# ResLysEmbed : A ResNet-Based Framework for Succinylated Lysine Residue Prediction Using Sequence and Language Model Embeddings

Souvik Ghosh[1], Md Muhaiminul Islam Nafi[1], and M Saifur Rahman[1,2]

[1]Department of CSE, BUET, Dhaka 1000, Bangladesh
[2]Corresponding author

October 8, 2024

**Abstract**

Lysine succinylation is a post-translational modification that can severely influence various cellular activities including cellular homeostasis and metabolic pathways and has been linked to several diseases by recent research. Despite its emerging importance, current computational methods are limited in performance for predicting succinylation sites. In this study, we propose a novel resnet based architecture, **ResLysEmbed**, for lysine succinylation site prediction that uses word embedding along with local embeddings from a protein language model. We also conducted a comparative analysis between different protein language models to determine the most effective one for the task. Additionally, we compared several deep learning architecture to find the most suitable architecture for processing word embedding features. From the above analysis we developed 3 hybrid architectures **ConvLysEmbed**, **InceptLysEmbed** and **ResLysEmbed**. Finally, our final model, **ResLysEmbed**, outperforms all existing methods with performance scores of 0.80, 0.39, 0.40 for accuracy, MCC and f1-score, respectively on independent test data.

## 1 Introduction

Post-translational modifications (PTMs) play a very crucial role to regulating localization, activities and interaction with other cellular molecules. Among more than 300 kinds of PTMs, lysine succinylation is relatively new one. Succinylation was first reported as a PTM by Zhang et al. [1]. Because succinyl group has a larger mass(100 Da) and is negatively charged, it causes a significant mass change and shift in charge(from +1 to -1) at the modified lysine residue. These changes heavily affect gene expression, cellular homeostasis and metabolic pathways [2].

Recent research on lysine succinylation has linked it to several pathological conditions, including cancer, metabolic disorders and infectious diseases. For example, studies have associated succinylation with metabolic disorders and heart disease [3], hereditary mitochondrial diseases [4], Alzheimer's disease [5].Recently Wang et al. [6] published a study that suggests that succinylation also plays a vital role in central nervous system diseases, including stroke, brain tumors, and Alzheimer's disease. Furthermore, lysine succinylation has been linked to SARS-CoV-2 infection, where succinylated host proteins play a vital role in the virus's interaction with its host [7].

Given the emerging importance of succinylation in health and disease, there has been a lot of experimental approaches for its detection [8], [9], [10]. However, because the experimental methods such as high-resolution liquid chromatography-tandem mass spectrometry (LC-MS/MS) are cumbersome, expensive, and time-consuming various computational methods have been developed for lysine succinylation detection.

The first computational method for detecting succinylated lysine residue prediction was iSuc-PseAAC [11], proposed by Xu et al. the predictor was based on SVM. Later on many other machine learning based predictors were proposed such as iSuc-PseAAC [11], SuccFind [12], iSuc-PseOpt [13], pSuc-Lys [14]. However, all of these studies used a relatively smaller dataset and did not take into account the proper distribution of lysine succinylataion. Later on Hasan et al. proposed a new method SuccinSite [15] along with a new dataset that has been used in a lost of studies later on, including this one. In the upcoming years Hasan et al. proposed 2 new methods namely SuccinSite2.0 [16] and GPSuc [17]. Both of these were generic and species-specific succinylation site predictors being among the very few who tried species specific prediction in this field.

Consequently, deep learning based methods started to emerge such as CNN-SuccSite [18], DeepSuccinylSite [19], pSuc-EDBAM [20], LMSuccSite [21]. CNN-SuccSite used convolutional neural network (CNN)-based architecture along with various feature encoding techniques as input for the task. DeepSuccinylSite [19] used the idea of word embedding for the first time and with various experiments determined the optimal window size. Later on LMSuccSite [21] used the idea of word embedding from DeepSuccinylSite [19] along with language model embedding for the target site to build an ensemble architecture which has been quite successful. During the same period Jia et al. proposed an ensemble dense blocks based method with attention module named pSuc-EDBAM [20] and Ahmed et al. proposed several CNN +

Biderectional LSTM ensemble architectures [22] based on different biophysico properties as input. Both of these in spite of attaining impressive performance could not outperform LMSuccSite [21].

Finally while the development of this study two new works came out namely PTM-CMGMS [23] and PTMGPT2 [24]. These are generic PTM detecting methods where a fixed architecture is trained with different PTM data to create different PTM prediction models. In PTM-CMGMS [23] structural information is utilized using various feature encoding techniques whereas in PTMGPT2 [24] a pre-trained GPT2 based model PROTGPT2 [25] was fine tuned for each PTM detection task. However, despite of being excellent approaches both of these methods require quite a lot computational resources with marginal improvement over existing predictors in terms of succinylation prediction. Moreover, upon closer inspection on PTMGPT2 [24] for comparison purposes we found multiple discrepancies on the succinylation dataset, which is elaborately discussed in Section *A Discussion on PTMGPT2*.

In our study, inspired by the success of LMSuccSite [21] we also decided on using protein language model embeddings as features. However, we have done a comparative analysis on three PLMs namely ProtT5 [26], ESM-650M [27] and ESM-3B [27] to determine which performs the best for this task. Additionally, We have done a comparative analysis with various architectures to determine which performs best for processing word embedding feature. Based on these analysis we have also established that the Conv2D architecture proposed in LMSuccSite [21] performs sub-optimally for capturing sequential information from word embeddings. Finally we proposed 3 hybrid architectures referred to as **ConvLysEmbed**, **InceptLysEmbed** and **ResLysEmbed** . Our final model **ResLysEmbed** outperform every existing methods including the recent generic predictors with an improvement varying around 2% to 15% on independent test data. Moreover, this new architecture has fewer parameters than most existing methods which makes it a computationally efficient and easy to use tool for predicting succinylation cites.

**Key Contributions :**

- Conducted a comparative analysis of different protein language model embeddings for lysine succinylation site prediction.
- Conducted a comparative analysis on different architectures for processing word embeddings.
- Proposed and evaluated three hybrid architectures, namely ConvLysEmbed, InceptLysEmbed, and ResLysEmbed.
- Proposed a new Resnet+MLP based hybrid architecture (ResLysEmbed) that outperforms every existing methods for lysine succinylation site prediction.
- Built a relatively simpler and computationally efficient method for lysine succinylation site prediction.

# 2 Methodology

## 2.1 Dataset

For dataset, we have taken the one used for developing LMSuccSite [21]. This dataset was originally provided by Hasan et al. [15] which was obtained from the UniProtKB/Swiss-Prot [28] database and the NCBI protein sequence database and contains experimentally verified succinylation sites.

