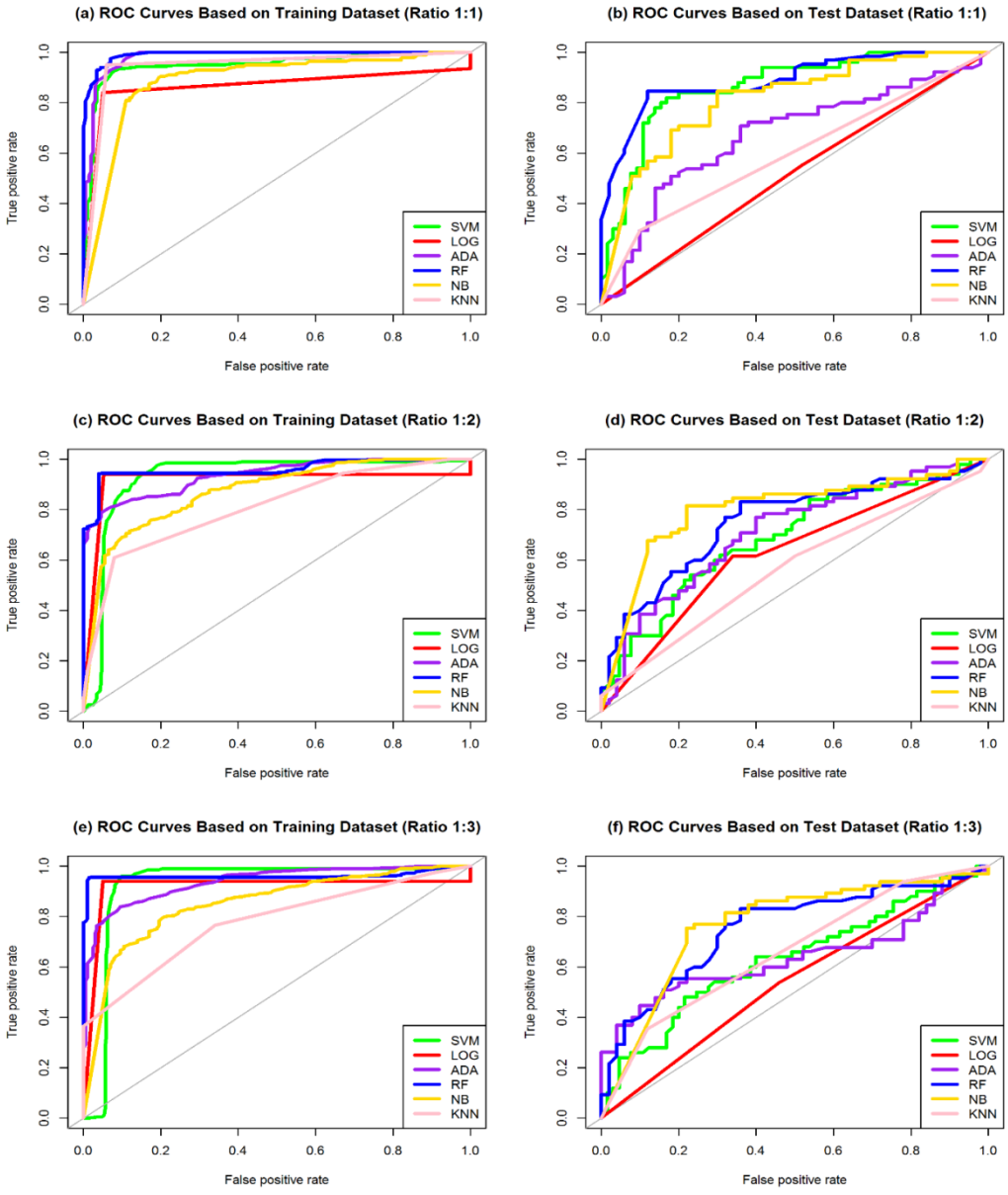

Figure 2(a-f): Figures a, c and e represents the ROC curves for 6 different classifiers based on 1:1, 1:2 and 1:3 ratios with training datasets respectively. Figures b, d and f represents the ROC curves for the independent test dataset with 6 different classifiers corresponding to 1:1, 1:2 and 1:3 ratios, respectfully.

