



## WEB SERVER

# DeepNitro: Prediction of Protein Nitration and Nitrosylation Sites by Deep Learning



Yubin Xie<sup>1,\*,a</sup>, Xiaotong Luo<sup>1,\*,b</sup>, Yupeng Li<sup>1,\*,c</sup>, Li Chen<sup>1,d</sup>, Wenbin Ma<sup>1,e</sup>  
 Junjiu Huang<sup>1,f</sup>, Jun Cui<sup>1,g</sup>, Yong Zhao<sup>1,h</sup>, Yu Xue<sup>2,i</sup>, Zhixiang Zuo<sup>1,\*,j</sup>  
 Jian Ren<sup>1,\*,k</sup>

<sup>1</sup> State Key Laboratory of Oncology in South China, Cancer Center, Collaborative Innovation Center for Cancer Medicine, School of Life Sciences, Sun Yat-sen University, Guangzhou 510060, China

<sup>2</sup> Department of Bioinformatics & Systems Biology, MOE Key Laboratory of Molecular Biophysics, College of Life Science and Technology and the Collaborative Innovation Center for Biomedical Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Received 7 January 2018; revised 12 April 2018; accepted 27 April 2018  
 Available online 27 September 2018

Handled by Hsien-Da Huang

## KEYWORDS

Protein nitration and nitrosylation;  
 Deep learning;  
 Web service;  
 Functional site prediction;  
 Feature extraction

**Abstract** Protein nitration and nitrosylation are essential post-translational modifications (PTMs) involved in many fundamental cellular processes. Recent studies have revealed that excessive levels of nitration and nitrosylation in some critical proteins are linked to numerous chronic diseases. Therefore, the identification of substrates that undergo such modifications in a site-specific manner is an important research topic in the community and will provide candidates for targeted therapy. In this study, we aimed to develop a computational tool for predicting nitration and nitrosylation sites in proteins. We first constructed four types of encoding features, including positional amino acid distributions, sequence contextual dependencies, physicochemical properties, and position-specific

\* Corresponding authors.

E-mail: [renjian@sysucc.org.cn](mailto:renjian@sysucc.org.cn) (Ren J), [zuozhx@sysucc.org.cn](mailto:zuozhx@sysucc.org.cn) (Zuo Z).

# Equal contribution.

<sup>a</sup> ORCID: 0000-0002-3566-4849.

<sup>b</sup> ORCID: 0000-0002-7367-9910.

<sup>c</sup> ORCID: 0000-0002-3933-066X.

<sup>d</sup> ORCID: 0000-0002-0139-6778.

<sup>e</sup> ORCID: 0000-0001-8774-7593.

<sup>f</sup> ORCID: 0000-0002-3417-1196.

<sup>g</sup> ORCID: 0000-0002-8000-3708.

<sup>h</sup> ORCID: 0000-0001-5068-2212.

<sup>i</sup> ORCID: 0000-0002-9403-6869.

<sup>j</sup> ORCID: 0000-0002-2492-2689.

<sup>k</sup> ORCID: 0000-0002-4161-1292.

Peer review under responsibility of Beijing Institute of Genomics, Chinese Academy of Sciences and Genetics Society of China.

<https://doi.org/10.1016/j.gpb.2018.04.007>

1672-0229 © 2018 The Authors. Production and hosting by Elsevier B.V. on behalf of Beijing Institute of Genomics, Chinese Academy of Sciences and Genetics Society of China.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

scoring features, to represent the modified residues. Based on these encoding features, we established a predictor called DeepNitro using **deep learning** methods for predicting protein nitration and nitrosylation. Using *n*-fold cross-validation, our evaluation shows great AUC values for DeepNitro, 0.65 for tyrosine nitration, 0.80 for tryptophan nitration, and 0.70 for cysteine nitrosylation, respectively, demonstrating the robustness and reliability of our tool. Also, when tested in the independent dataset, DeepNitro is substantially superior to other similar tools with a 7%–42% improvement in the prediction performance. Taken together, the application of deep learning method and novel encoding schemes, especially the position-specific scoring feature, greatly improves the accuracy of nitration and nitrosylation site prediction and may facilitate the prediction of other PTM sites. DeepNitro is implemented in JAVA and PHP and is freely available for academic research at <http://deepnitro.renlab.org>.

## Introduction

Proteins undergo various post-translational modifications (PTMs) to maintain their functions. Of the nearly 460 types of PTMs, protein nitration and nitrosylation are special epigenetic regulations that are mainly caused by unfavorable side reactions from metabolic processes during normal cellular activities. Under physiological conditions, nitric oxide (NO) is produced by NO synthases (NOS) [1] and subsequently diffuses through cell membranes. As a signaling molecule, NO is able to regulate many vascular and neuronal signaling pathways, as well as mitochondrial proliferation [2,3]. In the presence of oxidants, NO can be further converted into different reactive nitrogen species (RNS), such as nitrous acid (HNO<sub>2</sub>), dinitrogen trioxide (N<sub>2</sub>O<sub>3</sub>), nitrosyl anion (NO<sup>-</sup>), nitrosyl cation (NO<sup>+</sup>), and nitrogen dioxide radical (NO<sub>2</sub>) [4]. These RNS are able to induce nitration of proteins, thereby changing their chemical properties and reforming their tertiary structures. It has been reported that nitration can occur at amino acid residues such as tyrosine [5] and tryptophan [6]. In addition to protein nitration, RNS can also lead to the formation of protein *S*-nitrosylation by covalently attaching the nitrosyl group to the sulfur atom of cysteine [7].

The addition of nitro or nitrosyl groups to certain amino acid residues is an important mechanism for regulating the biological functions of specific cellular proteins by conferring particular physicochemical properties to the modified residues [8]. Recent reports have demonstrated that tyrosine nitration, tryptophan nitration, and *S*-nitrosylation play critical roles in multiple physiological and pathological processes, including cell signaling [9], immune response [10], cell death [11], transcriptional regulation [12], and protein activity [13]. The abnormal abundance of these modifications may lead to deleterious consequences. Many chronic diseases such as diabetes [14], atherosclerosis [15], chronic renal failure [16], cardiovascular diseases [17], and neurological disorders [18] are evidently linked to the aforementioned modifications. Therefore, the identification of substrates that undergo nitration or *S*-nitrosylation in a site-specific manner is important for providing potential guidance for the development of new therapeutic strategies and drugs.

At present, the large-scale identification of nitration or *S*-nitrosylation sites mainly relies on mass spectrometry-based methods [19]. However, because the level of endogenously nitrated or nitrosylated proteins in the cell is usually very low, highly efficient *in vivo* detection of individual nitrated or nitrosylated proteins has long remained a major methodological issue. Prior immunoprecipitation with specific antibodies is

helpful to improve efficiency, but the immunoprecipitation step usually requires complicated procedures, resulting in a laborious, inefficient, and expensive process. In this regard, further efforts are needed to improve the efficiency of current proteomic methodologies so that they can be applied in more research cases and facilitate the investigation of RNS-induced protein modifications.

In contrast to the time-consuming and expensive experimental methods, computational approaches for discovering PTM sites have attracted considerable attention because of their convenience and efficiency. To date, several programs have been developed for predicting the nitration and *S*-nitrosylation site, such as GPS-YNO2 [20] and iNitro-tyr [21] for tyrosine nitration, as well as GPS-SNO [22], iSNO-PseAAc [23], and SNOsite [24] for *S*-nitrosylation site prediction. However, many issues remain in these algorithms, leaving a lot of room for improvement. First, the existing tools generally rely on traditional shallow machine learning methods for prediction, which fail to learn the underlying biological features of protein modifications lacking a consensus sequence, such as nitration or nitrosylation, thus leading to inaccurate predictions of potential modification sites. Second, feature selection is critical for machine learning-based algorithms. However, the current feature extraction methods used in the methods above are unable to fully characterize the biological properties of the potential sites, resulting in disappointing performance. Third, up to now, there are no available tools for the prediction of tryptophan nitration, which limit the comprehensive study of protein nitration and nitrosylation in mammalian. Therefore, the development of a novel tool that is able to extract high-level features from the input sequences and produce reliable prediction results with sufficient accuracy remains an important problem to be solved.