The sequences were first processed using CD-HIT [29] with a 30% sequence similarity cut-off in order to create the dataset, ensuring that no sequence shared more than 30% similarity with any other sequence in the dataset. As a result, 5009 succinylation sites from total 2322 protein sequences were added to the dataset.

The dataset was then randomly split into training and testing sets. There were 2192 protein sequences in the training set, which included 4755 succinylation sites, and 124 protein sequences with a total of 253 succinylation sites in the test set. However since 5 succinylation sites were around the N- or C-termini, they were also excluded from the dataset as it was not possible to extract a 33 length window from those sites.

This dataset was also used in developing DeepSuccinylSite [19], pSuc-EDBAM [20] , PTM-CMGMS [23] (for Succinylation) and many others on the field. The dataset along with the language model embeddings and the pssm of all the corresponding proteins are available at **Souvik: Google drive link!**

| Dataset type | Positive (succinylated) | Negative (non-succinylated) |
|---|---|---|
| **Training data** | 4750 | 4750 |
| **Benchmark independent test data** | 254 | 2977 |

Table 1: Overview of the dataset used for training and testing.

## 2.2 Performance metrics

For evaluation, we used several widely used metrics for binary classification (e.g. Accuracy, MCC, AUROC, AUPRC, Precision, Recall, Specificity, F1-score). All of these metrics are based on the standard confusion matrix of binary classification. The confusion matix consists of four components :

- **True Positives(TP):** represents the number of predicted succinylated sites that were originally succinylated.
- **True Negatives(TN):** represents the number of sites predicted as non-succinylated sites that were originally non-succinylated.
- **False Positives(FP):** represents the number of originally non-succinylated sites falsely predicted as succinylated sites.
- **False Negatives(FN):** represents the number of originally succinylated sites falsely predicted as non-succinylated sites.

### 2.2.1 Small description and formula for each performance metrics

- **Accuracy:** Represents the proportion of data correctly predicted( both positive and negative class) out of all data-points.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** Represents the proportion of positive predictions that are correctly predicted.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** Represents the proportion of correctly predicted positive data out of all positive data.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-score:** F1-score is the harmonic mean of precision and recall. It represents how much the model can balance between precision and recall.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Specificity:** Represents the proportion of correctly predicted negative data out of all negative data.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- **MCC:** MCC represents the overall quality of binary classification by taking into account all 4 components of the confusion metrics.

$$\text{MCC} = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

- **AUROC:** Receiver Operating Curve(ROC) is a curve that plots the true positive rate and false positive rate of the model shifting the threshold of classifications from 0 to 1. The Area Under this curve provides a good representation of how the model distinguishes between positive and negative classes with area 1 indicating a perfect classification.
- **AUPRC:** Similar to previous one the Precision Recall Curve plots the precision and recall of the model at different threshold. The Area under Precision Recall Curve focuses on the trade off between precision and recall.

## 2.3 Features

In our study, we explore multiple features inspired by existing works to improve succinylated lysine residue prediction. The authors of LMSuccSite [21] utilized a combination of word embedding and language model embeddings for residue prediction.

word embedding is essentially a window of amino acids centered around the target site passed through Keras's embedding layer. The idea of encoding amino acid window in this way was first used in DeepSuccinylSite [19] where they performed experiments with different window sizes(ranging from 11 to 41). The optimal window size from there experiments was 33 (16-K-16) which was letter verified by the authors of LMSuccSite [21] by similar experiments.Each amino acid in the sequence window is transformed into a dense, continuous vector of fixed dimension by Keras's embedding layer. In our case, every amino acid is mapped into a 21-dimensional vector, which represents its location in a learnt embedding space, consisting of 20 canonical amino acids and 1 missing amino acid. Because the embedding layer is trained as a component of the larger model, it is able to develop optimal representations of the amino acids based on

their value in predicting succinylated lysine residues. Throughout training the embedding layer tries to encode the specific characteristics of amino acids and their context within the protein sequence. This improves the model's capacity to identify local sequence patterns associated with succinylation.

Additionally, we considered embeddings from three different language models: ProtT5, ESM-650M, and ESM-3B.

ProtT5 [26] ,originally called ProtT5-XL-UniRef50, is pre-trained using span-generation and teacher-forcing methods on the UniRef50 [30] dataset, which has over 45 million protein sequences, in a self- manner. It has a 24-layer encoder-decoder design with about 2.8 billion parameters, based on Google's T5-3B model [31]. The model is able to learn contextual links within protein sequences with the aid of it's span-generation process, which yields a 1024-dimensional embedding that incorporates both local and global sequence properties.

The ESM-650M model is a more recent PLM(protein language model) based on a transformer architecture created especially for protein sequences. It comes from the Evolutionary Scale Modeling (ESM) [27] family, which was also pre-trained in a self- manner. By learning to predict masked amino acids inside the sequence, it captures structural and evolutionary insights and generates a 1280-dimensional embedding.

Similarly, a larger model from the ESM family is the ESM-3B model, with almost 3 billion parameters. The same masked language modeling method as ESM-650M is used to train this model, which has a 2560-dimensional embedding , allowing it to capture far more intricate and distant interactions in protein sequences.

Furthermore, we included PSSM (Position-Specific Scoring Matrix) data as another feature. We hypothesize that the PSSM could complement the word embedding by emphasizing conserved regions because, PSSM profiles, obtained via sequence alignment against protein family databases, provide evolutionary conservation information by highlighting residues that are crucial for protein function or stability.

## 2.4 Feature selection

We incorporated different feature selection methods in order to select the best group from the above mentioned ones. In order to compare the feature importance of each module we first reduced the dimension of each group to just five feature using principal component analysis(PCA) [32]. In case of word embedding, because we cannot obtain the final outcome of the embedding layer without training it with a whole model, we simply used the one hot encoding of the amino acid window. Then We used the minimum redundancy maximum relevance(mRMR) [33] technique on the training to understand which feature has the most impact in predicting succinylated lysine residues.

We also incorporated feature importance ranking from XgBoost. In this case we used PCA to reduce each group of features dimension to 10 and ranked top 30 features based on feature importance.

In both of the above experiments most features were selected from the word embedding and ProtT5 embedding. Therefore, similar to LMSuccSite [21] we decided on using these as the final features for our model.

## 2.5 Model selection

For our model we decided to use separate architecture for the word embedding and protT5 embedding as they represent very different types of contextual information of the target site.

