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

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 probable phosphorylation sites are desirable to guide functional experiments or constrain search spaces for proteomics-based experimental pipelines. Currently, the most successful methods train deep learning models on amino acid composition representations. However, recently proposed protein language models provide enriched sequence representations that may 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 between 13.4% and 63.3% in terms of area under the precision-recall curve over the state-of-the-art predictors. Advanced model interpretation and model transferability experiments reveal that across models, protease-specific cleavage patterns give rise to a protease-specific training bias that results 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.

Availability and implementation: PhosphoLingo code, datasets, and predictions are available at https://github.com/jasperzuallaert/PhosphoLingo.

Contact: jasper.zuallaert@vib-ugent.be, 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 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 S/T 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, etc.

High-throughput LC-MS comes with its 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[28–32]. They have demonstrated value in various tasks, such as few-shot contact map prediction[33], zero-shot mutation impact prediction[34], or phylogenetic relationship modelling[35].

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 the publicly available state-of-the-art implementations. 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. To restrain this prediction bias, we provide an accurate PLM-based P-site predictor that is trained on a large multi-source P-site dataset, also including data from individual protease experiments.

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: the smallest model ESM-1_{small} has 43 million parameters, the larger model ESM-1b has 650 million parameters, and the largest model ProtT5-XL-U50 has three billion parameters[30,31] (Suppl. Table 1).

2.2 Neural network architectures

We evaluated two types of architectures on top of the protein representations:

(a) A simple dense layer (i.e., a logistic regression model), with a receptive field of 1. This means that it will only make use of the individual amino acid representation of the candidate P-site. We refer to this setup as the ***linear*** setup.
(b) A CNN with a receptive field that is predetermined in a hyperparameter search. The same architectural blueprint is used throughout the experiments. As shown in Figure 1, it 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. If only the CNN is trained, with the weights of the PLM fixed, we refer to this setup as the ***conv*** setup. If we allow for simultaneous fine-tuning of the PLM, we refer to it as the ***full*** setup.

The default one-hot encoded setup, as well as the three PLM based setups are schematically depicted in Suppl. Figs. 1-4. Furthermore, an extensive overview of hyperparameters and their ranges is given in Suppl. Table 2.

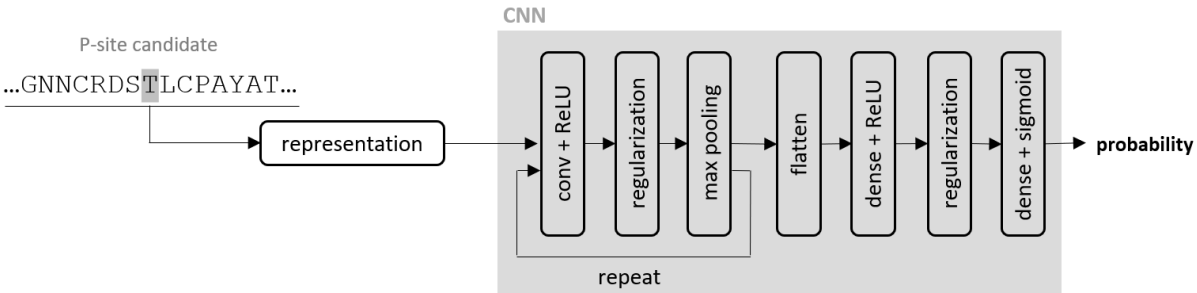

Figure 1. The CNN blueprint, highlighting the different components that are used throughout this study.

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[36] (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 in an independent dataset not used during training.

Importance scores are calculated using the DeepLiftSHAP implementation in the Captum[37] software package, at the protein representation level. 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

Prediction models were trained and evaluated on datasets that were compiled from three sources. The *Guo21* data was employed in previous P-site prediction research and was downloaded from the DeepPSP[12] GitHub repository. This data was compiled from UniProtKB/SwissProt, dbPTM, Phospho.ELM[38], and PhosphoSitePlus, with negative annotations defined as all remaining S/T/Y residues in the considered proteins. An important caveat with this standardized way of assembling phosphorylation datasets is that the negative set also contains S/T/Y residues for which phosphorylation can never be observed due to limitations in the data acquisition setups via MS/MS analysis.

The *Ramasamy22* data was obtained from our in-house and publicly accessible Scop3P[2,39] database, in which P-sites were annotated from large-scale re-processing of proteomics experiments[40] while requiring additional evidence from PhosphoSitePlus[41] and dbPTM[42]. 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 S/T/Y residues in the proteins for which no evidence was found in PhosphositePlus or dbPTM.

