



HOKKAIDO UNIVERSITY

Title	Accurate and fast predictions of protein structures and properties via protein language models
Author(s)	許, 仕傑
Degree Grantor	北海道大学
Degree Name	博士(環境科学)
Dissertation Number	甲第16231号
Issue Date	2025-03-25
DOI	https://doi.org/10.14943/doctoral.k16231
Doc URL	https://hdl.handle.net/2115/97665
Type	doctoral thesis
File Information	XU_Shijie.pdf



Accurate and fast predictions of protein structures and properties via
protein language models

(タンパク質言語モデルによるタンパク質の構造と特性の正確かつ
迅速な予測)

北海道大学大学院環境科学院

XU Shijie



HOKKAIDO UNIVERSITY

Accurate and fast predictions of
protein structures and properties via
protein language models

A DISSERTATION SUBMITTED

BY

SHIJIE XU

TO

GRADUATE SCHOOL OF ENVIRONMENTAL SCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE DIVISION OF

ENVIRONMENTAL SCIENCE

HOKKAIDO UNIVERSITY

SAPPORO, HOKKAIDO

MARCH 2025

©2025 – SHIJIE XU
ALL RIGHTS RESERVED.

Accurate and fast predictions of protein structures and properties via protein language models

ABSTRACT

The work presented in this thesis focuses on protein modeling using artificial intelligence. The field has become increasingly active in recent years due to the explosion of data produced by modern biological techniques. The huge amount of data undoubtedly stimulates the development of related fields and challenges the traditional methods used to process these data. Recent advances in deep learning have thus been introduced to reduce the gaps, but they require further exploration. The work presented in this thesis adopted supervised learning, unsupervised learning, transfer learning, and Bayesian optimization to understand the structures and properties of proteins, as well as their evolution. The primary aim of this thesis is to explore protein structures and properties through protein language models (PLMs). The specific problems addressed in the thesis are detailed below.

The introductory chapter discusses the foundational concepts of protein science and the importance of protein structure prediction, emphasizing challenges like protein complexity and data scarcity. Deep learning is highlighted as a promising but still evolving solution, with PLMs emerging as transformative tools that overcome traditional models' reliance on large, high-quality datasets. By leveraging both labeled and unlabeled data, PLMs offer a novel approach to protein modeling. The chapter also explores machine learning applications in phylogenetics, linking protein evolution to structure and function, and concludes with an outline of the thesis structure and its contributions.

Chapter 2 introduces an innovative, efficient, and accurate approach to predicting intrinsically disordered proteins (IDPs). IDPs, which lack stable structures, are crucial for understanding protein function and are associated with numerous human diseases. Traditional methods rely on multiple sequence alignments (MSAs) to predict IDPs, but these alignments are not universally available and necessitate exhaustive searches through ever-expanding protein sequence databases. The proposed PLM-based method replaces the laborious MSA reconstruction process with alternative information derived from PLMs. Furthermore, an ensemble learning strategy is employed to enhance prediction accuracy by combining multiple PLMs. A novel secondary structure predictor based on PLM is also introduced, further improving prediction accuracy. The functions of IDPs, including disordered linker regions and disordered protein-binding sites, are also predicted. These approaches not only achieve superior accuracy compared to state-of-the-art techniques but also operate at significantly faster speeds, making them ideal for genome-scale IDP predictions.

Chapter 3 presents a rapid and precise predictor for pK_a values. The pK_a values of ionizable groups in proteins are vital for understanding protein function and structure. However, numerous environmental factors, such as hydrogen bonding, ligand interactions, hydrophobicity, and electrostatics, can

significantly alter pK_a values, complicating accurate prediction. Traditional methods often fail or produce inaccurate pK_a predictions, particularly for shifted pK_a values. This chapter introduces a novel method that uses only the PLM representation of a single residue as input. Two isoelectric point (pI) prediction methods based on bidirectional long short-term (BiLSTM) networks are also proposed to improve pK_a prediction, achieving state-of-the-art accuracy of 0.3393 RMSE (root mean square error). The method supports multiple ionizable groups, including the sidechains of Asp, Glu, His, Lys, Cys, Tyr, and N- and C-termini. Testing on challenging datasets demonstrates the method's superior performance, particularly for deeply buried residues with significantly shifted pK_a values. The proposed method also exhibits rapid prediction speeds, making it suitable for large-scale applications in structure-unknown proteins.

Chapter 4 introduces a novel method for identifying metal-binding sites in proteins. Metalloproteins, which play pivotal roles in cellular functions, have profound implications in fields such as industry, medicine, and environmental science. Existing sequence- or structure-based methods often fall short in terms of precision. This chapter presents a multi-step approach: a sequence model for initial binding residue identification, a probe generation algorithm for candidate site prediction, a structure model for site refinement, and a clustering algorithm to eliminate redundancy. Protein language models are used to quickly and accurately identify metal-binding residues from sequence data, which are then refined in subsequent steps. The proposed method is rigorously evaluated and compared with state-of-the-art techniques, demonstrating superior accuracy. It also achieves rapid prediction speeds, approximately twenty seconds per protein, faster than most existing structure-based methods. Additionally, the method's robustness is demonstrated on AlphaFold2-predicted structures, suggesting its potential for large-scale applications.

Chapter 5 introduces a new partitioning algorithm designed to improve the accuracy of phylogenetic inference through partitioned models. Biological sequence heterogeneity, encompassing proteins and nucleotides, is often poorly modeled. Traditional partitioning methods rely heavily on expert knowledge or heuristic approaches. This chapter introduces a machine learning-based algorithm that optimizes the partitioning process via parametrization and Bayesian optimization. The novel parameterized sorting indices (PSI) proposed by the author guides the partitioning process. The method reconstructs phylogenetic trees with significantly more accurate topologies on several simulated datasets. It also yields better topologies, and higher bootstrapping support on real datasets, aligning more closely with the morphological and geological evidence in the literature. The proposed method also demonstrates acceptable computational efficiency compared to traditional methods.

In conclusion, this thesis demonstrates the power of protein language models in predicting protein structures and properties, achieving superior accuracy and efficiency in tasks such as intrinsically disordered protein prediction, pK_a prediction, and metalloprotein prediction. It also highlights the role of machine learning in phylogenetics for understanding and engineering proteins. The findings underscore the potential of these approaches to advance protein science and inspire further research in related fields.

Acknowledgments

This work was carried out under the supervision of Professor Akira Onoda at the Graduate School of Environmental Science, Hokkaido University. I am indebted to Professor Akira Onoda for his continuous guidance, helpful suggestions, and fruitful discussions through the past three years. I remember the first time I met Professor Akira Onoda online in the summer of 2021, when I suffered from the COVID-19 pandemic and was unable to travel to Japan. Professor Akira Onoda helped me to overcome the difficulties in doing research remotely as a research student. I deeply appreciate for his patience and encouragement throughout my transition of research fields from computer science to bioinformatics. I have greatly inspired by Professor Akira Onoda's vision and enthusiasm for artificial intelligence in the fields of biology and chemistry. I would also like to thank Professor Mehdi Davari Dolatabadi for his fruitful discussions, and suggestions from the perspective of interdisciplinary research. His experience in the interdisciplinary research encourages me to explore the challenging research field.

I would like to specially thank the administrative assistants of Onoda laboratory, Ms. Chiemi Katsuta, Ms. Fukiko Taguchi, and Ms. Chihiro Brennan, for their support and help to my efficient research and smooth life in the Onoda laboratory.

I sincerely thank all the members of the Onoda laboratory for their warm friendship and helpful discussions. Thanks to Yan Zhang for his support when I arrived in Japan. Thanks to Mariko Kojima, Shiyu Wang, MD Sadikur Rahman Shuvo, Enyu Dong, Lisa Connolly, Yuya Maeda, Fengshi Qi, Mengqin Wu, MD Badiruzzaman, MD Islam Molla, Yuri Ainai, Joel Muyomba, Aoi Nagamine, Shogo Sato, Gloria Kamwezi, Chikaze Kashima, Yuya Nagaoka, Kosuke Odani, Akie Terasaki, Hibiki Yamashita, Xiaolei Wang, and Ke Xu, for their help and friendship. Thanks to the alumni of Onoda laboratory, Haoyu Wang, Yu Ikarashi, Yuanxi Jiang, Taisei Kagaya, Junki Noda, and Haozhu Wang, for their help and friendship.

I give thanks to the Hokkaido University DX Doctoral Fellowship for its generous financial support during my study of the doctoral course, and its support for the challenging and interdisciplinary research field.

Thank my family for their continuous support and encouragement throughout my life.

Contents

1	GENERAL INTRODUCTION	I
2	ACCURATE AND FAST PREDICTION OF INTRINSICALLY DISORDERED PROTEINS FROM PROTEIN SEQUENCE VIA MULTIPLE PROTEIN LANGUAGE MODELS AND ENSEMBLE LEARNING	23
2.1	Introduction	23
2.2	Method	27
2.3	Results	33
2.4	Conclusion	46
	References	61
3	HIGH-THROUGHPUT AND ACCURATE pK_a PREDICTION VIA A PROTEIN LANGUAGE MODEL	62
3.1	Introduction	62
3.2	Method	67
3.3	Results and discussion	71
3.4	Conclusions	82
	References	101
4	METAL-BINDING PROTEIN IDENTIFICATION USING PROTEIN LANGUAGE MODEL AND DEEP REPRESENTATIONS	102
4.1	Introduction	102
4.2	Results	106
4.3	Method	109
4.4	Conclusion	114
	References	117
5	IMPROVED EVOLUTIONARY RATES ESTIMATION USING BAYESIAN OPTIMIZATION AND PARAMETERIZED SORTING INDICES	118
5.1	Introduction	118
5.2	Method	121
5.3	Results	127
5.4	Conclusion	132
	References	154

CONCLUSIONS AND FUTURE WORK	155
LIST OF PUBLICATIONS	157

Listing of figures

1.1	Protien and protein structure	2
1.2	Sequence to Structure	3
1.3	Protein structure prediction	3
1.4	Homology modeling and <i>ab initio</i> methods	4
1.5	AlphaFold2	5
1.6	Post-AlphaFold era	6
1.7	Deep learning	7
1.8	Architecture of BiLSTM	8
1.9	Architecture of Transformer	9
1.10	Protein language models	11
1.11	PLMs learn structural information	12
1.12	Prediction of protein structure and properties via PLMs	14
2.1	Intrinsically disordered proteins	24
2.2	Deep learning-based IDP prediction	25
2.3	This work of IDP-ELM	26
2.4	Architecture of the IDP-ELM predictor.	29
2.5	Architecture of the masked language model.	30
2.6	Architecture of the T5 model.	31
2.7	Accuracy of secondary structure predictors based on different PLMs.	35
2.8	Performance of top 10 predictors on nonredundant CAID test data set.	38
2.9	ROC and precision-recall curves on CAID data set.	39
2.10	Performance on nonredundant CAID2 data set.	40
2.11	ROC and precision-recall curves on CAID2 data set.	41
2.12	IDP prediction example 1	42
2.13	IDP prediction example 2	43
2.14	Inference time of IDP-ELM for human proteome prediction.	45
2.15	Speed comparison	46
3.1	Protein pK_a and isoelectric point (pI)	63
3.2	Computational methods for protein pK_a prediction without deep learning	64
3.3	Deep learning-based methods for protein pK_a prediction	65
3.4	This work of protein pK_a prediction	66
3.5	Overview of pKALM.	70
3.6	Performance comparison of different PLMs.	75

3.7	Performance comparison on different residues.	76
3.8	Performance comparison and ablation study.	78
3.9	Exemplary predictions.	80
3.10	The distribution of the pK_a values in the training and test data sets	85
3.11	The distributions of pI data sets.	86
3.12	Validated RMSE of peptide pI prediction	87
3.13	Validated RMSE of protein pI prediction	87
3.14	Performance of different PLMs on different pK_a shifts	88
3.15	Human proteome prediction	89
3.16	Attention maps for AhpC protein (head I-10)	90
3.17	Attention maps for AhpC protein (head II-20)	91
4.1	Metalloproteins	103
4.2	Motif- and Template-based metal binding prediction	104
4.3	Deep learning-based metal binding prediction	105
4.4	This work of metalloprotein prediction	106
4.5	Architecture of the PRIME predictor	107
4.6	First example of PRIME	108
4.7	Second example of PRIME	109
4.8	Relationship between the maximum number of binding residues and the chain length.	111
4.9	Statistics of the distances of the pairs of ions and N/O/S atoms.	112
4.10	Ablation study of structure pre-training	113
5.1	Phylogenetics and partitioned models.	119
5.2	Performance comparison on empirical DNA data sets	134
5.3	Bin distribution generated by PsiPartition.	135
5.4	Transfer distances between different partition methods.	136
5.5	Performance comparison on empirical protein data sets.	137
5.6	BIC and AICc curves of Bayesian optimization.	138
5.7	Tracing of BICs from PsiPartition.	139
5.8	RF distance on simulated DNA data set 1.	139
5.9	RF distances on simulated DNA data set 2.	140
5.10	Reconstruction comparison 1.	140
5.11	Reconstruction comparison 2.	141
5.12	Reconstruction comparison 3.	141
5.13	Bin distribution of DNA sites partitioned by PsiPartition	146
5.14	The reconstructed Noctuidae tree based on PsiPartition.	147
5.15	The reconstructed Noctuidae tree without partitioning.	148

List of Terms and Acronyms

GLOSSARY

Bayesian Optimization is a black-box optimization method used to optimize the parameters of target functions. 9, 17

CNN Convolutional Neural Network, a type of neural network that is well-suited for image and video recognition tasks. 6, 8

Deep Learning is a subfield of machine learning that uses neural networks with many layers to model and make sense of data. 4, 6

LSTM Long short-term memory, a type of recurrent neural network designed to overcome the vanishing gradient problem of traditional RNNs. 7

Protein Language Model is a language model that is pretrained on large-scale unlabeled protein sequences. It captures the statistical properties of protein sequences and can be fine-tuned on downstream tasks. 5, 10–12

ACRONYMS

BiLSTM bidirectional long short-term memory. 8

GPU graphics processing unit. 8

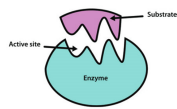
MSA multiple sequence alignment. 5, 14

1

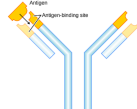
General Introduction

Proteins play an indispensable role in numerous biological processes (Figure 1.1). The function of a protein is primarily determined by its structure, which is pivotal for unraveling the molecular mechanisms of life. While traditional methods such as X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy offer highly precise structural data, they are costly, time-consuming, and labor-intensive. In contrast, computational methods, particularly the emergence of AlphaFold2 [1], can predict protein structures with remarkable accuracy, efficiency, and at a fraction of the cost. Approximately fifty years ago, Anfinsen's dogma [2] (Figure 1.2) proposed that the native structure of a protein is predominantly determined by its amino acid sequence, as evidenced by the ability of ribonuclease to refold spontaneously into its native form after denaturation by urea and β -mercaptoethanol. Although this principle has been challenged in recent years with discoveries such as the chaperone [3], which assists in proper protein folding, the notion that the amino acid sequence contains sufficient information to dictate the protein's structure remains widely accepted. Therefore, the problem of protein structure prediction continues to be a significant and complex challenge in the field.

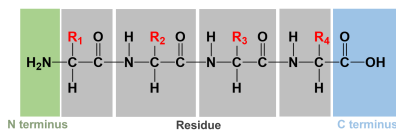
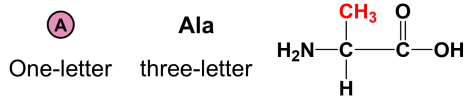
Protein



Enzyme



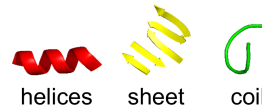
Antibody



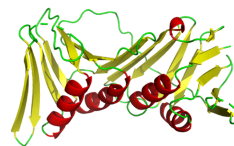
Protein structures



Primary structure



Secondary structure



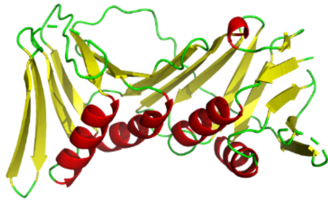
Tertiary structure

Figure 1.1: Twenty amino acids are the building blocks of proteins. Proteins play various functions in the biological systems such as enzymes, antibodies, myosin, and hemoglobin. Protein structures can be defined at different levels: primary, secondary, tertiary. The primary structure is the sequence of amino acids in a protein. The secondary structure refers to the local folding patterns of the protein, such as α -helices and β -sheets. The tertiary structure is the structure of the single protein chain.

Anfinsen's dogma



Protein folds in
msec to secs



Levinthal's paradox

- Protein folds rapidly
- Exhaustive search is impossible

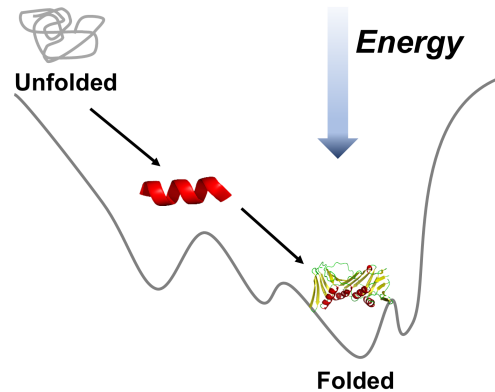
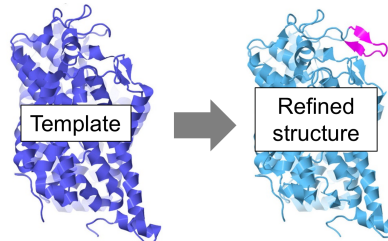
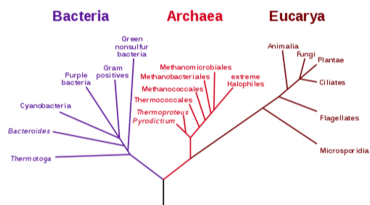


Figure 1.2: Anfinsen's dogma proposed that the native structure of a protein is predominantly determined by its amino acid sequence. The protein folding process is rapid while the possible conformations are vast. It is impossible to search all possibilities exhaustively.

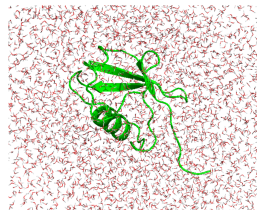
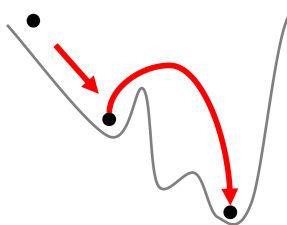
Before deep learning (1970-2010)

• Homology modeling



Zhexin Xiang, *Curr. Protein Pept. Sci.* 7.3 (2006): 217-227.

• Molecular dynamics simulation



Hao Geng *et al.*, *Comput. Struct. Biotechnol. J.* 17 (2019): 1162-1170.

Figure 1.3: Various computational methods have been developed to predict protein structures in the past decades, such as homology modeling and *ab initio* methods.

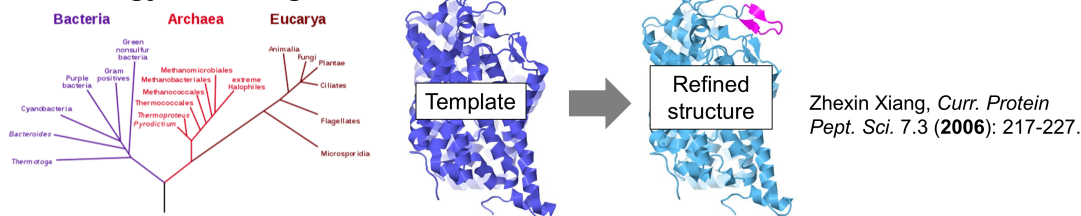
The twenty natural amino acids employed by living organisms to construct proteins lead to an

immense expanse of potential sequences and conformations. Even a relatively small protein, such as one with 100 amino acids, possesses $20^{100} \approx 10^{130}$ possible sequences, a number that far exceeds the total number of atoms in the universe. The conformational space of such a protein could be even more vast, owing to the continuous and flexible nature of protein structures. Currently, only small proteins can be solved by exhaustive searching through all possible conformations on a supercomputer. The protein structure prediction problem is fundamentally a combinatorial optimization problem, which is NP-hard [4] and necessitates the development of efficient algorithms to tackle it.

Over the past decades, various computational approaches have been developed to predict protein structures, encompassing homology modeling and *ab initio* methods (Figure 1.4). Homology modeling, the most prevalent approach, operates on the premise that proteins with similar sequences possess analogous structures. *Ab initio* methods, on the other hand, rely on fundamental physical principles of protein folding, such as the hydrophobic effect and hydrogen bonding. These techniques have profoundly influenced the field of bioinformatics, significantly advancing our ability to predict protein structures. In recent years, machine learning, particularly deep learning, has emerged as a transformative force, enabling highly accurate and efficient predictions. The AlphaFold series [5, 1, 6] has redefined the field, revolutionizing the paradigms of protein science.

Before deep learning (1970-2010)

- **Homology modeling**



- **Molecular dynamics simulation**

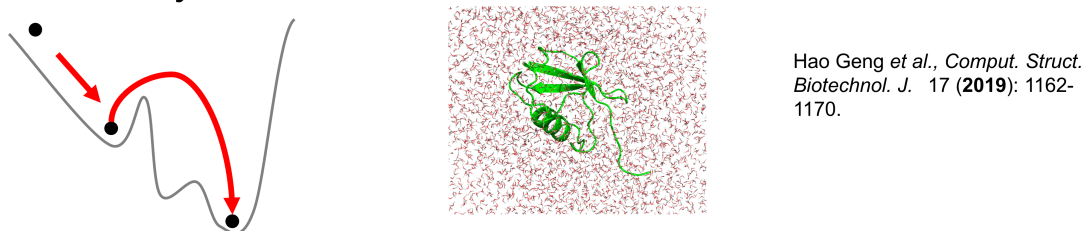


Figure 1.4: Before the development of deep learning-based methods, homology modeling and *ab initio* methods (e.g., Molecular dynamics simulation) are the most widely used methods for protein structure prediction.

The remarkable achievements of the AlphaFold series are underpinned by decades of interdisci-

plinary contributions (Figure 1.5). Foremost, protein researchers have provided an extensive repository of over a hundred thousand protein structures determined through diverse experimental methods, deposited in the Protein Data Bank (PDB) [7]. Additionally, the rapid advancements in sequencing technologies over the past half-century have yielded billions of protein sequences, furnishing high-quality data indispensable for AlphaFold. The advent of deep learning methodologies has facilitated the development of sophisticated AI systems capable of discerning the intricate relationships between protein sequences and their structures. Central to AlphaFold's success is the transformer architecture [8], introduced in 2017, which has found widespread applications in natural language processing (NLP). Furthermore, bioinformatics innovations, particularly advancements in multiple sequence alignment (MSA) techniques, have enabled AlphaFold2 to achieve unparalleled accuracy in protein structure prediction.

Deep learning (2010-present)

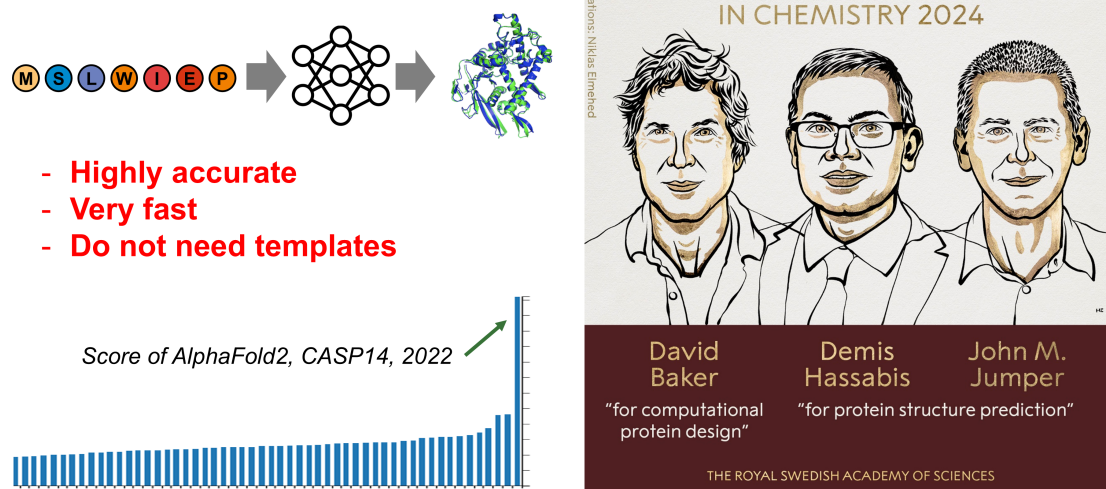


Figure 1.5: Deep learning-based methods such as AlphaFold2 have achieved remarkable success in protein structure prediction with high accuracy and efficiency. The creators of AlphaFold2 were awarded the 2024 Nobel Prize in Chemistry.

The transition from traditional bioinformatics methods to deep learning-based approaches has been revolutionary. The AlphaFold series has been a pivotal milestone in this journey, yet the field remains in its nascent stages (Figure 1.6). Among the pivotal advancements enabled by deep learning is the advent of protein language model (PLMs), which parallel large language models (LLMs) in natural language processing (NLP). PLMs hold the potential to redefine protein bioinformatics by addressing the limitations of traditional methods. Unlike their predecessors, they do not rely on external

databases yet encapsulate knowledge they learned within their parameters. Exploring the predictions and applications of PLM-based approaches in protein science presents a promising frontier for future research. This thesis seeks to investigate the potential of PLMs in protein science by proposing novel methods tailored to tasks where traditional approaches fall short or are not applicable.

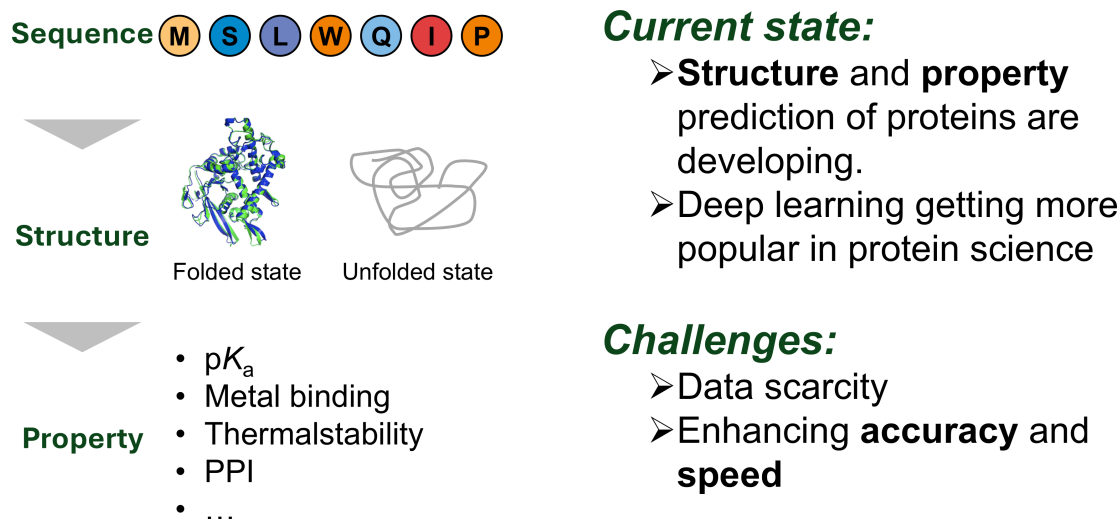
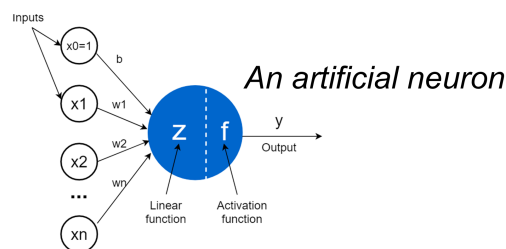
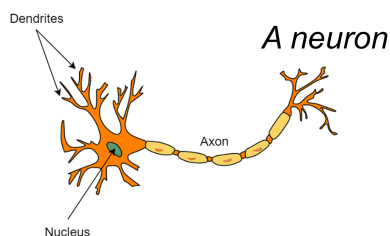


Figure 1.6: Though AlphaFolds have achieved remarkable success in protein structure prediction, there are still many challenges in the post-AlphaFold era, such as property prediction. The challenges in the field could be the enhancement of accuracy and speed, which are limited by the data scarcity.

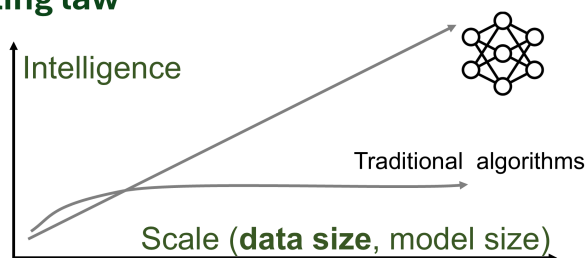
DEEP LEARNING

In recent decades, deep learning has emerged as an indispensable tool for data analysis, with numerous deep learning methodologies being both developed and applied in the realm of protein science. The most widely available protein data are stored in public repositories in the form of sequences and structures. Sequence data are typically processed using multi-layer perceptrons (MLPs) [9], convolutional neural networks (CNNs) [10], recurrent neural networks (RNNs) [11], or Transformers [8]. Spatial information inherent in protein structures is often analyzed using graph neural networks (GNNs) [12] or three-dimensional CNNs. In practice, hybrid models that integrate sequence and structural data by combining various types of neural networks (or their variants) are frequently employed. A prominent example is AlphaFold2 [1], which utilizes a transformer-like architecture, termed Evoformer, as its core component. This innovation enabled the prediction of highly accurate atomic-level protein structures, earning its creators the 2024 Nobel Prize in Chemistry.

Deep neural networks



Scaling law



➤ AlphaFold2 used about **100,000** protein structures to train.

Figure 1.7: Deep learning mimics the human brain's neural networks to process data and extract patterns. Its performance can be scaled up by increasing the size of the systems. However, the data scarcity heavily limits the performance of deep learning models in bioinformatics.

Recurrent Neural Networks (RNNs) are specifically designed to handle sequential data by leveraging the temporal information within input sequences to update hidden states, which are subsequently utilized for downstream tasks such as machine translation and speech recognition. However, the vanishing gradient problem inherent to RNNs limits their capacity to capture long-range dependencies, a critical issue for protein sequences that can extend to hundreds or thousands of amino acids, far surpassing the typical length of human language sentences. To address these challenges, architectures like LSTM (Long Short-Term Memory) networks [13] (Figure 1.8) and gated recurrent unit (GRU) networks [14] were developed.

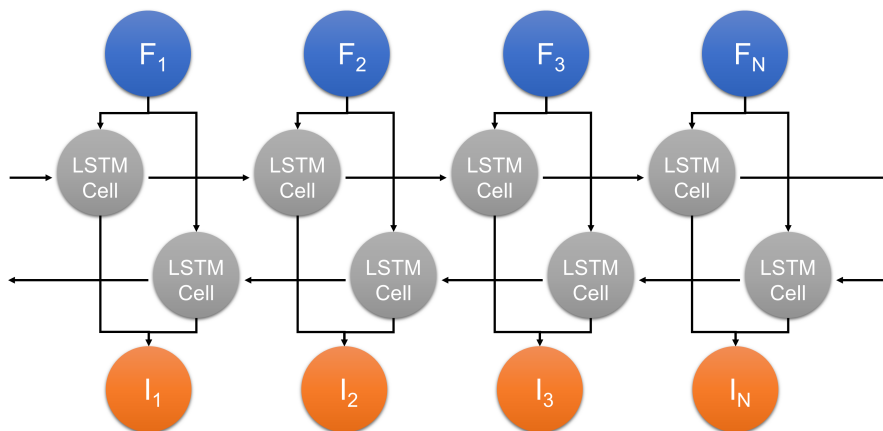


Figure 1.8: The architecture of Bidirectional Long Short-Term Memory (bidirectional long short-term memory (BiLSTM)). BiLSTM is a variant of RNN that can capture the long-range dependencies in both forward and backward directions. Here F_i represents the input features and I_i represents the output information. BiLSTM is widely used in sequence analysis tasks such as protein sequence analysis.