### 2.5.1 Word embedding model selection

Because the word embedding is essentially sequential data, we considered models that are designed to capture sequential relationships more effectively, such as Recurrent Neural Networks (RNN), Bidirectional Long Short-Term Memory networks (BiLSTM), Feed forward Neural Networks (FNN), and Convolutional Neural Networks with 1D convolutions (Conv1D). However, in development of LMSuccSIte [21], Neural Networks with 2D convolutions (Conv2D) were used in their word embedding module. This does not seem to be a good choice because the spatial dimension (like a 2D grid in images) is not as relevant as the sequential dimension (1D) in a amino acid sequence representation. Each amino acids position in a protein sequence has a specific meaning, and applying 2D convolutions could mix relationships in a way that might not respect the sequential nature of the data.

However we still included Conv2D in our model comparison process along with Conv1D and 2 other variations of Conv1D , inception and residual connection. We created an imbalanced validation dataset by splitting the training data in 1:9 ratio as validation data and training data. Then we randomly under-sampled the positive data points from the validation dataset to a a fraction of 0.1 to create an imbalanced validation data. Finally we trained all 7 models using the word embedding and tested there performances on the validation set.

From the results , we choose the Conv1D architecture and its two variations as all of them significantly outperform other deep learning architecture. with the inception and residual variation performing better than then the basic Conv1D model in most cases.

#### 2.5.2 ProtT5 embedding model selection

For the protT5 embedding part we compared the results of different ML and DL architectures. Based on the performance results , MLP (Multi-Layer Perceptron) was chosen for the ProtT5 embedding data due to its superior performance across multiple key metrics.

### 2.6 Short description of the models

Based on our preliminary analysis , we decided to further experiment with three Different ensemble architectures for succinylated lysine residue prediction : (1) a 1D Convolutional Network combined with MLP, (2) an Inception module combined with MLP, and (3) a Residual Network combined with MLP. These models will be referred to as **ConvLysEmbed**, **InceptLysEmbed**, and **ResLysEmbed**, respectively. Each of the 3 models have two different branches for separately processing word embedding features and language model embedding features.

#### 2.6.1 1D Convolutional Network (ConvLysEmbed)

The ConvLysEmbed model incorporates a simple 1D convolution branch that starts of with the keras's embedding layer which is essentially the key part of word embedding generation and maps the 33 length amino acid sequence into a 21-dimensional space. The embedding layer is followed by two 1D convolutional layers with 32 and 64 filters, respectively. In order to reduce the spatial dimension, each Conv1D layer is followed by a max-pooling operation. Finally after flattening the output from the convolutional layers, the branch ends with a fully connected dense layer of 32 units followed by a dropout of 30% to prevent overfitting.

#### 2.6.2 Inception Module (InceptLysEmbed)

The InceptLysEmbed model utilizes an inception module for capturing patterns at different scales from the output of the embedding layer. The inception module consists of multiple 1D convolutional layers with different kernel sizes (e.g., 1, 3, 5, 7, 9, and 11). The output from the embedding layer is simultaneously passed through all these layers . Additionally, a max-pooling branch is combined with a 1x1 convolution to further capture important local features. The outputs from all these branches are then concatenated and processed through max-pooling and flattening operations. Finally the branch ends with a fully connected dense layer of 32 units followed by a dropout of 30% to prevent overfitting.

#### 2.6.3 Residual Network (ResLysEmbed)

The ResLysEmbed model employs a residual network (ResNet) architecture which helps to preserve important information throughout the convolutional process. Each residual block consists of two Conv1D layers (each with kernel size 3) and a skip connection, which adds the input of the block back to the output. The output of the embedding layer is passed through two residual blocks with different number of filters (e.g. 32 and 64) and each residual block is followed by a max-pooling operation. Finally the branch end similarly to the above mentioned ones with a flattening operation followed by a dense layer and dropout.

#### 2.6.4 MLP Branch for ProtT5 Embeddings

The MLP branch handles the ProtT5 embedding of the target lysine site and its architecture is same across all three models. It simply passes the 1024-dimensional embedding through a fully connected layer with 32 units and ReLU activation, followed by a dropout layer. This simple architecture of the MLP branch is to ensure that after he ProtT5 embeddings are effectively processed without overshadowing the contributions of the word embedding branch.

#### 2.6.5 Model Output

The outputs from the word embedding branch (either Conv1D, Inception, or ResNet) and the MLP branch are concatenated and passed through an additional dense layer of 32 units to make the final prediction.

#### 2.6.6 Training and Comparing the results of the models

For training the models we decided to train the whole model with both branches altogether. This approach was taken to ensure that the embedding layer is trained with the final model so that it can create an optimal representations of the amino acids for the final model. On the contrary , training the branches separately and then freezing the layers for the final training might result on a sub optimal feature encoding in the word embedding branch.

We tested the aforementioned three models using 10-fold cross validation and also compared them using ROC and PR curve. The results demonstrate that the ResLysEmbed model is the most effective for succinylated lysine residue

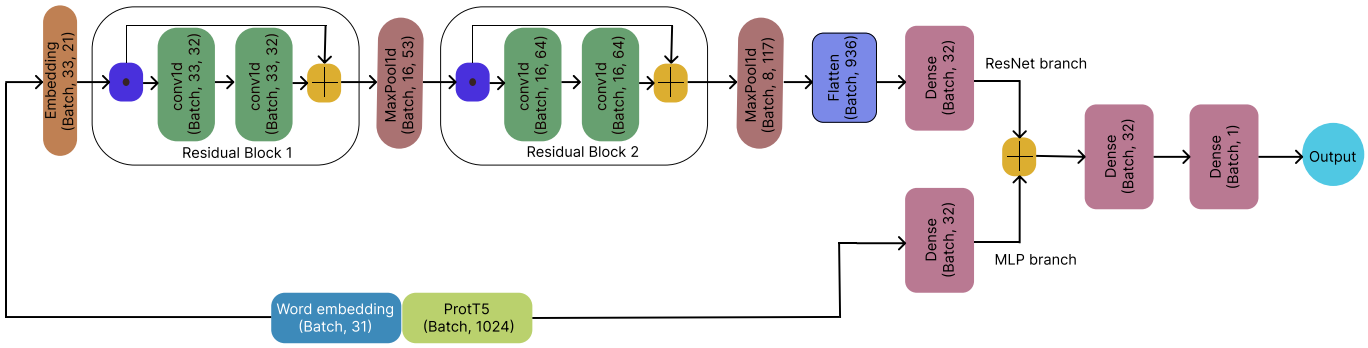
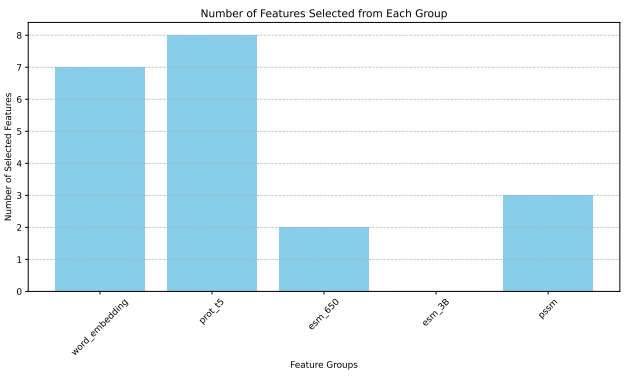