The *Giansanti15* data was obtained from experiments that involved different proteases[4]. Next to a multi-protease dataset containing all P-site annotations, we also compiled individual datasets for each protease. Only S/T annotations were considered, as Y annotations were too few. Negative labels were defined as all S/T residues in the annotated proteins that were not positively labelled in any of the single-protease experiment, Ramasamy22, or Guo21 datasets.

A comprehensive overview of the datasets is given in Table 1. All datasets are made available via GitHub, along with their training/validation/test splits.

dataset	residues	# of proteins	# of pos	# of neg
Ramasamy22[2]	S/T	8,759	54,408	682,214
	Y	8,588	4,884	125,331
Guo21[12]	S/T	13,599	184,375	981,620
	Y	9,710	32,213	149,501
dataset	**protease**	**# of proteins**	**# of pos**	**# of neg**
Giansanti15[4]	AspN	1,719	3,818	164,044
(S/T only)	Chymotrypsin	1,675	3,329	163,351
	GluC	1,827	4,103	178,817
	LysC	1,637	3,939	160,800
	Trypsin	3,112	8,713	290,746
	multi-protease	4,498	17,201	411,321

Table 1. An overview of datasets used in this study.

For the Guo21 data, the original data split between training and test was kept, and training data was divided into an actual training part and a validation part. This is done for S/T (Y) by randomly selecting 11,150 (7999) 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 Ramasamy22 and Giansanti15 data 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. Figure 5a and 5c.

3.2 Hyperparameters

Our setup includes a couple of fixed hyperparameters (Suppl. Table 2). Prediction models were optimized via a cross-entropy loss, using an AdamW optimizer. The loss on the held-out validation set is measured every 1/4th epoch, for a total of up to ten epochs. Early stopping was applied when the loss did not increase for five times, i.e. every 1.25 epochs, and the model with the lowest loss was used for evaluation.

In addition, for the one-hot encoding and each of the PLM + architecture setup (linear, conv only, full) combinations, an extensive hyperparameter search was performed on the Ramasamy22-S/T dataset. Hyperparameters were evaluated on the validation set, thus not hampering any test set independence in the results. As such, early stopping was done using a randomly held out 1/20th part of the training set. This scheme is illustrated in Suppl. Figure 5b.

A Bayesian optimization was run using the Weights and Biases framework[43]. Then, for each representation + setup combination, the top three scores were collected, and their architectures were collected for 10 more runs. Finally, the setup with the highest average AUPRC on the validation set was selected.

A detailed list of the hyperparameter ranges is given in Suppl. Table 2. The final hyperparameters are listed in Suppl. Table 3.

3.3 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. Figure 6.

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.

3.4 Hardware and software used

Programmatical frameworks used include PyTorch and PyTorch Lightning for model development and training and evaluation logic, WandB[43] for experiment logging and hyperparameter sweeps, and Captum[37] for calculating the SHAP values. Sequence logos were created using LogoMaker[44].

4. Results

4.1 Fine-tuning PLMs substantially improves P-site prediction performance

Figure 2 shows the prediction performance for the different amino acid representations and their corresponding optimized model architectures, as well as 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 4.

The *linear* setup performs worse than the baseline one-hot CNNs on the Ramasamy22-S/T dataset (with a drop in performance of 4.6% for the best performing PLM). Still, it performs around the same level on Guo21-ST (-1.1% to +0.2%) and Giansanti15 multi-protease data (-0.6% to +2.8%), and even slightly better on the Y datasets (-1% to +12.8%). Recall that the linear setup fits a simple logistic regression model on top of the PLM representation of only the candidate P-site amino acid. As such, these results strongly indicate that a PLM provides a rich source of contextual information that is particularly valuable for P-site prediction.

In the *conv* setup, training a CNN on top of a PLM representation of an extended receptive field substantially improves performance by up to 17.6% (Ramasamy22-S/T), 45.6% (Ramasamy22-Y), 18.6% (Guo21-S/T), 23.7% (Guo21-Y), and 16.1% (Giansanti15 multi-protease), compared to the baseline one-hot encoded architecture. A CNN clearly benefits from the optimized PLM representations.

Training a CNN while simultaneously fine-tuning the PLM that computes the representation in the *full* setup further improves performance for both the ESM-1b and ProtT5-XL-U50 PLM models on Ramasamy22-S/T and Giansanti15 multi-protease data (between +0.6% and +9.7%

compared to the *conv* setup), but underperforms slightly for the other datasets (between -7.8% and -0.1%), with one exception where variance in results was high (ESM-1b *conv*) on Ramasamy22-Y.

