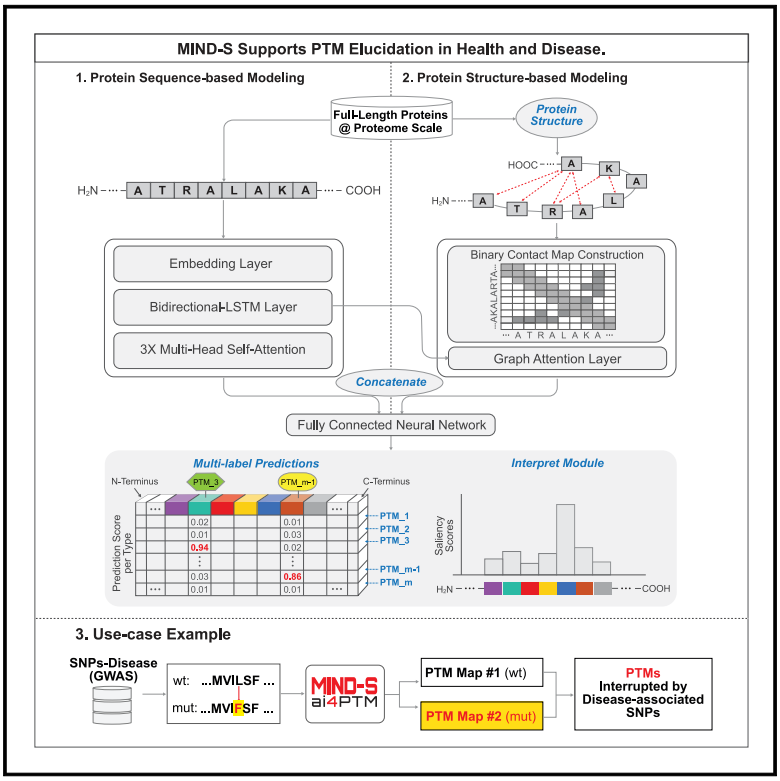
Cell Reports
Methods

Article

MIND-S is a deep-learning prediction model for elucidating protein post-translational modifications in human diseases

Graphical abstract



Highlights

- MIND-S predicts PTM based on protein sequence and structure by a deep-learning method
- MIND-S applies a multi-label strategy to predict multiple PTM types and sites
- MIND-S predicts important amino acids for a given PTM
- MIND-S examines the impact of SNP mutations at proteome level from a PTM perspective

Authors

Yu Yan, Jyun-Yu Jiang, Mingzhou Fu, ..., Dominic C.M. Ng, Wei Wang, Peipei Ping

Correspondence

weiwang@cs.ucla.edu (W.W.),
pping38@g.ucla.edu (P.P.)

In brief

Yan et al. develop a deep-learning-based post-translational modification (PTM) prediction tool, MIND-S. We demonstrate the performance of MIND-S via its prediction of 26 types of PTM at the proteome level utilizing both protein sequence and structure. MIND-S also provides an interpretation module to better understand PTM mechanisms and a mutation evaluation method to study the relationship between mutation (e.g., SNP) and PTM.

Yan et al., 2023, Cell Reports Methods 3, 100430
March 27, 2023
https://doi.org/10.1016/j.crmeth.2023.100430

CellPress

Cell Reports Methods

CellPress
OPEN ACCESS

Article

MIND-S is a deep-learning prediction model for elucidating protein post-translational modifications in human diseases

Yu Yan,[1,2,3] Jyun-Yu Jiang,[4] Mingzhou Fu,[2] Ding Wang,[1,3] Alexander R. Pelletier,[1,3,4] Dibakar Sigdel,[1,3] Dominic C.M. Ng,[1,3] Wei Wang,[1,2,4,*] and Peipei Ping[1,2,3,4,5,6,*]

[1]NIH BRIDGE2AI Center at UCLA & NHLBI Integrated Cardiovascular Data Science Training Program at UCLA, Suite 1-609, MRL Building, 675 Charles E. Young Dr. South, Los Angeles, CA 90095-1760, USA

[2]Medical Informatics Program, University of California at Los Angeles (UCLA), Los Angeles, CA 90095, USA

[3]Department of Physiology, UCLA School of Medicine, Suite 1-609, MRL Building, 675 Charles E. Young Dr., Los Angeles, CA 90095-1760, USA

[4]Scalable Analytics Institute (ScAi) at Department of Computer Science, UCLA School of Engineering, Los Angeles, CA 90095, USA

[5]Department of Medicine (Cardiology), UCLA School of Medicine, Suite 1-609, MRL Building, 675 Charles E. Young Dr. South, Los Angeles, CA 90095-1760, USA

[6]Lead contact

*Correspondence: weiwang@cs.ucla.edu (W.W.), pping38@g.ucla.edu (P.P.)

https://doi.org/10.1016/j.crmeth.2023.100430

MOTIVATION Post-translational modifications (PTMs) serve as key regulatory mechanisms in many cellular processes; altered PTMs contribute significantly to disease pathogenesis in humans. Due to the high complexity and a large quantity of PTM studies, computational methods to predict PTMs have been appreciated to be effective approaches to study PTMs. Here, we present MIND-S, a deep-learning-based PTM prediction tool utilizing protein-level information combined with its sequence and structure. MIND-S demonstrates excellent performance on simultaneous prediction of multiple types of PTM in a proteome setting. Integrating with SNP information identified from GWAS, MIND-S can offer molecular insights and evaluate the impact of SNP from a PTM perspective.

SUMMARY

We present a deep-learning-based platform, MIND-S, for protein post-translational modification (PTM) predictions. MIND-S employs a multi-head attention and graph neural network and assembles a 15-fold ensemble model in a multi-label strategy to enable simultaneous prediction of multiple PTMs with high performance and computation efficiency. MIND-S also features an interpretation module, which provides the relevance of each amino acid for making the predictions and is validated with known motifs. The interpretation module also captures PTM patterns without any supervision. Furthermore, MIND-S enables examination of mutation effects on PTMs. We document a workflow, its applications to 26 types of PTMs of two datasets consisting of ~50,000 proteins, and an example of MIND-S identifying a PTM-interrupting SNP with validation from biological data. We also include use case analyses of targeted proteins. Taken together, we have demonstrated that MIND-S is accurate, interpretable, and efficient to elucidate PTM-relevant biological processes in health and diseases.

INTRODUCTION

Protein post-translational modifications (PTMs) are covalent processing events that alter the biophysical properties of a protein through the addition of a modifying group to one or more amino acids. PTMs serve as key regulatory mechanisms governing a broad spectrum of sub-proteomes and are commonly involved in many disease phenotypes.[1,2] The diversity of PTM types and the large number of amino acid residues involved enable the greater regulatory capacity of PTMs, yet substantial challenges remain in detecting and understanding PTMs. Although large-scale PTM identification has been improved with proteomics tools,[3] they remain costly, labor-intensive, and time-intensive, especially when PTM-specific enrichment approaches are necessary for their detection.

Recently, computational approaches to predict PTM sites have gained traction.[2,4,5] A common and widely used prediction schema is to predict PTMs based on local amino acids spanning the target sites.[2] Specifically, amino acids flanking the PTM site are leveraged to make predictions on the target residual. However, this strategy relies heavily upon surrounding amino acids, whereas whole-protein-level information is less considered. Moreover, these approaches require selecting an optimal length of flanking amino acid sequence for different types of PTMs, limiting the transferability among PTM types.

Another major consideration is the interpretability pertaining to the underlying mechanism supporting model predictions. This is especially the case for deep-learning-based approaches, which often demonstrate excellent prediction results but without reasonable explanations (i.e., interpretation). Thus, the selection of interpretation methods becomes essential to help us understand the anticipated model output upon a certain set of input. The optimal interpretation methods enable us to uncover hidden patterns affecting PTM occurrence. For example, feature importance is one of the interpretation methods that evaluate which inputs (in this case, amino acids) are important for the anticipated output. Although several amino acid patterns related to phosphorylation have been uncovered,[6] many PTM patterns as well as their underlying mechanisms largely remain a mystery. Indeed, many phosphosites are orphans without information on their associated kinases.[7]

To overcome these challenges, we developed an artificial intelligence (AI)-based tool, MIND-S (multi-label interpretable deep-learning method for PTM prediction-structure version), which predicts PTMs at the protein level. Specifically, the protein sequence and structure are given as the input and the predictions are made on all possible residuals at the same time. This schema allows the model to make batch predictions across multiple protein sequences, multiple amino acid sites, and multiple PTM types at a proteome scale. We also adapted the integrated gradient method to interpret MIND-S by identifying residues important for prediction. We demonstrated that MIND-S achieves great performance for PTM prediction with excellent computational efficiency and interpretability. We present use cases of MIND-S, including an examination of how the SNP can affect PTM occurrences, which bridges the gap between genetic data and PTMs.

RESULTS

MIND-S model design and performance

We present a computation model, MIND-S, for protein PTMs prediction, utilizing graph neural network (GNN) and multi-head attention to extract information from protein structure and protein sequence. The overall design of MIND-S is detailed in the graphical abstract.

Our model is built at the protein level, where all PTMs pertaining to one protein are put within the same instance. All protein data were split into training, testing, and validation sets on the protein level as well. To ensure fair evaluation, proteins assigned in the testing set must share less than 50% sequence similarity with proteins in training and validation sets.[8] To increase model robustness, a bootstrap method was implemented, where multiple models are trained on sampled datasets and ensembled together at the end stage (Figure 1A). A fixed testing set (~5% of the whole dataset) was retained and the remaining data were split into training and validation sets at random at each iteration. To account for the various length of proteins and to alleviate the problem of redundant padding, the full-length protein with its PTM is split into multiple core sequences, on which the model will predict. Core sequences were then extended on both sides (up to 128 amino acids) to ensure sufficient contextual sequence information (Figure 1B). Extended core sequences were input to our model for multi-label training, and all PTMs falling within core sequences will be trained simultaneously (Figure 1C). The trained model was evaluated on the validation set by the area under the precision-recall curve (AUPR).[9] AUPR was chosen over the area under the receiver operating curve (AUC)[10] as AUPR yields a more informative evaluation when the data are imbalanced,[11] which is especially true for PTMs. The number of negative PTM samples (targeted residue without PTM) is far greater than the number of positive PTM samples (target residue with a PTM) (Table S1). The above training process was repeated 15 times to ensemble the final model, which is the weighted average of predictions from 15 models.

For the model architecture, MIND-S takes protein sequences and structures as the input and outputs PTM prediction scores (ranging from 0 to 1) for every targeted residue. One-hot encodings of these protein sequences are passed through a feedforward neural network, which converts the sparse representation into a dense numeric vector capturing biochemical properties (Figure S1). The embedding is then passed to a bidirectional long-short term memory (LSTM)[12] layer, which passes information along the sequence bidirectionally and encodes positional information. The LSTM embedding serves as the token embedding and node embedding for multi-head self-attention block[13] and graph attention layer,[14] a GNN model, respectively. The multi-head self-attention is designed to capture information about the protein sequence while the graph attention layer is employed to gather information from important and spatially close (close in 3D space) amino acids guided by the protein contact map. Last, the outputs of the two components are concatenated and converted to prediction score PTMs by a feedforward neural network layer. Detailed descriptions of our model architecture are described in the STAR Methods section. In addition, we also provide MIND, an alternative version to MIND-S that makes predictions solely based on the protein sequence. In the following paragraphs, we performed analyses to demonstrate the contribution of different modules or layers of our model.