Convolutional Neural Networks (CNNs) also play a pivotal role in bioinformatics. Protein sequences, often treated as one-dimensional signals, are well-suited for analysis by one-dimensional CNNs, which excel at identifying local patterns. Both CNNs and RNNs are adept at capturing sequential information, and numerous hybrid models capitalize on the strengths of both. Conversely, protein structures, regarded as three-dimensional signals, are effectively analyzed by three-dimensional CNNs, which can extract spatial information. For a protein with a 100-Å radius, preprocessing is required to convert three-dimensional coordinates into grids, potentially amounting to one million ($= 100^3$) grid points if the grid resolution is 1 Å. Processing such extensive data can be computationally expensive, even with state-of-the-art graphics processing units (GPUs), underscoring the importance of developing efficient and scalable models.

Transformer models (Figure 1.9), initially conceived for natural language processing (NLP) tasks, have become the foundation for large language models (LLMs) in recent years. Central to their architecture is the self-attention mechanism, which allows dynamic weighting of different positions in input sequences to capture long-range dependencies. This capability distinguishes transformers from RNNs and CNNs, enabling their remarkable scalability and computational efficiency—often referred to as “the scaling law” [15]. This principle underpins the success of systems like ChatGPT, the ground-breaking chat bot launched by OpenAI in 2022.

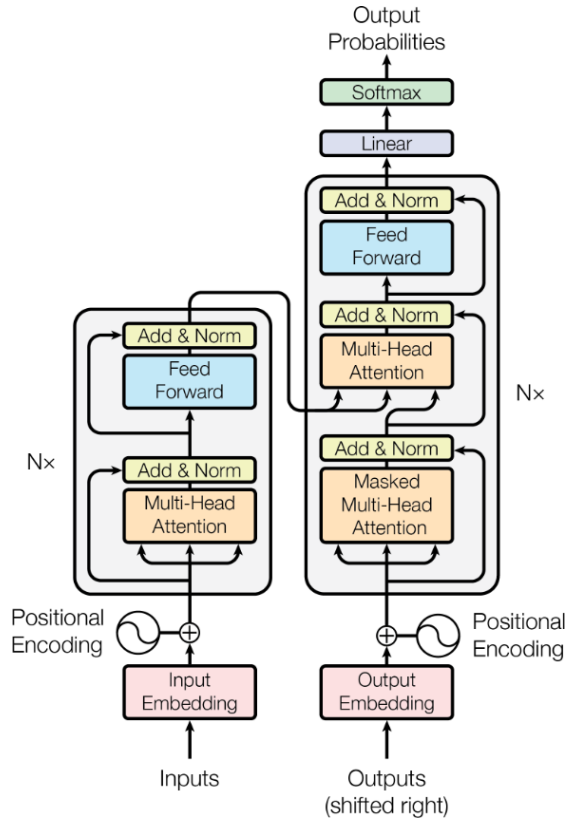


Figure 1.9: The architecture of the Transformer [8] is fundamental to numerous cutting-edge language models (LMs) and has become the prevailing framework for large language models (LLMs). Its self-attention mechanism enables the model to discern long-range dependencies within input sequences, a vital feature for various natural language processing (NLP) tasks.

Bayesian optimization (BO) is a black-box optimization technique designed for optimizing parameters of target functions when their analytical forms are either expensive to evaluate or unavailable. Widely applied across diverse domains, including hyperparameter tuning in machine learning models, BO is particularly effective for scenarios involving fewer than 20 parameters. Its core principle lies in balancing exploration and exploitation within the parameter space by employing a surrogate model to approximate the target function and a learning function to guide the search for optimal parameters. This sequential design strategy renders BO significantly more efficient than grid search or random search, especially under constraints on the number of trials.

PROTEIN LANGUAGE MODELS

Protein language models are the product of representation learning in the protein domain. The concept of “representations”, introduced in the 1980s by the 2024 Nobel Laureate G. Hinton *et al.* [11], initially emerged in the context of natural language processing (NLP). These representations are high-dimensional vectors whose spatial relationships encapsulate the contextual meaning of words. A compelling and intuitive illustration of this idea is as follows

$$\text{Queen} = \text{King} - \text{Man} + \text{Woman}.$$

Here, while discrete words cannot undergo arithmetic operations, their vector representations can. Another noteworthy property of these representations is their semantic clustering: vectors corresponding to similar words lie closer in space, whereas those of dissimilar words are more distant.

In the early 2010s, these representations were primarily derived using traditional machine learning techniques such as Word2Vec [16] and GloVe [17]. However, the advent of deep learning brought forth models like ELMo [18], BERT [19], and GPT [20], which utilized self-supervised learning to capture contextual information more effectively. With hundreds of millions or even billions of parameters, these models produced far more robust representations, particularly in terms of generalization capability. Today, these deep learning architectures, often referred to as large language models (LLMs), dominate NLP owing to their immense parameter scales and the expansive datasets on which they are trained.

The efficacy of such representations is assessed through various downstream tasks. Discriminative LLMs (e.g., BERT) are typically fine-tuned, either partially or fully, for specific applications such as sentiment analysis or named entity recognition. Generative LLMs, like GPT models, can generate text directly based on input prompts. For these models, both fine-tuning and prompt engineering are widely employed to adapt them to downstream tasks. Consequently, the training of LLMs often follows a two-step process: an initial, resource-intensive pre-training phase, followed by a subsequent, cost-effective fine-tuning phase. The term “fine-tuning” refers to the adjustment of models for specific tasks, which could involve altering their parameters (parameter tuning) or leveraging prompts (prompt tuning). This accessibility has contributed significantly to the widespread adoption of LLMs across diverse domains.

This paradigm has been successfully adapted to protein science. Protein language models, the protein science analogs of LLMs, were introduced by utilizing architectures and training strategies akin to those of LLMs (Figure 1.10). Protein language models are pre-trained on large protein sequence databases, enabling them to learn high-dimensional representations that are challenging for humans to

interpret. For example, transformer model attention maps, an intermediate output, have been shown to correspond to protein contact maps [21] (Figure 1.11). A contact map is a binary matrix indicating whether two residues are spatially proximate in a protein structure (e.g., within 8 Å). Though Protein language models do not explicitly learn protein structures, they appear to capture spatial information, such as binding sites [22]. Protein language models have demonstrated performance comparable to—and in some cases surpassing—conventional methods in a variety of applications [23, 21, 24, 25, 26]. Their rapid development continues to drive advancements in key tasks such as protein structure prediction [26] and protein design [27].

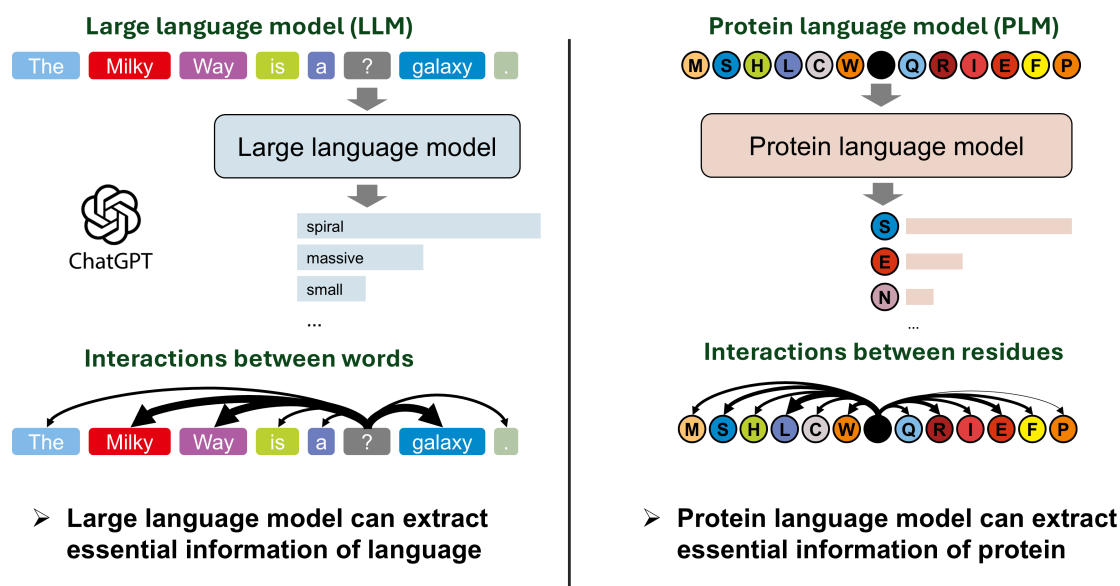


Figure 1.10: Protein language models (PLMs) are the protein science analogs of large language models (LLMs) in natural language processing (NLP). They are pre-trained on large protein sequence databases and learn interactions that are challenging for humans to capture.

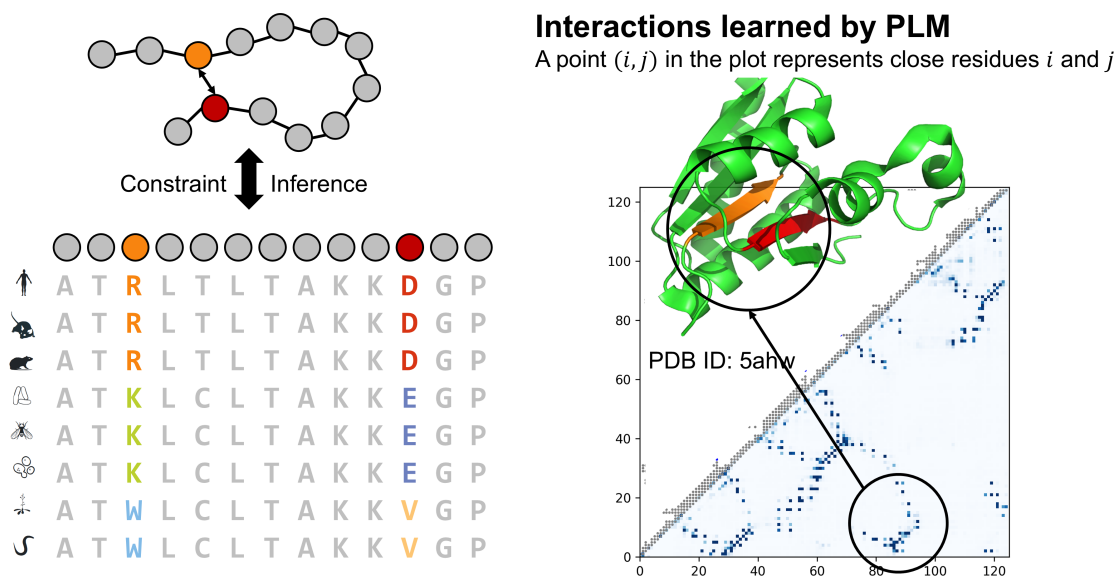


Figure 1.11: Protein language models (PLMs) have been shown to capture spatial information, such as protein contact maps, despite not being explicitly trained on protein structures. This capability underscores the potential of PLMs in protein science.

Analogous to their counterparts in computer science, protein language model can be categorized into two types: generative PLMs and discriminative PLMs. Generative PLMs, including ProGen [28, 29], ProGPT2 [30], RITA [31], and ESM3 [32], primarily focus on generating protein sequences to facilitate de novo protein engineering. Given the astronomical amount of possible protein sequences, generative PLMs significantly narrow the search space at the outset. Discriminative PLMs, such as SeqVec [23], ProtTrans [25], and the ESM series [24, 26, 33], are designed to construct high-dimensional representations of proteins, enabling various downstream tasks, including protein function prediction, protein-protein interaction analysis, and protein structure prediction. Both types of PLMs have demonstrated immense utility in protein science and have seen widespread adoption in recent years. This thesis mainly focuses on discriminative PLMs, as the problems addressed herein are protein structure and property prediction tasks. The details of the aforementioned PLMs are listed in the following Table 1.1.

Table 1.1: The comparison of various protein language models (PLMs).

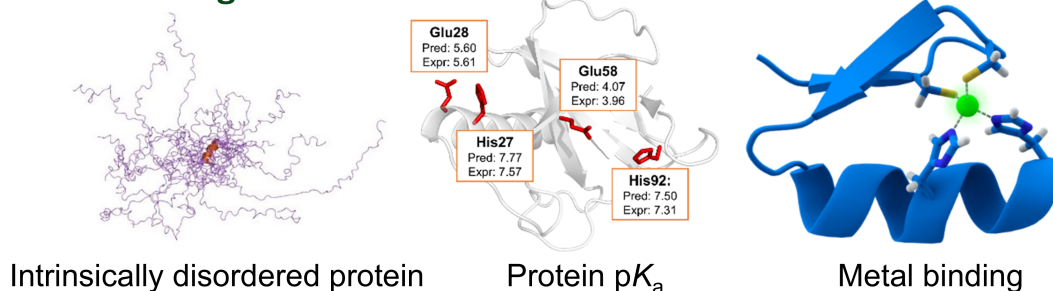
Type	Name	Model size	Dataset	Year	Reference
Generative	ProGen	1.2 B	280 M	2020	[28]
	ProtGPT ₂	738 M	50 M	2022	[30]
	RITA	1.2 B	280 M	2022	[31]
	ProGen2	6.4 B	> 1 B	2023	[29]
	ESM ₃	1.4/7/98 B	3.15 B	2024	[32]
Discriminative	SeqVec	93 M	UR ₅₀	2019	[23]
	ProtTrans	224M-11B	UR ₅₀ /UR ₁₀₀ /BFD	2021	[25]
	ESM	43 M - 670 M	UR ₅₀	2021	[24]
	MSAT	100 M	UR ₅₀ /UR ₁₀₀	2021	[33]
	ESM ₂	8 M-15 B	UR ₅₀	2023	[26]

Note: UR₅₀ refers to the UniRef₅₀ database, UR₁₀₀ refers to the UniRef₁₀₀ database, and BFD refers to the BFD database. MSAT refers to MSA Transformer.

PROTEIN STRUCTURE AND PROPERTY PREDICTION

In the post-AlphaFold era, the landscape of bioinformatics extends beyond predicting three-dimensional protein structures. Emerging challenges include predicting intrinsically disordered proteins (IDPs), protein pK_a values, and metal-binding sites, which are tasks for which AlphaFold programs are not explicitly designed (Figure 1.12). AlphaFold primarily employs high-quality X-ray crystallographic structures as training data, introducing a bias towards well-folded proteins. However, protein dynamics are equally critical, as even highly ordered proteins can exhibit disorder under specific conditions or in particular regions, which may be pivotal to their functions.

Prediction target



Algorithms

IDP-ELM

PLM +
ensemble learning

pKALM

PLM +
transfer learning

PRIME

PLM +
deep representations

Figure 1.12: Though AlphaFolds have achieved remarkable success in protein structure prediction, there are still many challenges in the post-AlphaFold era, such as predicting protein properties. These tasks are not explicitly designed in AlphaFold programs and require novel methods to address. Due to the data scarcity, the performance of current deep learning models is limited.

Another significant limitation of current methodologies lies in their reliance on extensive databases. For instance, AlphaFold2 and many other state-of-the-art techniques require MSA as input, which are derived from large protein databases. In nature, protein folding occurs independently of such databases. However, the performance of AlphaFold2 without MSAs is markedly inferior to its performance with MSAs. This conundrum highlights an incomplete understanding of the protein folding process in current methods. In fact, all MSA-dependent approaches warrant scrutiny, as MSAs are not always accessible, and their quality can be inconsistent.

Intrinsically disordered proteins, lacking stable three-dimensional structures in their native states, are omnipresent in nature, constituting approximately 6% of proteins in the SwissProt database and 30% in eukaryotic [34]. They play vital roles in biological processes, including signal transduction and protein-ligand interactions, and are implicated in human diseases such as Alzheimer's disease. IDPs frequently participate in protein-protein and lipid-binding interactions, making their identification and functional characterization indispensable. IDPs can often be recognized by "linear motifs", which are short sequence patterns derived from databases [35]. However, linear motifs alone are insufficient for identifying IDPs; evidence from protein structures is more reliable. Despite the wealth of structural data in the PDB, annotations for IDPs remain scarce, necessitating meticulous manual curation, as exemplified by the DisProt database [36], which aggregates thousands of IDPs from the literature.

The scarcity of data and the inherent dynamics of intrinsically disordered proteins (IDPs) present formidable challenges to their accurate prediction. Traditional methods, which often depend on multiple sequence alignments (MSAs), perform poorly in this domain due to the difficulty of constructing high-quality MSAs for IDPs. MSAs are generally derived by aligning and comparing amino acid sequences across homologous proteins; however, IDPs are typically less conserved and exhibit greater divergence. Moreover, the efficiency of these predictive approaches remains constrained, further limiting their applicability in genome-scale analyses.

The ionization states of specific chemical groups in proteins, such as the side chains of aspartic acid (Asp) and glutamic acid (Glu), are fundamental to understanding protein function and stability. The acid dissociation constant (pK_a) of these groups and the isoelectric point (pI) of proteins provide insight into their ionization states under varying pH conditions and are closely linked to conformational changes, ligand binding, and enzymatic activity. Deeply buried residues often exhibit pK_a values that deviate significantly from standard solvent values due to intricate interactions within the protein core, such as electrostatics, hydrophobicity, and ligand interactions. Conventionally, pK_a values are measured through nuclear magnetic resonance (NMR) titration experiments, which are laborious, time-consuming, and challenging to scale. To date, the PKAD-2 database [37] contains approximately 1,500 experimentally determined pK_a values, the most comprehensive dataset available.

Computational methods for predicting protein pK_a values have been devised to transcend the constraints of experimental techniques. These approaches are predominantly structure-based, leveraging electrostatic calculations or molecular dynamics simulations. However, they necessitate high-resolution protein structures, which are not always accessible. Furthermore, the accuracy of these methods hinges on the quality of the input structures, which can be influenced by factors such as crystallographic resolution and protein flexibility. Achieving precise and efficient pK_a predictions, reliant on protein structures, remains an imposing challenge in the post-AlphaFold era, especially considering the intrinsic flexibility and dynamic nature of proteins.

Metalloproteins, which bind metal ions, are indispensable for maintaining protein structure and function, with extensive applications in fields such as metalloenzyme design and the development of metal-based therapeutics. Despite the prevalence of metal ions in nature, which make approximately one-third of all proteins metalloproteins, the precise determination of metal-binding sites and specificities remains challenging. Metal-binding sites are often sequestered within the hydrophobic cores of proteins and are highly sensitive to conformational shifts. Even minor mutations can obliterate metal-binding capabilities. The specificity of metal-protein interactions is dictated by many factors such as coordination number, geometry, and ionic charge.

Traditional methods for identifying metal-binding sites rely on sequence or structural informa-

tion, each bearing limitations. Sequence-based approaches fail to predict the intricate coordination geometry of metal-binding sites, while structure-based methods are computationally expensive and necessitate high-resolution protein structures. Moreover, sequence-based methods frequently overlook metal-binding sites, whereas structure-based methods tend to overestimate their number. This discrepancy arises from the inherent complexity of metal-protein interactions, and identifying these sites amidst the vast structural landscape is akin to locating a needle in a haystack, thus demanding innovative and efficient computational strategies.

Molecular phylogenetics offers a powerful lens for reconstructing the evolutionary trajectories of proteins, elucidating the mutations that have shaped their functional landscapes over billions of years. Substitution rates at individual sites are typically modeled using homogeneous or partitioned frameworks. However, the heterogeneous evolutionary constraints that sculpt distinct site-specific dynamics necessitate careful partitioning for accurate phylogenetic reconstruction. Determining the optimal number of partitions remains a persistent challenge, as current methods rely heavily on expert heuristics, often lacking theoretical rigor. With machine learning demonstrating transformative potential in bioinformatics, its application to site partitioning in molecular phylogenetics represents a compelling and uncharted avenue of research.

DISSERTATION OUTLINE

Chapter 2 The author proposed a novel IDP prediction method called IDP-ELM to predict the intrinsically disordered regions (IDRs) as well as their functions, including disordered flexible linkers and disordered protein binding. This method uses high-dimensional representations extracted from several state-of-the-art PLMs and predicts IDRs by ensemble learning based on bidirectional recurrent neural networks. The performance of the method was evaluated on two independent test datasets from CAID (critical assessment of protein intrinsic disorder prediction) [38] and CAID2 [39], and showed significant improvements in terms of area under the receiver operating characteristic (AUC), Matthew’s correlation coefficient (MCC), and F1 score. In addition, IDP-ELM requires only protein sequences as input and does not require the time-consuming process of protein profiling that is a prerequisite for most existing state-of-the-art methods, providing an accurate, fast and convenient tool for proteome-level analysis. The corresponding reproducible source code and model weights are available at <https://github.com/xu-shi-jie/idp-elm>.

Chapter 3 The author proposed pKALM, a novel deep learning-based method for high-throughput protein pK_a prediction. pKALM uses a protein language model (PLM) to capture the complex sequence-structure relationship of proteins. While traditionally considered a structure-based problem, the re-

sults presented in this study show that a PLM pre-trained on large protein sequence databases can effectively learn this relationship and achieve state-of-the-art performance. pKALM accurately predicts the pK_a values of six residues (Asp, Glu, His, Lys, Cys, and Tyr) and two termini with high precision and efficiency. It excels at predicting both exposed and buried residues, which often deviate from standard solvent measured pK_a values. The author demonstrate a novel finding that predicted protein isoelectric points (pI) can be used to improve the accuracy of pK_a prediction. High-throughput pK_a prediction of the human proteome using pKALM achieves a speed of 4,965 pK_a predictions per second, which is several orders of magnitude faster than existing state-of-the-art methods. The case studies illustrate the effectiveness of pKALM in estimating pK_a values and the limitations of the method. pKALM will thus be a valuable tool for researchers in biochemistry, biophysics, and drug design. A web server for pKALM is available at <https://onodalab.ees.hokudai.ac.jp/pkalm>.

Chapter 4 An accurate metal binding site prediction method is proposed. The author leverage a protein language model to accurately and rapidly detect the possible binding residues, and then generate possible binding positions based on the information. A three-dimensional residual network is employed to check and refine possible binding positions. The two-stage method utilizes both sequence and structural information, thus achieving high recall, precision, and lowest RMSE compared to the existing methods. The pre-training strategy was employed to improve the metal binding site prediction, as it has been proven effective in capturing the structure information of protein structures in previous studies, and the ablation study was conducted and agreed with this conclusion. The case study on the metalloprotein dataset shows that the proposed method are able predict the metal binding sites efficiently and accurately. It also works for the AlphaFold2-predicted structures. The proposed method is expected to be a useful tool for researchers in the field of bioinformatics and biophysics. The corresponding reproducible source code and data are available at <http://github.com/xu-shi-jie/PRIME>.

Chapter 5 A novel partitioning approach, called PsiPartition, based on the parameterized sorting indices of sites and Bayesian optimization is proposed. The author applied the method to empirical data sets, and it performs significantly better than existing methods in terms of the Bayesian Information Criterion (BIC) and the corrected Akaike Information Criterion (AICc). The author tested PsiPartition on the simulated data sets with different site heterogeneity, alignment lengths, and number of loci. It is shown that PsiPartition clearly and stably outperforms other methods in terms of the Robinson-Foulds (RF) distance between the true simulated trees and the reconstructed trees, especially on the data with more site heterogeneity. More importantly, the proposed Bayesian optimization-based method provides, for the first time, a new general framework for efficiently determining the optimal number of partitions. The corresponding reproducible source code and data are available

at <http://github.com/xu-shi-jie/PsiPartition>.

References

- [1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.
- [2] Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.
- [3] Billy Tsai, Chiara Rodighiero, Wayne I Lencer, and Tom A Rapoport. Protein disulfide isomerase acts as a redox-dependent chaperone to unfold cholera toxin. *Cell*, 104(6):937–948, 2001.
- [4] Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 597–603, 1998.
- [5] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.
- [6] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, pages 1–3, 2024.
- [7] Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Henry Chao, Li Chen, Paul A Craig, Gregg V Crichlow, Kenneth Dalenberg, Jose M Duarte, et al. Rcsb protein data bank (rcsb.org): delivery of experimentally-determined pdb structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic Acids Res.*, 51(D1):D488–D508, 2023.

- [8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [9] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386, 1958.
- [10] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.*, 36(4):193–202, 1980.
- [11] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [12] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2008.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.
- [14] Kyunghyun Cho. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [16] Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [17] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [18] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

- [20] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [21] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pages 2020–12, 2020.
- [22] Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. Bertology meets biology: Interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.
- [23] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinf.*, 20:1–17, 2019.
- [24] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci.*, 118(15):e2016239118, 2021.
- [25] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghaliya Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10):7112–7127, 2021.
- [26] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
- [27] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.
- [28] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.
- [29] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

- [30] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nat. Commun.*, 13(1):4348, 2022.
- [31] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. Rita: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.
- [32] Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *bioRxiv*, pages 2024–07, 2024.
- [33] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.
- [34] Bin Xue, A Keith Dunker, and Vladimir N Uversky. Orderly order in protein intrinsic disorder distribution: disorder in 3500 proteomes from viruses and the three domains of life. *J. Biomol. Struct. Dyn.*, 30(2):137–149, 2012.
- [35] Manjeet Kumar, Sushama Michael, Jesús Alvarado-Valverde, András Zeke, Tamas Lazar, Juliana Glavina, Eszter Nagy-Kanta, Juan Mac Donagh, Zsofia E Kalman, Stefano Pascarelli, et al. Elm—the eukaryotic linear motif resource—2024 update. *Nucleic Acids Res.*, 52(D1):D442–D455, 2024.
- [36] Federica Quaglia, Bálint Mészáros, Edoardo Salladini, András Hatos, Rita Pancsa, Lucía B Chemes, Mátyás Pajkos, Tamas Lazar, Samuel Peña-Díaz, Jaime Santos, et al. Disprot in 2022: improved quality and accessibility of protein intrinsic disorder annotation. *Nucleic Acids Res.*, 50(D1):D480–D487, 2022.
- [37] Nicolas Ancona, Ananta Bastola, and Emil Alexov. Pkad-2: New entries and expansion of functionalities of the database of experimentally measured pK_a ’s of proteins. *J. Comput. Biophys. Chem.*, 22(05):515–524, 2023.
- [38] Marco Necci, Damiano Piovesan, and Silvio CE Tosatto. Critical assessment of protein intrinsic disorder prediction. *Nat. Methods*, 18(5):472–481, 2021.
- [39] CAID-2 Portal. CAID-2 references. <https://caid.idpcentral.org/challenge#Data>, 2024. [Online; accessed 29-September-2024].

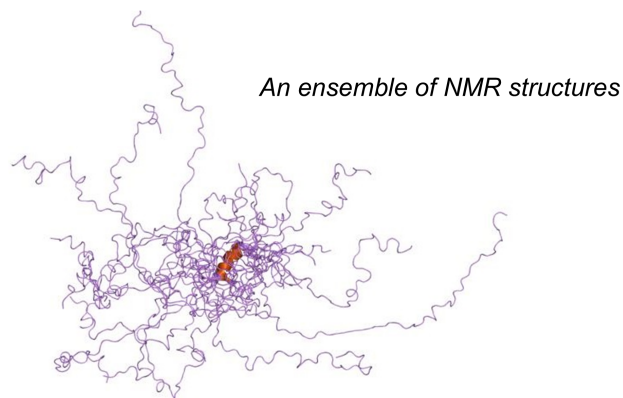
2

Accurate and fast prediction of intrinsically disordered proteins from protein sequence via multiple protein language models and ensemble learning

2.1 INTRODUCTION

Intrinsically disordered proteins (IDPs) and intrinsically disordered regions (IDRs) are characterized by the absence of stable structures yet play pivotal roles in various biological processes [1]. Studies have revealed their widespread presence across all domains of life [2]. For example, more than 6% of amino acid residues in annotated protein sequences are classified as disordered in SwissProt [3], with this figure exceeding 30% in eukaryotes [4] (Figure 2.1). Additionally, IDRs have been implicated in several major human diseases, including Alzheimer's [5], Parkinson's [6], and various cancers [7], making them a focus of interest for drug design [8]. Despite their significance, the identification and annotation of IDPs remain challenging, with only thousands currently cataloged in the DisProt database (V9.3, December 2022), primarily due to the high cost and labor-intensive nature of the process [9].

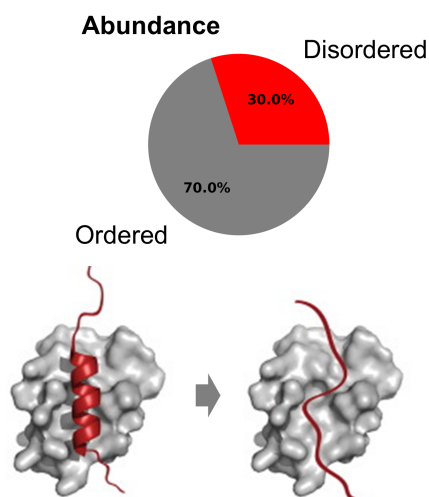
Intrinsically disordered protein (IDP)



IDPs as linear motifs

- 1ENHSALLRRLQDVANDRGSH
- 2ENDDVLLKRLQDVVSERANH
- 3GESKFQQKLAEMTSTRTRMQ

IDPs are very important



IDP occurs in protein-protein interaction

Figure 2.1: Intrinsically disordered proteins are characterized by the absence of stable structures yet play pivotal roles in various biological processes. They take a large portion of the human proteome and play important roles in various biological processes such as protein-protein interactions.

Over the past few decades, various computational approaches have been developed to predict intrinsically disordered regions (IDRs) directly from protein sequences, offering rapid and efficient tools for protein analysis. Deep-learning-based methods employing neural networks [10, 11, 12, 13] have shown significant utility in this domain (Figure 2.2). These methods leverage diverse architectures, including recurrent neural networks (RNNs) [14], convolutional neural networks (CNNs) [15], and self-attention mechanisms [16]. To enhance predictive accuracy, many existing approaches rely directly or indirectly on protein profiles generated via multiple sequence alignments (MSAs). MSAs are created by searching for homologous sequences across comprehensive protein sequence databases.

