Title

Harnessing AI/ML for Proteomics: Post-Translational Modification Prediction and Proteome Turnover Imputation

Author

Yan, Yu

Publication Date

2025

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Harnessing AI/ML for Proteomics: Post-Translational Modification Prediction and Proteome Turnover Imputation

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Medical Informatics

by

Yu Yan

2025

ABSTRACT OF THE DISSERTATION

Harnessing AI/ML for Proteomics: Post-Translational Modification Prediction and Proteome Turnover Imputation

by

Yu Yan
Doctor of Philosophy in Medical Informatics
University of California, Los Angeles, 2025
Professor Peipei Ping, Chair

The field of proteomics, encompassing the exhaustive study of proteins, their structures, functions, post-translational modifications, dynamics, and interactions, stands as a crucial domain in the quest to understand biological systems and disease mechanisms. The rise of high-throughput technologies, notably mass spectrometry, has exponentially increased the volume and complexity of proteomic data, posing both opportunities and challenges in large-scale data analysis and interpretation. In this context, the integration of Artificial Intelligence (AI) and Machine Learning (ML) methodologies presents a transformative strategy, promising to significantly enhance various facets of data analysis in proteomics. This dissertation is dedicated to exploring the application of AI/ML in the domain of proteomics.

The first theme of this dissertation introduces MIND-S, a deep-learning platform designed to predict protein post-translational modifications (PTMs). MIND-S utilized protein sequence and structure, modeling through combination of a transformer model and a graph neural network to efficiently predict multiple PTMs. It features an interpretation module that discerns the relevance of amino acids and uncovers PTM patterns without direct supervision. Additionally, it assesses the effects of mutations on PTMs and has been validated using biological data. This work demonstrates MIND-S's accuracy and efficiency in analyzing PTM processes in both health and disease.[1]

The second theme delves into gene representation through a comprehensive, task-agnostic

approach, aiming for a holistic understanding of molecular events. Traditional gene embeddings often have a narrow focus on specific tasks, missing the broader picture. This study evaluates nine gene embeddings across three categories: experimental, literature, and knowledge graph data. Using Singular Vector Canonical Correlation Analysis (SVCCA), it reveals that the representations contain unique, minimally overlapping information, fostering rich, multifaceted embeddings. This method outperforms task-specific approaches in various benchmark tests and successfully imputes missing data, enhancing individual embeddings. It offers a robust framework for comprehensive biomolecule characterization, with significant benefits for biomedical AI applications.

The third theme addresses the challenge of missing values in temporal proteomics datasets, which can obscure critical measurements and impair the understanding of biomedical processes. To address this, a Data Multiple Imputation (DMI) pipeline was developed to facilitate robust analysis of protein turnover rates in time-series data. This approach was applied to murine cardiac and human plasma datasets, greatly improving the detection of protein turnover rates and uncovering new biological insights. The imputed data provided a more comprehensive depiction of proteins, enhancing the understanding of biological pathways and disease associations. Notably, DMI outperformed single imputation methods in benchmark evaluations, demonstrating its effectiveness in managing missing data challenges in temporal proteomics.[2]

The dissertation of Yu Yan is approved.

Wei Wang

Xinshu Xiao

Alex Anh-Tuan Bui

Peipei Ping, Committee Chair

University of California, Los Angeles

2025

To my parents and my partner

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

This dissertation represents the culmination of my efforts and the invaluable support from many individuals who have been instrumental throughout my Ph.D. journey at UCLA. I am profoundly grateful to my advisor, collaborators, family, and friends for their integral roles in this accomplishment.

First and foremost, I wish to express my deepest gratitude to my advisor, Dr. Peipei Ping. Working alongside Dr. Ping has been the most valuable and transformative experience of my academic career. Since joining her lab six years ago, she has provided unwavering support both academically and personally. Her leadership and encouragement have been crucial in helping me complete this dissertation, and I am immensely thankful for her guidance.

I am also deeply appreciative of my dissertation committee members, Dr. Alex Bui, Dr. Xinshu Grace Xiao, and Dr. Wei Wang, for their constructive feedback and insightful comments on my research. Their expertise and suggestions have significantly enhanced the quality of my work.

I extend my heartfelt thanks to all current and former members of the Ping Lab for their assistance and stimulating discussions. In particular, I would like to acknowledge Mr. Alexander Pelletier, soon to be Dr. Pelletier, for sharing this entire Ph.D. journey with me and for the mutual support we have provided each other. I am also grateful to Dr. Dominic Ng, and Dr. Dean Wang for their valuable advice throughout this endeavor.

Finally, I am deeply thankful to my friends and family for their unwavering love and support. I owe a special debt of gratitude to my mother, Xinzhi Li, and my father, Guojiang Yan, for their unconditional love throughout my life. To my friends, Xing Zhang, Shufu Shi and Wenbin Guo, thank you for your steadfast friendship. A special acknowledgment goes to my partner, Mingzhou Fu, for being a constant source of comfort and joy during both the challenging and cheerful moments.

Thank you all for your profound contributions, which have made this achievement possible.

VITA

2015–2019	B.S. in *Biological Science*, School of Life Sciences, Wuhan University, Wuhan, China.
2019–2025	Graduate Student Researcher, *Medical Informatics*, University of California, Los Angeles.

PUBLICATIONS

(* indicates equal contribution. The list includes only those works where I am the lead author during my Ph.D. studies.)

Yan, Y., Sankar B. S., Mirza B., Ng D. C., Pelletier A. R., Huang S. D., Wang W., Watson K., Wang D. Ping, P. (2024). Missing Values in Longitudinal Proteome Dynamics Studies: Making a Case for Data Multiple Imputation.*Journal of Proteome Research*, 2024, 23, 9, 4151–4162.

Yan, Y.*, Wang D., Xin R., Soriano R. A., Ng D. C. M., Wang W., and Ping, P. (2023). Protocol for the prediction, interpretation, and mutation evaluation of post-translational modification using MIND-S. *Star Protocols*. 4.4 (2023): 102682.

Yan, Y.*, Jiang J., Fu M., Wang D., Pelletier A. R., Sigdel D., Ng D. C. M., Wang W. Ping, P. (2022). MIND-S is a deep-learning prediction model for elucidating protein post-translational modifications in human diseases. *Cell reports methods*, 3.3 (2023).

CHAPTER 1

Introduction

Proteomics is the large-scale study of the entire set of proteins produced by an organism, system, or biological context, collectively known as the proteome, at a specific point in time. Unlike genomics, which examines the static blueprint of DNA, proteomics captures the dynamic and functional aspects of cellular biology by analyzing protein expression levels, post-translational modifications, interactions, and functions. This comprehensive approach enables scientists to gain deeper insights into cellular processes, disease mechanisms, and the intricate regulatory networks that sustain life[3].

Advancements in analytical technologies have been pivotal in propelling the field of proteomics forward. Mass spectrometry (MS) stands out as a cornerstone technique, facilitating the precise identification and quantification of proteins with high sensitivity and accuracy. Additionally, bioinformatics tools and software have become indispensable for managing and interpreting the vast amounts of data generated, enabling the reconstruction of protein-protein interaction networks and the identification of post-translational modifications. Innovations such as tandem mass tags (TMT) and data-independent acquisition (DIA) have further enhanced the throughput and depth of proteomic analyses, making it feasible to study proteomes in unprecedented detail. However, these advancements also introduce challenges in extracting valuable information from the large volumes of available data[4].

Recently, Artificial Intelligence (AI) and Machine Learning (ML) have emerged as transformative fields that empower computers to perform tasks typically requiring human intelligence by learning from data and improving over time. AI encompasses a broad range of technologies, including natural language processing, computer vision, and robotics, aimed at replicating cognitive functions such as reasoning, problem-solving, and decision-making.

These technologies have revolutionized various industries by enhancing automation, optimizing processes, and enabling the analysis of vast datasets to uncover insights previously unattainable[5]. Their success in other fields has inspired their application in bioinformatics, where large datasets are prevalent and require processing at unprecedented scales.

This dissertation will focus on applying AI/ML methodologies to address some of the challenges in proteomics, which are listed below:

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

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 around 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.

1.2 Systematic Evaluation and Integration of Multi-Modal Gene Embeddings

Genes are fundamental for the specification of physical and biological traits; understanding their molecular events is often supported by a holistic examination of various aspects

including structure, interacting partners, and function. However, existing methods for representing genes as embeddings are often built towards specific tasks, where a particular aspect is focused, while the holistic view is not completely captured. In this study, we examined knowledge representations of genes from various aspects in a task-agnostic manner. Nine gene embeddings from three major categories: experimental data, literature data, and knowledge graph data are utilized and evaluated. Applying Singular Vector Canonical Correlation Analysis (SVCCA), we discovered that each representation possesses unique information with minimal overlap with other categories. Such sparsity in gene representation encourages embeddings that capture multi-faceted representation. To achieve a comprehensive view of biomolecules, we constructed an AI-based workflow using an Autoencoder to integrate these diverse knowledge representations into a unified multimodal embedding. To validate our results, we performed multiple analyses: our approach consistently ranked at the top in five diverse benchmark experiments, outperforming other embedding methods tested. As expected, the other methods only excelled in focused subsets of these benchmarks. Furthermore, our platform demonstrated the ability to impute missing embeddings by leveraging available datasets from non-missing embeddings, highlighting its utility in refining individual embeddings with incomplete information. Taken together, we have developed a method to characterize large set of biomolecules in a holistic fashion, presenting an efficient framework for integrating multimodal biomolecule embeddings, generating a comprehensive set of multimodal biomolecule embeddings optimized for machine learning and AI applications in biomedical research.

1.3 Missing Values in Longitudinal Proteome Dynamics Studies: Making a Case for Data Multiple Imputation

Temporal proteomics data sets are often confounded by the challenges of missing values. These missing data points, in a time-series context, can lead to fluctuations in measurements or the omission of critical events, thus hindering the ability to fully comprehend the underlying biomedical processes. We introduce a Data Multiple Imputation (DMI) pipeline

designed to address this challenge in temporal data set turnover rate quantifications, enabling robust downstream analysis to gain novel discoveries. To demonstrate its utility and generalizability, we applied this pipeline to two use cases: a murine cardiac temporal proteomics data set and a human plasma temporal proteomics data set, both aimed at examining protein turnover rates. This DMI pipeline significantly enhanced the detection of protein turnover rate in both data sets, and furthermore, the imputed data sets captured new representation of proteins, leading to an augmented view of biological pathways, protein complex dynamics, as well as biomarker–disease associations. Importantly, DMI exhibited superior performance in benchmark data sets compared to single imputation methods (DSI). In summary, we have demonstrated that this DMI pipeline is effective at overcoming challenges introduced by missing values in temporal proteome dynamics studies.

1.4 Summary

This dissertation explores the integration of artificial intelligence and machine learning techniques to advance proteomics. By leveraging state-of-the-art mass spectrometry and bioinformatics tools, the research addresses computational challenges of prediction of protein post-translational modifications (PTMs) with the novel MIND-S model. Additionally, the work systematically evaluates and integrates multi-modal gene embeddings derived from experimental data, literature, and knowledge graphs, providing a comprehensive, task-agnostic representation of gene function that outperforms traditional methods. Finally, the dissertation introduces a Data Multiple Imputation (DMI) pipeline to robustly tackle missing values in longitudinal proteome dynamics studies, significantly enhancing the analysis of protein turnover rates and biomarker–disease associations. Together, these contributions demonstrate the transformative potential of AI/ML methodologies in unraveling complex biological processes and improving biomedical research.

CHAPTER 2

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

2.1 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[6, 7]. 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[8], 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[7, 9, 10]. A common and widely used prediction schema is to predict PTMs based on local amino acids spanning the target sites. 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 mech-

anism 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[11], many PTM patterns as well as their underlying mechanisms largely remain a mystery. Indeed, many phosphosites are orphans without information on their associated kinases[12].

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. 4.1

2.2 Results

2.2.1 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 2.2.

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[13]. To increase model robustness, a bootstrap method was implemented, where multiple models are trained on sampled datasets and ensembled together at the end stage 4.1A. A fixed testing set (about 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 4.1B. Extended core sequences were input to our model for multi-label training, and all PTMs falling within core sequences will be trained simultaneously 4.1C. The trained model was evaluated on the validation set by the area under the precision-recall curve (AUPR)[14]. AUPR was chosen over the area under the receiver operating curve (AUC)[15] as AUPR yields a more informative evaluation when the data are imbalanced[16], 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) 2.6. 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 4.6. The embedding is then passed to a bidirectional long-short term memory (LSTM)[17] 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[18] and graph attention layer[19], 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 2.6. We also constructed a dataset consisting of 13 types of oxidative PTM (O-PTM) from a mass spectrometry project as an independent dataset 2.8. 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 the model is proportional to the amount of training data 4.2A. 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 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 4.2A.

Another feature, multi-label[20] 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 4.2B. 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)[21], improves model performance such that the representation of positional information is learnable and can be better utilized to improve predictions. Indeed, 4.7A 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[22]. As a result, 4.2C 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. 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[23]. 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 4.2D. 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 (4.2E). 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 2.10.

2.2.2 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[24], 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[25]. We first investigated its application on the -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[26]. 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" 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 4.3A. 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[27, 28]. 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 4.024; 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 4.3A. 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 4.8A. The clustering results are shown in 4.3B colored by clusters. We used the cluster center as a representation of the clusters (4.3C and 4.8) 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, 4.3C 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 4.3D. 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.

2.2.3 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[29, 30]. 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[31]. 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 (2.12A). In total, 51 SNPs that change the PTM prediction from positive to negative with a stringent criterion (wild-type prediction score greater than 0.8 and mutation prediction score smaller than 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[32]. 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[33]. The other SNP, P251S, on potassium voltage-gated channel subfamily H member 2 (Uniprot: Q12809), is a mutation found in long-QT syndrome[34]. 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 se-

quences 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[35]. 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[30] and therefore provides validation on the method. On the other hand, the effect of promoting a PTM is found in SNP R1628P on phosphosite 1627 with the prediction score changed greatly from 1.45e3 to 0.912 (4.4A). We visualized the PTM map of the wild-type and mutant LRRK2 in 4.4B. 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.

2.2.4 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[36, 37]. 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[38]. 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.

2.3 Methods

2.3.1 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[23] (09/2021), where 13 PTM types were included. For training/validation/testing split, to prevent information leakage from similar proteins, we applied Uniref. 50[13], 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 2.6.

The second dataset is an Oxidative PTM (O-PTM)-centric dataset we have collected in-house combined with a publicly available dataset.[39] 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 to guide the splitting. A summary of the O-PTM data in each split is shown in 2.8.

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.

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/)[40] 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[41] 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.

2.3.2 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.[42] 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, and a is the output of the layer.

Bidirectional LSTM layer

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

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

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

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

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

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

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

where x_t is the input of the t-th amino acid input embedding, h_{t-1} is the $t-1$-th hidden state, 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:

$$Out_t = Concat\left(h_t^f, h_t^b\right)$$

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

Multi-head self-attention

$$Q_i = W_{Q_i} X$$

$$K_i = W_{K_i} X$$

$$V_i = W_{V_i} X$$

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

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

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 represents 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.

2.3.3 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[43]. 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 a subsequence of Seq from position a to position $b-1$, s is the length of the sequence, c is the size of the 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

$$CoreSeq_i = ECSeq_i[l : r]$$

$$l = \begin{cases} 0, & i = 0 \\ \frac{c}{4}, & i \neq 0 \end{cases}$$

$$r = \begin{cases} s_i, & i = \frac{(s-1)}{c*2} - 1 \\ \frac{3c}{4}, & i \neq \frac{(s-1)}{c*2} - 1 \end{cases}$$

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 the 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[44].

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[45], 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:

$$emb_0 = 0$$

where the emb_0 is a zero vector.

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

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

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

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 the trapezoidal rule and scaled with respect to input to get the final saliency vector s:

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

The sum of all entries of s will be used as the 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[46] 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[47]. The sequence frequency plot of kinase was generated by aggregating the substrate sequences of that kinases retrieved from PhosphoSitePlus[31]. 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 PTM-Var[31] 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.

2.4 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. 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. 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. 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. 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 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, 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. Various neural networks have been proposed to process sequence data in the field of natural language processing (NLP). 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. 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 examined the PTM motif via interpretation module, however, the current analysis only focus on capturing known kinase motif. Another direction will be exploring the proportion of known motifs within the clustering patterns could serve as a validation of our model interpretability. However, conceptually, a direct comparison is somewhat challenging because kinase motifs are traditionally represented as position weight matrices (20-dimensional, corresponding to the 20 amino acids), whereas our model outputs a single scalar saliency score per position of the entire sequence. One potential solution is to use the saliency scores as weights for the corresponding amino acids, technically speaking, we could consider constructing a position weight matrix analogous to the established kinase motifs. This strategy may enable a comparison, and may present opportunities leading to discoveries of novel motifs. We are excited about this approach. We plan to explore this approach in our future investigations

We performed an ablation study to assess the contributions of the structure modules. The results indicated that the structure module contributes less to the overall model performance than the sequence module. We hypothesize several potential reasons for this observation: Prediction Limitations from AlphaFoldDB: The structures provided by AlphaFoldDB may have limited information and not necessarily perfectly reflect the true 3D conformations. PTMs might be sensitive to local structural variations which host them, thus are not adequately captured by a global reconstruction of that protein as shown in AlphaFoldDB. Redundancy Between Sequence and Structure: There may have been overlaps between the protein information contained in their sequence and structure. It is possible that when the model extracts the relevant features from the sequence, it automatically suppresses the additional value from the structural data related to the same protein. Incomplete Leverage of Structural Information: Although protein structures obtained from AlphaFoldDB contain rich information, their binary contact maps may capture only a subset of the 3D features. Recent studies have demonstrated alternative approaches for modeling protein 3D

structures[48–50], which may provide a more comprehensive representation;

Currently, MIND-S predicts whether an amino acid residue is a potential PTM target without conditioning on contextual variables such as cell type or disease state. The reason for MIND-S to focus on PTM prediction without conditions is to first reduce the potential search space for PTM sites and allow the model to learn a fundamental understanding of PTMs not complicated by other variables. However, PTMs have been shown to demonstrate differences across cell types or diseases in human samples. This heterogeneity occurs because the specificities of PTM depend on various intrinsic factors of the modification process within a given cell or tissue type.

NSCLC tumors express different combinations of active tyrosine kinases, resulting in distinct phosphorylation patterns that may lead to varied drug resistance profiles. This underscores the need for individualized therapies based on specific phosphorylation signatures [51]; Similarly, tumor cells often exhibit altered glycosylation on asparagine (N-linked) or serine/threonine (O-linked) residues with structures and expression levels that differ from those in normal cells—a hallmark of cancer that has been exploited as biomarkers (e.g., aberrantly glycosylated MUC1 in breast cancer) [52].

Therefore, a condition-specific PTM prediction model is essential for accurately mapping the PTM landscape in biological systems. In future iterations, we envision extending the model to incorporate additional context (e.g., cell type, disease conditions) to support more nuanced and condition-specific predictions.

2.5 Code and Data Availability

Training datasets and pig cardiac PTMome results have been deposited in Zenodo and are publicly available:

https://doi.org/10.5281/zenodo.7655709

https://doi.org/10.5281/zenodo.7655827

https://doi.org/10.5281/zenodo.7655835

All original code has been deposited on GitHub as well as Zenodo and is publicly available https://github.com/yuyanislearning/MIND

2.6 Acknowledgments

This chapter is based on my joint work with other collaborators, especially Dr. Jyun-yu Jiang and Dr. Wei Wang and my Ph.D. advisor Dr. Peipei Ping.

2.7 Figures

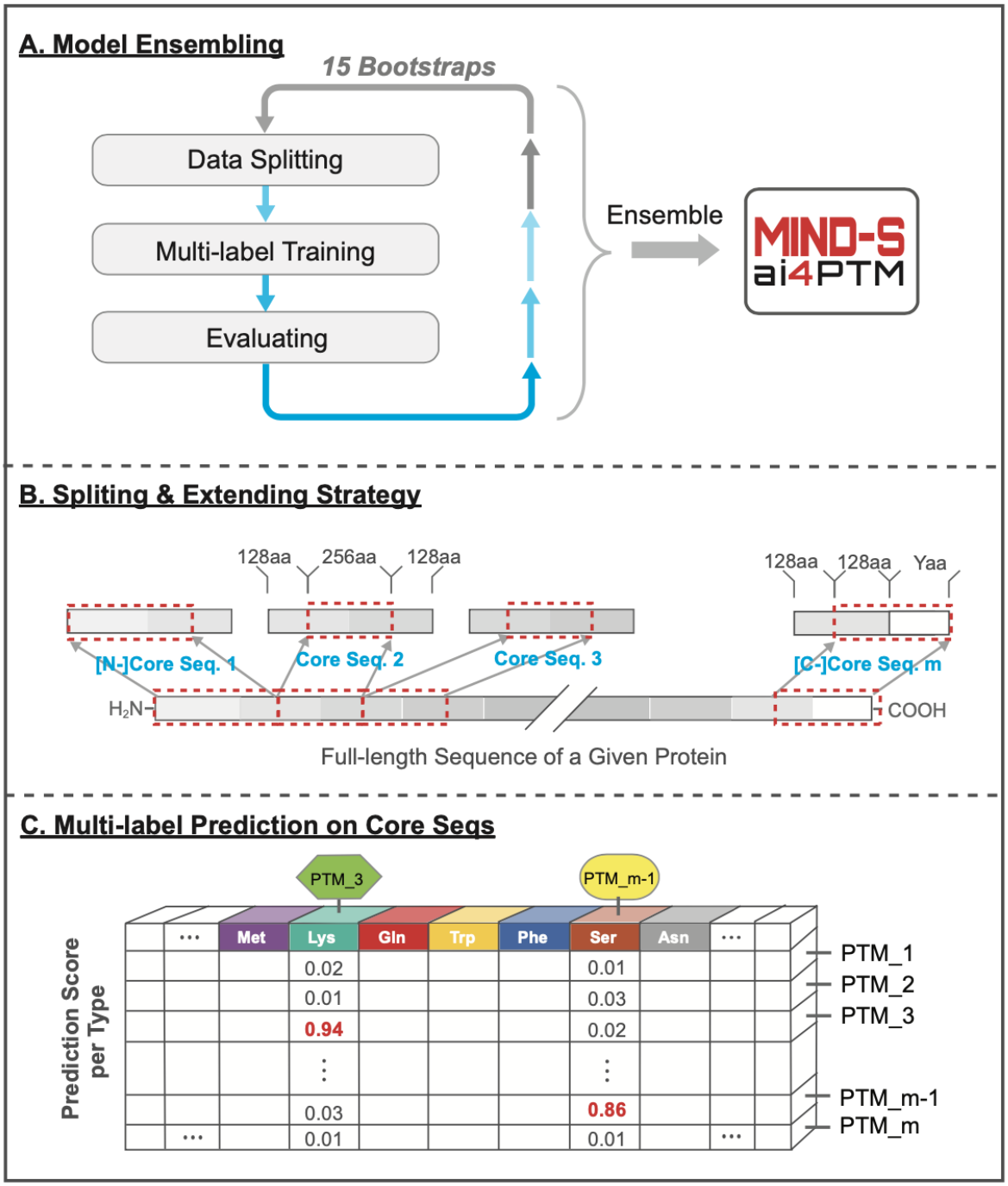


Figure 2.1: Design of MIND-S

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.

