PhosphoLingo: protein language models for phosphorylation site prediction

Jasper Zuallaert[1,2], Pathmanaban Ramasamy[1,2], Robbin Bouwmeester[1,2], Nico Callewaert[1,3], Sven Degroeve[1,2,§]

[1]VIB-UGent Center for Medical Biotechnology, VIB, Ghent, Belgium; [2]Department of Biomolecular Medicine, Faculty of Medicine and Health Sciences, Ghent University, Ghent, Belgium; [3]Department of Biochemistry and Microbiology, Faculty of Sciences, Ghent University, Ghent, Belgium

§ To whom correspondence should be addressed:
Tel: +32 9 224 98 54
Email: sven.degroeve@vib-ugent.be
Address: Technologiepark 75, 9052 Ghent, Belgium

Abstract

Motivation: With a regulatory impact on numerous biological processes, protein phosphorylation is one of the most studied post-translational modifications. Effective computational methods that provide a sequence-based prediction of phosphorylation sites are desirable to guide functional experiments. Currently, the most successful methods train neural networks on amino acid composition representations. However, recently proposed protein language models provide enriched sequence representations that contain higher-level pattern information on which more performant phosphorylation site predictions may be based.

Results: We explored the applicability of protein language models to general phosphorylation site prediction. We found that training prediction models on top of protein language models yield a relative improvement of up to 68.4% in terms of area under the precision-recall curve over the state-of-the-art predictors. Model interpretation and model transferability experiments reveal that protease-specific cleavage patterns give rise to a protease-specific training bias. This can result in an overly optimistic estimation of phosphorylation site prediction performance, an important caveat in the application of advanced machine learning approaches to protein modification prediction based on proteomics data. We show that improving data quality by negative sample filtering using experimental metadata can mitigate this problem.

Availability and implementation: The PhosphoLingo tool, with trained models, code, models, datasets, and predictions are available at https://github.com/jasperzuallaert/PhosphoLingo.

Contact: sven.degroeve@vib-ugent.be

Supplementary information: Supplementary materials are available at *bioRxiv*.

1. Introduction

A post-translational modification (PTM) is a covalent and generally enzymatic modification of a protein following protein biosynthesis. The most frequently studied PTM is phosphorylation[1]. During phosphorylation, phosphate groups are transferred by enzymes from adenosine triphosphate to specific residues in a protein sequence, with serine, threonine or tyrosine residues being the most common targets. Kinases and phosphatases are involved in virtually all biological processes by targeting specific sites for modification.

The substrate preferences of kinases depend on the structural and physicochemical properties of the target substrate. Phosphorylation sites (from here onward referred to as P-sites) are most often found at accessible and disordered regions of a protein[2]. In addition, the amino acid sequence and structure at the P-site are important, with for example proline-directed kinases predominantly targeting ST residues with a neighbouring proline downstream[3].

Proteome-wide analysis of protein phosphorylation is currently accomplished through liquid chromatography-mass spectrometry (LC-MS) experiments[4–6]. Herein, proteins are digested into peptides with a protease that cleaves the sequence at specific sites. Typically, trypsin is used, which cleaves proteins after arginine and lysine residues. This results in a positive charge at the peptide C-terminal side-chain, which is advantageous for MS and MS/MS analysis. Alternative proteases that cleave at other residues[4] are applied to increase the proportion of a proteome that can be analysed using LC-MS, which has limitations with regard to peptide length, charge states, and other proteotypic characteristics.

High-throughput LC-MS comes with limitations, as it requires expensive and laborious analysis processes using expensive specialized equipment, and it does not at all guarantee the detection of all P-sites in a proteome. In fact, sample preparation and LC-MS workflows introduce biases in terms of which P-sites are detected, which are poorly characterized. Sensitive and accurate *in-silico* P-site predictions complement LC-MS analysis, helping in down-stream MS data interpretation and further guide *in-vitro* and *in-vivo* experiments[7].

Most P-site prediction approaches use neighbouring sequence data to train machine learning models to predict which serine (S), threonine (T) or tyrosine (Y) residues are likely to be phosphorylated[7–26]. In addition to general P-site prediction, some approaches also include kinase family-specific or kinase-specific predictions[8,11,12,14,16,17,20–22,24]. Over the last two decades, different machine learning algorithms have been applied, such as random forests[9,13,23], support vector machines[8,15,23], and gradient boosting trees[8]. These classical approaches require numerical feature vector representations engineered from the amino acids surrounding a candidate P-site, which we will refer to as the receptive field. Manually crafted input features for phosphorylation prediction models include physicochemical properties[13,15,20,25], information theory features[15,17,28], as well as additional information such as structural features (e.g., secondary structure)[9,15,23], functional features[9,23], and protein-protein interactions[16,24].

Recently, deep learning based model architectures such as convolutional neural networks (CNNs)[12,16–19,21,25] and long short-term memory networks (LSTMs)[19,25] have been shown to

advance P-site prediction. These models learn low-level feature representations from the sequence input data during training, alleviating the need for preliminary manual feature engineering.

In recent years, the success of language models in the natural language processing domain[27] have inspired the advent of protein language models (PLMs). PLMs are pre-trained models that yield enriched, structure-aware sequence representations, instead of merely encoding the amino acid composition of a receptive field in a protein[28–34]. They have demonstrated value in various tasks, such as few-shot contact map prediction[35], protein structure prediction[33], zero-shot mutation impact prediction[36], or phylogenetic relationship modelling[37].

In this paper, we investigate whether the enhanced information content of different PLM representations contributes to the prediction of P-sites and extensively compare the resulting prediction models to the standard one-hot encoding-based representations, as well as to publicly available state-of-the-art tools. Our results show that PLM representations contain information that is highly relevant for P-site prediction, thereby substantially increasing prediction performance. Furthermore, we applied model interpretation methods to visualize what information was considered relevant for the prediction of P-sites. Model interpretation reveals that these, in addition to local and global sequence patterns, also exploit specific LC-MS protease patterns that can result in overly optimistic prediction performance estimations for *de novo* phosphorylation site prediction. To restrain this prediction bias, we suggest a negative sample filtering method, which we validate on phosphorylation data from individual experiments with different proteases.

2. Methods

All proposed predictors consist of two main components: a representation of the receptive field in the protein and a neural network that takes this representation as input. This section introduces the considered representations, the neural network models, the benchmark methods, and the methods to interpret the predictions computed by these models.

2.1 Protein representations

As a baseline, the receptive field was represented as a one-hot encoding of the amino acids, i.e., no PLM was used. Then, multiple PLMs were considered in this study: ESM-1_{small} and ESM-1b[30], ESM-2[33], ProtT5-XL-U50[31], CARP-640M[38], Ankh-base[34], and Ankh-large[34] (see Suppl. Table 1 for more details).