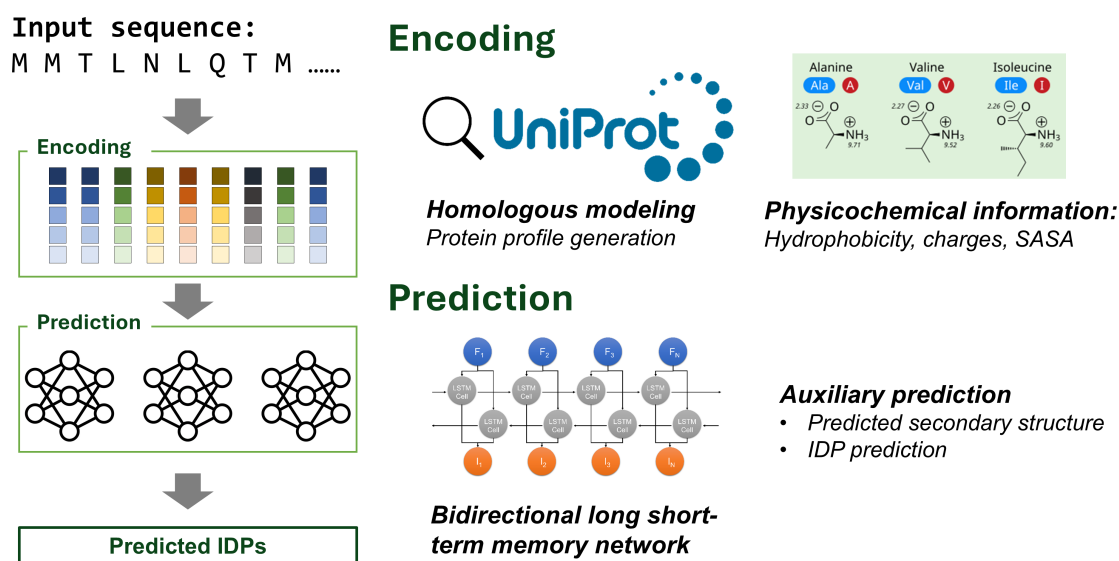


Figure 2.2: Deep learning-based methods have shown significant utility in predicting intrinsically disordered regions (IDRs) directly from protein sequences. These methods leverage various encoding methods and architectures, including bidirectional long short-term memory (BiLSTM) networks and auxiliary tasks like secondary structure prediction.

However, the use of MSAs introduces two key problems. First, MSAs are unavailable for certain proteins, such as those in the dark proteome [17, 18]. Second, constructing MSAs is computationally intensive, requiring exhaustive searches in databases like UniProtKB, which now contains over 227 million protein sequences [19]. As modern sequencing technologies continue to expand these databases rapidly, generating MSAs in a reasonable time frame has become increasingly difficult. These limitations restrict the broader application of IDR predictors. To address this, several MSA-free predictors have been proposed [20, 21], enabling the prediction of IDRs in large-scale analyses, such as the identification of IDRs across the human proteome, using only protein sequence data as input.

Recently, language models (LMs) [22] have gained prominence as powerful tools for sequence modeling in machine learning. These large-scale neural networks, often comprising hundreds of millions or even billions of parameters, are pretrained on vast unlabeled text corpora, such as Wikipedia, to automatically uncover latent sequence patterns. The outputs of these pretrained LMs are expressed as high-dimensional vectors, encapsulating rich contextual information about sequences. These “high-dimensional representations” position similar words closer in vector space, while dissimilar words are mapped farther apart. Pretrained model-based methods have demonstrated superior performance over traditional approaches in diverse machine-learning tasks, such as machine translation and text generation [22, 23]. Protein language models (PLMs) [24, 25, 26], a specialized form of LMs trained on protein sequences, efficiently and effectively extract essential sequence information. PLMs have

achieved competitive results in various protein-related tasks, including protein family classification [27], secondary structure prediction [26], residue-residue contact prediction [24], and three-dimensional structure prediction [25]. Early attempts to apply PLMs to intrinsically disordered protein (IDP) prediction tasks fell short of state-of-the-art performance, primarily due to the limited model size (typically measured by the number of parameters) and the constraints of the underlying architecture [28].

We present IDP-ELM, a novel method for predicting intrinsically disordered proteins (IDPs) and their functions, leveraging high-dimensional representations extracted from pretrained language models (PLMs). This approach utilizes cutting-edge PLMs established in previous studies [24, 25, 26]. Initially, we developed a secondary structure prediction module combining a single PLM with a bidirectional long short-term memory (BiLSTM) network, as incorporating predicted secondary structures has been shown to enhance IDP prediction performance [13]. Subsequently, IDPs are predicted using a bidirectional gated recurrent unit (BiGRU) network, with inputs comprising both high-dimensional representations and the predicted secondary structures. To further enhance performance, we integrated ensemble learning by averaging outputs from predictors based on different PLMs, a strategy proven effective for IDP prediction [29]. Evaluation results demonstrate that IDP-ELM surpasses existing predictors in AUC, MCC, and F1 metrics on challenging independent datasets, CAID [30] and CAID2 [31].

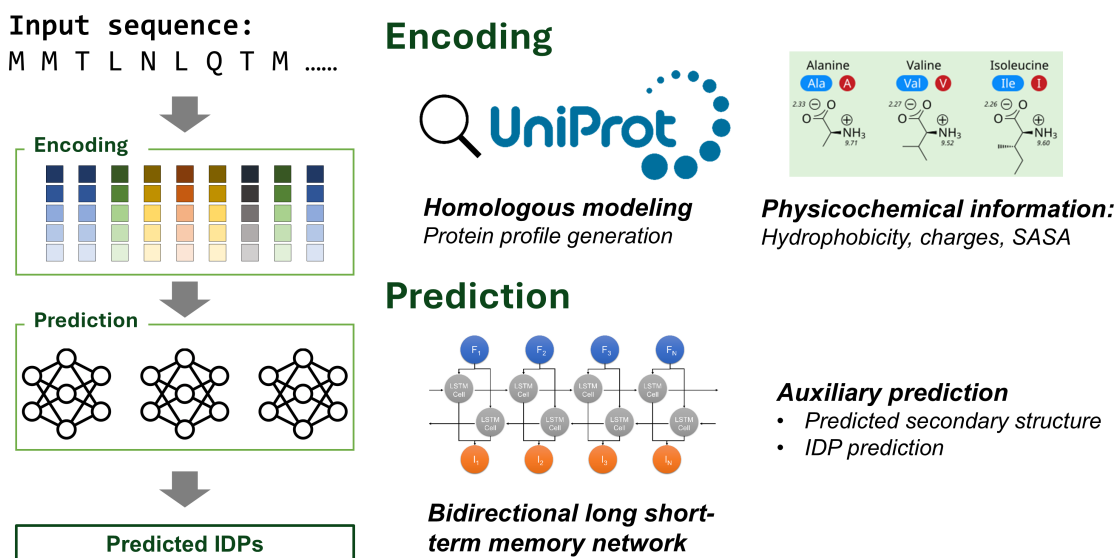


Figure 2.3: This work presents IDP-ELM, a novel method for accurate and fast IDP predictions leveraging high-dimensional representations extracted from pretrained language models (PLMs). IDP-ELM can also predict secondary structures and functions of IDPs.

In addition to achieving high accuracy, we highlight two additional advantages of PLM-based methods. Our innovative approach, which leverages PLM and predicted IDRs, identifies the functions of IDRs with state-of-the-art performance, surpassing existing methods. These functions include disordered protein binding (DP), where disordered regions interact with other proteins, and disordered flexible linkers (DFL), which connect two domains. Furthermore, PLM-based methods greatly improve the computational efficiency of IDP predictions. Leveraging these advantages, we conducted an accurate and efficient prediction of human proteomes, analyzing 34,877 sequences, to demonstrate the practical utility of IDP-ELM in large-scale proteome analyses.

2.2 METHOD

2.2.1 DATA SETS

For secondary structure prediction, we used the NetSurfP-3.0 data set from the reported work [32]. The training data set contains 10,337 nonredundant protein sequences, while the validation data set has 500 such sequences. The performance of the secondary structure predictors was evaluated using three test data sets: TS115, comprising 115 proteins; CB513, with 513 protein chains from 434 proteins; and CASP12, comprising 21 proteins.

The same data sets in the reported work in fIDPnn [13] were used for training, validating, and testing our IDP predictors. fIDPnn was constructed from the DisProt (V7.0) database and separated into three data sets: training (445 proteins), validation (100 proteins), and testing (200 proteins). The combined training and test data sets were then clustered by cd-hit algorithm [33] with 25% sequence similarity. Sequences in the test data set that shared <25% similarity to the training proteins were removed, resulting in 176 test proteins. We also employed the independent test data set from CAID (critical assessment of protein intrinsic disorder prediction) [30]. Furthermore, the sequences in CAID were clustered by cd-hit with 25% sequence similarity against fIDPnn training and validation data sets to ensure fairness in comparison with other predictors. The similarity within itself is also removed, resulting in 81 nonredundant sequences, with 9,008 (15.35%) disordered residues and 49,663 (84.65%) ordered residues. It is referred as “nonredundant CAID”, while the original one is referred to as “redundant CAID” later.

Furthermore, we employed another test data set from CAID2 [31]. Similarly, we removed the redundant sequences against the training and validation data sets of fIDPnn, which is referred to as “nonredundant CAID2”, while the original one is referred to as “redundant CAID2”. It is also noteworthy that the nonredundant CAID2 benchmark test data set was considerably different from the nonredundant CAID benchmark test data set, featuring a vast number of excluded residues (27,650;

60.03%) and disordered residues (4,825; 10.48%), as well as ordered residues (13,579; 29.48%) in 39 nonredundant sequences. In particular, these characteristics are quite different from fIDPnn training and validation data sets, thus being a greater challenge to predict.

For disordered flexible linker (DFL) prediction, we utilized benchmark data sets from the previously reported work [34] to train and evaluate our models, which include TR166 for training and validation, and TE82 for independent testing. To prevent the risk of overfitting, any two sequences in the training and validation data set shared less than 25% sequence similarities. TR166 was divided into training and validation data sets which contain 133 and 33 sequences, respectively. The independent test data set TE82 comprised 82 sequences collected from the DisProt (V8.0) database, as reported by the previous work [34]. The similarity between any two sequences in TR166 and TE82 was less than 25%. Detailed statistics are given in Table 2.1.

Table 2.1: Statistics of data sets for disordered flexible linker prediction

	Data set	# proteins	# DFL	# non-DFL	# residues
TR144	Training	133	2,762 (6.1%)	13,900 (30.5%)	45,528
	Validation	33	908 (6.5%)	3,652 (26.0%)	14,031
	TE82	82	2,223 (6.0%)	11,471 (31.1%)	36,938

For protein-binding prediction, the benchmark data sets from the reported work [35] were utilized to train and evaluate the models. These data sets were constructed from the DisProt database (V8.1), which includes IDRs that interact with other proteins. To avoid any overfitting, sequences within each data set share less than 30% similarity with sequences in other data sets. The detailed statistics can be found in the literature [35].

2.2.2 ARCHITECTURE OF IDP-ELM

Figure 2.4 illustrates the architecture of the IDP-ELM predictor. We employ nine distinct protein language models (PLMs) from the reported works [24, 25, 26] to extract high-dimensional representations for protein sequences. For each PLM, the representations are first fed into a bidirectional long short-term memory (BiLSTM) network, which is followed by a fully connected network and a softmax layer that predicts the secondary structure. The intermediate outputs from a fully connected network, namely, logits, are then concatenated with high-dimensional representations and fed into a bidirectional gated unit (BiGRU) network followed by a fully connected network to produce logits for IDRs. Similarly, the logits for IDR are also concatenated with high-dimensional representations

and fed into another BiGRU network, followed by a fully connected network to produce the logits of the function of IDRs. For secondary structure prediction, we employed the logits from the best-performing PLM (see Results section for details) and converted them into predicted secondary structures by a softmax layer.

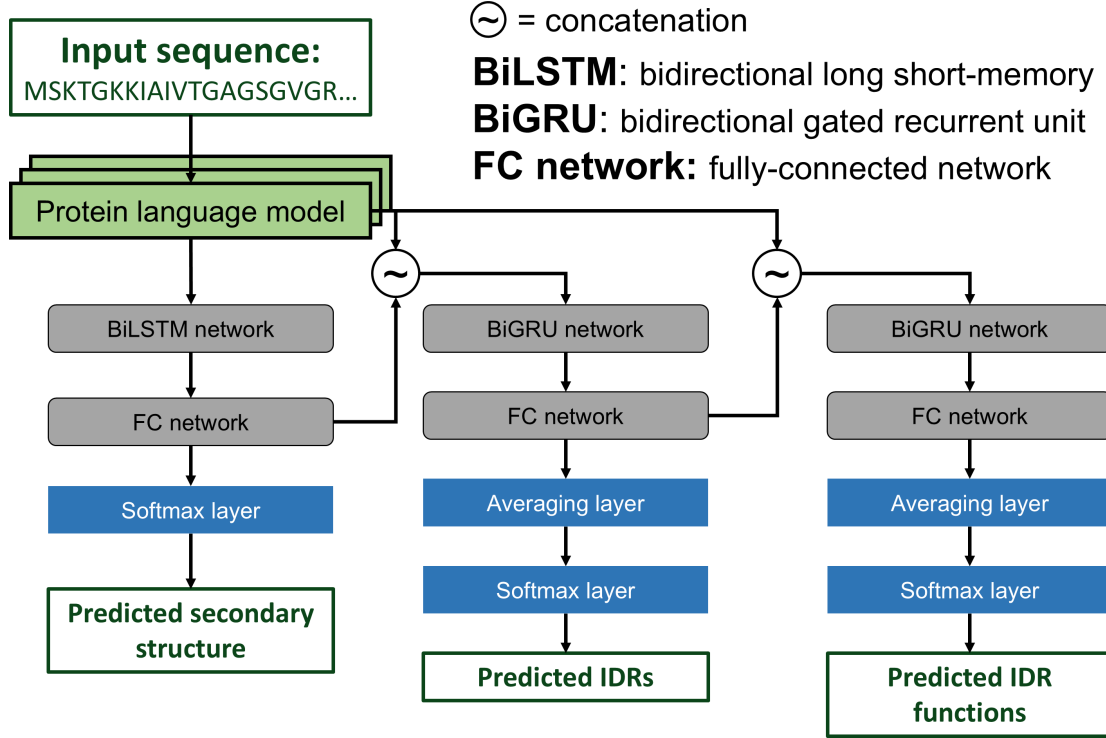


Figure 2.4: Architecture of the IDP-ELM predictor.

2.2.3 ENSEMBLE LEARNING

To predict IDRs as well as their functions, we utilize averaging layers and softmax layers. These layers have no parameters to train, and an averaging layer outputs the mean of the logits $x^i = (x_1^i, x_2^i)$ for $i = 1, 2, \dots, 9$ from each individual predictor, represented by $x = \frac{1}{9} \sum_{i=1}^9 x^i$. A softmax layer is then applied to convert the logits $x = (x_1, x_2)$ to a 2-dimensional vector (p_1, p_2) , where p_1 is the probability of non-IDRs and p_2 is the probability of IDRs. This is achieved by the following equation:

$$(p_1, p_2) = \text{softmax}(x_1, x_2) = \frac{(e^{x_1}, e^{x_2})}{e^{x_1} + e^{x_2}}$$

2.2.4 PROTEIN LANGUAGE MODELS

We utilized various PLMs to extract high-dimensional representations for protein sequences, with different architectures, model sizes, and pretraining data sets. ESM (evolutionary scale modeling) models from reported works [24, 25] are based on masked language models (MLM) [22], and their architecture is depicted in Figure 2.5. For each protein sequence x in the database X , a certain portion of 15% of its amino acid residues is masked, replaced, or kept, while the remaining 85% are unmasked. The MLM predicts the probability of x_i given masked sequences $x_{/M}$, training to recover the original sequence before masking as accurately as possible. This is done to minimize the following loss function [24]:

$$L_{MLM} = E_{x \sim X} E_M \sum_{i \in M} -\log p(x_i | x_{/M})$$

where p is the probability predicted by MLM and E represents the expectation.

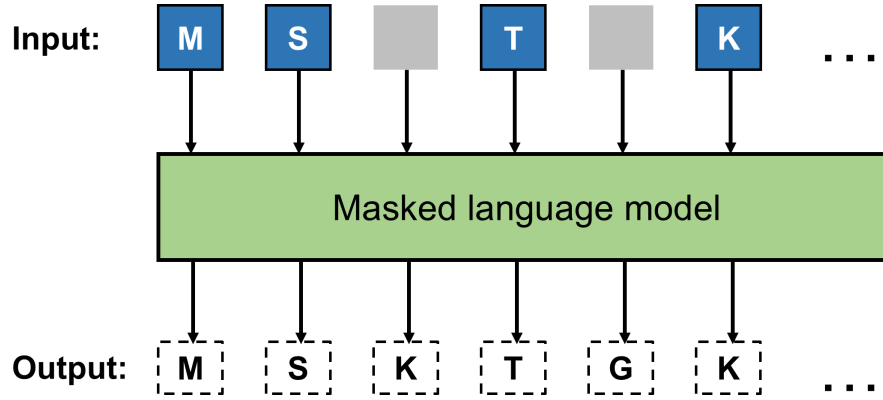


Figure 2.5: Architecture of the masked language model.

Through training on large-scale protein sequence data sets, the ESM models learn the latent relations between residues, which are called *attention maps* in MLM. The attention maps have been demonstrated to be related to the contact map of protein structures [36]. Additionally, ESM models produce high-dimensional representations for each residue in the sequences, which can be used for downstream tasks.

The T₅ (text-to-text transfer transformer) models were initially introduced in the reported work [37] to confirm the boundaries of transfer learning and attaining state-of-the-art proficiency in a variety of natural language processing tasks. These models have an encoder-decoder architecture, wherein the encoder is capable of learning from both directions of the sequence while the decoder is limited to

predicting tokens only from left to right (Figure 2.6). Typically, only the encoder is employed in practice, and T5 models for protein modeling [26], has been successfully utilized into various downstream tasks concerning protein modeling, exhibiting outstanding performance.

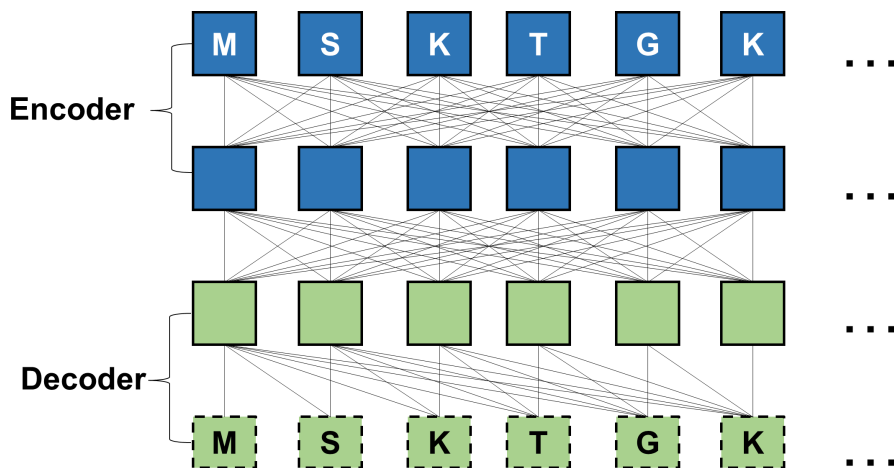


Figure 2.6: Architecture of the T5 model.

The details of PLMs we employed are listed in Table 2.2, including hidden dimensions (Dim), number of layers (No. of layers), number of parameters (Size), and pretraining data sets (Data set). For ESM models, UR50/S refers to the sparse UniRef50 data sets, and UR50/D refers to the dense UniRef50 data sets, which are constructed based on the UniParc data set containing 250 million diverse protein sequences [38]. For T5 models, BFD refers to the Big Fantastic Database [39, 40] which contains 2.1 billion protein sequences, and UR50 refers to UniRef50 data set which contains 45 million protein sequences [41]. For more information on these pretraining data sets, see the original papers [24, 25, 26].

Table 2.2: Details of Different Protein Language Models

Models	Dim	# layers	Size	Data set
esm1b_t33_650M_UR50S	1280	33	650M	UR50/S
esm2_t6_8M_UR50D	320	6	8M	UR50/D
esm2_t12_35M_UR50D	480	12	35M	UR50/D
esm2_t30_150M_UR50D	640	30	150M	UR50/D
esm2_t33_650M_UR50D	1280	33	650M	UR50/D
esm2_t36_3B_UR50D	2560	36	3B	UR50/D
esm2_t48_15B_UR50D	5120	48	15B	UR50/D
prot_t5_xl_bfd	1024	24	~3B	BFD
prot_t5_xl_uniref50	1024	24	~3B	UR50

2.2.5 EVALUATION METRICS

To comprehensively assess the efficacy of our approach, we employed the following metrics:

$$\left\{ \begin{array}{l}
 \text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \\
 \text{SP} = \frac{\text{TN}}{\text{TN} + \text{FP}} \\
 \text{SN(i.e., Rec)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \\
 \text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}} \\
 \text{Fi} = \frac{2 \cdot \text{Pre} \cdot \text{Rec}}{\text{Pre} + \text{Rec}} \\
 \text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \\
 \text{AUC} = \text{area under the receiver operating characteristic curve}
 \end{array} \right.$$

TP, FP, FN, and TN represent the true positive, false positive, false negative, and true negative, respectively. Here, the term “true” signifies the prediction that aligns with the actual outcomes, while “positive” denotes disordered states and vice versa. The Matthew’s correlation coefficient (MCC) ranges between -1 and 1 . A value of -1 indicates complete disagreement between the predictor and the actual results, while a value of 1 signifies a perfect predictor. A value of 0 corresponds to the performance of a random prediction. The Fi score lies between 0 and 1 . A value of 0 signifies complete disagreement between the predictor and the actual results, while a value of 1 represents a perfect prediction. A value of 0.5 corresponds to the performance of a random prediction. Other measures also fall within the

range of 0 to 1, with larger values indicating superior performance.

It is important to note that the AUC measure differs from the others as it remains unbiased in the face of imbalanced classification and does not depend on the choice of threshold. For the remaining metrics, including precision (Pre), recall (Rec), accuracy (ACC), and F1 score, we selected a threshold of 0.5 and do not explicitly mention it unless necessary.

2.3 RESULTS

2.3.1 HYPERPARAMETER TUNING, EARLY STOPPING, AND OPTIMIZERS

We have employed a grid search approach to select the best hyper-parameters for IDP and function predictions, including batch size, hidden size, and number of layers, while keeping the remaining hyper-parameters fixed. For IDP prediction, tuning ranges in [16,32,64] for batch size, [64,128,256] for hidden size, and [2,3] for the number of layers; for DFL prediction, tuning ranges in [8,16,32] for batch size, [16,32,64] for hidden size, and the number of layers is fixed with 1; for DP prediction, tuning ranges in [8,16,32] for batch size, [16,32,64,128] for hidden size, and the number of layers is fixed with 1. By performing the grid search, we chose the checkpoints that exhibit the maximum summation of validated AUC, F1, and MCC scores, resulting in the hyper-parameters in Table 2.7 for IDP prediction, Table 2.8 for DFL prediction, and Table 2.9 for DP prediction. The hidden size and batch size for secondary structure prediction is fixed as 1024 and 16. Note that hidden sizes refer to the hidden size of BiLSTM or BiGRU networks.

To avoid the risk of overfitting, we adopted an early stopping strategy. The validated AUC scores serve as the criterion for halting the training process when they stop increasing over five epochs. The AdamW optimizer [42] has been employed to train the secondary structure predictor, and Adam optimizer [43] has been employed for other training tasks. A learning rate of 10^{-3} and weight decays of 10^{-2} have been set for all computational experiments. Regarding the other optimizer options, we have utilized the default parameters as provided by the PyTorch library. For other detailed hyper-parameters, please refer to Table 2.10.

2.3.2 SEQUENCE PADDING AND SEGMENTATION

While the PLMs can effectively extract high-dimensional representations for a batch of sequences, we instead encoded protein sequences one by one, because user inputs for prediction are more likely to be sequential rather than concurrent. However, once the representations are extracted, they must be padded with zeroes within a mini-batch before being fed into the BiGRU and subsequent networks.

The previous report suggested the influence of sequence padding in protein prediction tasks [44]. We also observed a similar phenomenon in our situation, wherein accuracies slightly decrease from the batchified prediction to the single-sequence prediction. Some of the PLMs employed in our study impose limitations on the maximum sequences length. For instance, the esm1b model can only accept sequences with fewer than 1,024 amino acid residues, while accounting for the inclusion of two special tokens that are the beginning and end of the sequence. Therefore, we divide protein sequences longer than 960 amino acid residues into multiple segments, with an additional 31 amino acid residues neighboring their left and right sides. Consequently, neighboring segments overlap by 31 amino acids, while the first and last segments exhibit a single overlapping region on one side.

2.3.3 DOWN-SAMPLING STRATEGY

Due to the highly imbalanced nature of the training data sets, where the number of positive labels is significantly smaller than the number of negative labels, we have implemented a down-sampling strategy. This strategy is designed to foster the model’s focus on positive labels and prevent it from excessively learning from negative labels. Specifically, each mini-batch training was performed by randomly masking a certain portion of residues that are labeled as negative in the training data sets. For example, only 30% of the ordered residues is utilized for training the IDP predictor. It is worth noting that the down-sampling strategy employed in IDP-ELM has greatly enhanced its performance in terms of F1 score while keeping comparable AUC and MCC scores, as well as stabilizing the performance of a single predictor, and our ablation experiments support the conclusions (Table 2.11).

2.3.4 PLM-BASED SECONDARY STRUCTURE PREDICTIONS ACHIEVE STATE-OF-THE-ART PERFORMANCE

We implemented and evaluated the secondary structure predictors (SSPs) based on different PLMs, as depicted in Figure 2.7. Each SSP corresponds to a specific underlying PLM, and the evaluation encompasses the tasks of predicting Q8 and Q3 states. Q8 entails predicting eight distinct states of secondary structure, 3_{10} -helix (G), α -helix (H), π -helix (I), β -strand I, bright (B), turn (T), bend (S) and others (C) [45]. On the other hand, Q3 involves predicting three general states: helix (including G, H, and I), strand (including B and E) and coil (including T, S, and C). In our experiments, we solely focus on predicting Q8 states, while the Q3 states are derived from Q8 through the aforementioned replacements.

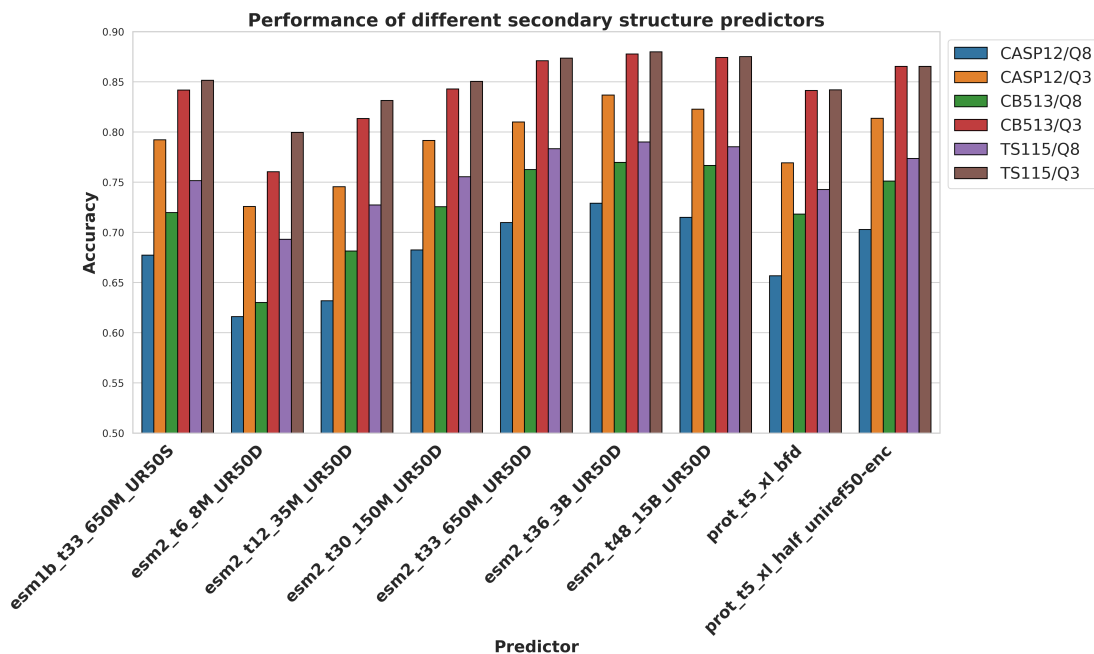


Figure 2.7: Accuracy of secondary structure predictors based on different PLMs.

We train multiple SSPs based on diverse PLMs and evaluate their performance on distinct test data sets. The results are displayed in Figure 2.7. Notably, the SSP utilizing a 3-billion-parameter PLM (esm2_t36_3B_UR50D) achieved the highest scores across all metrics, including Q8 and Q3, on CASP12, CB513, and TS115 test data sets. Consequently, we consider this model as our reported predictor. The Q3 scores on the CB513 and TS115 data sets reached 87.78% and 87.99%, respectively, which approach the theoretical limits of secondary structure prediction (88%–90%) [46]. Additionally, the Q3 score on the CASP12 test data set reached 83.66%, albeit this data set is small, containing only 21 sequences, and thus exhibiting a higher variance.

Furthermore, the performance of PLMs depends on factors such as model size, pretrained data set, and architecture. For instance, ESM2 PLMs (with prefix “esm2”) pretrained on the same UR50/D data set and MLM architecture may yield different performance due to the variations in their model sizes. It is worth noting that larger PLMs do not necessarily guarantee improved accuracy. In certain cases, an SSP based on a 15-billion-parameter ESM model (esm2_t48_15B_UR50D) may exhibit slightly lower accuracy than an SSP based on a 3-billion-parameter ESM model (esm2_t36_3B_UR50D) due to the limited size of the labeled training data set, which is not sufficient to fully leverage the capabilities of the larger PLM. Moreover, two SSPs based on PLMs with similar model architectures and parameter sizes (esm1b_t33_650M_UR50S and esm2_t33_650M_UR50D) display different accuracy levels,

attributed to the larger pretraining data set and longer pretraining duration of the latter. Additionally, two SSPs based on T5 models (prot_t5_xl_bfd and prot_t5_xl_half_uniref50-enc) underperformed compared to the SSP based on the three-billion-parameter ESM model (esm2_t36_3B_UR50D), despite having a similar model size of approximately 3 billion. This discrepancy arises because the T5 models, based on an encoder-decoder architecture, utilize only encoder components. In contrast, the 3-billion-parameter ESM model can harness the full parameters of the trained models.

Finally, we conducted a comparison with the existing secondary structure prediction methods. As shown in Table 2.3, our predictor based on the 3-billion-parameter ESM PLM achieves state-of-the-art results, surpassing both the traditional MSA-based method NetSurfP-2.0 and another predictor NetSurfP-3.0, which is also based on the PLM (esm1b_t33_650M_UR50S). It is also worth noting that by utilizing the same PLM, our implementation demonstrated similar performance to NetSurfP-3.0.

Table 2.3: Performance of Various SSPs

Name	CASP12		CB513		TS115	
	Q8	Q3	Q8	Q3	Q8	Q3
IDP-ELM, esm2_3B (this work)	0.7291	0.8368	0.7698	0.8778	0.7902	0.8799
IDP-ELM, esm1b (this work)	0.6774	0.7922	0.7197	0.8419	0.7515	0.8515
NetSurfP-2.0	0.699	0.810	0.713	0.845	0.740	0.849
NetSurfP-3.0	0.669	0.791	0.711	0.846	0.749	0.856

2.3.5 ENSEMBLE LEARNING IMPROVES THE PERFORMANCE OF IDP PREDICTION

We conduct a comparison of our IDP predictors based on different PLMs, evaluating various metrics on the test data set (Table 2.4). Different PLM-based predictors exhibit notable variations in performance. The smallest PLM (esm2_t6_8M_UR50D) performs unfavorably in all metrics due to its limited model size. However, larger PLMs do not consistently outperform smaller ones, e.g., when comparing the 150 M ESM model (esm2_t30_150M_UR50D) to the 650 M ESM model (esm2_t33_650M_UR50D). The latter has worse scores, except for the Rec score. The performance of each predictor also displays variability across different metrics, as seen in the comparison between the 3B ESM model (esm2_t36_3B_UR50D) and the 15B ESM model (esm2_t48_15B_UR50D). We conclude that this variability arises due to several factors: (1) IDRs exhibit an imbalanced distribution within proteins, leading to greater variance in performance. (2) The labeled data sets, encompassing training, validation, and test data, are relatively small, limiting the predictive capabilities of these PLM-based IDP predictors. (3) PLMs are primarily designed to capture self-interactions among residues, and larger PLMs possess greater capacity for

learning and storing such interaction patterns across billions of data points. Since IDRs, being unstructured regions in proteins, may lack strong correlations with other residues in the same protein, the advantages conferred by large model sizes offer limited benefits in the context of IDP prediction.