4.3 Selection of optimal window size

A window is defined by a sequence fragment of size 2n+1. In this study, different window sizes (25, 27, 29, 31, and 33) were considered to select the optimal window size. We constructed five predictors based on RF classifier with different five window sizes respectively using 1:1 ratio of training dataset with 1500 features. The Performance of these five predictors based on test dataset are shown in Table 4.

Table 4: Performance of the RF classifier for the test dataset for different window sizes with 1500 features.

Window Sizes	TPR (%)	TNR (%)	FDR (%)	MCR (%)	AC (%)	MCC (%)	AUC	pAUC (at FPR 0.3)
25	56	75.4	36.4	33	67	32	0.693	0.112
27	52	86.2	25.7	28.7	71.3	41.1	0.709	0.132
29	**88**	**83.1**	**20**	**14.8**	**85.2**	**70.5**	**0.891**	**0.214**
31	52	78.5	35	33	67	31.7	0.688	0.10
33	54	80	32.5	31.3	68.7	35.4	0.657	0.112

Table 4 shows that RF based predictor based on test dataset gives the highest performance with window size 29 than other window sizes.

4.4 Building Proposed Predictor

The proposed predictor was constructed combining sub-sections 4.1-4.3 that includes the RF classifier, top 1500 significant CKSAAP encoding features, window size 29 and the 1:1 ratio of SNO and non-SNO sites in the training dataset.

4.4.1 Average performance of the proposed predictor

To investigate the average performance of the proposed predictor, we took 90% data as the training dataset satisfying 1:1 ratio of SNO and non-SNO sites and 10% data as the test dataset without replacement from the original dataset and repeated this procedure 5 times. The average performance of the proposed predictor with the training and test datasets are shown in **Table 5, w**here the value of 1st bracket indicates that the value of the standard deviation of the performance scores of the repeated procedure.

Table 5: Average and standard error of performance measurements for test datasets.

Dataset	TPR (%)	TNR (%)	FDR (%)	MCR (%)	AC (%)	MCC (%)	AUC	pAUC (at FPR 0.3)
Training	93 (4)	98(1)	2(2)	4(3)	96(2)	91(7)	0.93(6)	0.29(1)
Test	89.4(7)	54.7(11)	31.8(4)	27.2(2)	72.8(2)	48.2(2)	0.788(2)	0.132(2)

Table 5 indicates that average performances of the predictor are TPR = 89.4%, TNR = 54.7%, FDR = 31.8%, MCC = 48.2%, MCR = 27.2%, AC = 72.8%, AUC= 0.788, pAUC(at FPR 0.3) = 0.132 for test dataset which are a significant result. All the above results clearly showed that proposed predictor provides the accurate predictions of SNO sites than existing methods.

4.4.2 Performance evaluation by cross-validation

For this analysis the total data set is split into 10 sets. One by one, a set is selected as test set and the 9 other sets are combined into the corresponding outer training set. This is repeated for each of the 10 sets. To investigate the performance of the proposed predictor by the area under the ROC curve (AUC), we perform cross validation. The Table 6 indicates the AUC value of the proposed method with both cross-validated training data and independent test datasets.

Table 6: Performance measurements of cross-validation and independent test.

Description	AUC
10 fold cross-validation for the training dataset	0.759
Independent test dataset	0.787

Table 6 indicates that our proposed method achieves an AUC score of 0.759 for 10-fold cross-validation with the training dataset and AUC score of 0.787 with the independent test dataset.

4.4.3 Performance comparison of the proposed method with existing methods

To compare the proposed method with the existing methods (SNOSite, GPS-SNO, iSNO-PseAAC and iSNO-AAPair) those were primary sequence based SNO site identification techniques (Li *et al.*, 2011; Li *et al.*, 2012; Xue, 2010; Xu *et al.*,

2013), we performed 10-fold cross-validation. But in the iSNO-PseAAC only accepted characters are 20 natural amino acid notations but not accepted "-" although '-' is a amino acid character. Table 7 indicates that the performance scores of TPR, TNR, AC and MCC are achieved 80.4%, 68%, 75% and 47% by the proposed method those are significantly improved than the existing methods. Thus, the proposed method outperformed the existing methods.

