Project 02. Prediction of protein S-Nitrosylation sites

Xác định vị trí S-Nitrosyl hóa trong protein

State-of-the-art: 2021

https://ieeexplore.ieee.org/abstract/document/9313999/media#media

A. Siraj, T. Chantsalnyam, H. Tayara and K. T. Chong, "RecSNO: Prediction of Protein S-Nitrosylation Sites Using a Recurrent Neural Network," in IEEE Access, vol. 9, pp. 6674-6682, 2021, doi: 10.1109/ACCESS.2021.3049142.

I. Bài toán

Sửa đổi S-Nitrosylation là một trong những sửa đổi sau dịch mã quan trọng nhất; nó đóng một vai trò quan trọng trong một loạt các quá trình sinh học và có liên quan đến các bệnh khác nhau. Việc xác định các vị trí S-Nitrosyl hóa trong protein là rất quan trọng để hiểu và kiểm soát các quá trình sinh học cơ bản. Các phương pháp xác định thực nghiệm thông thường tốn nhiều công sức và hiệu quả về chi phí. Để khắc phục những vấn đề này, các phương pháp tiếp cận sinh học tính toán đang được xem xét, bao gồm việc sử dụng các thuật toán học máy và học sâu.

II. Tập dữ liệu

Tập dữ liệu training: 6748 với 3383 positive and 3365 negative sites

Tập dữ liệu test: 3519 mẫu với 351 positive and 3168 negative sites

Website: http://nsclbio.jbnu.ac.kr/tools/RecSNO/

Định dạng dữ liệu: một chuỗi 41 ký tự với ký tự C ở giữa

Ví dụ: MAQDQGEKENPMRELRIRKLCLNICVGESGDRLTRAAKVLE

Phải phân loại chuỗi này thành 2 loại: nonSNO và SNO

III. Đánh giá kết quả

III.1. Các độ đo

$$\begin{cases} Sensitivity = \dfrac{T_p}{T_p + F_n} \\ Specificity = \dfrac{T_n}{T_n + F_p} \\ Accuracy = \dfrac{T_p + T_n}{T_p + T_n + F_p + F_n} \\ MCC = \dfrac{(T_p)(T_n) - (F_p)(F_n)}{\sqrt{(T_p + F_p)(T_p + F_n)(T_n + F_n)(T_n + F_p)}} \end{cases}$$

III.2. State-of-the-Arts

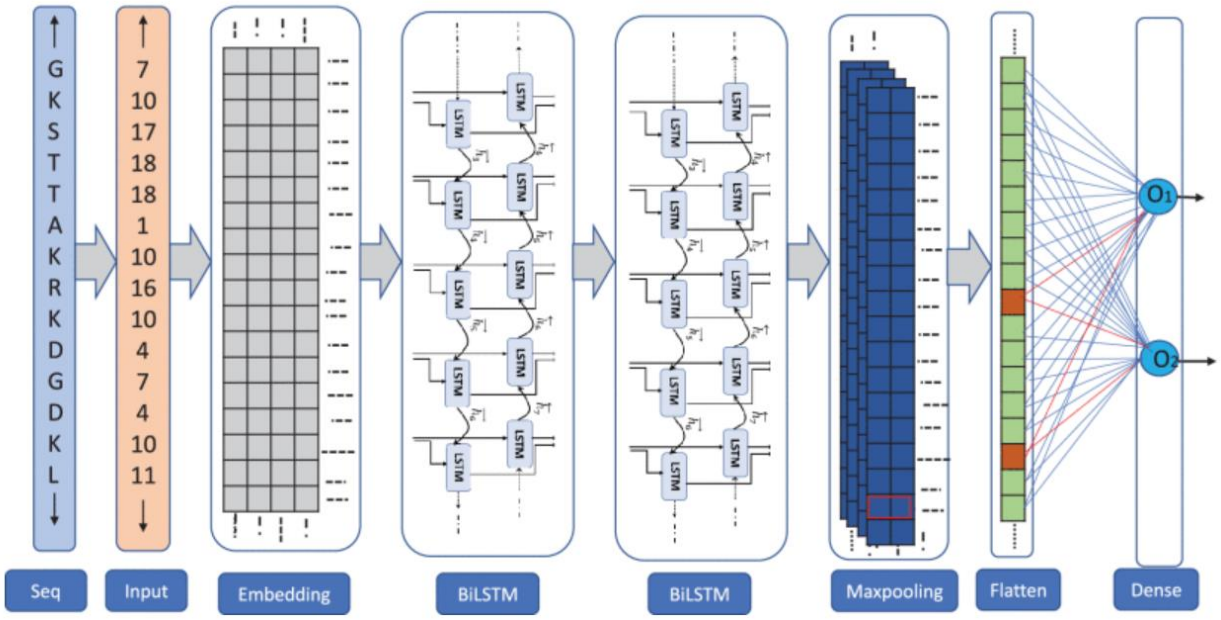
TABLE 5 Independent Dataset Comparison of RecSNO With Existing Predictors

Predictors	Sensitivity	Specificity	Accuracy	MCC	AUC
GPS-SNO	0.28	0.74	0.69	0.01	0.52
iSNOPseAAC	0.29	0.76	0.71	0.03	NA
SNOSite	0.67	0.45	0.47	0.07	NA
DeepNitro	0.58	0.76	0.73	0.22	0.73
PreSNO	0.60	**0.77**	**0.75**	0.25	0.76
RecSNO	**0.77**	0.71	0.71	**0.30**	**0.80**

IV. Phương pháp State-of-the-Arts

https://ieeexplore.ieee.org/abstract/document/9313999/media#media

A. Siraj, T. Chantsalnyam, H. Tayara and K. T. Chong, "RecSNO: Prediction of Protein S-Nitrosylation Sites Using a Recurrent Neural Network," in IEEE Access, vol. 9, pp. 6674-6682, 2021, doi: 10.1109/ACCESS.2021.3049142.



