Predicting effect of Post-Translational Modification on Protein-Protein Interaction

Ekaterina Geraskina, student number: 2636212, First Reader: Reza Haydarlou, and Second Reader: Anton Feenstra

Vrije Universiteit, Amsterdam, De Boelelaan 1011, Amsterdam, Netherlands
kgeraskina@gmail.com

Abstract. Post-translational modifications (PTMs) can considerably alter the structural or functional attributes of a protein, impacting, among others, protein-protein interactions (PPIs). This, in turn, can result in various pathogenic, degenerative and cancerous diseases. This research is aimed at developing an automated predictor for determining the effect of PTMs on PPIs, based on the PTMint database, a manually curated collection of experimentally verified effects of PTMs on PPIs. The LightGBM model was trained on a combination of ProtBERT embeddings of both interaction partner proteins, the Protein Site Window and a categorical feature representing whether or not PTM happened on the interface. The results show, that for the task of predicting the effect of Phosphorylation on PPIs, our approach performs better or just as well as previous attempts, while considerably decreasing computational costs. However, the performance of the model when applied to predicting the effect of multiple other PTM types on PPIs suggests that further research is needed to develop a way to represent PTMs. All of the code and datasets for this project can be found in this repository.

Keywords: Post-Translational Modification · Protein-Protein Interaction· ProtBERT · LightGBM · PTMint · Bett's Benchmark

1 Introduction

Post-translational modification (PTM) is a process of breakage of peptide bonds (proteolytic cleavage) between amino acids (AAs) and the addition of modifying chemical groups to one or more of AAs within a protein [21] [31]. According to the UniProt Knowledgebase [1], there are at least 700 distinct types of PTMs, with the six most-studied types (namely Phosphorylation, Acetylation, Succinylation, Methylation, Sumoylation, Glycosylation, and Ubiquitination) constituting nearly 90% of the database.

PTMs can considerably alter the structural or functional attributes of a protein, impacting aspects such as the structure, dynamics [26] [38], enzyme function, assembly [32], protein folding [9], and protein-protein interaction (PPI) [27]. PPIs are foundational to all biological processes at both molecular and cellular levels. Within the human body, they play critical roles in metabolism,

signalling, and immunity, among other functions. Disruptions or enhancements in PPIs have been associated with degenerative, pathogenic, or cancer-related diseases [14].

PTMs can influence PPIs through three primary mechanisms. The first mechanism is characterized by a PTM occurring at a protein's interface (interaction region), potentially disrupting or enhancing interaction with its partner, or inducing interaction with a previously non-interacting protein. Additionally, PTMs may indirectly affect PPIs by inducing structural changes or altering the subcellular localization of a protein, thereby, again, disrupting or enhancing existing interactions or fostering new ones.

Due to the significant effect altered PPIs can have on an organism's functionality, substantial scientific attention has been directed towards understanding PPIs. Numerous tools for laboratory investigation of PPIs have been developed, including yeast two-hybrid systems [13], mass spectrometry [30], Fluorescence Polarization (FP), Surface Plasmon Resonance (SPR)[5], and Nuclear Magnetic Resonance (NMR) [42]. Despite their utility, these approaches are labour-intensive, time-consuming, and costly, with each having its own set of limitations.

Moreover, PTM research is also expensive and time-consuming, although recent technological advancements have alleviated some challenges. For instance, Proximity Ligation Assay (PLA) [24] and immunoprecipitation (IP) [12] have emerged as viable techniques for PTM detection. Yet, the combination of mass spectrometry with an IP strategy remains the most effective, despite being a costly and challenging method [22].

Labarotory Investigation of the effects of PTMs on PPIs is particularly challenging and often unfeasible, as cleanly recreating proteins with specific PTMs is difficult. Additionally, some PTM-induced changes in PPIs are not pronounced enough to isolate them from the complex cellular milieu [36].

Needless to say, with the popularization of automated predictors in all spheres of science, numerous methods were created to predict PPIs and PTMs separately. PPI prediction is commonly regarded as a regression problem, binding affinity (strength of the interaction) prediction, and is usually based on the 3D structure of a protein complex[34]. However, only a fraction of complexes for interacting proteins have been solved, moreover, a protein's 3D structure is unstable and can change due to a wide variety of factors [10]. Another approach to PPI prediction is using sequence-based features, such as biophysical properties of AAs, Position-Specific-Scoring Matrices (PSSMs), and so on[6]. PTM prediction also received a lot of scientific attention, and a lot of advancements have been achieved to date[29]. The details of PTM prediction are, however, outside of the scope of this paper.

In line with the trend of utilizing automated predictors and classifiers for solving biological tasks, this research is aimed at developing an automated predictor for classifying the effect of PTMs on PPIs. A reliable predictor could substantially facilitate the understanding of PTM-related diseases on a molecular level, aiding in the identification of drug targets.

To date, several methodologies have been devised to predict the effect of PTMs on PPIs. For instance, Bett's method [3] focuses on the systematic examination of Phosphorylation switches, introducing a S_{switch} score to represent the likelihood of a Phosphosite acting as a switch to either enhance or inhibit a PPI. The score is derived from changes in interaction caused by Phosphorylation, the similarity of the 3D structure of the phosphorylated protein to its 3D template structure, and conservation information at the residue level. They also established a benchmark dataset known as Bett's Benchmark, encompassing known Phosphorylation switches and their effects on PPIs.

Other methods leveraging the 3D structure of proteins to predict the effect of PTM on PPI include FoldX[33] and Hawdock[39]. Initially introduced to calculate the binding affinity of protein complexes, these methods can be adapted to predict the effect of PTM on PPI by evaluating the change in binding affinity score between normal interactions and interactions involving PTM-affected proteins. Yet, to repeat, the employment of 3D structural data with tools like FoldX and HawkDock, while valuable, presents challenges including the requirement for high-resolution structural data, which is often elusive, and significant computational capacity.

Unfortunately, until recently, automated prediction of the effect of PTM on PPI was only possible through 3D structure-based methods. In March 2023, however, a PTMint database was published by Hong et al. [16], containing mined and manually curated experimental evidence of the effect different PTMs have on PPIs. PTMint contains more than 4500 records, describing the effect of six PTMs on PPIs in six different organisms. The effect is represented categorically, with classes 'Enhance', 'Induce' and 'Inhibit'. Not long after, Hong et al. [17] developed a model specifically for predicting the effect of Phosphorylation on PPI using sequence-based features. They used a subset of the PTMint database, describing the effect of Phosphorylation, and for prediction, constructed features such as Averaged Position-Specific-Scoring Matrices (PSSMs) of both proteins (phosphorylated protein and its interaction partner) along with one-hot encoding of the amino acids in the PTM Site window (PSW) (part of the protein sequence around the PTM Site). Their method outperformed Hawdock, FoldX, and Bett's approach on Bett's Benchmark dataset.

However, no tool currently exists for predicting the effect of PTMs on PPIs beyond Phosphorylation. Therefore, the research question of this paper is whether it is possible to construct a reliable model for classifying the effect of PTMs on PPIs (beyond Phosphorylation) using the complete PTMint Database. Additionally, the potential contribution of ProtBERT protein-level embeddings [11] to this task will be examined. Furthermore, it will be explored which dimensionality reduction technique retains the most information while reducing the feature space of ProtBERT embeddings.

The paper is structured as follows: first, the datasets, features, dimensionality techniques, algorithms and the experimental set-up will be described in the Methods section. The result of feature-related experiments and the performance of chosen models for the main classification task (predicting the effect of PTMs

on PPI) will be discussed in the Results section, as well as an in-depth analysis of factors, that might have affected the performance. In the Discussions section, the summary of the whole study will be provided, as well as the conclusions and critical analysis of the strengths and weaknesses of the proposed framework.

2 Methods

2.1 Data preparation and analysis

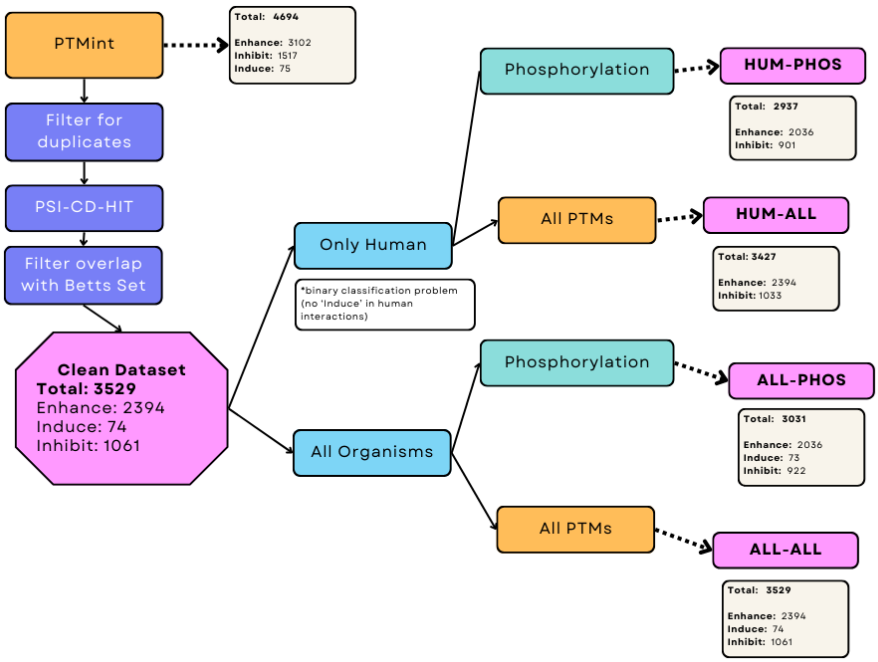


Fig. 1: Flow Chart of data preparation process

The comprehensive, manually curated evidence of PTM regulation of PPIs was obtained from the PTMint database, published in 2023 [16]. This dataset contains experimentally verified records of PTMs affecting PPIs in three distinct ways: Enhance (increase binding affinity), Inhibit (decrease binding affinity), and Induce (give rise to previously impossible interaction), across six organisms: Homo sapiens, Arabidopsis thaliana, Caenorhabditis elegans, Drosophila melanogaster, Saccharomyces cerevisiae, and Schizosaccharomyces pombe. The dataset describes six types of PTMs: Glycosylation, Phosphorylation, Ubiquitination, Methylation, Sumoylation, and Acetylation, with Phosphorylation constituting the majority of the database (86%).