Table 7: Performance comparison with existing methods.

Prediction methods	TPR	TNR	AC	MCC
SNOSite	84 %	30 %	57 %	0.49 %
GPS-SNO	46%	66%	56%	0.22 %
iSNO-PSeAAC	52%	58%	55%	0.19%
iSNO-AAPair	24%	64%	44%	-0.22 %
Proposed (1:1 ratio + CKSAAP + 29 window + 1500 feature) method	80.4%	68%	75%	47%

4.5 Sequence specificity of SNO site

By using Two Sample Logos software (Vacic *et al.*, 2006), the amino acid tendencies of neighboring SNO sites were compared to the non-SNO sites that are displayed for the training dataset in Figure 3. In this software, positive samples (SNO sites) represent its residues at each location of window size that is plotted above the X-axis. On the other hand, negative samples (non-SNO sites) represent its residues at each location of window size that are plotted under the X-axis. The proportion of positive (over display) or negative samples (under display) were showed by the height of the letter harboring the resultant residue. The cumulative percentage of these over/under displayed residues were plotted in the Y-axis. In the following calculation and operation, we selected 29-mer (-14, +14) window size and Figure 3 shows the position-specific difference of amino acid compositions between S-nitrosylation sites and non-S-nitrosylation sites. Figure 3 indicates that some amino acids are over/under represented at specific points and to identify the SNO sites, the positional amino acid encoding is an effective technique. Although we know that the binary encoding is a positional based encoding, but Table 1 shows that binary encoding is not sufficient to exactly identify the SNO Sites

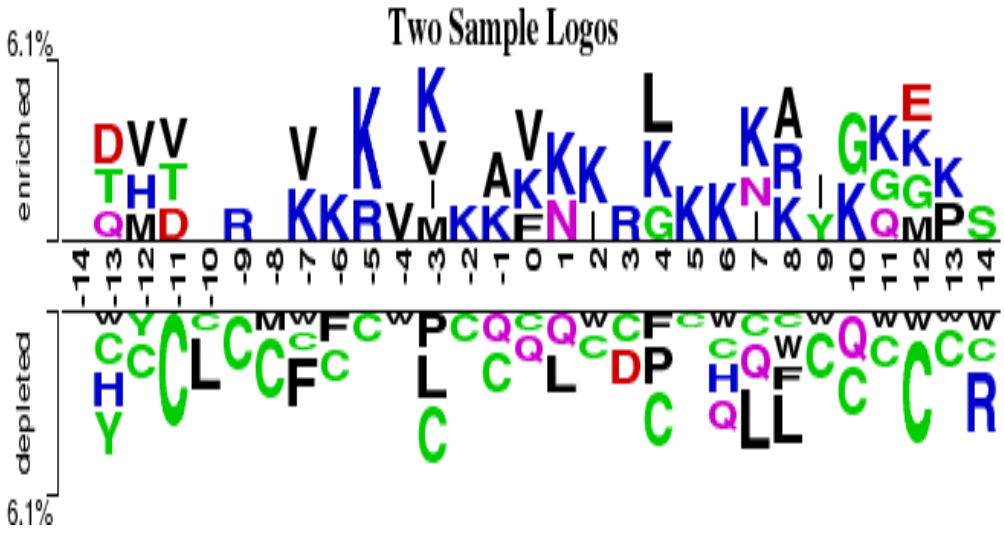

Figure 3: The amino acid propensities of surrounding SNO sites compared to non- SNO sites, as displayed with the Two Sample Logos software.

From the Figure 3 we visualized some interesting findings. There are: (i) the SNO site fragment has no Cysteine without middle position but in the non-SNO site fragment may have Cysteine at any position, (ii) in the SNO site fragment may have (D/T/V), R, (R/K), K, K at position -12, -10, -6, -3, 5 respectively but in the non-SNO site fragment may have only C at position -12, -10, -6, -3, 5.

