# SUMO-NN : a neural networks based method to predict SUMOylated lysines in human proteins

## Motivation

Over the last two decades, biochemical and genetics experiments have unearthed the biological significance of a covalent post-translational modification called SUMOylation. Disruption of SUMOylation has been linked to neurodegenerative diseases and cancer. Lysine residues in human proteins undergo SUMOylation. Experimental determination of SUMOylated lysines is very difficult. Hence, computational methods to predict SUMOylated lysines are very important. Present work proposes and demonstrates a new prediction tool, SUMO-NN.

## Dataset Creation

The pipeline used for dataset creation is summarized in following steps -

1. The list of human proteins and SUMOylated lysines therein was obtained from a mass-spectrometry coupled proteomics study Hendriks, et al, 2018 (https://www.nature.com/articles/s41467-018-04957-4).
2. Protein sequences were downloaded from the UniProt database (https://www.uniprot.org/).
3. Sequence redundancy was removed using linclust program in MMseqs2 suite of softwares (https://github.com/soedinglab/MMseqs2) .
4. For every lysine in all proteins in the dataset, 15-mers were extracted. This was done by including 7 amino acid residues before and 7 amino acid residues after the lysine and the lysine is at centre. In case of lysines at N-terminus or C-terminus, gaps were introduced as padding, to adjust the length of the 15-mer.

In the SUMO-NN source code, the folder “dataset_creation” contains files “positive_training_data.tsv”, “negative_training_data.tsv”, “positive_testing_data.tsv” and “negative_testing_data.tsv”. These files contain data for training and testing datasets respectively.

Details of training and testing datasets used in this study are given below (Table-1).

**Table-1**: Overview of training and testing datasets used by SUMO-NN

| **Term** | **Training Dataset** | **Testing Dataset** |
|---|---|---|
| Number of proteins | 3000 | 771 |
| Positive dataset / number of SUMOylated lysines | 10813 | 3035 |
| Negative dataset / all the remaining lysines | 123888 | 33785 |

For both training and testing datasets, the size of the negative dataset is more than 11 times the size of the positive dataset. If the entire negative dataset was given as input to the neural network, it would have introduced bias in predictions. Hence, the negative dataset was undersampled and was randomly chosen in a 1:1 ratio with respect to the positive dataset. Thus, for the negative training dataset, 10813 15-mers were randomly chosen from a total of 123888 15-mers. Similarly, for the negative testing dataset, 3035 15-mers were randomly chosen out of a total of 33785 15-mers.

**Method**

SUMO-NN has been implemented in the Python version 3.11.6 and PyTorch version 2.0.1+cpu. The model uses learning rate = 0.05 and takes 51 epochs to

train and make predictions. SGD optimizer and the entropy function BCEWithLogitsLoss were used. More details about SUMO-NN can be found in the program "predict_lysines.py" in the "src" folder of the source code. Predictions are saved to a file named "predictions_on_test_data.tsv".

**Performance**

Model performance was assessed using statistical metrics such as sensitivity (Sn), specificity (Sp), accuracy (Ac), F1 score (F1) and Matthews Correlation Coefficient (MCC).

Given below are equations for calculating above mentioned parameters. Here, TP = true positives, FP =false positives, TN = true negatives and FN = false negatives.

$$Sn = TP / (TP + FN)$$

$$Sp = TN / (TN + FP)$$

$$Ac = (TP + TN) / (TP + TN + FP + FN)$$

$$F1 = TP / (TP + 0.5 * (FP + FN))$$

$$MCC = (TP * TN - FP * FN) / denominator$$

$$denominator = ((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))^{0.5}$$

**Table-2:** Performance assessment of SUMO-NN on testing dataset

| **Metric** | **Value** |
|---|---|
| True Positives | 2211 |
| False Positives | 1254 |
| True Negatives | 1781 |

| | |
|---|---|
| False Negatives | 824 |
| Sensitivity | 0.729 |
| Specificity | 0.587 |
| Accuracy | 0.658 |
| F1 score | 0.68 |
| MCC | 0.319 |