Name	Description
Organism	Organism, in which the interaction was recorded
Gene	Gene of the modified protein (P1)
UniProt	UniProt ID of the modified protein (P1)
PTM	Type of the PTM (6 types in total)
Site-Pos	Position on the protein sequence where PTM happened
Site-AA	The residue at the PTM Site
Int_UniProt	UniProt ID of the interacting partner protein (P2)
Int_gene	Gene of the interacting partner protein (P2)
Effect	The effect which the PTM had on PPI (Enhance, Induce, Inhibit)
Method	Experimental technique used to establish the effect
Disease	Disease associated with the affected interaction
Co-localized	Whether P1 and P2 are co-localized and if so, where
PMID	PubMed ID of the research describing the interaction

Table 1: Original columns of the PTMint database

To prepare the dataset for the prediction task, the initial step is the removal of duplicates. The PTMint dataset comprises the following features: The *Organism*, in which the interaction was recorded, *UniProt ID*s of the affected protein (P1) and its interaction partner (P2), the *Genes* of both proteins, *Co-Localization* information, the *PTM Site* (integer), the AA at the PTM position (*Site-AA*), the *Effect* of the PTM on the interaction (Enhance, Induce, and Inhibit), *PubMed ID* of the source research paper and *Disease*, with which the affected interaction is associated. Given that the final models will be evaluated against Bett's benchmark dataset, which lacks *Co-localization* information, the corresponding column was omitted. Additionally, columns detailing *Genetic* information, *PubMed IDs* of the source research papers and the correlated *Disease* were removed. Given that some interactions were recorded by different research papers and in different cellular locations, the elimination of these columns resulted in numerous duplicates within the dataset. From any group of identical rows, only one was randomly retained.

Organism	UniProt	PTM	Site-Pos	Site-AA	Int_UniProt	Effect
A. thaliana	O23617	Phos	21	S	P48349	Induce
C. elegans	Q10666	Phos	118	S	P41932	Induce
C. elegans	Q967F4	Sumo	2845	K	O44326	Inhibit
D. melanogaster	Q8IPH9	Phos	54	S	P29310	Induce
D. melanogaster	Q9VKM1	Me	10	R	Q9VQ91	Induce

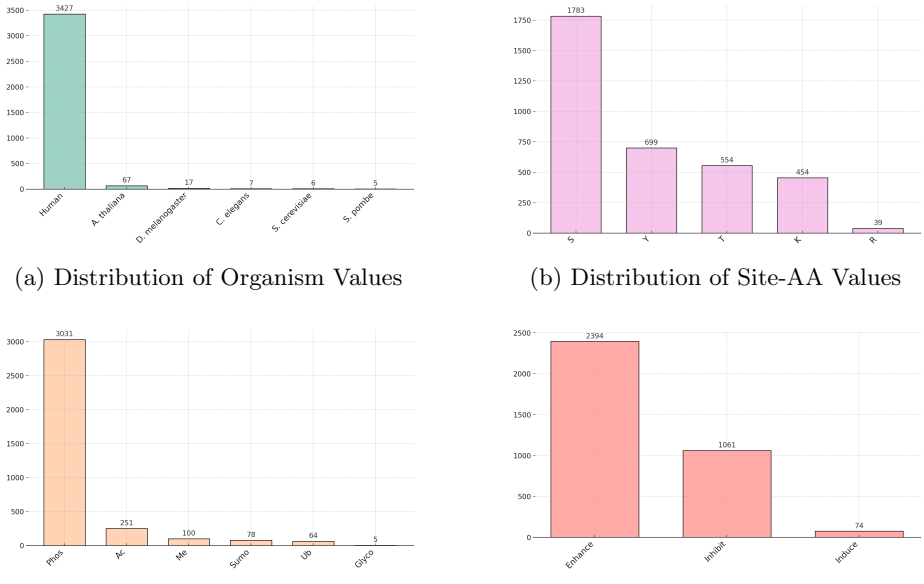
Table 2: Clean PTMint Dataset snippet

To further reduce redundancy, the dataset underwent filtering for sequence similarity among proteins. Since sequence-based features will be utilized, it is important to ensure that interaction partner proteins have differing sequences. Therefore, the dataset was filtered for homologous interacting proteins; if P1

(PTM affected protein) and P2 (P1's interaction partner) exhibited more than 25 % sequence similarity over more than 90% of the sequence length, the row describing that interaction was deleted from the dataset. Filtering was conducted using PSI-CD-HIT [19], following guidelines appropriated from the Hong's et al.[17] approach, mentioned earlier.

The dataset was also scrutinized for any overlap with Bett's benchmark set. Any row in PTMint that is identical to a row in Bett's benchmark set concerning *UniProt ID*s of P1 and P2, *PTM*, *Site-Pos*, *Site-AA*, and the *Effect* columns was removed. Additionally, rows with identical information but differing *Effect* values were discarded too.

Furthermore, certain individual proteins were removed from the dataset due to technical reasons. For example, entry A9UF07 was removed as its sequence could not be accessed using UniProt's Fasta database. All records describing the interaction of phosphorylated Claudin 16 protein (UniProtKB: Q9Y5I7) were eliminated due to a discrepancy between the dataset and the source paper [18]. The dataset indicated that the PTM occurred on the 303rd amino acid of the human Claudin 16 protein (which only has 235 AAs according to the UniProt Database). At the same time, the paper described the effect of Phosphorylation on a mutated murine Claudin 16 protein. Since the dataset did not provide full protein sequences, and they had to be retrieved from the UniProt Database [1] for this research, it was decided to delete records describing Claudin 16 protein (3 in total).



(a) Distribution of Organism Values

(b) Distribution of Site-AA Values

(c) Distribution of PTM Values

(d) Distribution of Effect Values

Fig. 2: Distribution graphs of Categorical columns of clean PTMint

The resulting clean dataset comprises 3529 data points. As illustrated in Figures 2a-2d, the dataset exhibits substantial imbalance for four major categorical features: *Site-AA*, *PTM*, *Organism*, and the *Effect*. While the imbalance in the *Organism* and *Site-AA* columns is arguably non-problematic, the imbalance in the remaining columns will likely pose challenges when considering the prediction of all types of PTM in all organisms, especially compounded by the fact that the 'Induce' effect is only recorded for non-human phosphorylated protein interactions, as stated in the supplementary data of the original publication[16].

The clean dataset was further partitioned into four sub-datasets to address specific research questions. The first dataset, HUM-PHOS, includes only records describing the effect of Phosphorylation on PPI in human proteins, aimed at comparing the efficacy of our method to the one developed by Hong et. al. [17] in prior research. The second dataset, ALL-PHOS, encompasses records describing the effect of Phosphorylation across all organisms listed in PTMint, intended to ascertain whether increasing protein variability aids prediction. The third dataset, HUM-ALL, will be utilized to determine the possibility of predicting the effect of any PTM on PPI, while the final dataset, ALL-ALL, will be used to extend this inquiry to all organisms.

A visualization of the data preparation process is provided in Figure 1.

2.2 Bett's Benchmark Set

To validate the performance of our model, it will be tested on Bett's Benchmark set, since it will allow us to compare its performance to PhosPPI-2, FoldX and Bett's approaches. Bett's benchmark set consists of 238 'Enhance' and 97 'Inhibit' records, describing the effect of Phosphorylation on PPI. It was comprised using UniProt and PhosphoSitePlus databases. Originally, it also contained negative records, describing 'no effect' of Phosphorylation on PPI, however, they were deleted from the dataset for the purposes of this study.

2.3 Feature engineering

For solving the classification task of predicting the effect of PTMs on PPI, sequence-based features will be utilized. As mentioned earlier, Hong et al. used PSSMs of both proteins as well as one-hot encoding of the AAs in the PSW. Moreover, they experimented with utilizing biophysical information, retrieved from the AAIndex database, to represent the PSW further. However, they established, that these features, while informative, did not aid prediction. As the utility of PSSMs was already established, and biophysical properties were proven to be not useful for this task, other approaches to extracting features from the sequence will be considered in this research.

ProtBERT Embeddings. ProtBERT [11], a variation of the BERT (Bidirectional Encoder Representations from Transformers) model, has been adapted for bioinformatics applications with a specific focus on protein sequences. While BERT was originally conceptualized for natural language processing to capture contextual information within texts, ProtBERT extends this approach to protein

sequences, enabling the identification of patterns and contextual relationships among amino acid sequences.