Table 2.4: Performance of IDP Prediction Based on Different PLMs on the Test Data Set

Models	AUC	F ₁	MCC	Rec	Pre
esm1b_t33_650M_UR50S	0.8345	0.6310	0.4785	0.7265	0.5577
esm2_t6_8M_UR50D	0.8108	0.5881	0.4150	0.8515	0.4491
esm2_t12_35M_UR50D	0.8271	0.6217	0.4626	0.7849	0.5147
esm2_t30_150M_UR50D	0.8334	0.6321	0.4812	0.7156	0.5660
esm2_t33_650M_UR50D	0.8321	0.6284	0.4746	0.8271	0.5066
esm2_t36_3B_UR50D	0.8278	0.6217	0.4625	0.7640	0.5241
esm2_t48_15B_UR50D	0.8332	0.5957	0.4301	0.8744	0.4518
prot_t5_xl_bfd	0.8351	0.6157	0.4589	0.8610	0.4792
prot_t5_xl_uniref50	0.8395	0.6031	0.4400	0.8639	0.4633
ensemble	0.8469	0.6325	0.4814	0.8344	0.5092

Despite the disparities among different PLM-based IDP predictors, we employ ensemble learning to augment their performance, named ensemble. As depicted in the final line of Table 2.4, we observe improvements in AUC, F₁, and MCC scores, as well as favorable results in Rec and Pre scores for the ensemble predictor, thereby underscoring the necessity of employing such a technique. It is important to note that the Rec and Pre are biases metrics: although the 15B ESM model achieves the highest Rec score, it attains the second-worst Pre score, while the 150 M ESM model achieves the highest Pre score but has a worse Rec score.

Furthermore, we compared the DFL predictors and DP predictors utilizing different PLMs (Tables 2.12 and 2.13). The results indicate that ensemble learning indeed enhances the performance compared to that of the single predictor, as evidenced by improved scores in AUC, F₁, and MCC scores.

2.3.6 IDP-ELM-BASED IDP PREDICTIONS ACHIEVED THE STATE-OF-THE-ART PERFORMANCE

We compared the performance of various IDP predictors by evaluating on the nonredundant CAID test data set. The predictions of the compared predictors were obtained from the CAID competition, and the results on the redundant CAID are also provided in Table 2.14. It is important to note that

we utilized the default threshold of 0.5 for evaluating all predictors, while the metrics reported in the CAID paper [30] are based on the threshold selection that maximizes the F1 score. Our IDP-ELM exhibits a remarkable advantage in terms of AUC (0.8554), F1 (0.5031), and MCC (0.4332), and it achieves top performance on Rec (0.8724), as shown in Table 2.15. IDP-ELM outperforms other state-of-the-art predictors except for the value of Pre (0.3535), which is relatively lower than the best achieved value of 0.5042 by fIDPnn. Figure 2.8 presents the top 10 IDP predictors plotted with the AUC, F1, and MCC metrics. Additionally, the ROC curve, displayed in Figure 2.9A, clearly illustrates the significant improvement of IDP-ELM compared to that of all other predictors. The precision-recall curve demonstrates that our IDP-ELM attains the highest F1 score when the optimal thresholds are chosen (Figure 2.9B), where the gray dashed curves represent the points with the same F1 scores. Furthermore, we conducted a Student's t-test to compare the predictions of IDP-ELM with those of other IDP predictors, and the results indicate that IDP-ELM significantly outperforms all existing methods (P-value < 0.05).

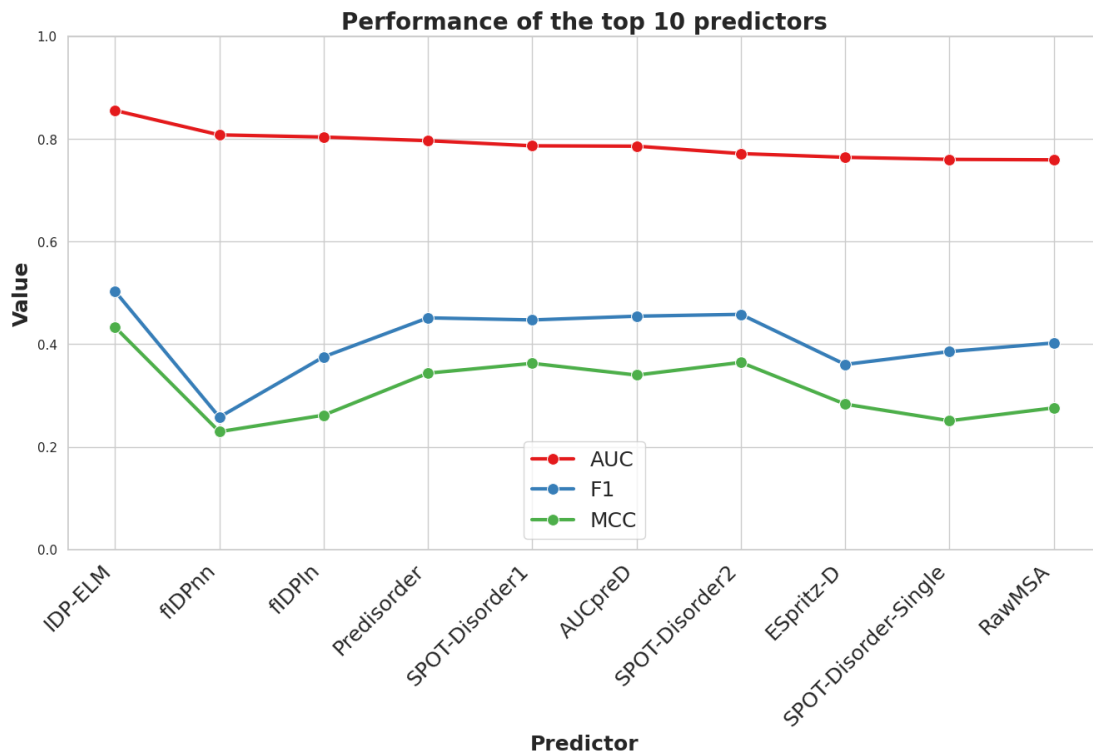


Figure 2.8: Performance of top 10 predictors on nonredundant CAID test data set.

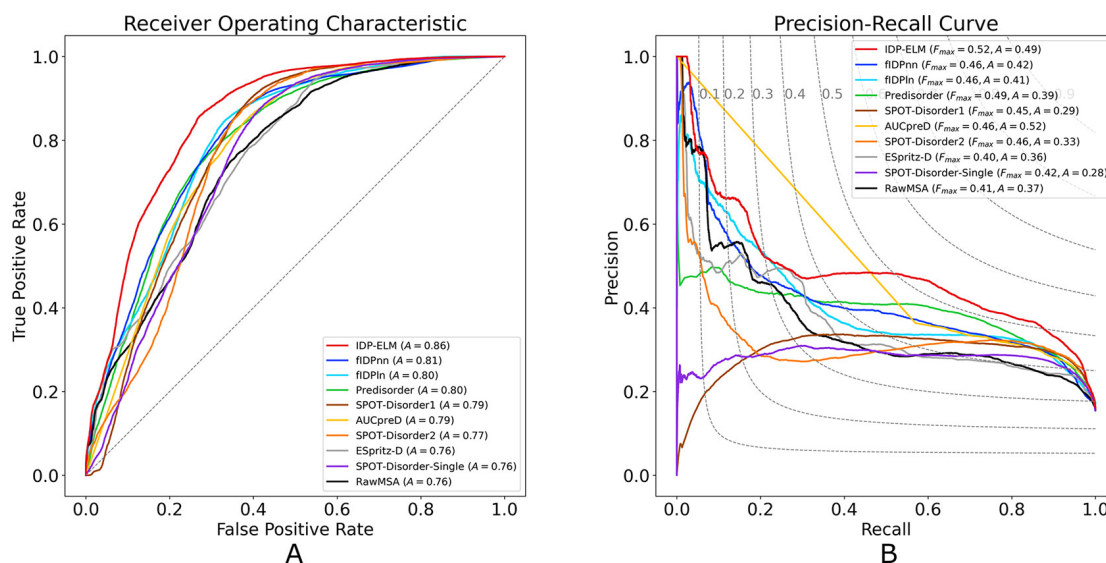


Figure 2.9: (A) Receiver operating characteristic curves and (B) precision-recall curves of top 10 IDP predictors on nonredundant CAID test data set.

To comprehensively assess the performance of various IDP predictors, we constructed a new test data set based on the CAID2 competition [31] test data set. The predictions of the compared predictors were obtained from the CAID2 competition, and the results on redundant CAID2 can also be found in Table 2.16. It is worth noting that some predictors, e.g., SPOT-Disorder2, failed on a large part of sequences (64 out of 284) since it relies on many input features from other complicated predictors that may fail to produce features for some sequences, while our IDP-ELM always produces reliable predictions since it requires only the sequence.

Our IDP-ELM exhibits a remarkable advantage in terms of AUC (0.9788), F_1 (0.9186), MCC (0.8717), and Rec (0.9457), while it achieves a top performance on Pre (0.8930), as shown in Table 2.17. Figure 2.10 presents the top 10 IDP predictors plotted with the AUC, F_1 , and MCC metrics. In addition, the ROC (receiver operating characteristic) curve in Figure 2.11A clearly illustrates the significant improvement of IDP-ELM compared to all other predictors. The precision-recall curve, depicted in Figure 2.11B, demonstrates that our IDP-ELM attains the highest F_1 score when the optimal thresholds are chosen, where the gray dashed curve represents points with the same F_1 scores. Furthermore, we conducted a Student's t-test to compare the predictions of IDP-ELM with those of other IDP predictors, and the results indicate that IDP-ELM outperforms all existing methods significantly (P-value < 0.05) except for RawMSA (P-value = 0.2361). However, RawMSA obviously underperforms our IDP-ELM on nonredundant CAID test data set.

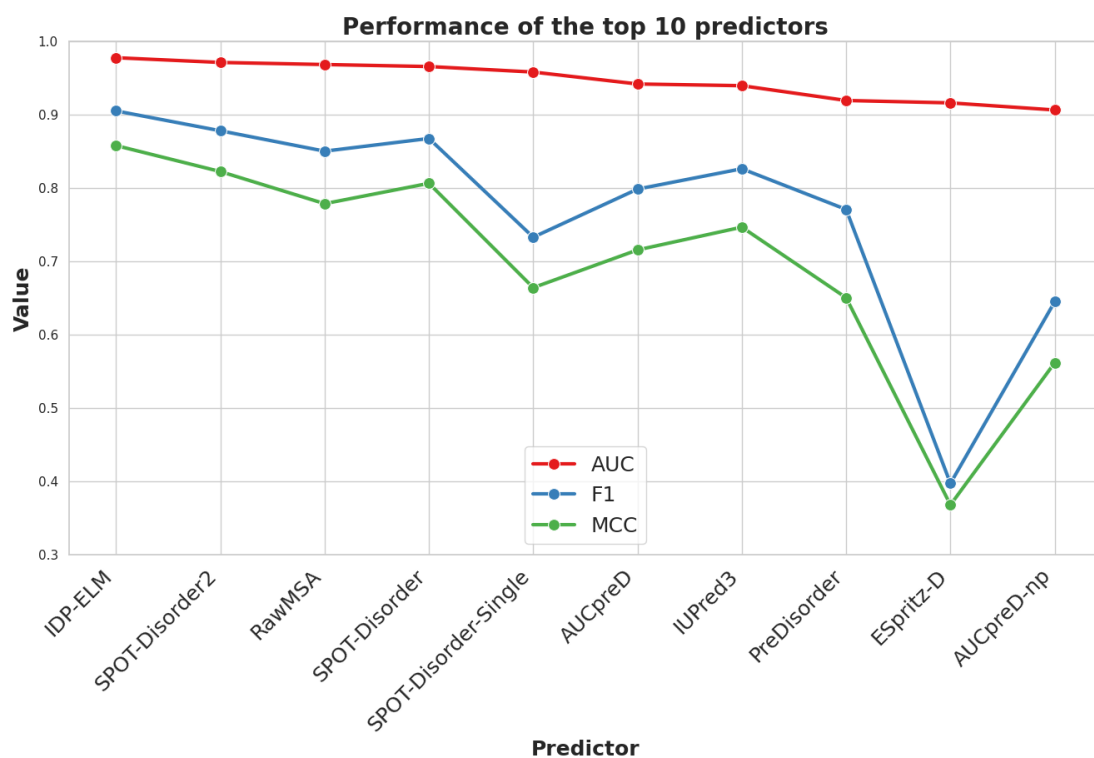


Figure 2.10: Performance of top 10 IDP predictors on nonredundant CAID2 test data set.

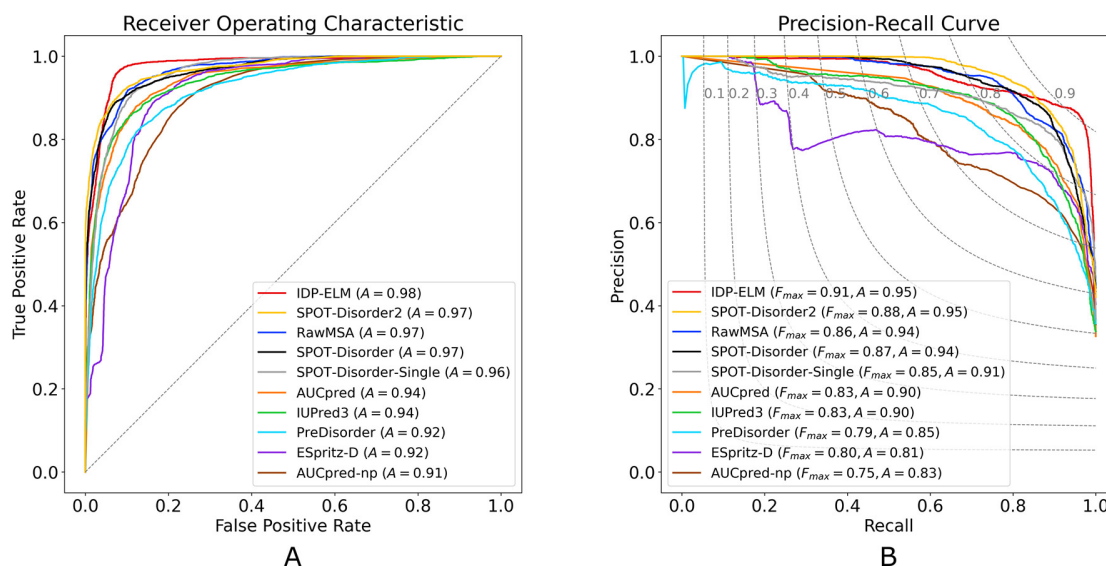


Figure 2.11: (A) Receiver operating characteristic curves and (B) precision-recall curves of top 10 IDP predictors on non-redundant CAID2 test data set.

2.3.7 EXEMPLARY VISUALIZATIONS

To test our IDP-ELM, two protein sequences were analyzed by different top predictors, and the results of their predictions are visualized in Figures 9 and 10, demonstrating the accuracy of our IDP-ELM. Note that some top predictors, e.g., SPOT-Disorder2, failed to predict these sequences, and we do not compare them here.

The first example showcases the HspB6 (Heat shock protein beta-6, UniProt: O14558, DisProt: DP01131), which has a long disordered region spanning from residues 1 to 56. Since the full-length human HSPB6 has not been crystallized by extensive attempts, this protein was crystallized by removing a poorly ordered region by limited proteolysis techniques [47]. Notably, only the prediction from our IDP-ELM matches with the sequence of a long-disordered region with a short false positive region in the tail of the sequence, while the others fail with predicting nothing in the IDR or too many false positives in ordered regions (Figure 2.13).

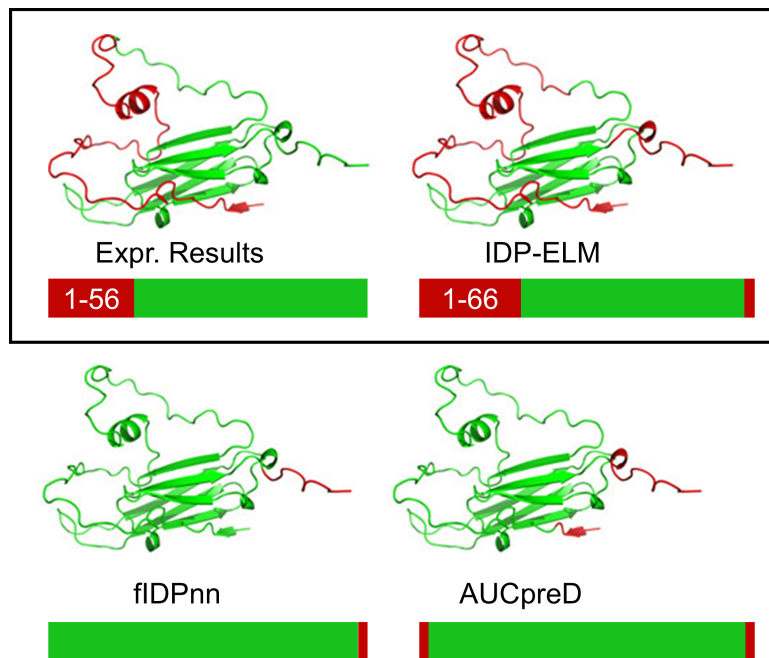


Figure 2.12: Visualization of selected example (DisProt ID: DP01131). 3D structures are predicted by AlphaFold2, and cartoons are visualized by PyMOL. The residues are colored in red (disordered) and green (ordered).

The second example is SNARE (vesicle transport through interaction with t-SNAREs homologue 1A, UniProt: Q9JI51, DisProt: DP01499), which was initially discovered to be unstructured (residue 115-192) in solution but assumed to be a helical conformation when engaged in interactions with other molecules [48]. The presence of diverse conformations within its structure makes it challenging to predict disorder accurately. However, our IDP-ELM successfully captures the disordered regions except a few false positives in the terminal, while other predictors fail, with predicting nothing in the IDR or too many false positives (Figure 2.12).

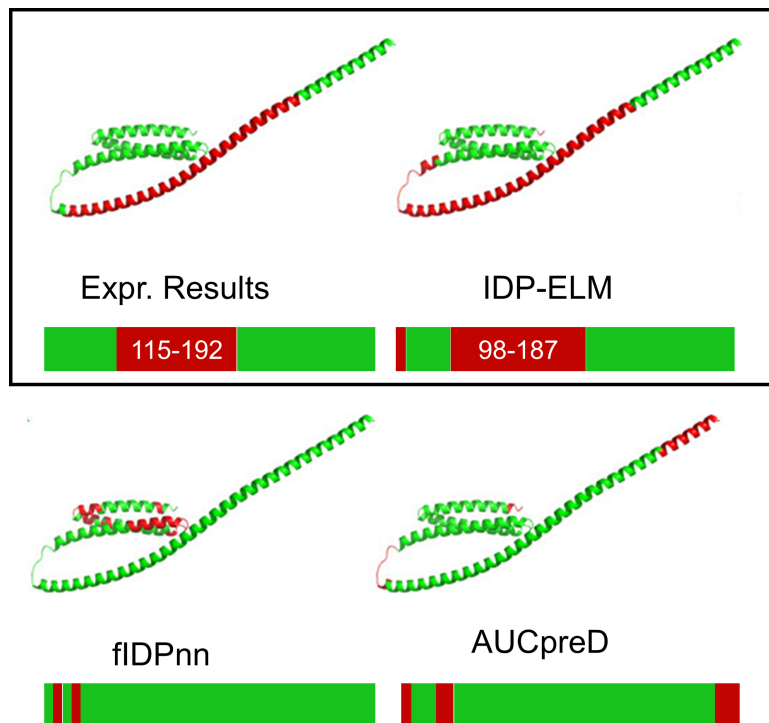


Figure 2.13: Visualization of selected example (DisProt ID: DP01499). 3D structures are predicted by AlphaFold2, and cartoons are visualized by PyMOL. The residues are colored in red (disordered) and green (ordered).

2.3.8 PLM-BASED IDR FUNCTION PREDICTIONS ACHIEVE STATE-OF-THE-ART PERFORMANCE

We thoroughly evaluate the performance of our disordered flexible linker (DFL) predictor on an independent test data set. Our implementation utilizing PLM attained the highest AUC, MCC, and Rec scores compared to other state-of-the-art methods and the second-best Pre score (Table 2.5). The other three methods including TransDFL [49], APOD [35], and DFLpred [50] employed various input features while our methods employed only high-dimensional representations as well as predicted IDRs. Considering the performance of DFL prediction are limited by the extremely limited labeled data size, our method shows the superior advantage of high-dimensional representations in such a scenario.

Table 2.5: Comparison of Different DFL Predictors on the Test Data Set

Predictor	AUC	MCC	Rec	Pre
IDP-ELM (this work)	0.8670	0.5221	0.6482	0.5647
TransDFL	0.835	0.435	0.452	0.586
APOD	0.816	0.418	0.512	0.512
DFLpred	0.637	0.145	0.179	0.337

Furthermore, we evaluate our disordered protein binding (DP) predictor on the independent test data set. Our PLM-based implementation outperforms other state-of-the-art methods in terms of AUC, F1, Rec, and Pre scores (Table 2.6). Note that here we employed thresholds that fit specificity as 0.8, which is the same as the settings of other compared methods in Table 2.6. The other four DP prediction methods including DeepDISOBind [35], MoRFChibiLight [51], DisoRDPbind [52], and ANCHOR₂ [53] employed various input features while our methods employed only high-dimensional representations as well as predicted IDRs. Our method shows the superior advantage of high-dimensional representations in such a scenario.

Table 2.6: Comparison of Different DP Predictors on the Test Data Set

Predictor	AUC	F1	Rec	Pre
IDP-ELM (this work)	0.8748	0.4202	0.6711	0.3058
DeepDISOBind	0.771	0.313	0.595	0.212
MoRFChibiLight	0.735	0.270	0.502	0.185
DisoRDPbind	0.727	0.249	0.456	0.171
ANCHOR ₂	0.719	0.270	0.501	0.185

2.3.9 FAST IDP PREDICTION FOR HUMAN PROTEOME

We executed our IDP-ELM on the human proteome to showcase the advantages of our methodology in terms of efficiency. The human proteome data set, encompassing 34,877 proteins, was obtained from MobiDB data set [54]. Since the human proteome has not been annotated experimentally, we employed the Mobi-lite annotations (V5.0 released in July 2022), which were derived from the consensus of eight distinct IDP predictors, demonstrating a 62.5% agreement rate and undergoing subsequent processing [29, 55]. It is worth noting that the Mobi-lite ranked 12th in CAID₂ and 20th in CAID, implying that they serve as mere references and cannot be considered absolute truths.

It is important to note the computational configurations that we adopted in human proteome prediction. Throughout the prediction pipeline, we utilized three NVIDIA 3090Ti 24 GB GPUs to accommodate the substantial demands of the ever-larger PLM `esm2_t48_15B_UR50D`, which necessitates approximately 70 GB of GPU memory. The distribution of all ESM models across multiple GPUs was facilitated using the Python package `accelerate 0.18.0` [56] with `Transformers 4.26.0`. Apart from the native automatic mixed precision in PyTorch, no further optimizations were implemented.

We generated a length–time curve with the x-axis representing the sequence length of protein sequences and the y-axis denoting the inference time for each length (Figure 2.14). The point on the graph suggests a linear increase in inference time as the sequence length expands. While a naïve self-attention mechanism would yield a time complexity of $O(N^2)$ where N is the length of the sequence, our approach involves truncating protein sequences to a constant length, resulting in a linear time complexity. For most protein sequences comprising hundreds of amino acid residues, IDP-ELM can deliver predictions in just one second. We also employed the Mobi-lite annotations to evaluate the predicted results, which yielded scores of 0.8823 for AUC, 0.6520 for F1, 0.5458 for MCC, 0.9030 for Rec, and 0.5101 for Pre.

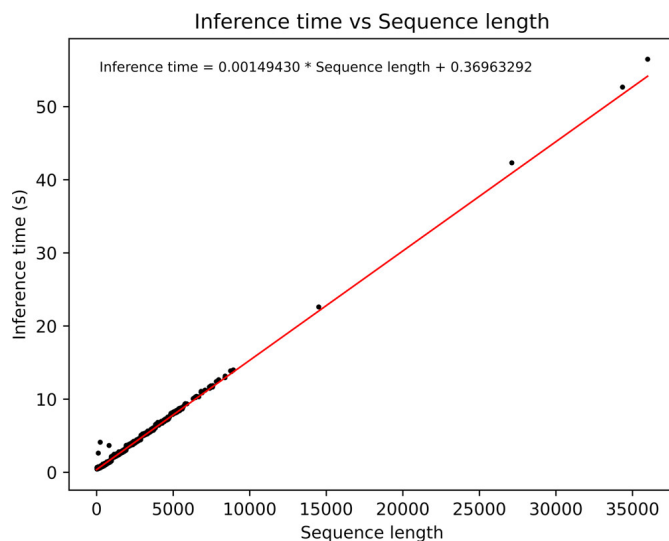


Figure 2.14: Inference time of IDP-ELM for human proteome prediction.

Furthermore, we compared the speed on the redundant CAID data set, consisting of 646 sequences and 336,595 residues. The average prediction time per sequence of our method was 0.8 s, outperforming several state-of-the-art methods mentioned in the cited work [57]: `fIDPnn` (0.5–1 min), `SPOT-Disorder2` (>10 min), `SPOT-Disorder-Single` (0.8–1 min), `RawMSA` (>10 min), `SPOT-Disorder` (5

min) and AUCpreD-np (0.5 min) (Figure 2.15).

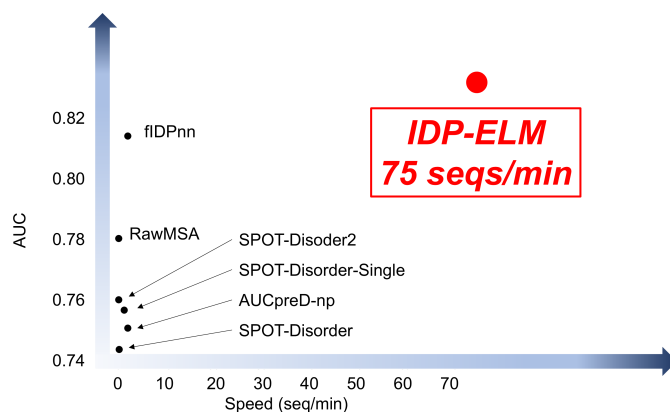


Figure 2.15: Speed comparison of IDP predictors on the CAID data set.

2.4 CONCLUSION

In conclusion, our study presents a novel IDP prediction method, named IDP-ELM, which leverages the power of protein language models (PLMs) and ensemble learning to accurately predict intrinsically disordered regions (IDRs) and their associated functions, including disordered flexible linkers and disordered protein binding. By utilizing high-dimensional representations from multiple PLMs and employing an averaging strategy from ensemble learning, IDP-ELM achieves the state-of-the-art performance of IDR prediction on two challenging independent test data sets CAID and CAID₂. Notably, IDP-ELM always offers a rapid and accurate prediction by solely utilizing protein sequences as inputs, while traditional approaches rely on time-consuming MSA generation and may fail to predict some sequences. This paradigm enables efficient proteome-level analysis without compromising accuracy. Furthermore, our predictions of IDR functions showcase advancements over existing IDR function prediction methods on several test data sets, underscoring the robustness and effectiveness of the IDP-ELM approach.

Overall, our research contributes to the growing field of IDP prediction and highlights the potential of PLMs for extracting vital information from protein sequences. The IDP-ELM method not only enhances the accuracies of IDP prediction but also opens avenues for further exploration and advancements in understanding the role of intrinsically disordered proteins in biological processes.

ACKNOWLEDGMENTS

This work was supported by Hokkaido University DX Doctoral Fellowship (JST SPRING, Grant Number JPMJSP2119) to S.X. and SATREPS Project “Recovering High-Value Bioproducts for Sustainable Fisheries in Chile (ReBiS)” funded by JST/JICA (Grant Number JPMJSA2206) and JSPS KAKENHI Grant Number JP24Ho2213 in Transformative Research Areas (A) JP24A202 Integrated Science of Synthesis by Chemical Structure Reprogramming (SReP) and JP24Ko1533 to A.O.

EXPERIMENTAL SECTION

Table 2.7: Hyper-parameters for Training IDP Predictors

Models	Batch size	Hidden size	# layers
esm1b_t33_650M_UR50S	64	64	2
esm2_t6_8M_UR50D	64	128	2
esm2_t12_35M_UR50D	64	128	3
esm2_t30_150M_UR50D	32	256	3
esm2_t33_650M_UR50D	16	64	3
esm2_t36_3B_UR50D	64	128	2
esm2_t48_15B_UR50D	16	256	2
prot_t5_xl_bfd	64	256	3
prot_t5_xl_uniref50	64	256	2

Table 2.8: Hyper-parameters for Training Disordered Flexible Linker Predictors

Models	Batch size	Hidden size	# layers
esm1b_t33_650M_UR50S	32	16	1
esm2_t6_8M_UR50D	8	16	1
esm2_t12_35M_UR50D	16	16	1
esm2_t30_150M_UR50D	32	16	1
esm2_t33_650M_UR50D	16	32	1
esm2_t36_3B_UR50D	8	16	1
esm2_t48_15B_UR50D	32	64	1
prot_t5_xl_bfd	8	32	1
prot_t5_xl_uniref50	16	64	1

Table 2.9: Hyper-parameters for Training Disordered Protein-binding Predictors

Models	Batch size	Hidden size	# layers
esm1b_t33_650M_UR50S	32	32	1
esm2_t6_8M_UR50D	8	32	1
esm2_t12_35M_UR50D	32	16	1
esm2_t30_150M_UR50D	16	64	1
esm2_t33_650M_UR50D	16	32	1
esm2_t36_3B_UR50D	8	16	1
esm2_t48_15B_UR50D	8	16	1
prot_t5_xl_bfd	8	16	1
prot_t5_xl_uniref50	32	128	1

Table 2.10: Hyper-parameters for Training Different Predictors

Models	Learning rate	Weight decay	Dropout	Mask ratio	Head
SSP	0.001	0.01	0.5	/	BiLSTM
IDP	0.001	0.01	0.5	0.3	BiGRU
DFL	0.001	0.01	0.5	/	BiGRU
Prot	0.001	0.01	0.5	/	BiGRU

refers to secondary structure prediction. IDP refers to intrinsically disordered protein prediction. DFL refers to disordered flexible linker prediction. Prot refers to the disordered protein-binding prediction. The mask ratio is only available for IDP prediction. We adopted a dropout layer between all two neighboring layers of the BiGRU, BiLSTM, and fully connected layers.