Both *conv* and *full* setups consistently outperform the state-of-the-art predictors. The top performing PLM-based setup shows a consistent increase of the AUPRC on the Ramasamy22-S/T data of 13.4% (compared to DeepPSP), on Ramasamy22-Y data of 63.4% (MusiteDeep), on Guo21-S/T data of 14.2% (DeepPSP), on Guo21-Y data of 19.5% (DeepPSP), and on Giansanti15 multi-protease data of 13.3% (DeepPSP).

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 model (DeepPSP) the precision increases by up to 13.7%, from 0.455 to 0.517 (ProtT5-XL-U50 *full*) on the Ramasamy22-S/T dataset. On the Ramasamy22-Y dataset, an increase of up to 46.3% is observed, from 0.164 to 0.240 (ProtT5-XL-U50 *conv*).

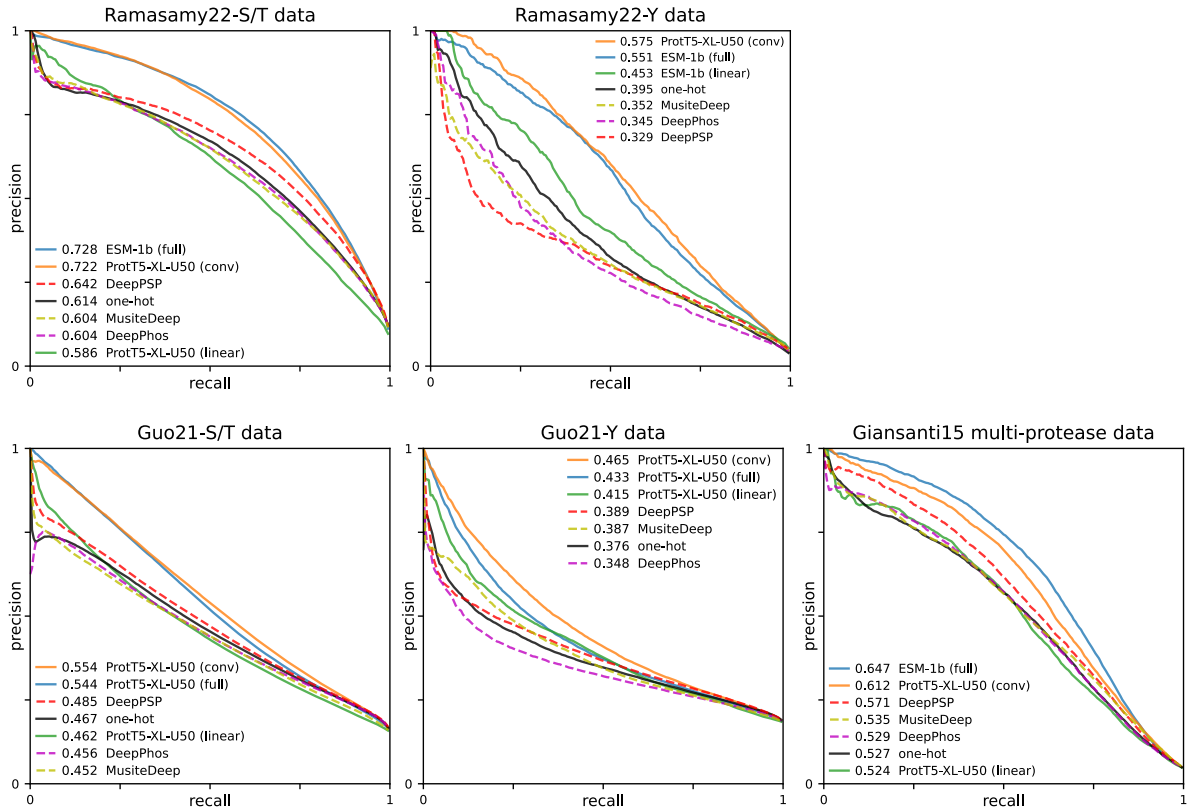

Figure 2. Precision-recall curves for the top PLM predictors per setup, a one-hot encoding, and existing predictors. Evaluation is done for five different datasets, each split up in a training set and a hold-out evaluation set. Average curves over 10 runs (per dataset) are depicted. In the legend, the 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 on Ramasamy22 and Guo21 data for the ESM-1b *full*

model, the best performing PLM-based model on the Ramasamy22-S/T dataset, are visualized in Figure 3.

For the S/T 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. Figure 7a-b). Other favourable residues in the S/T visualization are arginine (R) at positions P-3 and P-2, often seen for kinases in the PKa family (Suppl. Figure 7c), 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. Figure 7d). Visualizations for the one-hot encoded models, shown in Suppl. Fig. 8, confirm these findings.