2.2 Neural network architecture

To evaluate the different protein representations for phosphorylation prediction, we contructed a CNN architecture on top of the PLMs, which we then trained without fine-tuning the PLM further. Different approaches where the representations were further fine-tuned were considered, but ultimately discarded due to risks of overfitting on the huge models, as also suggested in literature[37]. An extensive hyperparameter search was done for each of the initial language models (ESM-1_{small}, ESM-1b, ProtT5-XL-U50), from which we then constructed a final hyperparameter setup that is used throughout all experiments. We chose to use a

single setup that works well for all datasets and representations to increase the robustness and reproducibility of our approach. Details are listed in Suppl. Table 2.

As schematically depicted in **Figure 1**, the CNN contains a series of convolutional blocks, with each convolutional block consisting of a convolutional layer followed by a regularization layer (either dropout or batch normalization), and finally a max pooling layer. The resulting activations are flattened, after which a fully-connected, a regularization layer, and an output layer are added. Furthermore, all convolutional and fully-connected layers are followed by a rectified linear unit (ReLU), and a sigmoid is added at the end to provide probabilities in output. The default one-hot encoded and PLM-based setups are schematically depicted in Suppl. Figs. 1-2.

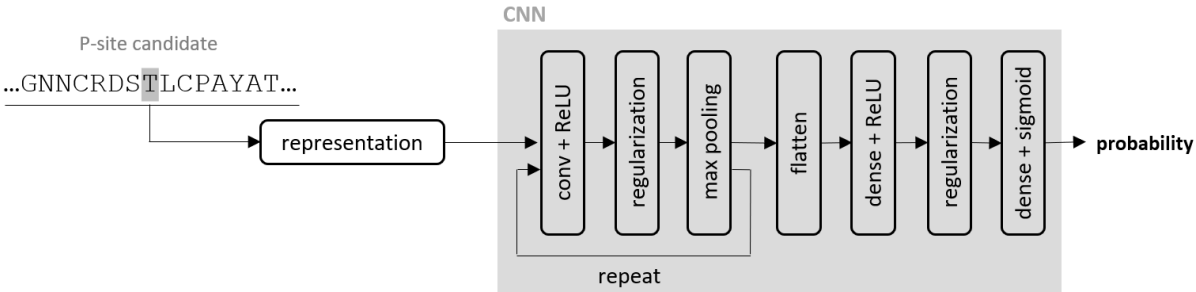


Figure 1. *The CNN blueprint, highlighting the different components that are used throughout this study.*

2.3 Benchmark methods

To evaluate the performance of the PLM-based predictors, we compare them to a one-hot encoded baseline, and to three state-of-the-art predictors: MusiteDeep[17], DeepPhos[18], and DeepPSP[12]. Code from their respective GitHub repositories was used to train their predictors from scratch on different data sources.

MusiteDeep implements two prediction modules: a CNN module enhanced with attention layers, and a capsule network module with dynamic routing, both built on a one-hot encoding with a receptive field of size 33 centered around the candidate P-site. Predicted probabilities for both modules are averaged to obtain the final P-site predictions.

In DeepPhos, three different densely connected CNN blocks are used with one-hot encoded inputs with receptive fields of size 15, 31 and 51, where each block consists of convolutional and intra-block concatenation layers. Blocks are combined using an inter-block concatenation layer, followed by a final fully connected layer.

Finally, DeepPSP uses two modules, each consisting of a convolutional block, a squeeze-and-excitation network, a bidirectional LSTM layer, and fully connected layers. The local module takes a one-hot encoding with a receptive field of length 51 in input, centered around the candidate P-site, while the global module uses a receptive field of length 2000 in input. Outputs of both modules are combined using a final fully connected layer.

2.4 Feature importance visualization

The fitted models are interpreted by applying SHapley Additive exPlanations[39] (SHAP) to compute importance scores for individual residues in the receptive field that estimate their

contribution to the predicted probability of a candidate P-site. For the visualization of all trained models we calculated the importance scores on the test set of the same dataset.

Importance scores are calculated using the DeepLiftSHAP implementation in the Captum[40] software package. SHAP values are computed relative to a reference baseline. For one-hot encoded models, an all-zero reference is used. For the PLM based representations, an average embedding baseline is constructed for each protein fragment. SHAP values are normalized on a dataset level, such that the absolute values of all importance scores in one sequence add up to 100 on average.

The importance scores, also referred to as saliency maps, are computed on an individual base per candidate P-site. These maps are then aggregated into an average saliency map that we visualize as a sequence logo centered around the candidate P-site, cropped to contain ten positions both up- and downstream.

3. Experimental Setup

3.1 Data

3.1.1 Data acquisition

Prediction models were trained and evaluated on datasets that were compiled via three different setups. Firstly, we used the *multi-source* data that was employed in previous P-site prediction research, which we downloaded from the DeepPSP[12] GitHub repository. This data was compiled from UniProtKB/SwissProt, dbPTM[41], Phospho.ELM[42], and PhosphoSitePlus[43], with negative annotations defined as all remaining STY residues in the proteins with positive annotations. The multi-source data is also split up in separate ST and Y datasets. An important caveat with this standardized way of assembling phosphorylation datasets is that the negative set also contains STY residues for which phosphorylation can never be observed due to limitations in the data acquisition setups of MS/MS analysis.

Secondly, *Scop3P* P-site datasets were obtained from our in-house and publicly accessible Scop3P[2,44] database, in which P-sites were annotated from large-scale reprocessing of proteomics experiments[45]. All annotations were matched with protein sequences from UniProtKB/SwissProt (version 2021-11), discarding proteins when one of its annotations did not match with the sequence data. Non-P-site annotations were defined as all remaining STY residues in the proteins in the reprocessed experiments for which no evidence was found in PhosphoSitePlus or dbPTM.

Thirdly, *single-protease* datasets were obtained from individual experiments that involved different proteases[4]. In addition to individual datasets for each protease, we also compiled a combined dataset with annotations from the individual sets, which we labelled as *multi-protease.* Only ST annotations were considered, as Y annotations were too few. As before, negative labels were first defined as all ST residues in the annotated proteins that were not positively labelled in any of the single-protease experiments, Scop3P, or multi-source datasets.