Table 2.11: Performance of IDP Prediction Based on Different PLMs on Test Dataset

Models	AUC	F1	MCC	Rec	Pre
esm1b_t33_650M_UR50S	0.8347	0.6226	0.4851	0.6191	0.6261
esm2_t6_8M_UR50D	0.8332	0.6277	0.4882	0.6373	0.6183
esm2_t12_35M_UR50D	0.8267	0.6134	0.4660	0.6326	0.5954
esm2_t30_150M_UR50D	0.8276	0.3174	0.3272	0.1967	0.8220
esm2_t33_650M_UR50D	0.8254	0.5320	0.4477	0.4156	0.7391
esm2_t36_3B_UR50D	0.8292	0.5905	0.4730	0.5218	0.6801
esm2_t48_15B_UR50D	0.8320	0.6061	0.4589	0.6140	0.5983
prot_t5_xl_bfd	0.8411	0.6338	0.4851	0.7053	0.5755
prot_t5_xl_uniref50	0.8299	0.6112	0.4563	0.6587	0.5700
ensemble	0.8484	0.6232	0.4987	0.5853	0.6664

Table 2.12: Performance of DFL Prediction on TE82 Test Datasets

Models	AUC	F1	MCC	Rec	Pre
esm1b_t33_650M_UR50S	0.8627	0.5192	0.4348	0.8286	0.3781
esm2_t6_8M_UR50D	0.7771	0.4496	0.3450	0.8084	0.3114
esm2_t12_35M_UR50D	0.8025	0.4162	0.3014	0.8165	0.2793
esm2_t30_150M_UR50D	0.8214	0.5052	0.4001	0.6847	0.4003
esm2_t33_650M_UR50D	0.8395	0.4732	0.3644	0.7319	0.3496
esm2_t36_3B_UR50D	0.8513	0.5384	0.4498	0.7782	0.4116
esm2_t48_15B_UR50D	0.8275	0.4720	0.3673	0.7670	0.3409
prot_t5_xl_bfd	0.8451	0.5215	0.4286	0.7706	0.3942
prot_t5_xl_uniref50	0.8067	0.4236	0.3079	0.7935	0.2889
ensemble	0.8670	0.6036	0.5221	0.6482	0.5647

Table 2.13: Performance of Disordered Protein-binding Prediction on Test Datasets

Models	AUC	F1	MCC	Rec	Pre
esm1b_t33_650M_UR50S	0.8426	0.3760	0.3487	0.7447	0.2515
esm2_t6_8M_UR50D	0.8330	0.3556	0.3190	0.6954	0.2389
esm2_t12_35M_UR50D	0.8375	0.3633	0.3300	0.7131	0.2437
esm2_t30_150M_UR50D	0.8283	0.3578	0.3219	0.6993	0.2404
esm2_t33_650M_UR50D	0.8426	0.3631	0.3298	0.7132	0.2436
esm2_t36_3B_UR50D	0.8777	0.3975	0.3800	0.7954	0.2650
esm2_t48_15B_UR50D	0.8742	0.3904	0.3697	0.7791	0.2604
prot_t5_xl_bfd	0.8673	0.3879	0.3659	0.7725	0.2590
prot_t5_xl_uniref50	0.8580	0.3787	0.3525	0.7508	0.2532
ensemble	0.8748	0.4202	0.3812	0.6711	0.3058

Table 2.14: Comparison of Various Predictors on Redundant CAID

Predictor	AUC	F1	MCC	Rec	Pre	Failed
IDP-ELM	0.8315	0.4843	0.3893	0.8098	0.3455	0
fIDPnn	0.8141	0.3878	0.3270	0.2999	0.5485	1
fIDPln	0.7929	0.4457	0.3266	0.5045	0.3991	1
RawMSA	0.7805	0.4459	0.3246	0.6820	0.3312	0
ESpritz-D	0.7741	0.3898	0.2888	0.3522	0.4365	1
DisoMine	0.7648	0.4127	0.2766	0.5740	0.3221	0
SPOT-Disorder2	0.7604	0.4684	0.3430	0.7589	0.3388	36
SPOT-Disorder-Single	0.7570	0.4135	0.2782	0.5564	0.3290	0
AUCpreD	0.7568	0.4329	0.3032	0.6871	0.3160	2
AUCpreD-np	0.7511	0.4183	0.2843	0.5731	0.3293	0
Predisorder	0.7474	0.4180	0.2920	0.8069	0.2820	4
IsUnstruct	0.7441	0.4128	0.2868	0.7478	0.2851	0
SPOT-Disorder1	0.7441	0.4299	0.3110	0.7478	0.3017	2
IUPred2A-short	0.7407	0.4134	0.2778	0.5627	0.3267	0
ESpritz-X	0.7395	0.4026	0.2644	0.5317	0.3240	1
IUPred-short	0.7394	0.4114	0.2752	0.5621	0.3244	1
IUPred-long	0.7368	0.4176	0.2847	0.6451	0.3087	1
MobiDB-lite	0.7367	0.3816	0.2525	0.4112	0.3560	1
IUPred2A-long	0.7353	0.4160	0.2819	0.6342	0.3095	0
VSL2B	0.7322	0.3910	0.2658	0.8147	0.2572	2
JRONN	0.7237	0.3935	0.2587	0.7413	0.2678	1
ESpritz-N	0.7143	0.3951	0.2560	0.6874	0.2772	1
DynaMine	0.7066	0.0322	0.0156	0.0174	0.2117	1
PyHCA	0.7065	0.3852	0.2399	0.6585	0.2723	0
DISOPRED-3.1	0.7010	0.3861	0.2405	0.6448	0.2756	0
DisEMBL-465	0.6855	0.3463	0.2057	0.3873	0.3132	2
S2D-1	0.6719	0.3219	0.1524	0.8667	0.1976	2
S2D-2	0.6541	0.3276	0.1633	0.8580	0.2025	2
DisEMBL-HL	0.6539	0.3321	0.1612	0.5298	0.2418	2
DisPredict-2	0.6371	0.3136	0.1470	0.4156	0.2519	0
GlobPlot	0.6244	0.3034	0.1383	0.3834	0.2510	1
DFLpred	0.4100	0.1135	-0.0168	0.0935	0.1442	0
FoldUnfold	/	0.3650	0.2113	0.6656	0.2514	2

Table 2.15: Comparison of Various Predictors on Non-redundant CAID

Predictor	AUC	F1	MCC	Rec	Pre
IDP-ELM	0.8554	0.5031	0.4332	0.8724	0.3535
fIDPnn	0.8078	0.2579	0.2296	0.1733	0.5042
fIDPln	0.8036	0.3754	0.2618	0.3761	0.3747
Predisorder	0.7965	0.4514	0.3436	0.8670	0.3051
SPOT-Disorder1	0.7865	0.4473	0.3628	0.8578	0.3025
AUCpreD	0.7858	0.4546	0.3399	0.7462	0.3269
SPOT-Disorder2	0.7714	0.4582	0.3646	0.8370	0.3154
ESpritz-D	0.7641	0.3606	0.2833	0.2948	0.4642
SPOT-Disorder-Single	0.7601	0.3857	0.2509	0.5927	0.2859
RawMSA	0.7592	0.4025	0.2760	0.6481	0.2919
AUCpreD-np	0.7554	0.4026	0.2745	0.6161	0.2990
DisoMine	0.7496	0.3640	0.2203	0.5698	0.2674
IsUnstruct	0.7463	0.3996	0.2887	0.7961	0.2667
IUPred-long	0.7358	0.3960	0.2715	0.7009	0.2759
IUPred2A-short	0.7350	0.4014	0.2726	0.6102	0.2990
DISOPRED-3.1	0.7345	0.3927	0.2724	0.7486	0.2662
IUPred2A-long	0.7342	0.3943	0.2681	0.6893	0.2761
IUPred-short	0.7332	0.3996	0.2702	0.6095	0.2973
VSL2B	0.7326	0.3814	0.2769	0.8719	0.2441
MobiDB-lite	0.7292	0.3738	0.2423	0.4476	0.3209
JRONN	0.7233	0.3793	0.2600	0.7965	0.2489
ESpritz-X	0.7188	0.3653	0.2229	0.5319	0.2782
ESpritz-N	0.7072	0.3752	0.2462	0.7385	0.2515
PyHCA	0.6960	0.3621	0.2217	0.6825	0.2465
DynaMine	0.6931	0.0252	-0.0103	0.0140	0.1254
S2D-1	0.6829	0.3087	0.1534	0.8969	0.1864
DisEMBL-465	0.6796	0.3203	0.1796	0.3691	0.2829
S2D-2	0.6631	0.3169	0.1705	0.8938	0.1926
DisEMBL-HL	0.6424	0.3124	0.1445	0.5168	0.2238
DisPredict-2	0.6253	0.2723	0.1238	0.3062	0.2452
GlobPlot	0.6065	0.2745	0.0993	0.3996	0.2090
DFLpred	0.3898	0.1025	-0.0251	0.0859	0.1270
FoldUnfold	/	0.3195	0.1540	0.6648	0.2103

Table 2.16: Comparison of Various Predictors on Redundant CAID2

Predictor	AUC	F ₁	MCC	Rec	Pre	Failed
SPOT-Disorder2	0.9494	0.8583	0.7938	0.8227	0.8971	64
IDP-ELM	0.9436	0.8365	0.7772	0.8008	0.8754	0
SPOT-Disorder	0.9314	0.8172	0.7553	0.7633	0.8792	0
AUCpred-profile	0.9239	0.7830	0.7154	0.7033	0.8830	2
SPOT-Disorder-Single	0.9172	0.7131	0.6670	0.5743	0.9404	0
rawMSA	0.9104	0.7132	0.6381	0.6143	0.8500	0
ESpritz-D	0.8989	0.4811	0.4746	0.3247	0.9279	0
AUCpred-no-profile	0.8939	0.6757	0.6141	0.5473	0.8827	0
PreDisorder	0.8930	0.7373	0.6061	0.8285	0.6642	16
disomine	0.8863	0.5544	0.4956	0.4150	0.8350	0
IUPred3	0.8854	0.7352	0.6521	0.6645	0.8227	0
fIDPnn	0.8833	0.5813	0.5399	0.4300	0.8970	0
ESpritz-X	0.8818	0.3706	0.4004	0.2305	0.9459	0
MobiDB-lite	0.8680	0.5839	0.5580	0.4232	0.9416	0
ESpritz-N	0.8586	0.6402	0.5628	0.5227	0.8258	0

Table 2.17: Comparison of Various Predictors on Non-redundant CAID2

Predictor	AUC	F1	MCC	Rec	Pre
IDP-ELM	0.9788	0.9186	0.8717	0.9457	0.8930
RawMSA	0.9717	0.8615	0.7868	0.8436	0.8802
SPOT-Disorder2	0.9700	0.8821	0.8197	0.8573	0.9083
SPOT-Disorder	0.9629	0.8708	0.8020	0.8479	0.8949
SPOT-Disorder-Single	0.9605	0.7549	0.6742	0.6405	0.9190
IUPred3	0.9499	0.8533	0.7730	0.8414	0.8655
AUCpreD	0.9401	0.8042	0.7124	0.7451	0.8735
fDPnn	0.9348	0.6940	0.6186	0.5550	0.9257
PreDisorder	0.9207	0.7913	0.6640	0.8849	0.7157
ESpritz-D	0.9132	0.4139	0.3660	0.2744	0.8413
DisoMine	0.9105	0.5451	0.4571	0.4037	0.8390
MobiDB-lite	0.9095	0.5838	0.5166	0.4308	0.9049
ESpritz-N	0.9071	0.6768	0.5607	0.5731	0.8262
AUCpreD-np	0.9063	0.6632	0.5687	0.5348	0.8729
ESpritz-X	0.8693	0.3800	0.3780	0.2390	0.9270

References

- [1] Johnny Habchi, Peter Tompa, Sonia Longhi, and Vladimir N Uversky. Introducing protein intrinsic disorder. *Chem. Rev.*, 114(13):6561–6588, 2014.
- [2] Zhenling Peng, Jing Yan, Xiao Fan, Marcin J Mizianty, Bin Xue, Kui Wang, Gang Hu, Vladimir N Uversky, and Lukasz Kurgan. Exceptionally abundant exceptions: comprehensive characterization of intrinsic disorder in all domains of life. *Cell. Mol. Life Sci.*, 72:137–151, 2015.
- [3] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine Michoud, Claire O’Donovan, Isabelle Phan, et al. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic Acids Res.*, 31(1):365–370, 2003.
- [4] Bin Xue, A Keith Dunker, and Vladimir N Uversky. Orderly order in protein intrinsic disorder distribution: disorder in 3500 proteomes from viruses and the three domains of life. *J. Biomol. Struct. Dyn.*, 30(2):137–149, 2012.
- [5] Martín Carballo-Pacheco and Birgit Strodel. Comparison of force fields for alzheimer’s a: A case study for intrinsically disordered proteins. *Protein Sci.*, 26(2):174–185, 2017.
- [6] Orkid Coskuner and Vladimir N Uversky. Intrinsically disordered proteins in various hypotheses on the pathogenesis of alzheimer’s and parkinson’s diseases. *Prog. Mol. Biol. Transl. Sci.*, 166:145–223, 2019.
- [7] Krithika Rajagopalan, Steven M Mooney, Nehal Parekh, Robert H Getzenberg, and Prakash Kulkarni. A majority of the cancer/testis antigens are intrinsically disordered proteins. *J. Cell. Biochem.*, 112(11):3256–3267, 2011.
- [8] Susmitha Ambadipudi and Markus Zweckstetter. Targeting intrinsically disordered proteins in rational drug discovery. *Expert Opin. Drug Discovery*, 11(1):65–77, 2016.
- [9] Federica Quaglia, Bálint Mészáros, Edoardo Salladini, András Hatos, Rita Pancsa, Lucía B Chemes, Mátyás Pajkos, Tamas Lazar, Samuel Peña-Díaz, Jaime Santos, et al. Disprot in 2022:

- improved quality and accessibility of protein intrinsic disorder annotation. *Nucleic Acids Res.*, 50(D1):D480–D487, 2022.
- [10] Yumeng Liu, Xiaolong Wang, and Bin Liu. Idp-crf: intrinsically disordered protein/region identification based on conditional random fields. *Int. J. Mol. Sci.*, 19(9):2483, 2018.
 - [11] Yi-Jun Tang, Yi-He Pang, and Bin Liu. Idp-seq2seq: identification of intrinsically disordered regions based on sequence to sequence learning. *Bioinformatics*, 36(21):5177–5186, 2020.
 - [12] Yumeng Liu, Xiaolong Wang, and Bin Liu. Rfpr-idp: reduce the false positive rates for intrinsically disordered protein and region prediction by incorporating both fully ordered proteins and disordered proteins. *Briefings Bioinf.*, 22(2):2000–2011, 2021.
 - [13] Gang Hu, Akila Katuwawala, Kui Wang, Zhonghua Wu, Sina Ghadermarzi, Jianzhao Gao, and Lukasz Kurgan. fldpnn: Accurate intrinsic disorder prediction with putative propensities of disorder functions. *Nat. Commun.*, 12(1):4438, 2021.
 - [14] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
 - [15] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.*, 36(4):193–202, 1980.
 - [16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
 - [17] Nelson Perdigão and Agostinho Rosa. Dark proteome database: studies on dark proteins. *High-throughput*, 8(2):8, 2019.
 - [18] Taraneh Zarin, Bob Strome, Alex N Nguyen Ba, Simon Alberti, Julie D Forman-Kay, and Alan M Moses. Proteome-wide signatures of function in highly diverged intrinsically disordered regions. *Elife*, 8:e46883, 2019.
 - [19] Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Res.*, 51(D1):D523–D531, 2023.
 - [20] Sheng Wang, Jianzhu Ma, and Jinbo Xu. Aucpred: proteome-level protein disorder prediction by auc-maximized deep convolutional neural fields. *Bioinformatics*, 32(17):i672–i679, 2016.

- [21] Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Accurate single-sequence prediction of protein intrinsic disorder by an ensemble of deep recurrent and convolutional architectures. *J. Chem. Inf. Model.*, 58(11):2369–2376, 2018.
- [22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [23] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [24] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci.*, 118(15):e2016239118, 2021.
- [25] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
- [26] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10):7112–7127, 2021.
- [27] Chi Jin and Robert I Cukier. Machine learning can be used to distinguish protein families and generate new proteins belonging to those families. *J. Chem. Phys.*, 151(17), 2019.
- [28] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinf.*, 20:1–17, 2019.
- [29] Marco Necci, Damiano Piovesan, Zsuzsanna Dosztányi, and Silvio CE Tosatto. Mobidb-lite: fast and highly specific consensus prediction of intrinsic disorder in proteins. *Bioinformatics*, 33(9):1402–1404, 2017.
- [30] Marco Necci, Damiano Piovesan, and Silvio CE Tosatto. Critical assessment of protein intrinsic disorder prediction. *Nat. Methods*, 18(5):472–481, 2021.

- [31] CAID-2 Portal. CAID-2 references. <https://caid.idpcentral.org/challenge#Data>, 2024. [Online; accessed 29-September-2024].
- [32] Magnus Haraldson Høie, Erik Nicolas Kiehl, Bent Petersen, Morten Nielsen, Ole Winther, Henrik Nielsen, Jeppe Hallgren, and Paolo Marcatili. Netsurfp-3.0: accurate and fast prediction of protein structural features by protein language models and deep learning. *Nucleic Acids Res.*, 50(W1):W510–W515, 2022.
- [33] Ying Huang, Beifang Niu, Ying Gao, Limin Fu, and Weizhong Li. Cd-hit suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, 26(5):680–682, 2010.
- [34] Zhenling Peng, Qian Xing, and Lukasz Kurgan. Apod: accurate sequence-based predictor of disordered flexible linkers. *Bioinformatics*, 36(Supplement_2):i754–i761, 2020.
- [35] Fuhao Zhang, Bi Zhao, Wenbo Shi, Min Li, and Lukasz Kurgan. Deepdisobind: accurate prediction of rna-, dna-and protein-binding intrinsically disordered residues with deep multi-task learning. *Briefings Bioinf.*, 23(1):bbab521, 2022.
- [36] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pages 2020–12, 2020.
- [37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.
- [38] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.
- [39] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nat. Commun.*, 9(1):2542, 2018.
- [40] Martin Steinegger, Milot Mirdita, and Johannes Söding. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods*, 16(7):603–606, 2019.
- [41] UniProt Consortium. The universal protein resource (uniprot). *Nucleic Acids Res.*, 36(suppl_1):D190–D195, 2007.

- [42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [43] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [44] Angela Lopez-del Rio, Maria Martin, Alexandre Perera-Lluna, and Rabie Saidi. Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction. *Sci. Rep.*, 10(1):14634, 2020.
- [45] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.
- [46] Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldeep Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings Bioinf.*, 19(3):482–494, 2018.
- [47] Stephen D Weeks, Ekaterina V Baranova, Michelle Heirbaut, Steven Beelen, Alexander V Shkumatov, Nikolai B Gusev, and Sergei V Strelkov. Molecular structure and dynamics of the dimeric human small heat shock protein hspb6. *J. Struct. Biol.*, 185(3):342–354, 2014.
- [48] Daniel Zwilling, Anna Cypionka, Wiebke H Pohl, Dirk Fasshauer, Peter J Walla, Markus C Wahl, and Reinhard Jahn. Early endosomal snares form a structurally conserved snare complex and fuse liposomes with multiple topologies. *EMBO J.*, 26(1):9–18, 2007.
- [49] Yihe Pang and Bin Liu. Transdfl: identification of disordered flexible linkers in proteins by transfer learning. *Genomics, Proteomics & Bioinformatics*, 21(2):359–369, 2023.
- [50] Fanchi Meng and Lukasz Kurgan. Dflpred: High-throughput prediction of disordered flexible linker regions in protein sequences. *Bioinformatics*, 32(12):i341–i350, 2016.
- [51] Nawar Malhis, Matthew Jacobson, and Jörg Gsponer. Morfchibi system: software tools for the identification of morfs in protein sequences. *Nucleic Acids Res.*, 44(W1):W488–W493, 2016.
- [52] Christopher J Oldfield, Zhenling Peng, and Lukasz Kurgan. Disordered rna-binding region prediction with disordpbinding. *RNA Chaperones: Methods and Protocols*, pages 225–239, 2020.

- [53] Bálint Mészáros, Gábor Erdős, and Zsuzsanna Dosztányi. Iupred2a: context-dependent prediction of protein disorder as a function of redox state and protein binding. *Nucleic Acids Res.*, 46(W1):W329–W337, 2018.
- [54] Damiano Piovesan, Alessio Del Conte, Damiano Clementel, Alexander Miguel Monzon, Martina Bevilacqua, Maria Cristina Aspromonte, Javier A Iserle, Fernando E Orti, Cristina Marino-Buslje, and Silvio CE Tosatto. Mobidb: 10 years of intrinsically disordered proteins. *Nucleic Acids Res.*, 51(D1):D438–D444, 2023.
- [55] Damiano Piovesan, Alessio Del Conte, Mahta Mehdiabadi, Maria Cristina Aspromonte, Matthias Blum, Giulio Tesei, Sören von Bülow, Kresten Lindorff-Larsen, and Silvio CE Tosatto. Mobidb in 2025: integrating ensemble properties and function annotations for intrinsically disordered proteins. *Nucleic Acids Res.*, 53(D1):D495–D503, 2025.
- [56] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. <https://github.com/huggingface/accelerate>, 2022.
- [57] Bi Zhao and Lukasz Kurgan. Deep learning in prediction of intrinsic disorder in proteins. *Comput. Struct. Biotechnol. J.*, 20:1286–1294, 2022.

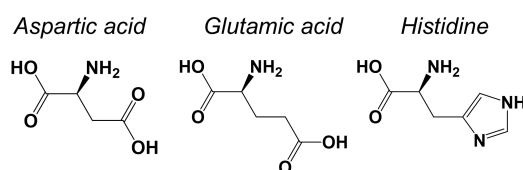
3

High-throughput and accurate pK_a prediction via a protein language model

3.1 INTRODUCTION

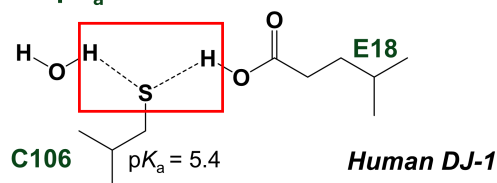
Protein pK_a is crucial for understanding various biological processes, including enzyme catalysis [1, 2, 3], protein folding [4, 5, 6], and ligand binding [7, 8]. Chemical groups in the side chain and terminals can be protonated or deprotonated depending on the pH of the environment, affecting protein structure, stability, and function (Figure 3.1). Currently, the most reliable pK_a values of proteins are only obtained through laborious, costly, and challenging NMR titration experiments [9].

Ionization of residue



pK_a can be measured by NMR titration

Protein pK_a can shift



Other factors also shift pK_a :

- Electrostatics
- Hydrophobicity
- Ligand interactions

Isoelectric point

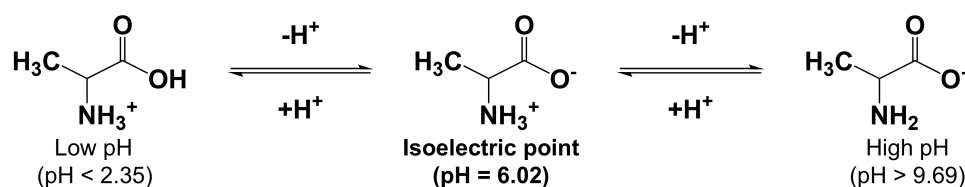


Figure 3.1: Ionizable residues such as Asp, Glu, and His can be protonated or deprotonated depending on the pH of the environment. The ability of these residues to accept or donate protons is quantified by the pK_a value and can be experimentally determined by NMR titration experiments. Many factors can affect the pK_a values of these residues, including the electrostatics, hydrophobicity, and ligand interactions. The isoelectric points of proteins are the pH values at which the proteins exhibit no net charge.

Computational methods for the prediction of protein pK_a have been developed to provide relatively fast and convenient pK_a estimations while maintaining reasonable accuracy (Figure 3.2). One widely used method is the empirical approach, PROPKA [10, 11], which employs a set of human-designed rules to predict pK_a values based on a single protein structure. PROPKA is considered one of the fastest in protein pK_a prediction due to the efficient computation, though its accuracy is limited by the simplicity of its rules. In contrast, constant pH molecular dynamics (CpHMD) simulation methods [12, 13, 14, 15, 16], based on the molecular dynamics (MD) simulations, are considered to be the most accurate computational methods for pK_a prediction. However, they require a long-time sampling of different protein conformations and protonation states, making them computationally intensive and unsuitable for high-throughput pK_a prediction. Furthermore, Poisson-Boltzmann (PB) methods [17, 18, 19], based on continuum electrostatics theory [20], provide the pK_a values for ionizable residues in proteins. PB methods reduce computational costs compared to CpHMD simulations, though at the expense of accuracy. This accuracy decreases due to theoretical assumptions on the basis of modeling the protein-solvent interface as a continuum medium [13] and single protein conformations in the calculations.

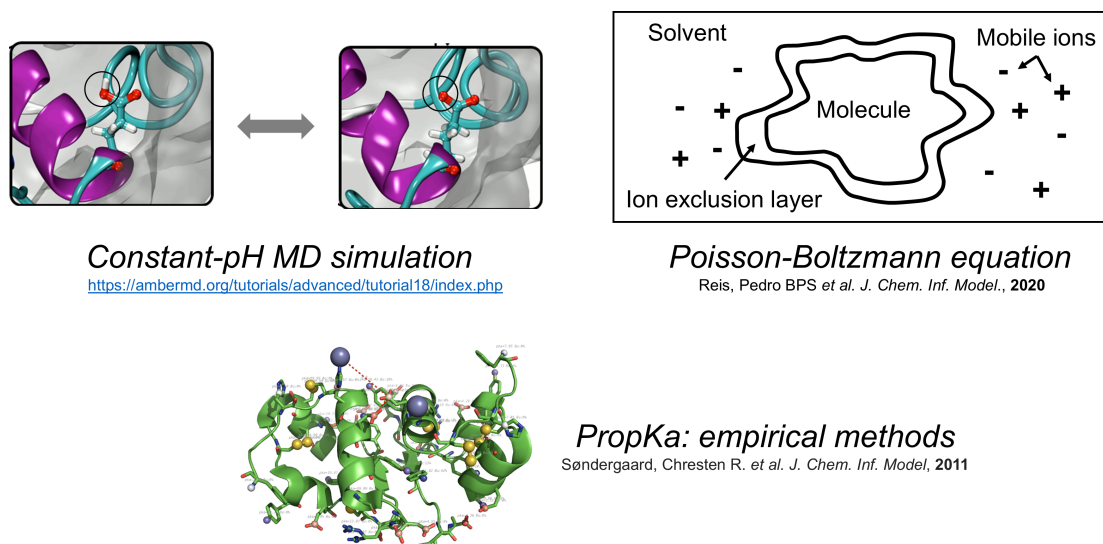


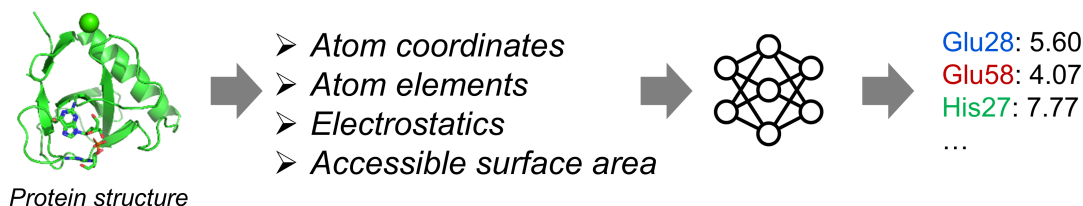
Figure 3.2: Computational methods such as molecular dynamics (MD) simulations, Poisson-Boltzmann (PB) methods, and empirical methods are widely used for protein pK_a prediction.

Protein pK_a prediction is typically approached as a structure-based problem, where the protein structure serves as input features of predictors. However, several issues arise with this approach, especially for high-throughput prediction: 1) Protein structures are not always available for all proteins, limiting the applicability of structure-based methods. Although structure predictors such as AlphaFold2 [21] and RoseTTAFold [22] can provide highly accurate atomic-level structures, their high computational cost makes them unsuitable for high-throughput pK_a prediction. Furthermore, studies indicate that the pK_a prediction based on structures predicted by AlphaFold2 tend to underestimate of the pK_a shifts [23], rendering predictions for significantly shifted pK_a values less reliable. 2) Protein structures are inherently flexible and dynamic, and specific regions can be ambiguous [24], affecting prediction accuracy. Some pK_a predictors use clustering of different conformations to enhance prediction accuracy [25]. However, this clustering process is time-consuming and does not fully address the aforementioned structural discrepancy issue. These discrepancies between structures [26] pose significant challenges to the prediction accuracy of the existing methods. These challenges highlight the need for improved methods in protein pK_a prediction that can account for structural discrepancies, flexibility, and the availability of protein structures.

Machine learning (ML) methods have recently emerged as a promising approach for protein pK_a prediction (Figure 3.3). ML models such as random forests [27], graph neural networks [28], and multilayer perceptron [29], have been applied to predict pK_a values based on protein structures [30, 31, 32, 25]. These ML methods are built with numerous parameters and trained on data sets consisting exper-

imentally determined pK_a values [33, 34] or pK_a values derived from other computational methods [23, 35]. Due to their flexible architectures and scalability, these methods effectively model the complex relationship between the protein structures and pK_a values of protein residues. ML methods achieve both high efficiency and high accuracy in pK_a prediction compared to traditional methods.

Workflow of DL-based pK_a prediction



Reported pK_a prediction methods

Name	Dataset	Year
DeepKa [1]	CpHMD data	2023
pKAI+ [2]	PypKa data	2022

[1] Cai, Zhitao *et al.* *ACS Omega*, **2021** [2] Reis, Pedro BPS *et al.* *J. Chem. Inf. Model.*, **2022**

Figure 3.3: The workflow of deep learning-based methods for protein pK_a prediction includes the pre-processing of input structures (e.g., the coordinates, element types, and the solvent-accessible surface area), the training of deep learning models, and the prediction of pK_a values. In recent years, several deep learning-based methods have been developed for protein pK_a prediction and have achieved state-of-the-art performance.

In recent years, large language models (LLMs) have achieved significant success in various modeling tasks, including natural language processing (NLP) and computer vision (CV). Analogously, protein language models (PLMs) have been developed using architectures such as transformers [36, 37] and the recurrent neural network (RNN) [38]. In a PLM, each residue in a sequence is represented as a high-dimensional vector encapsulating the essential information of the sequences and its context. For instance, residues in similar protein environments are likely represented by proximate vectors in the high-dimensional space. PLMs have demonstrated their effectiveness in various protein property prediction tasks, including secondary structure prediction [39], contact prediction [36], atomic-level structure prediction [37], and intrinsic disorder prediction [40], in terms of prediction accuracy and computational efficiency.

This study introduces pKALM, a novel protein pK_a prediction method based on a protein lan-

guage model. pKALM is a deep learning-based approach capable of predicting pK_a values solely from protein sequences (Figure 3.4). It has been trained to predict the pK_a values of the side chains of Asp, Glu, His, Lys, Cys, Tyr, as well as the N-terminus and C-terminus of proteins. The performance of pKALM was evaluated across a range of residue types, pK_a shifts and positions within proteins. The results demonstrated that pKALM outperformed existing methods. Two isoelectric point (pI) predictors were also developed, achieving state-of-the-art performance, and integrated into the pKALM framework to enhance pK_a prediction accuracy. Furthermore, we demonstrate the high-throughput prediction of pK_a values for all proteins in the human proteome, emphasizing the significantly faster prediction capabilities of pKALM. A case study is also provided to elucidate the reasons why sequence-based pKALM can effectively predict pK_a values, with the interpretability of the visualized attention maps of the PLM.