5. Conclusions

In this paper, we proposed a random forest based novel *in silico* predictor for predicting of protein SNO site of *Mus Musculus* species. To develop this novel predictor, we considered six different popular classifiers (KNN, SVM, NB, RF, ADA & LOGI) for the comparative study on the prediction of SNO sites using the experimentally identified S-nitrosylated protein sequences of *Mus Musculus*. The CKSAAP performed a better encoding scheme through a comparative study to develop the proposed predictor. We have observed that the proposed random forest classifier improves the performance over the other classifiers under the CKSAAP encoded dataset with 29 window size and 1500 important features out

of 2646. The performance was measured by ROC curves and AUC scores for both training and independent test datasets. We observed that our proposed predictor performs (AUC 75.9%, Accuracy 75% and MCC 47%) much better than the other existing predictors under the 10-fold cross-validation. The output of the proposed method may help to explore the SNO-related cellular functions in *Mus Musculus* as well as Human since *Mus Musculus* is evolutionary related to Human. To implement the proposed method, we provided the computational codes and instructions, which can be downloaded at http://www.bbcba.org/softwares/SnoPred.zip.

References

[1] Ahmed, F. F., Mosharaf, M. P., Sultana, A., Reza, M.S., Ahmed, M. S., Khatun, M. S., Hasan, M. M. and Mollah, M. N.H. (2017). Identification of protein S-Nitrosylation sites using composition of amino acid frequency. International conference on bioinformatics and biostatistics for agriculture, health and environment, ISBN: 978-984-34-0996-6.

[2] Breiman, L. (2001a). Random Forests, Machine Learning, 45, 5–32.

[3] Breiman, L. (2001b). SNP-based analysis of genetic substructure in the German population, Machine Learning, 45: 5–32.

[4] Chen, K., Kurgan, L. and Rahbari, M. (2007). Prediction of protein crystallization using collocation of amino acid pairs, Biochem. Biophys. Res. Commun, 355: 764–769.

[5] Cristianini, N. and Schölkopf, B. (2002). Support Vector Machines and Kernel Methods: The New Generation of Learning Machines, AI Magazine, Volume 23, Number 3.

[6] Daaka, Y., Lefkowitz, R. J. and Stamler, J. S. (2007). Regulation of beta-adrenergic receptor signaling by S-nitrosylation of G-protein-coupled receptor kinase 2, Cell, 129: 511–522.

[7] Foster, M. W., Douglas, T. H. and Jonathan, S. S. (2009). Protein S-nitrosylation in health and disease: A current perspective, Trends Mol. Med, 15: 391–404.

[8] Foster, M W., McMahon, T. J. and Stamler, J. S. (2003). S-nitrosylation in health and disease, Trends Mol. Med., 9: 160–168.

[9] Hasan, M. M., Guo, D. and Kurata, H. (2017). Computational identification of protein S-sulfenylation sites by incorporating the multiple sequence features information, MolBiosyst, 13: 2545-50.

[10] Hasan, M. M., Khatun, M. S. and Kurata, H. (2018). A comprehensive review of in silico analysis for protein S-sulfenylation sites, Protein & Peptide Letters, 9: 815 – 821.

[11] Hasan, M. M., Khatun, M. S. and Kurata, H. (2019a). Computational identification of microbial phosphorylation sites by the enhanced characteristics of sequence information, Sci Rep., 9(1):8258.

[12] Hasan, M. M., Khatun, M. S. and Kurat, H. (2019b). Large-scale assessment of bioinformatics tools for lysine succinylation sites, Cells, 8(2), 95.

[13] Hasan, M. M., Khatun, M. S., Mollah, M. N. H., Yong, C. and Guo, D. (2018a). A systematic identification of species-specific protein succinylation sites using joint element features information, Int J Nanomedicine, 12: 6303-15.

[14] Hasan, M. M., Khatun, M. S., Mollah, M. N. H., Yong, C. and Guo, D. (2018b). NTyroSite: computational identification of protein nitrotyrosine sites using sequence evolutionary features. Molecules, 23(7), 166.

[15] Hasan, M. M. and Kurata, H. (2018). GPSuc: global prediction of generic and species-specific succinylation sites by aggregating multiple sequence features, PLoS One, 13(10):e0200283.

[16] Hasan, M. M., Yang, S., Zhou, Y. and Mollah, M. N. H. (2016). SuccinSite: a computational tool for the prediction of protein succinylation sites by exploiting the amino acid patterns and properties, MolBiosyst., 12: 786-95.