Recently, the application of deep learning algorithm in machine learning research has attracted more and more attention. By learning a suitable representation of the input data, the raw vectors can be transformed into highly abstract features through propagating the whole model. Theoretically, superposition of sufficient level of neural network can increase the ability of feature extraction, resulting in more accurate interpretation of data. However, the so-called gradient diffusion problem has greatly restricted the depth of neural network model [25]. Until 2006, Hinton et al. proposed the layer-wise training strategy [26] to solve this problem properly and the deep neural network turned to practical account. Nowadays, many new techniques have been developed for the training of deep learning model, including Rectified Linear Unit (ReLU) [27], dropout training [28], regularization [29], and

momentum method [30]. With these advantages, deep learning has achieved state-of-the-art accuracy on many prediction tasks, such as image classification [31], speech recognition [32], and natural language processing [33]. Inspired by its excellent performance, deep learning is most recently applied in research field of computational biology. Typical applications include transcript factor binding site detection [34], protein structure prediction [35], RNA splicing prediction [36], and pathogenic variant identification [37]. In view of this, introducing deep learning algorithm into the prediction of nitration and nitrosylation sites would be a promising move to solve the current deficiencies in learning of in-depth biological features and provide more meaningful candidates for further experimental considerations.

In this work, by applying the deep learning algorithm together with effective feature extraction methods, we developed a novel software called DeepNitro to predict protein nitration and nitrosylation sites. For convenience, a webserver has also been developed and is freely available at <http://deep-nitro.renlab.org>.

## Methods

### Preparation of the dataset

To collect the training dataset, we first searched the scientific literature published before Jun 30th, 2015 from PubMed using the keywords “Nitration”, “Nitrated”, “Nitrosylation”, or “Nitrosylated”. By manually reading the retrieved articles, we collected the exact locations of all the experimentally-verified nitration and nitrosylation sites. After removing redundant sites, we collected a total of 1518 tyrosine nitration sites, 66 tryptophan nitration sites, and 4762 *S*-nitrosylation sites in 3113 unique proteins. As previously described [20,22], we treated the modified residues (Y for tyrosine nitration, W for tryptophan nitration, and C for *S*-nitrosylation) that have been reported in the published literature as positive data. Accordingly, the same types of non-modified residues present in the same sequence were considered as negative data. Generally speaking, if the modified residues included in the training dataset are redundant with too many homologous sites, the training process will carry a significant risk of model overfitting, leading to spurious prediction. To avoid this possibility, we first clustered the collected protein sequences using CD-Hit with an identity threshold of 40%. For proteins clustered in the same homologous group, we re-aligned them using the Smith-Waterman algorithm and checked the results manually. If two given positive sites or negative sites shared an identical flanking sequence around the modified residues, only one item was reserved for the model training. Finally, for tyrosine nitration, we constructed a training set of 1210 positive and 8043 negative sites. For tryptophan nitration, the training data contained 66 positive sites and 155 negative sites. For *S*-nitrosylation, we retained 3409 positive sites and 17,453 negative sites as the training set (Table S2).

Particularly, to evaluate the actual performance of our prediction model and compare it with other existing tools, we also constructed the independent test set for tyrosine nitration and *S*-nitrosylation. To ensure a fair comparison, the independent

test set does not contain any nitration or nitrosylation sites that are present in the training set of previously-published tools. To meet this criterion, we constructed the test set by focusing on the most recently collected data and removing duplicate sites that had been used in training sets of other tools. In total, 189 positive sites and 1182 negative sites were selected for the independent test set for tyrosine nitration. For the independent test set for *S*-nitrosylation, another 485 positive sites and 4947 negative sites were included (Table S2). However, due to the limited data availability, an additional test set for tryptophan nitration was not constructed.

### Feature encoding scheme

Before constructing the prediction model for protein nitration and nitrosylation, we first transferred the known modification sites into a set of feature vectors that can be directly recognized by classification algorithms. As presented in our previously published papers [20,22], the chemical properties and amino acid composition around the modified residue may affect the specificity of nitration and nitrosylation. Therefore, we constructed the encoding scheme by extracting sequence or biochemical features from the flanking sequence of length  $L$  in which the modified or non-modified site is located at the central position. Here, we defined the aforementioned region as a nitration or nitrosylation peptide. Specifically, the optimal window size  $L$  can be selected by the subsequent 4-fold cross-validation. A detailed description of the encoding scheme is provided below.

#### *One-hot encoding of the nitration or nitrosylation peptide*

To precisely describe the amino acid distribution at each position in the nitration or nitrosylation peptide, we transferred the 20 natural amino acid residues and a gap character filling the sequence termini into a 21-dimensional binary array. Accordingly, one-hot encoding for a peptide with a window size of  $L$  would result in a binary feature vector of  $L \times 21$  dimensions.

#### *The physical and chemical properties of the nitration or nitrosylation peptide*

The physiochemical environment is responsible for the formation of a covalent linkage between proteins and nitrogenous groups [38,39]. Thus, measurement of the physiochemical properties of the candidate residues may be particularly important for the accurate prediction of nitration and nitrosylation sites. Therefore, we introduced the property factor representation (PFR) [40] in our work and encoded the flanking regions into a set of physiochemical features. Using the encoding table (Table S3) from PFR, a given amino acid residue  $X$  can be encoded into a physiochemical feature vector  $x$  of dimensionality in 10 as defined below.

$$x = (f_x^1, f_x^2, f_x^3, \dots, f_x^{10}) \quad (1)$$

where  $f_x^i$  denotes the  $i$ -th property factor for amino acid residue  $x$ . Accordingly, a nitration or nitrosylation peptide with length of  $L$  can be represented as a  $10 \times L$  dimensional numeric vector. Similar to one-hot encoding, the gap characters filling the sequence termini are encoded with vectors of zeros.

### *k*-Space spectrum encoding

To depict the sequence context of a nitration or nitrosylation peptide, we calculated the *k*-space spectrum in our encoding scheme. When denoting a flanking region, we first scanned through the whole sequence with a sliding window of length *k* and counted the number of all possible amino acid pairs (e.g., LxxxE is a three-space amino acid pair) for subsequent use. Specifically, for peptide with gap characters, we simply ignored the gapped sequence and encoded *k*-space spectrum only in meaningful regions. We represented a specific amino acid pair as  $A_i A_j$ . A total of 400 amino acid pairs can be obtained for any given protein sequence. Then, the frequency of a *k*-space amino acid pair was calculated as follows.

$$f_{ij} = \frac{A_i A_j}{L - k + 1} \quad (2)$$

And the amino acid spectrum can be constructed by calculating the frequency of all possible amino acid pairs.

$$\text{spectrum} = (f_{1,1}, f_{1,2}, f_{1,3}, \dots, f_{10,10}, \dots, f_{20,20})_{400} \quad (3)$$

The *k* values can be adjusted in a range of 0–4 to capture more information about the sequence dependency. Finally we obtained a 2000 dimensional vector for the amino acid spectrum.

### Encoding with the position-specific scoring matrix

To depict the sequence conservation of nitration or nitrosylation sites, we developed the following position-specific scoring matrix (PSSM) encoding scheme in our prediction algorithm. Based on the model proposed by Vacic et al. [41], we first calculated the statistical significance of the differences in the frequencies of symbol occurrence between the positive and negative datasets. Let *P* and *Q* be the positive and negative samples, and  $|P|$  and  $|Q|$  be the numbers of sequences in the corresponding sample, respectively; and  $P_i$  be the *i*-th peptide in the positive dataset and  $P_{ij}$  be the *j*th position in peptide  $P_i$ . For each position in *P* and for each symbol *a* from the alphabet (including 20 natural amino acids and a gap character), a binary vector  $X_p^{j,a}$  can be generated as shown below.

$$X_p^{j,a} = (I_1, I_2, \dots, I_{|P|}) \quad (4)$$

where  $I_i$  can be calculated as follows.

$$I_i = \begin{cases} 1 & P_{ij} = a \\ 0 & P_{ij} \neq a \end{cases} \quad (5)$$

The vector  $X_Q^{j,a}$  for the negative sample can be generated in a similar way. Then, we constructed a null hypothesis that vectors  $X_p^{j,a}$  and  $X_Q^{j,a}$  were sampled from the same distribution, or, in other words, the occurrence probabilities for amino acid *a* are identical at position *j* in both samples. According to this hypothesis, we could compute the *P* value using a two-sample *t*-test [41]. However, since nitration and nitrosylation can only occur in specific amino acid residues, the central position in both the positive and negative data set would be the same. Therefore, when computing *P* value in such case, the variance in vectors  $X_p^{j,a}$  and  $X_Q^{j,a}$  would be zero, thereby making the *t* statistics incalculable. To fix this issue, the central position of the input peptide should be discarded. Accordingly, the significant PSSM *P* values are constructed as shown below,

$$P_{\text{PSSM}} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,k-1} & p_{1,k+1} & \cdots & p_{1,L} \\ p_{2,1} & \cdots & p_{2,k-1} & p_{2,k+1} & \cdots & p_{2,L} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{21,1} & \cdots & p_{21,k-1} & p_{21,k+1} & \cdots & p_{21,L} \end{bmatrix} \quad (6)$$

where *L* is the length of the flanking region,  $p_{ij}$  denotes the *P* value of the *i*th amino acid in the *j*th position for a given set of nitration or nitrosylation peptide, and *k* refers to the central position. Notably, for the peptide sequence, the matrix above would be in a dimension of  $21 \times (L - 1)$ .