A comprehensive overview of the datasets is given in Table 1**Error! Reference source not found.**.

name	source	residues	# of proteins	# of pos	# of neg	pos:neg ratio
multi-source-ST-NPF	Guo21[12]	ST	13,599	184,375	981,620	1:5.3
multi-source-Y-NPF		Y	9,710	32,213	149,501	1:4.6
name	**source**	**residues**	**# of proteins**	**# of pos**	**# of neg**	**pos:neg ratio**
Scop3P-ST-NPF	Ramasamy22[2]	ST	8,754	54,395	681,975	1:12.5
Scop3P-Y-NPF		Y	8,754	4,884	125,306	1:25.7
name	**source**	**protease**	**# of proteins**	**# of pos**	**# of neg**	**pos:neg ratio**
AspN-NPF	Giansanti15[4]	AspN	1,816	3,987	171,003	1:42.9
Chymotrypsin-NPF	(ST only)	Chymotrypsin	1,739	3,476	167,754	1:48.3
GluC-NPF		GluC	1,899	4,280	184,426	1:43.1
LysC-NPF		LysC	1,718	4,133	166,669	1:40.3
Trypsin-NPF		Trypsin	3,255	9,096	298,628	1:32.8
multi-protease-NPF		multi-protease	4,728	18,028	424,527	1:23.5

Table 1. *An overview of datasets used in this study. Following the default negative selection criterions, no peptide filtering is applied yet, hence datasets are labelled with NPF here.*

3.1.2 Peptide-filtering

As mentioned in Section 3.1.1, negative annotations can then contain sites for which phosphorylation can never be observed with MS/MS experiments (e.g., sites in long stretches without peptide cleavage sites). To provide a higher quality dataset for both model training and evaluation, we propose an extra negative sample filtering step for datasets where experimental metadata is available. The filtering step rules out non-detectable sites from both training and evaluation by only including residues that occurred in matched peptide spectra (PSMs) in Scop3P. We will refer to this method as *peptide filtering (PF)* from here on, whereas we will label to the non-filtered datasets as *non-peptide filtered (NPF)*. An overview of peptide-filtered datasets is given in Table 2. Note that contrary to the non-peptide-filtered single- and multi-protease datasets, proteins without positive annotations can be present in this data, as they might have been matched with in the PSMs, hence increasing the number of proteins.

name	source	residues	# of proteins	# of pos	# of neg	pos:neg ratio
Scop3P -ST-PF	Ramasamy22[2]	ST	8,754	54,395	326,584	1:6.0
Scop3P -ST-PF		Y	8,754	4,884	60,468	1:12.4
name	**source**	**protease**	**# of proteins**	**# of pos**	**# of neg**	**pos:neg ratio**
AspN-PF	Giansanti15[4]	AspN	2,182	3,987	5,475	1:1.4
Chymotrypsin-PF	(ST only)	Chymotrypsin	2,097	3,476	5,199	1:1.5
GluC-PF		GluC	2,374	4,280	6,280	1:1.5
LysC-PF		LysC	2,389	4,133	5,106	1:1.2
Trypsin-PF		Trypsin	3,923	9,096	9,479	1:1.0
multi-protease-PF		multi-protease	5,867	18,028	25,341	1:1.4

Table 2. *An overview of peptide-filtered datasets used in this study. Note that the filtering step was not done for the multi-source data due to the lack of metadata.*

3.1.3 Dataset selection

Datasets were selected and split for three main purposes in this manuscript: (a) model performance evaluation and comparison, (b) feature importance visualization, and (c) model transfer evaluation.

(a) The comparison between one-hot encoded models, PLM-based models and existing predictors is done on the multi-source datasets, the Scop3P datasets (both NPF and PF), and the non-filtered multi-protease dataset. As PSMs in the single-protease datasets are obtained

from individual experiments, their coverage is lower than in the multi-experiment Scop3P dataset, and thus, the number of non-P-site annotations is reduced drastically when applying peptide-filtering. This results in a smaller and much more balanced dataset (**Table 2**), which impacts model training. Therefore, we did not consider these NPF datasets for comparison.

(b) The visualization of important features is done by first training predictors on the datasets used for evaluation, and then calculating SHAP values on a fixed evaluation set. For models trained on ST data, we picked the test split of the multi-protease-PF (1,775 P-sites, 2,331 non-P-sites) for this purpose, given improved data quality via peptide filtering, while keeping computational costs low at the same time. For models trained on Y data, we calculated SHAP values using the Scop3P-Y-PF test set. Note that differences with experiments using different test sets for generating SHAP values were negligible.

(c) The evaluation of model transferability was done on ST data only, to compare the Scop3P-ST-PF to the Scop3P-ST-NPF dataset, the multi-protease-NPF/PF dataset, and the single-protease-NPF/PF datasets. Evaluation was done on the NPF versions of the single- and multi-protease datasets.

3.1.4 Dataset splits

For the multi-source data, the original data split between training and test was kept[12], and training data was divided into an actual training part and a validation part. This is done for ST (Y) by randomly selecting 11,150 (7,999) proteins in the training set for training, and 738 (743) of all proteins for validation, while keeping the remaining 1,361 (968) proteins for testing. The Scop3P, single- and multi-protease datasets were randomly split by dividing proteins over training, validation and test sets via a 85/5/10 distribution. Finally, for the model transfer analysis, a cross-validation scheme was used, where each dataset set was split up into ten folds. Then, for each fold, the proteins included in the validation set were removed from the training set, of which 1/10th was used for early stopping. These schemes are illustrated in Suppl. Fig. 3a and 3c. All datasets are made available via GitHub, along with their training/validation/test splits.

3.2 Proteins are split into fragments

Due to computational complexity of transformer-based PLMs, all proteins were split into fragments of at most 512 amino acids before generating the representations. Fragments of one protein can be divided over multiple batches, but will always belong to the same training, validation, or test set to avoid data leakage.

Furthermore, as we wanted to avoid that a candidate P-site lies at the border between two fragments, thus implying that it can only use information from upstream or downstream residues, we allowed for overlapping fragments. Every 256 residues, a new fragment starts, and P-sites are coupled to the fragment with the closest centre. The full batch split setup is illustrated in Suppl. Fig. 4.

For increased efficiency, we reduced the number of times the PLMs were used to generate representations. Per epoch, one protein fragment was forwarded through the PLM only once, even if it holds multiple P-site candidates. As a result, optimization was performed for a varying number of positive and negative P-sites per batch.

3.3 Hardware and software used

Programmatical frameworks used include PyTorch and PyTorch Lightning for model development and training and evaluation logic, WandB[46] for experiment logging, and Captum[40] for calculating the SHAP values. Sequence logos were created using LogoMaker[47].