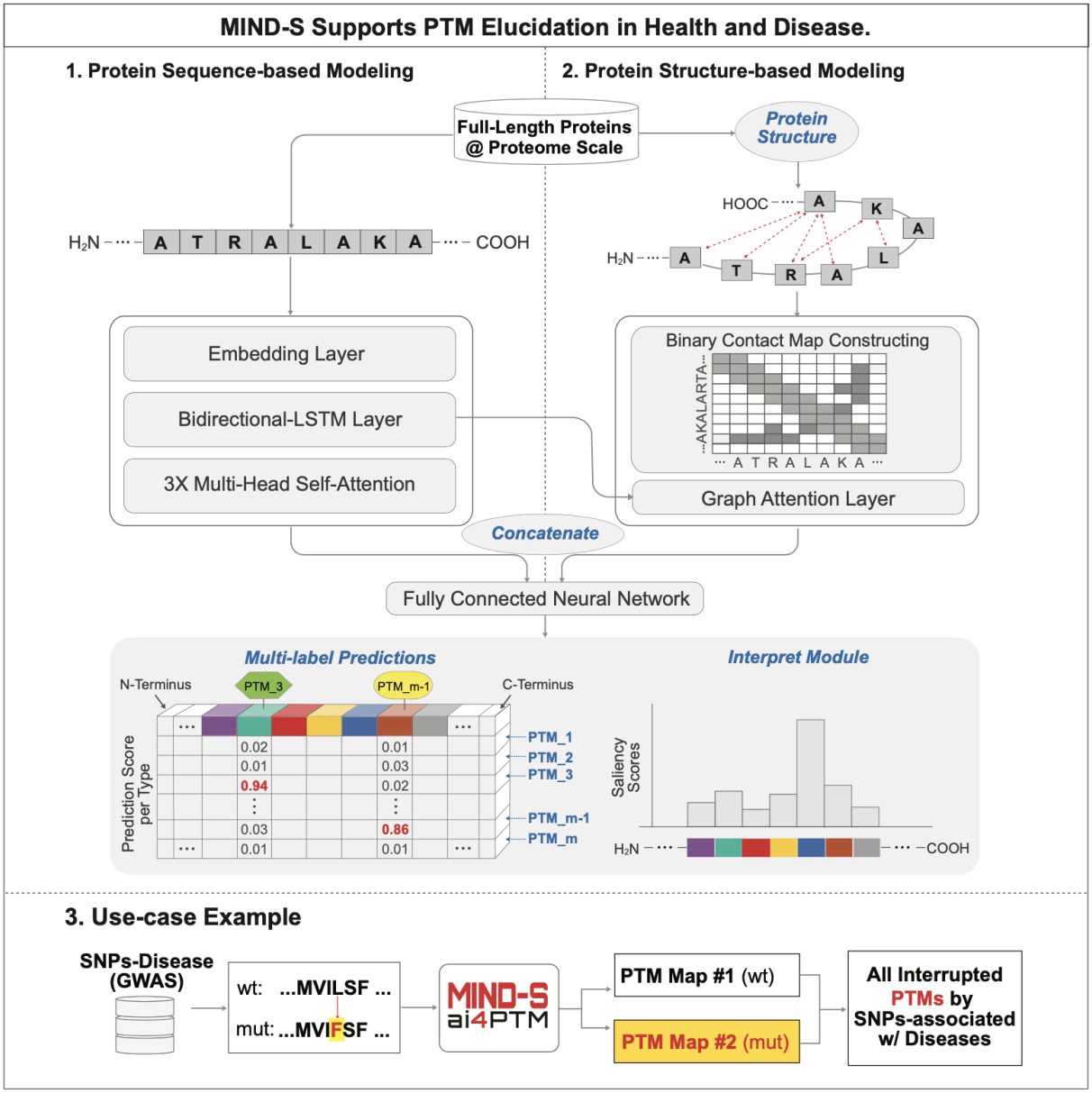
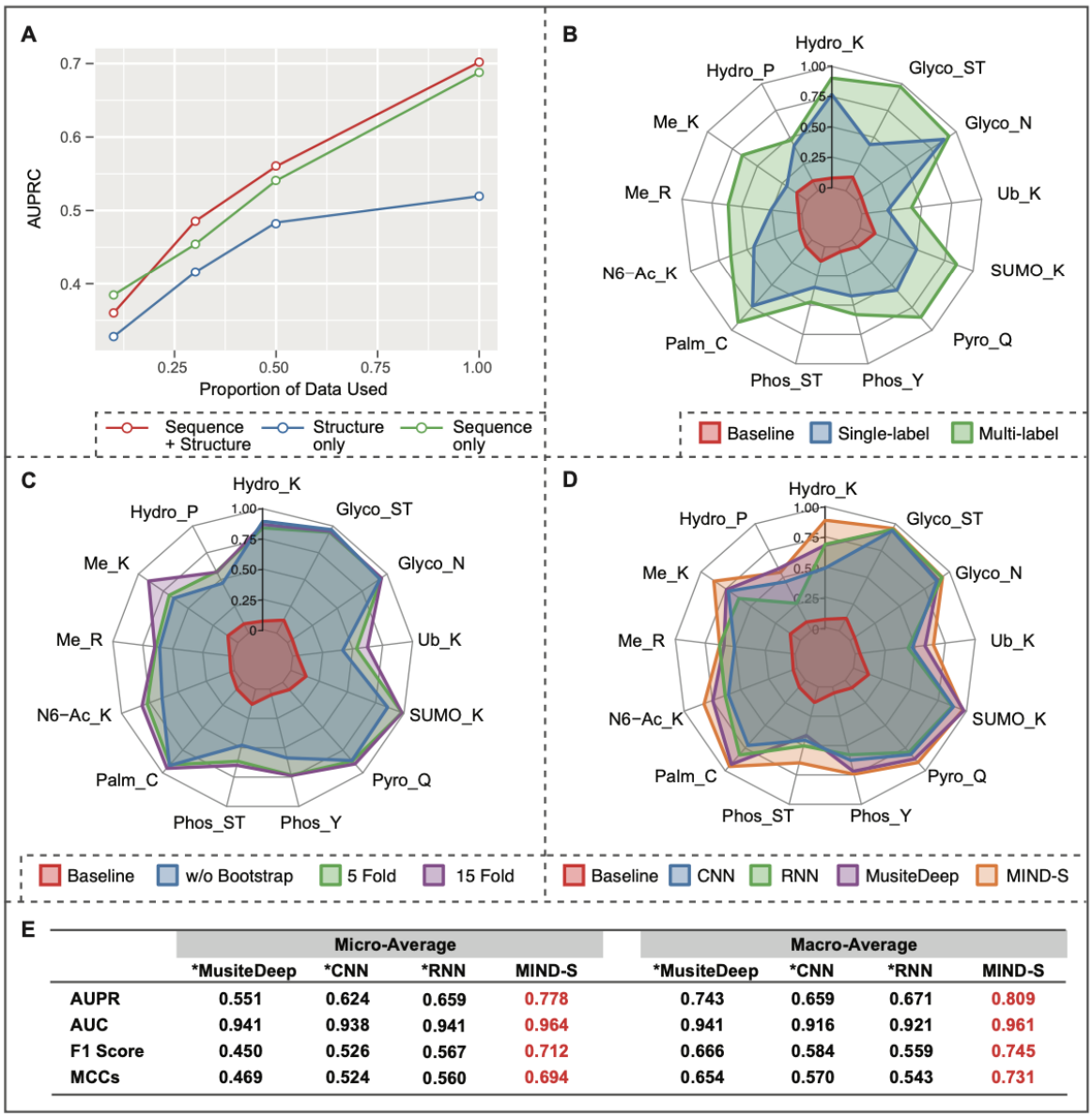


Figure 2.2: Graphical Abstract

MIND-S architecture overview



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 2.3: 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.**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; asteriod $p < 0.001$.

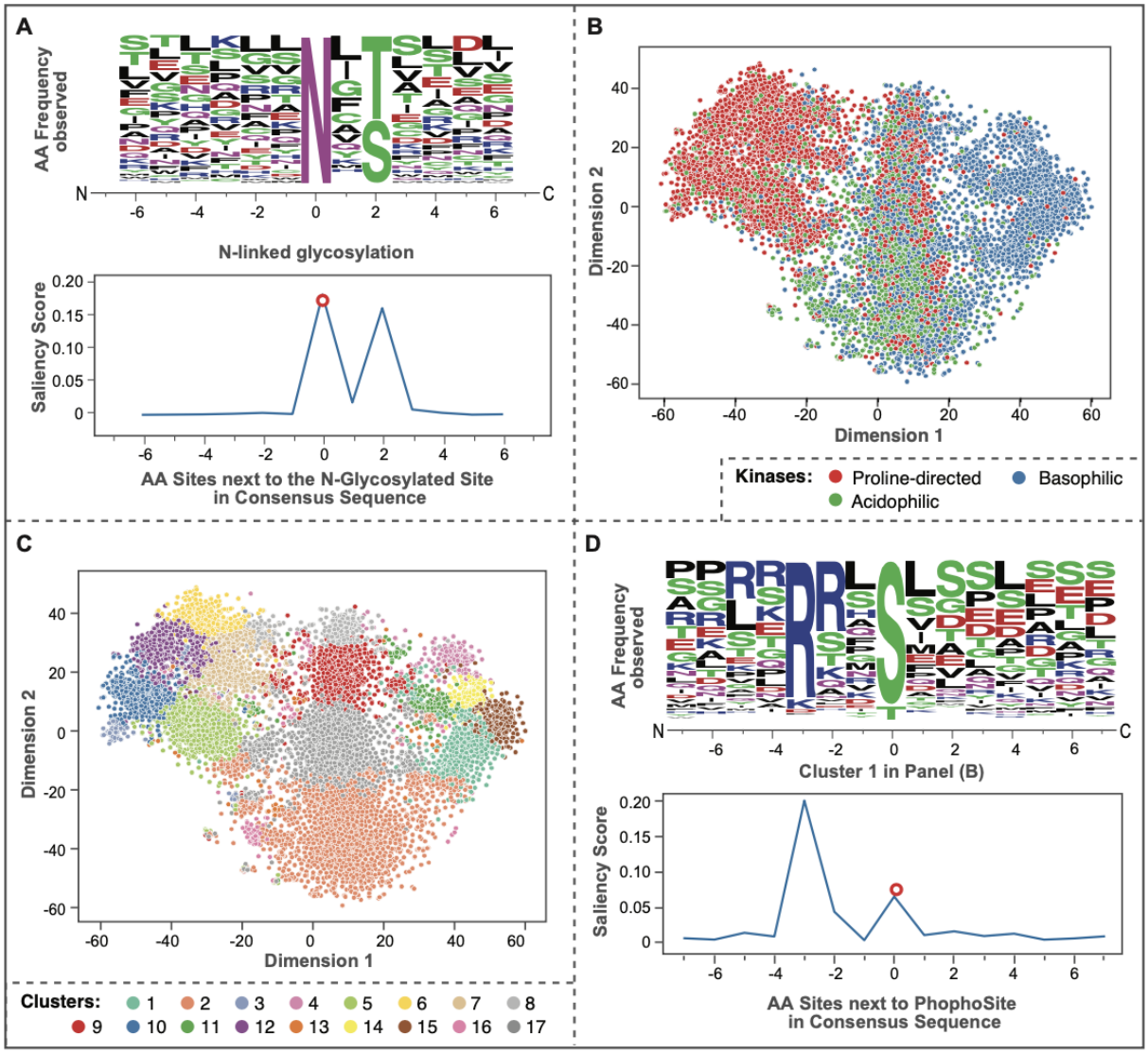


Figure 2.4: Validation of the interpretation module of MIND-S

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.

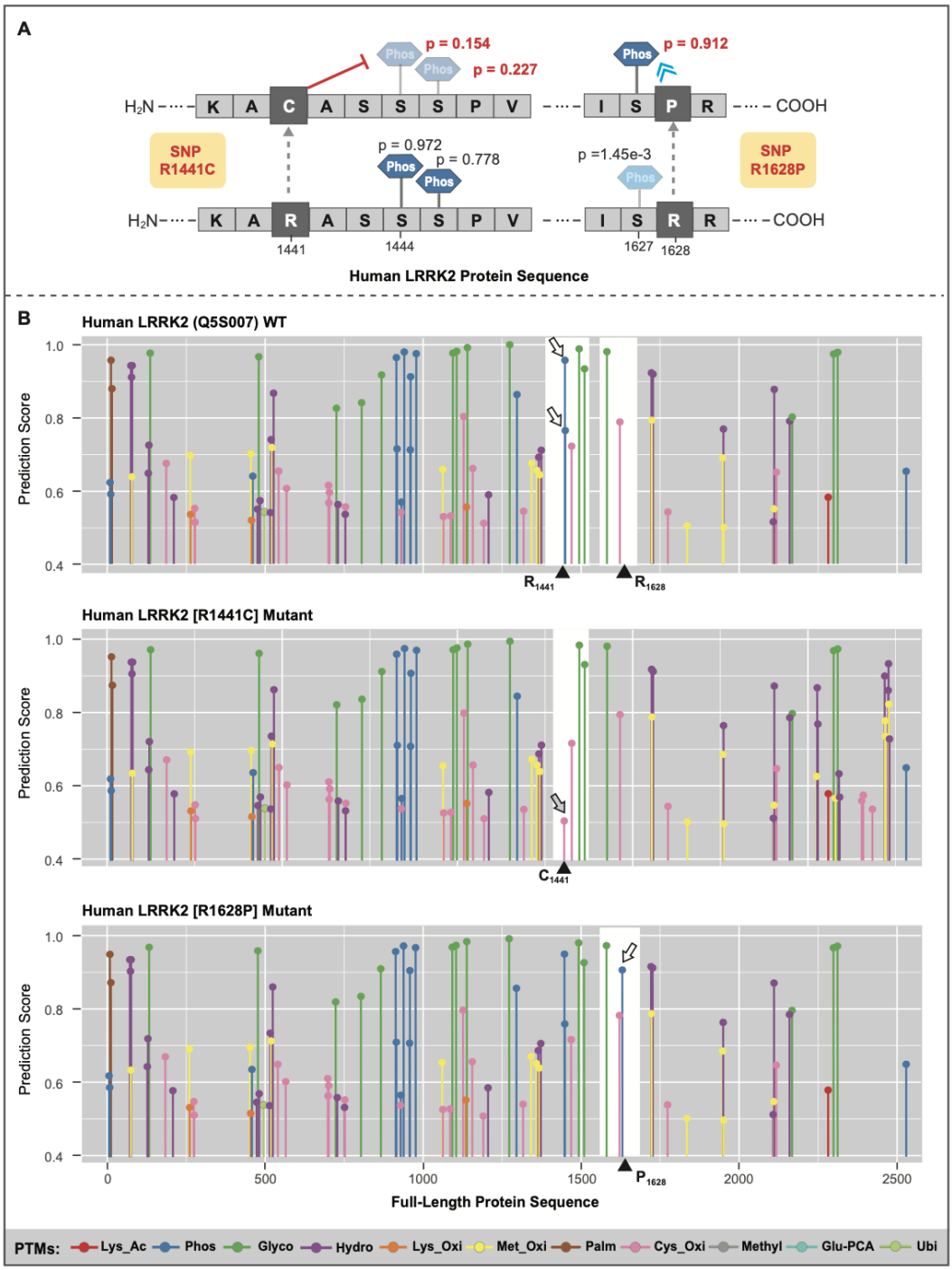


Figure 2.5: MIND-S examines the effect of SNP on LRRK2 PTM

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.

2.8 Supplementary materials

2.8.1 Supplementary figures

PTM types (amino acid)	Training (P/N)	Validation (P/N)	Testing (P/N)	Total (P/N)
Hydroxylysine (K)	431/3,074	60/368	20/187	511/3,629
Hydroxyproline (P)	3,812/14,522	903/2,334	87/666	4,802/17,522
Methylation (K)	1,547/30,207	102/2,216	30/1,053	1,679/33,476
Methylation (R)	4,334/94,851	241/7,170	140/3,894	4,715/105,915
N6-acetyllysine (K)	17,833/222,984	1,282/13,819	1,002/11,499	20,167/248,302
S-Palmitoylation (C)	2,518/13,606	168/802	115/668	2,801/15,076
Phosphorylation (S)	93,734/1,330,423	5,942/77,783	5,067/74,736	104,743/1,482,942
Phosphorylation (T)	18,043/820,514	1,281/49,501	1,004/46,149	20,328/916,164
Phosphorylation (Y)	7,682/72,942	514/4,597	412/4,363	8,608/81,902
Pyrrolidone-carboxylic-acid (Q)	1,086/8,385	44/581	43/240	1,173/9,206
SUMOylation (K)	953/18,839	32/765	28/641	984/20,245
Ubiquitin (K)	2,851/30,699	106/901	152/2,969	3,109/34,569
N-linked glycosylation (N)	59,143/353,879	2,876/17,759	3,741/22,966	65,760/394,604
O-linked glycosylation (S)	467/15,309	19/659	18/533	504/16,501
O-linked glycosylation (T)	187/10,131	3/591	16/433	206/11,155
Total number of PTMs	214,621/3,040,365	13,573/179,846	11,875/170,997	240,090/3,391,208
Number of proteins	43,907	2,393	2,511	48,811

Figure 2.6: PTM Dataset Summary

To best evaluate the performance of our tool, we have adopted the 13 PTM dataset from MusiteDeep (https://www.musite.net/). These datasets were used to validate outcomes predicted by MIND. They are split into training, validation, and testing sets. All amino acids detected either with a PTM (positive, P) or without PTMs (negative, N) are shown. The total number of proteins analyzed in each set is shown.

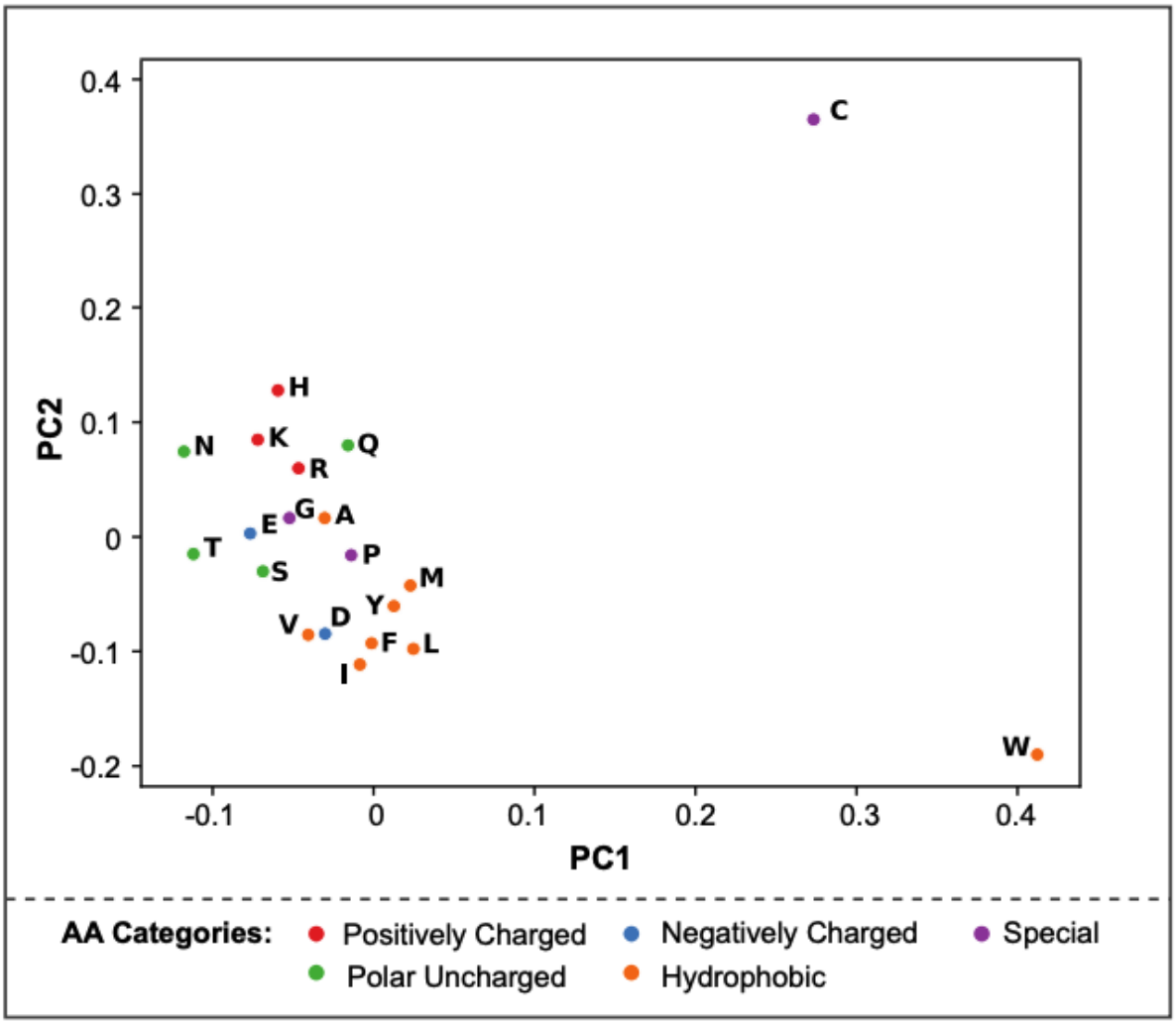


Figure 2.7: PCA plot of amino acid embedding from MIND

Embeddings of 20 amino acids are extracted from MIND and PCA is applied for visualization. Each dot represents one amino acid and is colored by category. Amino acids colored in red are the amino acids with a positive charge [Arginine (R), Histidine (H), Lysine (K)]; amino acids colored in blue are the amino acids with a negative charge [Aspartic acid (D), Glutamic acid (E)]; amino acids colored in green is the uncharged polar amino acids [Serine (S), Threonine (T), Asparagine (N), Glutamine (Q)]; amino acids colored in orange are the amino acids that are hydrophobic [Alanine (A), Valine (V), Isoleucine (I), Leucine (L), Methionine (M), Phenylalanine(F), Tyrosine (Y), Tryptophan (W)]; amino acids in colored in purple are the remaining amino acids [Cysteine (C), Glycine (G) and Proline (P)].