Although the significant PSSM can represent the differences in sequence conservation between positive and negative sites, the positive and negative tendency, unfortunately, cannot be inferred from such a matrix. To address this issue, we further established the following computational process to measure the conservation tendency of a given set of nitration or nitrosylation peptide. Specifically, we first calculated the observed frequency of an amino acid *a* at position *j* and constructed a frequency PSSM as shown below,

$$F_{\text{Pos}} = \begin{bmatrix} f_{1,1} & \cdots & f_{1,k-1} & f_{1,k+1} & \cdots & f_{1,L} \\ f_{2,1} & \cdots & f_{2,k-1} & f_{2,k+1} & \cdots & f_{2,L} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{21,1} & \cdots & f_{21,k-1} & f_{21,k+1} & \cdots & f_{21,L} \end{bmatrix} \quad (7)$$

where  $F_{\text{Pos}}$  denotes the frequency PSSM for the positive peptides, and  $f_{ij}$  represents the observed frequency of the *i*-th symbol from the alphabet in the *j*-th position. Accordingly, the frequency PSSM for negative peptides (denoted as  $F_{\text{Neg}}$ ) can also be calculated following the same approach. Based on  $F_{\text{Pos}}$  and  $F_{\text{Neg}}$ , we next computed the final encoding PSSM as shown below,

$$E_{\text{PSSM}} = \begin{bmatrix} E_{1,1} & \cdots & E_{1,k-1} & E_{1,k+1} & \cdots & E_{1,L} \\ E_{2,1} & \cdots & E_{2,k-1} & E_{2,k+1} & \cdots & E_{2,L} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ E_{21,1} & \cdots & E_{21,k-1} & E_{21,k+1} & \cdots & E_{21,L} \end{bmatrix} \quad (8)$$

where  $E_{ij}$  is calculated as below.

$$\delta_{ij} = \frac{f_{ij}^{\text{Pos}} - f_{ij}^{\text{Neg}}}{P_{ij}} \quad (9)$$

$$E_{ij} = \begin{cases} \ln(|\delta_{ij}| + 1) & \delta_{ij} \geq 0 \\ \ln(|\delta_{ij}| + 1) & \delta_{ij} < 0 \end{cases} \quad (10)$$

The final encoding PSSM represents the conservation tendency of the nitration or nitrosylation peptides. If  $E_{ij}$  is greater than zero, then the amino acid at this position is more likely to be observed in the positive sites; conversely, the amino acid would have a better chance of appearing in the negative sites. Using  $E_{\text{PSSM}}$  (Table S4), we encode a nitration or nitrosylation peptide into a numeric vector of dimension  $L - 1$ .

### Deep neural network for predicting potential nitration or nitrosylation sites

For the detection of potential nitration or nitrosylation sites, we introduced a deep neural network model in our prediction



algorithm. By coarse grid search, we optimized the network as an eight-layer architecture (Figure 1, Figure S1). Specifically, the first layer provided a module for data input, and numeric vectors were directly assigned to the neurons during training and predicting steps. From the second to the seventh layer, each layer was implemented as a fully connected dense layer according to the equation below,

$$o_j^l = \sigma \left( \sum_{i=1}^H w_{ij}^{l-1} o_i^{l-1} + b_j^{l-1} \right) \quad (11)$$

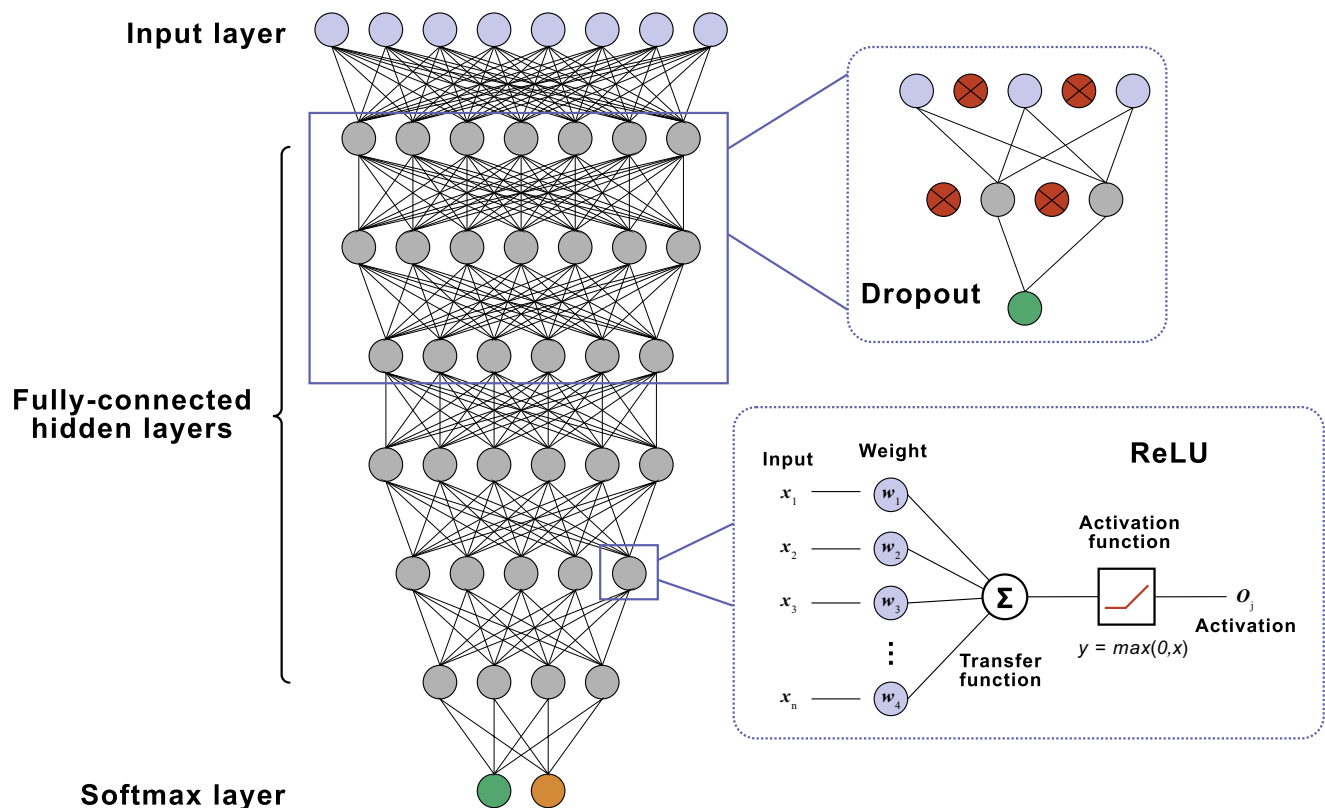
where  $H$  is the number of neurons in the  $(l-1)^{\text{th}}$  layer,  $w_{ij}^{l-1}$  and  $b_j^{l-1}$  are the weight and bias associated with the  $j^{\text{th}}$  neuron, respectively, and  $\sigma(\cdot)$  denotes a non-linear activation function in this neuron. We expected that high-level features might be extracted from the input vector when propagating the signal down these layers. The eighth layer is the output layer.