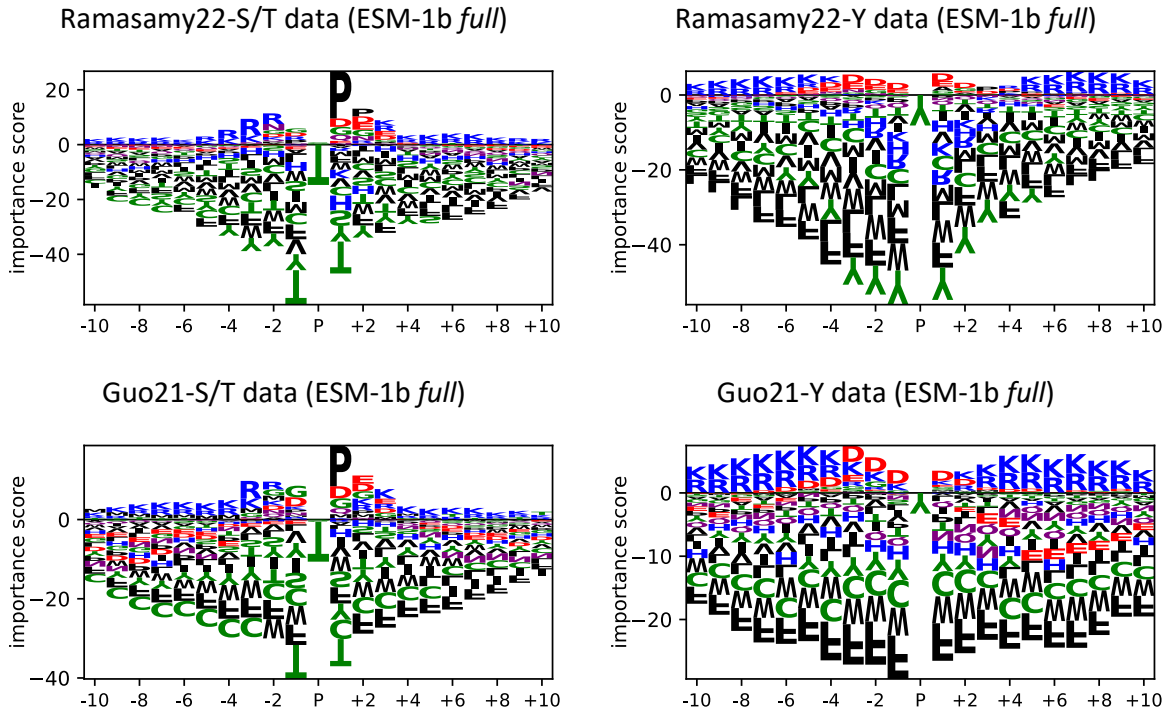

Figure 3. Average importance scores per position, calculated using DeepLiftSHAP, for the ESM-1b full prediction model setup, on the Ramasamy22 and Guo21 datasets. Visualizations are cropped to show the twenty positions surrounding the candidate P-site.

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

The average importance scores in Figure 3 suggest that the models disfavour hydrophobic amino acids in the proximity of the candidate P-site. On average, for the one-hot encoded model, W/C/F/Y/I/V/M/L are considered least favourable when in the proximity of a P-site, while for the PLM, this is the case for F/Y/C/V/I/W/L. 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[45] are more abundant in buried regions[45], 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[46,47] was used to obtain protein structures for all proteins in the dataset that were shorter than 2700 amino acids. Next, DSSP[48] 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].

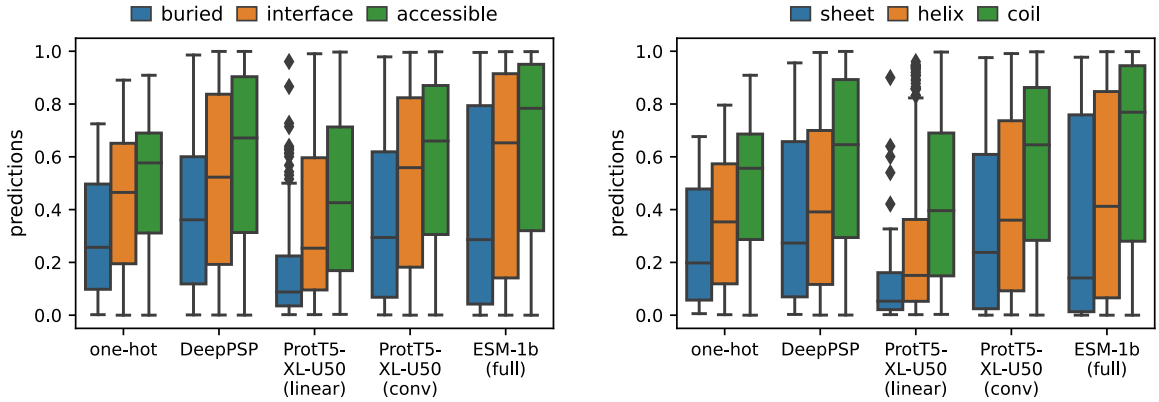

Figure 4. Distribution of predicted probabilities for positively labelled P-sites in the Ramasamy22-S/T test set, per model, divided by surface accessibility and secondary structure.