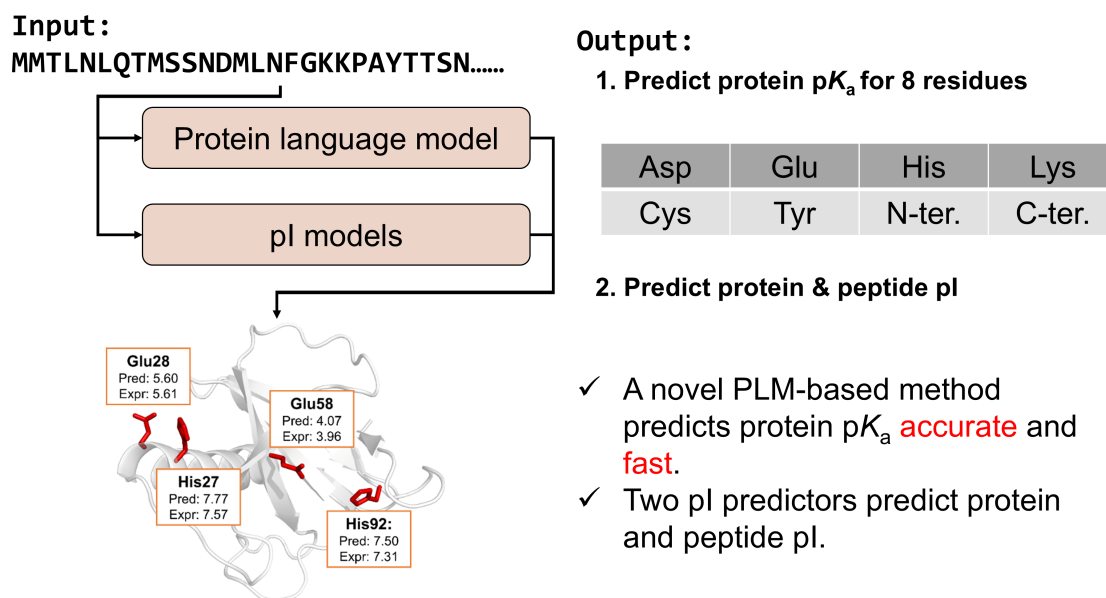


Figure 3.4: This work proposed a novel protein pK_a prediction method, pKALM, based on a protein language model and pI models. It can predict pK_a values solely from protein sequences and achieve state-of-the-art performance. The pKALM supports the prediction of six residues side chains and the N-terminus and C-terminus of proteins.

3.2 METHOD

3.2.1 DATA SETS

We utilized the recently released PKAD-2 data set [34] to train and test various protein pK_a prediction methods. PKAD-2 compiles experimental pK_a data from multiple references, providing 1,456 pK_a values for wild-type proteins and 269 pK_a values for mutant proteins. We meticulously revised the original data set to correct missing or erroneous information, excluding unverifiable or non-numerical pK_a values. The revised data set now includes 1,450 pK_a values for 165 wild-type proteins and 262 pK_a values for 47 mutant proteins. The revised PKAD-2 is available in the associated GitHub repository <https://github.com/xu-shi-jie/pKALM>.

The PKAD-2 data set exhibits considerable redundancy, with homologous proteins having very similar pK_a values for the same residues. This redundancy presents two critical issues: 1) it causes overlap between training and test sets, compromising the fairness of ML model evaluations, and 2) it hampers the generalization of ML models to novel, non-homologous proteins. To address these issues, we utilized the cd-hit program [41] to cluster the revised PKAD-2 data set at 30% sequence similarity, a critical threshold (compared to 30% in the work of Cai *et al.* [31] and 90% in the work of Reis *et al.* [32]). The resulting non-redundant data set was randomly divided into training and test sets in a 5:2 ratio, with 376 pK_a values in the training set and 152 pK_a values in test set. Notably, the size of test set closely aligns with those used in related studies [31].

The literature bias results in an extremely unbalanced distribution of pK_a values among different residues. Predictions for less abundant residues (Cys, Tyr, N-terminus, and C-terminus) are more challenging compared to abundant residues (Asp, Glu, His, and Lys). Table 3.1 presents the pK_a values for both abundant and less abundant residues. The detailed data set construction process is described in Supporting Information, and the pK_a value distribution (Figure 3.10).

Table 3.1: The numbers of pK_a values of different residues in the data set.

	Residue	Training set	Test set
Abundant	Asp	108	44
	Glu	121	48
	His	64	26
	Lys	38	15
Less abundant	Cys	11	4
	Tyr	14	6
	N-terminus	8	4
	C-terminus	12	5
Total		376	152

We utilized isoelectric point (pI) data from Kozlowski’s work [42], which includes both peptide and protein pI data sets. These data sets, derived from references containing experimental pI values, comprise 119,092 peptide pI values and 2,324 protein pI values. Redundancies in both data sets were removed, and they were split into training and test sets in a 3:1 ratio. Figure 3.11 illustrates the detailed distribution of pI values. Notably, compared to the peptide pI data, the protein pI data are significantly fewer and more concentrated, posing a greater challenge for accurate prediction.

3.2.2 PROTEIN LANGUAGE MODEL

Protein language models are large-scale neural networks pre-trained on extensive unlabeled protein sequences. Architectures such as to BERT [43] or T5 [44] are commonly employed for PLMs, characterized by parameters counts ranging from millions to billions, each exhibiting distinct properties. In this study, we evaluated nine PLMs of varied architectures and sizes, including the ESM series [36, 37] and ProtTrans series [39]. Detailed information are provided in Table 3.6.

The scaling law [45] of large language models demonstrates that their performance consistently improves with larger model and training data sizes. This finding aligns with previous research [40], indicating that performance increases with PLM model size but eventually plateaus due to the limited fine-tuning data. Consequently, selecting appropriate PLMs is critical for tasks involving protein property prediction, contingent upon the scale of available fine-tuning data.

3.2.3 ARCHITECTURE OF pKALM

The architecture of pKALM predictor is depicted in Figure 3.5a. The input comprises the protein sequence, encoded through a protein language model, a protein pI model, and a peptide pI model to derive feature vectors. These models are held frozen during pKALM training. The concatenated feature vectors are then fed into a bi-directional long short-term memory (BiLSTM) [46] network to capture sequential patterns in the protein sequences. To distinguish the residue types (i.e., Asp, Glu, His, Lys, Cys, Tyr, N-terminus, and C-terminus), they are encoded via an embedding layer and added to the BiLSTM network outputs. The combined vectors are subsequently processed by a simple fully connected layer to predict the pK_a shift. The final predicted pK_a is derived from the sum of the pK_a shift and the standard pK_a , consistent with prior work [11], as detailed in Table 3.7.

Furthermore, both pI models share identical architectures, as illustrated in Figure 3.5b. The input of each pI model includes either a protein or peptide sequence, encoded into vector sequences via a residue embedding layer. These vectors are then fed into a BiLSTM network to learn sequential patterns. From the BiLSTM outputs, only the hidden states corresponding to protonatable residues (i.e., Arg, Asp, Glu, His, Lys, Cys, and Tyr) and the termini are extracted and averaged to form the final feature vector. A fully connected layer converts this vector into the pI value. Each pI model is trained on distinct data sets: the peptide pI data set and the protein pI data set, respectively.

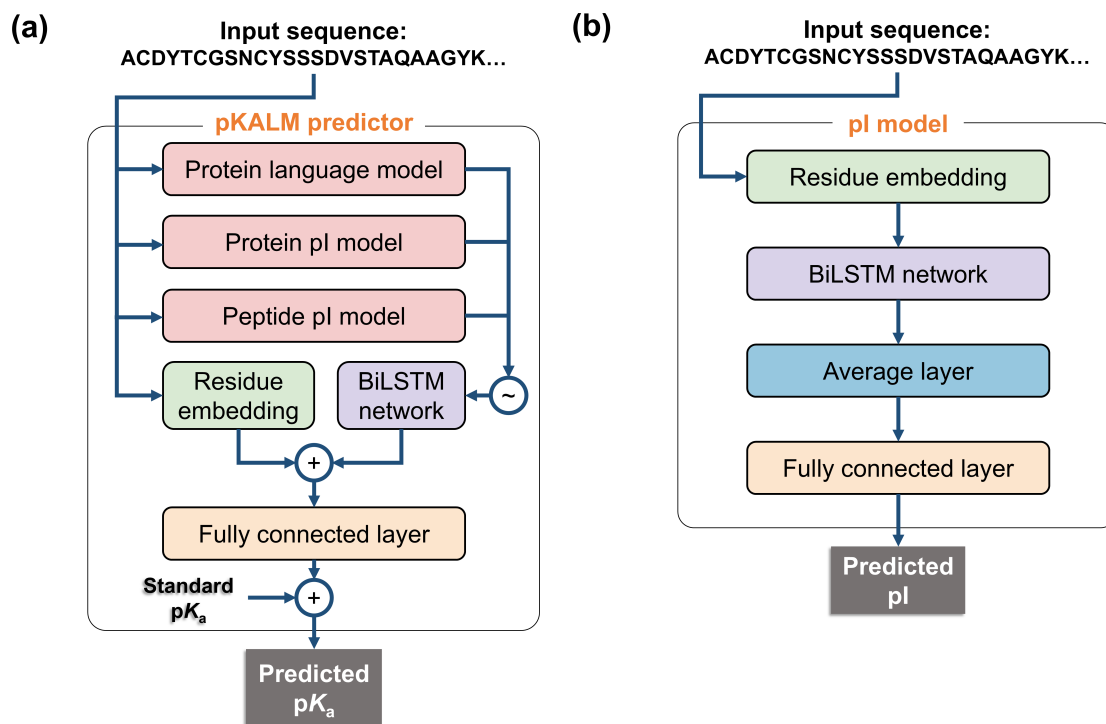


Figure 3.5: Overview of pKALM. (a) The architecture of pKALM. Arrows indicate data flow, while colored blocks represent different neural networks. The symbol “~” denotes concatenation of feature vectors and “+” denotes vector addition. Flow bifurcation indicates data duplication. (b) Architecture of the pI predictors. The pI model is shared by the peptide pI model and protein pI model.

3.2.4 COMPARED METHODS AND DATA PREPARATION

We compare pKALM with five existing pK_a prediction methods including PROPKA [11], PKAI [32], PypKa [19], DeepKa [31], and PKAI+ [32]. For DeepKa predictions, we utilized the online service available at <http://www.computbiophys.com/DeepKa/main>, while for the other methods, we installed their respective software packages locally and assessed their performance.

As both pK_a and pI prediction involve regression tasks, we employed root-mean-square error (RMSE) and mean absolute error (MAE) to evaluate method performance. RMSE, preferred for its sensitivity to data set outliers, served as the primary metric. Furthermore, for pI prediction, we utilized RMSE, MAE, and the coefficient of determination (R^2) to comprehensively assess method effectiveness.

All protein structures within our test set were prepared as below. We retrieved PDB files from the Protein Data Bank [47]. For proteins lacking PDB entries (UniProt IDs: P03230 and P53378), we employed ColabFold [48] to predict their structures. Repairing the protein structures involved using

openMM [49] to handle non-canonical residues (e.g. selenomethionine is replaced to methionine), mutation of wild-type residues, and removal of HETATM records and water molecules. Missing hydrogen atoms were added using the Amber 14 force field [50] and Modeller [51], assuming a pH of 7.0 and standard pK_a values for residues.

3.3 RESULTS AND DISCUSSION

3.3.1 HYPERPARAMETER TUNING AND OPTIMIZATION

We utilize a grid search methodology alongside the wandb package [52] to optimize the hyperparameters of models in the experiments. Specifically, we adjust the dimensions of the hidden states and the layers within the BiLSTM networks for both the pKALM and two pI models. We explore hidden state dimensions and the number of layers in different ranges (see Table 3.8). The optimal hyperparameters are selected based on the performance evaluated through 5-fold cross-validation on the training data set.

We employ the AdamW optimizer [53] throughout our experiments, with a fixed learning rate of 0.001 and a weight decay of 0.001 to mitigate overfitting. Remaining optimizer hyperparameters are set to their default values. Furthermore, we utilize a cosine annealing learning rate scheduler. The batch sizes are set to 32, with model training spanning 100 epochs for protein pK_a prediction, 20 epochs for peptide pI prediction, and 10 epochs for protein pI prediction. Loss functions employed are mean-absolute-error (MAE) for pK_a prediction and mean-squared-error (MSE) for pI prediction.

All experiments are conducted on a Linux workstation equipped with an AMD Ryzen 9 7950X3D 16-core processor, 64 GB memory, and dual NVIDIA RTX 3090Ti GPUs. Implementation utilizes Python 3.12.2 and PyTorch 2.2.1 with CUDA 11.8. For the PLM inference, we utilize Hugging Face’s transformers package (ver 4.39.3) [54] and accelerate package (ver 0.28.0) [55]. Except the native automatic mixed-precision feature of PyTorch [56], no additional optimization techniques are employed to accelerate the model training and inference.

During the training step of a mini-batch, the protein sequences varied in length were pad and truncated into a fixed length for efficient batch processing. We segment longer sequences into smaller segments with a maximum length of 960 tokens (i.e., residues), due to the constraint that some PLMs, such as ESM1B_650M, only accept sequences shorter than 1024 tokens. To retain contextual information, an additional 31 tokens at the end of the segments are appended to each segment, facilitating overlap between adjacent segments, which will be discarded. Final sequence encodings are derived from the concatenated non-overlapping of segment encodings.

3.3.2 PREDICTION OF PEPTIDE AND PROTEIN ISOELECTRIC POINT

The isoelectric point (pI) of a protein is the pH at which the protein exhibits no net charge, while the pK_a values denote the pH at which protein residues are half-protonated. The relationship between the pK_a values of proteins and its isoelectric point (pI) can be elucidated through the Henderson-Hasselbalch equation for weak acids [57]:

$$\text{pH} = pK_a + \log_{10} \left(\frac{[A^-]}{[HA]} \right)$$

The average charge of an ionizable residue at a specific pH can be calculated by $[A^-]/[HA] = 10^{\text{pH}-pK_a}$. The total charge of the protein can be determined by summing the charges of all ionizable groups. The isoelectric point (pI) is thus the pH at which the total charge of the protein is zero. For instance, the pI of a small molecule with two ionizable groups, with pK_a^1 and pK_a^2 , can be calculated as $\text{pI} = (pK_a^1 + pK_a^2)/2$.

Henderson-Hasselbalch equation suggests that pI is closely related to the pK_a values and this study initially trains peptide and protein pI predictors using certain data sets and integrates them into the pKALM framework. Performance comparisons against state-of-the-art methods for peptide pI prediction are summarized in Table 3.2. Our peptide pI predictor exhibits superior performance in terms of RMSE, MAE, and R^2 compared to IPC2, the previous leading method trained on identical data sets. 5-fold cross-validation results for peptide pI prediction are illustrated in Figure 3.12, with the optimal hyperparameters selected.

Table 3.2: Performance comparison of different methods for peptide isoelectric point prediction. Bold font indicates our method.

Method	RMSE	MAE	R^2
pKALM, this work	0.2204	0.1071	0.9765
IPC2 [42]	0.2216	0.1216	0.9761
Bjellqvist [58]	0.4051	0.2836	0.9204
Nozaki [59]	0.4083	0.2673	0.9191
DTASelect [60]	0.4235	0.2796	0.9130
Thurkill [61]	0.4466	0.2535	0.9033
Sillero [62]	0.4747	0.2696	0.8907
Dawson [63]	0.4910	0.2642	0.8831
Wikipedia ^a	0.5178	0.2974	0.8700
Grimsley [64]	0.5264	0.3796	0.8656

^aThe Wikipedia method refers to the pK_a values as presented in the Wikipedia page in 2005.

We develop a protein pI predictor and evaluate its performance against existing benchmarks. Performance comparisons against state-of-the-art methods for protein pI prediction are summarized in Table 3.3. Our method achieves the third-best RMSE and R^2 scores, and the fourth-best MAE among the methods evaluated. Notably, our approach employs deep learning, which typically demands larger data sets for excellent performance. The limited size of available protein pI data sets (see Section “Data set”) constrains the efficacy of our predictor despite employ a comparable architecture. 5-fold cross-validation results for protein pI prediction are depicted in Figure 3.13, with the optimal hyperparameters selected.

Table 3.3: Performance comparison of different methods for protein isoelectric point prediction. Bold font indicates our method.

Method	RMSE	MAE	R^2
IPC2 [42]	0.8479	0.5906	0.5934
IPC [65]	0.8677	0.6109	0.5760
pKALM, this work	0.9069	0.6468	0.5589
ProMoST [66]	0.9113	0.6444	0.5183
Toseland [67]	0.9278	0.6537	0.5095
Dawson [63]	0.9365	0.6586	0.4977
Bjellqvist [58]	0.9369	0.6536	0.5005
Wikipedia ^a	0.9484	0.6795	0.4860
Rodwell [68]	0.9579	0.6762	0.4706
Grimsley [64]	0.9588	0.6953	0.4779

^aThe Wikipedia method refers to the pK_a values as presented in the Wikipedia page in 2005.

3.3.3 PERFORMANCE OF DIFFERENT PROTEIN LANGUAGE MODELS

We assess the performance of multiple protein language models for predicting pK_a values, as depicted in the blue and green curves in Figure 3.6. The evaluation includes validation and test RMSE for nine PLMs: eight PLM-based configurations (with “ESM” or “PT” prefixes) and a compared configuration replacing PLM with an embedding layer (“No PLM”). The points in the curves representing the best configurations selected from validated RMSEs, which consistently decreases with an increase in model parameters from 8 to 650 millions, and eventually achieve the lowest for three largest PLMs: ESM2_650M, ESM2_3B, and ESM2_15B. Other PLMs (ESM1B_650M, PT_BFD, PT_UR), and the non-PLM configuration exhibit varying performances due to their distinct architectures and training data sets from ESM2 series, and their details can be found in the original literature [36, 39].

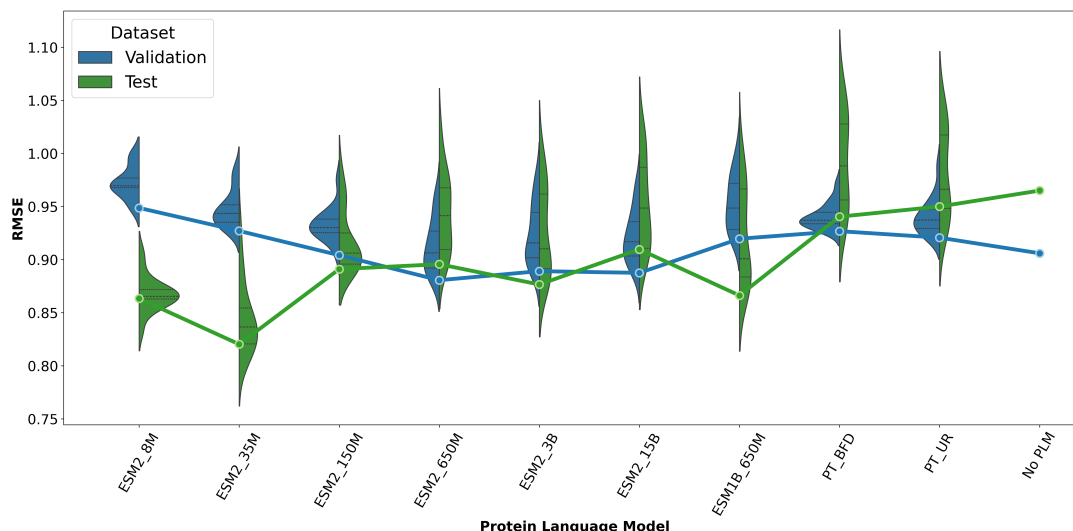


Figure 3.6: Comparison of the performance of different protein language models on the validation and test sets for all models and the best model selection. The violin graph illustrates the distribution of (Validation/Test) RMSEs for different PLMs with different hyperparameters (hidden states and layers). The curves represent the best (Validation/Test) RMSEs selected based on the validation set. Note that for the non-PLM configuration, the distribution of RMSEs across different configurations is very concentrated and we only show the best RMSE.

The test RMSEs, however, diverges notably from the validation RMSEs. We define the pK_a shift as the difference between model-predicted and experimentally determined pK_a , categorizing shifts into five ranges split by 0.5 pH units: 0-0.5, 0.5-1.0, 1.0-1.5, 1.5-2.0, and >2.0. Figure 3.14 illustrates the test RMSE of different PLMs across these ranges. Notably, larger PLMs (ESM1B_650M, ESM2_3B, and ESM2_15B) exhibits superior performance when pK_a shifts are small (<1.0 pH unit). In contrast, for shifts larger than 1.5, the configurations with smaller PLMs (ESM2_8M and ESM2_35M), or without PLM, demonstrate better performance. These findings suggest that larger PLMs excel in capturing evolutionary information from protein sequences and tend to predict normal pK_a values more accurately, while they may be less reliable for pK_a values that shift significantly. Thus, selecting an appropriate PLM is crucial to balance prediction accuracy of pK_a values that shift significantly and those that do not.

The distribution of RMSEs for small-size PLMs is more concentrated, while the gaps between the validation and test RMSEs are larger. In contrast, large-size PLMs exhibit less concentrated RMSEs with smaller gaps between validation and test RMSEs, as shown in the violin graph in Figure 3.6. Overall, the variance in performance across different PLMs and hyperparameters underscores the importance of selecting an appropriate PLM for protein pK_a prediction and highlights the necessity of

hyperparameter tuning for optimal performance. Notably, our choice of PLM is considered *a priori* knowledge rather than a hyperparameter. For subsequent experiments, we prioritize the RMSE of four abundant residues (Asp, Glu, His, and Lys) as the primary metric for selecting the best PLM. ESM2_35M achieves the best performance in terms of test RMSE among the evaluated models.

3.3.4 pKALM OUTPERFORMS EXISTING METHODS

We evaluate the performance of our pKALM method against reported method (Figure 3.7a and 3.7b). The results demonstrate that all methods can predict pK_a values for Asp, Glu, His, and Lys. Importantly, pKALM achieved the lowest test RMSE on four abundant residues (Asp 0.6929, Glu 0.6641, His 1.0468, Lys 0.7701). PKAI+, another deep learning-based method, showed the second-best performance, followed by DeepKa, PypKa, and PKAI. The empirical method PROPKA performs the worst, due to its simplistic empirical rules and limited model complexity. The predictions of Glu residues exhibited relatively small test RMSE among all predictors due to the concentrated pK_a values of glutamic acid. In contrast, the predictions of His residues had the highest RMSE, due to the limited and diverse distribution of pK_a values in the data set. Asp residues, with the second largest data (108/44 in training/test set) and a broader pK_a distribution, provided a fair and challenging to benchmark. Performance comparisons on different residues in terms of RMSE and MAE can be found with details in Table 3.9 and 3.10. Overall, pKALM significantly outperformed other methods.

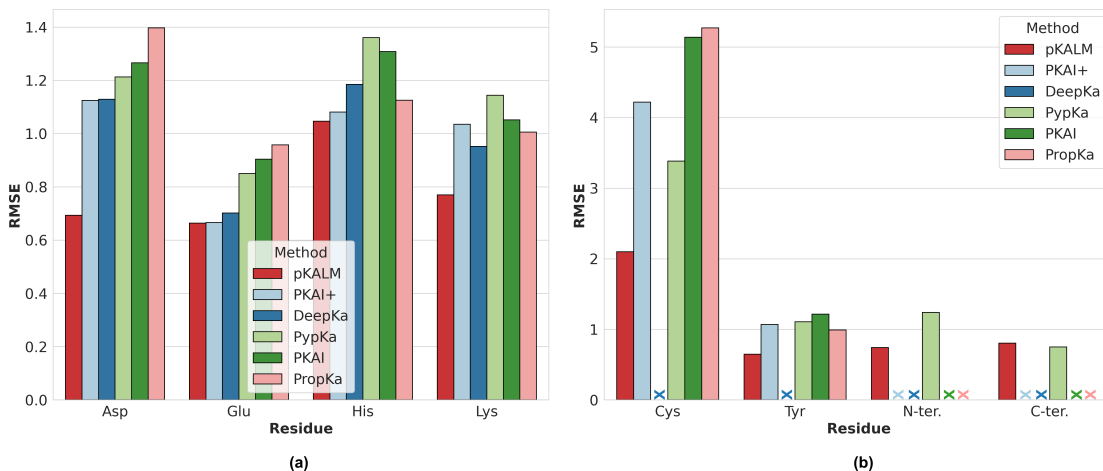


Figure 3.7: Performance comparison of pKALM with existing methods for (a) abundant residues: Asp, Glu, His, and Lys and (b) less abundant residues: Cys, Tyr, N-terminus (N-ter.), and C-terminus (C-ter.). Some methods are not available for certain residues, as indicated by “X”.

The pK_a values of less abundant residues Cys, Tyr, N-terminus, and C-terminus were predicted

in superior performance in terms of RMSEs (Cys 2.098, Tyr 0.6479, N-terminus 0.7419, C-terminus 0.8038) by pKALM relative to existing methods, as illustrated in Figure 3.7b. Predictions for Cys were particularly challenging due to their limited data and very diverse pK_a values.

We compared pKALM with the available methods, marking unavailable ones with an “X”. pKALM is unique in predicting pK_a values for Cys with a RMSE of approximately 2 pH units, whereas other methods showed significantly large RMSEs (PKAI+>4, PypKa>3, PKAI and PROPKA>5), rendering their predictions for Cys unreliable. Despite a limited test set, pKALM achieved the lowest RMSE for Tyr (14/6 in training/test set) and N-terminus (8/4 in training/test set). Detailed performance comparisons in terms of RMSE and MAE can be found in Table 3.11 and 3.12. pKALM and PypKa provided predictions for C-terminus with comparable accuracy (pKALM 0.8038 vs PypKa 0.7496). This indicates that not-machine-learning-based methods remain dependable for residues lacking sufficient data to train machine learning models.

3.3.5 PERFORMANCE ON pK_a SHIFTS

It is crucial to evaluate the performance of pK_a predictors on different pK_a shifts. According to the PKAD-2 dataset, most of pK_a values are within 1 pH unit of their standard values, though some are significantly shifted (e.g., >2 pH units). Predicting small pK_a shifts is relatively straightforward, while predicting large shifts is more challenging, necessitating evaluation across different pK_a shift ranges. According to the previous study [31], we split the pK_a shifts into five ranges with 0.5 pH unit interval: 0-0.5, 0.5-1.0, 1.0-1.5, 1.5-2.0 and >2.0 and assessed the performance of pKALM and existing methods on these ranges (Figure 3.8a).

pKALM achieves the lowest RMSE in ranges ≤ 1.5 while the third best in the range of 1.5-2.0 and the second-best in the range of >2.0. It is worth noting that, another two deep learning-based methods, PKAI and PKAI+, also lag behind the best performance for larger ranges. Deep learning methods are trained for minimizing the overall loss function and may not always focus on learning the largely shifted pK_a values. Considering the large part of small pK_a shifts in the data set, it is reasonable that the deep learning methods are not the best for the largely shifted pK_a values. On the other hand, DeepKa is optimized for relatively balanced distribution of pK_a shifts and thus performs better than PKAI and PKAI+ for the largely shifted pK_a values while still worse than pKALM in terms of accuracy. PROPKA is the best one for the pK_a shifts larger than 2.0 pH units while the data point of this range is very limited. Furthermore, its performance on small pK_a shifts is relatively low, suggesting such a method not reliable for the general pK_a prediction since we do not know which range the pK_a value belongs to before the prediction. In contrast, our method pKALM achieves balanced performance

across different pK_a shift ranges.

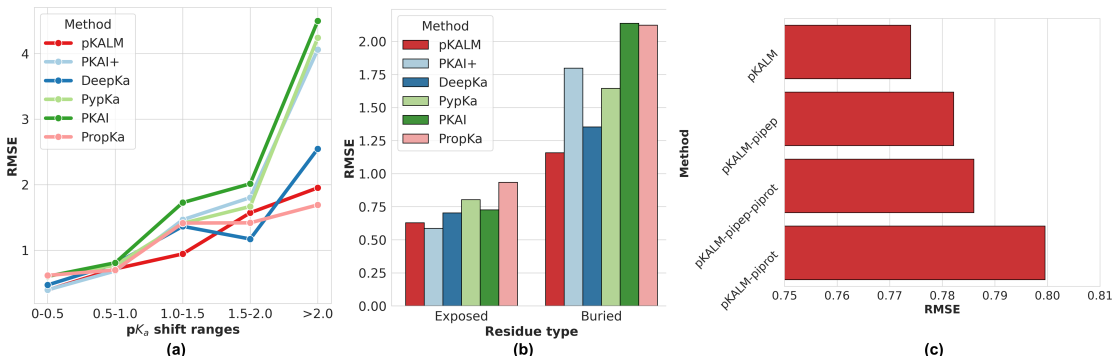


Figure 3.8: (a) Performance comparison of pKALM with reported methods for largely shifted pK_a values. (b) Performance comparison of pKALM with reported methods for exposed and buried residues. (c) Ablation study of pKALM by removing the pI models.

3.3.6 PERFORMANCE ON THE EXPOSED AND BURIED RESIDUES

We evaluate the performance of reported methods for exposed and buried residues separately. The definition of exposed and buried residues are based on the relative solvent accessibility (RSA) of the residues, which is already calculated in the PKAD-2 data set. We choose 50% as the threshold, same to the previous study [30] (Figure 3.8b). It is clear that pKALM achieves the lowest RMSE for challenging buried residues. For exposed residues, pKALM also achieves the second-lowest RMSE (0.6281), following PKAI+ (0.5862), while all predictors have relatively similar performance. This is reasonable since the exposed residues are more accessible to the solvent and thus their pK_a values are more close to the standard pK_a values, suggesting them much easier to predict. The slight difference between pKALM and PKAI+ may be due to the measurement uncertainty of pK_a values in the data set, which is usually around (0.01-0.1).

3.3.7 ABLATION STUDY AND COMPARISON WITH EXISTING METHODS

The necessity of adopting the pI models in pK_a prediction was confirmed by an ablation study of different modules (Figure 3.8c). To investigate the importance of modules inside a deep learning model, we remove the pI models one by one, retrain them with the same data sets, and evaluate them on the same test sets. Four versions of configurations are tested: 1) pKALM, 2) pKALM without peptide pI model “pKALM-pipep”, 3) pKALM without protein pI model “pKALM-piprot”, and 4) pKALM without both pI models “pKALM-pipep-piprot”. We observe the changes of the test RMSE of abun-

dant residues and the results show that the pI models are crucial for the pK_a prediction, though the version without any pI models are good. By removing the peptide pI model, the RMSE increases; removing the protein pI model, the RMSE also increases. When we remove only the protein pI model, the RMSE increases the most. This suggest that protein pI model is the most crucial for this architecture.

3.3.8 HIGH-THROUGHPUT PREDICTION

To evaluate the speeds of pKALM, the pK_a values of all proteins in the human proteome are predicted. The human proteome data set is download from the UniProt website (<https://www.uniprot.org/proteomes/UP000005640>) which contains a total of 20,598 proteins and 3,554,768 protonatable residues. The whole prediction process took 11 minutes 56 seconds on our machine, which produced an average speed of 4,965 pK_a per second. The distribution histograms of the predicted pK_a values of different residues (Figure 3.15) indicates the predictions have a very wide range of pK_a value distributions for all residues. This suggests that pKALM is capable of predicting the significantly shifted pK_a values in the proteins. The predicted pK_a of terminal groups have relatively concentrated distributions, which is consistent with the fact that N-terminal and C-terminal are more likely exposed to the solvent and thus have more concentrated pK_a to its standard pK_a values.