Figure 1: A visual representation of the ResLysEmbed Model architecture

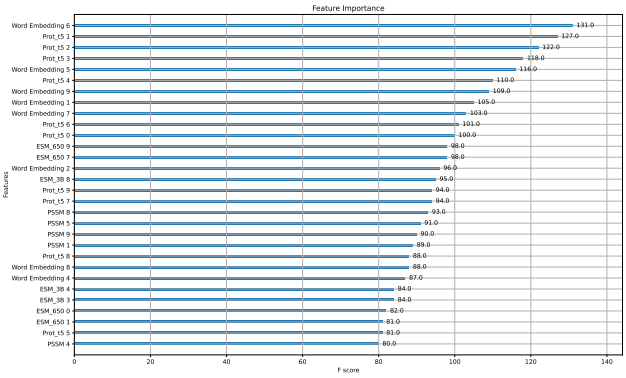
prediction, outperforming the ConvLysEmbed and InceptLysEmbed models across all evaluation metrics. Subsequently, we choose ResLysEmbed as our final model for the succinylated lysine residue prediction task.

# 3 Results

## 3.1 Feature Selection Results




(a) Feature selection using mRMR




(b) Top 30 featured selected based on feature importance

Figure 2: Feature Selection using mRMR and feature importance

The results from figure 2a depict how many features were selected from each group after using mRMR(Minimum Redundancy Maximum Relevance). It is clear from the figure that most of the features were selected from word embedding and protT5 embedding. Similarly, figure 2b plots the feature importance of top 30 features while training an XGBoost model. Here we can also see that among the top 30 features 10 were from protT5 and 8 from word embedding. From these results we can easily conclude that protT5 embedding performs much better in terms of succinylated lysine residue prediction than ESM-650M or ESM-3B. It should also be noted that the word embedding despite not being as sophisticated as a language model embedding still holds substantial relevance for this task, highlighting its importance alongside more advanced embeddings.

## 3.2 Model Selection Results

### 3.2.1 Performance comparison of selected deep learning models on Word embedding

In order to select the best model for processing word embedding data 7 deep learning architectures were used, namely , RNN, BiLSTM, FNN, Conv1D, Conv2D, Inception(Conv1D based) and Residual connection(Conv1D based).


Figure 3: Performance Comparison of different Deep Learning architecture on word embedding data

| Model | Accuracy | MCC | AUROC | AUPRC | Precision | Recall | Specificity | F1 Score |
|---|---|---|---|---|---|---|---|---|
| RNN | 0.651 | 0.0776 | 0.5612 | 0.1851 | 0.1325 | 0.449 | 0.6735 | 0.2047 |
| BiLSTM | 0.6714 | 0.3216 | 0.763 | 0.2823 | 0.2172 | **0.8776** | 0.6485 | 0.3482 |
| FNN | 0.7102 | 0.2267 | 0.6757 | 0.3765 | 0.2 | 0.6327 | 0.7188 | 0.3039 |
| Conv1D | 0.7959 | 0.3613 | 0.7596 | 0.3745 | 0.2893 | 0.7143 | 0.805 | 0.4118 |
| Conv2D | 0.7469 | 0.2402 | 0.678 | 0.3427 | 0.218 | 0.5918 | 0.7642 | 0.3187 |
| Inception | 0.7959 | **0.3823** | **0.7778** | 0.3344 | **0.296** | 0.7551 | 0.8005 | **0.4253** |
| ResNet | **0.798** | 0.3638 | 0.7608 | **0.3942** | 0.2917 | 0.7143 | **0.8073** | 0.4142 |

Table 2: Performance Comparison of different Deep Learning architecture on word embedding data

From the results of 2 we can see that Conv1D architecture and its two variations (inception and resnet) significantly outperform other deep learning architectures. Additionally, RNN and BiLSTM while having advantages in handling sequence data, they underperform compared to Conv1D in this scenario. It should also be noted that althoug the inception and resnet model perform slightly better in different metrics than the basic Conv1D , their improvement is marginal. This is why we decided to further experiment with all 3 Conv1D based model for the final model.

Interestingly, the results of 2 also show that Conv2D does not offer any advantage over Conv1D in this scenario. This confirms our earlier speculation that applying 2D convolutions could mix relationships between the spatial and temporal dimension and that could in turn result in an undesired representation of the sequential nature of the data.

### 3.2.2 Performance comparison of selected ML & DL models on ProtT5 embedding

For selecting the best model for ProtT5 embedding we tried different ML and DL models namely, Random Forest Classifier(RF), Support Vector Machine(SVM), Extreme Gradient Boosting(XGBoost), Multilayer Perceptron (MLP) and 1D Convolutional Neural Network(CNN).

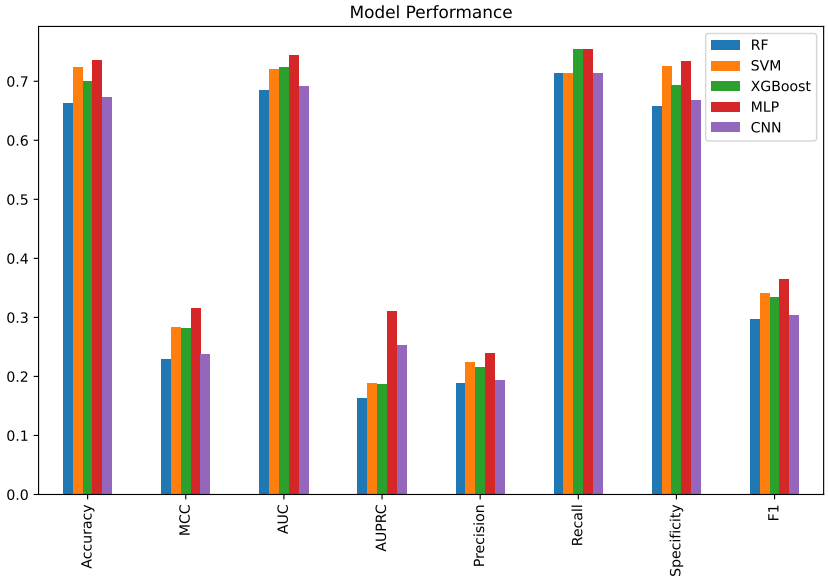



Figure 4: Performance Comparison of different models on protT5 embedding data

| Model | Accuracy | MCC | AUROC | AUPRC | Precision | Recall | Specificity | F1 |
|---|---|---|---|---|---|---|---|---|
| **RF** | 0.6633 | 0.2299 | 0.6859 | 0.1630 | 0.1882 | 0.7143 | 0.6576 | 0.2979 |
| **SVM** | 0.7245 | 0.2833 | 0.7200 | 0.1888 | 0.2244 | **0.7143** | **0.7256** | 0.3415 |
| **XGBoost** | 0.7000 | 0.2822 | 0.7245 | 0.1869 | 0.2151 | 0.7551 | 0.6939 | 0.3348 |
| **MLP** | **0.7367** | **0.3165** | **0.7449** | **0.3103** | **0.2403** | 0.7551 | 0.7347 | **0.3645** |
| **CNN** | 0.6735 | 0.2382 | 0.6916 | 0.2538 | 0.1934 | 0.7143 | 0.6689 | 0.3043 |

