Phosphorylation Site Prediction Using Deep Learning

Filip Carević, Edvin Maid, Natalija Mitić
Machine Learning CS-433, EPFL, Switzerland

***Abstract*—Predicting amino acid residues which are phosphorylation sites is an important problem from a biological and biomedical perspective. In this paper, we propose two neural network architectures for classifying whether residues are phosphorylation sites or not. These approaches are then compared. Additionally, we show that residue sequences have a strong representational power for the given problem. Lastly, by leveraging the outputs of the PESTO model for protein interface prediction we see that the protein interface prediction problem is not easily transferable to phosphorylation site prediction using our methods.**

I. Introduction

Proteins are by far the structurally and functionally most complex molecules known. [1] Because of this, most problems related to structural prediction are usually difficult. One important structural feature of proteins is the presence of Post-translational modification (PTM) sites. The most common form of PTM is phosphorylation [2]. Phosphorylation is usually the transfer of the terminal phosphate group of an ATP molecule to one of the residues of a given polypeptide chain [1]. Phosphorylation is a mechanism that by addition of a polar phosphate molecule activates or deactivates a protein. Abnormal phosphorylation is linked to many serious illnesses such as cancer, Alzheimer's and Parkinson's disease. [3] Because of its abundance and medical relevance, the study of phosphorylation is of immense importance. From a scientific point of view understanding which residues can and will accept a phosphate molecule can shed more light on pathological, as well as physiological, processes which take part in living organisms.

Task description

Polypeptide chains can be seen as sequences where each element is one of 20 different natural amino acids. These chains are not symmetric in the sense that one end of the chain is inherently different from the other (i.e. C-terminus and N-terminus). In the process of phosphorylation, phosphate molecules are most commonly attached to three possible amino acids: Serine, Threonine and Tyrosine (abbr. S, T and Y). The task is defined as binary classification for phosphorylation site detection applied to every residue (CNN model approach) or just S, T and Y residues (Linear model approach). This includes preparing the data, selecting, training and evaluating the models, with inter-model comparison in the end.

II. Data

For the purpose of training and model evaluation, we have used two datasets - Eukaryotic Phosphorylation Sites Database and the PESTO dataset.

A. Eukaryotic Phosphorylation Sites Database

First dataset is the preprocessed Eukaryotic Phosphorylation Sites Database (EPSD). EPSD contains 1,616,804 experimentally identified phosphorylation sites in 209,326 phosphoproteins from 68 eukaryotes.[4] EPSD was created by combining multiple independent phosphorylation databases, and, therefore, contains inconsistencies which are removed in the preprocessing phase. The EPSD dataset was then expanded using trained models for structure prediction. Features provided by the EPSD, as well as the expanded features, can be found in Table I. Additionally, EPSD also contains sequential representations of whole polypeptide chains stored in a separate (fasta) file.

feature name	explanation
UniProt ID	ID of the protein from the UniProt dataset [5]
AA	Amino acid on which PTM are found
Position	Position (index) of the amino acid in the chain
Source	Method of acquiring information. `Exp.` stands for experimental.
Reference	References to scientific literature describing observed amino acid chain
ss	Secondary structure indicator. Following the DSSP nomenclature [6] value 'C' stands for coil, 'H' for helix, while 'E' depicts extended strand in parallel and/or anti-parallel ß-sheet conformation
coor_x	X coordinate of AA in the 3D space
coor_y	Y coordinate of AA in the 3D space
coor_z	Z coordinate of AA in the 3D space
afold_conf_mask	Indicator if the phosphorylation appeared in the intrinsically unordered domain
EPSD ID	ID in Eukaryotic Phosphorylation Sites Database

Table I
EPSD base and expanded features

Sequential approaches presented in this paper require only features that describe the phosphorylated residue (i.e., amino acid) and its position in the chain. Therefore, only `Position` and amino acid sequences (fasta file) are used. Moreover, features referring to the same protein (i.e., same UniProt ID) are aggregated in a list converting residue-based datapoints to protein-based datapoints. Lastly, these protein-based datapoints are merged with the aforementioned files of polypeptide chain representations.