Datasets from experiments with other 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 Giansanti15 datasets, where predictors were trained and evaluated on data compiled from experiments that applied different proteases (Suppl. Table 5). Figure 5 shows that ESM-1b (*full*) 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. 9.

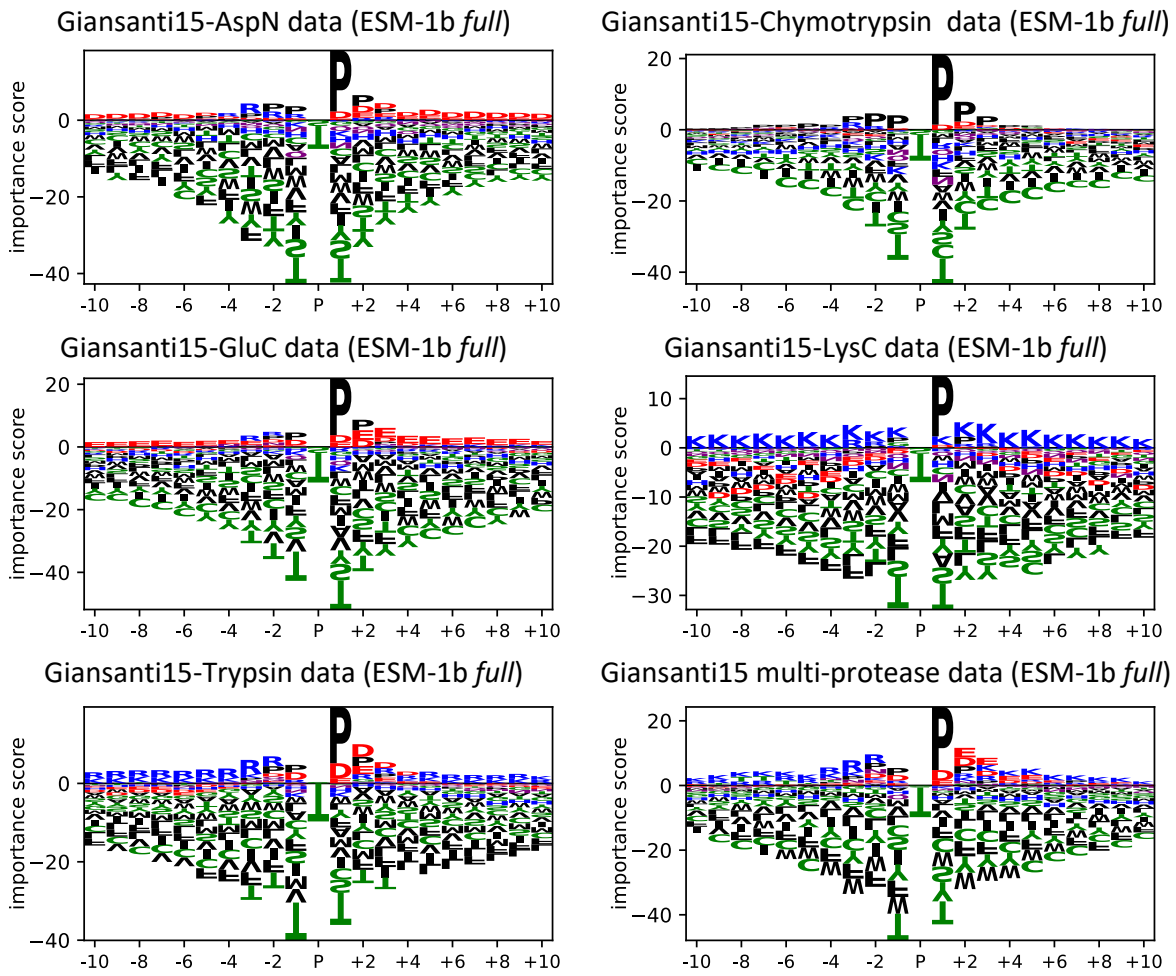

Figure 5. Average importance scores per position, calculated using DeepLiftSHAP, for the ESM-1b full prediction model setup, on the Giansanti15 datasets. Visualizations are cropped to show the twenty positions surrounding the candidate P-site.