4. Results

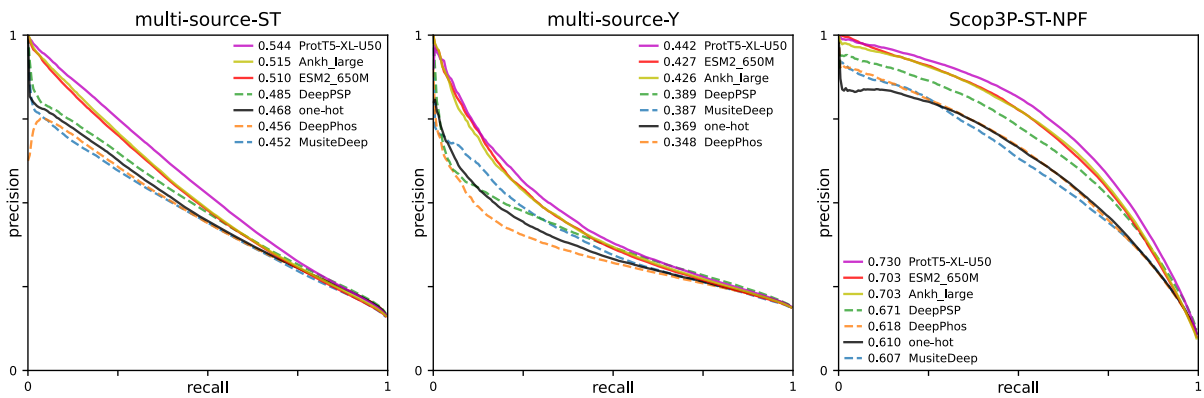
4.1 PLMs substantially improves P-site prediction performance

Figure 2 shows the prediction performance for the CNNs that were trained on top of different amino acid representations, as well as for the state-of-the-art predictors. To compare between predictors, we mainly looked at the area under the precision-recall curve (AUPRC) as a performance measure in order to produce meaningful results on the imbalanced datasets. More detailed results including other metrics are shown in Suppl. Table 3.

Remarkably, in the evaluation on six out of the seven datasets that were considered, the ProtT5-XL-U50 language model performed best, with improvements of 12.4% to 50.1% over the one-hot encoded CNN in terms of AUPRC, and improvements of 2.0% to 4.6% over the next best PLM-based model. Only on multi-source-Y-NPF ESM-1b performed roughly as good (+0.2%). Furthermore, the ProtT5-XL-U50 outperformed the state-of-the-art by 6.5% to 68.4% on all datasets considered.

To put the prediction performance gains into perspective, we looked at the precision of the predictor that correctly classifies four out of five true P-sites (i.e., when recall equals 0.8). Compared to the best existing model (DeepPSP) the precision for the ProtT5-XL-U50 models increases by up to 9.6%, from 0.727 to 0.797 on the Scop3P-ST-NPF dataset, or by up to 6.8%, from 0.792 to 0.846 when only considering observable P-site candidates in the Scop3P-ST-PF dataset. On both Scop3P-Y datasets, increase of up to 94.4% is observed, from 0.304 to 0.523 (NPF) and 0.305 to 0.593 (PF) respectively.

Furthermore, when comparing the performance on the Scop3P-ST-NPF and -PF datasets, the AUROC, which is more robust to dataset imbalance than the AUPRC, drops consistently for all representations by 1.6% to 3.1% when applying peptide filtering. For Scop3P-Y-NPF and -PF datasets, a drop in AUROC of 2.4% to 8.3% occurs. This suggests an increased level of complexity for the prediction task, as a subset of easily predictable targets is no longer considered in the evaluation. Further rationale for utilizing peptide filtering is discussed in Section 4.2.



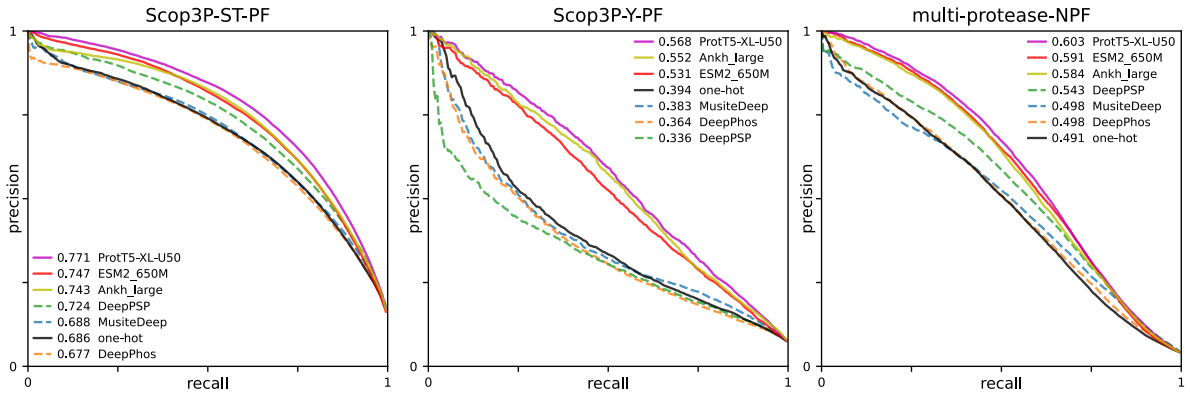


Figure 2. *Precision-recall curves for the ProtT5-XL-U50, ESM-2$_{650M}$ and Ankh_large PLMs, a one-hot encoding, and existing predictors. Average curves over 10 runs (per dataset) are depicted. In the legend, the mean AUPRC is reported.*

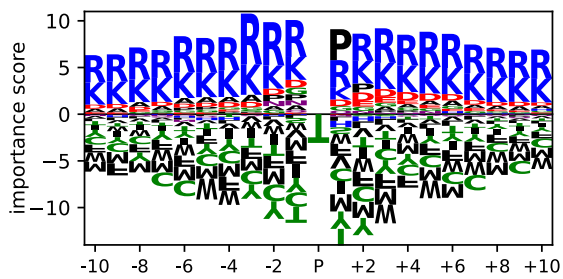
4.2 Feature importance analysis

Visualizations of P-site predictors show discriminative features from different kinases

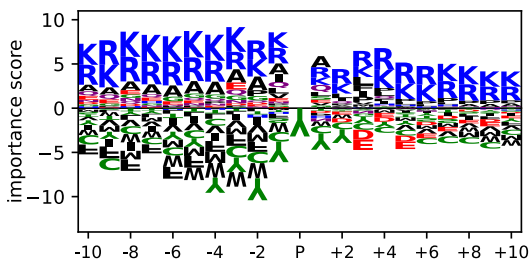
We computed SHAP values to estimate the average importance of individual residues in the proximity of candidate P-sites. Results for models trained on the Scop3P datasets using the ProtT5-XL-U50 PLM are visualized in Figure 3, and models using a one-hot encoding are visualized in Suppl. Fig. 6.

For the ST models, the most notable residue is proline (P) at position P+1, which is known to be a strong signal towards a P-site prediction, as several kinases in the CMGC group are known to be proline-directed kinases[3], such as those belonging to the MAPK and CDK kinase families (Suppl. Fig. 5a-b). Other favourable residues in the ST visualization are especially highlighted in the one-hot encoded visualizations in Suppl. Fig. 6), with arginine (R) at positions P-3 and P-2, as often seen for kinases in the PKa family (Suppl. Fig. 5c), and aspartic (D) and glutamic acid (E) on the first positions downstream of the candidate site, as often seen for kinases in the CK2 family (Suppl. Fig. 5d).