The preprocessing of this dataset consists of filtering out proteins whose PTM information were not experimentally extracted (i.e., `Exp.` value of the `Source` feature). Afterwards, proteins whose lengths of amino acid sequences does not correspond to the information available in the Universal Protein Resource (UniProt) [5] were also filtered out.

After filtering, the dataset size is reduced to 1,146,437 phosphorylation sites in 103,691 phosphoproteins.

B. PESTO dataset

PESTO is a deep learning model based on the attention mechanism [7] which analyzes spatial/geometric positions of atoms by applying attention to individual atoms of a given structure. PESTO is being developed at EPFL by the Laboratory for Biomolecular Modeling (yet to be published). The model is trained for the task of protein interface prediction. To be more precise, it classifies whether an atom is part of an interaction interface. The model architecture produces a latent space of 64 features per atom for a given protein. These vectors are then put through a classification layer.

Dataset Generation: For the purpose of predicting phosphorylation sites, PESTO was used to create a dataset of vector representations for each amino acid in all polypeptide chains of the first data set. This is done by running the protein interface prediction model without the classification layer on all sequences of the preprocessed EPSD dataset. At this point the model output is a set of vectors corresponding to each atom. However, the task described is for classifying whole amino acid residues (which are collections of atoms). For this reason, average pooling was applied on all atoms of a single residue to generate a vector representation of a single amino acid. This vector has 64 dimensions.

III. Models and Methods

For this task two models were constructed, both of which rely on a similar principle. It is assumed that for a given residue, neighbouring residues in geometric proximity should give us more information for the prediction. Residues which are in close proximity in the amino acid chain are also geometrically close. Thus using them should yield a part of the features needed for our task.

A. CNN model

As show in [8], the surrounding amino acids of the primary structure represent an important feature. Since the data is sequential and neighbouring elements influence each other, a convolutional neural network should be suitable for such a problem. The input into the model is a sequence of one-hot vector representations with the adequate number of channels. The number of channels is 20. The output of the model is a sequence of probability estimations. Each element corresponds to the probability of it being a phosphorylation site. This model has to learn that only certain amino acids can be phosphorylation sites (S, Y and T). The model architecture consists of:

- one 1d CNN input layer with 20 input channel and 128 output channels,
- four 1d CNN layers with 128 in and out channels on which the Exponential Linear Unit (ELU) activation function was applied in an alternating manner
- three linear layers of shape [128 x 128] on which the Exponential Linear Unit (ELU) activation function was applied in an alternating manner
- one linear output layer of shape [128 x 1].

All CNN layers use a kernel of size 21 and zero padding, which preserves the length of the sequence. The idea behind this architecture is that the CNN layers should be able to extract sequential features. The output of these layers will be a sequence of the same length as the input, i.e. polypeptide chain length. Each element will then correspond to a vector representation of its respective amino acid. Finally, dense feed-forward layers are applied to classify the given amino acid.

B. Linear model

The second model architecture used for phosphorylation classification is the Linear model. This approach is inspired by the sequential representation [8] of protein structure used for the task of detecting PTM sites.

After analysing the raw data, it was found that phosphorylation appears exclusively on three different amino acid residues:

- Serine (S)
- Threonine (T)
- Tyrosine (Y)

Features used for training are extracted from every protein (i.e., sequence of amino acids). More precisely, for every amino acid that has the potential to be a phosphorylation site (S, T or Y), the nearby `N` amino acids (`N/2` in each direction) are also used as input. The motivation behind this is that adjacent amino acids of phosphorylated residues are important sequence-based features for predicting phosphorylation sites [9]. Subsequently, the list of neighboring amino acids is one-hot encoded. Since there are 20 amino acids in total, an indicator is set on the `i`-th index of the 20-dimensional binary vector for every amino acid in the neighborhood. For each of the `M` possible phosphorylation sites of a single protein, the extraction of the neighborhood of size `N` with one-hot encoding results in a neighborhood list (i.e, matrix) with the shape of $[M \text{ x } (20 * (N + 1))]$. Such a neighborhood list is used as one input datapoint for training. During the model evaluation phase on the test data, the same preprocessing of amino acids is performed.

This linear model is a multilayered, dense feed-forward neural network whose architecture consists of:

- one linear input layer of shape [420 x 128],

- eight hidden dense layers of shape [128 x 128] on which the Exponential Linear Unit (ELU) activation function was applied alternating,
- one linear output layer of shape [128 x 1].