In probability theory, the output of a softmax function can be used to represent a probability distribution over  $k$  different possible outcomes. Therefore, it is widely used in various multiclass classification algorithms, such as softmax regression, naive Bayes classifiers, and artificial neural networks. Specifically, in neural network, softmax function was first introduced by Bridle and his colleagues [42]. Comparing to simple logistic regression, deep neural network with a softmax classifier can vastly improve the performance in the case of not being able

to perform feature selection, which is important to find out features that significantly affect the outcome of the classification. In view of this, to achieve a classification of the extracted features, we therefore introduced a softmax classifier in the output layer,

$$o_k = \frac{e^{y_k}}{\sum_{i=1}^H e^{y_i}} \quad (12)$$

where  $o_k$  is the output of the  $k$ -th neuron, representing the observed probability of class  $k$ ;  $y_k$  is the associated linear output from the previous hidden layers; and  $H$  is the total number of output neurons in the softmax layer. Specifically, for all the layers above, the ReLU function (Equation (13)) was adopted as the activation function to avoid gradient diffusion during the training process. The gradient diffusion, also known as vanishing gradient problem, is a challenge in the training of artificial neural network by gradient-based learning methods and backpropagation [43]. In such method, the update of neuron's weights and biases requires to pass backwards the error signal from the previous layers. When the network is deep enough, the gradient of the loss function would be vanishingly small, effectively preventing the change of the weight. In the worst case, this may completely stop the neural network from further training, which explains the hindered usage of deep neural network in solving complex problems until recently when ReLU [27] function was introduced as a solution. Since



**Figure 1** The deep neural network model of DeepNitro

A total of eight neural levels were implemented. To reduce overfitting, a dropout process was introduced in the first three hidden layers. Additionally, for each dense layer (the fully-connected layer), ReLU was applied to avoid gradient diffusion. Blue circles indicate neurons in the input layer, the green and orange circles are neurons in the output layer. Red circles refer to neurons that are blocked by the dropout approach. ReLU, rectified linear unit.

the derivative of ReLU is constant at 1 for any input value greater than zero, the error signal would never attenuate when propagating down the network. This makes ReLUs favorable for the complex neural network model, and we can have very deep neural networks with ReLUs without the vanishing gradient problem.

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (13)$$

Before using the deep neural network for prediction, we must first optimize the network parameters for the training dataset. To achieve this goal, the negative log-likelihood (Equation (14)) was considered as a loss function in our optimization steps,

$$L = -\frac{1}{H} \sum_{i=1}^H o_i \log \hat{o}_i \quad (14)$$

where  $\hat{o}_i$  represents the predicted label, while  $o_i$  is the real label. Again,  $H$  is the total number of output neurons in the final layer. According to Equation (14), a mini-batch gradient descent algorithm is used to update network weights and biases during the back-propagating process. The gradients can then be computed as shown below,

$$w^l \leftarrow m \times w^l - \eta \frac{\partial E}{\partial w^l} \quad (15)$$

$$b^l \leftarrow m \times b^l - \eta \frac{\partial E}{\partial b^l} \quad (16)$$

where  $\eta$  is the learning rate. The batch size was optimized to be 30 for protein nitration and set at 50 for protein S-nitrosylation, respectively. To reduce overfitting, the momentum item was adopted for updating the weights and bias, and both the L1 and L2 regularization were introduced in the loss function (Equation (17)),

$$L = L + \lambda_1 \sum |\theta| + \lambda_2 \sum \theta^2 \quad (17)$$

where  $L$  is the loss function;  $\lambda_1$  and  $\lambda_2$  are coefficients for  $L_1$  and  $L_2$  regularization, respectively; and  $\theta$  is the weights and biases in a given layer.

Additionally, the dropout method was also implemented from the second to the fourth layer to improve the generalization capacity for the unknown dataset. Detailed parameter settings are also optimized by coarse grid search and are listed in Table S5.

The aforementioned deep neural network was implemented and trained by the *deeplearning4j* library in JAVA.

### Evaluation of the feature abstraction ability in the deep neural network

To further decipher the underlying mechanism of our constructed models, we have designed the following method to evaluate the feature abstraction abilities in the deep neural network. The abstracted features for each training and test dataset from the second to the seventh layer were computed based on the previously trained model. The abstracted features from each layer were then input into a simple multilayer perceptron (MLP) for training and testing. This simple MLP consisted of four fully-connected layers, and the detailed parameters are listed in Table S6. Using the training dataset, this MLP model

was retrained to classify the abstracted features. Next, the performance of the retrained model was evaluated on the independent test dataset, and the AUC value was calculated as criteria for quantifying the abstraction ability of each layer. To compare with other traditional feature selection approaches, the principle component analysis (PCA) was also performed on the raw features. Particularly, in order to avoid any bias, the raw features were compressed to the same dimension as each hidden layer from the deep neural network model. Similarly, the abstraction ability of PCA method was evaluated using the same procedure.

The abstraction ability per unit of feature is computed as below,

$$S_i = \frac{A_i - 0.5}{D_i} \quad (18)$$

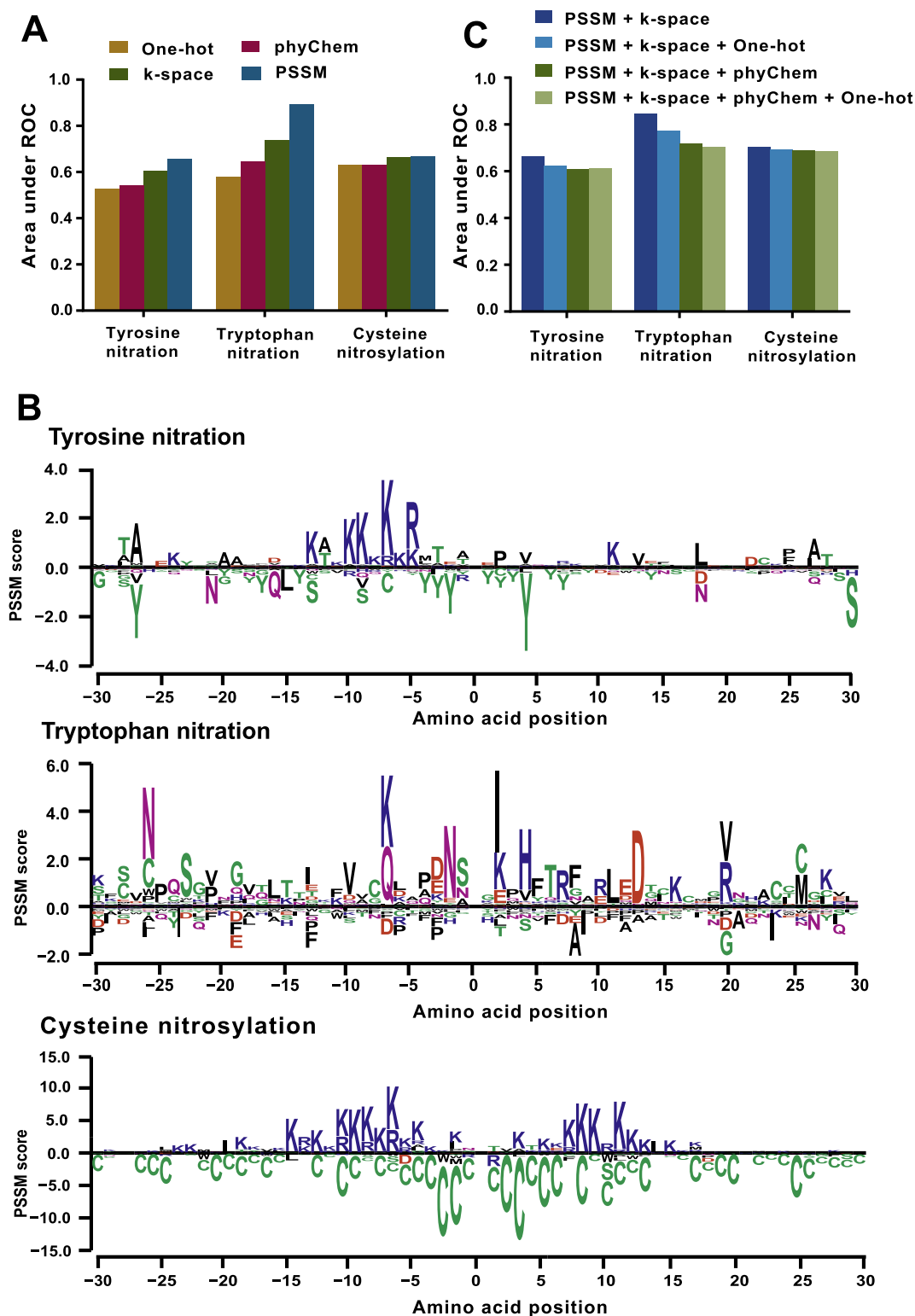
where  $A_i$  is the AUC value of the abstracted features from the  $i$ -th layer, and  $D_i$  is the number of abstracted features at this layer. Theoretically, an AUC value of 0.5 indicates a random classification model, and thus, AUC value greater than 0.5 would contribute to classification. Therefore, we subtract 0.5 from the calculated AUC value to quantify the real contribution, and further divided it by the feature's dimensionality to compute the abstraction ability per unit of feature.