MIND-S has several unique design features that facilitate its performance of the prediction task, as demonstrated by our experiments. We have selected 13 types of curated PTM as the benchmark dataset to test the model performance (Table S1). We also constructed a dataset consisting of 13 types of oxidative PTM (O-PTM) from a mass spectrometry project as an independent dataset (Table S2). We investigated the contribution of sequence and structure components of MIND-S to gain a deeper understanding of the PTM prediction task. We showed that adequate data are a vital part of the model. We trained and evaluated MIND-S with different amounts of data sampling from the whole dataset, and the results indicated that the performance of

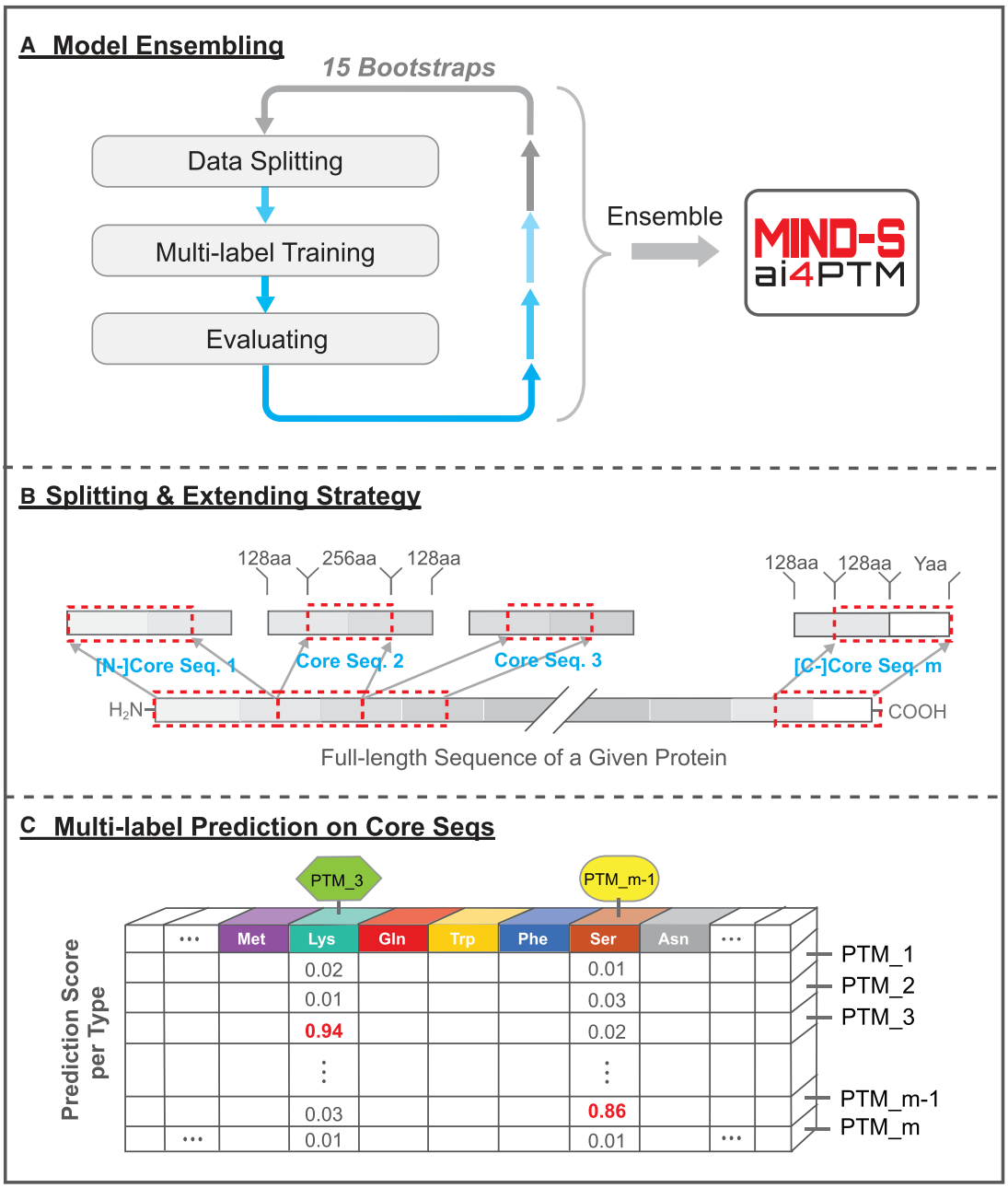


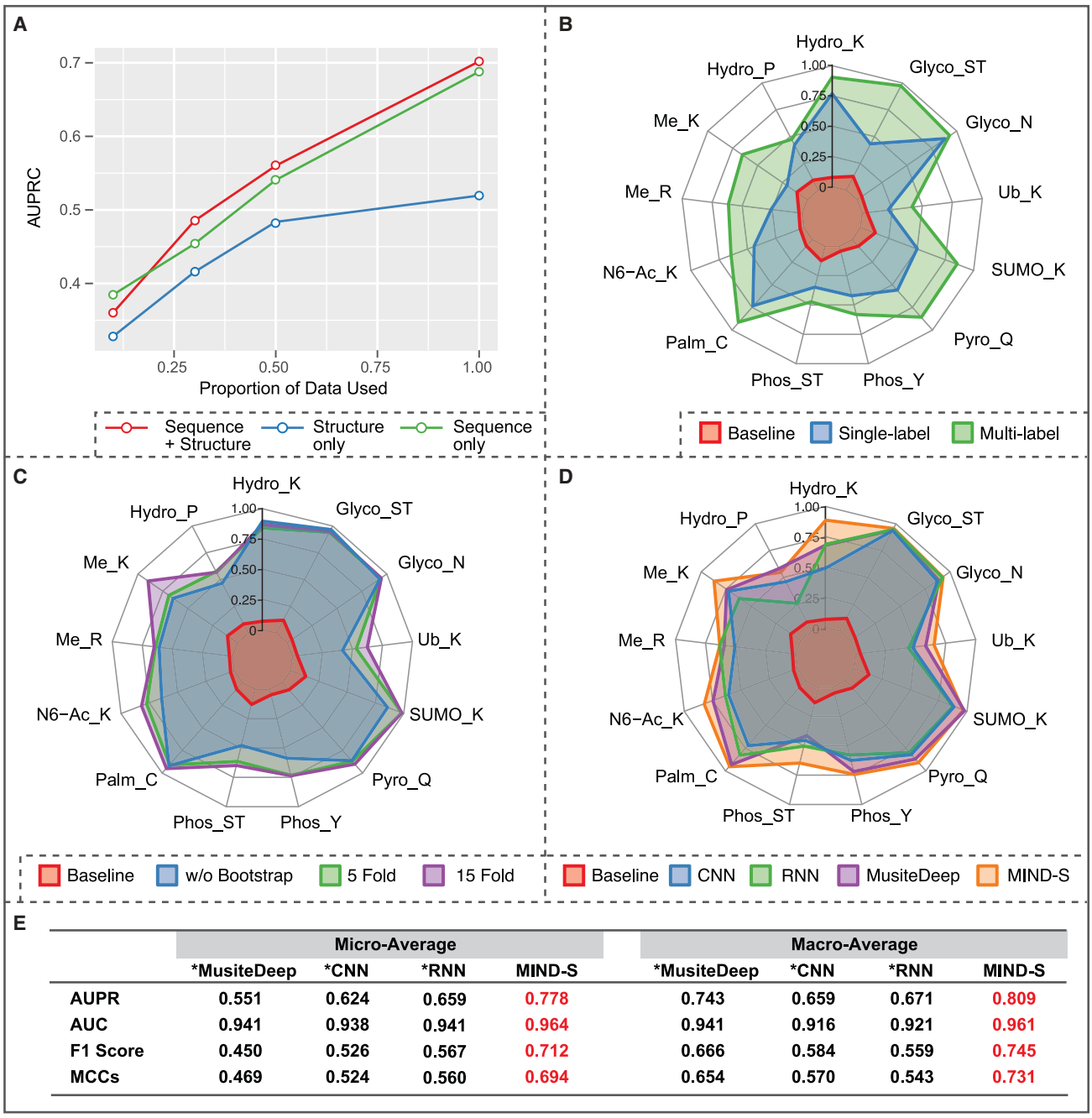
Figure 1. Design of MIND-S

(A) A workflow on ensemble MIND-S. A fixed dataset is retained for testing, with the remaining data split into a training set and a validation set in the data splitting step; PTM data at the protein level are mapped to core sequences using the split and extend strategy detailed in (B); each individual model is trained on the processed data under the multi-label setting as detailed in (C); each model is subsequently evaluated in the evaluation steps. This process is repeated 15 times to ensemble the final model, MIND-S.

(B) The splitting and extending strategy. The full-length protein is first split into multiple core sequences. To ensure sufficient information for prediction, each core sequence is then extended (additional 128 amino acid residues on both C and N termini; on only one side when it is the N terminus or C terminus core sequence).

(C) The multi-label training on the core sequence. The prediction score matrix representing one core sequence is shown. Columns of the matrix correspond to amino acid residues, rows of the matrix correspond to PTM types, and each cell corresponds to the prediction score of the specific PTM.

the model is proportional to the amount of training data (Figure 2A). Furthermore, we uncovered that the sequence (modeled by multi-head attention) and structure (modeled by graph attention layer) components together provide the best performance. We constructed an ablated version of MIND-S with either multi-head attention or graph attention layer removed as structure-only and sequence-only models. Individually, the sequence-only model performs better than the structure-only



E

	Micro-Average				Macro-Average			
	*MusiteDeep	*CNN	*RNN	MIND-S	*MusiteDeep	*CNN	*RNN	MIND-S
AUPR	0.551	0.624	0.659	0.778	0.743	0.659	0.671	0.809
AUC	0.941	0.938	0.941	0.964	0.941	0.916	0.921	0.961
F1 Score	0.450	0.526	0.567	0.712	0.666	0.584	0.559	0.745
MCCs	0.469	0.524	0.560	0.694	0.654	0.570	0.543	0.731

Figure 2. MIND-S performance on PTM prediction

(A) The line plot presents the relationship between the number of data points and performance. The x axis is the proportion of training data employed to train the model, and the y axis is the performance (macro-average AUPR) of the model. The red line shows the performance of the model with both the sequence and the structure components; the green line and the blue line show the performance of the model with only the sequence component and the model with only the structure component, respectively. All models achieve performances with more data, and the model with both components performs the best.