Such a model is implicitly aware of the fact that phosphorylation sites can only be on S, T and Y residues, resulting in a more balanced task.

The second linear model has been implemented to perform classification task on the PESTO features. It is also based on the neighborhood principle. The only difference is that, instead of creating neighborhood using one-hot representations of amino acids, we used the amino acid representations from the PESTO embedding of the protein. PESTO embeddings should encode, among other things, the position in space of each amino acid and therefore, the embeddings for each amino acid are unique, i.e. Serine residues at different positions inside the same protein chain do not have the same embedding representation. Since one embedding contains 64 features, then for each of the `M` possible phosphorylation sites of a single protein, extraction of the neighborhood of size `N` and one-hot expansions result in the neighborhoods list of the shape $[M$ x $(64 * (N + 1))]$. This is used as one datapoint inside of the minibatch training.

Consequently, the architecture of the model had to be changed in order to fit the new data representation. The architecture of the linear model with PESTO embeddings consists of:

- one linear input layer of shape [64 x 32],
- two hidden dense layers of shape [32 x 32] on which the Exponential Linear Unit (ELU) activation function was applied alternating,
- one linear output layer of shape [32 x 1].

Model Training

The training process uses the Adam optimizer [10] with the initial learning rate of 0.001. The reason why Adam was preferred over the Stochastic Gradient Descent (SGD) is due to its adaptive learning rate. Moreover, in this case, empirically it was shown that the use of the Adam optimizer achieves a faster convergence than the use of SGD.

For the loss function, the Binary Cross-entropy loss (BCE) is used in combination with the sigmoid in one layer. This functionality is implemented in PyTorch library within the `BCEWithLogitsLoss` class. Note that this loss function [11] is more numerically stable than using sigmoid followed by a BCELoss as it utilizes the `log-sum-exp` trick for numerical stability.

Another issue that needed to be addressed is the heavily imbalanced data. The analysis showed that the total number of amino acids in the whole dataset is 59,451,528, of those 10,002,126 are S, T and Y residues, while total number of actual phosphorylation sites is only 1,146,437. For the task being solved by the CNN model only 2% are actual phosphorylation sites. However, the Linear model approach is notably more balanced. That is, around 11.46% of all residues are phosphorylation sites. Because of these imbalances, during training, the model would converge to output only 0 values. In order to mitigate this problem the loss function was weighted. This was done by setting the `pos_weight` attribute of the `BCEWithLogitsLoss`.

IV. Results

Model	Performance		
	Precision/ Recall AUC	Average Precision	ROC AUC
CNN model with one-hot encoding	0.379	0.379	0.969
Linear model with one-hot encoding	0.339	0.339	0.767
Linear model with PESTO embeddings	0.198	0.198	0.684

Table II
PERFORMANCES OF MODELS WITH BASE FEATURES

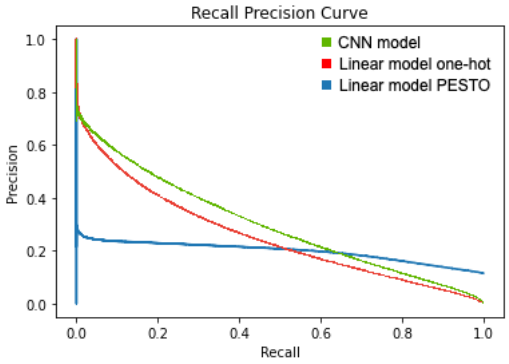

Figure 1. Comparison of Precision/Recall curves

As can be seen from the table, the CNN based model outperforms the Linear model on the AUC (area under the curve) metric of the Precision/Recall curve. In addition, metrics show that our models match the industry standards [3]. Very low values of the precision metric are obtained on ours, but also on state-of-the art models. This shows that the problem is inherently difficult.

On the other hand, due to the fact that the CNN-based approach takes into account all of the 20 amino acids during the calculations of FPR (false-positive rate) and TPR (true-positive rate), and linear-based approach, after manual extraction in the preprocessing phase, considers only Serine, Tyrosine, Threonine, the AUC measurements of ROC are not comparable between the models.

As shown in II, the Linear model with PESTO embeddings showed a considerably lower performance, which suggests that the usage of PESTO embeddings requires a different approach.