PTM types (amino acid)	Training (P/N)	Validation (P/N)	Testing (P/N)	Total (P/N)
Arginine hydroxylation (R)	904/22,393	75/1,737	79/1,997	1,058/26,127
Asparagine hydroxylation (N)	2,016/23,324	234/2,849	170/1,732	2,240/27,905
Aspartate hydroxylation (D)	2,639/36,217	297/4,243	250/3,882	3,186/44,342
Cysteine 4-HNE (C)	213/6,106	16/164	23/500	252/6,770
Cysteine sulfination (C)	486/8,280	59/848	57/913	602/10,041
Cysteine sulfonation (C)	702/9,643	72/671	64/927	838/11,241
Lysine carbonylation (K)	455/16,239	57/1,318	31/675	543/18,232
Methionine sulfonation (M)	493/3,711	56/508	48/232	597/4,451
Methionine sulfoxide (M)	7,396/28,814	891/3,624	732/3,267	9,019/35,705
Phenylalanine hydroxylation (F)	1,368/17,201	158/2,128	128/1,800	1,654/21,129
Tryptophan hydroxylation (W)	779/2,719	87/360	82/337	948/3,416
Tyrosine hydroxylation (Y)	1,391/11,580	151/1,324	139/1,064	1,681/13,968
Valine hydroxylation (V)	2,999/48,744	333/5,044	278/4,407	3,610/58,195
Total number of PTMs	21,841/234,971	2,486/24,818	2,081/21,773	26,228/281,522
Number of proteins	3,163	397	397	3,957

Figure 2.8: O-PTM Dataset Summary

A total of 13 O-PTM types of mouse cardiac proteome were collected in-house using established methods. They were split into training, validation, and testing sets. All amino acids detected either with a PTM (positive, P) or without PTMs (negative, N) are shown. The total number of proteins analyzed in each set is shown.

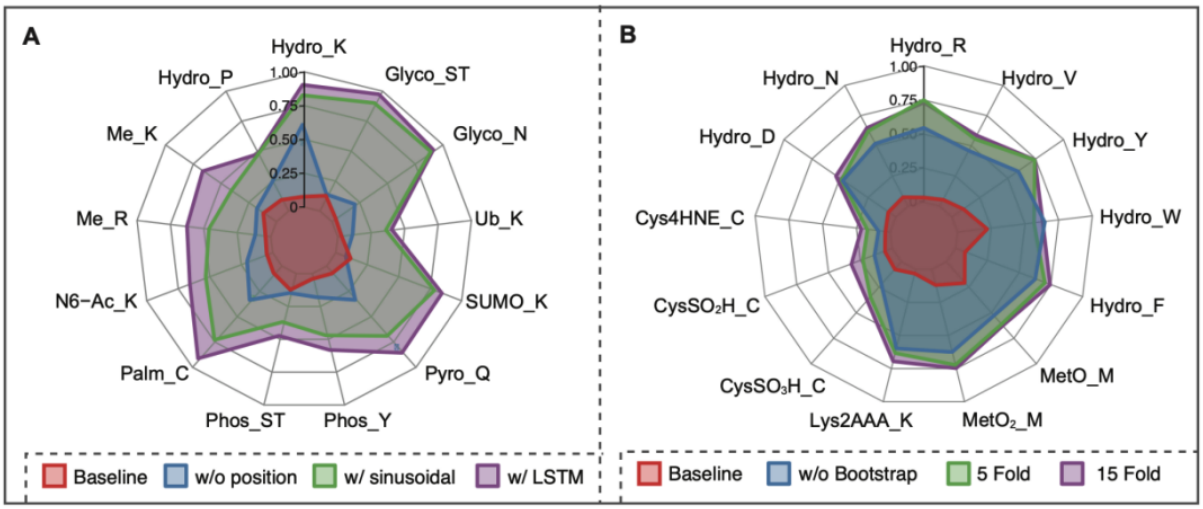


Figure 2.9: MIND-S Performance on PTM Prediction

Panel **A** and **B**, 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. Panel **A**, the AUPR for PTM prediction utilizing different position encoding methods. The AUPR of the model without any positional information provided is shown in blue; The AUPR of the model using sinusoidal positional encoding is shown in green; The AUPR of the model using LSTM as a way to encode positional information is shown in purple. Panel **B**, the AUPR for Oxidative PTMs (OPTMs) 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.

Model	MIND-S	# attn=4	#attn=2	h_d=32	h_d=64	#head=4	#head=16
Hydroxylysine (K)	0.868	0.08	0.834	0.492	0.678	0.729	0.614
Hydroxyproline (P)	0.534	0.147	0.488	0.409	0.438	0.524	0.406
Methylation (K)	0.739	0.027	0.612	0.518	0.63	0.669	0.758
Methylation (R)	0.465	0.067	0.441	0.318	0.476	0.512	0.415
N6-acetyllysine (K)	0.597	0.089	0.619	0.482	0.589	0.632	0.636
S-Palmitoylation (C)	0.849	0.238	0.816	0.824	0.864	0.838	0.851
Phosphorylation (ST)	0.483	0.058	0.458	0.419	0.452	0.489	0.492
Phosphorylation (Y)	0.564	0.1	0.595	0.507	0.592	0.596	0.591
Pyrrolidone-carboxylic-acid (Q)	0.772	0.219	0.854	0.805	0.817	0.84	0.932
SUMOylation (K)	0.904	0.051	0.75	0.801	0.859	0.888	0.856
Ubiquitin (K)	0.471	0.075	0.574	0.257	0.342	0.489	0.416
N-linked glycosylation (N)	0.925	0.158	0.918	0.915	0.917	0.921	0.921
O-linked glycosylation (ST)	0.954	0.019	0.903	0.878	0.941	0.928	0.956
Macro-average AUPR	**0.701**	0.102	0.681	0.586	0.661	0.696	0.680

Figure 2.10: Hyperparameters tuning Summary

A table summarizes the PTM-specific performance of different hyperparameters choice. The base model, MIND-S, is the final model used in the paper except that it is not ensembled. Only one hyperparameters were changed on the base model to test the hyperparameter's impact on performance. Specifically, "attn=4" changes the number of multi-head self-attention layers to 4 (3 in the base model), similar, "attn=2" changes it to 2; "h_d=32" changes the hidden dimension to 32 (128 in the base model), "h_d=64" changes it to 64; "head=4" changes the attention head to 4 (8 in the base model), "head=16" changes it to 16. From the average AUPR, we chose the hyperparameters setting in base model as our final model setting since it achieves the best performance.

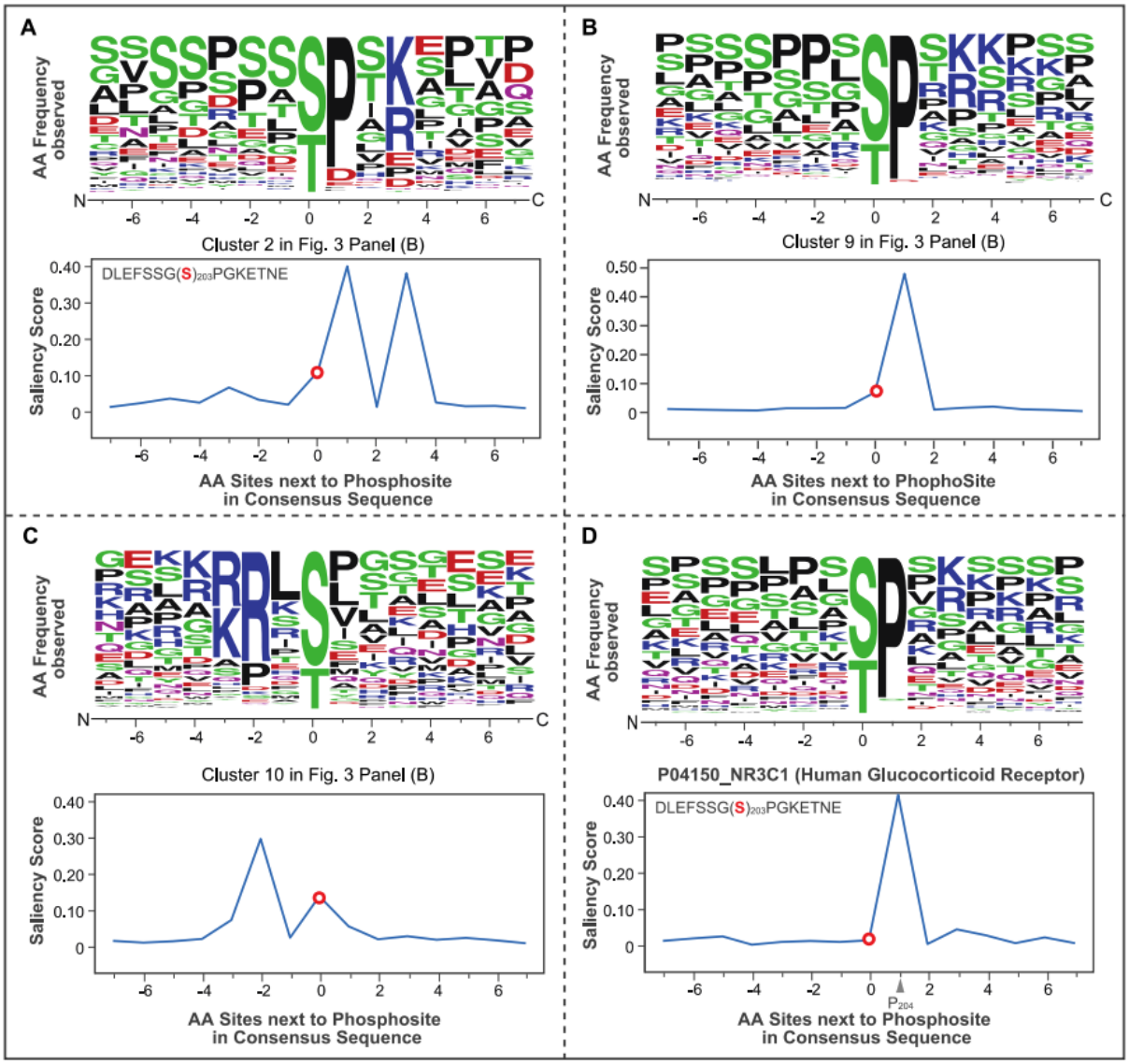


Figure 2.11: Saliency clustering results

Panel **A**, the upper panel is the sequence frequency plot of all sequences from cluster 2, where the +1 and +3 position shows enrichment of proline (P) and arginine (R) respectively. The bottom panel is the saliency scores of the representative of cluster 2, where the +1 and +3 positions have peak saliency scores. Panel **B**, the upper panel is the sequence frequency plot of all sequences from cluster 9, where the +1 position shows enrichment of proline (P). The bottom panel is the saliency scores of the representative of cluster 9, where the +1 position has a peak saliency score. Panel **C**, the upper panel is the sequence frequency plot of all sequences from cluster 10, where the -3 position shows enrichment of arginine (R). The bottom panel is the saliency scores of the representative cluster 10, where the -3 position has a peak saliency score. Panel **D**, the upper panel is the sequence frequency plot of all sequences from substrates of kinase CDK-1, where the +1 position shows enrichment of proline. The bottom panel is the flanking saliency scores of protein P04150 site 203, where the +1 position has a peak saliency score. The two panels show a matching pattern on position +1.

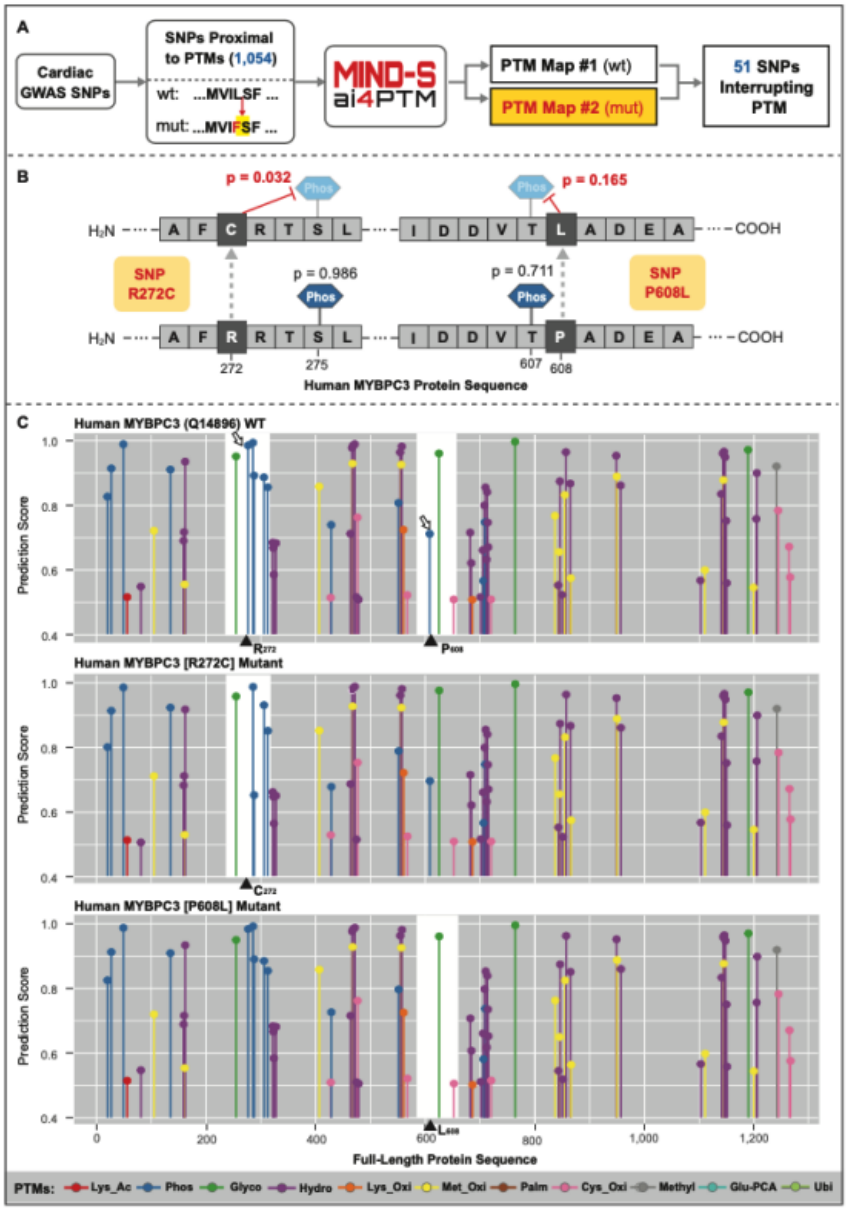


Figure 2.12: Illustration of SNP affecting PTMs

Panel **A**, MIND-S Workflow for determining the impact of disease-specific SNPs on PTMs (e.g., cardiac disease-associated SNPs that affect PTMs shown). Among the cardiac diseaseassociated SNPs identified in GWAS, after translation, 1,054 SNPs are proximal to PTMs. Accordingly, we examined their impacts on PTMs. In silico mutations on protein sequences were guided by these SNPs. With the input of mutant and wild-type protein sequences, MIND-S outputs scores of PTMs and determines whether this SNP affects any PTMs. In total, 51 SNPs were found to interrupt the PTMs. Panel **B**, An illustration of SNPs interrupting or promoting PTM occurrences on myosin binding protein C (MYBPC3). SNP R272C on protein MYBPC3 is found to have reduced phosphorylation scores on site 275 from 0.986 to 0.032. SNP P608L is found to have an elevated score of phosphorylation on site 607 from 0.711 to 0.165. Panel **C**, PTM maps of wild-type and two mutant MYBPC3. 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 R272C mutant, one phosphorylation site is interrupted. In the P608L mutant, one phosphorylation site is interrupted.

CHAPTER 3

Systematic Evaluation and Integration of Multi-Modal Gene Embeddings

3.1 Introduction

Artificial Intelligence (AI) has exhibited remarkable capabilities in various fields, notably in natural language processing and computer vision. Its growing prominence is also evident in numerous biomedical fields, including genomics, genetics, proteomics, drug discovery and precision medicine[1, 53–58]. A critical factor underlying this advancement is the effective representation of biomedical concepts in a machine-interpretable format, facilitating the integration of diverse AI and machine learning (ML) methodologies. Constructing embedding for genes as well as their products including transcripts and proteins (refer to gene embedding for simplicity) are one of the essential components of it. Gene embeddings are typically compact numerical vectors, encapsulating the intrinsic properties of genes and elucidating their interrelationships depending on the data and model utilized. Gene embeddings can serve as building blocks within biomedical AI/ML as gene/transcript/protein are the fundamental concepts in the biological domain. For example, gene embeddings have been utilized in predicting gene characteristics and interactions[59–61] and adapted to general tasks such as single-cell RNA-seq analysis[62, 63].

Gene embeddings demonstrate considerable versatility across a range of biomedical artificial intelligence and machine learning (AI/ML) tasks; however, their efficacy is largely influenced by the alignment between the gene embeddings and the specific downstream applications. To maximize the utility of gene embeddings for particular research inquiries, it is essential to evaluate whether the underlying data and method used for embedding generation

encompasses the relevant information, a process typically tailored to specific ML tasks. However, the burgeoning diversity of biomedical data modalities and applications necessitates gene embeddings that offer a comprehensive representation of genes. Assessments based narrowly on specific task performances may lose the generalizability and prove insufficient for gauging the multi-faceted representational capacities of gene embeddings. Additionally, developing a fair and biologically reasonable evaluation task is inherently challenging, let alone devising a series of tasks that cover multiple aspects.

In this context, we propose an alternative evaluation framework, that involves assessing gene embeddings in a task-agnostic manner, eliminating the need to predefine a ML task. Essentially, gene embeddings that represent different dimensions of genes are compared to each other to identify the overlap shared. This is grounded in the premise that a holistic gene embedding will capture diverse aspects of genes and therefore exhibit substantial overlap with other gene embeddings. While methods to compare representations from neural networks have been mainly focused on learning the dynamics of neural networks, we adapt the method to compare gene representations generated from distinct sources. We developed a workflow utilizing Single-Value Canonical Correlation Analysis (SVCCA) to quantify the similarity between pairs of gene embeddings. We assessed the underlying similarities between embeddings derived from disparate data sources or methods, providing insights into the multi-faceted representation ability of gene embeddings.

To ensure diverse perspectives of genes are included, we investigated nine different embeddings, which we broadly categorize them into three main categories based on their respective data sources, namely: Knowledge Graph Embeddings, Molecular Data Embeddings, and Natural Language Embeddings. Knowledge Graph Embeddings are derived from knowledge graphs comprising entities, such as genes and other biomedical components, along with their interrelationships. These embeddings effectively capture both the structural and semantic dimensions of the corresponding biomedical knowledge stored in the Knowledge Graph. Molecular Data Embeddings, on the other hand, are constructed from diverse molecular datasets, such as omics data and sequence data. These embeddings encapsulate the intrinsic biological insights present within the molecular data. Lastly, Natural Language Embeddings

are recently developed with the advances in natural language processing. They are usually generated from textual sources, distilling the semantic meanings of genes as reflected in the scientific literature. These embeddings emphasize various aspects of biomedical data and knowledge, representing a diverse array of gene-related information.

Our investigation revealed that the majority of gene embeddings we assessed did not correlate well with gene embeddings from different modalities. This is expected as these embeddings are usually generated from single data sources. To address this issue, we developed a Multi-Modal Gene Embedding (MMGE) by integrating gene embeddings across various data sources and methods using a customized autoencoder model. Our results demonstrate that MMGE exhibits integrated information that highly overlaps with existing gene embeddings. Furthermore, MMGE consistently ranks among the top-performing embeddings across a diverse array of ML evaluation tasks, underscoring its capacity to capture a comprehensive spectrum of gene information. Additionally, we demonstrate that leveraging diverse information, MMGE can be used to impute missing gene embeddings for incomplete gene embeddings. These findings suggest that MMGE could serve as an effective and widely adoptable data in biomedical AI/ML applications.

3.2 Results

3.2.1 Canonical Correlation Analysis Between Gene Embeddings

To evaluate the multi-faceted representation capabilities of gene embeddings, we examined the shared information between these sets that are generated from diverse data sources. We employed SVCCA to perform the comparison between gene embeddings. At its core, SVCCA identifies linear transformations that maximize the correlations between the two sets of embeddings. Although SVCCA calculated correlations of the embeddings in a transformed space, this correlation is not to be confused with the convention correlations, but should seen as an indicator of information overlap in the representations from the two embedding sets. We structured the two gene embedding sets with genes aligned and applied SVCCA to compute the correlation score (ranging from 0 to 1, reflecting the degree of similarity between the two gene embedding sets) (see 3.1). Additionally, we devised a shuffling method to compute a background correlation to adjust the correlation value for fair comparison.

In this study, we selected a range of representative gene embedding sets from three distinct categories for detailed investigation. From the molecular data embedding category, our selection includes gene2vec[59], omics[64], geneformer[65], and prottrans[66]. Gene2vec employs a concept analogous to word2vec[67] on gene expression data from the GEO database to construct gene embeddings. Omics is a composite embedding, integrating expression data, genome-wide essentiality screening data, and protein sequence encodings. Geneformer represents a transformer-based model, specifically trained on extensive single-cell RNA-seq data. Prottrans is a protein language model developed through self-supervised masked language modeling (MLM) on protein sequences. In the knowledge graph embedding category, we chose struc2vec[68] on the protein-protein interaction (PPI) database which demonstrated better performance[60]. We included two gene embeddings from a heterogeneous biological knowledge graph, know2Bio, created using two knowledge graph embedding techniques (cite TransE and MurE). For the natural language embeddings, we selected genePT[69] and BioLinkBERT[70]. In these embeddings, the representation of a gene is extracted as the embedding of the gene's description, utilizing GPT embedding mode and BioLinkBERT as the

underlying language models. From each of these gene embedding methods, we extracted all available gene embeddings for our analysis. The broad and diverse range of gene embeddings enables comprehensive evaluations on representation similarity between gene embeddings.