## Results

### Construction of predictors

We first optimized the length of the flanking regions of a potential nitration or nitrosylation site by traversing the window size  $L$  from 10 to 50 using a 4-fold cross validation. Then various encoding schemes, including the one-hot binary encoding, PFR, k-space spectrum, and PSSM encoding, were applied to capture the sequence and physiochemical features of the flanking regions. As shown in Figure S2, increases in the window size improved the prediction performance of tyrosine nitration, tryptophan nitration, and S-nitrosylation. Taking into account both the prediction accuracy and computational burden, we finally selected a peptide length of 41 aa for subsequent training and prediction. Accordingly, a given nitration or nitrosylation site peptide could be encoded into a 3311-dimensional feature vector, when all the four feature encoding schemes above were used.

To capture all available biological properties, we designed a set of feature-encoding schemes for protein nitration and nitrosylation. However, we speculate that different types of features might contribute divergently to the prediction performance, and there might be a dependency relationship present among different features. Thus, we first examined the prediction capacities of the four encoding schemes by 4-fold cross-validation. The evaluation results of tyrosine nitration, tryptophan nitration, and S-nitrosylation showed a high coincidence. PSSM and k-space encoding were the two most effective feature capturing schemes for protein nitration and nitrosylation, enabling better classification than the other two schemes (Figure 2A). Although experimental evidence has shown that the formation of protein nitration and nitrosylation is mainly regulated by chemical side reactions produced via NO-related pathways, herein, we unexpectedly observed a weak contribution of physiochemical features to the prediction of



**Figure 2** Establishment of the optimal encoding scheme for DeepNitro

**A.** The prediction capabilities of the four different types of encoding schemes. **B.** The modified PSSM encoding features for protein nitration and nitrosylation. The PSSM scores were calculated according to our modified methods (see Methods section). The amino acid profiles were generated using seq2logo [49] based on the calculated PSSM scores. Position 0 denotes the nitrated or nitrosylated residue. **C.** The evaluation results of different combination of feature-encoding schemes. Feature-encoding schemes were added sequentially according to their prediction capabilities, and 4-fold cross-validation was applied to evaluate their prediction performance. PSSM, position specific scoring matrix.

potential nitrogen-containing modifications (Figure 2A). Among all the feature-coding schemes, we found that our modified PSSM scheme achieved an outstanding efficacy for extracting underlying features of the protein modification sites that were lack of consensus motifs. Instead of only calculating the conservation degree for positive data, we also took into account the contribution of negative data. In fact, by measuring the subtle differences between positive modification sites and negative residues, we could further extract the underlying rules for classification in comparison to the traditional PSSM method. For example, in our PSSM scheme, we observed that adjacent basic amino acid residues, such as arginine and lysine, were favorable for producing tyrosine nitration, while adjacent aromatic amino acid residues, such as tyrosine, hindered tyrosine nitration (Figure 2B, Figure S2). Similar patterns were also observed for cysteine S-nitrosylation. The proximal basic amino acid (arginine or lysine) residues showed a striking preference for positive data. Additionally, the cysteine residues located around the flanking region were deleterious to the modification (Figure 2B, Figure S2). Interestingly, the amino acid preference for tryptophan nitration seemed to be more site-specific compared with the other two types of modifications. The aliphatic or basic amino acid residues at positions -7, -2, +2, +4, and +13 contributed positively to the modification of the nitro-group on tryptophan. In contrast, the alanine residue at position +8 and isoleucine residue at position +9 contributed negatively to tryptophan nitration (Figure 2B, Figure S2).

Based on the observations above, we next sought to test the prediction performance for different combinations of the feature-encoding schemes. We evaluated the prediction capacities of the combined schemes using 4-fold cross-validation by adding the schemes sequentially into the prediction model based on the order of their contributions. As expected, using a combination of schemes could improve the prediction accuracy for tyrosine nitration and cysteine nitrosylation (Figure 2C). In particular, incorporation of the PSSM and k-space schemes exhibited optimal performance. Therefore, herein we chose PSSM and k-space as the final feature-encoding schemes and constructed a 2040-dimensional numeric vector for both tyrosine nitration and cysteine nitrosylation prediction. For the tryptophan nitration, there seems to be a different trend between the scheme combinations and the prediction performance. PSSM encoding showed the strongest capacity for prediction, whereas the integration of other features seemed to weaken its prediction capacity. Notably, the prediction performance of the constructed model decreased with increased number of integrated schemes (Figure 2C). Due to the limited number of known nitration sites available in our training dataset, the application of multiple features during the encoding process will introduce extra noise for classification, which may hinder the improvement of the prediction accuracy. Therefore, to obtain the most appropriate prediction model, we selected only the PSSM as the final scheme for feature encoding of tryptophan nitration. A 40-dimension feature vector was inputted to the deep neural network model for training and prediction.

Based on the aforementioned feature selection strategies, we introduced a deep neural network model to construct the predictor called DeepNitro for the detection of potential nitration or nitrosylation sites (Figure 1).

## Evaluation of the feature abstraction abilities in the constructed predictors

In deep learning-based method, as a benefit of its hierarchical architecture, input features can be precisely abstracted along the successive levels and thereby discovering informative patterns for subsequent classification or regression tasks. The following method was developed to evaluate the feature abstraction abilities of our constructed predictors. Firstly, we extracted the output signals from each hidden layer as the abstracted features, and input them into a simple multilayer perceptron (MLP) to test their classification performance. Using our collected dataset, we retrained the MLP model and calculated the AUC value under the independent testset. To compare with other feature selection approaches, the principle component analysis (PCA) was also performed on the raw input data and the selected components were then propagated forward the same MLP to compute its performance. Expectedly, the abstracted features selected from the nitration and nitrosylation models all outperformed those from the PCA method, suggesting that our method have a better feature abstraction abilities than the traditional approaches (Figure S3). We further computed the abstraction efficiency per unit of feature for both the deep neural network and PCA method. Obviously, the abstraction efficiencies in each hidden layer showed that deep neural network is able to extract effective features by transforming signals through successive layers. As the number of layers increased, more informative features were obtained. Moreover, compared to PCA method, deep neural network can preserve more information under the same compression ratio.