(B–D) Both positive and negative data points in each PTM type were applied and analyzed. Radar plots present PTM prediction and model performance. The baseline AUPR (total detected PTMs divided by total available AA residues) for each PTM type is shown in red. (B) The AUPR for PTM under single-label setting is shown in blue, and the AUPR under multi-label setting is shown in green. The multi-label model shows better performance than the single-label setting in all PTM types. (C) The AUPR for benchmark PTMs on the model trained with 5- and 15-fold bootstrapping is shown in green and purple, respectively. The AUPR of the model without bootstrapping is shown in green. The bootstrap method shows better performance, and the 15-fold bootstrap method achieves the best performance. (D) The AUPR for benchmark PTMs of MIND-S, MusiteDeep, CNN, and RNN are shown in orange, purple, blue, and green respectively. Overall, MIND-S shows the best performance in most of the PTM types.

(legend continued on next page)

model, which suggests that protein sequence is most informative for PTM prediction. However, the two components combined achieve the best performance, suggesting that graph attention layer can provide valuable information not captured in the protein sequence (Figure 2A).

Another feature, multi-label[15] training and prediction, in MIND-S was shown to improve the overall performance of the model. Unlike the conventional approaches, where one model is trained to predict one type of PTM, multi-label allows one model to be trained to predict multiple types of PTM. This strategy can benefit PTM with fewer samples available. Such PTMs are usually more challenging to predict due to the limited availability of relevant datasets; as demonstrated in the previous section, the amount of data is proportional to the performance. MIND-S addresses this issue by employing multi-label prediction such that the learning process (parameters in the network) of different types of PTM is shared during training; PTM types with fewer data can "borrow" knowledge learned from other PTM types. Moreover, instead of separately training each PTM type, all PTM types were trained and predicted simultaneously, which speeds up the training and predicting processes, alleviating the computational burden. We evaluated model performance under the single-label settings (where training is performed separately for each PTM) compared with the multi-label setting on each PTM type. Our results reveal that multi-label substantially improved the prediction performance for most PTM types, especially for PTMs with limited data, such as hydroxyl lysine and O-linked glycosylation on serine and threonine (Figure 2B). Using a multi-label strategy, MIND-S greatly improves the prediction of these PTM types. To a lesser extent, commonly studied PTM types also benefit from a multi-label strategy. However, the performance of one such PTM, N-linked glycosylation, showed little improvement, suggesting the improvement from adding data for this PTM was saturated.

In addition, we showed that utilizing a bi-LSTM layer, instead of fixed positional encoding methods to capture positional information (i.e., the sequential order of amino acid),[16] improves model performance such that the representation of positional information is learnable and can be better utilized to improve predictions. Indeed, Figure S2A shows that the model performs poorly without any position information; amino acid composition by itself is not sufficient for prediction.

Last, a bootstrap method is applied: the dataset is split 15 times to generate 15 training and validation sets, where the size of the validation set is 5% of the total size of the dataset. The 15 models were trained on each set, and an ensemble model was obtained by averaging the output prediction scores from N models weighted by AUPR scores on validation sets. The bootstrap step is to enhance the robustness of MIND-S, and the weighting is to adjust for the variation of the performance by the models.[17] As a result, Figure 2C highlights that the bootstrap method enhanced the model's performance, and the model achieved its best performance at N = 15. Therefore, we chose N = 15 for bootstrapping in MIND-S.

Few tools are available for multiple PTM predictions.[2] MIND-S is benchmarked against MusiteDeep, which is a valid PTM prediction tool that allows multiple PTMs prediction, outperforming several other single-PTM prediction tools.[18] MusiteDeep is a convolutional neural network (CNN)-based model for multiple PTM predictions, and it takes in the one-hot encoded flanking sequences of length 33 and passes to an ensemble of multi-CNN and Capsnet models. We also construct a straightforward CNN and a recurrent neural network (RNN) under our protein-level prediction schema as a comparison between the two schemas. The performances of MIND-S, MusiteDeep, CNN, and RNN models are shown in Figure 2D. MIND-S has the best performance in most types of PTM when evaluated by AUPR. MIND-S also shows the best performances on all aggregate metrics (Figure 2E). Moreover, thanks to the multi-label and protein-level training design, MIND-S has a far smaller size (698,765 parameters for 13 types of PTM together) compared with MusiteDeep (2,342,680 parameters for each PTM), which renders superior computation speed for the training process and demands fewer computational resources. In addition, the CNN model outperforms MusiteDeep in terms of micro-average metrics even though it is simple in terms of model design, indicating that our protein-level prediction schema may help the model better capture the information needed. In addition, analyses on the hyperparameters of MIND-S can be found in Table S4.

MIND-S provides biological interpretation through integrated gradients

Given that MIND-S can accurately predict PTM occurrences, we seek to interpret its predictions to gain insight into how PTMs occur. MIND-S adapts a *post hoc* interpretation method, integrated gradients,[19] to provide a way to interpret the model prediction. This interpretation method can evaluate to what extent each amino acid residue can affect the final prediction. In other words, it can identify the important amino acids for PTM. The integrated gradients method was originally designed for continuous values; we adapted this approach to amino acid residue embeddings. Since each amino acid is mapped to a multi-dimension embedding, integrated gradients of each dimension of the embedding were summed to generate a single saliency score for that amino acid as a measurement of importance to prediction.

To evaluate if the interpretation method can capture biologically relevant information, we compared the interpretation with the known PTM motif and consensus sequence patterns of the flanking sequence of a PTM.[20] We first investigated its application on the N-linked glycosylation, which possesses a relatively stable recognition pattern: Asn-X-Ser/Thr, where Asn is the PTM site and X is any amino acid except proline.[21] To evaluate the robustness of our interpretation method, we apply it to all confident and correct predictions in the test set, as the model has not "seen"

(E) Table of model performances. Micro- and macro-aggregated metrics (AUPR, AUC, F1 score, Matthews correlation coefficient [MCC]) on benchmark PTM data of four models: MIND-S, CNN, RNN, and MusiteDeep. MIND-S shows the best performance in every metric measured. One-sided paired t test was performed on binary cross-entropy loss between MIND-S and other models; *$p < 0.001$.
See also Table S5.

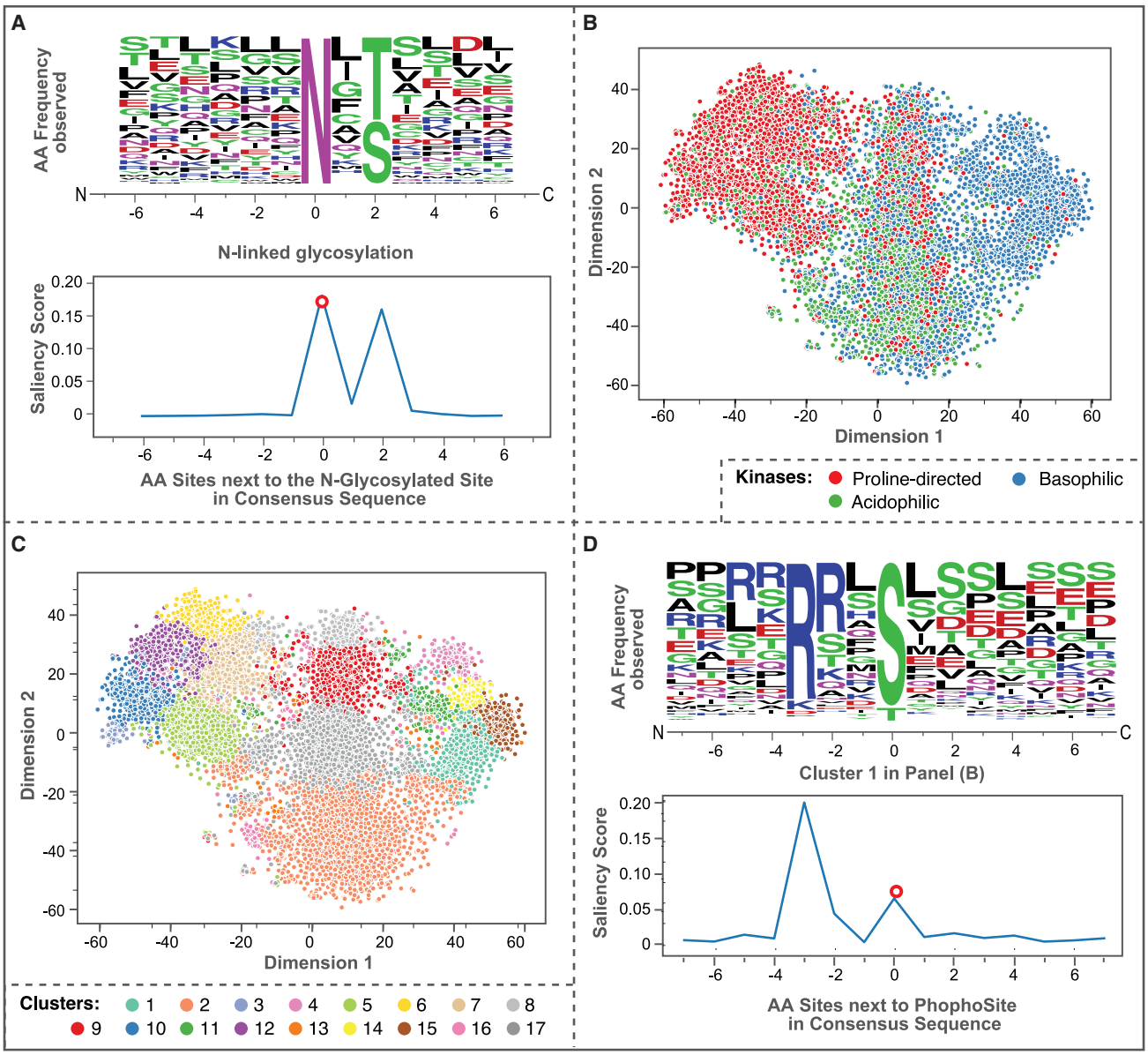


Figure 3. Validation of the interpretation module of MIND-S

(A) The upper panel shows the sequence frequency plot of all sequences from glycosylation sites investigated, where the +2 position shows enrichment of serine (S) or threonine (T). The bottom panel shows the averaged saliency scores of the same glycosylation sites, where the 0 and +2 position has a peak saliency score. The two panels show a matching of emphasis at the +2 position.