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. Figs. 10-11.

ProtT5-XL-U90 *conv* models trained on the Ramasamy22 data are consistently outperformed by single-protease trained models with a relative AUPRC increase of 3% to 11%, except for the trypsin data. Here, the models trained on Ramasamy22 data slightly outperform the models trained on only Giansanti15 trypsin data by 2%. This is likely because the former is largely constructed from experiments using trypsin digestion, and it has more than 6 times as many positive annotations, hence more data were available in training, allowing for better prediction performance.

For the AspN, Chymotrypsin, and GluC evaluation sets, the multi-protease predictors outperform the single-protease predictors by 1% to 7%. On LysC, they reach roughly the same performance, and on Trypsin data, performance drops by 2%. Overall, this suggests that models trained on data obtained from multiple proteases generalize better. In addition, we

can expect that training on multi-protease data results in predictors that are less prone to the protease specific bias observed before.

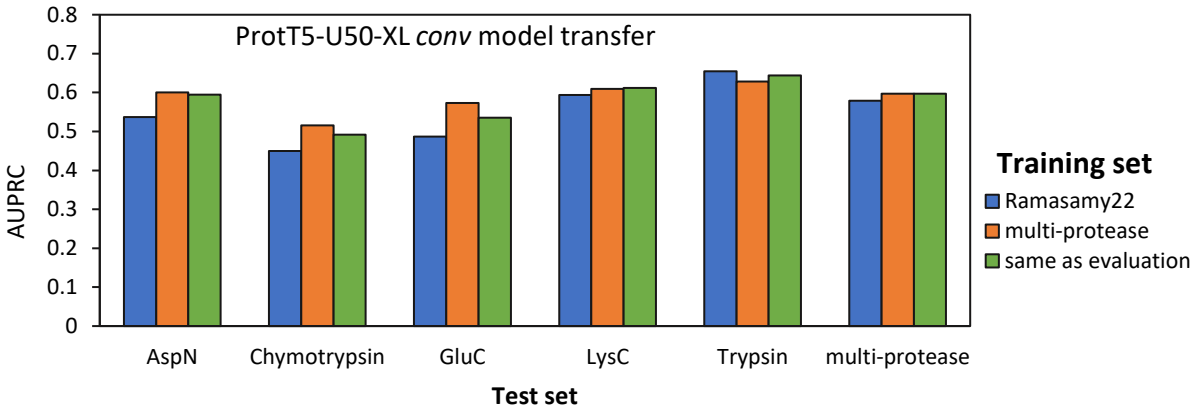

Figure 6. *Evaluation on Giansanti15 single-protease and multi-protease datasets (x-axis), when training on a different data source (legend). Predictions are computed from ten folds of the evaluation set, where training is performed on either Ramasamy22-S/T data, Giansanti15 multi-protease data, or on the dataset matching the data source on the x-axis. Note that proteins present in the test fold are omitted from training. The average AUPRC over these ten test folds is shown, for ProtT5-XL-U50 conv models.*

5. Discussion

With PLMs showing their potential in various structure-related prediction tasks, we applied them to general phosphorylation prediction. We showed that PLMs compute information rich protein representations that substantially boost P-site prediction accuracy as evaluated on different data sources. Of the proposed training strategies, both training of the CNN (*conv* setup) only and simultaneously fine-tuning the PLM (*full* setup) proved successful. Even for a simple linear setup that applies logistic regression on a PLM encoding of just the candidate P-site already compares on par with more complex CNN models with one-hot encoded receptive fields.

Furthermore, we compared our approach to currently available state-of-the-art approaches that apply non-standard building blocks such as squeeze-and-excitation networks in combination with bidirectional LSTM layers (DeepPSP), capsule networks (MusiteDeep), and densely connected CNN blocks with block concatenation (DeepPhos). Our approach substantially outperforms these architectures. On all five datasets considered, each of the *conv* and *full* setups that were trained on the two largest PLMs outperformed all considered baselines. Overall, PLMs clearly yield richer representations that contribute towards more effective phosphorylation prediction.

Inspecting average feature importance scores confirmed 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. Both for one-hot encoded and PLM-based models, predicted probabilities are found to be higher on average for accessible P-sites than for buried ones, and higher for P-sites within random coil regions than for ones in alpha helices or beta sheets.

We also found the prediction models to favour the presence of amino acids that are targeted by proteases for cleavage, in the proximity of candidate P-sites. Rather than a biological justification for phosphorylation, this is largely a result of statistical bias in the positive and negative labels in the datasets that emerge from the nature of protein digestion in bottom-up LC-MS/MS proteomics experiments. Most P-sites are determined after MS/MS experiments where trypsin is used for digestion[3]. As more missed cleavages occur near P-sites[4,6], the set of phosphorylated peptides contains more R/K-enriched peptides that are cleaved in fragments that are too short for MS/MS analysis if they are not phosphorylated. Additionally, R/K-poor regions result in longer peptides, which suffer from lower localization scores on average (see Suppl. Figure 12), or which can be too long for MS/MS analysis.