For any given protein sequence input into ProtBERT, the model yields a matrix of size $L \times 1024$, where L is the length of the protein sequence. Each row corresponds to the embedding of an amino acid residue, and each embedding is a vector of 1024 dimensions.

$$\text{Per residue embedding} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{11024} \\ x_{21} & x_{22} & \dots & x_{21024} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{n1024} \end{bmatrix}$$

$$\text{Protein-level embedding} = \begin{bmatrix} \overline{x}_1 & \overline{x}_2 & \dots & \overline{x}_{1024} \end{bmatrix}$$

In the scope of this research, however, protein-level ProtBERT embeddings will be utilized, namely the mean of each column across all AAs. This operation results in a vector of 1024 dimensions, containing the overall contextual information of the entire protein sequence. A ProtBERT embedding was generated for each protein in the dataset, and it resulted in 2048 new features (1024 for P1 and 1024 for P2).

Utilizing ProtBERT embedding already proved to be extremely effective for a wide variety of tasks based on sequence-derived features[7][40], even more so than PSSMs both on residue [37] and protein [23] levels. Whether or not this will hold for this task will be established below.

Positional Encoding is a way to represent positional information in a matrix shape. It is widely used in transformer models to convey positional information of each word in a sentence. The length of the sentence defines the number of rows in a matrix, while the dimensionality parameter (usually 1024 to match the transformer's embedding size) defines the number of columns. For each row a unique vector is created by taking sine or cosine functions (for even and non-even columns, respectively) with unique frequencies, calculated by combining the column and row values. A more detailed description of the positional encoding calculation can be found in the Appendix 1.

In this research, positional encoding will be used to represent the PTM Site in the protein sequence. Discussed below in more detail, two approaches will be used: a pure 10-dimensional positional encoding of the PTM Site and a weighted mean of positional encoding and ProtBERT embedding of the PTM Site. That way it will be tested, whether positional encoding aids prediction, either in pure form or when combined with ProtBERT embedding, merely highlighting that information.

Interface Information. As was established in Bett's work, whether or not Phosphorylation happens on the protein interface, can have tremendous weight when establishing its effect on the PPI. Assuming this holds for all PTMs, a categorical variable, decoding whether or not an interface residue was modified, was added. The information in question was acquired from PIPPEN predictions [35], a tool for predicting the probability of each AA in a protein sequence being

on the interface. For each PTM-affected residue, a probability was extracted from a per-protein prediction and then converted to a categorical variable using a threshold of 0.75.

PTM Site Window (PSW) Representation. Several approaches were considered for representing the PSW. Primarily, it is important to establish that the window in question is represented by 31 AAs in total, 15 AAs upstream of the PTM Site and 15 AAs downstream, with Site-AA in the middle. As mentioned earlier, Hong et al. represented PSWs using the one-hot encoding of each AA in a given sequence snippet. However, this approach is problematic, since for a window of 9 AAs (used in their research) one-hot encoding of every AA in a window would result in 189 variables only to represent the PTM Site window (20 features for 20 standard AAs and 1 feature for a missing AA if window spans beyond sequence length per residue in the window). Note that this sparsity of features might lead to lowered computational efficiency, noise and over-fitting.

To examine whether this holds for the given situation, the first two approaches to consider were therefore set to be one-hot encoding (further referred to as OBC approach) of categorical representation of each AA in PSW (651 features in total) and mere categorical representation of each AA (further referred to as CAT approach) in PSW (31 features). Alternatively, ProtBERT Window (PW) and ProtBERT Site (PS) approaches were proposed within this research. Both are rather straightforward, PW being a ProtBERT embedding of the PSW (averaged over the column axis), and PS being ProtBERT embedding of the PTM Site (residue-level embedding).

2.4 Dimensionality Reduction Techniques

As described above, ProBERT protein-level embeddings are represented by 1024-long vectors. Since each row in the dataset describes the interaction of two proteins, incorporating ProtBERT embeddings will create 2048 new features in total, 3072 if PW or PS are chosen to represent PSW. This is highly problematic since none of the PTMint subdatasets are bigger than 3500 entries, and after the 90/10 train/test split some of them are only 2000 entries long, which means that the feature space is much more dimensional than the length of the dataset, which can result in overfitting. To avoid that, several dimensionality reduction techniques were considered as data transformation tools. Below one can find a detailed description of every technique in question.

Principal Component Analysis(PCA)[25] is a multivariate statistical technique that is often utilized to reduce the dimensionality of datasets while retaining significant variations within the data. This technique is particularly useful in scenarios where the datasets show multicollinearity since it allows one to turn linearly related variables into a set of unrelated variables via orthogonal transformation. One of the primary advantages of PCA is its high computational efficiency, as well as interpretability considering the ratio of variance, explained by the transformed components. On the downside, the principal components generated by PCA can be difficult to interpret in a meaningful manner, which

might pose challenges in understanding the transformed data, however, this is an issue for all transformation techniques.

Uniform Manifold Approximation and Projection (UMAP)[28] is a technique that became popular for its ability to handle complex data structures. Unlike PCA, which seeks to preserve the global structure, UMAP aims to maintain local neighbourhood relationships in the dataset, therefore preserving both local and global patterns within the dataset. It does that by calculating distances between neighbours in high-dimensional space and then mapping these relations to lower dimensions. UMAP has been recognized for its scalability, computational efficiency and utility for both visualization and transformation purposes. On the downside, however, it is extremely sensitive to the hyperparameters (number of neighbours and number of reduced features). If optimized incorrectly, these can dramatically change the representation of the data, resulting in a biased and uninformative feature set. In this research, the number of neighbours was set to 6 with 8 reduced features per embedding, inspired by Zhang et al.[41]

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a probabilistic dimensionality reduction technique primarily used for visualizing high-dimensional data. Similarly to UMAP, it aims to maintain the pairwise distances between data points in the high-dimensional space when they are mapped to a lower-dimensional space. While it is a very popular and useful technique for data visualisation, t-SNE is rarely used for transformation purposes. Moreover, its computational efficiency is rather low, and the algorithm is very sensitive to hyperparameters, which can lead to disturbed visualisation if applied incorrectly.

Recursive Feature Elimination (RFE) is a feature selection method, aimed at enhancing the predictive Accuracy of models. RFE operates by recursively building models and evaluating the performance of each feature, followed by the exclusion of the least significant feature at each iteration. This process continues until a predefined number of features is retained [15].

This is the most straightforward dimensionality reduction approach, however, the most computationally expensive. To tackle this problem in this research, RFE will be applied in two steps: first, a model will be trained on all features and only the 500 most informative ones will be kept. Then, an RFE process will be applied, with a step of 10 (the 10 least informative features will be excluded after each iteration), therefore decreasing the computational power needed for the process.

2.5 Models

Different Algorithms will be examined in application to the classification problem of predicting the effect of PTM on PPI, namely Random Forests, LightGBM, XGBoost and TabNet. Below, a summary of each algorithm is provided, as well as their main strengths and weaknesses.

Random forests is an ensemble learning method that was first introduced by Breiman in 2001 [4]. The model builds upon the concept of decision trees and works by creating random trees on top of each other, each consecutive tree aiming at decreasing the error of the previous one.

One of the major strengths of the Random Forests Algorithm is its robustness to overfitting, making it effective in handling both high-dimensional and noisy data. Due to this factor, it will also be used for all feature-related experiments, since, as described below, all of them will be performed on highly-dimensional feature sets.

LightGBM short for Light Gradient Boosting Machine, is an ensemble learning framework introduced by Ke et al. in 2017 [20]. Unlike traditional gradient boosting algorithms that grow trees depth-wise, LightGBM grows trees leaf-wise, which allows for more complex models and better Accuracy

One of the key strengths of LightGBM is its efficiency. It is optimized for speed and is capable of handling large datasets with lower memory usage. Additionally, the framework is highly customizable, offering various hyperparameters for tuning the model to improve its performance on different tasks. LightGBM also provides features like categorical variable support without the need for prior encoding, which simplifies the preprocessing steps. However, LightGBM has its shortcomings. It can be sensitive to overfitting, especially on smaller datasets. The model also requires careful tuning, as inappropriate hyperparameters can lead to poor performance.

XGBoost short for eXtreme Gradient Boosting, is an optimized distributed gradient boosting library introduced by Chen and Guestrin in 2016 [8]. The framework is designed to be highly efficient, flexible, and portable. It builds upon the principles of gradient-boosted trees and grows depth-wise.

One of the notable strengths of XGBoost is its scalability. It can handle large datasets and high-dimensional feature spaces effectively. The framework also provides a wide range of hyperparameters for model tuning, making it highly customizable for different tasks. However, it is generally much slower than LightGBM or Random Forests, which, in turn, can make the hyperparameter tuning process time-consuming.

Tabnet is a deep learning model explicitly designed for tabular data, introduced by Arik and Pfister in 2021 [2]. One of the distinguishing features of TabNet is its use of attention mechanisms to perform feature selection at each decision step. This enables the model to focus on different subsets of features for different tasks, making it highly flexible and interpretable. However, the model can be tedious in practice, since it is not yet supported by sklearn's pipelines, therefore the hyperparameter process has to be set up manually. Additionally, it generally requires a considerably large dataset for effective training.

2.6 Evaluation Metrics

To evaluate the performance of the models, classic classification metrics were chosen: Accuracy, Precision, Recall, F1-score, Matthews Correlation Coefficient (MCC), and Receiver Operating Characteristic (ROC-AUC).