Here, we develop a DL-based classifier for SNO prediction using a combined word embedding and BiLSTM approach. This classifier contains six layers, as shown in Figure 3. These layers include: (1) an input layer, in which a residue fragment of length 41 (including the pseudo-amino acid '−') is converted via integer encoding; (2) an embedding layer, which is used to represent properties in the form of a word vector such that, every peptide in the sequence is converted into a 64-dimensional word vector; (3) two consecutive BiLSTM layers, one with 32-memory units and the other with 24. The first BiLSTM layer takes n-dimensional word vectors as input and extracts the features of those inputs. The result of the first BiLSTM layer is passed as input to the second BiLSTM layer, which extracts the features more deeply; (4) a max-pooling layer, which reduces the dimensions to half. The max-pooling layer preserves the features with maximum values in pool size; and (5) a prediction layer, which contains two neurons activated by the 'softmax' activation function and, provides a probability score for each class. We use dropout layers with different probabilities. After finding the best hyper-parameters for each layer with grid search, the hyper-parameter setting information for each layer is defined (shown in Table 1), except the given hyper-parameters values for each layer set as default.

TABLE 1 Proposed Model Layer Details

Layers	hyperparameter Settings	Output shape
Embedding	Input dim = 24 Output dim = 64 Input shape = (41,)	(41,64)
BiLSTM	LSTM units = 32 Kernal reg = L2($1e^{-4}$) Recurrent reg = L2($1e^{-4}$) Bias reg = L2($1e^{-4}$)	(41,64)
Dropout	Rate = 0.1	(41,64)
BiLSTM	LSTM units = 24 Kernal reg = L2($1e^{-2}$) Recurrent reg = L2($1e^{-2}$) Bias reg = L2($1e^{-2}$)	(41,48)
Dropout	Rate = 0.2	(41,48)
Max Pooling	Pool size = 2	(20,48)
Flatten	Just flatten the matrix	(960)
Dropout	Rate = 0.2	(960)
Dense	Activation = softmax Units = 2	(2)

In our proposed model, We used batch size of 12 and applied Adam optimizer to our framework, which merges the dividend of both the adaptive gradient algorithm and root mean square propagation, resulting in effective training [68]. We also used early stopping to monitor validation loss with a patience of 5 for stop training because further training would increase the variance of the model and lead to overfitting. We also used a learning rate schedular after 20 epochs, which decreased the learning rate by multiplying it by (e−1). The architecture was implemented using the Keras (https://keras.io/) deep learning library. Since we used softmax-based prediction, a categorical cross-entropy function was used as the loss function and the results were obtained by applying a threshold of 0.5.

V. Model Evaluation and Performance Metrics

The present study uses stratified k-fold cross validation, the folds are generally formed in such a way as to be consisted of almost the same proportion of predictor labels as original dataset. Studies have shown that stratified cross validation

generates comparative upshots with lower bias and lower variance when compared to regular cross validation [69]. we used 5-fold strategy, in which the data are divided into 5 equal bunches by which one part is used for validation and the remaining four parts are used for training. The technique persists untill each fold is sorted out as validation data and assesses the performance of the model using different types of matrices, including a confusion matrix, matthew's correlation coefficient (MCC), receiver operating characteristics (ROC) curve and precision-recall curve (PRC). A confusion matrix is one of the basic matrix used to assess the quality of the classification predictor. A confusion matrix envisages the results in the form of a matrix where each column constitutes the predicted result and each row indicates the actual class of the sample. A confusion matrix relies on four values, the number of true positives (Tp), the number of true negatives (Tn), the number of false-positive (Fp), and the number of false negatives (Fn). Another performance matrices used confusion matrix as.

$$\begin{cases} Sensitivity = \dfrac{T_p}{T_p + F_n} \\ Specificity = \dfrac{T_n}{T_n + F_p} \\ Accuracy = \dfrac{T_p + T_n}{T_p + T_n + F_p + F_n} \\ MCC = \dfrac{(T_p)(T_n) - (F_p)(F_n)}{\sqrt{(T_p + F_p)(T_p + F_n)(T_n + F_n)(T_n + F_p)}} \end{cases}$$

Sensitivity (SN) is a measure of the accurate positive rate and Specificity (SP) represents the true negative rate of the classifier. Accuracy (ACC) is the proportion of all accurately predicted samples, both positive and negative. MCC is a balanced measure in which true and false negatives are both used in the evaluation.The area under the ROC curve is used to indicate the degree of quality and separability of the classification models. The PRC is the tradeoff between precision and recall using different threshold. The higher area under the curve is the representation of both the high recall and the high precision. As the high value of precision is due to a low false positive rate,while the high recall is due to low false negative rate.