Table 3: Model performance metrics comparison on ProtT5 embedding data

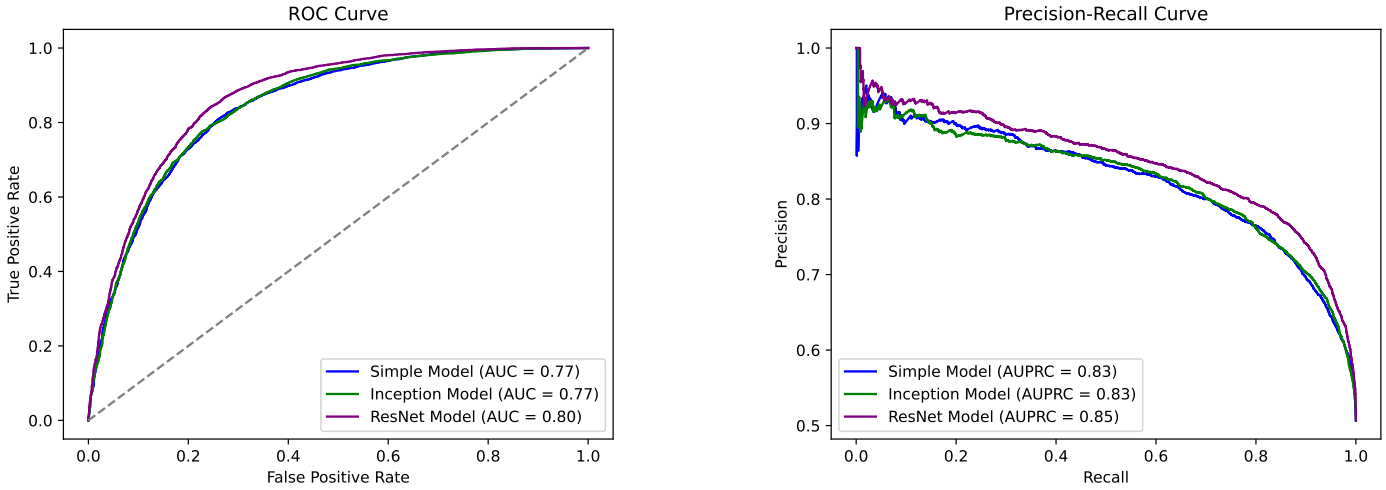
From the results of 3 we can see that MLP achieved the highest performance across most metrics, including Accuracy (0.7367), AUROC (0.7449), AUPRC (0.3103), Precision (0.3103), Recall (0.7551), and F1 score (0.3746). SVM also performed well, especially in terms of MCC (0.2833) and Specificity (0.7256). However, it is also evident that MLP and CNN, tend to outperform traditional machine learning models (RF, SVM, XGBoost) in most of the performance metrics. This suggests that Deep Learning are more effective for task of capturing proper information from language model embeddings.

### 3.2.3 Performance comparison of ConvLysEmbed, InceptLysEmbed and ResLysEmbed

For final model selection we proposed 3 models : ConvLysEmbed, InceptLysEmbed and ResLysEmbed. We compared their performance using 10-fold cross validation on the training set and also their ROC an PR curve.

| Model | Accuracy | MCC | AUROC | AUPRC | Precision | Recall | F1 |
|---|---|---|---|---|---|---|---|
| **ConvLysEmbed** | 0.7736 ± 0.0100 | 0.5476 ± 0.0200 | 0.7732 ± 0.0100 | 0.8281 ± 0.0149 | 0.7653 ± 0.0163 | 0.7983 ± 0.0249 | 0.7811 ± 0.0131 |
| **InceptLysEmbed** | 0.7742 ± 0.0110 | 0.5487 ± 0.0225 | 0.7740 ± 0.0109 | 0.8329 ± 0.0151 | 0.7689 ± 0.0085 | 0.7930 ± 0.0270 | 0.7805 ± 0.0129 |
| **ResLysEmbed** | **0.7965 ± 0.0130** | **0.5941 ± 0.0273** | **0.7960 ± 0.0133** | **0.8521 ± 0.0165** | **0.7887 ± 0.0165** | **0.8189 ± 0.0383** | **0.8028 ± 0.0151** |

Table 4: Performance comparison of ConvLysEmbed, InceptLysEmbed and ResLysEmbed




(a) ROC curve

(b) PR curve

Figure 5: ROC (Receiver Operating Characteristic) curve and PR (Precision Recall) curves with AUC (Area Under Curve) for ConvLysEmbed, InceptLysEmbed and ResLysEmbed

From table 4 we can see that ResLysEmbed model outperforms the other two models across all metrics. On the other hand, InceptLysEmbed performs similarly to ConvLysEmbed, with only slight improvements in some metrics. This suggests that while the inception architecture adds some benefits, they are marginal compared to the simple ConvLysEmbed model.

Figure 5 visualizes these models performance differences using the ROC and PR curves. In Figure 5a, the ROC curves show that ResLysEmbed consistently has a higher true positive rate at different false positive rates, resulting in its higher AUC score of 0.80 compared to the 0.77 of both ConvLysEmbed and InceptLysEmbed. Similarly, the PR curves in Figure 5b shows that ResLysEmbed maintains highest AUPRC score of 0.85.

These results conclude that the ResLysEmbed model is the most effective at predicting succinylated lysine residues. While ConvLysEmbed and InceptLysEmbed are competent models, their performance is slightly inferior to ResLysEmbed across multiple metrics. The outcome of this comparison suggests that the residual branch pairs up well with the MLP branch for succinylated lysine residue prediction.