Accuracy is a straightforward yet arguably the least informative metric. It computes the overall ratio of correct predictions to the total number of predictions,

given by:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Precision is a metric signifying the percentage of correct classifications to a specific class out of all instances that were classified to be that class by a model, therefore penalizing for False Positives. For model comparison purposes, weighted Precision will be used, given by:

$$\text{Weighted Precision} = \sum_{i=1}^{n} \left(\frac{\text{True Positives}_i}{\text{True Positives}_i + \text{False Positives}_i} \right) \times \text{Weight}_i$$

Recall, in turn, penalizes for False Negatives. Hence, it denotes the percentage of correctly identified class representatives out of true class representatives. Again, the weighted version of Recall will be used, given by:

$$\text{Weighted Recall} = \sum_{i=1}^{n} \left(\frac{\text{True Positives}_i}{\text{True Positives}_i + \text{False Negatives}_i} \right) \times \text{Weight}_i$$

The F1 score is mostly employed for binary classification problems, especially in scenarios where false positives and false negatives bear different costs, or when the class distribution is skewed. The formula for the F1 score is:

$$\text{Weighted F1} = \sum_{i=1}^{n} \left(2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \right) \times \text{Weight}_i$$

MCC offers a balanced measure of a model's true positive, true negative, false positive, and false negative rates, presenting a nuanced view of the model's performance, unaffected by class imbalance.

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

The ROC-AUC score represents the area under the ROC curve, typically depicted as a plot of the true positive rate (Sensitivity) versus the false positive rate (1-Specificity) for different thresholds. The ROC-AUC score interprets the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. It is frequently employed in a graph form to establish the optimal classification threshold balancing false and true positive rates, but can also serve as a classification metric due to its robustness against class imbalance.

$$\text{True Positive Rate (Sensitivity)} = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate (1-Specificity)} = \frac{FP}{FP + TN}$$

Since the ROC-AUC value applies solely to binary classification problems, for the ALL-ALL and ALL-PHOS datasets, a weighted one-vs-one ROC-AUC score was utilized. For each pair of classes, the ROC-AUC value will be computed and the weighted mean of these values will adopted as a metric.

2.7 Experimental set-up

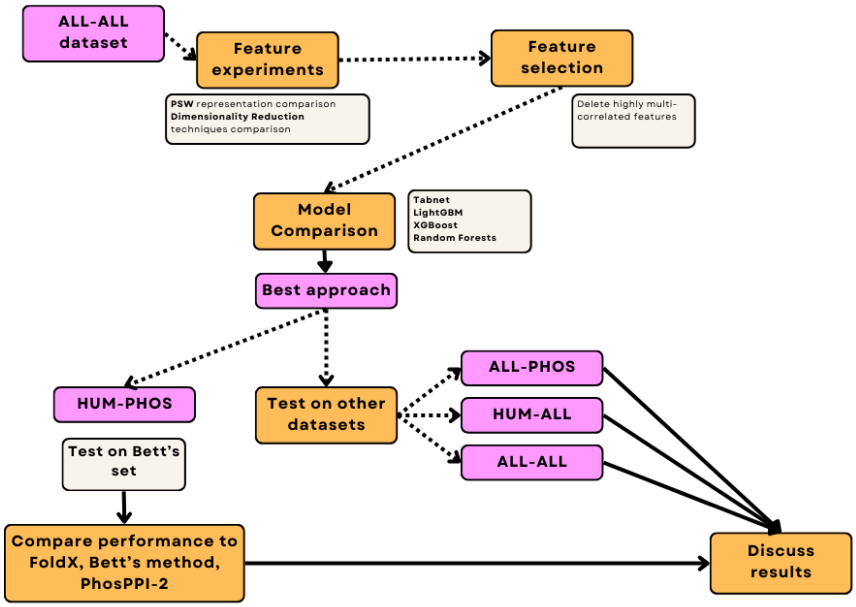


Fig. 3: Flow Chart of the Experimental Procedure

Given that the ultimate goal of the current research is to develop a predictor of the effect of PTMs on PPIs (beyond Phosphorylation), all preliminary experiments were performed on the ALL-ALL dataset, which was split into 90% train and 10% test sets using sklearn's `train_test_split` function with random state 42. The experiments consider comparing PSW representations, comparing different dimensionality reduction techniques and choosing the best-performing algorithm based on the resulting feature set.

For the first PSW experiment, four approaches were considered (described in detail in Table 3). Firstly, one-hot encoding (OBC) and Categorical representations (CAT) of the AAs in the (-15, +15) PSW were considered. The other two approaches were based on full ProtBERT embeddings, of the PTM Site only (PS) and of the averaged PSW embedding (PW). To keep the comparison fair, all PSW representations were used in combination with a full protein-level representation of the Interaction partner protein (P2). Four resulting feature sets were then used to train Random Forests algorithms (random state 10) and the performance of resulting models was used to evaluate PSW representation informativeness.

The second PSW experiment was aimed at establishing whether or not utilizing positional encoding of the PTM Site aids prediction. Based on which approach

		Feature Set	N_{feat}
1st PSW experiment	CAT	Integer encoding of each AA in the PSW (31), full protein-level ProtBERT embedding of P2 (1024) PTM type (1)	1056
	OBC	One-hot Encoding of each AA in the PSW (651), full protein-level ProtBERT embedding of P2 (1024) PTM type (1)	1676
	PW	Mean of ProtBERT embeddings of the AAs in the PSW (1024), full protein-level ProtBERT embedding of P2 (1024) PTM type (1)	2049
	PS	ProtBERT embedding of the Site-AA (1024), full protein-level ProtBERT embedding of P2 (1024) PTM type (1)	2049
2nd PSW experiment	CPP	Winning approach from 1st PSW experiment (31/651/1024), 10-dimensional positional encoding of the PTM Site (10), full protein-level ProtBERT embedding of P2 (1024) PTM type (1)	1066/ 1686/ 2059
	MPP	average of PS or PW (depending on whether one of them is the best PSW representation) and 1024-dimensional Positional encoding or PTM Site (1024), ProtBERT embedding of P2 (1024), PTM type (1)	2049
Dimensionality Reduction experiments	UMAP	UMAP transformation of protein-level ProtBERT embeddings of P1 (8), P2(8), winning approach from PSW experiments, PTM type (1)	+16/ +24
	t-SNE	t-SNE transformation of protein-level ProtBERT embeddings of P1 (3) and P2(3),winning approach from PSW experiments, PTM type (1)	+6/ +9
	PCA	PCA transformation of protein-level ProtBERT embeddings of P1 (200) and P2 (200), winning approach from PSW experiments, PTM type (1)	+400/ +600
	RFE	RFE of protein-level ProtBERT embeddings of P1 and P2, winning approach from PSW experiments, PTM type (1)	To be established

Table 3: Set-up of feature-related experiments

is proven to be the best in the first PSW experiment, different experimental procedures were considered. For OBC and CAT approaches the only tested scenario will be adding 10-dimensional positional encoding of the PTM Site to the model. Two strategies will be tested if PW or PS prove to be the most informative: adding 10-dimensional encoding (CPP), and using a weighted mean of positional encoding of the PTM Site on the matrix level for PW and vector level for PS (ProtBERT embedding weighing twice as much as the positional encoding) (MPP). For PW, after positional encoding is incorporated into the PTM Site row, the mean of all AAs will be taken over 1024 columns, flattening the matrix the same way as it is for protein-level embeddings.

The last Feature experiment considered comparing different dimensionality reduction techniques for ProtBERT embedding transformation. Again, the comparison was based on the performance of models trained on datasets containing winning PSW representation strategy and differently transformed ProtBERT embeddings of P1 and P2. Random Forests algorithm was used for performance comparison, trained and tested on the ALL-ALL dataset.

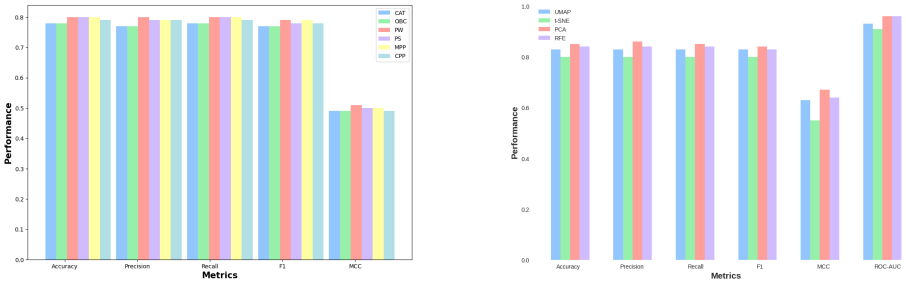
After the best feature set is established, the performance of LightGBM, TabNet, XGBoost and Random Forests will be compared when trained on ALL-ALL dataset with these features. Then, the best model will be applied to the other three sub-datasets, with separate hyperparameter tuning performed for each dataset. The resulting four predictors based on four PTMint sub-datasets will be tested on respective test sets (one per sub-dataset). The model based on the HUM-PHOS sub-dataset will be additionally tested on the Bett's benchmark test set to compare it to other methods for predicting the effect of Phosphorylation on PPI, such as FoldX, Hawdock and PhosPPI-2.

3 Results

3.1 Feature experiments

The first PSW experiment revealed that the PW approach provides the best results for classification, with Precision, Recall, F1 and Accuracy metrics of 0.80 and 0.50 for MCC. Therefore, the 2nd PSW experiment considered two ways of incorporating positional encoding information: combining it with the ProtBERT embedding on the matrix level using the mean transformation, and then flattening the resulting matrix, and using the 10-dimensional positional encoding of the PTM Site separately. The results revealed that both positional representations in fact decreased performance, with the second approach making performance slightly worse than the first. It was therefore decided to not use positional encoding since ProtBERT embeddings already hold that information and attempts to highlight it do not prove themselves useful. The visualisation of the results can be found in Figure 4a.