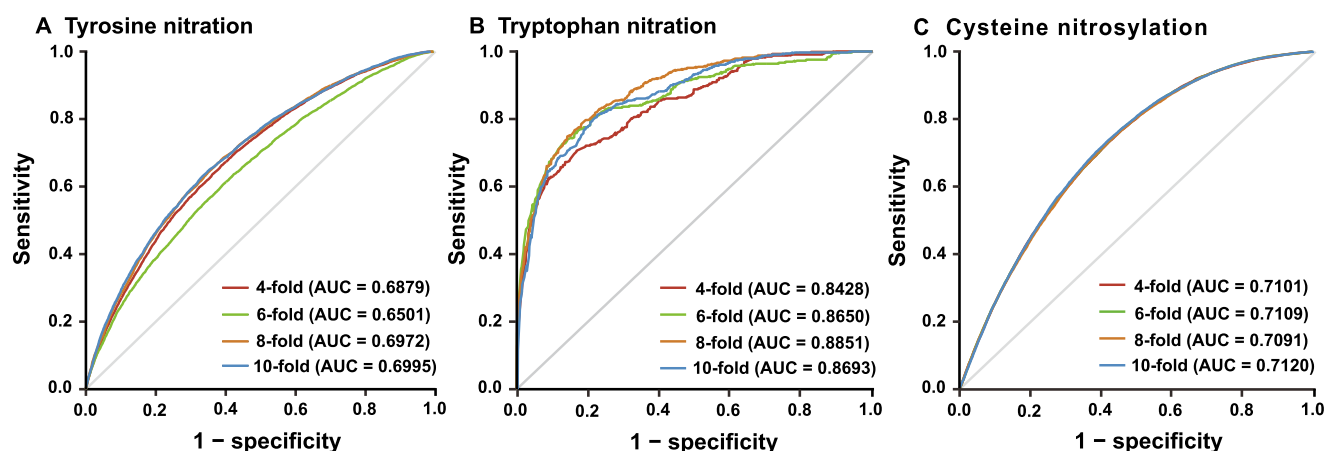
Taken together with the observations above, the application of a deep neural network in our study could automatically extract high-level recognition patterns for protein post-translational modifications, and help to eliminate irrelevant features or reduce noise in the training process.

## Evaluation of the prediction performance

To evaluate the prediction performance of DeepNitro, we performed 4-, 6-, 8-, and 10-fold cross-validation of the training dataset. As a result, DeepNitro showed an acceptable performance in n-fold cross-validation with the area under the ROC curve (AUC) close to 0.7 for tyrosine nitration (Figure 3A). For tryptophan nitration, the AUC values were mostly greater than 0.85, indicating a satisfactory prediction performance (Figure 3B). For S-nitrosylation, all the tested AUC values were greater than 0.7 (Figure 3C). Furthermore, for both nitration and nitrosylation, the ROC curves of the 4-, 6-, 8-, and 10-fold cross-validations were very close to each other, supporting the robustness of our constructed predictors. Since the positive and negative datasets were highly imbalanced in our training dataset, we then calculated the precision-recall curves to further evaluate the performance of our prediction models (Figure S4). The results further indicate that DeepNitro is accurate and robust in predicting novel nitration and nitrosylation sites, even in the case of an imbalanced data dataset.

In our prediction models, we expect that application of deep neural network may help to uncover the underlying





**Figure 3** Performance evaluations of the prediction models for tyrosine nitration, tryptophan nitration, and the *S*-nitrosylation. The 4, 6, 8, and 10-fold cross-validations were performed on the tyrosine nitration (A), the tryptophan nitration (B), and the cysteine nitrosylation (C) models.

sequence features of protein nitration and nitrosylation from the training dataset. To prove this point, we further compared our deep neural network models to other shallow machine learning methods for their abilities to interpret the training dataset. The two most widely-adopted algorithms, support vector machine (SVM) and random forest (RF), were compared and a 4-fold cross-validation was carried out to evaluate their performance. As shown in Figure S5, applying the eight-layer deep neural network substantially improved the prediction capability for tyrosine nitration and *S*-nitrosylation over SVM and RF, indicating that the deep neural network models are more powerful in interpreting the underlying information for training dataset. However, for tryptophan nitration, random forest seemed to be the optimal algorithm. Generally speaking, training deep neural network typically requires a certain amount of training dataset to maintain the robustness and accuracy of the model. But in tryptophan nitration, the sample size is quite small, and therefore, limiting the performance of deep neural network. The advantage of deep neural network can be further demonstrated as the training dataset expands in the near future.

To rigorously evaluate the prediction performance of DeepNitro, we next compared it with other state-of-art predictors using an independent dataset. For tyrosine nitration, we selected iNitro and our previous tool, GPS-YNO2, for comparison. To ensure that the comparison was unbiased, all the different thresholds for these tools were used. The evaluation results of DeepNitro in the independent dataset agreed well with those from the *n*-fold cross-validation, indicating that our model provides robust results for new data. In comparison to other available tools, DeepNitro also showed superior prediction performance (Figure 4A). For *S*-nitrosylation, iSNO-AAPair, SNOsite, and GPS-SNO were selected to perform the comparison. Since iSNO-AAPair and SNOsite did not provide prediction scores for all potential cysteine residues, we could only compute the prediction performance under their default thresholds. As presented in Figure 4B, our model achieved the highest AUC value of 0.7437, outperforming other prediction software. To our knowledge, our study is the first attempt to establish a prediction model for tryptophan

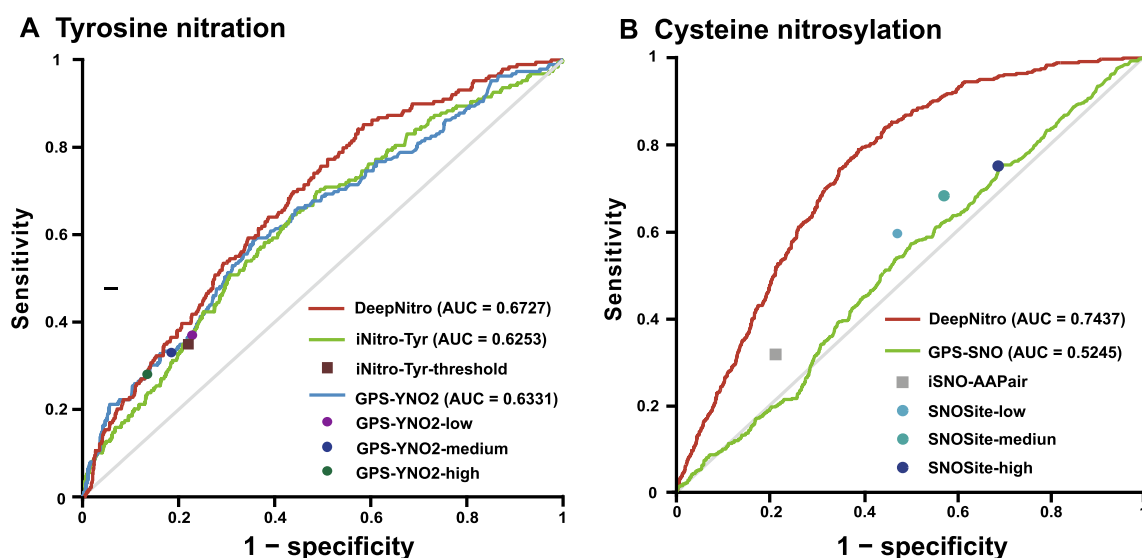
nitration for the biology community, and therefore a performance comparison was not performed for tryptophan nitration.

#### Development of the DeepNitro web server

To facilitate the use of our prediction models, we next developed an online predictor called DeepNitro for the community, which is freely available at <http://deepnitro.renlab.org>. DeepNitro only requires protein sequences to run a prediction. The prediction of tyrosine nitration, tryptophan nitration, and cysteine nitrosylation are well supported in our predictor, and users can select the modification types of their interest in the options panel. To balance the prediction accuracy of each modification type, we selected three thresholds with high, medium, and low stringencies based on the evaluation results (Figure 5A). The detailed performance values under these three thresholds are shown in Table S7.

After the query sequences are submitted to DeepNitro, users can check its running status in the result panel in real time. When the prediction is complete, a button that links out to the result page automatically appears. Figure 5B provides a snapshot for the result page of human dynamin-1-like (DNM1L) protein and heterogeneous nuclear ribonucleoprotein D-like (HNRNPDL) protein. The prediction position, score, and modification type of the input proteins were first listed in an interactive table, which allows the users to easily search and sort the results. Remarkably, to facilitate a further analysis of the protein function, we also implemented an automatic pipeline for visualizing the prediction results. By integrating IBS [44] and InterProScan [45] into the web server, DeepNitro can present the graphical representation of the input proteins together with their predicted sites and domain organization in the visualization panel (Figure 5C). In order to allow a mass prediction of protein nitration and nitrosylation sites, a standalone package, which is available at <http://deepnitro.renlab.org>, was also developed. Like the web server, the prediction of multiple modification types and visualization of the predicted results was supported.