Pairwise SVCCA correlations were computed within selected gene embeddings (3.2). Most gene embeddings exhibited a relatively similar degree of similarity of representation with other embeddings except for struc2vec on PPI. The distinct position of struc2vec could suggest a lower overlap of information in its representation. Notably, genePT and BiolinkBERT exhibited the highest similarity among all comparisons, indicating a substantial overlap of information between these two pairs. This finding aligns with the fact that both genePT and BiolinkBERT gene embeddings were generated from the same data source and format (text descriptions of genes). A similar pattern was observed in the two Know2BIO embeddings. Although these embeddings generally showed lower similarity with other gene embedding sets, they exhibited the highest similarity with each other. These observations validate the effectiveness of our approach in assessing the shared information between gene embeddings. The remaining gene embeddings showed less similarity, suggesting distinct representations. In addition, it is shown that computing a background distribution is necessary, given that background correlations varied considerably across different comparison pairs. These results show that there is information shared between different gene embeddings, the level of which is relatively low, with each capturing particular aspects of genes.

3.2.2 MMGE (Multi-Modal Gene Embedding) Integrates Multi-Modal Gene Embeddings

As AI/ML approaches become increasingly prevalent in the biomedical field, there is a growing demand for information-rich molecular embeddings. However, as the gene embeddings examined exhibit relatively low information overlap. This challenge motivated new approaches that can present a holistic view of a molecule; we envisioned that this approach should have the following characteristics: (1) It should be able to incorporate information of a molecule from multiple data modality to capture comprehensive characteristics. (2) It

should demonstrate versatile ability to accommodate a variety of downstream applications. (3) And it should also leverage the existing pretrained models and embeddings. (4) Lastly, it should be easily scaled with the increasing number of representations.

To address this challenge, we developed the Multi-Modal Gene Embedding (MMGE), designed to encapsulate and integrate multi-modal gene information captured by diverse gene embeddings, making it applicable to a broader range of applications. MMGE employs a customized autoencoder model to refine and distill this information into a condensed form (3.3A). For each gene, embeddings from different modalities are combined as the input and compressed by the encoder to form a condensed gene embedding. The condensed gene embedding is converted back to the original embeddings by a decoder. The model is trained to have the gene embedding to be condensed while maintaining the capacity to reconstruct the diverse original gene embeddings. After the model is trained, the output from the encoder will be utilized as the MMGE.

To ascertain whether the MMGE encapsulates comprehensive information, we calculated its SVCCA correlations with other gene embeddings with the same procedure. The results showed that MMGE exhibited significantly higher correlations in comparison to other gene embedding approaches (3.3B), implying that MMGE effectively integrates information from a variety of gene embedding sets.

To further evaluate whether the high correlation of MMGE to other embeddings is driven mainly by the fact that the embeddings to be compared are also inputs of the MMGE, we employed a "leave-one-out" test. Specifically, for each gene embedding utilized, we calculated the correlation with a reduced MMGE whose input excluded that particular gene embedding, such that the reduced MMGE to be compared does not "know" the gene embedding prior to comparison. From the comparison, we assess how much shared information MMGE could still capture without having been exposed to the specific gene embedding during training. The reduced versions of MMGE continued to achieve the highest similarity scores, regardless of the exclusion of particular gene embeddings (3.3C). This finding reinforces the notion that different gene embeddings contain overlapping information, and MMGE is capable of amalgamating these diverse data sources to a certain degree.

3.2.3 MMGE Demonstrates Strong Performance across Various Downstream Tasks

To highlight the comprehensive information captured in MMGE compared to other gene embeddings, we assessed its performance across a range of gene-related ML tasks. Selecting tasks that highlight diverse attributes of gene data was crucial in evaluating MMGE's capacity to encapsulate multifaceted information. Thus, five distinct ML tasks were chosen: Gene Ontology (GO) prediction, protein subcellular location prediction, gene dosage sensitivity prediction, protein-protein interaction (PPI) prediction, and gene-gene interaction prediction (see details in Methods). The GO database[71] is a comprehensive resource designed to systematically describe the functional attributes of gene products, significantly aiding in the characterization of gene functions. Protein Subcellular Localization Prediction is crucial for understanding a protein's function and physicochemical properties, with accurate prediction reducing the need for labor-intensive experimental determination. Gene Dosage Sensitivity Prediction is vital for interpreting the impact of copy number variants in genetic diagnostics, offering insights into gene sensitivity to dosage variations. Protein-Protein Interaction (PPI) Prediction entails forecasting the likelihood of interactions between pairs of proteins. Lastly, Gene-Gene Interaction Prediction focuses on predicting whether pairs of genes exhibit functional overlap.

Our focus was on the generalizability of MMGE across various tasks rather than excelling in a single one. Some tasks, such as GO prediction, might inherently favor certain embeddings like those based on natural language, as the original gene descriptions may already include functional information. Therefore, assessing the overall performance across all tasks provides a more comprehensive evaluation of the gene embeddings' generalizability. Performance rankings for each task revealed that MMGE consistently achieved a top-three position (3.4), whereas other gene embeddings exhibited more variable rankings. This consistent high performance indicates that MMGE effectively integrates multi-modal information, proving beneficial across a wide range of downstream tasks.

3.3 Methods and Data

3.3.1 Gene Embedding Collection

GenePT: GenePT's precomputed GPT-3.5 embeddings ('text-embedding-ada-0020' model) for each gene were sourced directly from genePT's Zenodo repository

(https://zenodo.org/records/10030426)

Additionally, the NCBI gene summaries used by GenePT were also retrieved from this repository.

BiolinkBert: BioLinkBERT-base is a model that has been pretrained on PubMed abstracts, incorporating citation link information. In this study, the NCBI gene summaries utilized by GenePT, were tokenized using the BioLinkBERT tokenizer and adjusted the tokenized input to a length of 512 through padding and truncation. We then processed these inputs through the pretrained model

(https://huggingface.co/michiyasunaga/BioLinkBERT-base)

The embedding for each gene was represented by the '[CLS]' token embedding obtained from the model output.

String struc2vec gene embedding were retrieved from BioNEV

(https://github.com/xiangyue9607/BioNEV).

Omics gene embedding were retrieved from the supplementary data from.

Prottrans: Prottrans embeddings were retrieved from the UniProt protein database. These embeddings are generated using the ProtT5 protein language model. We obtain the per-protein embeddings (UP000005640_9606, Per-protein embeddings for Homo sapiens reference proteome). In this process, a fixed-length embeddings vector is computed for each protein sequence. UniProt IDs were converted to gene names via UniProt ID mapping API. In cases where a protein was associated with multiple genes, the first gene was used.

Geneformer embeddings were retrieved from the gene embedding layer from "geneformer-12L-30M" model

(https://huggingface.co/ctheodoris/Geneformer).

While in its original publication, gene embeddings were retrieved from averaging output gene embeddings from each cell. As we are interested in the universality of gene embedding instead of genes in particular cell types or cellular context, gene embedding layers which are invariant to the cellular context were used to generate the embeddings. Specifically, we retrieved gene IDs geneformer used, tokenized them and extracted the corresponding gene embedding from the model.

Gene2vec embedding was directly obtained from

(https://github.com/jingcheng-du/Gene2vec).

Know2BIO Knowledge Graph Embeddings were retrieved for all genes found in the Know2BIO biomedical knowledge graph. Know2BIO integrates biomedical data from 30 biomedical knowledge bases describing relationships between genes, proteins, drugs, diseases, biomedical pathways, and other data types. Data was accessed on August 18, 2023 and consists of 219k nodes and 6.18M edges. Knowledge graph embeddings were trained on this knowledge graph, using an 80:10:10 split for training, validation, and test set. Two knowledge graph embedding models were chosen based on their superior performance in the benchmark evaluation: MurE[72] and TransE[73]. Both models were trained with negative sampling of 150, learning rate of 0.001, using Adam optimizer with 1000 max epochs.

All gene IDs from various embeddings were converted to the HGNC symbol using mygene.info (https://mygene.info/) or Ensembl REST API (https://rest.ensembl.org/) or Uniprot (https://www.uniprot.org/).

3.3.2 Correlation analysis

SVCCA calculation

SVCCA, a variant of CCA, is tailored to focus on singular values that capture the majority of variance in gene embeddings. This is based on the premise that minor fluctuations in gene embeddings may not yield meaningful insights. A key advantage of SVCCA is its computational efficiency, which allows for repeated measurements on large gene embedding

datasets.

A particular gene embedding dataset is structured as a matrix, where rows represent different genes and columns represent the embedding dimensions. Before conducting correlation analyses, the embeddings are standardized by column (embedding features). To compare two gene embedding sets, we retain the intersection of their gene sets, resulting in two gene embedding matrices with matching rows (i.e., each row represents the same gene in both embeddings) but different columns. Specifically, let EMB1 have dimensions $X(m, d_1)$ and EMB2 have dimensions $Y(m, d_2)$. For each gene i in the gene set, X_i and Y_i represent the embeddings in the two gene embedding sets.

SVCCA was implemented as in its original publication[74]. SVCCA first performs singular value decomposition on matrices X and Y. This results in singular vectors X' and Y' with associated singular values λ_{1-d_1} for X and Y, respectively. From these d_1 singular vectors, we retain the top d_1', where d_1' is the smallest value that satisfies $\sum_{i=1}^{d_1'} |\lambda_i| \geqslant 0.99 \sum_{i=1}^{d_1} |\lambda_i|$, thereby retaining vectors that account for 99% of the variance in X. CCA is then applied to these top singular vectors, X' and Y', producing transformed subspaces. The average of the correlations between each transformed subspaces is used as the final correlation value.

Background correlation calculation

To assess baseline correlations between different gene embedding pairs, we calculated a background correlation distribution. We randomly permuted gene sets in the two embedding methods to ensure unmatched gene pairs, calculating the SVCCA correlation between these shuffled embeddings to represent background correlations. This permutation process was repeated 100 times to derive an empirical background correlation distribution. The p-value was defined as the proportion of permutations where the correlation exceeded the actual correlation: $p = \frac{\text{number of permutations with correlation} > \text{actual correlation}}{\text{total number of permutations}}$. We used a p-value threshold of 0.05, indicating statistical significance if fewer than 5 permutations showed higher correlation than the actual correlation. We further utilized this background distribution to calculate an adjusted correlation by subtracting the mean background correlation from the unpermuted correlation.

Correlation plots

The correlation heatmap was generated from the correlation matrix, where the diagonal was set to zero to indicate self-comparisons for visualization purposes. Hierarchical clustering was applied to this heatmap to cluster similar gene embeddings. The figure was produced using the Python package Seaborn's Clustermap function. The correlation ranking boxplot was created using the adjusted SVCCA correlations, excluding self-correlations, to illustrate the correlation range with other gene embedding sets.

3.3.3 MMGE generation

The autoencoder model employed comprises two primary components: an encoder and a decoder. The encoder is designed as a three-layer multilayer perceptron (MLP) with the specific function of compressing the diverse information inherent in gene embeddings. Conversely, the decoder is a two-layer MLP responsible for reconstructing the dense representation produced by the encoder back to its original input form.

Implementation of the Autoencoder Model: The autoencoder model is implemented using PyTorch version 1.13.1. The encoder component consists of two linear layers with dimensions of 1024 and 512, respectively. Leaky ReLU activation functions with a negative slope of 0.01 are applied following these linear layers. The decoder, in contrast, comprises a single linear layer matching the input dimensionality, and no activation function is utilized in this layer. The output generated by the encoder serves as the Multimodal Gene Embedding (MMGE).

The input comprises nine concatenated gene embedding sets: OMICS, Gene2vec, BioLinkBert, GenePT, ProtTrans, K2B-MurE, K2B-TransE, Geneformer, and STRING-struc2vec. A total of 12,528 genes, common to all nine embedding sets, were employed for training. The gene embeddings are standardized on the feature side. For reconstruction, mean squared error (MSE) loss is utilized, and the optimization is conducted using the Adam optimizer. The model undergoes training for 100 epochs, with early stopping criteria triggered if there is no decrease in loss over the course of three consecutive epochs (patience = 3).

Because the nine gene embedding sets vary in size, direct calculation of the MSE loss

may bias results towards embeddings with larger dimensions. To address this, the loss was weighted based on the size of each input embedding during optimization: the weight for each embedding is calculated as the total size divided by the size of dimension.

To assess the similarity of the MMGE on unseen gene embedding sets, we trained a series of reduced models, each omitting one of the nine gene embedding sets in input during the autoencoder training process. Consequently, nine reduced MMGE models were generated, each corresponding to a scenario with one missing gene embedding set. SVCCA similarity scores were computed between these nine reduced MMGEs and their respective missing gene embeddings, which were then compared to the similarity scores obtained between the missing gene embeddings and the remaining gene embeddings.

All training processes were executed utilizing an NVIDIA A100 GPU.

3.3.4 Gene embedding benchmark

We selected in total five major tasks for benchmark: GO prediction, PPI prediction, gene dosage prediction, protein location prediction, Gene gene interaction prediction.

GO prediction: GO21 is an ontology of gene, with detailed annotation of gene's function, involving biological processes (BP), molecular functions as well as cellular location. We chose to predict biological processes of genes. We only retained the GO terms with experimental evidence by filtering on evidence code: 'EXP', 'IDA', 'IPI', 'IMP', 'IGI', 'IEP','TAS', 'IC'. As we are interested in the general performance of gene embeddings in universal tasks, we selected the high level biological process and chose 6 of them that have more than 100 positive cases in the gene sets: Biological Regulation, Metabolic Process, Localization, Cellular Process, Response to Stimulus, Developmental Process. Specifically, we retrieved the BP associated with the gene and traced back to these 6 root BPs using the hierarchy of GO terms. We trained the model for each BP separately as a binary classification task.

PPI prediction. PPI were retrieved from STRING24, with a focus of experimental validated interaction in humans ("9606.protein.physical.links.v12.0.txt"). For each PPI pair, we retained it when the two proteins both exist in the intersected gene set (genes that appear

in all gene embeddings). We chose to multiply the two gene embeddings to get the input for the model to avoid the directionality of which gene embedding to be put in the front for concatenating gene embeddings.

Gene dosage prediction. Gene dosage is a task adopted from geneformer, where gene dosage indicates whether the copy number of the gene is important for survival.

Protein location prediction, protein location is a task adopted from deeploc[75] to predict the cellular location of proteins. For each cellular location, we trained a separate model to predict it as a binary classification task.

Gene gene interaction prediction is a task defined in gene2vec8, which is to predict whether two genes have shared function annotation. We used a similar approach as in PPI, to multiply the two gene embeddings to serve as the input for the prediction.

3.3.5 Pycaret ML

In this study, we employed a systematic approach to train and evaluate for the different tasks with different gene embeddings, using the PyCaret library, to evaluate the gene embedding benchmark tasks. We selected Logistic Regression for all tasks. These methods were chosen for their diversity in modeling techniques, which provided a comprehensive evaluation of their applicability to gene embedding tasks. The metric used for model comparison was the area under the precision-recall curve (AUPRC), a robust measure for evaluating model performance. This metric was selected due to its effectiveness in reflecting the precision and recall balance of the models, thus providing a more nuanced understanding of their predictive capabilities, especially in class-imbalanced predictions. To ensure a fair and rigorous evaluation of each model, StratifiedKFold validation approach with 10 folds was used. This method of cross-validation maintains the proportion of samples for each class, which is particularly important in minimizing training bias due to class imbalance. By employing this approach, we could assess the generalizability and stability of each model across different subsets of the data. Models with longer training times were excluded from our analysis to optimize computational efficiency without compromising the integrity of the results. The

best-performing model for each gene embedding task was identified based on the highest AUPRC score.

3.4 Discussion

The growing understanding of omics and the rapid advancement of computational techniques underscore the potential for gene embeddings to significantly influence applications ranging from precision medicine to drug discovery. The evolution of artificial intelligence (AI) and machine learning (ML) has enabled the transformation of gene representations into embeddings that are more accessible for computational applications. These developments have even empowered biologists with minimal modeling expertise to implement machine learning models, thereby democratizing the use of advanced computational methods.

Despite these advancements, the evaluation of gene embeddings remains predominantly single-task oriented, limiting their broader applicability. Such trend has been observed in our experiment utilizing SVCCA, a method to calculate canonical correlation in the transformed spaces where correlation is maximized. We discovered that most gene embeddings share little information overlap. This single-focus evaluative approach necessitates an additional process to align the appropriate gene embedding for a given task. Ideally, a robust, universal gene embedding should serve as a pre-computed component that encapsulates diverse information while minimizing resource requirements. Such versatile embeddings could function as foundational elements for more complex downstream tasks or models. Our objective is to systematically explore these possibilities by examining gene embeddings in a task-agnostic manner. We posit that a single task cannot capture the complexity inherent in biological data, while evaluating multiple tasks can be both resource- and time-intensive.

The efficacy of gene embeddings in capturing biological meaning is contingent upon both the dataset used and the method employed to generate the embedding. Ideally, embeddings should represent essential information while minimizing noise; however, achieving or measuring this balance is often challenging. Thus, our study incorporates various data sources and diverse methods applied to the same datasets.

While several benchmarking studies have assessed gene embeddings, fewer efforts have focused on directly comparing the embeddings themselves. Our investigation aims to discern the extent to which information is shared among these embeddings. Specifically, we seek to determine whether different gene embeddings encapsulate distinct information and which embedding would be optimal for developing a new AI/ML model. Nine different gene embeddings are evaluated and little overlap is discovered in this study, indicating distinct aspects of genes captured in various embeddings. Thus, to build an embedding that captures more comprehensive information of genes, we develop the MMGE model to integrate these embeddings into condensed embedding, which removes the redundancy while combining the different aspects effectively. The integrated MMGE has shown higher information overlap with different gene embedding sets as well as demonstrating better performance in a range of downstream gene relevant benchmark tasks. By addressing these questions, we hope to enhance the utility and understanding of gene embeddings within the scientific community.

3.5 Code and data availability

The code is available in github: https://github.com/erikazhengyilin/uge_capybara

3.6 Acknowledgments

I would like to thank my advisor Dr. Peipei Ping for this work.

3.7 Figures

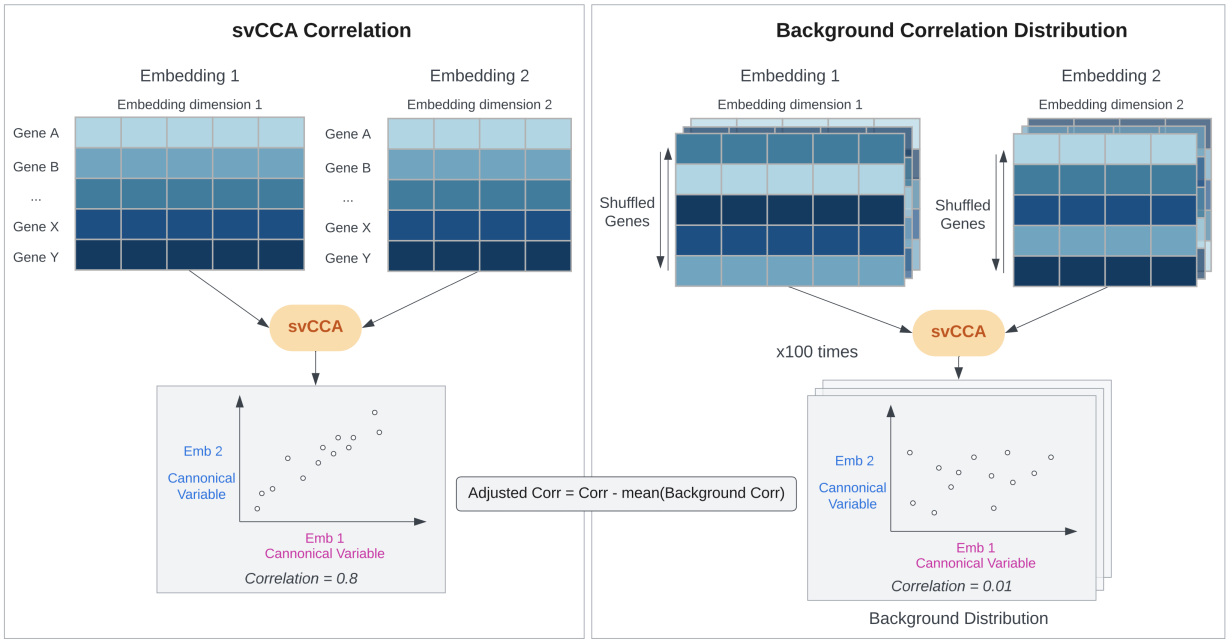


Figure 3.1: SVCCA correlation calculation and background distribution calculation.

The left panel shows the calculation of SVCCA between two gene embedding sets. The two matrices are formatted as gene by embedding dimension (the embedding dimensions can be distinct for each gene embedding set) and the genes are aligned (same color). SVCCA is then applied to obtain the canonical space for each gene embedding. One canonical variable is shown in a scatter plot where x and y-axis represent transformed values from the two gene embeddings and each scatter represents one gene. The right panel shows the process of calculating background distribution SVCCA correlation. The order of genes in each matrix are shuffled (different colors). The shuffle process is repeated for 100 times and SVCCA is calculated for each shuffle, resulting in an empirical background correlation distribution. The adjusted correlation is obtained from subtracting the average background correlation from the original correlation from the unshuffled matrices.