[17] Hasan, M. M., Zhou, Y., Lu, X., Li, J., Song, J. and Zhang, Z. (2015). Computational Identification of Protein Pupylation Sites by Using Profile-Based Composition of k-Spaced Amino Acid Pairs, PLoS One, 10: e0129635.

[18] Hess, D. T., Matsumoto, A., Kim, S. O., Marshall, H. E. and Stamler, J. S. (2005). Protein S nitrosylation: purview and parameters, Nat Rev Mol Cell Biol, 6: 150–166.

[19] Keller, J. M., Michael, R. G. and James, A. G. (1985). A fuzzy k-nearest neighbor algorithm, Syst. Man Cybern. IEEE, Trans.4: 580–585.

[20] Khatun, M. S., Hasan, M.M. and Kurata, H. (2019). Efficient computational model for identification of anti-tubercular peptides by integrating amino acid patterns and properties, FEBS Lett, doi: 10.1002/1873-3468.13536. (in press)

[21] Li, B. Q., Hu, L. L., Niu, S., Cai, Y. D. and Chou, K. C. (2012). Predict and analyze S-nitrosylation modification sites with the mRMR and IFS approaches, Journal of Proteomics, 75:1654–1665.

[22] Lipton, S. A., Choi, Y.B., Pan, Z. H., Lei, S. Z., Chen, H. S. V., Sucher, N. J., Singel, D. J., Loscalzo, J. and Stamler, J. S. (1993). A redox-based mechanism for the neuroprotective and neurodestructive effects of nitric oxide and related nitroso-compounds, Nature, 364(6438): 626-632.

[23] Li, Y. X., Yuan, H. S., Ling, J. and Nai, Y. D. (2011). An efficient support vector machine approach for identifying protein S-nitrosylation sites, Protein Pept. Lett., 18: 573–587.

[24] Lugovskoy, A. A., Zhou, P., Chou, J. J., McCarty, J. S., Li, P. and Wagner, G. (1999). Solution structure of the CIDE-N domain of CIDE-B and a model for CIDE-N/CIDE-N interactions in the DNA fragmentation pathway of apoptosis, Cell, 99: 747–755.

[25] Nakamura, T., Cieplak, P., Cho, D. H., Godzik, A. and Lipton, S. A. (2006). S-nitrosylation of Drp1 links excessive mitochondrial fission to neuronal injury in neurodegeneration, Mitochondrion, 10: 573–578.

[26] Sperandei, S. (2014). Understanding logistic regression analysis, Biochem Med (Zagreb), 24(1): 12–18.

[27] Vacic, V., Iakoucheva, L. M. and Radivojac, P. (2006). Two Sample Logo: a graphical representation of the differences between two sets of sequence alignments, Bioinformatics, 22: 1536–1537.

[28] Wang, R. (2012). AdaBoost for Feature Selection, Classification and Its Relation with SVM, Physics Procedia, Volume 25, Pages 800-807.

[29] Whalen, E. J., Foster, M. W., Matsumoto, A., Ozawa, K., Violin, J. D., Que, L. G., Nelson, C. D., Benhar, M., Keys, J. R., Rockman, H. A., Koch, W. J., Schonhoff, C. M., Matsuoka, M., Tummala, H., Johnson, M. A., Estevéz, A. G., Wu, R., Mannick, J. B., (2006). S-nitrosothiol depletion in amyotrophic lateral sclerosis,. Proc. Natl. Acad. Sci. USA, 103: 2404–2409.

[30] Whitley, E. and Ball, J. (2002). Statistics review 6: Nonparametric methods, Critical Care, 6(6): 509–513.

[31] Xu, Y., Ding, J., Wu, L. Y., Chou, K. C. (2013). iSNO-PseAAC: Predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition, PLoS One, 8: e55844.

[32] Xue, Y., Liu, Z., Gao, X., Jin, C., Wen, L., Yao, X. and Ren, J. (2010). GPS-SNO: Computational prediction of protein S-nitrosylation sites with a modified GPS algorithm, PLoS One, 5: e11290.

[33] Zhanga, W. and Gao, F. (2011). An Improvement to Naive Bayes for Text Classification, Procedia Engineering, 15: 2160 – 2164.

[34] Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors, Ann Transl Med., 4(11): 218.