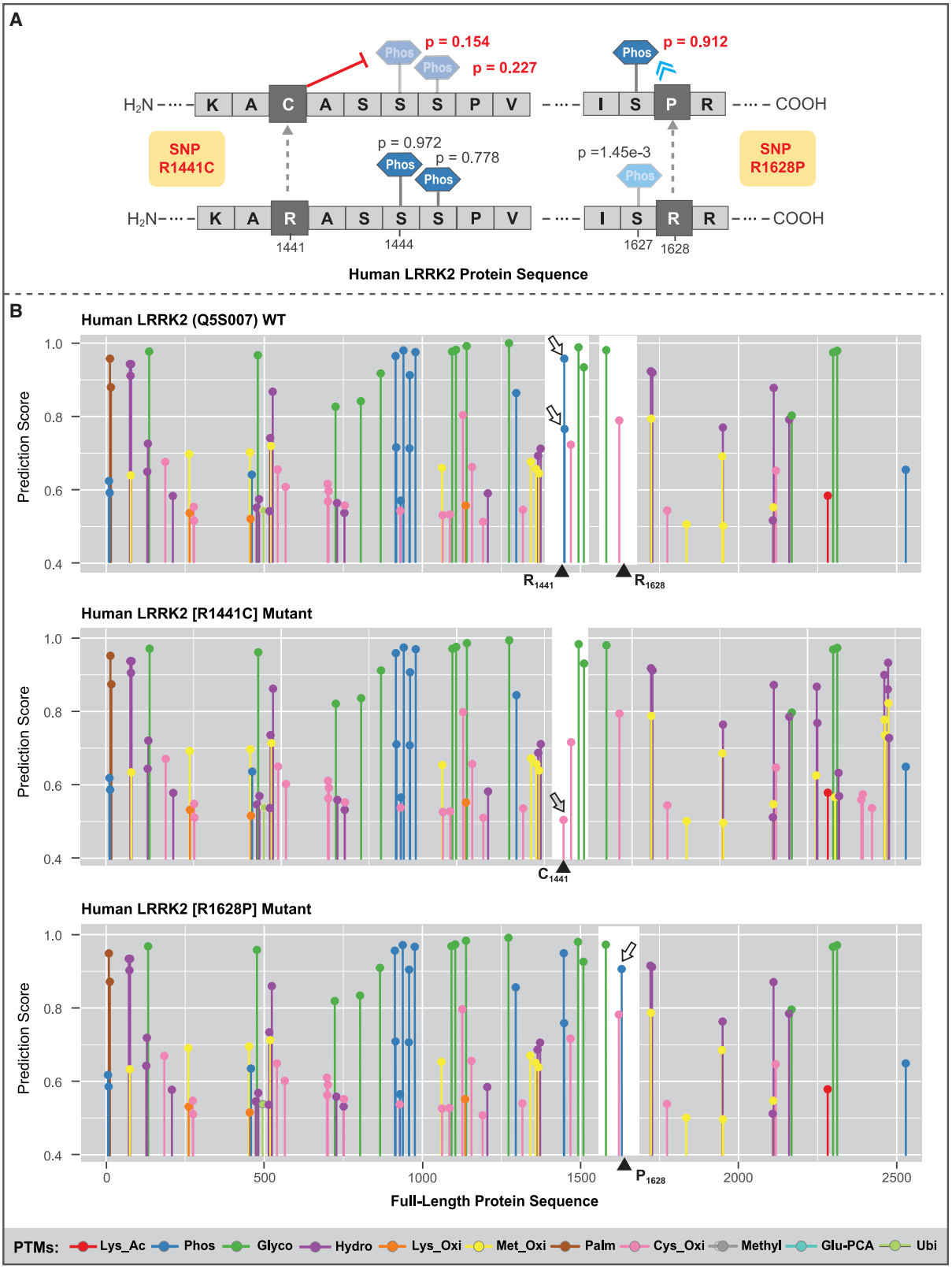
(B and C) The t-SNE plot of flanking saliency scores of phosphosites. Points in (A) are colored by the kinase group of the phosphosites: proline-directed kinase (red), basophilic kinase (blue), and acidophilic kinase (green). The three kinase groups are roughly distributed in three regions: left, right, and middle. Points in (B) are colored by the clusters (17 clusters in total). (C) The upper panel is the sequence frequency plot of all sequences from cluster 1, where the −3 position shows enrichment of arginine (R). The bottom panel is the saliency scores of the representative of cluster 1, where the −3 position has a peak saliency score. The two panels show a matching of emphasis at the −3 position.

the test set during training. Saliency scores of the amino acids surrounding the N-linked glycosylation site were calculated and averaged by their relative position to the PTM sites (Figure 3A). Obvious peaks at the position of 0 (the glycosylation site) and the position of +2, matching the consensus recognition pattern Asn-X-Ser/Thr, were shown in the averaged saliency scores. The comparison with the sequence frequency plots from the corresponding flanking sequences of the PTM sites further demonstrates that our model is able to faithfully capture recognition patterns. We then evaluate the method in the scenario that there is a mixture of various recognition patterns. We focused on phosphorylation where a variety of recognition patterns exist.[22,23] Kinases are responsible for phosphorylating proteins with specific recognition patterns. We searched for protein sequences in our

dataset to find all the phosphorylation sites with a studied motif region through Scansite 4.0[24]; 13,689 phosphosites were found with at least one motif, and MIND-S predicted 13,377 of them correctly. To evaluate if our interpretation method can distinguish phosphorylations introduced by different kinase groups, we calculated the saliency scores of the flanking amino acid (of length 21, including the PTM site itself) of each phosphorylation. We applied t-SNE (t-distributed stochastic neighbor embedding) on the saliency scores of each phosphorylation to reduce the dimension for visualization, and we colored them based on the associated kinases motif. Three groups of kinases were depicted: proline-dependent, basophilic, and acidophilic kinases (Figure 3A). From the t-SNE plot, the three groups of kinases are roughly separated, with proline-dependent kinases falling on the left, basophilic kinases falling on the right, and acidophilic falling in the middle. This suggests that MIND-S's interpretation module can mostly separate the phosphorylation originating from different kinase groups. However, the interpretation module is not expected to make perfect separation due to the following reasons: (1) Scansite is a tool to predict phosphorylation based on the motif, which does not represent ground-truth, and (2) the interpretation module is developed to identify the important amino acid for prediction and not for motif discovery. To further understand the results from the interpretation module, we performed clustering on the saliency scores, such that different patterns can be separated. K-means clustering was applied, and the number of clusters (17) was determined by the elbow methods (Figure S3A). The clustering results are shown in Figure 3B colored by clusters. We used the cluster center as a representation of the clusters (Figures 3C and S3) and gathered the sequence from the corresponding cluster to create a frequency plot. We found several clusters with an obvious consensus sequence pattern. Correspondingly, the interpretation module is able to highlight those positions. For example, Figure 3C shows cluster 0's saliency scores where, at the −3 position, saliency score reached the peak, indicating that MIND-S considered the −3 position as important. We then investigated the sequence pattern and found an enrichment of arginine at −3 position, indicating that the arginine there is important for phosphorylation. We also found other matching patterns (+1 proline, −2 arginine, −2 and 3 arginine), suggesting that MIND is able to capture the sequence consensus pattern hidden in the input even though we did not explicitly design a module to detect the consensus pattern. Other patterns exist in the clusters, while not exactly matching the enrichment of amino acid, which suggests that MIND-S has other ways in addition to the consensus sequence for making a prediction. Last, we also show one specific example of saliency scores that exhibit the same trend as the phosphorylation motif. MIND-S correctly predicted the phosphorylation site on protein P04150 site 203 (glucocorticoid receptor), which falls in a *CDK1* motif. We compared the consensus sequence frequency of the *CDK1* motif with the saliency scores of the flanking amino acids (Figure 3D). The interpretation module detects that the proline on position +1 is important for phosphorylation, which is in accord with the pattern shown in the kinase motif, where position +1 is a highly conserved proline. Similar analysis can be performed on any predictions made by MIND-S for users pursuing details of the prediction.

MIND-S examinates SNP effects on PTMs

Dysregulation of PTMs could potentially lead to disease, and the identification of the disease mechanism is of vital importance. Non-synonymous mutation can interrupt the recognition of the corresponding enzyme responsible for PTM addition and may therefore interrupt the PTM occurrence.[25,26] Although genome-wide association studies (GWAS) have associated genetic variants with various traits, including disease phenotypes, less has been done to investigate associations between SNP and PTM. This may be due to the lack of coupling datasets from the two modalities. MIND-S is able to identify SNP candidates that affect PTM occurrences without requiring such datasets. We demonstrate two use scenarios here: in scenario 1, if a PTM is given, to predict whether an SNP will interfere with a known PTM (identified experimentally); in scenario 2, if a PTM is not known, to predict the change of the PTM landscape. We demonstrated scenario 1 with 1,054 non-synonymous cardiac-related SNPs that are proximal to PTMs (within five amino acids; limits to four common PTMs: phosphorylation, methylation, ubiquitination, and sumoylation) retrieved from PhosphoSitePlus PTMVar.[27] The protein sequence was mutated *in silico* based on the SNPs and input to MIND-S. The prediction score of the proximal PTM was compared against the one from the unmutated protein sequence (Figure S4A). In total, 51 SNPs that change the PTM prediction from positive to negative (Table S3) with a stringent criterion (wild-type prediction score >0.8 and mutation prediction score <0.2) For example, SNP R272C on myosin-binding protein C (Uniprot: Q14896), is an SNP found in hypertrophic cardiomyopathy and is responsible for a decrease in the phosphorylation level on the protein.[28] MIND-S examined this SNP and revealed a change in the score of phosphorylation on site 275S from 0.986 to 0.0031. This is also in accord with decreased phosphorylation level of myosin-binding protein C in heart failure.[29] The other SNP, P251S, on potassium voltage-gated channel subfamily H member 2 (Uniprot: Q12809), is a mutation found in long-QT syndrome.[30] MIND-S detects this SNP to interrupt the phosphorylation on site 250S, with the score dropping from 0.898 to 0.016. This may suggest a potential role of the mutation. We next demonstrated the use case when PTM information is not known in advance. Similar to scenario one, the original protein sequence and mutated protein sequences are both used as input to MIND-S. Instead of making prediction on each single PTM, MIND-S makes predictions on every amino acid that can be targeted by PTM, which generates PTM maps for both original protein and mutated protein. Providing a PTM map can not only examine PTM that may be distal but also discover PTM that might be promoted by SNP. We demonstrate the effects of both interference and promotion of SNPs on protein leucine-rich repeat kinase 2 (*LRRK2*), a protein associated with familial and sporadic Parkinson disease (PD) and also shown to be associated with cardiac diseases.[31] Eleven SNPs on *LRRK2* were retrieved from UniProt and examined, and four of them are found to potentially affect PTMs. SNP R1441C was predicted to interfere with the phosphorylation on site 1444 with the prediction score changed from 0.972 to 0.154. Such interference has been reported[26] and therefore provides validation on the method. On the other hand, the effect of promoting a PTM is found in SNP R1628P



(legend on next page)

on phosphosite 1627 with the prediction score changed greatly from 1.45e−3 to 0.912 (Figure 4A). We visualized the PTM map of the wild-type and mutant *LRRK2* in Figure 4B. Through comparisons of PTM maps between wild type and mutant, full-length protein effects of SNP can be examined and protein-wide distribution of PTMs over the protein sequence can be comprehensively viewed. For example, in SNP R1441C, in total, two phosphorylations were predicted to be interfered and carbonylation on cysteine is predicted to be promoted. In summary, MIND-S can effectively examine the effect of protein mutation from a PTM perspective.