V. Discussion

Given the performances in the result section, several remarks can be made.

Firstly, it can be concluded that our models are comparable to industrial standards. Moreover, the sequential approaches described in this paper, have demonstrated that sequential features account for most of the representational power for this classification task.

Secondly, when it comes to the comparison of our models, we can conclude that CNN-based sequential models outperform the One-hot based linear models. Recall that the CNN model's architecture includes several convolutional layers, which as a consequence have the ability to extract more information from the amino acid sequence as a whole. To be more precise, the sliding window of the first layer extracts the information regarding the direct neighborhood of each amino acid, while deeper ones extract the information regarding indirect neighborhood (I.e., the second layer extracts the first degree of indirectness, the third one extracts the second degree of indirectness, and so forth). Nevertheless, the linear model observes only direct neighborhoods and requires the size of the neighborhood to be manually set. Therefore, the linear model, unlike the CNN, cannot gain insight in the whole protein sequence. This might be reasoning behind the under-performance of the linear model.

Lastly, regarding the under-performance using PESTO embeddings. We believe that the predominant loss of information occurred after the aggregation (pooling) of vector representations of individual atoms to produce a vector for each residue (recall the process of PESTO features extraction described in the Data section). In combination with the fact that PESTO was not fine-tuned for the given task, nor was it trained with the pooling layer, the expected result is a considerable drop in representational power.

VI. Conclusion

In this paper, we have demonstrated that sequential features, relying on one-hot encoding of amino acids, provide sufficient information to match the performances of the current state-of-the-art models [3]. In addition to this, we have shown that the CNN model outperforms the Linear model in this kind of classification task, i.e., the classification of the phosphorylation sites. On the other hand, our models, when trained on the features obtained as the output from the last hidden layer of the PESTO model (recall that PESTO model was trained for the protein interface prediction), exhibited under-performing results. In other words, the transfer learning from PESTO to CNN and Linear models was not very successful.

References

[1] B. Alberts, *Molecular Biology of the Cell*. W.W. Norton, 2017. [Online]. Available: https://books.google.ch/books?id=jK6UBQAAQBAJ

[2] G. A. Khoury, R. C. Baliban, and C. A. Floudas, "Proteome-wide post-translational modification statistics: frequency analysis and curation of the swiss-prot database," *Sci. Rep.*, vol. 1, Sep 2011. [Online]. Available: https://doi.org/10.1038/srep00090

[3] M. S. Wolfe, "The role of tau in neurodegenerative diseases and its potential as a therapeutic target," *Scientifica*, vol. 2012, pp. 796 024–796 024, 2012, 24278740[pmid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/24278740

[4] S. Lin, C. Wang, J. Zhou, Y. Shi, C. Ruan, Y. Tu, L. Yao, D. Peng, and Y. Xue, "EPSD: a well-annotated data resource of protein phosphorylation sites in eukaryotes," *Briefings in Bioinformatics*, vol. 22, no. 1, pp. 298–307, 02 2020. [Online]. Available: https://doi.org/10.1093/bib/bbz169

[5] The UniProt Consortium, "UniProt: the universal protein knowledgebase," *Nucleic Acids Research*, vol. 45, no. D1, pp. D158–D169, 11 2016. [Online]. Available: https://doi.org/10.1093/nar/gkw1099

[6] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983. [Online]. Available: http://dx.doi.org/10.1002/bip.360221211

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[8] S. Jamal, W. Ali, P. Nagpal, A. Grover, and S. Grover, "Predicting phosphorylation sites using machine learning by integrating the sequence, structure, and functional information of proteins," *Journal of Translational Medicine*, vol. 19, no. 1, p. 218, May 2021. [Online]. Available: https://doi.org/10.1186/s12967-021-02851-0

[9] T. Li, P. Du, and N. Xu, "Identifying human kinase-specific protein phosphorylation sites by integrating heterogeneous information from various sources," *PLOS ONE*, vol. 5, no. 11, pp. 1–8, 11 2010. [Online]. Available: https://doi.org/10.1371/journal.pone.0015411

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[11] "BCEWithLogitsLoss — PyTorch 1.10.1 documentation," https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html, (Accessed on 12/20/2021).