We evaluate the speed of different predictors that compared in this study. We cite the reported speeds of DeepKa in the published work [69]. We have tested the speed on human proteome data set for our pKALM. Other methods are installed with their official programs on our machine. Furthermore, we randomly sample 1,000 structures from the PDB database and predict the pK_a values of all residues in the structures by PKAI, PKAI+, PROPKA. Since PypKa is very slow, we sampled only 100 structures and predicted their pK_a values by PypKa. The speed of these predictors are shown in Table 3.4.

It is clear that pKALM is the fastest method among the compared methods, which is more than 37 times faster than PKAI/PKAI+, 130 times faster than PROPKA, 694 times faster than DeepKa, and 5773 times faster than PypKa. The speed of pKALM is mainly due to its sequence-based design and the parallel computation of the deep learning models on the GPUs. PLMs are considered as the bottleneck of the deep learning models. However, the pK_a prediction of pKALM is performed for all protonatable residues in the same protein simultaneously. The price of PLM inference is paid only once for each protein, instead of each residue, which is more efficient than the residue-by-residue prediction. Another deep learning methods, PKAI/PKAI+ also provide the second and the third best speed. Surprisingly, the empirical method PROPKA predicts more slowly than the deep learning

methods, possibly due to its implementation on the CPU and the sequential prediction of the residues' pK_a values.

Table 3.4: Speed comparison of different methods for pK_a prediction.

Method	Total pK_a	Total time	pK_a per sec
pKALM, this work	3,554,768	11 min 56 s	4,964.76
PKAI	187,312	23 min 31 s	132.75
PKAI+	187,312	23 min 38 s	132.10
PROPKA	219,198	96 min 14 s	37.96
DeepKa	1,537	3 min 35 s	7.15
PypKa	11,565	3 h 43 min 49 s	0.86

^a Note that different methods are tested on different data sets.

3.3.9 CASE STUDY AND DISCUSSION

PREDICTION FOR RIBONUCLEASE T_1

We use pKALM to predict the pK_a values of ribonuclease T_1 (PDB ID: IRGA) to have a case study. Ribonuclease T_1 is a small protein consisting of 104 residues (Figure 3.9a). It has been reported to have pK_a values as shown in Table 3.5. The predicted pK_a values from pKALM are very close to the experimental values. This example demonstrates that pKALM accurately predict the pK_a values of the residues in the protein.

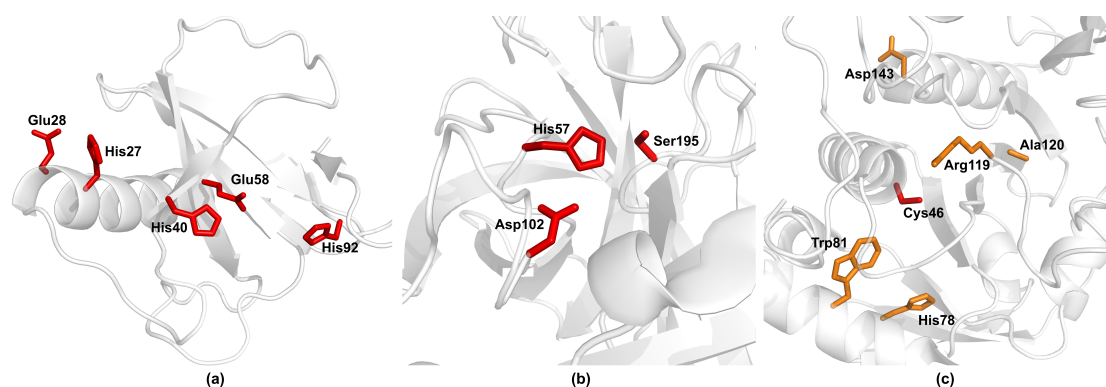


Figure 3.9: Exemplary predictions. (a) The five titrated residues of ribonuclease T_1 . (b) The catalytic triad of chymotrypsin. (c) The bacterial peroxiredoxin AhpC. The catalytic Cys46 is shown in red. Residues with high attention maps to Cys46 are shown in orange.

Table 3.5: The experimental, predicted, and standard pK_a values of selected residues in ribonuclease T₁.

Residue	Experimental	Predicted	Standard
His27	7.57	7.77	6.5
Glu28	5.61	5.60	4.5
His40	7.44	6.19	6.5
Glu58	3.96	4.07	4.5
His92	7.31	7.50	6.5

^a The experimental pK_a values are reported in the literature [70].

PREDICTION FOR CHYMOTRYPSIN

pKALM is employed to predict the catalytic triad motif of chymotrypsin, a serine protease. This motif is comprised of Asp, His, and Ser (Figure 3.9b). The input for pKALM is the B chain of the apoenzyme structure (PDB ID: 4CHA). Given the strong interaction between these three residues, the pK_a values of His57 are significantly altered from their standard pK_a value of 6.5. The predicted pK_a value of His57 from pKALM is 7.65, which is in close agreement with the experimental value of 7.5 [2]. This indicates that pKALM is effective in capturing the intricate interactions between the residues and accurately predicting the pK_a values of the residues in the catalytic triad. It is worth noting that Ser in the catalytic triad is not on the input chains but the prediction is still accurate, possibly due to the strong interactions between the residues in the catalytic triad, which are captured by the protein language model.

WHY PLM-BASED DESIGN WORKS?

We conduct a case study to demonstrate why the PLM-based design of pKALM works very well. We use pKALM to predict the pK_a values of Cys46 in the bacterial peroxiredoxin AhpC (PDB ID: 4MA9), a large protein consisting of five chains and hundreds of residues. The key residues for analysis are shown in Figure 3.9c. The pK_a value of Cys46 is 5.94 [71], which is largely shifted from the standard pK_a value of 9.0. The predicted pK_a value from pKALM is 6.67, which is close to the experimental value, while other methods predict the values including PROPKA (9.0), PKAI (9.64), and PKAI+ (9.08), close to the standard pK_a value of 9.0. This suggests that pKALM is more reliable for the prediction of the pK_a values of Cys residues in proteins.

To visualize the effectiveness of pKALM, the attention maps are considered as the useful tool to interpret the deep learning models [72]. We thus visualize the attention maps of all 20 heads of the

PLM ESM2_35M when the input is the sequence of AhpC (Figure 3.16 and Figure 3.17). Most heads only capture the short-distance interactions between the residues, while some heads capture the long-distance interactions. Here we take the 14th head as an example to analyze the attention map, since it has the most significant long-distance interactions. The most important residues that Cys46 pays attention to are His78, Trp81, Arg119, Ala120, and Asp143. These residues are distant from Cys46 in sequence and close in the 3D structure. The attention map imply that pKALM may successfully build the sequence- pK_a relationship by capturing the long-distance interactions and possibly more essential information of the residues in the protein sequences.

3.4 CONCLUSIONS

The understanding of protein functions in biological processes is contingent upon the knowledge of protein pK_a . It is a significant challenge to accurately measure the pK_a of specific residues in a protein. However, computational methods provide an alternative approach to predict the pK_a values. In this study, we propose a deep learning-based method, pKALM, which employs a protein language model to achieve highly accurate predictions for Asp, Glu, His, Lys, Cys, Tyr, N-terminus, and C-terminus. Furthermore, two pI models were trained and integrated into the pKALM architecture. The pI models are of particular importance for the prediction of pK_a , with the protein pI model being especially crucial in this regard. The performance of pKALM was evaluated on the PKAD-2 dataset, and it was compared with existing methods. The results demonstrate that pKALM outperforms existing methods in terms of root mean square error (RMSE) and mean absolute error (MAE). Furthermore, the performance of pKALM was evaluated on data sets with significantly shifted pK_a values, exposed and buried residues, and the human proteome. Since the current version of pKALM is designed for single-chain inputs, the development of the multi-chain adaptable version is currently underway.

ACKNOWLEDGMENTS

This work was supported by Hokkaido University DX Doctoral Fellowship (JST SPRING, Grant Number JPMJSP2119) to S.X. and SATREPS Project “Recovering High-Value Bioproducts for Sustainable Fisheries in Chile (ReBiS)” funded by JST/JICA (Grant Number JPMJSA2206) and JSPS KAKENHI Grant Number JP24H02213 in Transformative Research Areas (A) JP24A202 Integrated Science of Synthesis by Chemical Structure Reprogramming (SReP) and JP24K01533 to A.O.

EXPERIMENTAL SECTION

Data processing for constructing the training and test data sets

Firstly, we selected all pK_a values in the PKAD-2 dataset for Asp, Cys, Glu, His, Lys, Tyr, C-terminus, and N-terminus residues, including both wild-type and mutant proteins. Protein structures were obtained from the PDBj website <https://pdbj.org/> using the following command:

```
1  rsync -rlpt -v -z --delete \
2  ftp.pdbj.org::ftp_data/structures/divided/pdb/ ./wwpdb
```

(up to 7/2/2024). All structures collected were then fixed using the PDBFixer tool under the openMM platform.

Secondly, we verified that the residues recorded in the PKAD-2 dataset matched the PDB structures (residue ID, residue name, and chain ID). We checked for missing residues in their structures, marking unknown residues with an “X” symbol. Sequences were extracted based on tHis analysis, rather than directly from the “pdbres” field in the PDB files.

Next, we removed redundancy with the following command:

```
1  ./cdhit/psi-cd-hit/psi-cd-hit.pl -i $in_path -o $db30 \
2  -c 0.3 -ce 1e-6 -G 0 -g 1 -para 8 -blp 4
```

and retained only the representative sequences in each cluster for subsequent analysis. The clustered dataset was then split into training, validation, and test sets with a ratio of 4:1:2 using the following Python code:

```
1  test_dfs, val_dfs, train_dfs = [], [], []
2  for res in df['Res Name'].unique():
3      res_df = df[df['Res Name'] == res]
4      # res_df is the dataframe of the residue data for each residues
5      test_dfs.append(res_df[0::7])
6      test_dfs.append(res_df[1::7])
7
8      val_dfs.append(res_df[2::7])
9
```

```
10         train_dfs.append(res_df[3::7])
11         train_dfs.append(res_df[4::7])
12         train_dfs.append(res_df[5::7])
13         train_dfs.append(res_df[6::7])
14     test_df = pd.concat(test_dfs)
15     val_df = pd.concat(val_dfs)
16     train_df = pd.concat(train_dfs)
```

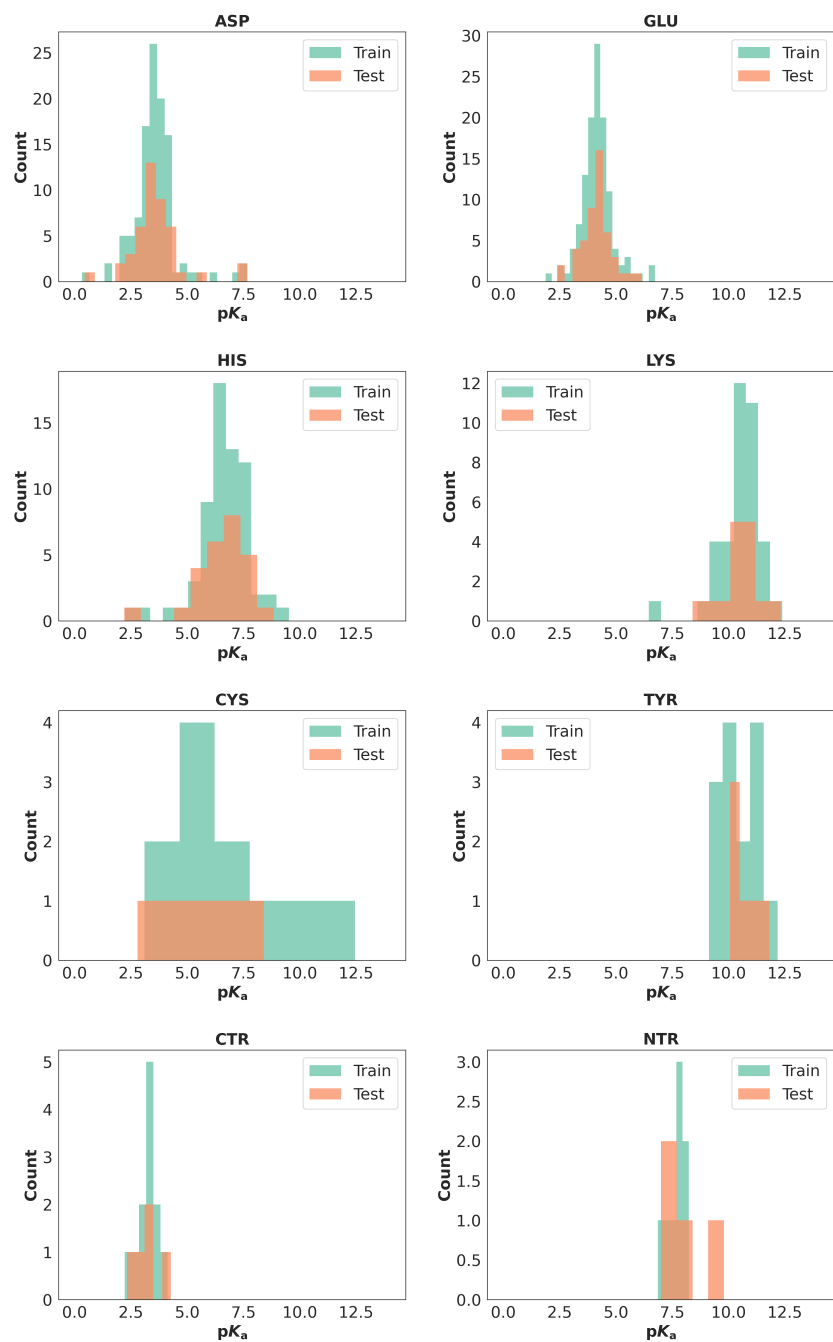


Figure 3.10: The distribution of the pK_a values in the training and test data sets

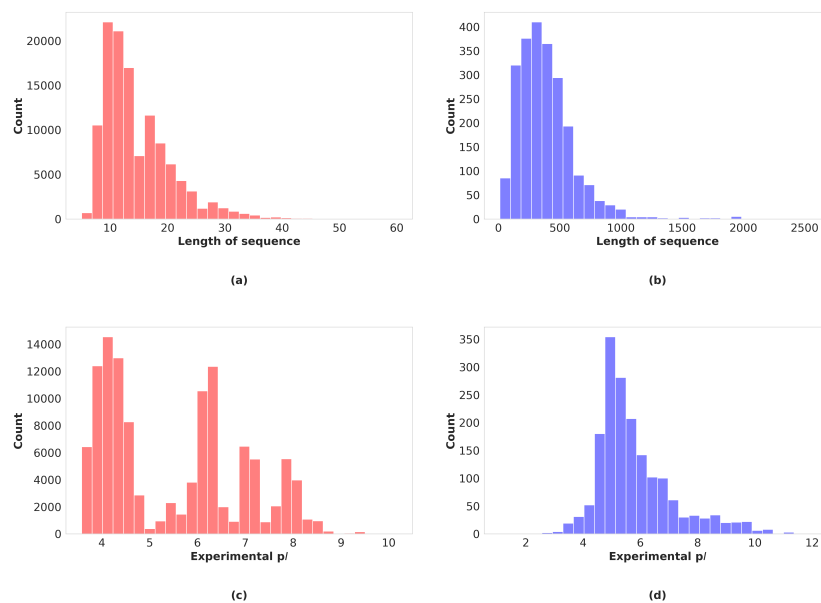


Figure 3.11: (a) The distribution of the sequence lengths of IPC2 peptide data set. (b) The distribution of the sequence lengths of IPC2 protein data set. (c) The distribution of the isoelectric points of IPC2 peptide data set. (d) The distribution of the isoelectric points of IPC2 protein data set.

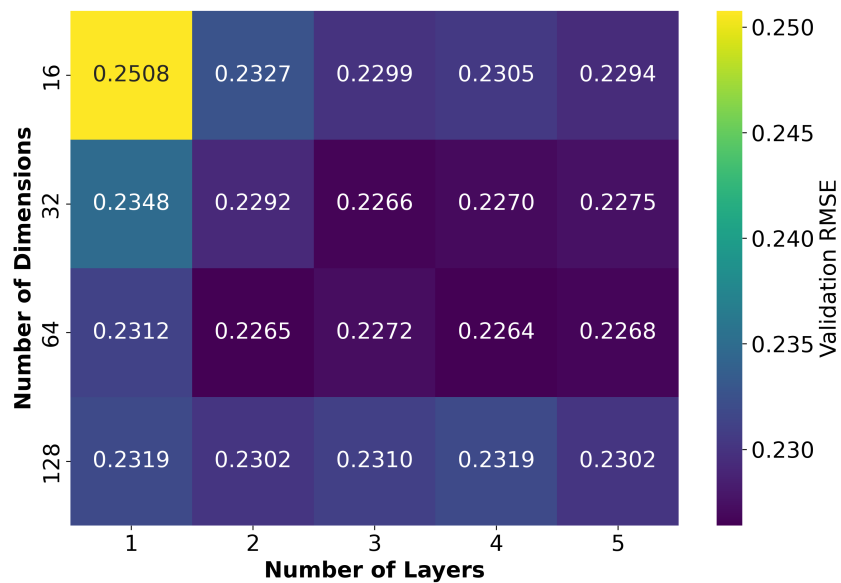


Figure 3.12: Heatmap of validated RMSE of peptide isoelectric point prediction with different hyperparameters

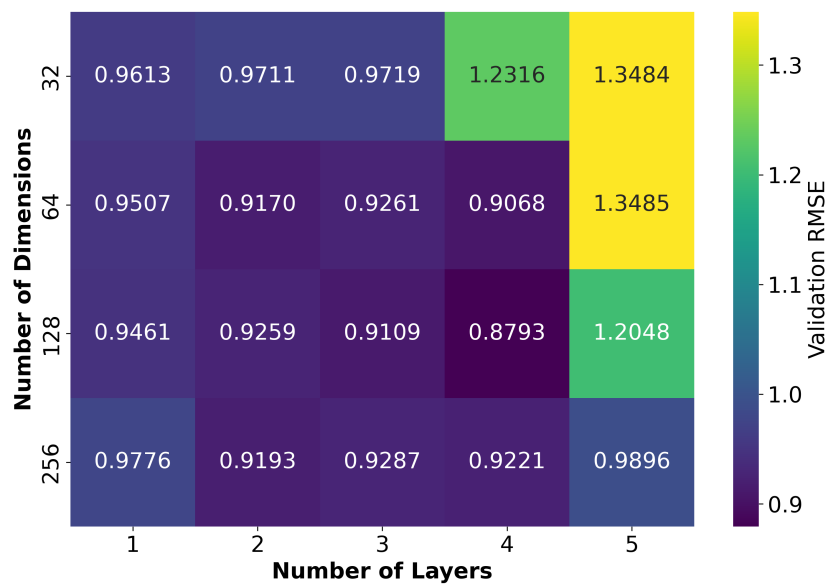


Figure 3.13: Heatmap of validated RMSE of protein isoelectric point prediction with different hyperparameters

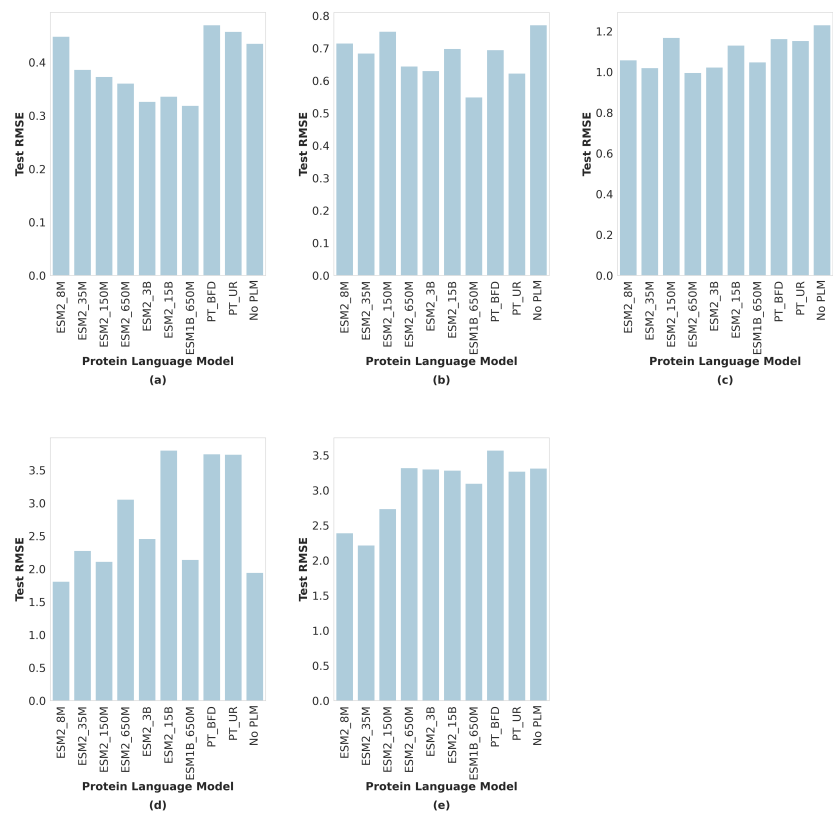


Figure 3.14: Performance of different protein language models on different pK_a shift ranges: (a) 0-0.5, (b) 0.5-1.0, (c) 1.0-1.5, (d) 1.5-2.0, and (e) > 2.0

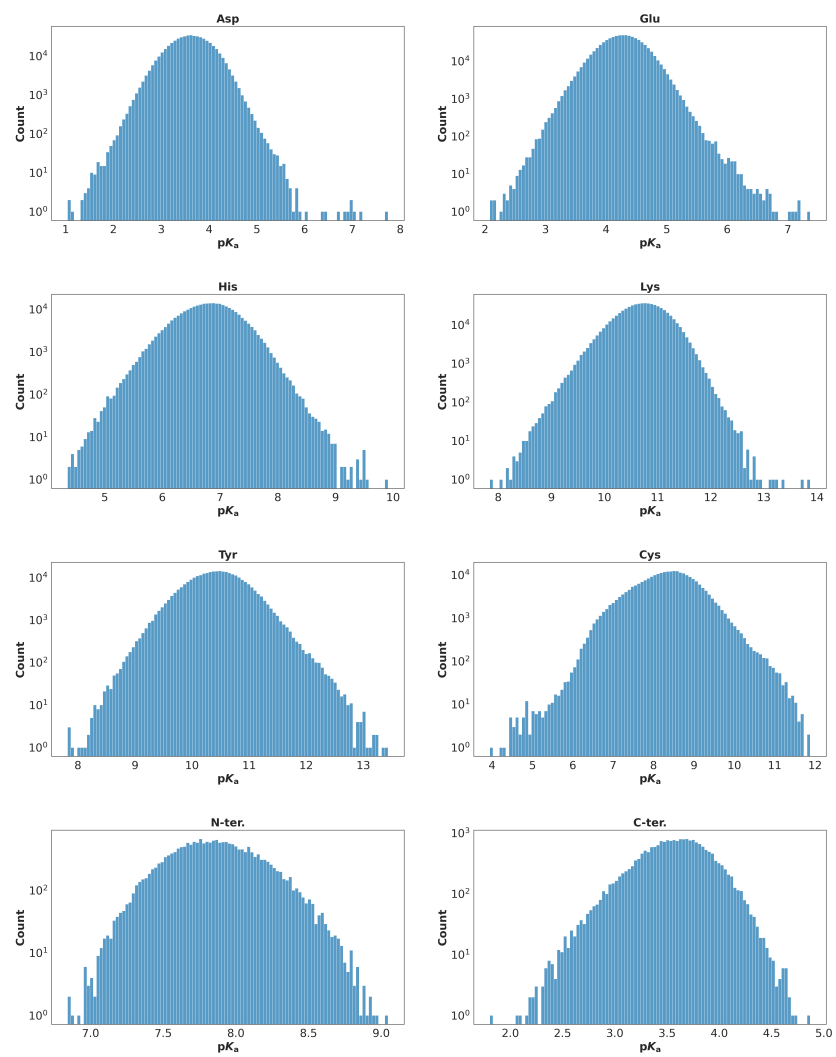


Figure 3.15: Distribution of predicted pK_a values in the human proteome for different residues

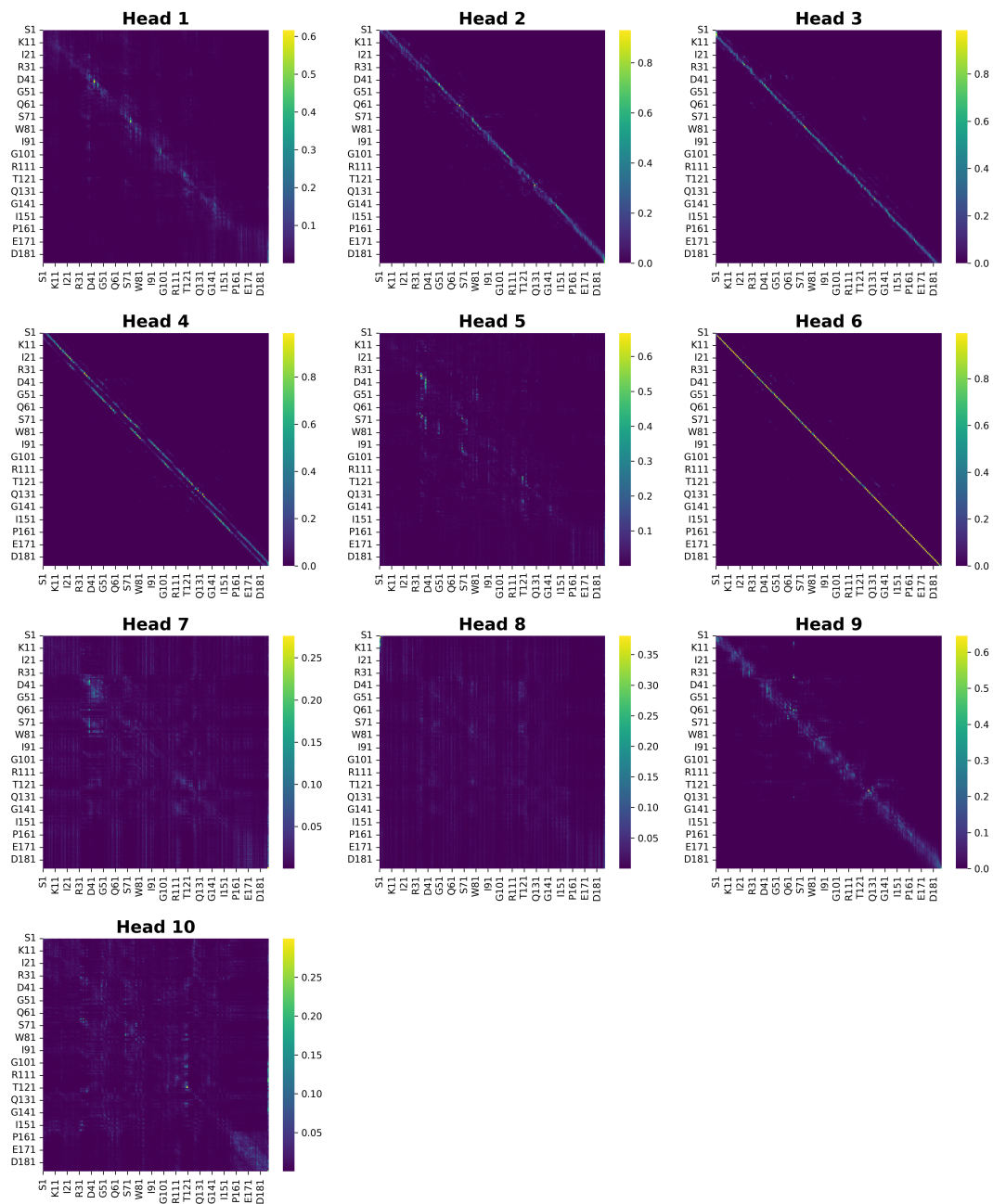


Figure 3.16: The attention maps of from the last layer of ESM2_35M model for AhpC protein (head 1-10)

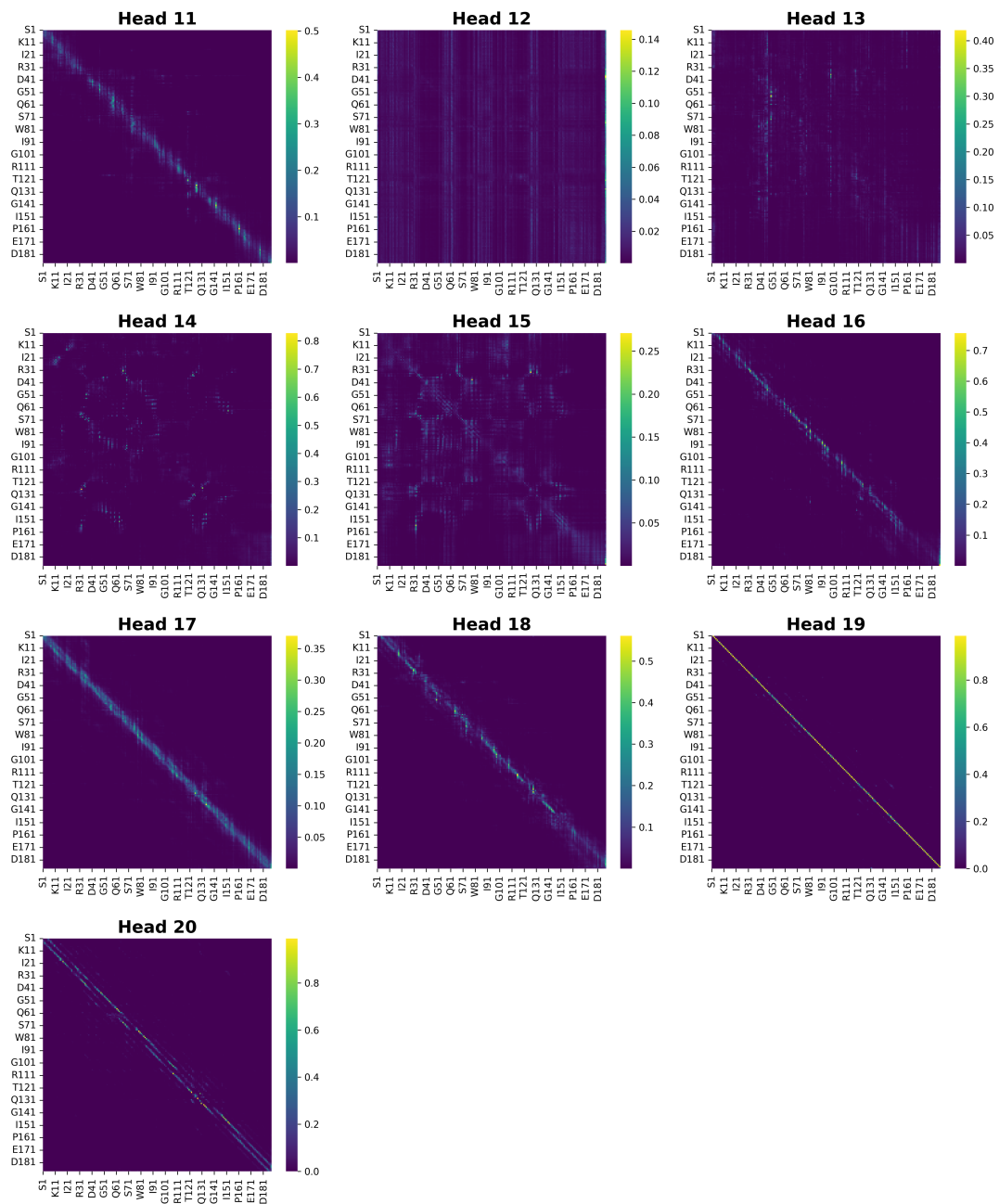


Figure 3.17: The attention maps of the last layer of ESM2_35M model for AhpC protein (head 11-20)

Table 3.6: The information of protein language model used in this study

Model	Abbr.	# layers	# params	Dim.	Data set
esm1b_t33_650M_UR50S	ESM1B_650M	33	650M	1280	UR50/S
esm2_t6_8M_UR50D	ESM2_8M	6	8M	320	UR50/D
esm2_t12_35M_UR50D	ESM2_35M	12	35M	480	UR50/D
esm2_t30_150M_UR50D	ESM2_150M	30	150M	640	UR50/D
esm2_t33_650M_UR50D	ESM2_650M	33	650M	1280	UR50/D
esm2_t36_3B_UR50D	ESM2_3B	36	3B	2560	UR50/D
esm2_t48_15B_UR50D	ESM2_15B	48	15B	5120	UR50/D
prot_t5_xl_bfd	PT_BFD	24	~3B	1024	BFD
prot_t5_xl_uniref50	PT_UR	24	~3B	1024	UR50

Table 3.7: The standard pK_a for different residues

Residue	Standard pK_a
Arg	12.5
Glu	4.5
Lys	10.5
Cys	9.0
His	6.5
Asp	3.8
Tyr	10.0
C-ter.	3.2
N-ter.	8.0

Table 3.8: The tuned hyperparameters for the pKALM model and pl models

Model	Hidden state dimension	Number of layers
pKALM	[64, 128, 256, 512]	[1, 2, 3, 4]
peptide pI model	[16, 32, 64, 128]	[1, 2, 3, 4, 5]
protein pI model	[32, 64, 128, 256]	[1, 2, 3, 4, 5]

Table 3.9: The comparison of test RMSE of different models for different residues.