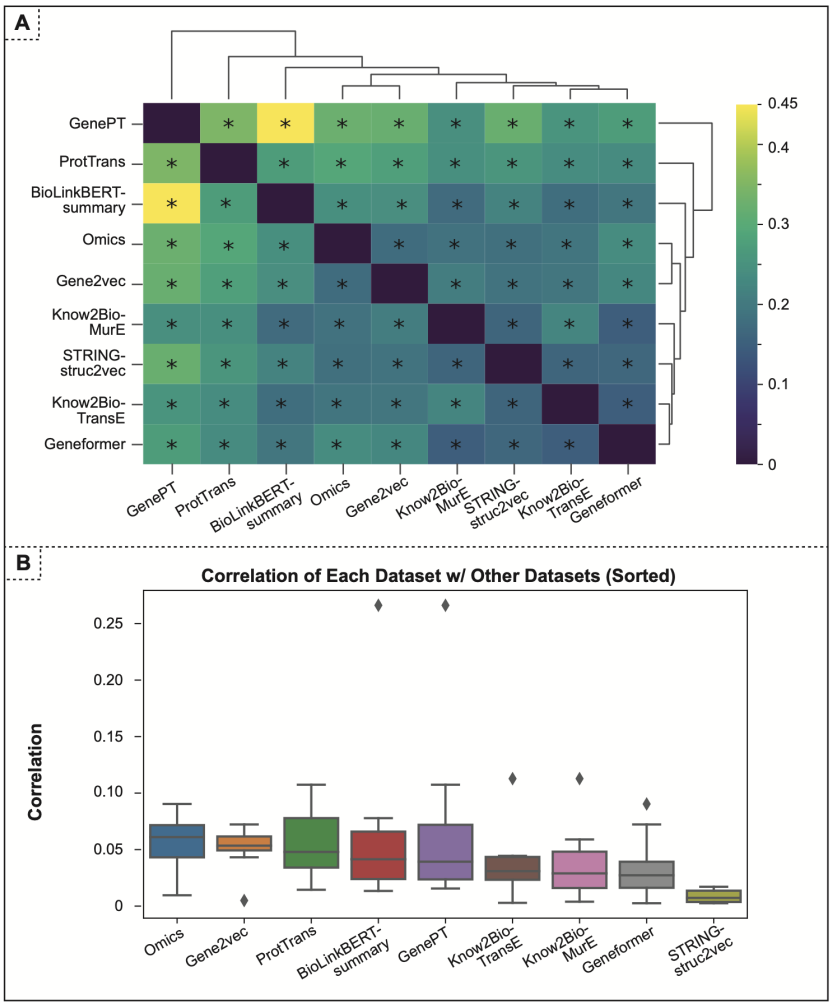


Figure 3.2: Correlations of gene embeddings

A) SVCCA correlation heatmap of gene embeddings. Each entry represents the original SVCCA correlation between the two corresponding gene embeddings. Hierarchical clustering is performed to arrange the row and column of the heatmap. Self comparison of gene embeddings (the diagonal) was not performed and set as 0. Statistical test (see details in Methods Section) was performed for each comparison and the "*" indicates a p-value less than 0.05. B) Boxplot of adjusted SVCCA correlation against other gene embeddings.

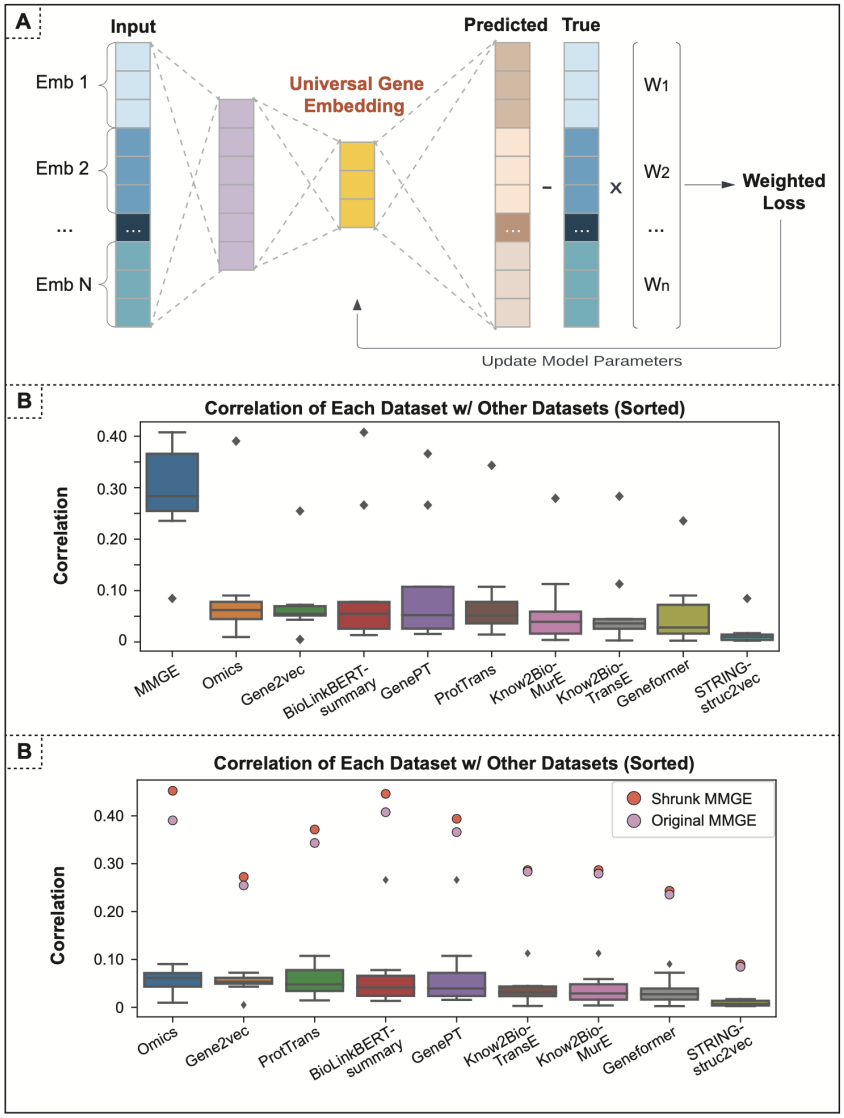


Figure 3.3: MMGE shows high correlations with other embeddings

A) Building MMGE from diverse gene embeddings. Nine gene embeddings of the same genes are concatenated to large vectors and served as the input of the model. The encoder of the model with the reducing layer size serves to condense the input and the decoder of the model is responsible to predict the input from the condensed vector. The difference between the predictions and input will be weighted and used to guide the model optimization. After training is finished, the consed layer (yellow) will be retrieved as MMGE. B) A boxplot of adjusted SVCCA correlation against other gene embeddings. MMGE (blue) stands out as it possesses the highest overall SVCCA correlations. C) A boxplot of adjusted SVCCA highlighting the shrunk MMGEs. The SVCCA correlation of the corresponding shrinked MMGE is shown as a purple circle for each embedding; correlation with the intact MMGE is shown as a red circle.

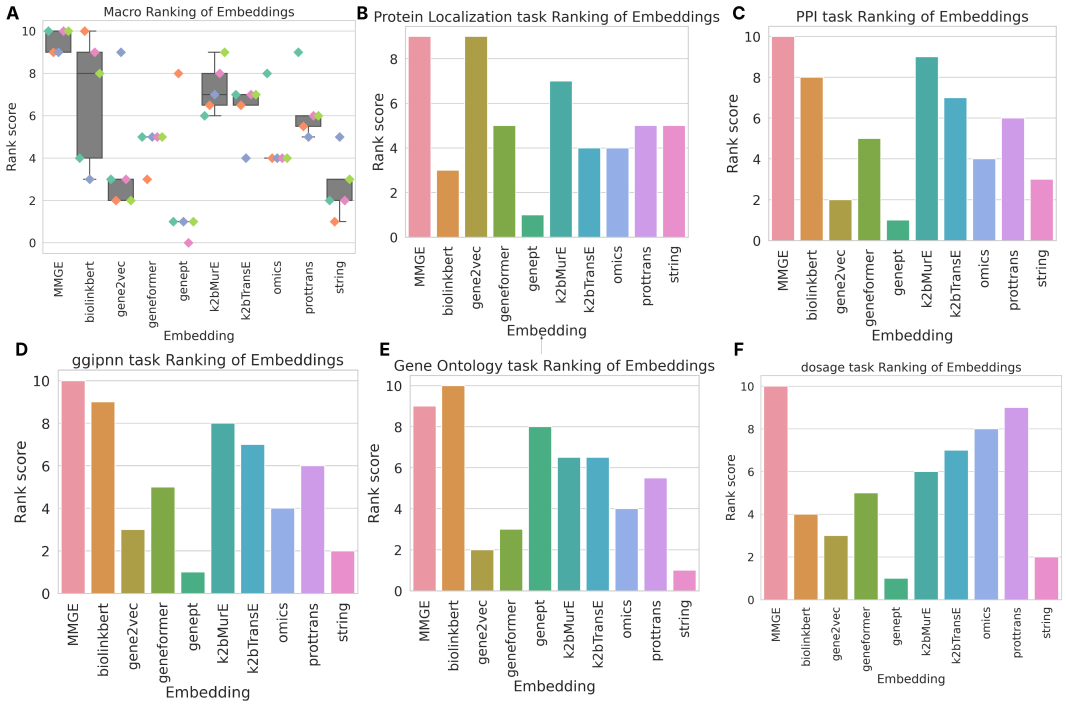


Figure 3.4: Ranking of Embeddings Across Five Downstream Tasks.

(A) A boxplot shows the overall ranking of each embedding across all tasks. The x-axis represents the different embeddings, and the y-axis shows their rank scores (with higher scores indicating better performance). **(B)** In the protein localization prediction task, MMGE and gene2vec tied for the best performance. **(C)** In protein-protein interaction (PPI) prediction, MMGE ranked first, with k2bMurE coming in second. **(D)** For gene-gene regulatory interaction (ggipnn) prediction, MMGE again led the rankings, while Biolinkbert ranked second. **(E)** In gene ontology prediction, Biolinkbert outperformed all other embeddings, achieving the top rank, with MMGE following closely in second place. **(F)** In the gene dosage sensitivity task, MMGE achieved the best performance, with prottrans ranking second. Although each embedding shows unique strengths for specific tasks, overall, MMGE demonstrates superior performance across all tasks tested.

CHAPTER 4

Missing Values in Longitudinal Proteome Dynamics Studies: Making a Case for Data Multiple Imputation

4.1 Introduction

Missing values, absence of observations for one or more variables in the data set, is a common challenge across a wide range of biomedical data sets[76–79], including proteomics data sets[80, 81]. Missing values can adversely impact data quality, subsequent downstream analysis and/or modeling, resulting in biased outcomes, and incomplete conclusions[82]. Overcoming missing data points is essential for rendering a data set to be "AI-ready", which refers to the data operations performed to meet the requirements of AI models[83]. To appropriately address missing values, it is necessary to explore the factors contributing to them, including the conditions under which data sets were collected (e.g., experimental equipment [77, 84, 85]). In particular, missing values in temporal data sets, i.e., data sets with repeated measurements at multiple time points are further complicated by (1) the continuity of time series data, which might be hampered due to the proportion of missing values; and (2) any intrinsic temporal patterns, which are yet to be revealed. Ostensibly, addressing these complexities in temporal data sets requires context specific solutions.

The advancement of proteomics technologies, e.g., tandem mass spectrometry (MS)[58, 86], has rendered proteome-wide examinations and measurements of protein dynamics feasible with unprecedented detail[87, 88]. Despite significant advancements in technology, MS-based proteomics often grapples with the issue of missing values. Missing values in proteomics can arise from a variety of factors, including peptide abundances that fall below the detection limit, error from laboratory preparation or instrumentation and/or data pro-

cessing[89, 90]. When/if a significant portion of peptide data are absent, the subsequent quantification of protein expressions as well as measurements of protein turnover rates will be affected[91]. Accordingly, missing turnover rates and inaccurate turnover rate estimation may occur with incomplete time series when the number of peptides quantified across time points is insufficient for model fitting. This issue introduces biases in subsequent analyses, thus hindering biological discovery and understanding[80, 81, 92]. Seminal works have been implemented to tackle these issues in protein expression data[77, 80, 81, 93–97], whereas effective approaches specifically addressing missing values in the context of temporal dynamics profiling are lacking. Accurate estimation of protein turnover rate is contingent upon a complete time-series data set and is more vulnerable to missing values[98, 99].

Generally, data imputation methods can be classified into single- and multiple imputation approaches. Most imputation methods applied in proteomics are single imputation techniques, where each missing value is filled by one imputed value[77, 80, 81]. Although single-imputation approaches are widely adopted, estimates from single imputation are treated as observed values, making them indistinguishable in downstream analyses. Single imputation falls short of capturing the uncertainty associated with missing values, often resulting in unrealistically narrow standard errors[100]. In contrast, Data Multiple Imputation (DMI) methods address these challenges and have been applied on nontemporal proteomics data set[97]. DMI generates multiple imputations for each missing value, allowing for the aggregation of these imputations to derive a final imputed value. DMI considers variability across imputed data sets, thereby reflecting the inherent uncertainty in missing values, an aspect not addressed by single imputation methods. Moreover, DMI methods can be seamlessly integrated with downstream analysis. For example, for protein turnover rate estimation, imputed values will not be distinguished from observed values, leading to potential overreliance on the imputed data and skewing estimates. DMI imputes multiple values for the same missing values via sampling from posterior distributions of the parameters, better capturing the uncertainty during the process. Then the protein turnover rate can be inferred from each imputed data set individually and then pooled to derive final parameter estimates, therefore better addressing the potential variation from the imputation. In addition, DMI

utilizes time series from other peptides to capture the potential temporal dependency via Fully Conditional Specification (FCS)[101]. Therefore, the DMI integrated workflow takes into consideration temporal dependencies, uncertainties at single time point, as well as time series levels to address the multilevel challenges introduced by missing data in temporal proteomics studies.

We have developed a DMI pipeline to effectively address missing values in estimating protein turnover rates from time series proteomics data. Our workflow 4.1B showcased its effectiveness and generalizability on a cardiac temporal proteomics data set from mice and a temporal plasma proteomics data set from humans.

4.2 Methods and Data

4.2.1 Data Sets

Murine Data Set

A temporal proteomics data set characterizing large-scale cardiac protein turnovers across multiple mouse strains[99]. To summarize, this study is divided into two groups: Isoproterenol (ISO) treated mice and Controlled (Ctrl) mice were metabolically labeled with deuterium water. Within each group, six mouse strains were used: A/J, BALB/cJ, C57BL/6J, CE/J, DBA/2J, and FVB/NJ. From each experimental group, two mice were euthanized on each day: 0, 1, 3, 5, 7, 10, and 14 to collect heart and plasma samples. In the cardiac hypertrophy groups, surgical implanted subcutaneous micro-osmotic pumps (Alzet) were calibrated to deliver 15 $mg{\cdot}kg^{-1}{\cdot}d^{-1}$ of isoproterenol over 14 days.

Human Data Set

A human temporal proteomics data set that performed high-throughput quantification of protein turnover in ten human subjects[102]. This proteomics data was acquired from healthy human plasma samples collected at ten defined intervals: days 0, 1, 2, 4, 5, 8, 9, 10, 12, and 14.

The peptide samples from both data sets were analyzed by liquid chromatography-tandem

mass spectrometry (LC-MS/MS) to discern peptide abundance, isotope incorporation, and sequences. Protein turnover kinetics and estimated fitting errors were analyzed through "Proturn"[103]. Additional details of the data set can be found in previous publications[98, 99].

4.2.2 Construction of the Data Multiple Imputation (DMI) Pipeline

We incorporated FCS in our pipeline using the R package "MICE"[104]. We formatted the data from both data sets as a proteome-wide time series of A0 (the fraction of the zeroth isotopomer of a peptide isotope envelope, which is used to estimate the protein turnover rate). For the murine data set, this was done for each mouse strain in each condition (ISO/CTRL), and for the human data set, for each healthy subject. Missing A0 values at any given time point were imputed based on the remaining time points. If multiple A0s from different peptides in the same proteins exist, the median of the A0s was used. The imputation was performed on the peptides that have at least two observed time points; this is not to be confused with the requirement of four time points to perform the turnover rate estimation. We used FCS to reproduce the correlations over time and set the number of imputed data sets, m, to 10. Subsequently, we performed half-life computation with "Proturn" on the 10 resulting data sets separately, with identical settings. For any given protein, the final turnover rate constant k is the average rate constant estimated from 10 runs of half-life analyses. This process is repeated for each of the 12 samples, i.e., 6 samples under both ISO-treated and CTRL conditions, in the murine data set and for each of the 10 healthy subjects in the human data set. Compared to previous work, this pipeline is flexible to accommodate other types of DMI techniques and larger m, and provides a platform for comparing different approaches for missing data.

Proturn for Computing Protein Turnover Rates

"Proturn" was used to calculate protein turnover kinetics and estimated fitting errors as previously described[105]. "Proturn" automatically retrieved identified peptides that were uniquely assigned to proteins for the area integration. The "Proturn" parameters were

set as follows: area-under-curve integration width: 60 ppm, extracted ion chromatogram smoothing: Savitzky–Golay filter over 7 data points. To further control against peptide false positive identifications, only peptides that were explicitly identified (1% FDR) and integrated in greater than 4 time points were accepted for the calculation of protein abundance and turnover.

Evaluation Framework for Missing Data Imputation

To simulate missing data scenarios, we first retrieved peptides from the murine cardiac temporal proteomics data set that contained a complete time series in A0 with no missing values, such that we can ensure that the turnover rate is estimated without missing values and can serve as a ground truth for evaluating the imputation methods. To simulate the different levels of missingness, we create five masked data sets where 1 up to 5 time points out of the complete 7 time points were randomly masked. On each of these masked data sets, we applied three imputation methods: (1) DMI; (2) Single imputation with mean; and (3) Single imputation with k-nearest neighbor (KNN) using 30 neighbors. Each masked data set that underwent the Data Single Imputation (DSI) workflow produced one imputed data set. Each masked data set that underwent the DMI workflow produced 10 imputed data sets for each of the masked configurations. Subsequently, we conducted kinetic analysis to quantify the turnover rates on each masked data set for each imputation method independently. The accuracy of the imputation methods was quantified using the normalized root-mean-square error (NRMSE) comparing the actual values versus the imputed values for A0 and turnover rates.

4.2.3 Impact of DMI on Biomedical Insights

Summary of Number of Samples Available for Turnover Calculation with a Barplot

For each time series of a specific protein from different experimental conditions (6 strains $\times$ 2 treatments = 12 conditions), the number of nonmissing data points were counted (ranging from 0 to 7) by picking the peptide with the least missing values in the time series. The

counts from different experimental conditions for the same protein are then aggregated to yield the total number of observations and the number of missing observations imputed for that protein. Proteins are sorted by the number of observations in the barplot. The barplot showing the numbers of proteins recovered by DMI under different conditions follows the same procedure.

Protein Expression Comparison on Proteins Quantifiable with or without DMI

Violin plots compare the abundance value (normalized spectral abundance factor, NSAF) and turnover rates between proteins only quantifiable by DMI and those quantifiable without DMI. The area of each violin is adjusted to reflect the number of proteins. A two-sample two-sided Wilcoxon test is performed, and the p-value is shown in the figure. The Wilcoxon test is performed in R using wilcox.test.

Reactome Pathway Enrichment Analysis

Reactome database was used to analyze the biological processes associated with the identified proteins, including those recovered through imputation methods. We performed Reactome Pathway enrichment analysis with the following settings: Mus musculus genes as the reference list; biological process complete as the annotation data set; Fisher's Exact test and calculate FDR. The analysis was specifically designed to pinpoint biological processes that are significantly enriched in our data set of proteins, with an emphasis on contrasting those proteins identified through DMI with those not subjected to DMI. Biological processes that are only enriched in the protein list subjected to DMI are shown.

Protein Complex Stability Analysis

Protein complex information was retrieved from Complex Portal[106]. We selected complexes for which all protein interactors in the complex were represented in the proteomics data set and focused on heterocomplexes, i.e., complexes with multiple protein interactors. Stability is calculated as the standard deviation of the average protein turnover rates within the protein complex. To compare against proteins sampled from the proteome, we account for the number of proteins in the complex by sampling from the proteome with the em-

pirical frequency of the number of proteins in complexes. A Wilcoxon Test was performed to calculate the p-values. We also analyzed the dynamics of individual protein complexes across the experimental groups. Using one-way Analysis of Variance (ANOVA), we examined differences in the mean turnover rates of protein interactors in four complexes.

Biomarker Analysis on Human Temporal Proteomics Data set

MarkerDB is a professionally curated database of preclinical biomarkers[107]. From this database, we identified 137 unique protein biomarkers and retrieved their UniProt IDs using UniProt KB API[108]. We identified the intersection of these biomarkers and proteome quantified with and without imputation in the human temporal proteomics data set. We then queried MarkerDB to map the biomarker lists of each human subject to their disease associations in order to identify new or corroborated disease associations revealed by the additional imputed proteins.

4.3 Results and Discussion

4.3.1 The DMI Pipeline to Recover Temporal Proteomics Data with Flexibility

We developed a DMI pipeline capable of imputing missing values in temporal proteomics data, rendering greater coverage of protein turnover rates. Our workflow (4.1B) first pre-processes the temporal proteomics data set to fit the format required by DMI. DMI is then performed to impute missing values for m rounds, where m is predefined. The resulting m imputed data sets allow quantification of protein turnover rates for all identified proteins, a task that would have been challenging, and sometimes infeasible, with incomplete data sets. Kinetic analyses are performed on these data sets separately, leading to M estimates of protein turnover rates. Finally, all estimates are pooled to generate the final turnover rates, proteome wide.

4.3.2 The DMI Pipeline Enhances the Final Determination of Protein Turnover Rates

Our DMI pipeline is able to fully utilize the information that can be extracted about proteome dynamics from the temporal proteomics data sets. In the previous analysis, peptides identified at least 4 times were selected to control false discovery rate of protein turnover quantification. The requirement for a minimal number of time points is to ensure adequate information for accurate turnover rate estimation. Our DMI pipeline captures a more complete proteome-wide turnover rate in both data sets. Thus, proteins that were previously quantifiable (¿4 time points) but not present in the full time points also benefit from inclusion of DMI-imputed data for more accurate kinetic analysis. A detailed number of imputed samples and original samples for both data sets are shown in 4.2.

We evaluated the performance of DMI on imputing missing values in comparison to single imputation methods (DSI). We developed an approach to introduce missing values by masking experimentally observed values for peptides' with a complete time-series. To examine the temporal aspects of the imputation, we evaluated how well each imputation method can recover masked values and subsequently estimate turnover rates from the imputed time series. Across various levels of missingness, DMI consistently outperformed k-nearest neighbor (KNN) imputation and mean imputation in accurately imputing experimentally observed A0 values and turnover rates as measured by NRMSE (4.6).

4.3.3 The DMI Pipeline Ensures a Comprehensive View of Protein Turnover Rates

A detailed number of proteins quantifiable after imputation in each mouse strain under two conditions is shown in 4.3C. Around 50% improvement of coverage is shown in all strains under both conditions. With the improved coverage, we have a more comprehensive view of the proteome dynamics landscape during cardiac hypertrophy pathogenesis.