Consequently, three sets of ProtBERT embeddings were used for dimensionality reduction experiments: window-level embedding of the PTM Site and protein-level embeddings of P1 and P2, with 3076 features in total. The experiment revealed (Figure 4b), that the PCA-transformed feature set seems to



(a) Performance comparison of different PSW representation strategies

(b) Performance comparison of various dimensionality reduction techniques

Fig. 4: Results of feature experiments

produce the best results (Accuracy and Recall of 0.85, Precision of 0.86, F1 of 0.84 and MCC and ROC-AUC of 0.67 and 0.96, respectively). However, PCA transformation also produced the most features, with 200 transformed variables per embedding set (the number chosen based on the amount of the explained variance, the corresponding graph can be found in the Appendix), which results in 600 features per dataset, increasing the chance of overfitting. Therefore, the second-best technique was chosen, namely RFE, which left 90 features scored 0.84 for Accuracy, Precision and Recall, 0.83 for F1 and 0.64 and 0.96 for MCC and ROC-AUC, respectively. Most importantly, RFE revealed that 90 features are already enough for an Accuracy of 0.83 (Figure 5). However, before proceeding to the final experiments, the features with high VIF scores (a measure of multicollinearity) had to be removed to avoid an overly complex model.

The final feature set (presented in Table 1) consisted of 74 ProtBERT embedding features chosen with RFE (25 of them representing the PSW embedding, 19 representing P1 and 24 for P2). Moreover, a categorical feature representing whether or not PTM happened on the interface was used, and, for ALL-ALL and HUM-ALL datasets, the categorical variable representing the type of the PTM was included for prediction.

	Description	N_{feat}
PW	RFE-selected ProtBERT PTM window embedding features	25
P1	RFE-selected ProtBERT embedding features of P1	19
P2	RFE-selected ProtBERT embedding features of P2	24
Is_int	Categorical representation of whether or not PTM happened on the interface	1
PTM type	Categorical representation of the type of PTM	1

Table 4: The Final Feature Set

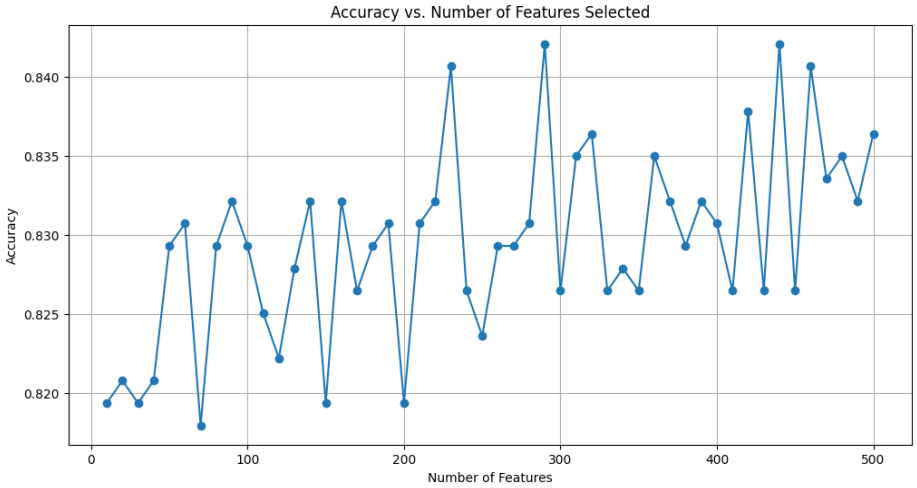
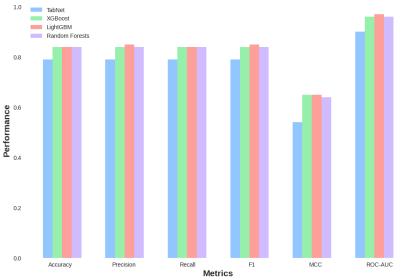


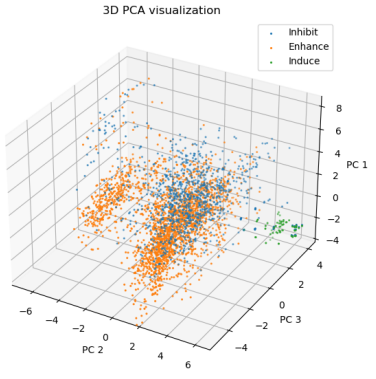
Fig. 5: Graph of Accuracy by number of Features (RFE)

3.2 Algorithms comparison

Lastly, the performances of four different algorithms were compared when applied to the final feature set for the ALL-ALL dataset (Figure 6a). Results revealed LightGBM to perform slightly better than the rest, with Accuracy and Recall of 0.84, Precision and F1 of 0.85, MCC of 0.65 and ROC-AUC of 0.97. Therefore, it was chosen as a final algorithm and applied to four PTMint sub-datasets using the final feature set.



(a) Comparison of the behaviour of different models when applied to ALL-ALL dataset with final feature set



(b) Distribution of 3 Principal Components of the final feature set coloured by Effect Values

3.3 Classification results on PTMint sub-datasets

Table 5 presents the results of final models trained on different PTMint sub-datasets and applied to the respective test sets. For each sub-dataset, the hyperparameters were optimized separately using Random Search, the optimal parameters for each sub-dataset can be found in the Appendix. One can see that in terms of Accuracy, Precision, Recall and F1, the models trained on different sub-datasets score almost the same (0.84 or 0.83). This can mean that the predictor is generalizable to predicting the effects of different PTMs in different organisms, despite the imbalance of data. However, it will be further explored with feature importance graphs and ROC-curves graphs analysis.

It is important to consider the fact, that the final ALL-ALL-based model, which underwent hyperparameter tuning, actually performs slightly worse than the untuned version used for Algorithm comparisons. This means that the hyperparameter tuning process converged to an optimal, but not the best solution, most likely because Random Search was performed, instead of a Grid one.

Method	Accuracy	Precision	Recall	F1	MCC	ROC
HUM-PHOS 0	0.84	0.84	0.84	0.84	0.61	0.91
HUM-ALL	0.84	0.84	0.84	0.84	0.59	0.89
ALL-PHOS	0.84	0.84	0.84	0.84	0.64	0.95
ALL-ALL	0.84	0.83	0.84	0.84	0.63	0.95

Table 5: Performance of the LightGBM trained and tested on different PTMint sub-datasets with final feature sets

When considering MCC and ROC-AUC scores, one can see that models trained on ALL-PHOS and ALL-ALL datasets seem to score slightly higher, than those trained on the Human subsets. In case with ROC-AUC curves this can be explained by 'Induce' class having the ROC-AUC of 1 (as shown in figures 7c and 7d. This, in turn, can be the consequence of the 'Induce' class being the most distinguished in the dataset, as shown in Figure 6b, representing the scatterplot of three PCA components of the final feature set. One can see that the 'Induce' class holds a very specific space in the 3D graph, whilst 'Enhance' and 'Inhibit' are highly mixed together on components 1 and 2, with only slight distinction on the 3rd component ('Enhance' class seems to have lower values). Interestingly enough, the 'Induce' class representatives seem to be closer to 'Inhibit' representatives on the 3rd component. This is surprising, since by definition, 'Induce' should be closer to 'Enhance', however, given that ProtBERT embeddings are purely based on the sequence of proteins, and our dataset, in turn, is based mostly on ProtBERT embeddings, the distinct pattern of 'Induce' representatives can be better explained by them belonging to non-human interactions, rather than anything else.

Finally, to validate the proposed framework, the HUM-PHOS-based model was applied to Bett's Benchmark test set, and its classification results were

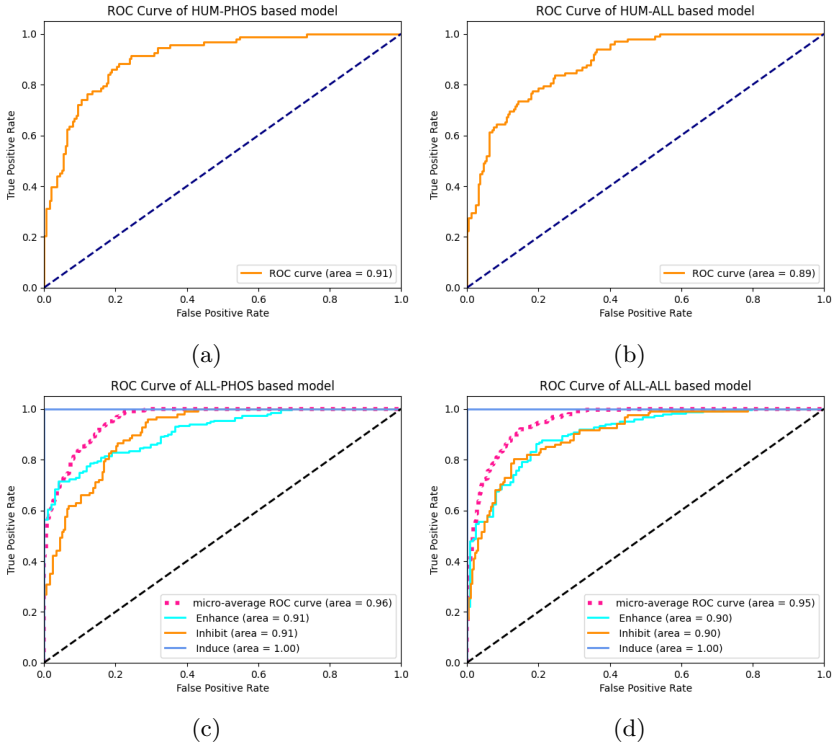


Fig. 7: ROC curves of LightGBM models trained and tested on PTMint sub-datasets

compared to previous methods, namely FoldX, Bett's method and PhosPPI-2. As can be seen in Table 6, our method considerably outperforms both Bett's method and FoldX on all the metrics. However, when considering the comparison to PhosPPI-2, the situation is less straightforward. Our method slightly outperforms PhosPPI-2 on Accuracy, MCC and ROC-AUC (0.79, 0.48, 0.81 and 0.78, 0.47, 0.80, respectively). However, PhosPPI-2 considerably outperforms our method on Recall and F1 values. Yet it is, unfortunately, unclear whether they use a weighted, macro or micro-averaging technique, therefore one cannot say with certainty, whether the presented comparison is fair.