Besides the functionalities above, we also integrated a database covering all the nitration and nitrosylation sites we



**Figure 4** Performance comparison of the tyrosine nitration and the *S*-nitrosylation prediction models for the independent benchmarking dataset. An independent test set was used to evaluate the prediction capability of DeepNitro and other existing tools on protein nitration (A) and nitrosylation sites (B). The default prediction thresholds of each compared tool were also marked on the ROC curves. Since three prediction thresholds with low, medium and high stringencies were provided in GPS-YN02 and SNOSite, we separately marked them on the corresponding ROC curves. ROC, receiver operating characteristic curve.

collected into the web server. By searching with annotation keywords or protein sequences, users can easily get access to the modification data of interest and perform further functional analysis to uncover the potential role of protein nitration and nitrosylation.

## Discussion

Protein nitration and nitrosylation are widespread modifications that modulate diverse aspects of cellular functions. Unlike other PTMs, nitration and nitrosylation are induced by a series of chemical reactions rather than enzymatic processes. As a consequence, this kind of modification usually lacks consensus motifs, making it difficult to predict the exact sites using bioinformatics algorithms. To overcome this challenge, we present a novel computational tool, DeepNitro, for predicting potential nitration and nitrosylation sites. First, we constructed new schemes for encoding the potential modification sites based on primary sequence features. Then, the deep learning algorithm was applied for model training and prediction.

DeepNitro shows superior performance for both protein nitration and nitrosylation compared with existing tools. Its prediction capability is enhanced mainly by the new encoding schemes adopted in this study, especially the PSSM method. In contrast to the traditional process, our method not only measures the conservation in positive data, but also takes into account the comparison of residue preference between positive and negative data. The subtle differences between positive and negative data may act as a key factor to improve the distinguishing capability of our model.

In addition to the encoding schemes, a new model training method is also introduced in our study. Recently, the deep artificial neural network has received increasing attention in the field of machine learning. By propagating raw data down the

deep networks, underlying features and highly complicated functions can be effectively encapsulated, increasing the classification and regression capabilities for the input data. Currently, the deep neural network has been shown to improve performance in image [46] and speech recognition [32], natural language understanding [47], and most recently, in computational biology [34,35]. As we expected, the application of the deep neural network in the current study has introduced remarkable performance gains into our model. The deep neural network allows us to better handle high-dimensional encoding vectors by training complex networks with multiple layers that capture their internal relationship. Compared to other traditional machine learning algorithms, application of this new method can discover high-level features and increase interpretability of protein nitration and nitrosylation.

Although promising performance was obtained using DeepNitro, there is still room for refinement. First, to reduce the computational burden, we only considered the primary sequence feature in the current algorithm. Recent studies have indicated that the tertiary structure is another key feature for determining the occurrence of protein nitration or nitrosylation [38]. Therefore, considering sequence features only will introduce bias into the prediction model. Consequently, we will further introduce a structural encoding scheme, such as features for peptide secondary and three-dimensional structures, in our future versions. Second, novel deep learning architectures will also be applied in the next version of DeepNitro to improve its performance. We are currently working to predict the potential nitration and nitrosylation sites through a deep fully-connected network, which neglects the contextual dependencies of a given residue [48]. To measure the contextual dependencies in nitration and nitrosylation site prediction, the recurrent neural network (RNN) will be integrated in future developments. With the help of RNN model in measuring sequence contextual dependency, we expect that

**A**

or select local files to upload.

100%

+ Select Files

Examples:

◀ Fasta example

\* fasta format supported  
\* Paste file size < 2M

Clear

Options

S-Nitrosylation Threshold:

Medium

Tyr-Nitration Thredshold:

Medium

Trp-Nitration Threshold:

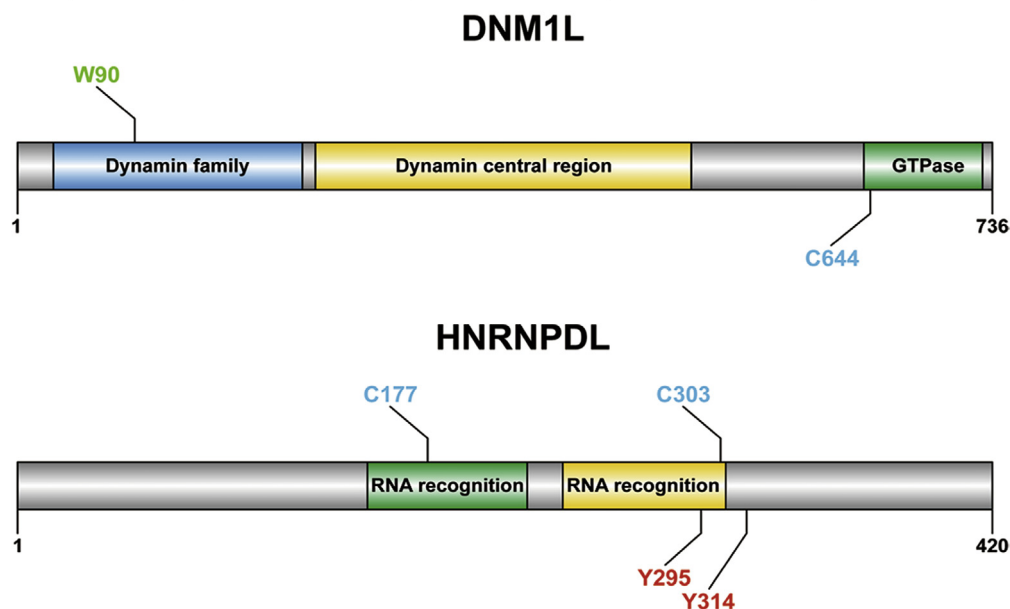
Medium

**B**

Download ▾

Search

ID	Position	Peptide	Prediction Score	Prediction Cutoff	Modification Type
DNM1L	90	NGVEAEEWGF <sup>W</sup> LHTK	0.759	0.575	Tryptophan Nitration
DNM1L	345	TKFATEY <sup>C</sup> NTIEGTA	0.477	0.37	S-Nitrosylation
DNM1L	644	SAREQ <sup>R</sup> DC <sup>E</sup> VIERLI	0.454	0.37	S-Nitrosylation
HNRNPDL	177	RFGEVVD <sup>C</sup> TIKTDPV	0.476	0.37	S-Nitrosylation
HNRNPDL	277	TNERRG <sup>F</sup> CFITYTDE	0.412	0.37	S-Nitrosylation
HNRNPDL	295	KKLLES <sup>R</sup> YHQIGSGK	0.513	0.295	Tyrosine Nitration
HNRNPDL	303	HQIGSGK <sup>C</sup> EIKVAQP	0.461	0.37	S-Nitrosylation
HNRNPDL	314	VAQPKEV <sup>R</sup> RQQQQQQ	0.296	0.295	Tyrosine Nitration

**C****Figure 5** A snapshot of the DeepNitro server

**A.** The main interface of DeepNitro. Protein sequences can be input into the text area or uploaded as a single FASTA file. Thresholds with high, medium, and low stringencies are provided in the options panel. Detailed information for the predicted modification sites, such as protein name, modified position, flanking peptide, prediction score, and prediction threshold, is listed in the interactive table (**B**) and visualized using IBS and InterProScan (**C**). DNM1L, dynamin-1-like protein; HNRNPDL, heterogeneous nuclear ribonucleoprotein D-like; IBS, illustrator for biological sequences.

such an advanced architecture can greatly improve the prediction capability of non-consensus protein modifications such as nitration and nitrosylation.

## Authors' contributions

JR and ZZ conceived the project and supervised the study. YBX designed the algorithm and drafted the manuscript. XL and YL implemented the algorithm and performed the evaluation. LC developed the software and established the web service. WM, JH, JC, YZ, and YX performed the data analysis and help to draft the discussion section. All authors read and approved the final manuscript.

## Competing interests

The authors have declared no competing interests.

## Acknowledgments

This work was supported by grants from the National Natural Science Foundation of China (Grant Nos. 91753137, 31471252, 31771462, 81772614, and U1611261); National Key Research and Development Program of China (Grant No. 2017YFA0106700); Guangdong Natural Science Foundation (Grant Nos. 2014TQ01R387 and 2014A030313181); Science and Technology Program of Guangzhou, China (Grant Nos. 201604020003 and 201604046001); and China Postdoctoral Science Foundation (Grant No. 2017M622864).

## Supplementary material

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.gpb.2018.04.007>.