ProtT5-XL-U50, trained on **Scop3P-ST-NPF** data



ProtT5-XL-U50, trained on **Scop3P-Y-NPF** data



ProtT5-XL-U50, trained on **Scop3P-ST-PF** data

ProtT5-XL-U50, trained on **Scop3P-Y-PF** data

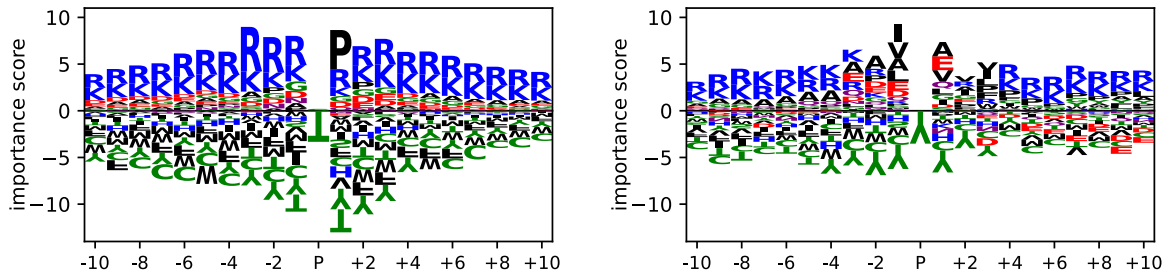


Figure 3. *Average importance scores per position calculated using DeepLiftSHAP, for the ProtT5-XL-U50 prediction model setup, on the Scop3P datasets. Visualizations are cropped to show the twenty positions surrounding the candidate P-site.*

CNNs trained for P-site prediction disfavour inaccessible P-sites

The average importance scores in Figure 3 and Suppl. Fig. 6 indicate that the models disfavour hydrophobic amino acids in the proximity of the candidate P-site. On average, for the one-hot encoded model trained on Scop3P-ST-PF data, W/C/Y/F/M are considered least favourable when in the proximity of a P-site, while for the PLM, this is the case for Y/C/F/W/M. As P-sites need to be reached by kinases for phosphorylation to take place, the majority of P-sites are found at the solvent exposed area of the protein (more than 80%, while less than 2% are located in buried regions)[2]. Hydrophobic (A/F/I/L/M/V/Y/W) and C residues[48] are more abundant in buried regions[48], suggesting that the models implicitly learn that P-sites occur less frequently in buried regions.

This is confirmed by considering prediction distributions for candidate P-sites in buried, interface and accessible regions. Alphafold[49,50] was used to obtain protein structures for all proteins in the dataset that were shorter than 2700 amino acids. Next, DSSP[51] was used to calculate the solvent accessibility and secondary structure for all candidate P-sites. Figure 4 shows the distribution of predicted probabilities for positively labelled P-sites in the evaluation set. Predicted scores of P-sites located in buried regions tend to be lower than those in interface regions, and much lower than those in accessible regions. Similar conclusions can be drawn from secondary structure analysis, where P-sites in random coil regions generally score higher than those in helices and sheets, which is in line with statistical analysis of phosphorylation data[2].

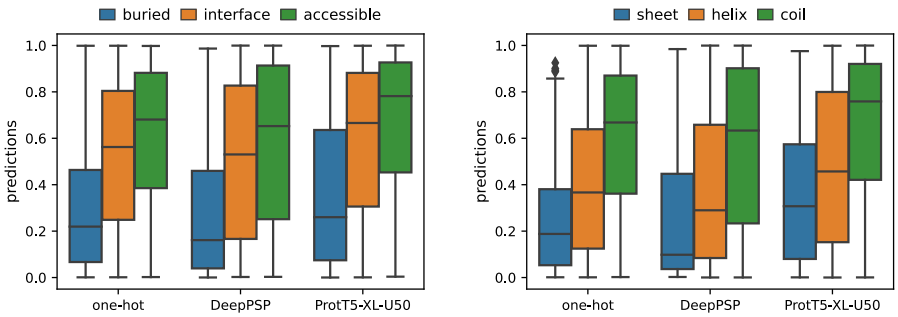


Figure 4. *Distribution of predicted probabilities for positively labelled P-sites in the Scop3P-ST-PF test set, per model, divided by surface accessibility and secondary structure.*

Datasets from experiments with individual proteases indicate biased learning

Most notably, **Figure 3** shows a high sensitivity to R and K residues in the proximity of the candidate P-sites. These residues correspond to tryptic cleavage sites. With trypsin being the most prevalent protease in bottom-up proteomics, this observation strongly points to a protease-related bias in phosphorylation datasets. We hypothesized that the models indeed learn an enriched presence of tryptic cleavage sites near P-sites.

To test this hypothesis, we performed experiments on the single-protease datasets, where we trained predictors on data compiled from experiments that applied different proteases. **Figure 5** shows that models trained on the single-protease datasets demonstrate consistent enrichment of their corresponding cleavage sites. The amino acids with highest average importance within the twenty positions surrounding the P-sites are D for AspN, F for chymotrypsin, E for GluC, K for LysC, and R for trypsin, consistent with the most frequent cleavage sites of the respective proteases. Similar results for one-hot encoded models are shown in Suppl. Fig. 7.

Additionally, the visualizations in Figure 3 suggest a decrease of the bias levels when applying peptide filtering. We compare the average importance score of R and K residues in the proximity of the P-site (disregarding the 7 central amino acids containing other strong signals) to the average importance of the most important residue, P at P+1. For the Scop3P-ST dataset, we observe that the average scores for R and K are 79.9% and 100.3% of the average score for P at P+1, and that this drops to 40.6% and 49.4% respectively when applying peptide filtering. This indicates that the models are less inclined to learn the protease-induced bias, which we highlight further in Section 4.3.