All things considered, the results of this research show, that the proposed framework performs comparably well to the previous sequence-based framework when trained on the HUM-PHOS dataset and tested on Bett's Benchmark. Of course, for complete comparison, PhosPPI-2 predictions should have been provided to the HUM-PHOS test set, yet unfortunately, it is not possible, since their web server is currently unavailable.

Nevertheless, the results of our approach on the HUM-PHOS dataset support the findings of Hong et al. regarding the utility and performance of automated

Method	Accuracy	Precision	Recall	F1	MCC	ROC
Our method	0.79	0.79	0.79	0.79	0.48	0.81
PhosPPI-2	0.78	NA	0.87	0.85	0.47	0.80
Bett's	0.74	NA	0.35	0.51	0.53	0.62
FoldX	0.78	NA	0.01	0.02	0.41	0.46

Table 6: Performance comparison among our method (LightGBM trained on HUM-PHOS sub-dataset), PHOSPPI-2, Bett's Method and FoldX

predictors of the effect of Phosphorylation on PPIs. Additionally, our approach retains its performance when extended to predicting the effect of Phosphorylation in different organisms, proving the generalizability of the prediction. Moreover, it proves to be useful for predicting the effect of different PTMs in human and non-human organisms, too.

3.4 Feature importances

To further investigate the performance of the model trained and tested on the ALL-ALL sub-dataset, it is important to consider feature importance graphs. For the LightGBM model, there are two types of importance metrics for the feature set: split and gain. Split feature importance shows how often a specific feature was used to split data over trees (Figure 8). Gain feature importance quantifies the reduction in loss after the feature is used to split the data over trees (Figure 9).

As can be seen from the figures, the feature representing the type of the PTM does not seem to provide much information, at least, according to the model. This most likely stems from the huge imbalance of the PTMs in the dataset, with Phosphorylation constituting most of the dataset. The implications of this phenomenon, as well as possible solutions, will be discussed below. Another thing to consider when looking at feature importance graphs is the different positions the of *Is_int* feature in graphs representing split and gain importance. In the split importance graph (Figure 8), it takes the last position, while in the gain importance graph (Figure 9), it is the third feature on top. Therefore, even though it is rarely used by the model to split the data over trees, it still considerably improves the prediction capacity. This finding further supports Bett's findings regarding the importance of knowing whether or not Phosphorylation (or, possibly, any PTM) happened on the interface of the protein to predict the effect.

4 Discussion

This work presents a systematic in-depth analysis of different features, dimensionality reduction techniques and algorithms for the prediction of the effect of PTM on PPI. The final model has LightGBM architecture and is trained on a combination of ProtBERT embeddings of PTM-affected protein, PSW and Interaction Partner protein, as well as information about whether or not PTM

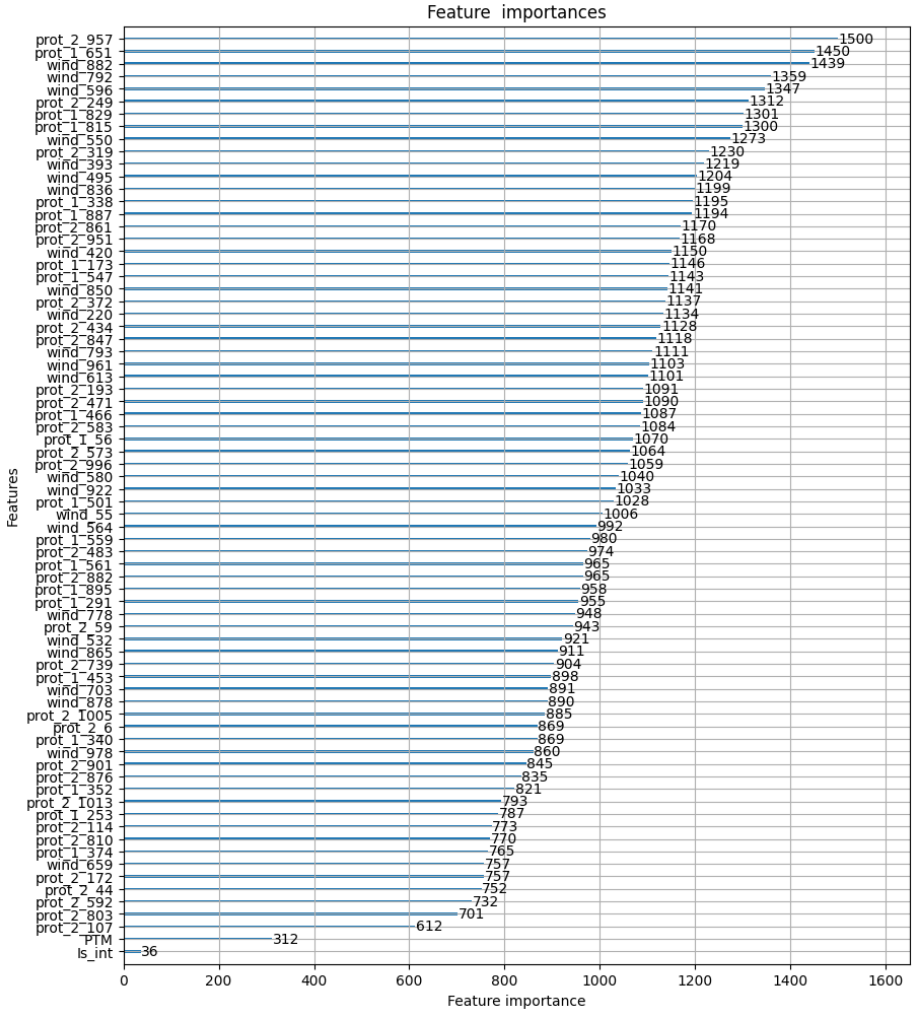


Fig. 8: Split feature importances for LightGBM model trained on ALL-ALL dataset

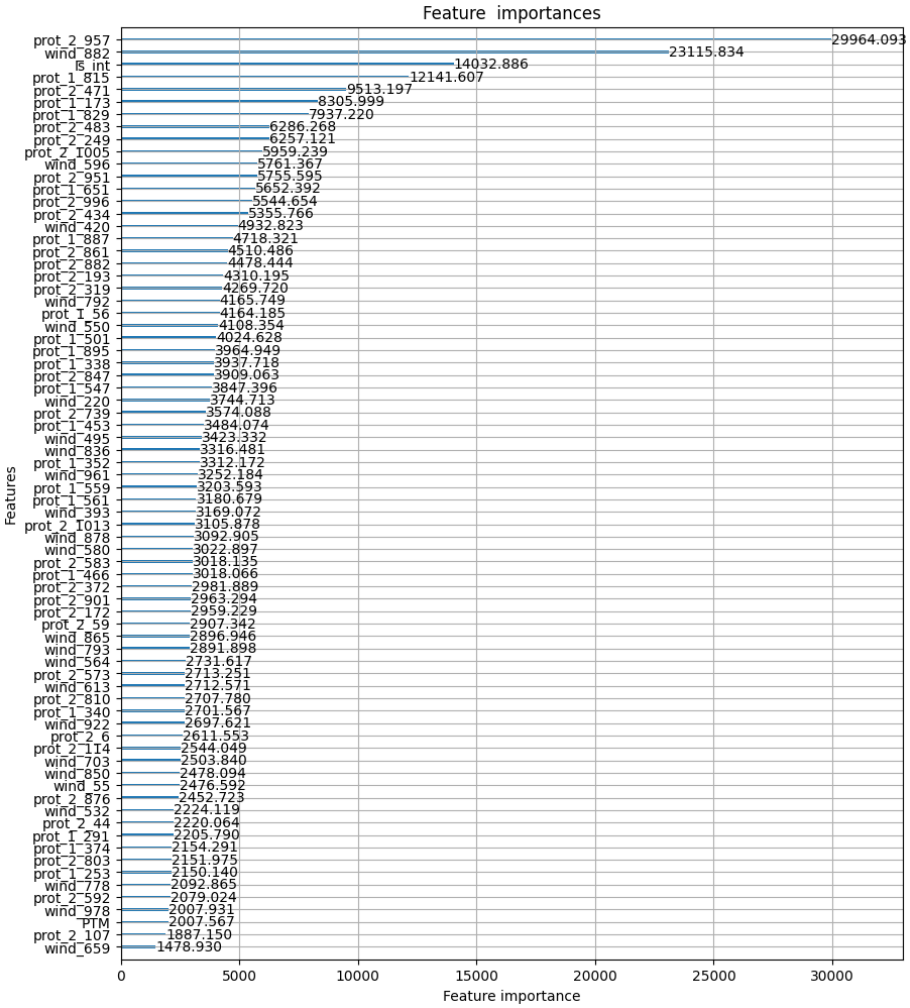


Fig. 9: Gain feature importances for LightGBM model trained on ALL-ALL dataset

happened on the interface. For ALL-ALL and HUM-ALL datasets, the type of PTM is also used for prediction. The study revealed, that while predicting the effect of only Phosphorylation in human and non-human organisms is possible with considerable Accuracy, further research is needed to establish a way to represent the PTM as a feature to make the prediction of any PTM on PPI possible.