Other use cases of MIND-S

Furthermore, we demonstrate MIND-S's usage by providing several use cases in cardiovascular research. while similar approaches can be adapted to other research fields as well. MIND-S is able to make high-throughput predictions on unannotated proteins. We chose pig cardiac proteome for prediction because of its high research value in cardiac disease modeling but relatively few PTM annotations.[32,33] MIND-S predicted PTMs on the pig cardiac proteome from text mining (unpublished data), which consists of 7,016 proteins where 6,596 of them have no PTM reported. MIND-S identified 48,841 PTMs with high confidence (prediction score >0.8) as a pig cardiac PTMome. MIND-S can also be utilized as an approach to determine the exact PTM location. Some experimental approaches, such as antibody-based approaches, can confirm the existence of PTMs while being unable to determine the exact location of PTM on the protein. MIND-S can serve as a follow-up analysis that provides putative locations. Pyruvate dehydrogenase complex (PDH) is reported to be modified by O-linked-N-acetylglucosamine (O-GlcNAc) in mice, while the sites were not determined.[34] We used MIND-S to identify the glycosylation sites and found three O-GlcNAc sites: Uniprot: P35486 site 232, Uniprot: P35486 site 300, and Uniprot: Q8BKZ9 site 200. Uniprot: P35486 is a pyruvate dehydrogenase E1 component subunit alpha, somatic form. in mitochondria. and both sites 232 and 300 can be modified by kinase for phosphorylation. This may suggest a crosstalk between glycosylation and phosphorylation. Uniprot: Q8BKZ9 is a pyruvate dehydrogenase protein X component in mitochondria, where site 200 falls in the peripheral subunit-binding (PSBD) domain. PSBD domain, consisting of ∼35 residues, binds to the E1 or E3 subunit of PDH. This suggests the regulatory role glycosylation may play in regulating the PDH functionality.

DISCUSSION

In this report, we describe a PTM prediction schema with its coupled modeling method, MIND-S. Most existing PTM tools are based on local amino acids spanning the target sites.[2] Specifically, several amino acids flanking the PTM site are taken as the input, with predictions on the target residual as the output. Several major limitations exist in this approach (discussed below). Our workflow with MIND-S has overcome these limitations. We have applied a strategy to train and predict at the protein level, which provides a much larger receptive field to the model and relieves the burden of tuning window size. Moreover, it converts the single-site, single-PTM prediction task to a multiple-site and multiple-PTM prediction task, allowing the features learned to be shared across PTM types and improving training and predicting efficiency. In addition, reframing the peptide-level question into a protein-level question opened up opportunities for us to address the question on integration with other protein-level features and/or tasks available. One important element in our workflow architect design is the application of GNN to overcome the challenges on the integration of protein structure with protein sequence. This was not trivial since protein structure data are 3D data, whereas the protein sequence is 1D. We employed GNN to model the protein structure as a contact map, which provided spatial closeness relationship between any pairs of amino acids. We demonstrated that integration of GNN offered new information and enhanced the prediction performance. We believe that, with the growing computing power and rapid development of deep learning, modeling at the protein level will make the model interoperable among different applications involving proteins in the future.

Over the past decade, many pioneer studies contributed significantly to the growing field of machine learning applications to decode PTMs.[2,35,36] One popular area is the amino acid recognition-domain-based PTM predictions. This direction has offered important information, e.g., associated kinases, to the targeted sequences.[18,37] However, they also bear several limitations: (1) information outside of the flanking region is often lost. Short flanking sequences may not be able to capture longer sequence information or protein-level information. For example, docking sites on the substrate increase the binding affinity of the kinase for the substrate, which increases the likelihood of phosphorylation.[38] Docking sites can be far away from the phosphosites and would be missed if only local flanking sequences were considered. (2) Training and predicting are inefficient when input amino acid sequences are overlapping. For PTM prediction, both training and predicting will be done on a large scale given the large amount of existing PTM data and proteins with no PTM annotation. In addition, methods such as deep neural networks are time-costly on training. These require a less redundant dataset, while overlapping flanking sequences create redundancy in both training and prediction processes. (3) Different PTM types may have distinct

Figure 4. MIND-S examines the effect of SNP on *LRRK2* PTM

(A) An illustration of SNPs interrupting or promoting PTM occurrences on a particular molecule, *LRRK2*. SNP R1441C on protein *LRRK2* is found to have reduced phosphorylation scores on site 1444 from 0.972 to 0.754 and site 1445 from 0.778 to 0.227. SNP R1628P is found to have an elevated score of phosphorylation on site 1627 from 2.6e-4 to 0.903.

(B) PTM maps of wild-type and two mutant *LRRK2*. PTM types are annotated. The mutation amino acid (aa) is highlighted by the black triangles on the x axis. The area affected is shown with a white background; the major changes in the PTM prediction score are indicated by black arrows. In the R1441C mutant, two phosphorylation sites are interrupted, and an O-PTM on cysteine is promoted. In the R1628P mutant, one phosphorylation site is promoted.

See also Table S3.

optimal sizes of the flanking sequence; the fixed window size of the flanking sequence may limit the model's ability to transfer between PTM types. In addition, studies of PTM motifs are limited, which makes it difficult to determine the optimal size by existing biological knowledge.

With the above considerations, we created MIND-S using a deep-learning method to perform the prediction under the schema. While machine learning approaches such as support vector machines and random forest have been applied to PTM site prediction,[4,18,39–44] these methods often rely heavily on engineered features such as amino acid composition profiles, position-specific scoring matrix profiles, and surface accessibility. These engineered features are computationally expensive to build, store, and predict, and are often unavailable. On the other hand, while protein sequences are widely available, they are difficult to encode as numeric values to be "machine-readable." Compared with conventional machine learning approaches, deep-learning methods can accept a wider range of raw input, such as sequence data and graph data. Features important for prediction are thus implicitly extracted and utilized, relieving the feature extraction burden and enhancing performance.[5] Various neural networks have been proposed to process sequence data in the field of natural language processing (NLP).[45] However, model architectures that succeed in general NLP tasks may not be generalizable to tasks directed toward protein amino acid sequence. For example, the amino acid sequence comprising a protein is usually much longer than a natural language sentence, the "vocabulary" of protein sequences (i.e., 20 common amino acids) is much less complex than the word dictionary, but the amino acid sequence order of a given protein offers important insights.

We applied an LSTM layer to effectively deliver sequential information instead of using fixed positional encoding, such that the sequential information can be represented in a way that the model can best utilize. As for protein structure data, as the number of experimental determined structure data grows and computational methods for structure prediction improve, various methods are becoming available to model the protein structure data with deep learning.[46–48] Here we utilized the AlphaFold DB as one example to illustrate the utilities of structure data. AlphaFold DB has an excellent coverage of protein structure, and we converted the structure to a protein contact map to adapt for GNNs. We observed benefits from incorporating structure information for enhanced PTM prediction. In addition, we provide another version, MIND, that takes only protein sequence as the input as an alternative for proteins without reliable structure data. We also examined strategies in addition to model architecture, such as multi-label learning and bootstrap methods, which can considerably enhance MIND-S's ability to accurately predict PTMs, demonstrating the need for consideration from both computational elements as well as protein features and for developing methods on PTM predictions. The architecture design of the model is mostly driven by functionality. Specifically, the embedding layer is to vectorize the protein sequence, the bidirectional-LSTM layer is to encode positional information, the multi-head self-attention layer is to process the sequence data, the graph attention layer is to process the structure data, and the final fully connected neural network is to construct the embedding above to multi-label output.

By model design, MIND-S is able to process arbitrarily long protein sequences as input, but we truncated protein sequences when they were at excessive length due to practical restrictions. First, protein sequences need to be padded to the same length as the longest sequence in the batch, even though the padding does not provide any meaningful information; second, computational memory cost is quadratic to the protein length. Thus, we split the protein into sufficiently long subsequences to ensure memory efficiency and still allow the model learning on long-distance interactions between amino acids. We also developed an approach to split the sequence to ensure the interaction between amino acids falls into two subsequences that will not be lost. While such restriction can be loosened if preferred during inference time, e.g., when only a few proteins are investigated, full-length proteins can be used as input since parameters in MIND-S are independent of protein length.

MIND-S also provides a way to evaluate the contribution from individual residuals to the final prediction. Although deep learning is powerful on complicated tasks, the internal decision-making process is complex and less understood by humans. We use integrated gradients to simplify the interpretation process from tracking complicated decision-making processes to calculating saliency scores associated with every residual, which is easier to be understood by humans. Overall, three important features define a good model on PTM predictions: it implicitly detects innate patterns, it makes reasonable predictions with effective model interpretation, and it will unveil underlying patterns in a human-understandable fashion. Specifically, we considered two types of mechanism insights: (1) for a biologist who is interested in a specific PTM, MIND-S can point out the amino acids that might be important for the occurrence of that PTM. Thus, further experimental investigations can be performed on those prioritized amino acids instead of every amino acid in that protein. (2) When predictions and interpretations of PTM are performed on a proteome scale, the results can be treated as a database to mine the recognition pattern. For example, our analysis of phosphorylation recognition pattern not only discovered known recognition patterns but also revealed recognition patterns that have not yet been found. Different from regular motif-finding tools, which mine patterns from sequences bearing PTM (positive case), MIND-S utilized both positive and negative PTM cases (sequence without PTM site) to identify amino acids that are essential to the PTM occurrence. Thus, MIND-S provides a perspective on mining recognition/modification patterns.

Last, we showed several use cases of MIND-S. MIND-S is capable of studying the functionality of SNPs by identifying SNPs that disrupt PTM occurrences. We prioritized the SNPs most likely to change the PTM occurrence from a large pool of disease-associated SNPs. In this study, only mutations in the protein sequence due to SNPs are considered; however, any mutational processes that result in a mutant protein can be studied with MIND-S. For example, RNA splicing of introns and exons results in multiple different isoforms of the same protein and therefore can affect the occurrence of a PTM on a given site.

Taken together, our tool empowers researchers to understand how sequence variation can affect downstream biological processes and their PTM landscape.

Limitations of the study

We envision the following important tasks will further elevate the performance and broaden the applications of MIND-S: (1) improve the structure modeling to better present spatial information for PTM predictions, (2) incorporate additional PTM types to further benefit from multi-label training and allow broader usage, (3) further investigate results from saliency scores to mine patterns for PTM occurrences, and (4) evaluate effects of biological processes altering protein sequence (e.g., RNA splicing) from a PTM perspective.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Dataset
  - Model architecture
  - Fully connected neural network layer
  - Bidirectional LSTM layer
  - Multi-head self-attention
  - Model training
  - Model comparisons
  - Amino acid embedding
  - Saliency scores
  - SNP PTM association
- QUANTIFICATION AND STATISTICAL ANALYSIS

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.crmeth.2023.100430.

ACKNOWLEDGMENTS

We thank Dylan Steinecke and Eric Zhang for their support and helpful discussions. This work was supported by NIH R35 HL135772 to P.P., Y.Y., and D.C.M.N.; NIH T32 HL139450 to A.R.P and D.S.; NIH R01 HL146739 to Y.Y., D.W., and D.C.M.N.; NSF NRT 1829071 to A.R.P.; and the TC Laubisch Endowment to P.P. at UCLA.