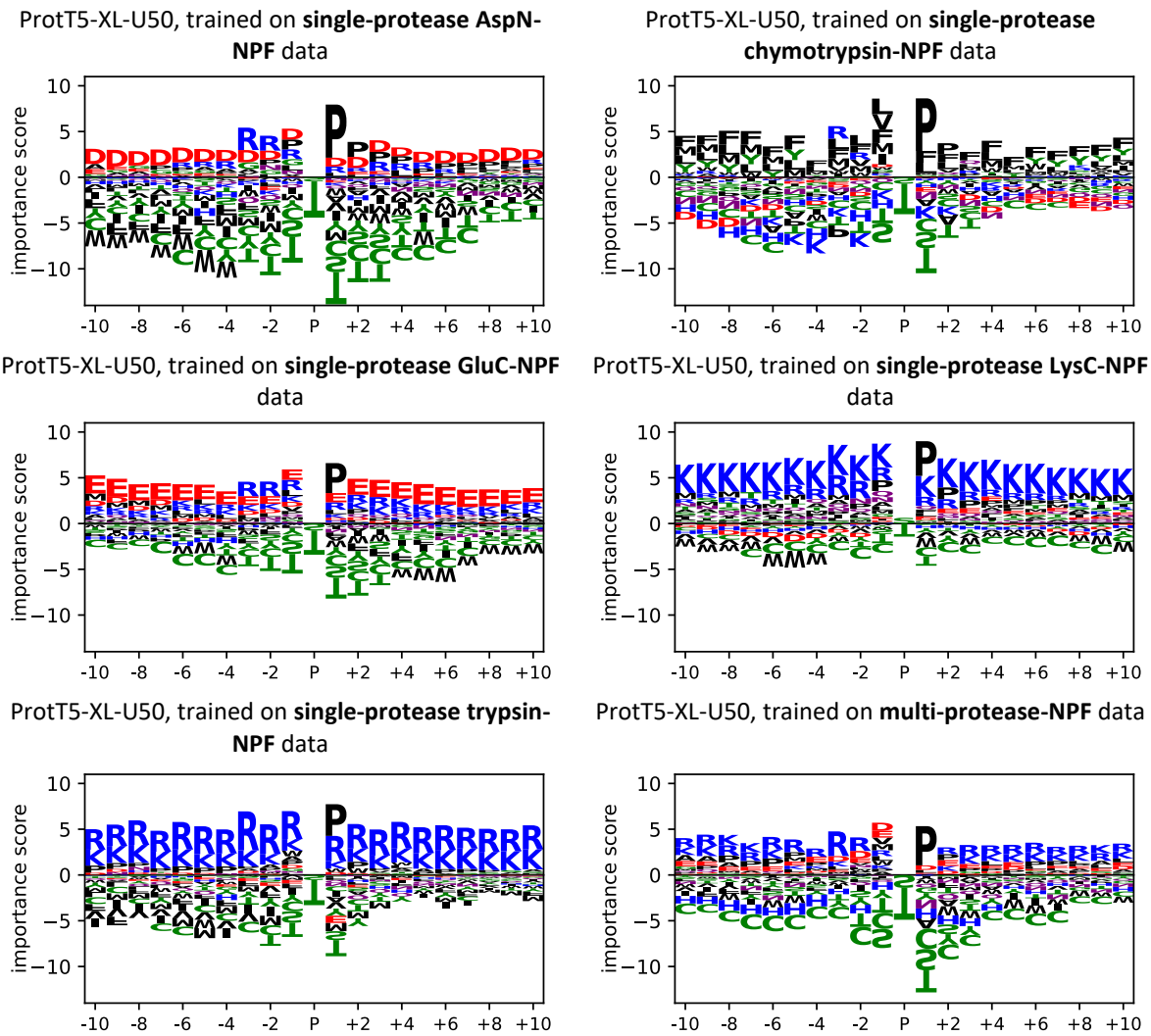


Figure 5. *Average importance scores per position, calculated using DeepLiftSHAP, for the ProtT5-XL-U50 prediction model setup, on the single- and multi-protease datasets. Visualizations are cropped to show the twenty positions surrounding the candidate P-site.*

4.3 Model transferability analysis

To further investigate the observed protease-specific bias, we evaluated predictors that were trained on different data sources, across different test sets. Concise results for the PLM-based models are shown in Figure 6, with additional results in Suppl. Fig. 8.

ProtT5-XL-U50 models trained on the Scop3P-ST-NPF dataset are consistently outperformed by the respective single-protease trained models with a relative AUPRC increase of 15.9% to 21.0%, for the AspN, chymotrypsin, and GluC data. For LysC data, where cleavage occurred on lysine residues (partly sharing the bias with the trypsin-based Scop3P data), the performance difference was smaller (7.3%). For trypsin data, the models trained on Scop3P data almost reached the same level (0.5%), as the former is exclusively constructed from experiments using trypsin digestion.

Additionally, we compared the models trained on the Scop3P-ST-NPF to models trained on Scop3P-ST-PF data, to investigate the effect of the endeavoured quality improvement in the filtered data. We observe that the trypsin bias learned by the models is reduced, as

performance improves on AspN, chymotrypsin, and GluC data by 2.1% to 7.9%. Simultaneously, the performance drops for the LysC and trypsin datasets by 6.1% and 4.8% respectively, as part of the - in this case - beneficial bias is reduced.

Models trained on multi-protease-NPF data further improve on the Scop3P-ST-PF-based models by 3.2% to 15.3%. Compared to the models trained on single-protease data, the difference in performance is more limited, between -4.4% and +3.9%. Overall, this suggests that models trained on data obtained from multiple proteases generalize better. Extra model transfer results can be found in Suppl. Fig. 8, where we can also see that models trained on the multi-source data generalize worse than the Scop3P models.

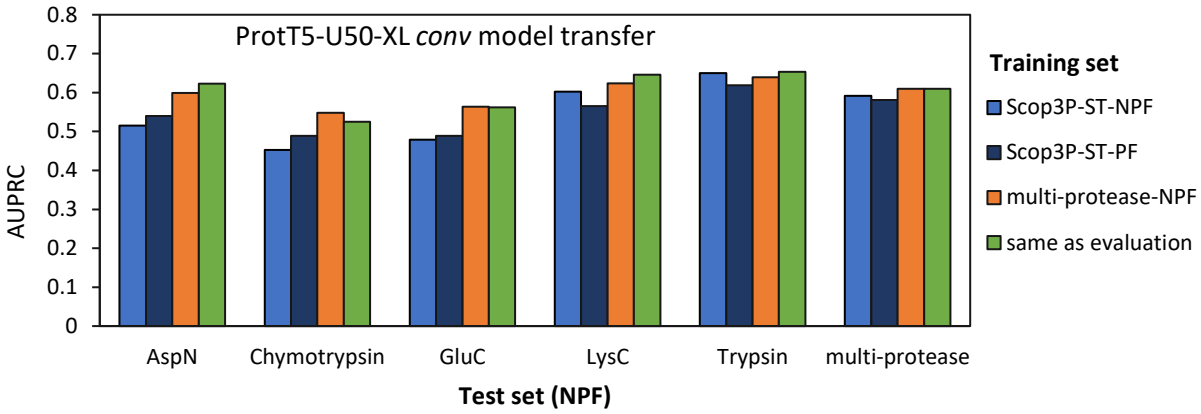


Figure 6. *Evaluation on non-peptide-filtered single-protease and multi-protease datasets (x-axis), when training on a different data source (legend). Predictions are computed from ten folds of the evaluation set, where training is performed on either Scop3P-ST data, multi-protease data, or on the dataset matching the data source on the x-axis. Note that proteins present in the test fold are always omitted from training. The average AUPRC over these ten test folds is shown, for ProtT5-XL-U50 conv models.*

5. Discussion

In this study, we examined the potential of fine-tuning PLMs to predict phosphorylation events from protein sequence. We propose a novel approach that utilizes CNNs on top of protein representations generated by the language models. Our results show a significant improvement in prediction performance compared to current state-of-the-art neural network architectures that rely on one-hot encoded representations. This improvement was observed across various datasets obtained from diverse sources, indicating that PLMs offer informative protein sequence representations that enhance phosphorylation prediction accuracy.