Specifically, the results of the model trained on the subset of data considering Phosphorylation data in human (HUM-PHOS sub-dataset) proteins on Bett's Benchmark show that not only does our approach significantly outperform FoldX and Bett's methods, but also offers just as much predictive power with less computational costs than that of Hong et al. Specifically, computing PSSMs requires aligning any given sequence against an NCBI-nr dataset, which is larger than 150GB, while creating ProtBERT embeddings, even though time-consuming and provides a lot more features, doesn't require as much memory. Therefore, while our method does add to the efficiency by training a model on a considerably smaller feature set and avoiding the PSSM computation, it does not seem to add predictive power. Moreover, given the gap in the performance of our method on the HUM-PHOS test set and Bett's benchmark set, one can argue, that the PTMint dataset is not necessarily representative enough to build a good classifier. This assumption can be further supported by the fact, that Hong et al. [17], who developed the PhosPPI-2 tool, also experienced a drop in performance when applying their method to the Bett's Benchmark set.

However, when considering the performance of our approach extended to other sub-datasets, the results should be treated with caution. For example, the model trained on the ALL-PHOS dataset shows high performance on the test set, yet one should keep in mind, that the 'Induce' class is only present for non-human interactions, and datapoints describing the 'Induce' class, therefore, show a very specific pattern, while 'Enhance' and 'Inhibit' are harder to distinguish. However, it is unclear, whether with the increased representation of that class in the dataset the pattern would remain as specific. Moreover, as mentioned earlier, 'Induce' values seem to be closer to 'Inhibit' values on the only axis that shows some distinction between 'Enhance' and 'Inhibit' values. Given that the PCA analysis used to construct the graph was applied only to RFE-selected ProtBERT embeddings, which do not provide any information about the PTM, one possible explanation for such a picture could be that if the binding affinity of two proteins is low, PTM is more likely to decrease it. However, at this stage, this is merely speculation, especially given that there is no 'Stop' class in the dataset, which makes it impossible to say whether the same pattern holds vice versa. Moreover, it is unclear whether 'Induce' should be considered a subclass of 'Enhance'.

Lastly, despite our model achieving quite high results on HUM-ALL and ALL-ALL datasets, the fact that the PTM type feature produces very little gain and is very rarely used for splitting the data over decision trees, is problematic. This shows that somehow the model gets enough information from the sequence-based features to predict the effect with high Accuracy, yet it entirely misses the objective of the task. This phenomenon is not unexpected, since the dataset is

severely imbalanced, with Phosphorylation data constituting most of the dataset. It is possible, that the model does not receive enough alternative information to discover the importance of the *PTM* feature. However, another way to explain such behaviour would be that mere categorical representation of the PTM type is not enough.

When discussing the PTMint dataset, one's attention should be drawn to the fact that all records concerning one of the proteins were removed from the dataset since the mutated version was researched in the source paper, therefore the correct sequence could not be accessed from the UniProt Fasta file. However, that was only found to be the case since the mutated version was longer, and the PTM happened at the end of the sequence, which disrupted the code. Therefore, it is not possible to say, whether or not there were other such instances in the dataset, which in turn means, that some of the ProtBERT embeddings could have been describing a standard protein sequence, while the mutated one should have been considered, which might have led to bias in the model. For further research, it is important to manually check every record in the PTMint dataset before retrieving the protein sequences, if they are needed.

Coming back to our method, a few things should be considered for further research. Firstly, as mentioned in the Introduction section, PTM can affect a PPI in three ways: change structure, change localization or happen on the interface and directly affect the binding affinity. Whilst our method does attempt to account for the information that might help detect the third type of influence, the first two types of influence are not represented. Subsequently, ways to convey that information should be explored, such as incorporating biophysical properties of the chemical residue added, or, perhaps, a change of those for the AA to which it was added. Alternatively, a BERT model could be trained on sequences with PTMs, perhaps, representing PTMs as syllables. Given the state-of-the-art performance of ProtBERT on numerous tasks, such an approach could prove fruitful for the field of predicting and analyzing the effect of PTMs on different processes.

Secondly, a more extensive exploration of how to incorporate interface information is needed, especially for the interaction partner protein. Perhaps, positional encoding of the ten most likely interface Sites (as predicted by PIPENN) could be incorporated. Another way to examine that could be to use PPI interface prediction tools directly, and later use ProtBERT embeddings of those AAs.

Moreover, on a more technical side, it is important to note, that Random Search was used for the hyperparameter optimization, and as shown in the Results, the hyperparameter values found are merely optimal. On top of that, the RFE procedure was performed with a step of 10, which could mean that certain informative features could have been omitted due to random fluctuations in the model. Additionally, no cross-validation was performed for the final results of the models trained on different PTMint sub-datasets, which could result in the different performance of the proposed framework when faced with a differently sampled test set.

All things considered, predicting the effect of PTM on PPI is a promising field. Despite all the limitations of the PTMint dataset, lack of scientific attention and specific pitfalls of the proposed approach, this study shows that it is not only possible to predict the effect of Phosphorylation on PPI, but that of other PTMs as well. Moreover, throughout this study, it was established that PCA transformation is most informative when reducing the dimensionality of the ProtBERT embeddings, yet when taking efficiency into the equation, using feature elimination algorithms can be more beneficial.

5 Data availability

All code used for the research is provided in the github repository.

6 Appendix

6.1 Positional Encoding calculation

1. Angle Rate Calculation: A unique angle rate is calculated for each dimension of the model. This is achieved through the formula:

$$\texttt{angle\_rates} = \frac{1}{10000^{(2i/d)}}$$

   where i ranges from 0 to $d_{\text{model}} - 1$. This formula ensures a geometric progression in the frequency of positional encodings across dimensions, which is believed to provide a better basis for learning positional relationships.
2. Angle Radian Calculation: The angle in radians for each position in the sequence is then calculated by multiplying the `position` by the `angle_rates`. This operation aligns with the positional encoding formula, applying a scaling factor to the position of each token:

$$\texttt{angle\_rads} = \texttt{position} \times \texttt{angle\_rates}$$

3. Application of Trigonometric Functions: The sine function is applied to the angle radians of even-indexed dimensions, while the cosine function is applied to the angle radians of odd-indexed dimensions:

$$\texttt{angle\_rads}_{[:,0::2]} = \sin(\texttt{angle\_rads}_{[:,0::2]})$$

$$\texttt{angle\_rads}_{[:,1::2]} = \cos(\texttt{angle\_rads}_{[:,0::2]})$$

   This step is in alignment with the original positional encoding formula, ensuring each position gets a unique encoding which can be utilized by the model to learn the ordering of tokens.

6.2 Hyperparameter optimization procedure

Since LightGBM was chosen for the prediction task throughout the performance comparison with other models, only LightGBM's hyperparameters were optimized for further experiments. Specifically,

1. n_estimators. Refers to the number of boosting rounds or the number of trees to build. Higher values can lead to better performance but might increase the risk of overfitting and also require more computational time. values considered: [100, 200, 300, 400, 500]
2. max_depth, Indicates the maximum depth of a tree. Deeper trees can model more complex patterns but might lead to overfitting. values considered: [-1, 10, 15, 20, 25]
3. num_leaves. Specifies the maximum number of leaves for each tree. This parameter is crucial for controlling model complexity.More leaves allow the model to learn finer details but increase the risk of overfitting. values considered: [15, 31, 63, 127]
4. learning_rate. Determines the step size at each iteration while moving toward a minimum of a loss function. Smaller values require more boosting rounds but can lead to better Accuracy and less overfitting. values considered: [0.1, 0.05, 0.01, 0.005]
5. min_child_samples. The minimum number of data samples required to be in a leaf node (child). A larger number can prevent the model from learning relations which might be highly specific to a particular sample. values considered: [10, 20, 30, 40]
6. subsample. Fraction of the training data to be used for fitting each tree. Values less than 1.0 make the algorithm more conservative and prevent overfitting but values that are too small can lead to underfitting. values considered: [0.6, 0.7, 0.8, 0.9, 1.0]
7. colsample_bytree. The fraction of features (columns) to be used for each tree. A smaller value can provide more diversity among the trees, enhancing the model's ability to generalize, but can also make individual trees weaker. values considered: [0.6, 0.7, 0.8, 0.9, 1.0]
8. reg_alpha. L1 regularization term on weights (equivalent to Lasso regression).It can be used to reduce the complexity of the model and prevent overfitting by penalizing large coefficients. values considered:[0, 0.1, 0.5, 1]
9. reg_lambda. L2 regularization term on weights (equivalent to Ridge regression). It's used to smooth the learned weights and generally has the effect of reducing overfitting values considered: [0, 0.1, 0.5, 1]

For each sub-dataset the hyperparameter optimization was performed separately, using Random Search with 150 different hyperparameter combinations and Startified K-Fold validation with 5 rounds.

Hyperparameter optimisation results on the HUM-PHOS dataset subsample = 0.8, reg_lambda = 0.1, reg_alpha = 0.1, n_estimators = 400, num_leaves = 63, min_child_sample = 30, max_depth = 25, learning_rate = 0.005, colsample_bytrees = 0.6

Hyperparameter optimisation results on ALL-PHOS dataset subsample = 0.7, reg_lambda = 0.1, reg_alpha = 0.1, n_estimators = 200, num_leaves = 127, min_child_sample = 40, max_depth = 15, learning_rate = 0.01, colsample_bytrees = 0.7

Hyperparameter optimisation results on HUM-ALL dataset subsample = 0.7, reg_lambda = 0.1, reg_alpha = 0.1, n_estimators = 400, num_leaves = 127, min_child_sample = 20, max_depth = 25, learning_rate = 0.005, colsample_bytrees = 0.9

Hyperparameter optimisation results on ALL-ALL dataset subsample = 0.7, reg_lambda = 0.1, reg_alpha = 0.1, n_estimators = 300, num_leaves = 63, min_child_sample = 40, max_depth = 20, learning_rate = 0.005, colsample_bytrees = 0.7

PCA results. While the hyperparameters for UMAP and t-SNE were based on previous research, the number of Principal Components had to be established experimentally. For that, ProtBERT embeddings of all the unique proteins from P1 and P2 columns were considered, and a PCA analysis was performed 20 times, with starting number of PCs of 10 and the final number of PCs of 200. The ratio of explained variance was saved for each iteration and then plotted against the number of components. Based on the graph, it was decided to keep 200 components, since it explained the most amount of variance. From the graph below it can also be seen, that the explained ratio curve rises very slowly, which first two components explaining only 20% of ratio.