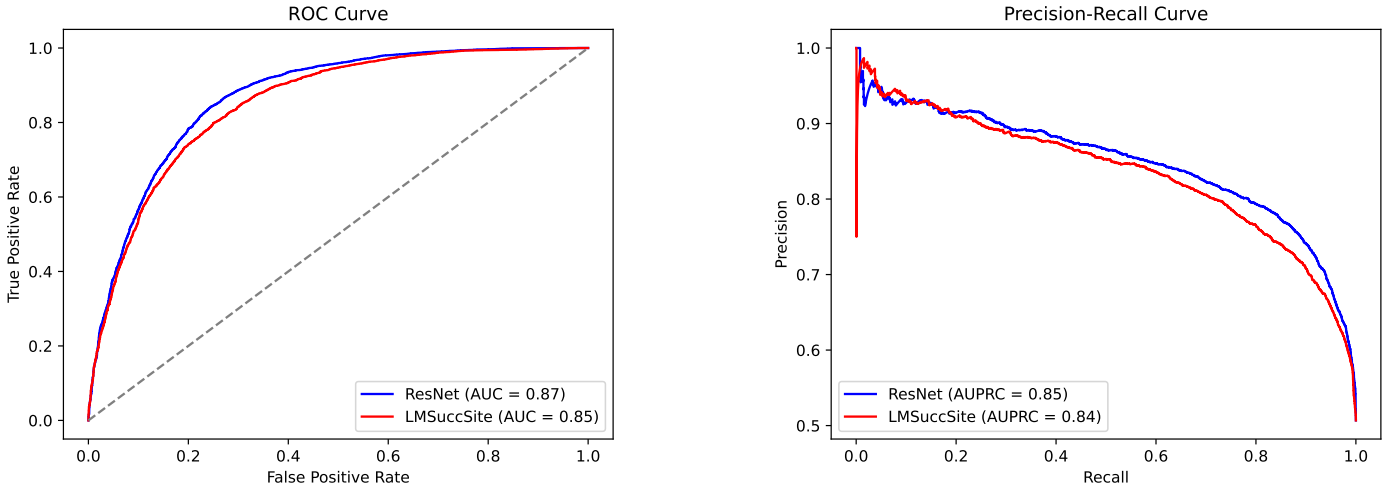
### 3.2.4 Comparison of ResLysEmbed with other predictors

In this section, we compare the performance of our proposed ResLysEmbed model with other state-of-the-art models for succinylated lysine residue prediction. For comparison we had chosen the (CBL+BLC+CBL_BLC)-E model from Ahmed et al. [22] as their best performing model, pSuc-EDBAM [20] , the results for succinylation from PTM-CMGMS [23] and LMSuccSite [21]. We were unable to load the saved model from the LMSuccSite repository. We tried contacting the authors via email regarding this issue but did not get any response from them. Therefore, we re-implemented their work based on the provided methodology and hyper-parameters.
Table 5 represents the key performance metrics including accuracy, MCC, AUC, AUPRC, precision, recall, and F1 score of different models along with ResLysEmbed on the independent test set from Table 1. It should be noted that the same training and independent test set from Table 1 were used to develop all these models.

| Model | Accuracy | MCC | AUROC | AUPRC | Precision | Recall | Specificity | F1-Score |
|---|---|---|---|---|---|---|---|---|
| LMSuccSite(Reproduced) | 0.7626 | 0.3345 | 0.7808 | 0.3155 | 0.2209 | 0.8024 | 0.7591 | 0.3464 |
| pSuc-EDBAM | 0.7559 | 0.2927 | - | - | - | 0.7559 | 0.7413 | - |
| PTM-CMGMS | - | 0.3072 | 0.8306 | 0.3024 | - | - | - | - |
| (CBL+BLC+CBL_BLC)-E | 0.696 | 0.271 | - | - | - | 0.791 | 0.787 | - |
| ResLysEmbed | **0.8053** | **0.3893** | **0.8733** | **0.3482** | **0.2624** | **0.8182** | **0.8042** | **0.3973** |

Table 5: Performance comparison of ResLysEmbed with other predictors on independent test set.

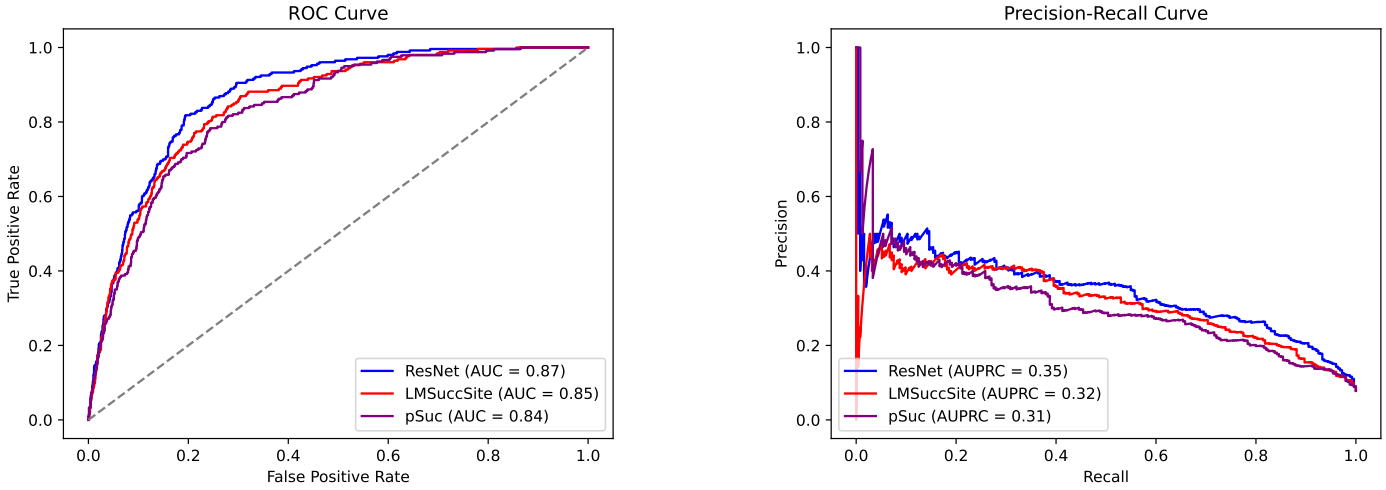
From the results it is evident that the ResLysEmbed model outperforms existing methods in nearly all metrics. It achieves almost 2% to 15% improvement in almost every metric compared to the next best predictor LMSuccSite. The high improvement in F1-score indicates that the model can strike a better balance between precision and recall, leading to improved overall performance.