As previously demonstrated with proteomics data, missing values are correlated with low abundances of the protein, i.e., proteins with low abundance were prone to contain missing

values. We investigated whether low abundance also correlates with the missing values in the protein turnover rate. We further explored this relationship in the context of protein turnover rates. Specifically, we compared the abundance levels and turnover rates of proteins that can be quantified without DMI to those that are only quantifiable with DMI (4.3A and B). A significant difference in both the abundance and turnover rate between these two groups in all strains and two treatments suggests that the proteins with lower abundance and higher turnover are prone to be missing in the turnover rate calculation. Thus, DMI enables proteins with lower expression to be captured, ensuring a more comprehensive view of proteome wide protein dynamics (4.3B).

4.3.4 The DMI Pipeline Captures a Broad Representation of Biological Processes

To investigate how an imputed data set can better capture the comprehensive biological processes of the proteome, we performed the Reactome Pathway enrichment analysis on both the protein sets before and after imputation to determine the potential loss of biological processes if no imputation is performed. There were 199 and 238 biological processes enriched from the proteins recovered with imputation in health and disease, respectively (4.7). In the healthy group, biological processes related to localization, autophagy, splicing and so on are enriched. In the disease group, biological processes related to transportation, splicing, and autophagy are enriched. While the recovered biological processes in the two groups were not the same, they share common pathways in terms of high-level processes such as splicing, localization and autophagy.

4.3.5 The DMI Pipeline Reveals a Dynamic Landscape on Protein Complexes

The turnover rate of individual proteins within protein complexes offers insights into their stability, regulatory mechanisms, and functional lifespans, enhancing our understanding of cellular biology[109]. We investigated the turnover rate landscape of multiple heterocomplex interactors, revealing the dynamic view of protein complexes.

We first explored the impact of DMI on proteome-wide turnover rates, revealing that DMI elucidates a detailed proteome turnover landscape (4.4A). While the majority of proteins show relatively consistent turnover rates before and after DMI, we observed increases and decreases of turnover rates as a result of increased time points imputed by DMI. The proteins that have lower turnover rates after imputation seem to have a large discrepancy before and after DMI. This discrepancy likely arises because these protein turnover rates, when quantified without DMI, are challenging to measure due to the high proportion of missing values that lead to fewer data points and greater variation across replicates. Subsequently, we investigated the turnover rates of proteins within heterocomplexes, characterized by the Complex Portal database. We defined a metric, the standard deviation of turnover rates, as a measure of the synchronization of turnovers within protein complexes. A lower standard deviation signifies a more coordinated complex, characterized by similar level protein turnover rates. Our analysis demonstrated that the synchronization of protein complexes was significantly greater than that observed for proteins sampled from the proteome, suggesting a coordinated regulation of turnover within the complexes (4.4B).

The turnover landscape offered by DMI allowed for an understanding of how individual complex dynamics may be coordinated across experimental groups (4.4C). Importantly, the ability to assess the dynamics of all protein interactors in certain heterocomplexes is only made possible by DMI (e.g., CPX-5868, 4921, 3035, 3027). We observed that, in some cases, DMI quantified turnover rates demonstrate alignment with the quantified turnover rates obtained without DMI in terms of the synchronization among heterocomplex interactors in the ISO and CTRL conditions (e.g., CPX-2055, 16).

We also observed DMI quantified turnover to provide insight into the change in complex synchronization between the ISO and CTRL conditions. We zoomed in to analyze a select number of these complexes where turnover exhibited incoherence in the ISO experimental group, yet suggested coherence in the CTRL group: (1) UBC13-UEV1A ubiquitin-conjugating enzyme E2 complex; (2) Mitochondrial NIAUFX iron–sulfur cluster (ISC) assembly complex; (3) AP-2 Adaptor complex, alpha1 variant; (4) Laminin-211 complex (4.4D). We further compared the change in coherence (one way ANOVA). The analysis indicated

a decrease in coherence across all four complexes, suggesting that mismatches in turnover rates within complexes critical to cardiac function could play a role in the pathophysiology of heart failure: (1) The ubiquitin-conjugating enzyme complex plays a key role in the process of eliminating damaged and/or misfolded proteins in response to cardiac stress[110]; (2) The Mitochondrial NIAUFX iron–sulfur cluster (ISC) assembly complex is required for the de novo synthesis of iron–sulfur (Fe–S) clusters within mitochondria. Defects in ISC biogenesis are associated with disorders of mitochondrial import, export, and translation and have been linked with cardiomyopathies[111]; (3) AP2, a membrane-bound complex, interacts with clathrin in the plasma membrane to form clathrin-coated vesicles, controlling intracellular trafficking in endocytosis and playing a crucial role in autophagy and lysosomal protein degradation[112]; (4) Laminin 211, an extracellular matrix protein, functions to stabilize the basement membrane and muscle fibers during cardiac contraction[113]. This analysis underscores the utility of DMI in proteomics, providing preliminary insights into protein dynamics that merit further investigation.

4.3.6 The DMI Pipeline Recovers Dynamics of Potential Biomarkers

To further demonstrate the capabilities and effectiveness of our Data Multiple Imputation (DMI) pipeline, we applied our workflow to a human plasma temporal proteomics data set. Similarly, DMI significantly enhanced the number of proteins that can be quantified in each subject by an additional 60% (4.5A). This substantial improvement in protein coverage allows for an improved understanding of the proteome dynamics landscape, thereby broadening the scope of potential clinical applications.

To illustrate a clinical application, we investigated whether additional DMI recovered biomarkers can be quantified. A list of biomarkers from MarkerDB was retrieved and compared with the protein list generated with and without the application of DMI. Our analysis revealed that DMI successfully recovered an additional 2–3 biomarkers per subject on top of the original 10 biomarkers (4.5B). To assess the potential of the additionally identifiable biomarkers to impact diagnostic and prognostic assessments, we obtained biomarker–disease

associations curated from MarkerDB. We observed that certain biomarkers can be highly specific to particular diseases when outside their normal ranges. However, most biomarkers can be less specific and indicative of a family of diseases (e.g., C-reactive protein can be associated with any host of inflammation-related diseases, while human growth hormone can be linked to growth deficiency or acromegaly). Therefore, the ability of imputation to capture additional plasma biomarkers has high clinical utility. It can provide additional corroboration for a specific disease differential, confirm the absence of disease, or indicate the potential of other disease (4.5C and 4.8). This comprehensive biomarker profile helps strengthen the overall differential diagnosis and directs the clinician toward further clinical investigation.

Temporal cardiovascular proteome dynamics studies often suffer from missing data problems, and it hinders our ability to gain insights from these valuable data resources. In many cases, mechanisms contributing to missing values are complex and typically stem from a combination of Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not At Random (MNAR). Therefore, methods that can accommodate various combinations of missing data patterns are necessary. The DMI method discussed herein is effective for handling MCAR and MAR data but can also accommodate MNAR patterns followed by some sensitivity analysis, thus addressing various types of missing data scenarios. However, it is advisable to select specific imputation methods tailored to the nature of the missing mechanism when such information is known or strongly assumed.

Our DMI pipeline allows users to adjust the parameters of imputation to meet the demands of their proteomics data analysis in the following aspects: it provides a default regression model but allows users to choose preferred regression methods in the multiple imputation process; allows users to specify the minimum samples required for imputation, which depends on the specific experimental design; allows selection of the number of data sets, m, for multiple imputation, which should be chosen based on the computational resources available and reliability desired.

As demonstrated in our study, a primary advantage of the DMI pipeline is to better address uncertainties in handling missing data compared with ad hoc or single imputation

methods. We showed the benefit of our DMI pipeline for protein turnover rates inference by applying it to the cardiac temporal data sets.

4.4 Discussion

Missing values is a common issue in MS-based proteomics studies and especially in proteome dynamics data sets. Our DMI pipeline successfully addressed missing data challenges and demonstrated its utility on two distinct existing temporal proteomics data set. In brief, the DMI pipeline captured additional protein turnover rates. These recovered protein dynamics enable a more detailed view of biological pathways, protein complexes, and plasma biomarkers previously obscured, thereby enhancing our understanding of biological insights into the underlying protein dynamics in cardiovascular diseases. In summary, our DMI pipeline can expand the scope of proteome characterization in temporal data sets.

Additionally, Pretraining large omics datasets to learn inherent biological patterns has recently gained traction—particularly in single-cell transcriptomics[65, 114, 115]. We have followed the development of this direction with enthusiasm. We envision such approaches can potentially be adapted to proteomics investigations for tasks such as missing value imputation. Despite its potential promise, challenges for proteomic applications remain problematic, including limited data points (typically, each experiment contributes only a single data point) and batch effects. Unfortunately, implementing a pretraining strategy for missing value imputation in proteomics is beyond the scope of the current study. We are committed to explore this intriguing question in our future endeavors.

4.5 Code and data availability

The code of the work is available at:
https://github.com/yuyanislearning/temporal-preteomics-multiple-imputation.git.

4.6 Acknowledgments

This chapter is based on my joint work with Baradwaj Simha Sankar, Dr. Bilal Mirza, and my Ph.D. advisor Dr. Peipei Ping.

4.7 Figures

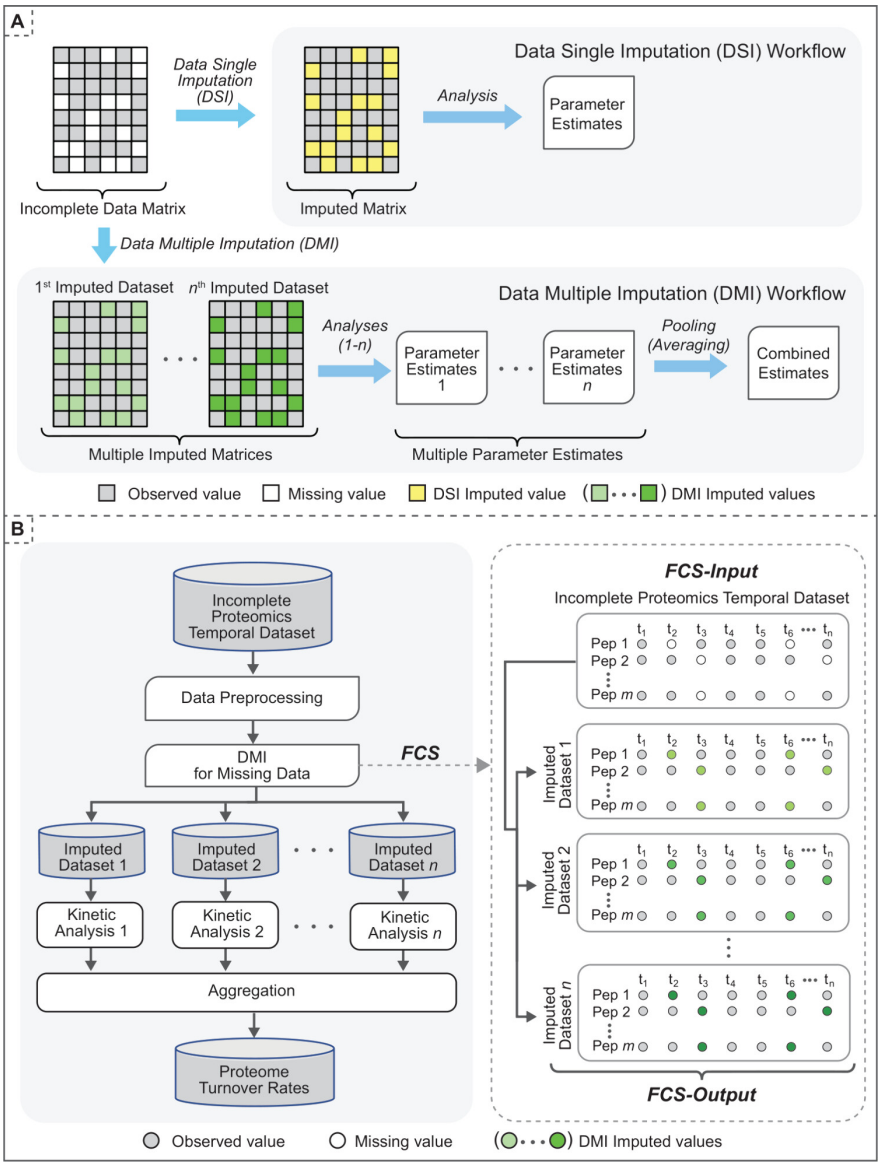


Figure 4.1: Data imputation workflows.

(A) Data Single Imputation (DSI) and Data Multiple Imputation (DMI). In the DSI approach, each missing value (white cell) in the incomplete data matrix is replaced with a single estimate (yellow cells). Imputed values are treated as observed values in the imputed data matrix for downstream analysis. In the DMI approach, multiple values are imputed for each missing value in the incomplete data matrix. Consequently, there are multiple imputed data matrices with the same observed values but different imputed values (green cells). Analysis of each imputed data matrix is performed separately, and the final estimates are obtained by pooling the results from multiple analyses. (B) DMI for missing values in Proteome turnover Data set. The DMI pipeline computed protein turnovers from an incomplete temporal data set (peptide isotope intensities). As data preprocessing, we included peptides detected at >2 time points. The missing values were imputed using Fully Conditional Specification (FCS). DMI generated 10 imputed data sets in which peptide isotope intensity values are imputed at each of the time points (t1 – tn) when/if the data was missing. Each data set has the same observed values but slightly different imputed values. Kinetic analysis (25) was performed on each imputed data set independently, and the protein turnover rates were obtained by averaging the results of multiple analyses.

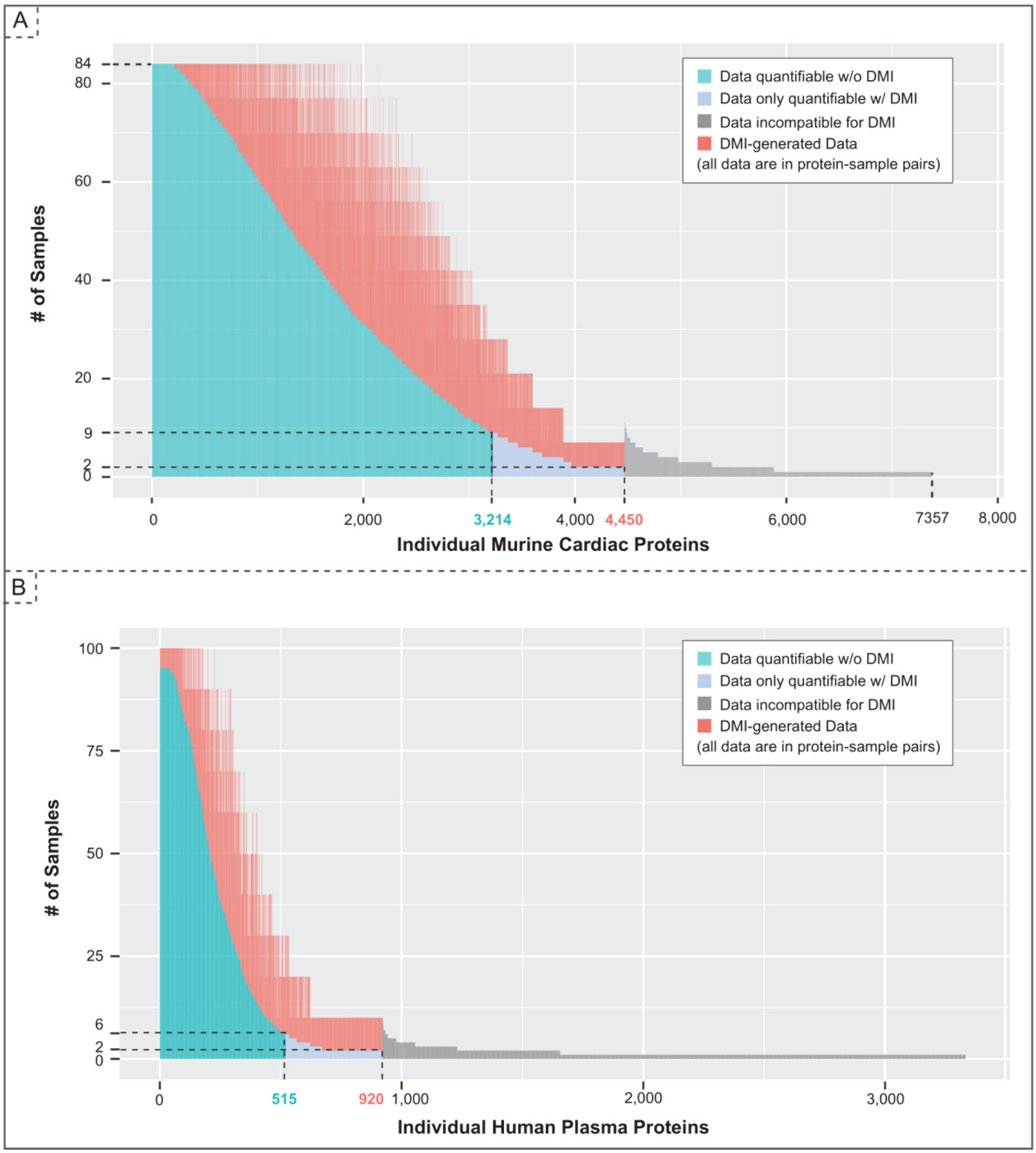


Figure 4.2: DMI improves coverage of the proteome turnover rates

Supporting evidence from two independent data sets are presented here. (A) The mouse data set contains 84 samples (6 strains × 2 treatments × 7 time points). The individual proteins are represented in the x-axis in decreasing order of samples, where their turnover rates were quantifiable without (blue) DMI and with DMI (red). Without DMI, the turnover rate of 3,214 proteins (in dark blue) were quantified. With DMI, the turnover rates of 1,236 (38%) additional individual proteins were quantified (in light blue), capturing a total of 4,450 protein turnover rates. Only a small fraction of samples (in gray, 2,907 proteins) did not satisfy our minimum requirement for imputation. (B) The human plasma data set consists of 100 samples (10 subjects × 10 time points). Similarly, without DMI, the turnover rates of 515 proteins (in dark blue) were quantified. With DMI, the turnover rate of 405 (78%) additional individual proteins were quantified (in light blue), capturing a total of 920 protein turnover rates.

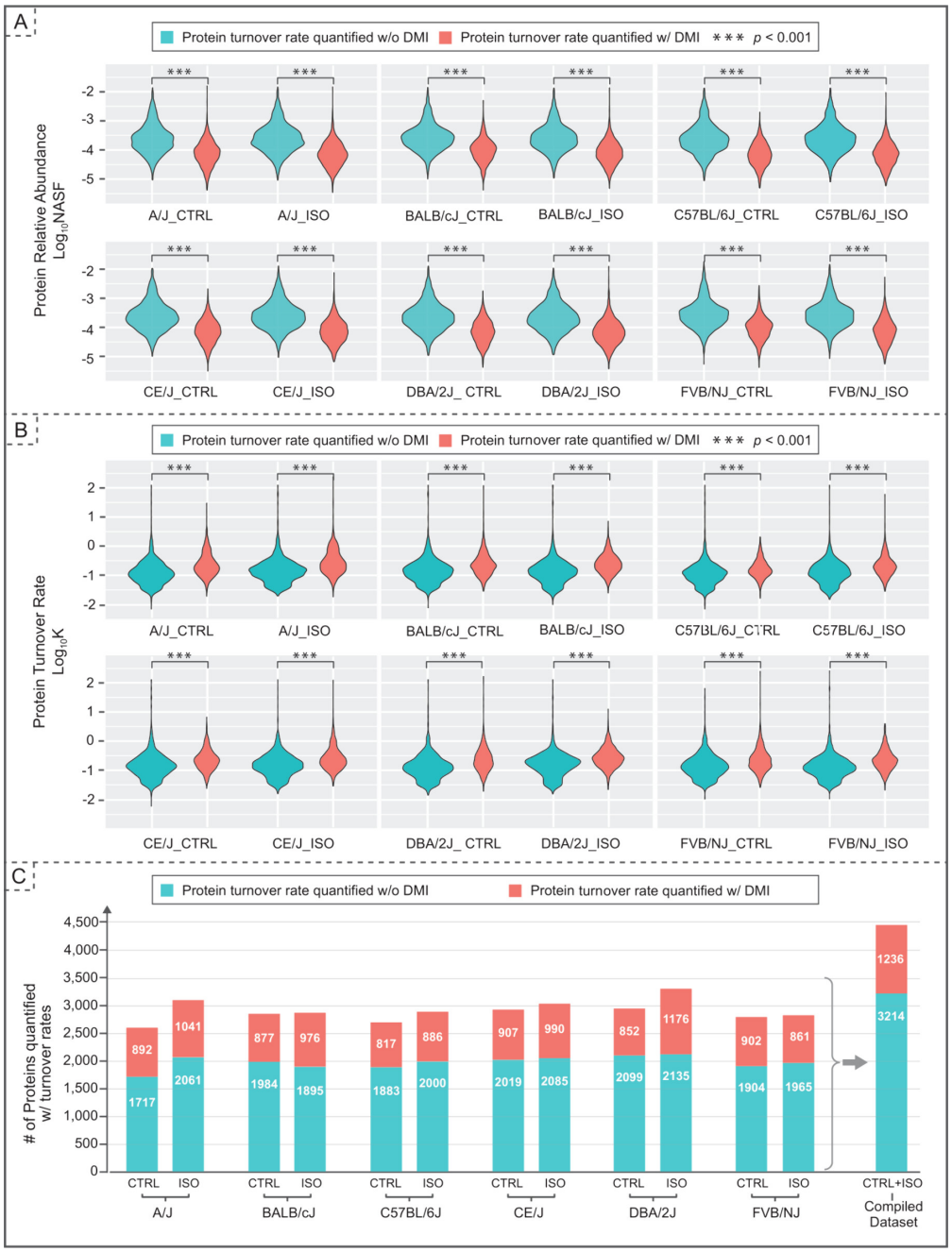


Figure 4.3: Impact of DMI on protein expression and turnover rate

(A) Violin plot shows the protein relative abundance of those with DMI (orange) and those without (blue), indicating that DMI has a more pronounced impact on proteins of lower abundance. (B) Violin plot shows the protein turnover rate computed from the data set with or without DMI, illustrating the DMI has a bigger influence on proteins with faster turnover rates. Statistical significance between groups in both violin plot is determined using the Wilcoxon test (***p-value <0.001). (C) Bar chart compares the quantifiable protein turnover rates with and without data imputation across six mouse strains. Data imputation leads to a 40–50% increase (orange) in the quantifiable turnover rates in each strain.