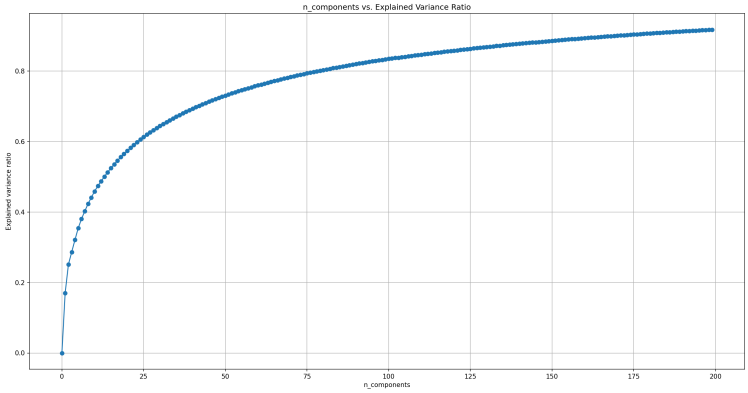


Fig. 10: Explained variance ratio by number of components

References

1. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023.
2. Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.
3. Matthew J Betts, Oliver Wichmann, Mathias Utz, Timon Andre, Evangelia Petsalaki, Pablo Minguez, Luca Parca, Frederick P Roth, Anne-Claude Gavin, Peer Bork, et al. Systematic identification of phosphorylation-mediated protein interaction switches. *PLoS computational biology*, 13(3):e1005462, 2017.
4. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
5. Fiona Browne, Huiru Zheng, Haiying Wang, and Francisco Azuaje. From experimental approaches to computational techniques: a review on the prediction of protein-protein interactions. *Advances in Artificial Intelligence (16877470)*, 2010.
6. Rita Casadio, Pier Luigi Martelli, and Castrense Savojardo. Machine learning solutions for predicting protein–protein interactions. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(6):e1618, 2022.
7. Abel Chandra, Alok Sharma, Iman Dehzangi, Tatsuhiko Tsunoda, and Abdul Sattar. Deep learning for protein peptide binding prediction: Incorporating sequence, structural and language model features. *bioRxiv*, pages 2023–09, 2023.
8. Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.
9. Federica Del Monte and Giulio Agnetti. Protein post-translational modifications and misfolding: New concepts in heart failure. *PROTEOMICS–Clinical Applications*, 8(7-8):534–542, 2014.
10. Ken A Dill. Dominant forces in protein folding. *Biochemistry*, 29(31):7133–7155, 1990.
11. Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.
12. Stephen M Fuchs and Brian D Strahl. Antibody recognition of histone post-translational modifications: emerging issues and future prospects. *Epigenomics*, 3(3):247–249, 2011.
13. Anne-Claude Gavin, Markus Bösche, Roland Krause, Paola Grandi, Martina Marzioch, Andreas Bauer, Jörg Schultz, Jens M Rick, Anne-Marie Michon, Cristina-Maria Cruciat, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, 2002.
14. Mileidy W Gonzalez and Maricel G Kann. Chapter 4: Protein interactions and disease. *PLoS computational biology*, 8(12):e1002819, 2012.
15. Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.
16. Xiaokun Hong, Ningshan Li, Jiyang Lv, Yan Zhang, Jing Li, Jian Zhang, and Hai-Feng Chen. Ptmint database of experimentally verified ptm regulation on protein–protein interaction. *Bioinformatics*, 39(1):btac823, 2023.

17. Xiaokun Hong, Jiyang Lv, Zhengxin Li, Yi Xiong, Jian Zhang, and Hai-Feng Chen. Sequence-based machine learning method for predicting the effects of phosphorylation on protein-protein interactions. *International Journal of Biological Macromolecules*, page 125233, 2023.
18. Jianghui Hou, Vijay Renigunta, Mingzhu Nie, Abby Sunq, Nina Himmerkus, Catarina Quintanova, Markus Bleich, Aparna Renigunta, and Matthias Tilmann Florian Wolf. Phosphorylated claudin-16 interacts with trpv5 and regulates transcellular calcium transport in the kidney. *Proceedings of the National Academy of Sciences*, 116(38):19176–19186, 2019.
19. Ying Huang, Beifang Niu, Ying Gao, Limin Fu, and Weizhong Li. Cd-hit suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, 26(5):680–682, 2010.
20. Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.
21. DG Knorre, NV Kudryashova, and TS Godovikova. Chemical and functional aspects of posttranslational modification of proteins. *Acta Naturae ( )*, 1(3 (3)):29–51, 2009.
22. Martin R Larsen, Morten B Trelle, Tine E Thingholm, and Ole N Jensen. Analysis of posttranslational modifications of proteins by tandem mass spectrometry: Mass spectrometry for proteomics analysis. *Biotechniques*, 40(6):790–798, 2006.
23. Hansol Lee, Songyeon Lee, Ingoo Lee, and Hojung Nam. Amp-bert: Prediction of antimicrobial peptide function based on a bert model. *Protein Science*, 32(1):e4529, 2023.
24. Karl-Johan Leuchowius, Irene Weibrecht, and Ola Söderberg. In situ proximity ligation assay for microscopy and flow cytometry. *Current protocols in cytometry*, 56(1):9–36, 2011.
25. Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993.
26. Matthias Mann and Ole N Jensen. Proteomic analysis of post-translational modifications. *Nature biotechnology*, 21(3):255–261, 2003.
27. Christopher J Marshall. Protein prenylation: a mediator of protein-protein interactions. *Science*, 259(5103):1865–1866, 1993.
28. Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
29. Lingkuan Meng, Wai-Sum Chan, Lei Huang, Linjing Liu, Xingjian Chen, Weitong Zhang, Fuzhou Wang, Ke Cheng, Hongyan Sun, and Ka-Chun Wong. Mini-review: Recent advances in post-translational modification site prediction based on deep learning. *Computational and Structural Biotechnology Journal*, 2022.
30. Zainab Noor, Seong Beom Ahn, Mark S Baker, Shoba Ranganathan, and Abidali Mohamedali. Mass spectrometry–based protein identification in proteomics—a review. *Briefings in bioinformatics*, 22(2):1620–1638, 2021.
31. Shahin Ramazi, Abdollah Allahverdi, and Javad Zahiri. Evaluation of post-translational modifications in histone proteins: a review on histone modification defects in developmental and neurological disorders. *Journal of biosciences*, 45:1–29, 2020.
32. Helena Ryšlavá, Veronika Doubnerová, Daniel Kavan, and Ondřej Vaněk. Effect of posttranslational modifications on enzyme function and assembly. *Journal of proteomics*, 92:80–109, 2013.

33. Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, and Luis Serrano. The foldx web server: an online force field. *Nucleic acids research*, 33(suppl_2):W382–W388, 2005.
34. Till Siebenmorgen and Martin Zacharias. Computational prediction of protein–protein binding affinities. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 10(3):e1448, 2020.
35. Bas Stringer, Hans de Ferrante, Sanne Abeln, Jaap Heringa, K Anton Feenstra, and Reza Haydarlou. Pipenn: protein interface prediction from sequence with an ensemble of neural nets. *Bioinformatics*, 38(8):2111–2118, 2022.
36. Shu Wang, Arianna O Osgood, and Abhishek Chatterjee. Uncovering post-translational modification-associated protein–protein interactions. *Current opinion in structural biology*, 74:102352, 2022.
37. Xiao Wang, Lijun Han, Rong Wang, and Haoran Chen. Dadl-schlo: protein subchloroplast localization prediction based on generative adversarial networks and pre-trained protein language model. *Briefings in Bioinformatics*, 24(3):bbad083, 2023.
38. Geoffrey I Webb, Eamonn Keogh, and Risto Miikkulainen. Naïve bayes. *Encyclopedia of machine learning*, 15:713–714, 2010.
39. Gaoqi Weng, Ercheng Wang, Zhe Wang, Hui Liu, Feng Zhu, Dan Li, and Tingjun Hou. Hawkdock: a web server to predict and analyze the protein–protein complex based on computational docking and mm/gbsa. *Nucleic acids research*, 47(W1):W322–W330, 2019.
40. Zexi Yang, Yan Wang, Xinye Ni, and Sen Yang. Deepdrp: Prediction of intrinsically disordered regions based on integrated view deep learning architecture from transformer-enhanced and protein information. *International Journal of Biological Macromolecules*, 253:127390, 2023.
41. Yaqi Zhang, Gancheng Zhu, Kewei Li, Fei Li, Lan Huang, Meiyu Duan, and Fengfeng Zhou. Hlab: learning the bilstm features from the protbert-encoded proteins for the class i hla-peptide binding prediction. *Briefings in Bioinformatics*, 23(5):bbac173, 2022.
42. Mi Zhou, Qing Li, and Renxiao Wang. Current experimental methods for characterizing protein–protein interactions. *ChemMedChem*, 11(8):738–756, 2016.