(a) ROC curve

(b) PR curve

Figure 6: ROC (Receiver Operating Characteristic) curve and PR (Precision Recall) curves with AUC (Area Under Curve) for ResLysEmbed and LMSuccSite on 10-fold CV of Training Set




(a) ROC curve

(b) PR curve

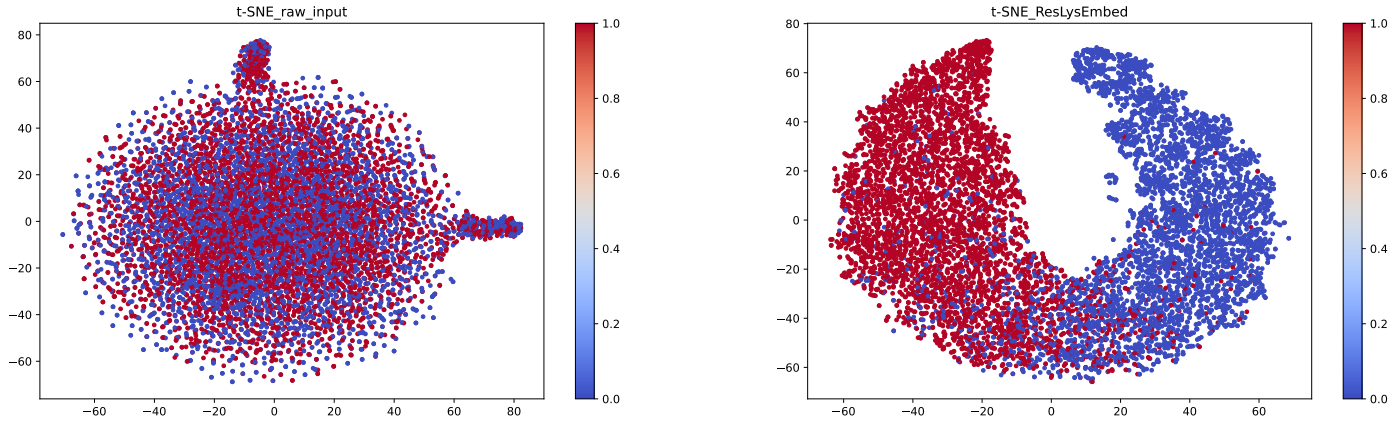
Figure 7: ROC (Receiver Operating Characteristic) curve and PR (Precision Recall) curves with AUC (Area Under Curve) for ResLysEmbed, LMSuccSite and pSuc-EDBAM on independent test set

We also compared the performance of ResLysEmbed against LMSuccSite [21] using the ROC and PR curve for the tenfold cross-validation on training data and against both LMSuccSite [21] and pSuc-EBDAM [20] on independent test data. Figure 6 and figure 7 illustrates the ROC and PR curves for the 10 fold cross validation result and independent test result respectively. As reflected in the higher AUC scores in both the ROC and PR curves, ResLysEmbed model demonstrates better performance in classification compared to both LMSuccSite and pSuc-EDBAM.

### 3.2.5 Further analysis using t-SNE plots

To get a more visual look at the model's learning capability, we visualized the high-dimensional embeddings using t-distributed stochastic neighbor embedding (t-SNE) [34] , as shown in Figure 8.

The figure on the left (Figure 8a) shows the raw input features(from the training data) before training, where there is no clear separation between the succinylated and non-succinylated lysine residue data. After training the ResLysEmbed model, we plotted the features from the final hidden layer which is shown in (Figure 8b). Interestingly, it shows significantly improved separation, with almost distinct clusters forming for positive and negative classes which confirms that the model was able to learn distinctive features from the input data.




(a) Before training

(b) After training the ResLysEmbedModel

Figure 8: t-SNE to visualize the high-dimensional embedding learned by the Model

# 4 Adding PSSM to the mix : A protein specific analysis

# 5 A Discussion on PTMGPT2

We had tried following the same procedure as PTMGPT2 for extracting the training sequences from dbPTM. We retrieved the training proteins from dbPTM website and downloaded the FASTAs from uniprot. Then we used CD-HIT with a 30% sequence similarity cutoff on the training set. There we got 3012 proteins. We got the test proteins from mapping dbPTM Benchmark dataset with PTMGPT2's test set. We got 1049 proteins. It contained redundant proteins. When we ran CD-HIT with a 30% sequence identity limit, we received 747 proteins. After checking, we saw that there were duplicate and redundant proteins between our newly created nonredundant training and test set.

To find out how much duplicate and redundant proteins were present between the training and test set, we used BLAST-CLUST. After running BLAST-CLUST [35] on the newly created training set (with 3012 proteins), we got 2979 proteins. For the test set (747 proteins), we got 744 proteins. Then, we ran BLAST-CLUST between the training set and test set. We always used a 30% sequence identity limit for BLAST-CLUST. Finally, we saw that there were 377 duplicate proteins and 264 redundant proteins among the 747 proteins inside the test set.

# 6 Discussion and Conclusion

In this study, we have proposed a novel method, **ResLysEmbed**, which uses a combination of resnet-based architecture combined with MLP for lysine succinylation site prediction. **ResLysEmbed** outperforms the existing methods with promising results and also has fewer parameters than other methods. We believe, **ResLysEmbed** can serve as a reliable and resource efficient method for researchers studying succinylation and related PTMs.

The improved performance of this method proves that resnet-based architectures can be very effective when processing features like word embeddings. At the same time, it highlights the importance of using simpler architectures while processing PLM embeddings.

The results of this study highlight some valuable insights into lysine succinylation site prediction. One of the most important aspect is the key role that word embeddings play in this scenario. Additionally, protein language model (PLM) embeddings further improve the performance when combined with word embeddings. Moreover, our analysis proves that training the embedding layer with the final model can produce better results than pre-training and freezing it during the final training.

Finally, we believe, even with the success of this method, there are areas for future exploration. The integration of structural data may provide improved prediction results. Additionally, as more and more protein language models emerge, more comprehensive analysis into the impact of different PLMs in lysine succinylation cite prediction may provide more improvements.

# 7 Declarations

We have to follow the submitting the journal for this

# References


[1] Z. Zhang, M. Tan, Z. Xie, L. Dai, Y. Chen, and Y. Zhao, "Identification of lysine succinylation as a new post-translational modification," *Nature chemical biology*, vol. 7, no. 1, pp. 58–63, 2011.

[2] Y. Yang and G. E. Gibson, "Succinylation links metabolism to protein functions," *Neurochemical research*, vol. 44, no. 10, pp. 2346–2359, 2019.

[3] L. Yang, S. Miao, J. Zhang, P. Wang, G. Liu, and J. Wang, "The growing landscape of succinylation links metabolism and heart disease," *Epigenomics*, vol. 13, no. 4, pp. 319–333, 2021.