AUTHOR CONTRIBUTIONS

Conceptualization, Y.Y.; methodology Y.Y., J.-Y.J., M.F., and W.W.; software, Y.Y. and J.-Y.J.; formal analysis, Y.Y.; investigation, Y.Y., J.-Y.J., W.W., and P.P.; resources, W.W. and P.P.; data curation, Y.Y.; writing – original draft, Y.Y.; writing – review & editing, Y.Y., J.-Y.J., M.F., D.W., D.S., A.R.P., D.C.M.N., W.W., and P.P.; visualization, Y.Y., M.F., D.W., and P.P.; supervision, W.W. and P.P.; project administration, Y.Y.; funding acquisition, P.P.

DECLARATION OF INTERESTS

The authors declare no competing interests.

INCLUSION AND DIVERSITY

We support inclusive, diverse, and equitable conduct of research.

Received: October 7, 2022
Revised: January 19, 2023
Accepted: February 24, 2023
Published: March 27, 2023

REFERENCES

1. Knorre, D.G., Kudryashova, N.V., and Godovikova, T.S. (2009). Chemical and functional aspects of posttranslational modification of proteins. Acta Naturae *1*, 29–51.
2. Ramazi, S., and Zahiri, J. (2021). Post-translational modifications in proteins: resources, tools and prediction methods. Database *2021*, baab012. https://doi.org/10.1093/database/baab012.
3. Olsen, J.V., and Mann, M. (2013). Status of large-scale analysis of post-translational modifications by Mass Spectrometry. Mol. Cell. Proteomics *12*, 3444–3452. https://doi.org/10.1074/mcp.O113.034181.
4. Chen, X., Qiu, J.-D., Shi, S.-P., Suo, S.-B., Huang, S.-Y., and Liang, R.-P. (2013). Incorporating key position and amino acid residue features to identify general and species-specific Ubiquitin conjugation sites. Bioinformatics *29*, 1614–1622. https://doi.org/10.1093/bioinformatics/btt196.
5. Naseer, S., Hussain, W., Khan, Y.D., and Rasool, N. (2021). Optimization of serine phosphorylation prediction in proteins by comparing human engineered features and deep representations. Anal. Biochem. *615*, 114069. https://doi.org/10.1016/j.ab.2020.114069.
6. Ubersax, J.A., and Ferrell, J.E., Jr. (2007). Mechanisms of specificity in protein phosphorylation. Nat. Rev. Mol. Cell Biol. *8*, 530–541. https://doi.org/10.1038/nrm2203.
7. Needham, E.J., Parker, B.L., Burykin, T., James, D.E., and Humphrey, S.J. (2019). Illuminating the dark phosphoproteome. Sci. Signal. *12*, eaau8645. https://doi.org/10.1126/scisignal.aau8645.
8. Suzek, B.E., Wang, Y., Huang, H., McGarvey, P.B., and Wu, C.H.; UniProt Consortium (2015). UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics *31*, 926–932. https://doi.org/10.1093/bioinformatics/btu739.
9. Ozenne, B., Subtil, F., and Maucort-Boulch, D. (2015). The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. J. Clin. Epidemiol. *68*, 855–859. https://doi.org/10.1016/j.jclinepi.2015.02.010.
10. Ling, C.X., Huang, J., and Zhang, H. (2003). AUC: A statistically consistent and more discriminating measure than accuracy. In IJCAI'03: Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp. 519–524.
11. Dou, L., Yang, F., Xu, L., and Zou, Q. (2021). A comprehensive review of the imbalance classification of protein post-translational modifications. Brief. Bioinform. *22*, bbab089. https://doi.org/10.1093/bib/bbab089.
12. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural Comput. *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (Curran Associates, Inc.).
14. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. Preprint at arXiv. https://doi.org/10.48550/arXiv.1710.10903.

15. Murphy, K.P. (2012). Machine Learning: A Probabilistic Perspective (MIT Press).
16. Dufter, P., Schmitt, M., and Schütze, H. (2021). Position information in transformers: An overview. Preprint at arXiv. https://doi.org/10.48550/arXiv.2102.11090.
17. Sagi, O., and Rokach, L. (2018). Ensemble learning: a survey. WIREs Data Mining Knowl. Discov. *8*, e1249. https://doi.org/10.1002/widm.1249.
18. Gao, J., Thelen, J.J., Dunker, A.K., and Xu, D. (2010). Musite, a tool for global prediction of general and kinase-specific phosphorylation sites. Mol. Cell. Proteomics *9*, 2586–2600. https://doi.org/10.1074/mcp.M110.001388.
19. Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. Preprint at arXiv. https://doi.org/10.48550/arXiv.1703.01365.
20. Chou, M.F., and Schwartz, D. (2011). Biological sequence motif discovery using motif-x. CP. in Bioinformatics *35*, 13. https://doi.org/10.1002/0471250953.bi1315s35.
21. Gavel, Y., and von Heijne, G. (1990). Sequence differences between glycosylated and non-glycosylated Asn-X-Thr/Ser acceptor sites: implications for protein engineering. Protein Eng. *3*, 433–442. https://doi.org/10.1093/protein/3.5.433.
22. Pearson, R.B., and Kemp, B.E. (1991). [3] Protein kinase phosphorylation site sequences and consensus specificity motifs: tabulations. In Methods in Enzymology Protein Phosphorylation Part A: Protein Kinases: Assays, Purification, Antibodies, Functional Analysis, Cloning, and Expression (Academic Press), pp. 62–81. https://doi.org/10.1016/0076-6879(91)00127-I.
23. Pinna, L.A., and Ruzzene, M. (1996). How do protein kinases recognize their substrates? Biochim. Biophys. Acta *1314*, 191–225. https://doi.org/10.1016/S0167-4889(96)00083-3.
24. Obenauer, J.C., Cantley, L.C., and Yaffe, M.B. (2003). Scansite 2.0: proteome-wide prediction of cell signaling interactions using short sequence motifs. Nucleic Acids Res. *31*, 3635–3641. https://doi.org/10.1093/nar/gkg584.
25. Martin, D.D.O., Kay, C., Collins, J.A., Nguyen, Y.T., Slama, R.A., and Hayden, M.R. (2018). A human huntingtin SNP alters post-translational modification and pathogenic proteolysis of the protein causing Huntington disease. Sci. Rep. *8*, 8096. https://doi.org/10.1038/s41598-018-25903-w.
26. Manschwetus, J.T., Wallbott, M., Fachinger, A., Obergruber, C., Pautz, S., Bertinetti, D., Schmidt, S.H., and Herberg, F.W. (2020). Binding of the human 14-3-3 isoforms to distinct sites in the leucine-rich repeat kinase 2. Front. Neurosci. *14*, 302.
27. Hornbeck, P.V., Zhang, B., Murray, B., Kornhauser, J.M., Latham, V., and Skrzypek, E. (2015). PhosphoSitePlus, 2014: mutations, PTMs and recalibrations. Nucleic Acids Res. *43*, D512–D520. https://doi.org/10.1093/nar/gku1267.
28. Ehlermann, P., Weichenhan, D., Zehelein, J., Steen, H., Pribe, R., Zeller, R., Lehrke, S., Zugck, C., Ivandic, B.T., and Katus, H.A. (2008). Adverse events in families with hypertrophic or dilated cardiomyopathy and mutations in the MYBPC3gene. BMC Med. Genet. *9*, 95. https://doi.org/10.1186/1471-2350-9-95.
29. El-Armouche, A., Pohlmann, L., Schlossarek, S., Starbatty, J., Yeh, Y.-H., Nattel, S., Dobrev, D., Eschenhagen, T., and Carrier, L. (2007). Decreased phosphorylation levels of cardiac myosin-binding protein-C in human and experimental heart failure. J. Mol. Cell. Cardiol. *43*, 223–229. https://doi.org/10.1016/j.yjmcc.2007.05.003.
30. Napolitano, C., Priori, S.G., Schwartz, P.J., Bloise, R., Ronchetti, E., Nastoli, J., Bottelli, G., Cerrone, M., and Leonardi, S. (2005). Genetic testing in the long QT SyndromeDevelopment and validation of an efficient approach to genotyping in clinical practice. JAMA *294*, 2975–2980. https://doi.org/10.1001/jama.294.23.2975.
31. Liu, Y., Hao, C., Zhang, W., Liu, Y., Guo, S., Li, R., Peng, M., Xu, Y., Pei, X., Yang, H., and Zhao, Y. (2022). Leucine-rich repeat kinase-2 deficiency protected against cardiac remodelling in mice via regulating autophagy formation and degradation. J. Adv. Res. *37*, 107–117. https://doi.org/10.1016/j.jare.2021.07.004.
32. Schüttler, D., Tomsits, P., Bleyer, C., Vlcek, J., Pauly, V., Hesse, N., Sinner, M., Merkus, D., Hamers, J., Kääb, S., and Clauss, S. (2022). A practical guide to setting up pig models for cardiovascular catheterization, electrophysiological assessment and heart disease research. Lab Anim. *51*, 46–67. https://doi.org/10.1038/s41684-021-00909-6.
33. Gabriel, G.C., Devine, W., Redel, B.K., Whitworth, K.M., Samuel, M., Spate, L.D., Cecil, R.F., Prather, R.S., Wu, Y., Wells, K.D., and Lo, C.W. (2021). Cardiovascular development and congenital heart disease modeling in the pig. J. Am. Heart Assoc. *10*, e021631. https://doi.org/10.1161/JAHA.121.021631.
34. Li, T., Zhang, Z., Kolwicz, S.C., Abell, L., Roe, N.D., Kim, M., Zhou, B., Cao, Y., Ritterhoff, J., Gu, H., et al. (2017). Defective branched-chain amino acid (BCAA) catabolism disrupts glucose metabolism and sensitizes the heart to ischemia-reperfusion injury. Cell Metab. *25*, 374–385. https://doi.org/10.1016/j.cmet.2016.11.005.
35. Mm, H., and Khatun, M.S. (2017). Prediction of protein Post-Translational Modification sites: an overview. Ann. Proteom. Bioinform. *2*, 049–057. https://doi.org/10.29328/journal.apb.1001005.
36. Audagnotto, M., and Dal Peraro, M. (2017). Protein post-translational modifications: in silico prediction tools and molecular modeling. Comput. Struct. Biotechnol. J. *15*, 307–319. https://doi.org/10.1016/j.csbj.2017.03.004.
37. Xu, Y., Song, J., Wilson, C., and Whisstock, J.C. (2018). PhosContext2vec: a distributed representation of residue-level sequence contexts and its application to general and kinase-specific phosphorylation site prediction. Sci. Rep. *8*, 8240. https://doi.org/10.1038/s41598-018-26392-7.
38. Sharrocks, A.D., Yang, S.-H., and Galanis, A. (2000). Docking domains and substrate-specificity determination for MAP kinases. Trends Biochem. Sci. *25*, 448–453. https://doi.org/10.1016/S0968-0004(00)01627-3.
39. Chauhan, J.S., Rao, A., and Raghava, G.P.S. (2013). In silico platform for prediction of N-O- and C-glycosites in eukaryotic protein sequences. PLoS One *8*, e67008. https://doi.org/10.1371/journal.pone.0067008.
40. Deng, W., Wang, C., Zhang, Y., Xu, Y., Zhang, S., Liu, Z., and Xue, Y. (2016). GPS-PAIL: prediction of lysine acetyltransferase-specific modification sites from protein sequences. Sci. Rep. *6*, 39787. https://doi.org/10.1038/srep39787.
41. Zhao, Q., Xie, Y., Zheng, Y., Jiang, S., Liu, W., Mu, W., Liu, Z., Zhao, Y., Xue, Y., and Ren, J. (2014). GPS-SUMO: a tool for the prediction of sumoylation sites and SUMO-interaction motifs. Nucleic Acids Res. *42*, W325–W330. https://doi.org/10.1093/nar/gku383.
42. Wei, L., Xing, P., Shi, G., Ji, Z., and Zou, Q. (2019). Fast prediction of protein methylation sites using a sequence-based feature selection technique. IEEE/ACM Trans. Comput. Biol. Bioinform. *16*, 1264–1273. https://doi.org/10.1109/TCBB.2017.2670558.
43. Ismail, H.D., Newman, R.H., and Kc, D.B. (2016). RF-Hydroxysite: a random forest based predictor for hydroxylation sites. Mol. Biosyst. *12*, 2427–2435. https://doi.org/10.1039/C6MB00179C.
44. Ren, J., Wen, L., Gao, X., Jin, C., Xue, Y., and Yao, X. (2008). CSS-Palm 2.0: an updated software for palmitoylation sites prediction. Protein Eng. Des. Sel. *21*, 639–644. https://doi.org/10.1093/protein/gzn039.
45. Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning (MIT Press).
46. Li, J., Luo, S., Deng, C., Cheng, C., Guan, J., Guibas, L., Peng, J., and Ma, J. (2022). Directed weight neural networks for protein structure representation learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.2201.13299.
47. Fasoulis, R., Paliouras, G., and Kavraki, L.E. (2021). Graph representation learning for structural proteomics. Emerg. Top. Life Sci. *5*, 789–802. https://doi.org/10.1042/ETLS20210225.
48. Wang, Z., Combs, S.A., Brand, R., Calvo, M.R., Xu, P., Price, G., Golovach, N., Salawu, E.O., Wise, C.J., Ponnapalli, S.P., and Clark, P.M.