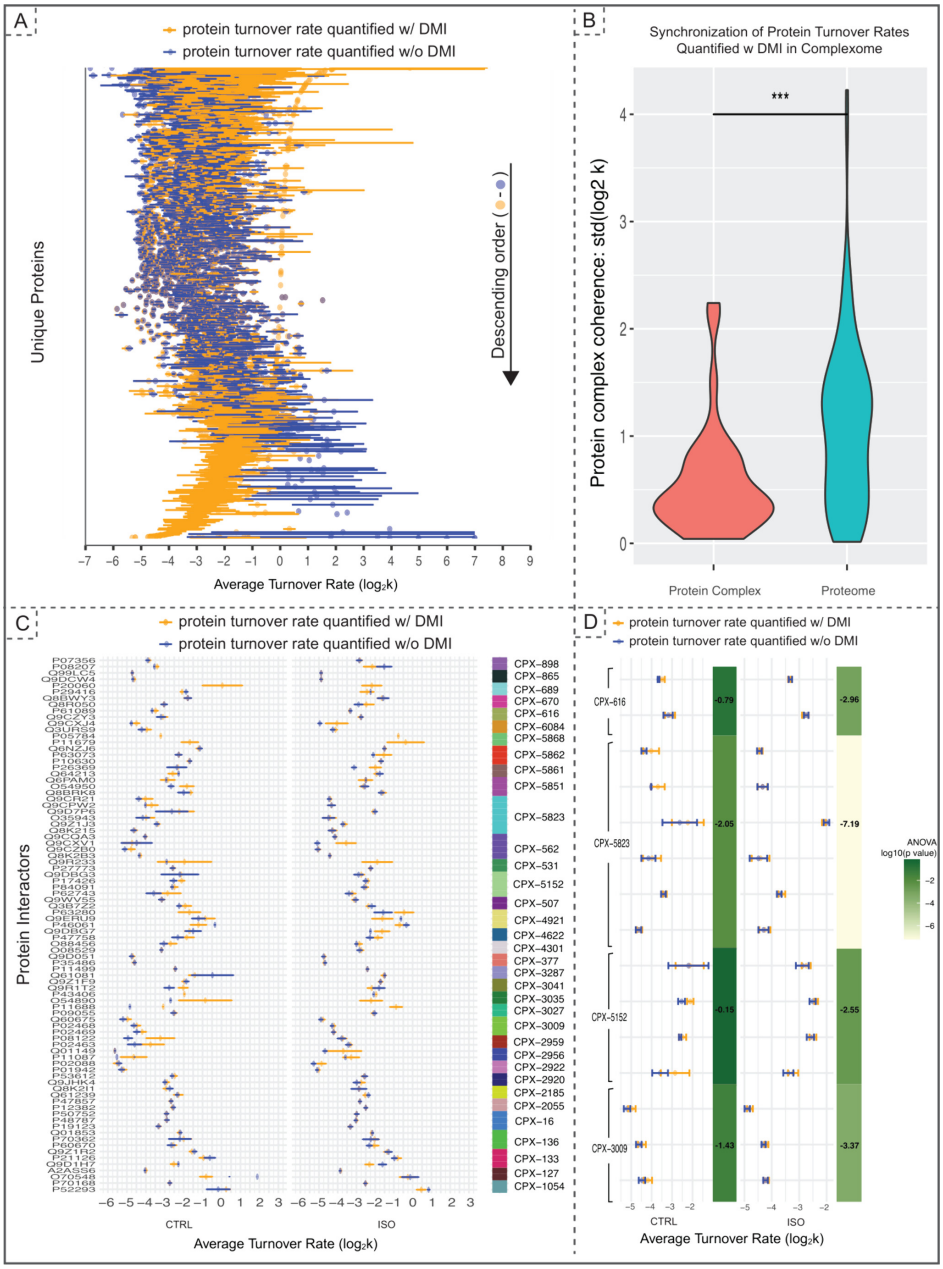


Figure 4.4: Impact of DMI on protein complex dynamics

(A) Scatter plot of proteome turnover rates from top to bottom based on the absolute impact of DMI on turnover rate estimations: enhancement (on the top), agreement (in the middle), reduction (in the lower part), or the assignment of an imputed value, previously unquantifiable in the absence of DMI. Each row represents a protein, and the rows are organized in a descending order of the difference between protein turnover rates estimated after and before imputation. Error bars represent standard error mean (SEM); they are 0 if $n < 2$. (B) A violin plot shows a pronounced synchronization of turnover rates among proteins within complexes, as evidenced by the standard deviation of the turnover rates quantified post-DMI compared to the broader proteome. "***" indicates a p-value <0.001. (C) A scatter plot of protein turnover rate within individual complexes, showing the impact of DMI on assessing the dynamic behavior of proteins within the same complex. A color bar indicates the protein complex the protein interactors belong to. Detailed examples are given in panel D. (D) A zoom in view of four protein complexes selected from panel C: UBC13-UEV1A ubiquitin-conjugating enzyme E2 complex; Mitochondrial NIAUFX iron–sulfur cluster assembly complex; AP-2 Adaptor complex, alpha1 variant; Laminin-211 complex, where DMI provides insight into the synchronized protein turnover behavior in CTRL which was disrupted in ISO.

Figure 4.5: DMI pipeline enhances protein quantification in human samples

The bar chart presents a comparison of protein turnover rates quantified with and without Data Imputation (DMI) across 10 human subjects when examining both the plasma proteome (A) and the biomarkers it carries (B). The application of data imputation results in a significant increase in quantifiable protein turnover rates, with a 60–70% improvement observed in the proteome and a 10% enhancement noted in individual biomarkers. (C) We elucidate biomarker–disease associations in the data set gained with DMI, it reveals three patterns: (1) new disease associations (e.g., HS8: C-reactive protein → Hypertension); (2) new evidence supporting existing disease association (e.g., HS3: Lysozyme C and IL-2 receptor subunit alpha → Pulmonary Sarcoidosis); and (3) adding to the list of markers pre-DMI (e.g., HS3: Ferritin or HS8: Fibrinogen).

4.8 Supplementary materials

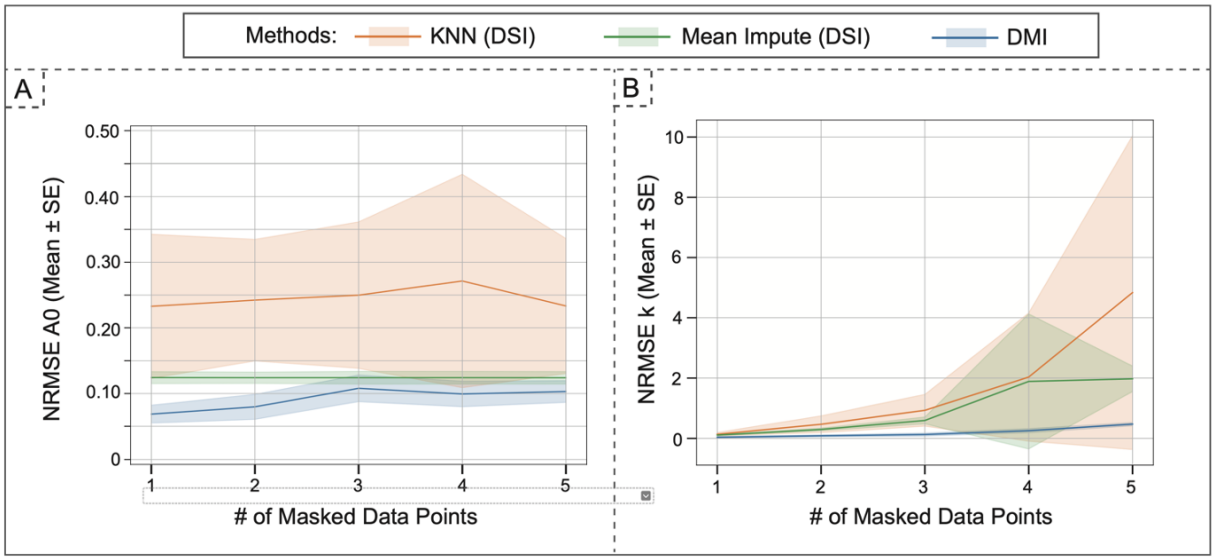


Figure 4.6: DMI improves coverage of the proteome turnover rates

To simulate the different levels of missingness, we have created five masked scenarios in silico, where 1 time point and up to 5 time points out of the complete 7 time points were randomly masked. On each of these masked datasets (scenarios), we applied three imputation methods to recover the datasets: 1) DMI; 2) Single Imputation using Mean (DSI); and 3) Single Imputation using K-nearest neighbor (DSI). Each masked dataset undergone the DSI workflow produced one imputed dataset, resulting in a total of 5 imputed datasets. Each masked dataset that underwent the DMI workflow produced 10 imputed datasets for each of the masked configurations, resulting in a total of 50 imputed datasets. The imputed A0 values were compared against the ground-truth A0 values to evaluate the ability of imputation methods that can faithfully recover the original data. Subsequently, we conducted kinetic analysis to quantify the turnover rates on each masked dataset for each imputation method independently. The turnover rate estimated using imputed A0 time series were compared against the ones using completely observed A0 time series to evaluate the ability of imputation methods to capture the temporal dependencies among values. The accuracy of the imputation methods was quantified using the normalized root mean square error (NRMSE). As illustrated in the figure, DMI (blue) consistently demonstrates superior performance in recovering missing data, evidenced by the lower Normalized Root Mean Square Error (NRMSE) values and narrower Standard Error (SE) range in comparison to DSI methods in either single time point level (A0 in panel A) or time series level (turnover rate in panel B). In addition, DMI is more resilient to higher levels of missingness compared to DSI, as its NRMSE and SE are less affected with the increasing number of masked data points.

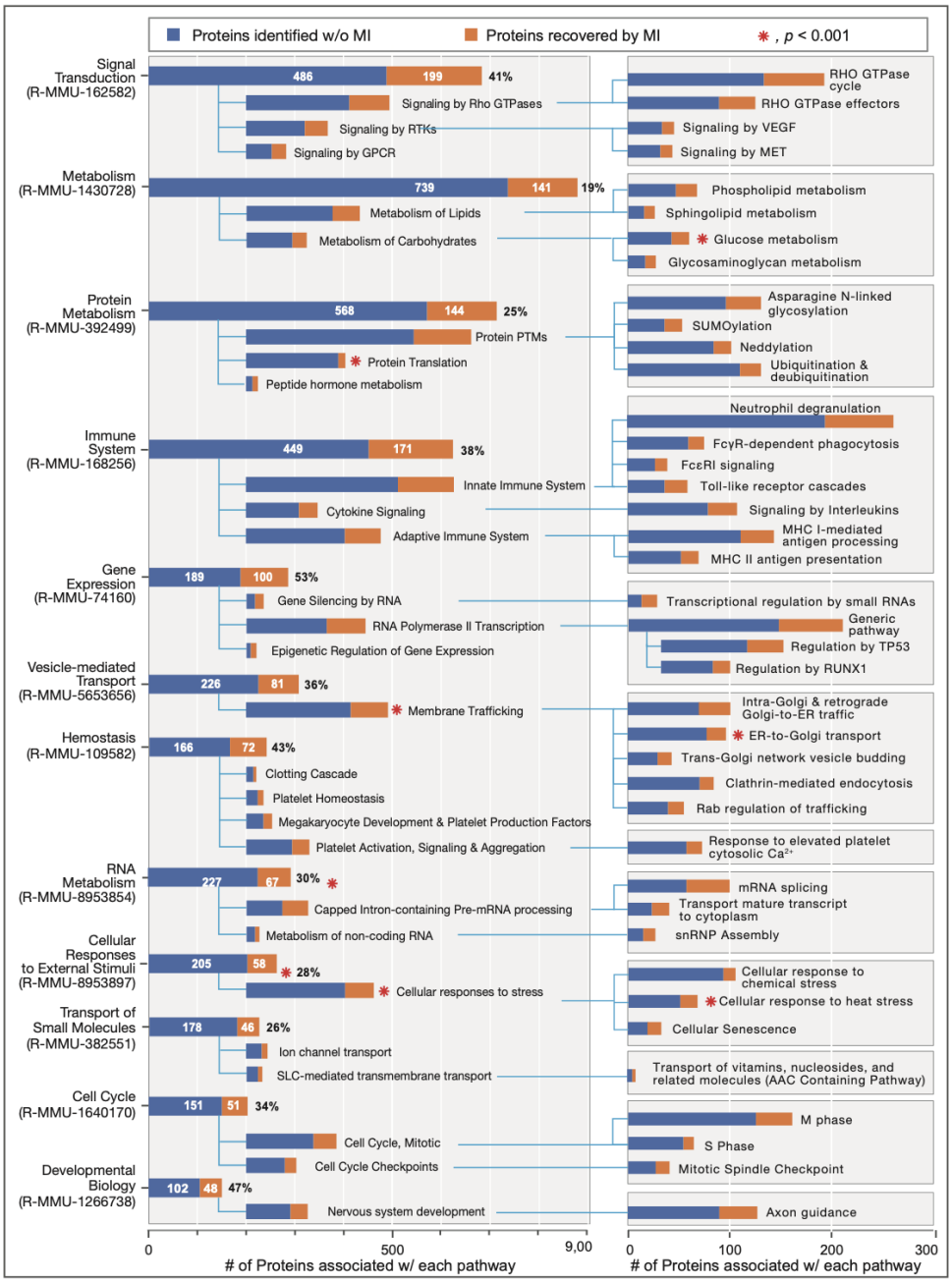


Figure 4.7: DMI enhances protein turnover rate detection in biological pathways

The DMI pipeline leads to enhanced protein turnover rate detection in multiple biological pathways, but it has limited impact on others. For example, in the "Protein Metabolism" pathway, DMI captures the turnover rates of additional 144 proteins (in orange) on top of the 568 proteins (in blue) without imputation. A "*" sign indicates a significant enrichment of the pathway with the recovered proteins based on pathway enrichment analysis.

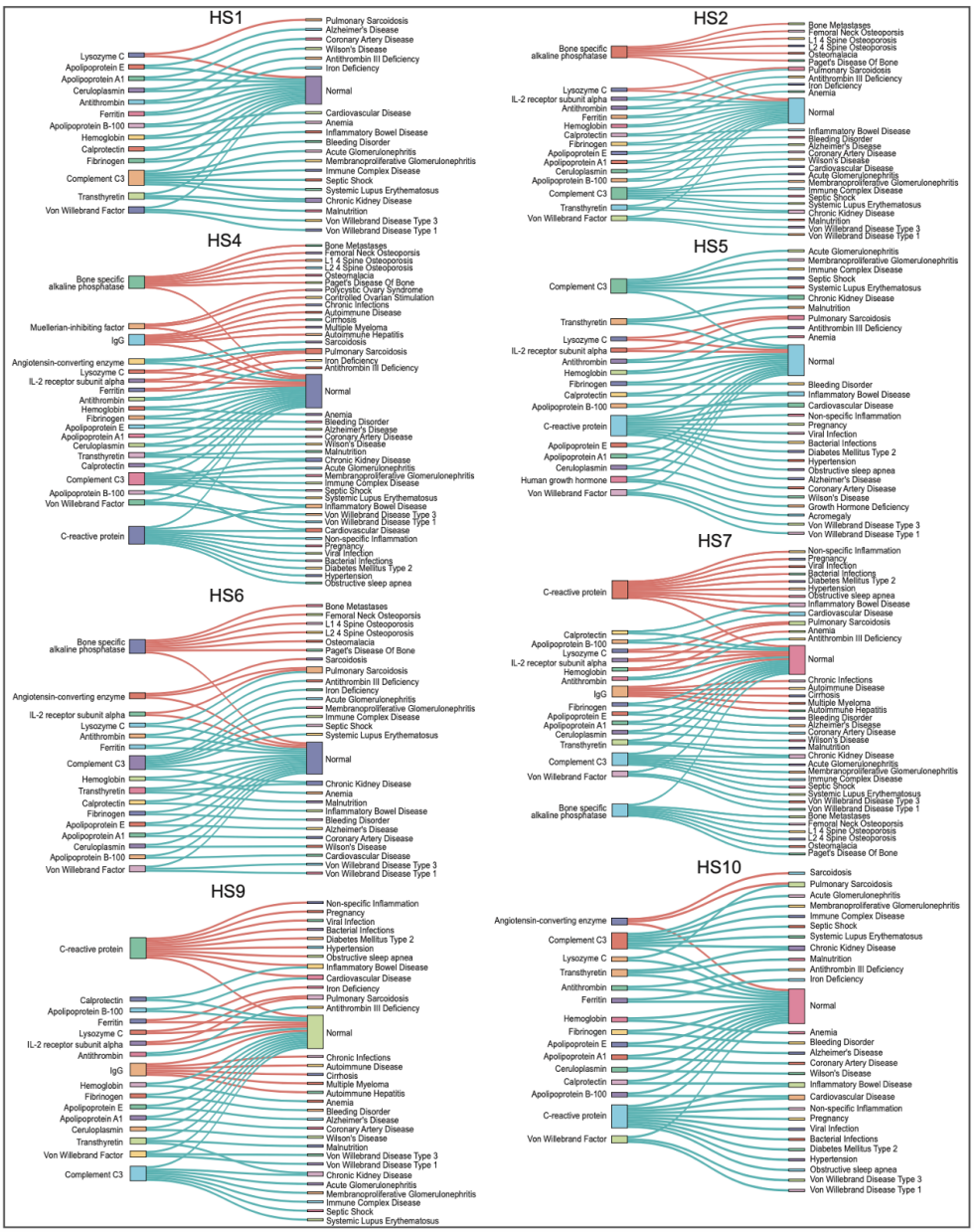


Figure 4.8: DMI recovers dynamics of potential biomarkers

Analysis of biomarker-disease associations across subjects reveals that DMI-imputed biomarkers either strengthen existing disease associations or provide evidence for new ones. A total of 30 additional biomarkers are discovered, 2 of which are new biomarkers only discovered with DMI across 10 subjects. This leads to a total of 116 additional biomarker-disease associations, 7 of which are new biomarker-disease associations that were not detected pre-imputation.

CHAPTER 5

Summary and future directions

In this article, we discussed three topics regarding AI/ML applications in proteomics and their corresponding applications. In this section, I summarize the three computational methods that I developed and discuss some potential future directions.

5.1 MIND-S: A Deep-Learning Prediction Model for Elucidating Protein Post-Translational Modifications in Human Diseases

In Chapter 2, we proposed MIND-S, a deep learning model for predicting protein post-translational modifications (PTMs) with high accuracy, efficiency, and interpretability. MIND-S leverages information from protein sequences and structures to predict potential PTM locations for a total of 26 different PTM types.

We have three future directions for this project. Our ablation study indicates that the structure module in MIND-S can be further enhanced to fully leverage the information retained in protein structures. We plan to incorporate 3D structural data by considering residual sidechain directions, secondary structure elements, and local 3D conformations, in addition to spatial proximity.

Another direction will be to further investigate the proportion of known PTM motif with the saliency-weighted positional weight matrix. Exploring the proportion of known motifs within the saliency patterns generated by MIND-S could serve as a validation of our model interpretability. Furthermore, it could potentially be a new approach for novel PTM motif discovery.

A remaining challenge is predicting PTMs in a condition-specific manner. PTMs have

been shown to differ across cell types and disease states in human samples, due to intrinsic factors that affect the modification process within a given cell or tissue type. As such, a condition-specific PTM prediction model is essential for accurately mapping the PTM landscape in biological systems. Although MIND-S can statically predict PTMs or identify potential target sites, it cannot yet determine whether a PTM will occur under a specific cellular condition. Incorporating context-specific information, such as gene expression data, may enhance these predictions and lead to more accurate, condition-specific outcomes.

5.2 Systematic Evaluation and Integration of Multi-Modal Gene Embeddings

In Chapter 3, we systematically investigated the overlap, measured by SVCCA, a method calculate the maximized correlation in a transformed space, between different gene embeddings and uncovered the diversity inherent in these representations. We further integrated gene embeddings to create a more holistic view of genes and proteins, thereby benefiting downstream tasks in the proteomics domain.

For future directions, it will be valuable to incorporate additional gene, protein, and RNA embeddings to further enrich the representation. We plan to extend our integration framework to include state-of-the-art embeddings, thereby capturing a more comprehensive picture of biological systems. This expanded representation will enable us to explore how different embeddings complement one another.

Given the model's efficiency, it is feasible to develop a user-friendly service that generates customized embeddings. Such a service would allow researchers to select specific representations tailored to their preferences or particular downstream tasks, with the flexibility to adjust parameters in real time. Additionally, incorporating condition-specific information, such as tissue-specific gene expression profiles or environmental factors, could further refine these embeddings to support more targeted predictions.

5.3 Missing Values in Longitudinal Proteome Dynamics Studies: Making a Case for Data Multiple Imputation

In Chapter 4, we proposed a multiple imputation pipeline for temporal proteomics, specifically designed to estimate proteome turnover rates. Our method demonstrated superior performance in recovering missing data, thereby improving the overall proteome coverage and enabling a more detailed understanding of protein dynamics.

For future directions, we aim to adapt and extend our pipeline to accommodate a wider variety of data formats and protein abundance measurements. One area of focus will be enhancing the pipeline's flexibility to integrate data from different experimental platforms, such as label-free quantification, SILAC, and TMT, as well as other emerging technologies. This will not only increase the robustness of our approach but also ensure its applicability across diverse experimental setups. Additionally, we plan to incorporate advanced statistical and machine learning techniques in the DMI to refine the imputation process, which could further improve the accuracy and reliability of proteome turnover estimations. By enabling more precise modeling of protein dynamics, these enhancements will pave the way for deeper insights into cellular processes, disease mechanisms, and potential therapeutic targets.

We will consider adapting a different schema that leveraging the existing proteomics dataset to boost the imputation of incoming proteomics dataset. Potentially, a model that pretrained on large corpus of proteomics datasets can learn inherent biological patterns among proteins and be used to impute missing data leveraging the learnt relationship. We are committed to explore this intriguing question in our future endeavors.

Bibliography

[1] Yu Yan et al. "MIND-S is a deep-learning prediction model for elucidating protein post-translational modifications in human diseases". In: *Cell reports methods* 3.3 (2023).

[2] Yu Yan et al. "Missing Values in Longitudinal Proteome Dynamics Studies: Making a Case for Data Multiple Imputation". In: *Journal of Proteome Research* 23.9 (2024), pp. 4151–4162.