During training, machine learning models learn to exploit these kinds of biases as a distinctive feature between positive and negative samples. Thus, general phosphorylation prediction models do not just learn P-site fingerprints to predict the likeliness of phosphorylation, but also the likeliness that they are detected with current experimental setups. We are currently only able to identify a delineated subset of all possible P-sites, as with current standard high-throughput experimental methodologies, any prediction model trained on resulting data will attempt to learn that delineation, rather than the exact boundary between actual and pseudo P-sites.

This was further illustrated by transferring prediction models between phosphorylation datasets that resulted from experiments with different proteases and showing that this was detrimental for prediction performance. Hence, 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 using the same protease should be used.

We provide an accurate PLM-based P-site predictor on our GitHub repository (https://github.com/jasperzuallaert/PhosphoLingo), trained on a large multi-source dataset that combines all annotations from the considered datasets in this paper. A user-friendly framework is released to reproduce the results in this manuscript, as well as to train and evaluate models on other phosphorylation datasets, or even for other post-translational modifications. Furthermore, we supply predictions for all S, T and Y residues in the human proteome, using an ensemble of the best prediction models discussed in this manuscript.

Acknowledgements

Funding

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

Conflict of interest: none declared

414 # References

415 1. Ramazi, S. & Zahiri, J. Post-translational modifications in proteins: resources, tools and
416 prediction methods. *Database* **2021**, (2021).
417 2. Ramasamy, P., Vandermarliere, E., Vranken, W. & Martens, L. Panoramic Perspective on
418 Human Phosphosites. *Journal of Proteome Research* **21**, 1894–1915 (2022).
419 3. Pinna, L. A. & Ruzzene, M. How do protein kinases recognize their substrates?
420 *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* **1314**, 191–225 (1996).
421 4. Giansanti, P. *et al.* An Augmented Multiple-Protease-Based Human Phosphopeptide
422 Atlas. *Cell Reports* **11**, 1834–1843 (2015).
423 5. Dephoure, N., Gould, K. L., Gygi, S. P. & Kellogg, D. R. Mapping and analysis of
424 phosphorylation sites: A quick guide for cell biologists. *Molecular Biology of the Cell* **24**,
425 535–542 (2013).
426 6. Tsiatsiani, L., Heck, A. J. R., Heck, A. J. R. & Bijvoet, P. Proteomics beyond trypsin. *The*
427 *FEBS Journal* **282**, 2612–2626 (2015).
428 7. Trost, B. & Kusalik, A. Computational prediction of eukaryotic phosphorylation sites.
429 *Bioinformatics* **27**, 2927–2935 (2011).
430 8. Ma, R. *et al.* KinasePhos 3.0: Redesign and Expansion of the Prediction on Kinase-specific
431 Phosphorylation Sites. *bioRxiv preprint* (2021) doi:10.1101/2021.11.02.467032.
432 9. Song, J. *et al.* PhosphoPredict: A bioinformatics tool for prediction of human kinase-
433 specific phosphorylation substrates and sites by integrating heterogeneous feature
434 selection. *Scientific Reports* **7**, (2017).
435 10. Xu, Y., Song, J., Wilson, C. & Whisstock, J. C. PhosContext2vec: A distributed
436 representation of residue-level sequence contexts and its application to general and
437 kinase-specific phosphorylation site prediction. *Scientific Reports* **8**, (2018).