## References

- [1] Forstermann U, Sessa WC. Nitric oxide synthases: regulation and function. *Eur Heart J* 2012;33:829–37, 37a–d.
- [2] Ferrer-Sueta G, Radi R. Chemical biology of peroxynitrite: kinetics, diffusion, and radicals. *ACS Chem Biol* 2009;4:161–77.
- [3] Gladwin MT, Kim-Shapiro DB. Vascular biology: nitric oxide caught in traffic. *Nature* 2012;491:344–5.
- [4] Mikkelsen RB, Wardman P. Biological chemistry of reactive oxygen and nitrogen and radiation-induced signal transduction mechanisms. *Oncogene* 2003;22:5734–54.
- [5] Greenacre SA, Ischiropoulos H. Tyrosine nitration: localisation, quantification, consequences for protein function and signal transduction. *Free Radic Res* 2001;34:541–81.
- [6] Nuriel T, Hansler A, Gross SS. Protein nitrotryptophan: formation, significance and identification. *J Proteomics* 2011;74:2300–12.
- [7] Stamler JS, Lamas S, Fang FC. Nitrosylation. the prototypic redox-based signaling mechanism. *Cell* 2001;106:675–83.
- [8] Zaragoza R, Torres L, Garcia C, Eroles P, Corrales F, Bosch A, et al. Nitration of cathepsin D enhances its proteolytic activity during mammary gland remodelling after lactation. *Biochem J* 2009;419:279–88.
- [9] Adams L, Franco MC, Estevez AG. Reactive nitrogen species in cellular signaling. *Exp Biol Med (Maywood)* 2015;240:711–7.
- [10] Bonavida B, Garban H. Nitric oxide-mediated sensitization of resistant tumor cells to apoptosis by chemo-immunotherapeutics. *Redox Biol* 2015;6:486–94.
- [11] Kasten D, Mithofer A, Georgii E, Lang H, Durner J, Gaupels F. Nitrite is the driver, phytohormones are modulators while NO and H<sub>2</sub>O<sub>2</sub> act as promoters of NO<sub>2</sub>-induced cell death. *J Exp Bot* 2016;67:6337–49.
- [12] Gonzalez R, Cruz A, Ferrin G, Lopez-Cillero P, Fernandez-Rodriguez R, Briceno J, et al. Nitric oxide mimics transcriptional and post-translational regulation during alpha-tocopherol cytoprotection against glycochenodeoxycholate-induced cell death in hepatocytes. *J Hepatol* 2011;55:133–44.
- [13] Bajor M, Zareba-Kozioł M, Zhukova L, Goryca K, Poznanski J, Wyslouch-Cieszyńska A. An interplay of S-nitrosylation and metal ion binding for astrocytic S100B protein. *PLoS One* 2016;11:e0154822.
- [14] Chen HJ, Yang YF, Lai PY, Chen PF. Analysis of chlorination, nitration, and nitrosylation of tyrosine and oxidation of methionine and cysteine in hemoglobin from type 2 diabetes mellitus patients by nanoflow liquid chromatography tandem mass spectrometry. *Anal Chem* 2016;88:9276–84.
- [15] Upmacis RK. Atherosclerosis: a link between lipid Intake and protein tyrosine nitration. *Lipid Insights* 2008;2008:75.
- [16] Piroddi M, Palmese A, Pilolli F, Amoresano A, Pucci P, Ronco C, et al. Plasma nitroproteome of kidney disease patients. *Amino Acids* 2011;40:653–67.
- [17] Turko IV, Murad F. Protein nitration in cardiovascular diseases. *Pharmacol Rev* 2002;54:619–34.
- [18] Nakamura T, Tu S, Akhtar MW, Sunico CR, Okamoto S, Lipton SA. Aberrant protein S-nitrosylation in neurodegenerative diseases. *Neuron* 2013;78:596–614.
- [19] Cook SL, Jackson GP. Characterization of tyrosine nitration and cysteine nitrosylation modifications by metastable atom-activation dissociation mass spectrometry. *J Am Soc Mass Spectrom* 2011;22:221–32.
- [20] Liu Z, Cao J, Ma Q, Gao X, Ren J, Xue Y. GPS-YNO2: computational prediction of tyrosine nitration sites in proteins. *Mol Biosyst* 2011;7:1197–204.
- [21] Xu Y, Wen X, Wen LS, Wu LY, Deng NY, Chou KC. iNitro-Tyr: prediction of nitrotyrosine sites in proteins with general pseudo amino acid composition. *PLoS One* 2014;9:e105018.
- [22] Xue Y, Liu Z, Gao X, Jin C, Wen L, Yao X, et al. GPS-SNO: computational prediction of protein S-nitrosylation sites with a modified GPS algorithm. *PLoS One* 2010;5:e11290.
- [23] Xu Y, Ding J, Wu LY, Chou KC. iSNO-PseAAC: predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition. *PLoS One* 2013;8:e55844.
- [24] Lee TY, Chen YJ, Lu TC, Huang HD, Chen YJ. SNOsite: exploiting maximal dependence decomposition to identify cysteine S-nitrosylation with substrate site specificity. *PLoS One* 2011;6:e21849.
- [25] Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies', a field guide to dynamical recurrent neural networks. Piscataway: IEEE Press; 2001.
- [26] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Comput* 2006;18:1527–54.
- [27] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks.. *Proc 14th Intl Conf Artif Intell Stat* 2011;315–23.
- [28] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 2014;15:1929–58.
- [29] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521:436.
- [30] Sutskever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning. *Intl Conf Mach Learn* 2013;1139–47.



- [31] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. *Proc IEEE Conf Comput Vis Pattern Recognit* 2015;1–9.
- [32] Hinton G, Deng L, Yu D, Dahl GE, Mohamed AR, Jaitly N, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process Mag* 2012;29:82–97.
- [33] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. *Proc 25th Intl Conf Mach Learn* 2008;160–7.
- [34] Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nat Biotechnol* 2015;33:831.
- [35] Di Lena P, Nagata K, Baldi P. Deep architectures for protein contact map prediction. *Bioinformatics* 2012;28:2449–57.
- [36] Leung MK, Xiong HY, Lee LJ, Frey BJ. Deep learning of the tissue-regulated splicing code. *Bioinformatics* 2014;30:i121–9.
- [37] Quang D, Chen Y, Xie X. DANN: a deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics* 2015;31:761–3.
- [38] Radi R. Protein tyrosine nitration: biochemical mechanisms and structural basis of functional effects. *Acc Chem Res* 2013;46:550–9.
- [39] Bartesaghi S, Ferrer-Sueta G, Peluffo G, Valez V, Zhang H, Kalyanaraman B, et al. Protein tyrosine nitration in hydrophilic and hydrophobic environments. *Amino Acids* 2007;32:501–15.
- [40] Kidera A, Konishi Y, Oka M, Ooi T, Scheraga HA. Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *J Protein Chem* 1985;4:23–55.
- [41] Vacic V, Iakoucheva LM, Radivojac P. Two sample logo: a graphical representation of the differences between two sets of sequence alignments. *Bioinformatics* 2006;22:1536–7.
- [42] Bridle JS. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: Fogelman Soulié Françoise, Hérault Jeanny, editors. *Neurocomputing*. Berlin Heidelberg: Springer; 1990. p. 227–36.
- [43] Goh GB, Hodas NO, Vishnu A. Deep learning for computational chemistry. *J Comput Chem* 2017;38:1291–307.
- [44] Liu W, Xie Y, Ma J, Luo X, Nie P, Zuo Z, et al. IBS: an illustrator for the presentation and visualization of biological sequences. *Bioinformatics* 2015;31:3359–61.
- [45] Jones P, Binns D, Chang HY, Fraser M, Li W, McAnulla C, et al. InterProScan 5: genome-scale protein function classification. *Bioinformatics* 2014;30:1236–40.
- [46] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 2012;1097–105.
- [47] Sarikaya R, Hinton GE, Deoras A. Application of deep belief networks for natural language understanding. *IEEE-ACM Trans Audio Speech* 2014;22:778–84.
- [48] Liu N, Han J. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *IEEE Trans Image Process* 2018;27:3264–74.
- [49] Thomsen MC, Nielsen M. Seq2Logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. *Nucleic Acids Res* 2012;40:W281–7.