(2022). LM-GVP: an extensible sequence and structure informed deep learning framework for protein property prediction. Sci. Rep. *12*, 6832. https://doi.org/10.1038/s41598-022-10775-y.

49. Lau, E., Cao, Q., Lam, M.P.Y., Wang, J., Ng, D.C.M., Bleakley, B.J., Lee, J.M., Liem, D.A., Wang, D., Hermjakob, H., and Ping, P. (2018). Integrated omics dissection of proteome dynamics during cardiac remodeling. Nat. Commun. *9*, 120. https://doi.org/10.1038/s41467-017-02467-3.

50. UniProt Consortium (2021). UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Res. *49*, D480–D489. https://doi.org/10.1093/nar/gkaa1100.

51. Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and de Hoon, M.J.L. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics *25*, 1422–1423. https://doi.org/10.1093/bioinformatics/btp163.

52. Grattarola, D., and Alippi, C. (2020). Graph neural networks in TensorFlow and Keras with Spektral. Preprint at arXiv. https://doi.org/10.48550/arXiv.2006.12138.

53. Lin, M.M., and Zewail, A.H. (2012). Hydrophobic forces and the length limit of foldable protein domains. Proc. Natl. Acad. Sci. USA *109*, 9851–9856. https://doi.org/10.1073/pnas.1207382109.

54. Johnson, J.M., and Khoshgoftaar, T.M. (2019). Survey on deep learning with class imbalance. J. Big Data *6*, 27. https://doi.org/10.1186/s40537-019-0192-5.

55. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., et al. (2018). Scikit-learn: Machine learning in Python. Preprint at arXiv. https://doi.org/10.48550/arXiv.1201.0490.

56. Crooks, G.E., Hon, G., Chandonia, J.-M., and Brenner, S.E. (2004). WebLogo: a sequence logo generator. Genome Res. *14*, 1188–1190. https://doi.org/10.1101/gr.849004.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
PTM benchmark dataset	This manuscript	https://doi.org/10.5281/zenodo.7655709
O-PTM dataset	This manuscript	https://doi.org/10.5281/zenodo.7655827
Pig cardiac proteome	This manuscript	https://doi.org/10.5281/zenodo.7655835
Predicted protein structure	AlphafoldDB	https://alphafold.ebi.ac.uk/
PTMVar	PhosphoSitePlus	https://www.phosphosite.org/homeAction
Protein sequence	Uniprot	https://www.uniprot.org/
Software and algorithms		
MIND-S, MIND	This manuscript	https://doi.org/10.5281/zenodo.7659116
MusiteDeep	MusiteDeep	https://www.musite.net/
Scansite	Scansite	https://scansite4.mit.edu/#home
WebLogo	WebLogo	https://weblogo.berkeley.edu/logo.cgi

RESOURCE AVAILABILITY

Lead contact

Further requests for resources should be directed to and will be fulfilled by the Lead Contact, Peipei Ping (pping38@g.ucla.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- Training datasets and pig cardiac PTMome results have been deposited in Zenodo and are publicly available as of the date of publication. The DOI are listed in the key resources table.
- All original code has been deposited on GitHub as well as Zenodo and is publicly available as of the date of publication. The DOI is listed in the key resources table.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Dataset

Two separate large PTM datasets encompassing a total of 26 PTM types, including 50,000 + proteins with 260,000 + total PTMs, are employed in this study. First, we selected the PTM dataset previously published by MusiteDeep[18] (09/2021), where 13 PTM types were included. For training/validation/testing split, to prevent information leakage from similar proteins, we applied Uniref. 50 [8], where protein in the clusters has at least 50% sequence identify to and 80% overlap with the longest sequence in the cluster; during splitting, we enforced proteins from the same Uniref. 50 cluster go into the same split. The 13 PTM types are detailed in Table S1.

The second dataset is an Oxidative PTM (O-PTM)-centric dataset we have collected in-house combined with a publicly available dataset.[49] In short, O-PTM was searched from MS raw file through IP2 program, and 13 types of O-PTMs are included in this study. Similarly, we applied Uniref. 50 [8] to guide the splitting. A summary of the O-PTM data in each split is shown in Table S2.

An amino acid residue with/without a PTM will be treated as a positive/negative label for that PTM, respectively. Both positive and negative labels were utilized to train our model. An example is shown in Figure S5.

In addition, we only considered negative PTM when there is at least one positive PTM in the same protein; for example, if a protein has one phosphorylation on serine or threonine, all other serine and threonine that do not bear phosphorylation will be treated as negative samples; if the same protein has no ubiquitination, all lysine will neither be treated as positive or negative labels for ubiquitination. This is to ensure the integrity of the negative samples, since a protein with no positive PTM may indicate no PTM identification experiment has been performed on that protein.

Protein sequences were downloaded from the UniProt website (https://www.uniprot.org/)[50] by UniProt ID.

Protein structures were downloaded from AlphaFoldDB by UniProt ID from Google Cloud Public Datasets, with name as "AF-UID-F1-model_v3.cif". In total, 38,947 proteins have predicted structure from AlphaFold. We used Biopython[51] to parse the protein structure and built the contact map. Specifically, model 0 and chain A of each protein was used, "CA" atom in each amino acid was used to calculate the pairwise distance between amino acids. From the pairwise distance matrix, we filtered out amino acid pairs with distance greater than 10 Å and binarized it as our contact map. We regarded each amino acid is close to itself.

Model architecture

The MIND-S architecture consists of one embedding layer, one bidirectional LSTM layer, three multi-head self-attention blocks, one graph attention layer and one fully connected layer. The embedding layer converts protein sequence to an embedding of the size of 128 through a feedforward dense layer. The bidirectional LSTM layer has a dimension of 64 for each direction. Tanh activation is used for cell and hidden state; sigmoid activation is used for gate activation. After the LSTM layer, a dropout layer is added. The feature vector from the LSTM layer is passed to the multi-head self-attention block. The multi-head self-attention block consists of a multi-head self-attention layer, a dropout layer, a layer normalization layer, two feedforward dense layers, another dropout layer, and another normalization layer in sequential. Graph attention layer used multi-head attention with the number of heads equals to 8 and a dropout rate equals to 0.5. Lastly, output from the multi-head self-attention block and graph attention layer will be concatenated at the last dimension and passed to a feedforward neural network with an output dimension of 13 (number of PTM types) for each amino acid. Sigmoid activation is applied to output a final score. Unless specified otherwise, all layers have 128 hidden dimensions and the dropout score is set to 0.1. The model is built using Tensorflow 2.0, Keras API and spektral.[52] The detail of each layer is descripted below.

Fully connected neural network layer

$$a = relu(WX + b)$$

where X is the input matrix, W is the weight matrix, b is the bias term, $relu$ is the rectified linear activation function, a is the output of the layer.

Bidirectional LSTM layer

Hidden states of each amino acid in LSTM are calculated following the sequential order:

$$f_t = sigmoid(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = sigmoid(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = sigmoid(W_o x_t + U_o h_{t-1} + b_o)$$

$$c'_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet c'_t$$

$$h_t = o_t \bullet \tanh(c_t)$$

where x_t is the input of the t-th amino acid input embedding, h_{t-1} is the $t-1$-th hidden state, h_{t-1} is the $t-1$-th hidden state, f_t is the t-th forget gate, i_t is the t-th input gate, o_t is the t-th output gate, c_t is the t-th cell state, W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c are the weight matrices and b_f, b_i, b_o, b_c are the biases.

Bidirectional LSTM are combined by LSTM from N-terminal to C-terminal and LSTM from C-terminal to N-terminal:

$$Out_t = Concat(h_t^f, h_t^b)$$

where Out_t is the t-th output, h_t^f and h_t^b are hidden state from forward and backward direction respectively.

Multi-head self-attention

$$Q_i = W_{Qi} X$$

$$K_i = W_{Ki} X$$

$$V_i = W_{Vi} X$$

$$A_i = softmax\left(\frac{Q_i K_i.T}{\sqrt{d_k}}\right)$$

$$Out = Concat_{i=1}^{h}(A_i V_i)$$

where X is the input matrix, W_{Q_i}, W_{K_i}, W_{V_i} are the weight matrices to generate query matrix Q_i, key matrix K_i and value matrix V_i, respectively. i represent the i-th head. A_i is the matrix of scaled attention of the i-th head, T is the matrix transpose operation, d_k is the second dimension of matrix K_i. Out is the output from concatenating all heads.