438 11. Wang, C. *et al.* GPS 5.0: An Update on the Prediction of Kinase-specific Phosphorylation
439 Sites in Proteins. *Genomics, Proteomics and Bioinformatics* **18**, 72–80 (2020).
440 12. Guo, L. *et al.* DeepPSP: A Global-Local Information-Based Deep Neural Network for the
441 Prediction of Protein Phosphorylation Sites. *Journal of Proteome Research* **20**, 346–356
442 (2021).
443 13. Wei, L., Xing, P., Tang, J. & Zou, Q. PhosPred-RF: A Novel Sequence-Based Predictor for
444 Phosphorylation Sites Using Sequential Information only. *IEEE Transactions on*
445 *Nanobioscience* **16**, 240–247 (2017).
446 14. Li, F. *et al.* Quokka: A comprehensive tool for rapid and accurate prediction of kinase
447 family-specific phosphorylation sites in the human proteome. *Bioinformatics* **34**, 4223–
448 4231 (2018).
449 15. Dou, Y., Yao, B. & Zhang, C. PhosphoSVM: Prediction of phosphorylation sites by
450 integrating various protein sequence attributes with a support vector machine. *Amino*
451 *Acids* **46**, 1459–1469 (2014).
452 16. Yang, H., Wang, M., Liu, X., Zhao, X.-M. & Li, A. PhosIDN: an integrated deep neural
453 network for improving protein phosphorylation site prediction by combining sequence
454 and protein–protein interaction information. *Bioinformatics* **37**, 4668–4676 (2021).
455 17. Wang, D. *et al.* MusiteDeep: A deep-learning based webserver for protein post-
456 translational modification site prediction and visualization. *Nucleic Acids Research* **48**,
457 W140–W146 (2021).
458 18. Luo, F., Wang, M., Liu, Y., Zhao, X. M. & Li, A. DeepPhos: Prediction of protein
459 phosphorylation sites with deep learning. *Bioinformatics* **35**, 2766–2773 (2019).
460 19. Lv, H., Dao, F.-Y., Zulfiqar, H. & Lin, H. DeepIPs: comprehensive assessment and
461 computational identification of phosphorylation sites of SARS-CoV-2 infection using a

462 deep learning-based approach. *Briefings in Bioinformatics* (2021)
463 doi:10.1093/bib/bbab244.
464 20. Deznabi, I., Arabaci, B., Koyuturk, M. & Tastan, O. DeepKinZero: Zero-shot learning for
465 predicting kinase-phosphosite associations involving understudied kinases.
466 *Bioinformatics* **36**, 3652–3661 (2020).
467 21. Kirchoff, K. & Gomez, S. EMBER: multi-label prediction of kinase-substrate
468 phosphorylation events through deep learning. *Bioinformatics* **38**, (2022).
469 22. Blom, N., Sicheritz-Pontén, T., Gupta, R., Gammeltoft, S. & Brunak, S. Prediction of post-
470 translational glycosylation and phosphorylation of proteins from the amino acid
471 sequence. *Proteomics* **4**, 1633–1649 (2004).
472 23. Jamal, S., Ali, W., Nagpal, P., Grover, A. & Grover, S. Predicting phosphorylation sites
473 using machine learning by integrating the sequence, structure, and functional
474 information of proteins. *Journal of Translational Medicine* **19**, (2021).
475 24. Ma, H., Li, G. & Su, Z. KSP: An integrated method for predicting catalyzing kinases of
476 phosphorylation sites in proteins. *BMC Genomics* **21**, (2020).
477 25. Thapa, N. *et al.* A deep learning based approach for prediction of Chlamydomonas
478 reinhardtii phosphorylation sites. *Scientific Reports* **11**, (2021).
479 26. Ismail, H. D., Jones, A., Kim, J. H., Newman, R. H. & Kc, D. B. RF-Phos: A Novel General
480 Phosphorylation Site Prediction Tool Based on Random Forest. *BioMed Research*
481 *International* **2016**, (2016).
482 27. Brown, T. B. *et al.* Language Models are Few-Shot Learners. *Advances in Neural*
483 *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. 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.
34. 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).
35. Detlefsen, N. S., Hauberg, S. & Boomsma, W. Learning meaningful representations of protein sequences. *Nat Commun* **13**, 1914 (2022).
36. Lundberg, S. M. & Lee, S. A unified approach in interpreting model predictions. *Advances in Neural Information Processing Systems* 4765–4774 (2017).

37. Kokhlikyan, N. *et al.* Captum: A unified and generic model interpretability library for PyTorch. *arXiv preprint* (2020) doi:10.48550/arXiv.2009.07896.

38. Dinkel, H. *et al.* Phospho.ELM: a database of phosphorylation sites—update 2011. *Nucleic Acids Research* **39**, D261–D267 (2011).

39. Ramasamy, P. *et al.* Scop3P: A Comprehensive Resource of Human Phosphosites within Their Full Context. *Journal of Proteome Research* **19**, 3478–3486 (2020).

40. 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).

41. Hornbeck, P. V. *et al.* PhosphoSitePlus, 2014: mutations, PTMs and recalibrations. *Nucleic acids research* **43**, D512–D520 (2015).

42. 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).

43. Biewald, L. Experiment Tracking with Weights and Biases. (2020).

44. Tareen, A. & Kinney, J. B. Logomaker: Beautiful sequence logos in python. *bioRxiv preprint* (2019) doi:10.1101/635029.

45. 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.

46. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature 2021 596:7873* **596**, 583–589 (2021).

47. 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).

48. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**, 2577–2637 (1983).