Upon inspection of feature importance, we found evidence supporting known kinase family-specific residue patterns, such as proline at P+1, arginine at P-3 and P-2, and aspartic and glutamic acid at the first positions downstream of the candidate P-site. These findings suggest that the proposed models recognize the characteristic features of an 'average' P-site across all kinases. However, it also indicates that rare kinases with unique motifs might not be effectively captured during training, thereby limiting the efficacy of a general phosphorylation prediction model.

Further analysis incorporating protein 3-D structure revealed that predicted probabilities for P-sites were generally higher within accessible regions than within buried regions, and higher

within random coil regions compared to those within alpha helices or beta sheets. Our analysis also reveals a strong preference of the prediction models towards amino acids that are targeted by proteases for cleavage, particularly in the vicinity of candidate P-sites. We attribute this pattern to the inherent biases present in bottom-up LC-MS/MS proteomics experiments, rather than being a phosphorylation-specific pattern. The bias arises from at least two sources.

Firstly, in the case of trypsin-based digestion, candidate P-sites in long stretches of protein sequence lacking lysine or arginine residues cannot be detected in LC-MS/MS experiments. For datasets compiled without applying our proposed peptide filtering, all candidates in these stretches will be labelled as non-P-sites. Secondly, a higher frequency of missed cleavages near P-sites[4,6] results in more arginine- and lysine-enriched peptides that would typically be cleaved into fragments that are too short for MS/MS analysis were they not phosphorylated. A prediction model can exploit these technical biases by utilizing the lack of arginine and lysine around non-P-sites to steer its prediction to a lower probability, and by increasing the P-site probability for occurrences within arginine- and lysine-rich regions. Thus, general phosphorylation prediction models do not just learn P-site fingerprints to predict the likeliness of phosphorylation, but also the likeliness of detection with current experimental setups.

The technical bias is further evidenced by the poor generalization performance of prediction models when transferred between datasets compiled from experiments using different proteases for digestion. Models trained on datasets obtained using one protease show reduced accuracy when evaluated on datasets obtained using a different protease.

To partially address this bias, we investigated a peptide filtering method to improve data quality by removing negatively labelled data points that were not present in the observed peptides during MS/MS analysis. In addition to model evaluation on data with fewer false negative data points, models trained on filtered data generalized better when transferring to phosphorylation datasets derived using different proteases for digestion. However, the proposed filtering method does not resolve all potential sources of dataset bias, such as the increased frequency of missed cleavages near P-sites, which require further investigation in future research.

As a final remark, the application for which P-site predictions are done steers the choice of training data. When doing *de novo* predictions, protease bias is detrimental, and the training data should be as heterogeneous as possible. However, when predicting detectable phosphorylation for LC-MS/MS data interpretation purposes, a model trained on data acquired using the same protease should be used.

We have released the PhosphoLingo tool, which contains the most precise PLM-based P-site predictors in this study on our GitHub repository (https://github.com/jasperzuallaert/PhosphoLingo), trained on the Scop3P-ST-PF and Scop3P-Y-PF datasets. We also provide a user-friendly framework to replicate the results reported in this manuscript, and to train and evaluate models on additional phosphorylation

446 datasets or other post-translational modifications. Additionally, we furnish predictions for all
447 S, T and Y residues in the human proteome.

448 # Acknowledgements

449 ## Funding

450 This work was supported by the Research Foundation – Flanders (FWO) [1274021N for J.Z.]; and the Vlaams
451 Agentschap Innoveren en Ondernemen [HBC.2020.2205 for R.B.]. Research at our lab is core-funded by the VIB
452 - Center for Medical Biotechnology and Ghent University.

453 Conflict of interest: none declared

454 # References

455 1. Ramazi, S. & Zahiri, J. Post-translational modifications in proteins: resources, tools and
456 prediction methods. *Database* **2021**, (2021).
457 2. Ramasamy, P., Vandermarliere, E., Vranken, W. & Martens, L. Panoramic Perspective on
458 Human Phosphosites. *Journal of Proteome Research* **21**, 1894–1915 (2022).
459 3. Pinna, L. A. & Ruzzene, M. How do protein kinases recognize their substrates?
460 *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* **1314**, 191–225 (1996).
461 4. Giansanti, P. *et al.* An Augmented Multiple-Protease-Based Human Phosphopeptide
462 Atlas. *Cell Reports* **11**, 1834–1843 (2015).
463 5. Dephoure, N., Gould, K. L., Gygi, S. P. & Kellogg, D. R. Mapping and analysis of
464 phosphorylation sites: A quick guide for cell biologists. *Molecular Biology of the Cell* **24**,
465 535–542 (2013).
466 6. Tsiatsiani, L., Heck, A. J. R., Heck, A. J. R. & Bijvoet, P. Proteomics beyond trypsin. *The*
467 *FEBS Journal* **282**, 2612–2626 (2015).
468 7. Trost, B. & Kusalik, A. Computational prediction of eukaryotic phosphorylation sites.
469 *Bioinformatics* **27**, 2927–2935 (2011).
470 8. Ma, R. *et al.* KinasePhos 3.0: Redesign and Expansion of the Prediction on Kinase-specific
471 Phosphorylation Sites. *bioRxiv preprint* (2021) doi:10.1101/2021.11.02.467032.
472 9. Song, J. *et al.* PhosphoPredict: A bioinformatics tool for prediction of human kinase-
473 specific phosphorylation substrates and sites by integrating heterogeneous feature
474 selection. *Scientific Reports* **7**, (2017).
475 10. Xu, Y., Song, J., Wilson, C. & Whisstock, J. C. PhosContext2vec: A distributed
476 representation of residue-level sequence contexts and its application to general and
477 kinase-specific phosphorylation site prediction. *Scientific Reports* **8**, (2018).
478 11. Wang, C. *et al.* GPS 5.0: An Update on the Prediction of Kinase-specific Phosphorylation
479 Sites in Proteins. *Genomics, Proteomics and Bioinformatics* **18**, 72–80 (2020).
480 12. Guo, L. *et al.* DeepPSP: A Global-Local Information-Based Deep Neural Network for the
481 Prediction of Protein Phosphorylation Sites. *Journal of Proteome Research* **20**, 346–356
482 (2021).