Model training

Sequence preprocessing

The input sequence is one-hot encoded into a matrix with the shape as (length of sequence +2, 26), where 2 is the "<START>" and "<END>" added before and after the sequence; 26 is the total number of tokens: 22 amino acids (20 common amino acids plus Selenocysteine and any) and four special tokens "<OTHER>", "<PAD>", "<START>" and "<END>". '<START>' and '<END>' will be added before and after the input sequence to indicate the start and end of the sequence; "<OTHER>" will be used if the amino acid in the sequence is not in the 22 amino acid tokens mentioned earlier; "<PAD>" is used to pad the sequence to the maximum length, which is 512 amino acids with "<START>" and "<END>" tokens in our study. The padding is to batch the data for computation; we mask the attention involving padding during multi-head attention calculation by adding negative 1e3 to the corresponding attention scores before softmax, rendering the value close to 0 after softmax. A binary protein contact map is used as the adjacency matrix for the graph attention layer. To match the sequence length, the protein contact map is also padded to a dimension of (514, 514) with zeros.

We have chosen 512 amino acids as the maximum segment length with three primary considerations: first, we anticipate that a 512 amino acid long segment has sufficient length to encompass various protein domains, which are on average 100 amino acids long, typically of length between 50 and 200 amino acids.[53] Second, our multi-head self-attention and structure graph layers require quadratic computational memory with respect to length, restricting the protein segment length. Third, about 60% of the 48,811 proteins in our dataset are shorter than 512, which is not affected by the maximum length.

Thus, we selected a computation segment length of around 512 amino acids.

PTM mapping strategy

To address long sequences, we arranged proteins into extended core sequences with overlap. We set the segments with maximum length (i.e., 512), where the N terminal of the extended core sequence has 128 amino acids overlapping with the C terminal of the last extended core sequence. The PTM data falling in the core sequence (i.e., the middle 256 amino acids of the extended core sequence) were selected for training, assuring interactions within a distance of at least 128 (mostly longer) will not be lost.

Specifically, we cut the whole protein sequence Seq into extended core sequences $ECSeq_i$:

$$ECSeq_i = Seq\left[\frac{i*c}{2} : MaxSeq_i\right],$$

$$MaxSeq_i = \begin{cases} s, i = \frac{s-1}{2c} - 1 \\ \frac{(i+2)*c}{2}, i \neq \frac{s-1}{2c} - 1 \end{cases},$$

where $Seq[a:b]$ represents subsequence of Seq from position a to position $b-1$, s is the length of the sequence, c is the size of core sequence (i.e., 256).

To ensure enough context for each instance, for each $ECSeq_i$, we only consider the $CoreSeq_i$ within the positions from l to r of $ECSeq_i$, where

$$CoreSeq_i = ECSeq_i[l:r] \quad l = \begin{cases} 0, i = 0 \\ \frac{c}{4}, i \neq 0 \end{cases} \quad r = \begin{cases} s_i, i = \frac{(s-1)}{c*2} - 1 \\ \frac{3c}{4}, i \neq \frac{(s-1)}{c*2} - 1 \end{cases}$$

s_i is the length of the i-th subsequence $ECSeq_i$.

Label and sample weight preprocessing

To adapt to the multi-label setting, we constructed a label matrix with the shape as (length of sequence, the number of PTM types); where the rows correspond to the residues in the protein sequence while columns correspond to the PTM types. We used "1" to present the positive labels and "0" to represent the negative labels. The entry was "1" if/when the amino acid hosts PTM(s) or, "0" if the amino acid is naked or not the target of the PTM. We also constructed the sample weight matrix to (a) inform which PTMs to be included during training and evaluation; and (b) apply class weights during training. We weighted different PTM labels

to provide higher weight for PTM types with fewer samples to address the class imbalance issue. The sample weight matrix has the same shape as the label matrix, where entry will be weights if/when the PTM was hosted. The entry will be "1" if no weighting is applied.

Model loss

We have chosen the weighted binary cross-entropy loss for each label. Specifically,

$$Loss = -\frac{1}{\sum_{j=1}^{N} N_j}\sum_{j=1}^{N}\sum_{i=1}^{N_j}\left(y_{ij}\log\left(p_{ij}\right) * w_j + \left(1 - y_{ij}\right) * \log\left(1 - p_{ij}\right)\right)$$

where N_j is the number of samples in the class j, N is the total number of PTM classes, y_{ij} is the true label in i sample of PTM class j, p_{ij} is the output prediction scores from model in i sample of the PTM class j, w_j is the positive class weight for PTM class j.

Considering that many amino acids do not host any PTMs, we applied sample masks to retain only positive or negative samples during loss calculation. All other amino acids that are not targets of PTM will not be included in the loss calculation.

One challenge in PTM site prediction is the class imbalance issue; the number of negative samples, target residuals without PTMs, is much larger than the number of positive samples, target residuals with PTMs. To ensure the model learns from balanced data, we computed the loss with the inverse proportion of positive or negative samples as the weights.

Specifically, the weight was calculated as:

$$w_j = \frac{n_{pos_j} + n_{neg_j}}{2n_{pos_j}}$$

where n_{pos_j} is the number of positive samples in the PTM class j, n_{neg_j} is the number of negative samples in the PTM class j. The consideration of such weighting is to assign higher weights for classes with fewer data and lower weights for classes with more data. So the weight is set proportional to the inverse of the number of samples in negative or positive samples and normalized by the total number of samples.[54]

This calculation was only performed during training; no weighting was involved during evaluation.

Training settings

We set the number of epochs as 300 and batch size as 64. An early stopping strategy with patience equal to 2 was enforced, and loss was monitored for early stopping. The model will stop training after 2 epochs if there was no improvement in the loss. And the model with the least loss during the training was accepted as the final model. We utilized the Adam stochastic optimization method with the following parameters: learning rate 1e-3, the decay rate for the first moment estimate as 0.9, and exponential decay rate for the second moment estimate as 0.999. We employed the AMSgrad variant. The model evaluation metric was calculated through the scikit-learn package,[55] where the average precision score was selected to determine the AUPR. Metric using micro-average was to calculate the metric for all predictions made together, whereas macro-average calculated the metric for each PTM type first and averaged them. AUC, f1 score, and Matthews correlation coefficient (MCC) were determined through the scikit-learn package as well.

Bootstrapping was performed by splitting the dataset randomly into two separate sets iteratively, where one set was one-fifth of the total size and was ultimately used as the validation set whereas the remaining was the training set. 15 different training/validation sets were generated to train 15 models. After training, AUPR was calculated for the validation set. The AUPR scores were applied to weight the score outputted by corresponding models to ensemble a final model for each PTM type for each amino acid.

Two models were trained individually for the 13 PTMs and 13 O-PTMs as these two PTM datasets were generated from different sources.

Model comparisons

For evaluating the effects of data size, we had the testing set fixed, and randomly sampled the remaining data without replacement with a proportion of 10%, 30%, 50%, and 100%. For the sequence only model, we remove the input of protein structure and graph attention layer. For the structure only model, we removed the multi-head self-attention blocks. For the single label setting, we changed the final output of the model to one dimension and train the model for each PTM type individually. For evaluating the positional information, we constructed a model without the biLSTM layer as the no positional information model; we constructed a model without the biLSTM layer but instead we add the sinusoidal positional encoding before next layers as the sinusoidal model. MusiteDeep was trained on the combined training and validation sets and tested on the testing dataset with default settings. CNN model is constructed under our schema with protein mapped to Core sequences. The CNN model consists of four layers of 1D convolution with the size of dimension as 256, same padding, and kernel size as 3, 6, 9, and 12 respectively. A LeakyReLU layer with alpha = 0.01 and a dropout layer with probability = 0.6 are added after each 1D convolution layer. The remaining setting was identical in MIND-S without bootstrap application. RNN model is constructed similar as CNN model, instead of using CNN layers, RNN model use three layer of Bidirectional LSTM layer with identical setting as MIND without bootstrapping.

Amino acid embedding

The amino acid embeddings are generated by the embedding layer in MIND-S, where each amino acid has one corresponding embedding associated. We extracted the embeddings from 20 amino acids and used principal component analysis (PCA) to extract the first two components for visualization.

Saliency scores

Integrated gradients were selected to determine saliency scores. The integrated gradients method requires the integration of gradients from a series of interpolated values from background to actual input. However, one-hot encoding cannot be interpolated. Therefore, we applied the embedding from one-hot encoding instead to perform the interpolation. This would not interfere with the saliency attribution given that the embedding layer is unique for each amino acid. We utilized the same vectors as input for background embedding, with the exception that the corresponding embedding of the residual to be evaluated as zero. We calculated 50 interpolations between the background and the actual embedding, specifically:

A background embedding emb_0 of the amino acid residue interested is created first:

$$emb_0 = \mathbf{0}$$

where the emb_0 is a zero vector.

A series of interpolated embedding emb_i are generated from the background embedding and the original embedding emb:

$$emb_i = emb_0 + \alpha_i(emb - emb_0)$$

$$\alpha_i = \frac{1}{N_\alpha} i$$

where N_α is the total number of interpolations to be performed and emb_i is the i th embedding.

Interpolated embeddings and the original embeddings were then input into the model to arrive at the prediction. The prediction of a PTM was then determined to calculate the gradients of the interpolated inputs. The gradient of interpolated embedding emb_i is calculated as s_i

$$s_i = \frac{\partial loss}{\partial emb_i}$$

Finally, the gradients of interpolated embeddings are accumulated with trapezoidal rule and scaled with respect to input to get the final saliency vector s:

$$s = \frac{\sum_{i=1}^{N_\alpha - 1} s_i + \sum_{2}^{N_\alpha} s_i}{2}(emb - emb_0)$$

The sum of all entry of s will be used as saliency score for this amino acid residue.

Flanking sequence of length equal to 21 (including the PTM site) is used to calculate the flanking saliency scores and perform t-SNE plot. Only phosphosites that have prediction scores greater than 0.8 and have associated kinases defined were selected. Kinases are determined by scansite4[24] web service to scan protein sequences in our datasets to locate the phosphorylation motifs. t-SNE plot is generated by sklearn with default setting except that perplexity is set to 100. Clustering analysis is performed by kmeans in sklearn with the default setting and the number of clusters (17) is determined by the elbow method using the sum of squared distances of samples to the corresponding cluster center. The sequence frequency plot of the cluster was generated by aggregating the sequence of samples in that cluster by WebLogo.[56] The sequence frequency plot of kinase was generated by aggregating the substrate sequences of that kinases retrieved from PhosphoSitePlus.[27] The representative of the cluster is the cluster center from kmeans.

SNP PTM association

Human disease-associated SNPs proximal to PTM sites were downloaded from PTMVar[27] and UniProt. SNP related to cardiovascular diseases were selected for analysis. In silicon mutated protein was generated according to SNP. The prediction scores of the same PTM site were compared between the mutated and wild-type proteins. An SNP-PTM pair was set to be confident when the wild type has a prediction score higher than 0.8 and the subsequent mutation prediction score is lower than 0.2 or vice versa.

QUANTIFICATION AND STATISTICAL ANALYSIS

Statistical test was performed to compare the cross-entropy loss made by MIND and the other models. All predictions on the test set from the models were used to calculate the binary cross-entropy loss with true labels (n = 182,872 losses). We then performed a one-sided t-test (the alternative hypothesis is loss from MIND is smaller than the other model) on cross-entropy losses from MIND the other model compared. We used ttest_rel from scipy.stats in python to perform the t-test.