[3] Ruedi Aebersold and Matthias Mann. "Mass-spectrometric exploration of proteome structure and function". In: *Nature* 537.7620 (2016), pp. 347–355.

[4] Jürgen Cox and Matthias Mann. "MaxQuant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification". In: *Nature biotechnology* 26.12 (2008), pp. 1367–1372.

[5] Ian Goodfellow. *Deep learning*. 2016.

[6] DG Knorre, NV Kudryashova, and TS Godovikova. "Chemical and functional aspects of posttranslational modification of proteins". In: *Acta Naturae ( )* 1.3 (3) (2009), pp. 29–51.

[7] Shahin Ramazi and Javad Zahiri. "Post-translational modifications in proteins: resources, tools and prediction methods". In: *Database* 2021 (2021), baab012.

[8] Jesper V Olsen and Matthias Mann. "Status of large-scale analysis of post-translational modifications by mass spectrometry". In: *Molecular & cellular proteomics* 12.12 (2013), pp. 3444–3452.

[9] Xiang Chen et al. "Incorporating key position and amino acid residue features to identify general and species-specific Ubiquitin conjugation sites". In: *Bioinformatics* 29.13 (2013), pp. 1614–1622.

[10] Sheraz Naseer et al. "Optimization of serine phosphorylation prediction in proteins by comparing human engineered features and deep representations". In: *Analytical Biochemistry* 615 (2021), p. 114069.

[11] Jeffrey A Ubersax and James E Ferrell Jr. "Mechanisms of specificity in protein phosphorylation". In: *Nature reviews Molecular cell biology* 8.7 (2007), pp. 530–541.

[12] Elise J Needham et al. "Illuminating the dark phosphoproteome". In: *Science signaling* 12.565 (2019), eaau8645.

[13] Baris E Suzek et al. "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches". In: *Bioinformatics* 31.6 (2015), pp. 926–932.

[14] Brice Ozenne, Fabien Subtil, and Delphine Maucort-Boulch. "The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases". In: *Journal of clinical epidemiology* 68.8 (2015), pp. 855–859.

[15] Charles X. Ling, Jin Huang, and Harry Zhang. "AUC: a Statistically Consistent and more Discriminating Measure than Accuracy". In: *International Joint Conference on Artificial Intelligence*. 2003. URL: https://api.semanticscholar.org/CorpusID:118673880.

[16] Lijun Dou et al. "A comprehensive review of the imbalance classification of protein post-translational modifications". In: *Briefings in Bioinformatics* 22.5 (2021), bbab089.

[17] S Hochreiter. "Long Short-term Memory". In: *Neural Computation MIT-Press* (1997).

[18] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[19] Petar Velickovic et al. "Graph Attention Networks". In: *ArXiv* abs/1710.10903 (2017). URL: https://api.semanticscholar.org/CorpusID:3292002.

[20] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[21] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. "Position Information in Transformers: An Overview". In: *Computational Linguistics* 48 (2021), pp. 733–763. URL: https://api.semanticscholar.org/CorpusID:231986066.

[22] Omer Sagi and Lior Rokach. "Ensemble learning: A survey". In: *Wiley interdisciplinary reviews: data mining and knowledge discovery* 8.4 (2018), e1249.

[23] Jianjiong Gao et al. “Musite, a tool for global prediction of general and kinase-specific phosphorylation sites”. In: *Molecular & Cellular Proteomics* 9.12 (2010), pp. 2586–2600.

[24] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic Attribution for Deep Networks”. In: *International Conference on Machine Learning*. 2017. URL: https://api.semanticscholar.org/CorpusID:16747630.

[25] Michael F Chou and Daniel Schwartz. “Biological sequence motif discovery using motif-x”. In: *Current protocols in bioinformatics* 35.1 (2011), pp. 13–15.

[26] Ylva Gavel and Gunnar von Heijne. “Sequence differences between glycosylated and non-glycosylated Asn-X-Thr/Ser acceptor sites: implications for protein engineering”. In: *Protein Engineering, Design and Selection* 3.5 (1990), pp. 433–442.

[27] Richard B Pearson and Bruce E Kemp. “[3] Protein kinase phosphorylation site sequences and consensus specificity motifs: Tabulations”. In: *Methods in enzymology* 200 (1991), pp. 62–81.

[28] Lorenzo A Pinna and Maria Ruzzene. “How do protein kinases recognize their substrates?” In: *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research* 1314.3 (1996), pp. 191–225.

[29] Dale DO Martin et al. “A human huntingtin SNP alters post-translational modification and pathogenic proteolysis of the protein causing Huntington disease”. In: *Scientific Reports* 8.1 (2018), p. 8096.

[30] Jascha T Manschwetus et al. “Binding of the human 14-3-3 isoforms to distinct sites in the leucine-rich repeat kinase 2”. In: *Frontiers in Neuroscience* 14 (2020), p. 302.

[31] Peter V Hornbeck et al. “PhosphoSitePlus, 2014: mutations, PTMs and recalibrations”. In: *Nucleic acids research* 43.D1 (2015), pp. D512–D520.

[32] Philipp Ehlermann et al. “Adverse events in families with hypertrophic or dilated cardiomyopathy and mutations in the MYBPC3 gene”. In: *BMC medical genetics* 9 (2008), pp. 1–11.

[33] Ali El-Armouche et al. “Decreased phosphorylation levels of cardiac myosin-binding protein-C in human and experimental heart failure”. In: *Journal of molecular and cellular cardiology* 43.2 (2007), pp. 223–229.

[34] Carlo Napolitano et al. “Genetic testing in the long QT syndrome: development and validation of an efficient approach to genotyping in clinical practice”. In: *Jama* 294.23 (2005), pp. 2975–2980.

[35] Yuan Liu et al. “Leucine-rich repeat kinase-2 deficiency protected against cardiac remodelling in mice via regulating autophagy formation and degradation”. In: *Journal of Advanced Research* 37 (2022), pp. 107–117.

[36] Dominik Schuettler et al. “A practical guide to setting up pig models for cardiovascular catheterization, electrophysiological assessment and heart disease research”. In: *Lab Animal* 51.2 (2022), pp. 46–67.

[37] George C Gabriel et al. “Cardiovascular development and congenital heart disease modeling in the pig”. In: *Journal of the American Heart Association* 10.14 (2021), e021631.

[38] Tao Li et al. “Defective branched-chain amino acid catabolism disrupts glucose metabolism and sensitizes the heart to ischemia-reperfusion injury”. In: *Cell metabolism* 25.2 (2017), pp. 374–385.

[39] Edward Lau et al. “Integrated omics dissection of proteome dynamics during cardiac remodeling”. In: *Nature communications* 9.1 (2018), p. 120.

[40] The UniProt Consortium. “UniProt: the universal protein knowledgebase in 2021”. In: *Nucleic acids research* 49.D1 (2021), pp. D480–D489.

[41] Peter JA Cock et al. “Biopython: freely available Python tools for computational molecular biology and bioinformatics”. In: *Bioinformatics* 25.11 (2009), p. 1422.

[42] Daniele Grattarola and Cesare Alippi. “Graph Neural Networks in TensorFlow and Keras with Spektral”. In: *IEEE Comput. Intell. Mag.* 16 (2020), pp. 99–106. URL: https://api.semanticscholar.org/CorpusID:219966138.

[43] Milo M Lin and Ahmed H Zewail. "Hydrophobic forces and the length limit of foldable protein domains". In: *Proceedings of the National Academy of Sciences* 109.25 (2012), pp. 9851–9856.

[44] Justin M Johnson and Taghi M Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of big data* 6.1 (2019), pp. 1–54.

[45] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *ArXiv* abs/1201.0490 (2011). URL: https://api.semanticscholar.org/CorpusID:10659969.

[46] John C Obenauer, Lewis C Cantley, and Michael B Yaffe. "Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs". In: *Nucleic acids research* 31.13 (2003), pp. 3635–3641.

[47] Gavin E Crooks et al. "WebLogo: a sequence logo generator". In: *Genome research* 14.6 (2004), pp. 1188–1190.

[48] Pablo Gainza et al. "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning". In: *Nature Methods* 17.2 (2020), pp. 184–192.

[49] Fabian Fuchs et al. "Se (3)-transformers: 3d roto-translation equivariant attention networks". In: *Advances in neural information processing systems* 33 (2020), pp. 1970–1981.

[50] Bowen Jing et al. "Learning from protein structure with geometric vector perceptrons". In: *arXiv preprint arXiv:2009.01411* (2020).

[51] Klarisa Rikova et al. "Global survey of phosphotyrosine signaling identifies oncogenic kinases in lung cancer". In: *Cell* 131.6 (2007), pp. 1190–1203.

[52] Salomé S Pinho and Celso A Reis. "Glycosylation in cancer: mechanisms and clinical implications". In: *Nature Reviews Cancer* 15.9 (2015), pp. 540–555.

[53] Raquel Dias and Ali Torkamani. "Artificial intelligence in clinical and genomic diagnostics". In: *Genome medicine* 11.1 (2019), p. 70.

[54] Jia Xu et al. “Translating cancer genomics into precision medicine with artificial intelligence: applications, challenges and future perspectives”. In: *Human genetics* 138.2 (2019), pp. 109–124.

[55] Kit-Kay Mak, Yi-Hang Wong, and Mallikarjuna Rao Pichika. “Artificial intelligence in drug discovery and development”. In: *Drug discovery and evaluation: safety and pharmacokinetic assays* (2024), pp. 1461–1498.

[56] Kevin B Johnson et al. “Precision medicine, AI, and the future of personalized health care”. In: *Clinical and translational science* 14.1 (2021), pp. 86–93.

[57] Bhavneet Bhinder et al. “Artificial intelligence in cancer research and precision medicine”. In: *Cancer discovery* 11.4 (2021), pp. 900–915.

[58] Bilal Aslam et al. “Proteomics: technologies and their applications”. In: *Journal of chromatographic science* (2016), pp. 1–15.

[59] Jingcheng Du et al. “Gene2vec: distributed representation of genes based on co-expression”. In: *BMC genomics* 20 (2019), pp. 7–15.

[60] Xiang Yue et al. “Graph embedding on biomedical networks: methods, applications and evaluations”. In: *Bioinformatics* 36.4 (2020), pp. 1241–1251.

[61] Yijia Xiao et al. “Know2BIO: A Comprehensive Dual-View Benchmark for Evolving Biomedical Knowledge Graphs”. In: *arXiv preprint arXiv:2310.03221* (2023).

[62] Fan Yang et al. “scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data”. In: *Nature Machine Intelligence* 4.10 (2022), pp. 852–866.

[63] Yanay Rosen et al. “Universal cell embeddings: A foundation model for cell biology”. In: *bioRxiv* (2023), pp. 2023–11.

[64] Felix Brechtmann et al. “Evaluation of input data modality choices on functional gene embeddings”. In: *NAR Genomics and Bioinformatics* 5.4 (2023), lqad095.

[65] Christina V Theodoris et al. “Transfer learning enables predictions in network biology”. In: *Nature* 618.7965 (2023), pp. 616–624.

[66] Ahmed Elnaggar et al. “Prottrans: Toward understanding the language of life through self-supervised learning”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.10 (2021), pp. 7112–7127.

[67] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[68] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. “struc2vec: Learning node representations from structural identity”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 385–394.

[69] Yiqun Chen and James Zou. “GenePT: A Simple But Effective Foundation Model for Genes and Cells Built From ChatGPT”. In: *bioRxiv* (2024). DOI: 10.1101/2023.10.16.562533. eprint: https://www.biorxiv.org/content/early/2024/03/05/2023.10.16.562533.full.pdf. URL: https://www.biorxiv.org/content/early/2024/03/05/2023.10.16.562533.

[70] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. “Linkbert: Pretraining language models with document links”. In: *arXiv preprint arXiv:2203.15827* (2022).

[71] Michael Ashburner et al. “Gene ontology: tool for the unification of biology”. In: *Nature genetics* 25.1 (2000), pp. 25–29.

[72] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. “Multi-relational Poincaré Graph Embeddings”. In: *Neural Information Processing Systems*. 2019. URL: https://api.semanticscholar.org/CorpusID:168633605.

[73] Antoine Bordes et al. “Translating Embeddings for Modeling Multi-relational Data”. In: *Neural Information Processing Systems*. 2013. URL: https://api.semanticscholar.org/CorpusID:14941970.

[74] Maithra Raghu et al. “SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability”. In: *Neural Information Processing Systems*. 2017. URL: https://api.semanticscholar.org/CorpusID:23890457.

[75] Almagro Armenteros. "DeepLoc: prediction of protein subcellular localization using deep learning". In: *Bioinformatics* 33 (2017), pp. 3387–3395. URL: https://api.semanticscholar.org/CorpusID:3789627.

[76] Kin Yau Wong, Donglin Zeng, and DY Lin. "Robust score tests with missing data in genomics studies". In: *Journal of the American Statistical Association* (2019).

[77] Cosmin Lazar et al. "Accounting for the multiple natures of missing values in label-free quantitative proteomics data sets to compare imputation strategies". In: *Journal of proteome research* 15.4 (2016), pp. 1116–1125.

[78] Jiang Li et al. "Imputation of missing values for electronic health record laboratory data". In: *NPJ digital medicine* 4.1 (2021), p. 147.

[79] Simon B Goldberg, Daniel M Bolt, and Richard J Davidson. "Data missing not at random in mobile health research: Assessment of the problem and a case for sensitivity analyses". In: *Journal of Medical Internet Research* 23.6 (2021), e26749.

[80] Liang Jin et al. "A comparative study of evaluating missing value imputation methods in label-free proteomics". In: *Scientific reports* 11.1 (2021), p. 1760.

[81] Bobbie-Jo M Webb-Robertson et al. "Review, evaluation, and discussion of the challenges of missing value imputation for mass spectrometry-based label-free global proteomics". In: *Journal of proteome research* 14.5 (2015), pp. 1993–2001.

[82] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons, 2019.

[83] Shazia Afzal et al. "Data Readiness Report". In: *2021 IEEE International Conference on Smart Data Services (SMDS)* (2020), pp. 42–51. URL: https://api.semanticscholar.org/CorpusID:222341544.

[84] Yuliya V Karpievitch, Alan R Dabney, and Richard D Smith. "Normalization and missing value imputation for label-free LC-MS analysis". In: *BMC bioinformatics* 13 (2012), pp. 1–9.

[85] Joost R Van Ginkel et al. "Rebutting existing misconceptions about multiple imputation as a method for handling missing data". In: *Journal of personality assessment* 102.3 (2020), pp. 297–308.

[86] Yaoyang Zhang et al. "Protein analysis by shotgun/bottom-up proteomics". In: *Chemical reviews* 113.4 (2013), pp. 2343–2394.

[87] Baljash S Cheema et al. *Protein turnover in the failing heart: an ever-changing landscape*. 2017.

[88] Björn Schwanhäusser et al. "Global quantification of mammalian gene expression control". In: *Nature* 473.7347 (2011), pp. 337–342.

[89] Qian Li et al. "GMSimpute: a generalized two-step Lasso approach to impute missing values in label-free mass spectrum analysis". In: *Bioinformatics* 36.1 (2020), pp. 257–263.

[90] Shisheng Wang et al. "NAguideR: performing and prioritizing missing value imputations for consistent bottom-up proteomic analyses". In: *Nucleic acids research* 48.14 (2020), e83–e83.

[91] Izumi V Hinkson and Joshua E Elias. "The dynamic state of protein turnover: It's about time". In: *Trends in cell biology* 21.5 (2011), pp. 293–303.

[92] Mingyi Liu and Ashok Dongre. "Proper imputation of missing values in proteomics datasets for differential expression analysis". In: *Briefings in Bioinformatics* 22.3 (2021), bbaa112.

[93] Miranda L Gardner and Michael A Freitas. "Multiple imputation approaches applied to the missing value problem in bottom-up proteomics". In: *International journal of molecular sciences* 22.17 (2021), p. 9650.

[94] Jonathon J O'Brien et al. "The effects of nonignorable missing data on label-free mass spectrometry proteomics experiments". In: *The annals of applied statistics* 12.4 (2018), p. 2075.

[95] Nicolai Bjødstrup Palstrøm, Rune Matthiesen, and Hans Christian Beck. "Data imputation in merged isobaric labeling-based relative quantification datasets". In: *Mass Spectrometry Data Analysis in Proteomics* (2020), pp. 297–308.

[96] Minjie Shen et al. "Comparative assessment and novel strategy on methods for imputing proteomics data". In: *Scientific reports* 12.1 (2022), p. 1067.

[97] Xiaoyan Yin et al. "Multiple imputation and analysis for high-dimensional incomplete proteomics data". In: *Statistics in medicine* 35.8 (2016), pp. 1315–1326.

[98] Alison Barbara Ross, Julian David Langer, and Marko Jovanović. "Proteome Turnover in the Spotlight: Approaches, Applications, and Perspectives". In: *Molecular & Cellular Proteomics : MCP* 20 (2020). URL: https://api.semanticscholar.org/CorpusID:231862757.

[99] Edward Lau et al. "A large dataset of protein dynamics in the mammalian heart proteome". In: *Scientific Data* 3 (2016). URL: https://api.semanticscholar.org/CorpusID:18855339.

[100] Roger E.Millsap, Alberto Maydeu-Olivares, and Paul D. Allison. "Missing Data". en. In: *The SAGE Handbook of Quantitative Methods in Psychology*. SAGE Publications Ltd, 2009, pp. 72–90. ISBN: 978-0-85702-099-4. DOI: 10.4135/9780857020994. URL: https://methods.sagepub.com/hnbk/edvol/sage-hdbk-quantitative-methods-in-psychology/chpt/missing-data (visited on 01/31/2025).

[101] Panteha Hayati Rezvan, Katherine J. Lee, and Julie Anne Simpson. "A review of the reporting and implementation of multiple imputation in medical research". In: *missing* (2015). URL: https://api.semanticscholar.org/CorpusID:63044619.

[102] Ding Wang et al. "Characterization of human plasma proteome dynamics using deuterium oxide". In: *PROTEOMICS – Clinical Applications* 8.7-8 (2014), pp. 610–619. DOI: https://doi.org/10.1002/prca.201400038. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prca.201400038. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prca.201400038.

[103] Maggie P.Y. Lam et al. "Protein kinetic signatures of the remodeling heart following isoproterenol stimulation". In: *The Journal of Clinical Investigation* 124.4 (Apr. 2014), pp. 1734–1744. DOI: 10.1172/JCI73787. URL: https://doi.org/10.1172/JCI73787.

[104] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45.3 (2011), pp. 1–67. DOI: 10.18637/jss.v045.i03. URL: https://www.jstatsoft.org/index.php/jss/article/view/v045i03.

[105] Tae-Young Kim et al. "Metabolic Labeling Reveals Proteome Dynamics of Mouse Mitochondria*". In: *Molecular Cellular Proteomics* 11.12 (2012), pp. 1586–1594. ISSN: 1535-9476. DOI: https://doi.org/10.1074/mcp.M112.021162. URL: https://www.sciencedirect.com/science/article/pii/S1535947620334551.

[106] Birgit HM Meldal et al. "Complex Portal 2018: extended content and enhanced visualization tools for macromolecular complexes". In: *Nucleic Acids Research* 47.D1 (Oct. 2018), pp. D550–D558. ISSN: 0305-1048. DOI: 10.1093/nar/gky1001. eprint: https://academic.oup.com/nar/article-pdf/47/D1/D550/27436068/gky1001.pdf. URL: https://doi.org/10.1093/nar/gky1001.

[107] David S Wishart et al. "MarkerDB: an online database of molecular biomarkers". In: *Nucleic Acids Research* 49.D1 (Nov. 2020), pp. D1259–D1267. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa1067. eprint: https://academic.oup.com/nar/article-pdf/49/D1/D1259/35364158/gkaa1067.pdf. URL: https://doi.org/10.1093/nar/gkaa1067.

[108] The UniProt Consortium. "UniProt: the Universal Protein Knowledgebase in 2023". In: *Nucleic Acids Research* 51.D1 (Nov. 2022), pp. D523–D531. ISSN: 0305-1048. DOI: 10.1093/nar/gkac1052. eprint: https://academic.oup.com/nar/article-pdf/51/D1/D523/48441158/gkac1052.pdf. URL: https://doi.org/10.1093/nar/gkac1052.

[109] Miguel Martin-Perez and Judit Villén. "Determinants and Regulation of Protein Turnover in Yeast". In: *Cell Systems* 5.3 (2017), 283–294.e5. ISSN: 2405-4712. DOI: https://doi.org/10.1016/j.cels.2017.08.008. URL: https://www.sciencedirect.com/science/article/pii/S2405471217303411.

[110] Yasuhiro Maejima and Junichi Sadoshima. "SUMOylation". In: *Circulation Research* 115.8 (2014), pp. 686–689. DOI: 10.1161/CIRCRESAHA.114.304989. eprint: https://www.ahajournals.org/doi/pdf/10.1161/CIRCRESAHA.114.304989. URL: https://www.ahajournals.org/doi/abs/10.1161/CIRCRESAHA.114.304989.

[111] Majid Alfadhel et al. "Mitochondrial iron-sulfur cluster biogenesis from molecular understanding to clinical disease". In: *Neurosciences Journal* 22.1 (2017), pp. 4–13. DOI: 10.17712/nsj.2017.1.20160542. eprint: https://nsj.org.sa/content/22/1/4.full.pdf. URL: https://nsj.org.sa/content/22/1/4.

[112] Arti Nile Juhyun Shin and Jae-Wook Oh. "Role of adaptin protein complexes in intracellular trafficking and their impact on diseases". In: *Bioengineered* 12.1 (2021). PMID: 34565296, pp. 8259–8278. DOI: 10.1080/21655979.2021.1982846. eprint: https://doi.org/10.1080/21655979.2021.1982846. URL: https://doi.org/10.1080/21655979.2021.1982846.

[113] Quynh Nguyen, Kenji Rowel Q. Lim, and Toshifumi Yokota. "Current understanding and treatment of cardiac and skeletal muscle pathology in laminin-α2 chain-deficient congenital muscular dystrophy". In: *The Application of Clinical Genetics* 12 (2019), pp. 113–130. URL: https://api.semanticscholar.org/CorpusID:196610430.

[114] Haotian Cui et al. "scGPT: toward building a foundation model for single-cell multi-omics using generative AI". In: *Nature Methods* (2024), pp. 1–11.

[115] Yanay Rosen et al. "Toward universal cell embeddings: integrating single-cell RNA-seq datasets across species with SATURN". In: *Nature Methods* (2024), pp. 1–9.