13. Wei, L., Xing, P., Tang, J. & Zou, Q. PhosPred-RF: A Novel Sequence-Based Predictor for Phosphorylation Sites Using Sequential Information only. *IEEE Transactions on Nanobioscience* **16**, 240–247 (2017).
14. Li, F. *et al.* Quokka: A comprehensive tool for rapid and accurate prediction of kinase family-specific phosphorylation sites in the human proteome. *Bioinformatics* **34**, 4223–4231 (2018).
15. Dou, Y., Yao, B. & Zhang, C. PhosphoSVM: Prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine. *Amino Acids* **46**, 1459–1469 (2014).
16. Yang, H., Wang, M., Liu, X., Zhao, X.-M. & Li, A. PhosIDN: an integrated deep neural network for improving protein phosphorylation site prediction by combining sequence and protein–protein interaction information. *Bioinformatics* **37**, 4668–4676 (2021).
17. Wang, D. *et al.* MusiteDeep: A deep-learning based webserver for protein post-translational modification site prediction and visualization. *Nucleic Acids Research* **48**, W140–W146 (2021).
18. Luo, F., Wang, M., Liu, Y., Zhao, X. M. & Li, A. DeepPhos: Prediction of protein phosphorylation sites with deep learning. *Bioinformatics* **35**, 2766–2773 (2019).
19. Lv, H., Dao, F.-Y., Zulfiqar, H. & Lin, H. DeepIPs: comprehensive assessment and computational identification of phosphorylation sites of SARS-CoV-2 infection using a deep learning-based approach. *Briefings in Bioinformatics* (2021) doi:10.1093/bib/bbab244.
20. Deznabi, I., Arabaci, B., Koyuturk, M. & Tastan, O. DeepKinZero: Zero-shot learning for predicting kinase-phosphosite associations involving understudied kinases. *Bioinformatics* **36**, 3652–3661 (2020).
21. Kirchoff, K. & Gomez, S. EMBER: multi-label prediction of kinase-substrate phosphorylation events through deep learning. *Bioinformatics* **38**, (2022).
22. Blom, N., Sicheritz-Pontén, T., Gupta, R., Gammeltoft, S. & Brunak, S. Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence. *Proteomics* **4**, 1633–1649 (2004).
23. Jamal, S., Ali, W., Nagpal, P., Grover, A. & Grover, S. Predicting phosphorylation sites using machine learning by integrating the sequence, structure, and functional information of proteins. *Journal of Translational Medicine* **19**, (2021).
24. Ma, H., Li, G. & Su, Z. KSP: An integrated method for predicting catalyzing kinases of phosphorylation sites in proteins. *BMC Genomics* **21**, (2020).
25. Thapa, N. *et al.* A deep learning based approach for prediction of Chlamydomonas reinhardtii phosphorylation sites. *Scientific Reports* **11**, (2021).
26. Ismail, H. D., Jones, A., Kim, J. H., Newman, R. H. & Kc, D. B. RF-Phos: A Novel General Phosphorylation Site Prediction Tool Based on Random Forest. *BioMed Research International* **2016**, (2016).
27. Brown, T. B. *et al.* Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* **33**, 1877--1901 (2020).

28. Alley, E. C., Khimulya, G., Biswas, S., Alquraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods* **16**, 1315–1322 (2019).
29. Rao, R. *et al.* Evaluating Protein Transfer Learning with TAPE. *Advances in Neural Information Processing Systems* **32**, 9689--9701 (2019).
30. Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences of the United States of America* **118**, (2021).
31. Elnaggar, A. *et al.* ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 1–1 (2021).
32. Strodthoff, N., Wagner, P., Wenzel, M. & Samek, W. UDSMProt: universal deep sequence models for protein classification. *Bioinformatics* **36**, 2401–2409 (2020).
33. Lin, Z. *et al.* Evolutionary-scale prediction of atomic level protein structure with a language model. 2022.07.20.500902 Preprint at https://doi.org/10.1101/2022.07.20.500902 (2022).
34. Elnaggar, A. *et al.* Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling. Preprint at https://doi.org/10.48550/arXiv.2301.06568 (2023).
35. Rao, R., Meier, J., Sercu, T., Ovchinnikov, S. & Rives, A. *Transformer protein language models are unsupervised structure learners*. http://biorxiv.org/lookup/doi/10.1101/2020.12.15.422761 (2020) doi:10.1101/2020.12.15.422761.
36. Meier, J. *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. in *Advances in Neural Information Processing Systems* vol. 34 29287–29303 (2021).
37. Detlefsen, N. S., Hauberg, S. & Boomsma, W. Learning meaningful representations of protein sequences. *Nat Commun* **13**, 1914 (2022).
38. Yang, K. K., Lu, A. X. & Fusi, N. *Convolutions are competitive with transformers for protein sequence pretraining*. http://biorxiv.org/lookup/doi/10.1101/2022.05.19.492714 (2022) doi:10.1101/2022.05.19.492714.
39. Lundberg, S. M. & Lee, S. A unified approach in interpreting model predictions. *Advances in Neural Information Processing Systems* 4765–4774 (2017).
40. Kokhlikyan, N. *et al.* Captum: A unified and generic model interpretability library for PyTorch. *arXiv preprint* (2020) doi:10.48550/arXiv.2009.07896.
41. Li, Z. *et al.* dbPTM in 2022: an updated database for exploring regulatory networks and functional associations of protein post-translational modifications. *Nucleic Acids Research* **50**, D471–D479 (2022).
42. Dinkel, H. *et al.* Phospho.ELM: a database of phosphorylation sites—update 2011. *Nucleic Acids Research* **39**, D261–D267 (2011).

43. Hornbeck, P. V. *et al.* PhosphoSitePlus, 2014: mutations, PTMs and recalibrations. *Nucleic acids research* **43**, D512–D520 (2015).
44. Ramasamy, P. *et al.* Scop3P: A Comprehensive Resource of Human Phosphosites within Their Full Context. *Journal of Proteome Research* **19**, 3478–3486 (2020).
45. Perez-Riverol, Y. *et al.* The PRIDE database resources in 2022: a hub for mass spectrometry-based proteomics evidences. *Nucleic Acids Research* **50**, D543–D552 (2022).
46. Biewald, L. Experiment Tracking with Weights and Biases. (2020).
47. Tareen, A. & Kinney, J. B. Logomaker: Beautiful sequence logos in python. *bioRxiv preprint* (2019) doi:10.1101/635029.
48. Gowder, S. M., Chatterjee, J., Chaudhuri, T., Paul, K. & Wang, J. Prediction and Analysis of Surface Hydrophobic Residues in Tertiary Structure of Proteins. *The Scientific World Journal* (2014) doi:10.1155/2014/971258.
49. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature 2021 596:7873* **596**, 583–589 (2021).
50. Varadi, M. *et al.* AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research* **50**, D439–D444 (2022).
51. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**, 2577–2637 (1983).