[4] P. Gut, S. Matilainen, J. G. Meyer, P. Pällijeff, J. Richard, C. J. Carroll, L. Euro, C. B. Jackson, P. Isohanni, B. A. Minassian, *et al.*, "Sucla2 mutations cause global protein succinylation contributing to the pathomechanism of a hereditary mitochondrial disease," *Nature communications*, vol. 11, no. 1, p. 5927, 2020.

[5] Y. Yang, V. Tapias, D. Acosta, H. Xu, H. Chen, R. Bhawal, E. T. Anderson, E. Ivanova, H. Lin, B. T. Sagdullaev, *et al.*, "Altered succinylation of mitochondrial proteins, app and tau in alzheimer's disease," *Nature communications*, vol. 13, no. 1, p. 159, 2022.

[6] C. Wang, W. Cui, B. Yu, H. Zhou, Z. Cui, P. Guo, T. Yu, and Y. Feng, "Role of succinylation modification in central nervous system diseases," *Ageing Research Reviews*, p. 102242, 2024.

[7] Q. Liu, H. Wang, H. Zhang, L. Sui, L. Li, W. Xu, S. Du, P. Hao, Y. Jiang, J. Chen, *et al.*, "The global succinylation of sars-cov-2–infected host cells reveals drug targets," *Proceedings of the National Academy of Sciences*, vol. 119, no. 30, p. e2123065119, 2022.

[8] H. Yuan, J. Chen, Y. Yang, C. Shen, D. Xu, J. Wang, D. Yan, Y. He, and B. Zheng, "Quantitative succinyl-proteome profiling of chinese hickory (carya cathayensis) during the grafting process," *BMC plant biology*, vol. 19, pp. 1–10, 2019.

[9] H. Zhou, I. Finkemeier, W. Guan, M.-A. Tossounian, B. Wei, D. Young, J. Huang, J. Messens, X. Yang, J. Zhu, *et al.*, "Oxidative stress-triggered interactions between the succinyl-and acetyl-proteomes of rice leaves," *Plant, Cell & Environment*, vol. 41, no. 5, pp. 1139–1153, 2018.

[10] W. Jin and F. Wu, "Proteome-wide identification of lysine succinylation in the proteins of tomato (solanum lycopersicum)," *PloS one*, vol. 11, no. 2, p. e0147586, 2016.

[11] Y. Xu, Y.-X. Ding, J. Ding, Y.-H. Lei, L.-Y. Wu, and N.-Y. Deng, "isuc-pseaac: predicting lysine succinylation in proteins by incorporating peptide position-specific propensity," *Scientific reports*, vol. 5, no. 1, p. 10184, 2015.

[12] H.-D. Xu, S.-P. Shi, P.-P. Wen, and J.-D. Qiu, "Succfind: a novel succinylation sites online prediction tool via enhanced characteristic strategy," *Bioinformatics*, vol. 31, no. 23, pp. 3748–3750, 2015.

[13] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "isuc-pseopt: identifying lysine succinylation sites in proteins by incorporating sequence-coupling effects into pseudo components and optimizing imbalanced training dataset," *Analytical biochemistry*, vol. 497, pp. 48–56, 2016.

[14] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "psuc-lys: predict lysine succinylation sites in proteins with pseaac and ensemble random forest approach," *Journal of theoretical biology*, vol. 394, pp. 223–230, 2016.

[15] M. M. Hasan, S. Yang, Y. Zhou, and M. N. H. Mollah, "Succinsite: a computational tool for the prediction of protein succinylation sites by exploiting the amino acid patterns and properties," *Molecular BioSystems*, vol. 12, no. 3, pp. 786–795, 2016.

[16] M. M. Hasan, M. S. Khatun, M. N. H. Mollah, C. Yong, and D. Guo, "A systematic identification of species-specific protein succinylation sites using joint element features information," *International journal of nanomedicine*, pp. 6303–6315, 2017.

[17] M. M. Hasan and H. Kurata, "Gpsuc: Global prediction of generic and species-specific succinylation sites by aggregating multiple sequence features," *PloS one*, vol. 13, no. 10, p. e0200283, 2018.

[18] K.-Y. Huang, J. B.-K. Hsu, and T.-Y. Lee, "Characterization and identification of lysine succinylation sites based on deep learning method," *Scientific reports*, vol. 9, no. 1, p. 16175, 2019.

[19] N. Thapa, M. Chaudhari, S. McManus, K. Roy, R. H. Newman, H. Saigo, and D. B. Kc, "Deepsuccinylsite: a deep learning based approach for protein succinylation site prediction," *BMC bioinformatics*, vol. 21, pp. 1–10, 2020.

[20] J. Jia, G. Wu, M. Li, and W. Qiu, "psuc-edbam: Predicting lysine succinylation sites in proteins based on ensemble dense blocks and an attention module," *BMC bioinformatics*, vol. 23, no. 1, p. 450, 2022.

[21] S. Pokharel, P. Pratyush, M. Heinzinger, R. H. Newman, and D. B. Kc, "Improving protein succinylation sites prediction using embeddings from protein language model," *Scientific reports*, vol. 12, no. 1, p. 16933, 2022.

[22] S. S. Ahmed, Z. T. Rifat, M. S. Rahman, and M. S. Rahman, "Succinylated lysine residue prediction revisited," *Briefings in Bioinformatics*, vol. 24, no. 1, p. bbac510, 2023.

[23] Z. Li, M. Li, L. Zhu, and W. Zhang, "Improving ptm site prediction by coupling of multi-granularity structure and multi-scale sequence representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 188–196, 2024.

[24] P. Shrestha, J. Kandel, H. Tayara, and K. T. Chong, "Post-translational modification prediction via prompt-based fine-tuning of a gpt-2 model," *Nature Communications*, vol. 15, no. 1, p. 6699, 2024.

[25] N. Ferruz, S. Schmidt, and B. Höcker, "Protgpt2 is a deep unsupervised language model for protein design," *Nature communications*, vol. 13, no. 1, p. 4348, 2022.

[26] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, *et al.*, "Prottrans: Toward understanding the language of life through self-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 10, pp. 7112–7127, 2021.

[27] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, *et al.*, "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, no. 6637, pp. 1123–1130, 2023.

[28] U. Consortium, "Uniprot: a hub for protein information," *Nucleic Acids Res*, vol. 43, no. D1, pp. D204–D212, 2015.

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

[30] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu, "Uniref: comprehensive and non-redundant uniprot reference clusters," *Bioinformatics*, vol. 23, no. 10, pp. 1282–1288, 2007.

[31] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[32] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[33] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[34] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[35] M. Johnson, I. Zaretskaya, Y. Raytselis, Y. Merezhuk, S. McGinnis, and T. L. Madden, "Ncbi blast: a better web interface," *Nucleic acids research*, vol. 36, no. suppl_2, pp. W5–W9, 2008.