Method	Asp	Glu	His	Lys	Cys	Tyr	C-ter.	N-ter.
PypKa	1.2130	0.8499	1.3601	1.1436	3.3835	1.1068	0.7496	1.2390
PropKa	1.3966	0.9574	1.1254	1.0058	5.2693	0.9896	/	/
PKAI	1.2659	0.9040	1.3081	1.0515	5.1371	1.2156	/	/
PKAI+	1.1244	0.6662	1.0809	1.0347	4.2194	1.0679	/	/
DeepKa	1.1292	0.7014	1.1848	0.9522	/	/	/	/
pKALM	0.6929	0.6641	1.0468	0.7701	2.0984	0.6479	0.8038	0.7419
–pipep	0.7066	0.6615	1.0565	0.7905	2.0854	0.6127	0.7765	0.7208
–piprot	0.7432	0.6580	1.0820	0.7959	1.8850	0.6891	0.8171	0.6382
–pp	0.7297	0.6317	1.0726	0.8030	1.8866	0.6697	0.8131	0.7100

* The rows “–pipep”, “–piprot”, and “–pp” indicate the performance of the pKALM models without peptide pI models, without protein pI models, and without both models, respectively.

Table 3.10: The comparison of test MAE of different models for different residues.

Method	Asp	Glu	His	Lys	Cys	Tyr	C-ter.	N-ter.
PypKa	0.7755	0.6515	0.9030	0.9354	3.3827	0.9904	0.6542	0.9180
PropKa	0.8863	0.7265	0.8865	0.8193	4.9575	0.6833	/	/
PKAI	0.7545	0.6350	0.9365	0.8015	4.8325	1.1533	/	/
PKAI+	0.6511	0.4902	0.7665	0.7980	3.8975	0.8983	/	/
DeepKa	0.7530	0.5306	0.7924	0.6039	/	/	/	/
pKALM	0.4730	0.4702	0.7133	0.5625	1.7706	0.4415	0.6783	0.5745
–pipep	0.4957	0.4543	0.7432	0.5812	1.7462	0.4020	0.6404	0.6266
–piprot	0.5219	0.4595	0.7391	0.5578	1.5620	0.4589	0.6508	0.5278
–pp	0.5054	0.4486	0.7550	0.5868	1.5867	0.4319	0.6564	0.6150

* The rows “–pipep”, “–piprot”, and “–pp” indicate the performance of the pKALM models without peptide pI models, without protein pI models, and without both models, respectively.

Table 3.11: The comparison of test RMSE of different models for different shift ranges, buried/exposed, and major/total residues.

Method	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-	Buried	Exposed	Abundant	Total
PypKa	0.6048	0.7703	1.4103	1.6671	4.2418	1.6453	0.8032	1.1340	1.1936
PropKa	0.6167	0.6986	1.4145	1.4181	1.6909	2.1226	0.9337	1.1617	1.4507
PKAI	0.6015	0.8084	1.7266	2.0137	4.4972	2.1368	0.7266	1.1345	1.4168
PKAI+	0.3953	0.6834	1.4651	1.8030	4.0603	1.7972	0.5862	0.9631	1.1868
DeepKa	0.4699	0.7924	1.3643	1.1713	2.5462	1.3534	0.7036	0.9959	0.9959
pKALM	0.3928	0.7156	0.9439	1.5716	1.9515	1.1580	0.6281	0.7740	0.8321
–pipep	0.3748	0.6980	0.9424	1.8229	1.8591	1.1682	0.6215	0.7822	0.8355
–piprot	0.3613	0.6866	0.9352	1.6556	2.0967	1.1722	0.6165	0.7995	0.8392
–pp	0.4017	0.6496	0.9549	1.6373	2.0386	1.1434	0.6295	0.7860	0.8288

* The rows “–pipep”, “–piprot”, and “–pp” indicate the performance of the pKALM models without peptide pI models, without protein pI models, and without both models, respectively.

Table 3.12: The comparison of test MAE of different models for different shift ranges, buried/exposed, and major/total residues.

Method	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-	Buried	Exposed	Abundant	Total
PypKa	0.4719	0.7045	1.2493	1.5622	4.0462	1.2108	0.6134	0.7792	0.8294
PropKa	0.4365	0.6038	1.2937	1.2925	1.6650	1.3886	0.6928	0.8243	0.9388
PKAI	0.4549	0.6585	1.5575	1.7850	4.2850	1.4359	0.5501	0.7515	0.8844
PKAI+	0.3077	0.6015	1.3913	1.7550	3.9950	1.2387	0.4277	0.6322	0.7347
DeepKa	0.3607	0.6545	1.2728	1.0341	2.4758	0.8363	0.5223	0.6682	0.6682
pKALM	0.2726	0.5821	0.8367	1.5696	1.8849	0.8126	0.4551	0.5291	0.5644
–pipep	0.2474	0.5660	0.8045	1.8224	1.7903	0.8278	0.4553	0.5388	0.5708
–piprot	0.2465	0.5587	0.7818	1.6538	2.0343	0.8412	0.4472	0.5459	0.5722
–pp	0.2678	0.5370	0.8132	1.6209	1.9923	0.8074	0.4671	0.5429	0.5716

* The rows “–pipep”, “–piprot”, and “–pp” indicate the performance of the pKALM models without peptide pI models, without protein pI models, and without both models, respectively.

References

- [1] Jeremy R Knowles and William P Jencks. The intrinsic pK_a -values of functional groups in enzymes: Improper deductions from the ph-dependence of steady-state parameter. *CRC Crit. Rev. Biochem.*, 4(2):165–173, 1976.
- [2] Thomas K Harris and George J Turner. Structural basis of perturbed pK_a values of catalytic groups in enzyme active sites. *IUBMB Life*, 53(2):85–98, 2002.
- [3] Effie C Kisgeropoulos, Vivek S Bharadwaj, David W Mulder, and Paul W King. The contribution of proton-donor pK_a on reactivity profiles of [feFe]-hydrogenases. *Front. Microbiol.*, 13: 903951, 2022.
- [4] An-Suei Yang and Barry Honig. On the ph dependence of protein stability. *J. Mol. Biol.*, 231(2):459–474, 1993.
- [5] Yee-Joo Tan, Mikael Oliveberg, Ben Davis, and Alan R Fersht. Perturbed pK_a -values in the denatured states of proteins. *J. Mol. Biol.*, 254(5):980–992, 1995.
- [6] John Bradley, Fergal O’Meara, Damien Farrell, and Jens Erik Nielsen. Highly perturbed pK_a values in the unfolded state of hen egg white lysozyme. *Biophys. J.*, 102(7):1636–1645, 2012.
- [7] Alexey V Onufriev and Emil Alexov. Protonation and pK changes in protein–ligand binding. *Q. Rev. Biophys.*, 46(2):181–209, 2013.
- [8] Robert L Jones and W David Wilson. Effect of ionic strength on the pK_a of ligands bound to dna. *Biopolymers: Original Research on Biomolecules*, 20(1):141–154, 1981.
- [9] Zeynep Pinar Haslak, Sabrina Zareb, Ilknur Dogan, Viktorya Aviyente, and Gerald Monard. Using atomic charges to describe the pK_a of carboxylic acids. *J. Chem. Inf. Model.*, 61(6): 2733–2743, 2021.
- [10] Delphine C Bas, David M Rogers, and Jan H Jensen. Very fast prediction and rationalization of pK_a values for protein–ligand complexes. *Proteins: Struct., Funct., Bioinf.*, 73(3):765–783, 2008.

- [11] Mats HM Olsson, Chresten R Søndergaard, Michal Rostkowski, and Jan H Jensen. Propka3: consistent treatment of internal and surface residues in empirical pK_a predictions. *J. Chem. Theory Comput.*, 7(2):525–537, 2011.
- [12] Michael S Lee, Freddie R Salsbury Jr, and Charles L Brooks III. Constant-ph molecular dynamics using continuous titration coordinates. *Proteins: Struct., Funct., Bioinf.*, 56(4):738–752, 2004.
- [13] Jason A Wallace and Jana K Shen. Predicting pK_a values with continuous constant ph molecular dynamics. In *Methods in enzymology*, volume 466, pages 455–475. Elsevier, 2009.
- [14] Sarah L Williams, César Augusto F de Oliveira, and J Andrew McCammon. Coupling constant ph molecular dynamics with accelerated molecular dynamics. *J. Chem. Theory Comput.*, 6(2):560–568, 2010.
- [15] Wei Chen, Brian H Morrow, Chuanyin Shi, and Jana K Shen. Recent development and application of constant ph molecular dynamics. *Mol. Simul.*, 40(10-11):830–838, 2014.
- [16] Robert C Harris, Ruibin Liu, and Jana Shen. Predicting reactive cysteines with implicit-solvent-based continuous constant ph molecular dynamics in amber. *J. Chem. Theory Comput.*, 16(6):3689–3698, 2020.
- [17] Jens E Nielsen and Gerrit Vriend. Optimizing the hydrogen-bond network in poisson–boltzmann equation-based pK_a calculations. *Proteins: Struct., Funct., Bioinf.*, 43(4):403–412, 2001.
- [18] Jim Warwicker. pK_a predictions with a coupled finite difference poisson–boltzmann and debye–hückel method. *Proteins: Struct., Funct., Bioinf.*, 79(12):3374–3380, 2011.
- [19] Pedro BPS Reis, Diogo Vila-Vicosa, Walter Rocchia, and Miguel Machuqueiro. Pypka: A flexible python module for poisson–boltzmann-based pK_a calculations. *J. Chem. Inf. Model.*, 60(10):4442–4448, 2020.
- [20] Jiahui Chen, Jingzhen Hu, Yongjia Xu, Robert Krasny, and Weihua Geng. Computing protein pK_a s using the tabi poisson–boltzmann solver. *J. Comput. Biophys. Chem.*, 20(02):175–187, 2021.
- [21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Židek, Anna Potapenko, et al.

- Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.
- [22] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.
 - [23] Zhitao Cai, Tengzi Liu, Qiaoling Lin, Jiahao He, Xiaowei Lei, Fangfang Luo, and Yandong Huang. Basis for accurate protein pK_a prediction with machine learning. *J. Chem. Inf. Model.*, 63(10):2936–2947, 2023.
 - [24] Johnny Habchi, Peter Tompa, Sonia Longhi, and Vladimir N Uversky. Introducing protein intrinsic disorder. *Chem. Rev.*, 114(13):6561–6588, 2014.
 - [25] Hatice Gokcan and Olexandr Isayev. Prediction of protein pK_a with representation learning. *Chem. Sci.*, 13(8):2462–2474, 2022.
 - [26] Sergiy O Garbuzynskiy, Bogdan S Melnik, Michail Yu Lobanov, Alexei V Finkelstein, and Olexana V Galzitskaya. Comparison of x-ray and nmr structures: is there a systematic difference in residue contacts between x-ray-and nmr-resolved protein structures? *Proteins: Struct., Funct., Bioinf.*, 60(1):139–147, 2005.
 - [27] Leo Breiman. Random forests. *Mach. Learn.*, 45:5–32, 2001.
 - [28] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2008.
 - [29] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.
 - [30] Ada Y Chen, Juyong Lee, Ana Damjanovic, and Bernard R Brooks. Protein pK_a prediction by tree-based machine learning. *J. Chem. Theory Comput.*, 18(4):2673–2686, 2022.
 - [31] Zhitao Cai, Fangfang Luo, Yongxian Wang, Enling Li, and Yandong Huang. Protein pK_a prediction with machine learning. *ACS omega*, 6(50):34823–34831, 2021.
 - [32] Pedro BPS Reis, Marco Bertolini, Floriane Montanari, Walter Rocchia, Miguel Machuqueiro, and Djork-Arné Clevert. A fast and interpretable deep learning approach for accurate electrostatics-driven pK_a predictions in proteins. *J. Chem. Theory Comput.*, 18(8):5068–5078, 2022.

- [33] Swagata Pahari, Lexuan Sun, and Emil Alexov. Pkad: a database of experimentally measured pK_a values of ionizable groups in proteins. *Database*, 2019:baz024, 2019.
- [34] Nicolas Ancona, Ananta Bastola, and Emil Alexov. Pkad-2: New entries and expansion of functionalities of the database of experimentally measured pK_a 's of proteins. *J. Comput. Biophys. Chem.*, 22(05):515–524, 2023.
- [35] Pedro BPS Reis, Djork-Arné Clevert, and Miguel Machuqueiro. pkpdb: a protein data bank extension database of pK_a and pI theoretical values. *Bioinformatics*, 38(1):297–298, 2022.
- [36] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci.*, 118(15):e2016239118, 2021.
- [37] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
- [38] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, 16(12):1315–1322, 2019.
- [39] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10):7112–7127, 2021.
- [40] Shijie Xu and Akira Onoda. Accurate and fast prediction of intrinsically disordered protein by multiple protein language models and ensemble learning. *J. Chem. Inf. Model.*, 64(7):2901–2911, 2023.
- [41] Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.
- [42] Lukasz Pawel Kozlowski. Ipc 2.0: prediction of isoelectric point and pK_a dissociation constants. *Nucleic Acids Res.*, 49(W1):W285–W292, 2021.

- [43] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.
- [45] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.
- [47] Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Henry Chao, Li Chen, Paul A Craig, Gregg V Crichlow, Kenneth Dalenberg, Jose M Duarte, et al. Rcsb protein data bank (rcsb.org): delivery of experimentally-determined pdb structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic Acids Res.*, 51(D1):D488–D508, 2023.
- [48] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold: making protein folding accessible to all. *Nat. Methods*, 19(6):679–682, 2022.
- [49] Peter Eastman, Jason Swails, John D Chodera, Robert T McGibbon, Yutong Zhao, Kyle A Beauchamp, Lee-Ping Wang, Andrew C Simmonett, Matthew P Harrigan, Chaya D Stern, et al. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput. Biol.*, 13(7):e1005659, 2017.
- [50] James A Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E Hauser, and Carlos Simmerling. ff14sb: improving the accuracy of protein side chain and backbone parameters from ff99sb. *J. Chem. Theory Comput.*, 11(8):3696–3713, 2015.
- [51] Benjamin Webb and Andrej Sali. Comparative protein structure modeling using modeller. *Curr. Protoc. Bioinform.*, 54(1):5–6, 2016.
- [52] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL <https://www.wandb.com/>. Software available from wandb.com.

- [53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [54] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.
- [55] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. <https://github.com/huggingface/accelerate>, 2022.
- [56] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.
- [57] Henry N Po and NM Senozan. The henderson-hasselbalch equation: its history and limitations. *J. Chem. Educ.*, 78(11):1499, 2001.
- [58] Bengt Bjellqvist, Bodil Basse, Eydfinnur Olsen, and Julio E Celis. Reference points for comparisons of two-dimensional maps of proteins from different human cell types defined in a ph scale where isoelectric points correlate with polypeptide compositions. *Electrophoresis*, 15(1): 529–539, 1994.
- [59] Yasuhiko Nozaki and Charles Tanford. The solubility of amino acids and two glycine peptides in aqueous ethanol and dioxane solutions: establishment of a hydrophobicity scale. *J. Biol. Chem.*, 246(7):2211–2217, 1971.
- [60] David L Tabb, W Hayes McDonald, and John R Yates. Dtaselect and contrast: tools for assembling and comparing protein identifications from shotgun proteomics. *J. Proteome Res.*, 1(1): 21–26, 2002.
- [61] Richard L Thurlkill, Gerald R Grimsley, J Martin Scholtz, and C Nick Pace. pk values of the ionizable groups of proteins. *Protein Sci.*, 15(5):1214–1218, 2006.
- [62] Antonio Sillero and João Meireles Ribeiro. Isoelectric points of proteins: theoretical determination. *Anal. Biochem.*, 179(2):319–325, 1989.

- [63] Rex Malcolm Chaplin Dawson, Daphne C Elliott, William H Elliott, and Kenneth Milner Jones. Data for biochemical research. 1959.
- [64] Gerald R Grimsley, J Martin Scholtz, and C Nick Pace. A summary of the measured pK values of the ionizable groups in folded proteins. *Protein Sci.*, 18(1):247–251, 2009.
- [65] Lukasz P Kozlowski. Ipc–isoelectric point calculator. *Biol. Direct*, 11:1–16, 2016.
- [66] Brian D Halligan. Promost: A tool for calculating the pI and molecular mass of phosphorylated and modified proteins on two-dimensional gels. *Phospho-Proteomics: Methods and Protocols*, pages 283–298, 2009.
- [67] Christopher P Toseland, Helen McSparron, Matthew N Davies, and Darren R Flower. Ppd v1.0—an integrated, web-accessible database of experimentally determined protein pK_a values. *Nucleic Acids Res.*, 34(suppl_1):D199–D203, 2006.
- [68] John D Rodwell. Heterogeneity of component bands in isoelectric focusing patterns. *Anal. Biochem.*, 119(2):440–449, 1982.
- [69] Zhitao Cai, Hao Peng, Shuo Sun, Jiahao He, Fangfang Luo, and Yandong Huang. Deepka web server: High-throughput protein pK_a prediction. *J. Chem. Inf. Model.*, 64(8):2933–2940, 2024.
- [70] Normann Spitzner, Frank Löhr, Stefania Pfeiffer, Assen Koumanov, Andrey Karshikoff, and Heinz Rüterjans. Ionization properties of titratable groups in ribonuclease T₁. pK_a values in the native state determined by two-dimensional heteronuclear nmr spectroscopy. *Eur. Biophys. J.*, 30:186–197, 2001.
- [71] Kimberly J Nelson, Derek Parsonage, Andrea Hall, P Andrew Karplus, and Leslie B Poole. Cysteine pK_a values for the bacterial peroxiredoxin ahpc. *Biochemistry*, 47(48):12860–12868, 2008.
- [72] Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. Bertology meets biology: Interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.

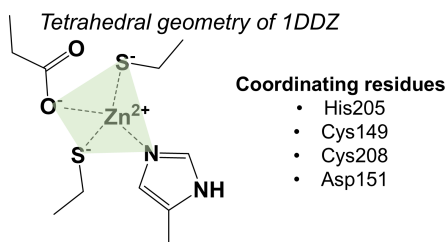
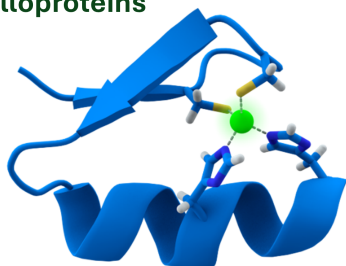
4

Metal-binding protein identification using protein language model and deep representations

4.1 INTRODUCTION

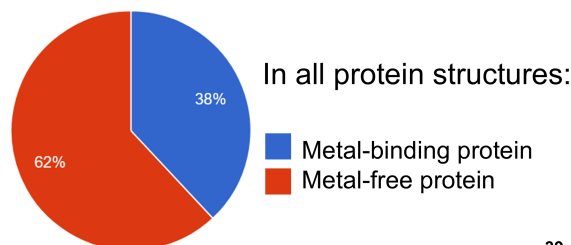
Metalloproteins play a central role in various biological processes such as structure stabilization, catalysis, electron transfer and signal transduction (Figure 4.1). Metalloproteins account for about one-third of the protein structures deposited in the Protein Data Bank (PDB) [1], with enormously diverse functions. The lack of metal ions can lead to critical human diseases. For example, zinc-binding proteins for the immune response would lose their functions if zinc deficiency occurs during aging.

Metalloproteins



Protein binds many metal ions

Ion	Sites	Percentage (%)
Zn ²⁺	64191	33.45
Mg ²⁺	47884	24.95
Ca ²⁺	43469	22.65
Mn ²⁺	12663	6.6
Fe ³⁺	7442	3.88
Cu ²⁺	6493	3.38



39

Figure 4.1: Metalloproteins are proteins that contain metal ions. They play a central role in various biological processes such as structure stabilization, catalysis, electron transfer and signal transduction. Metalloproteins account for about one-third of the protein structures deposited in the Protein Data Bank (PDB) including different types of metal ions such as Zn²⁺, Ca²⁺, Mg²⁺, Mn²⁺, Fe³⁺, Cu²⁺, and Co²⁺.

Metalloproteins are also important in industrial applications. Metalloenzymes use protein as a scaffold, incorporating metal ions to catalyze reactions with desired stereo-selectivity and efficiency. The binding of metal ions to proteins can be promiscuous, inhibiting one another. If ions compete with essential metal ions, this can lead to toxicity in cells. Calcium-binding proteins can also be used as sensors to detect the concentration of calcium ions in the environment [2, 3], and uranium-binding proteins can be used to depollute uranium-contaminated water [4]. His-tagged proteins are widely used in protein purification and detection by exploiting the high affinity binding of histidine to Ni²⁺ ions [5], where the six histidine residues are often located in the N-terminus and/or C-terminus of the protein. The phylogeny of metalloproteins may reflect changes in metal abundances on Earth over geological time scales [6].

As simple metal ion binding sites can be relatively easily implemented on the surface of proteins through the coordination of specific amino acids (i.e. the nitrogen atoms of histidine side chains, the sulfur atoms of cysteine side chains), the identification of metal binding sites inside proteins is more difficult, especially for deeply buried regions. Metal ions are often added to the buffer solution to stabilize the structure of proteins in X-ray crystallography, which can be misidentified as metal binding sites.

Traditional molecular dynamics (MD) simulations are usually difficult to use with transition metals

due to the lack of accurate force field parameters. While QM/MM simulations are computationally expensive and time consuming, it is important to accurately and efficiently identify the metal-binding site in proteins for both biological and experimental purposes [7].

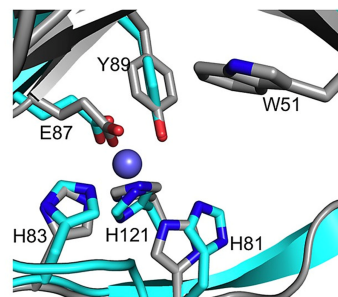
A number of tools have been developed to predict metal-binding proteins [8, 9, 10, 11, 12]. Non-deep learning methods often use templates of homologous proteins with known metal-binding sites to predict the binding sites of the query protein (Figure 4.2). Deep learning-based methods can be divided into sequence-based and structure-based methods (Figure 4.3). Their input is usually the sequence and/or structure of the protein. Structure-based methods use the three-dimensional structure of proteins and attempt to identify the metal-binding sites from the geometric and physicochemical properties, while sequence-based methods rely on sequence pattern recognition. The structure-based methods often provide geometric predictions, such as the positions of metal ions in the protein, while the sequence-based methods are more likely to predict the binding residues alone. In addition, the sparsity of sequence patterns restrict the accuracy of sequence-based methods, as they often miss the actual binding sites while the structure-based methods often overestimate the number of binding sites due to the plausible geometric positions of the metal ions.

Sequence motifs

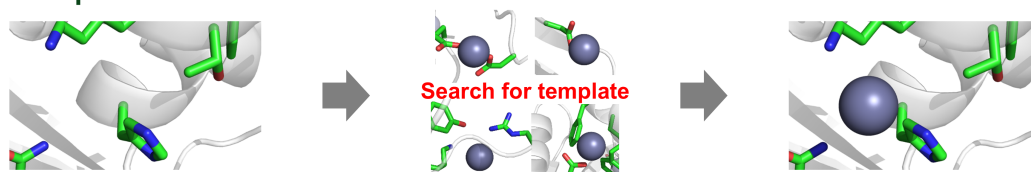
Cupin motif 1													Cupin motif 2																																													
G		H		H		E		G				G		P		H		N																																								
W1	---	F	G	P	G	H	O	L	R	P	R	T	---	P	P	F	L	G	L	E	---	S	G	I	V	T	I	D	---	E	I	R	---	A	C	V	A	L	I	P	G	D	A	E	---	G	T	V	A	---	P	---						
W2	---	F	E	P	H	G	R	L	P	R	D	---	P	P	E	F	L	G	L	E	---	S	G	V	V	T	I	D	---	E	I	R	---	F	C	V	A	I	V	P	A	N	A	E	---	D	T	Q	A	---	P	---						
Q1	---	M	P	A	G	L	I	Y	P	F	Q	P	---	A	E	E	I	F	I	L	A	---	E	A	E	F	L	M	E	---	R	R	L	G	F	G	D	H	V	F	H	P	S	G	H	P	---	A	T	R	Y	D	R	---	P	---		
Q2	---	W	G	A	G	L	D	Y	D	W	S	Q	---	A	E	E	L	Y	L	T	L	A	---	G	A	V	F	K	V	D	---	A	F	V	G	A	E	G	T	R	L	H	A	S	W	Q	S	---	A	M	S	T	G	D	Q	---	P	---
L1	---	F	A	P	G	C	T	Y	P	A	A	---	K	G	I	T	E	S	Y	V	C	L	S	---	A	V	S	E	---	G	V	Y	V	F	G	S	M	I	F	N	P	P	E	H	L	---	R	I	T	V	G	D	R	E	---	P	---	
L2	---	F	A	P	S	T	T	Y	P	Q	S	---	K	D	I	E	S	Y	I	S	V	A	---	A	W	S	E	---	A	V	H	A	F	G	S	L	I	L	N	R	P	G	L	E	---	R	I	T	T	G	D	L	S	---	P	---		
L3	---	F	A	P	R	T	T	Y	P	Q	S	---	S	E	I	E	S	Y	I	S	V	S	---	A	W	S	E	---	A	V	Y	A	F	G	S	L	I	L	N	R	P	G	D	E	---	R	I	T	T	G	D	Q	E	---	P	---		
L4	---	F	A	P	R	T	T	Y	P	A	A	---	S	G	I	T	E	S	Y	I	C	L	S	---	A	T	S	E	---	G	V	Y	V	F	G	A	M	I	L	N	V	P	E	H	D	---	A	I	T	T	S	D	R	E	---	P	---	

Cupin motifs and metal binding residues of DddW.

Brummett, Adam E., et al. *PLoS one* 10.5 (2015): e0127288.

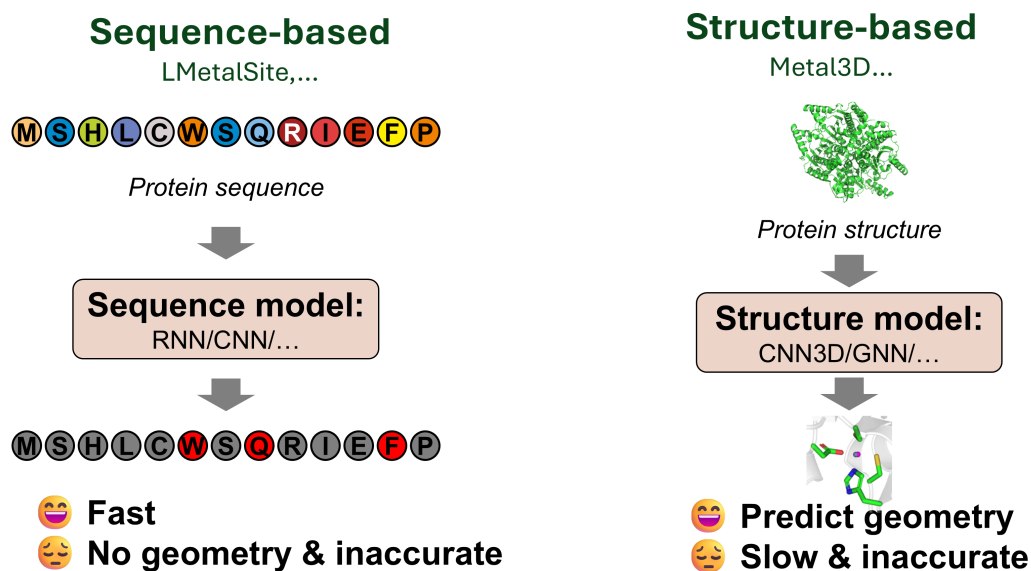


Templates



Chih-Hao Lu et al. *Bioinformatics* 38.18 (2022): 4428-4429

Figure 4.2: Motif- and template-based prediction search for homologous proteins with known metal-binding sites. The sequence of the query protein is aligned to the template, and the metal-binding sites are predicted based on the alignment. The template-based method is limited by the availability of homologous proteins with known metal-binding sites.



[1] Simon L. Dürr *et al. Nat. Commun.* 14.1 (2023): 2713.

Qianmu Yuan *et al. Briefings Bioinf.* 23.6 (2022): bbac444.

Figure 4.3: Deep learning-based metal binding prediction methods are either sequence-based or structure-based. Sequence-based methods use the sequence of the protein to predict the metal-binding residues. Structure-based methods use the three-dimensional structure of the protein to predict the metal-binding sites. They have their own advantages and disadvantages. Sequence-based methods are computationally efficient but often miss the actual binding sites, while structure-based methods are computationally expensive and often overestimate the number of binding sites.

In this study, we present a two-step framework that combines sequence and structural information to predict protein metal-binding sites. The framework uses the protein structure as input, from which sequence information is derived, and outputs the predicted metal-binding positions. It operates as follows: (1) a sequence-based predictor, built on a protein language model, identifies binding residues; (2) a structure-based predictor, using a pre-trained three-dimensional residual network [13], determines the geometry of the binding sites. The first step is designed to reduce the number of false positives (residues predicted as binding sites but are not) in a computationally efficient manner, which is a disadvantage of many existing structure-based methods. The second step is designed to refine the results from the first step, further reducing the number of missed binding sites. Testing reveals that our method significantly outperforms current approaches, particularly in F1 score and root mean square error (RMSE). PRIME is implemented in a metal-wise manner, and the current version supports the most common ions such as Zn^{2+} , Ca^{2+} , Mg^{2+} , Mn^{2+} , Fe^{3+} , Cu^{2+} , and Co^{2+} .

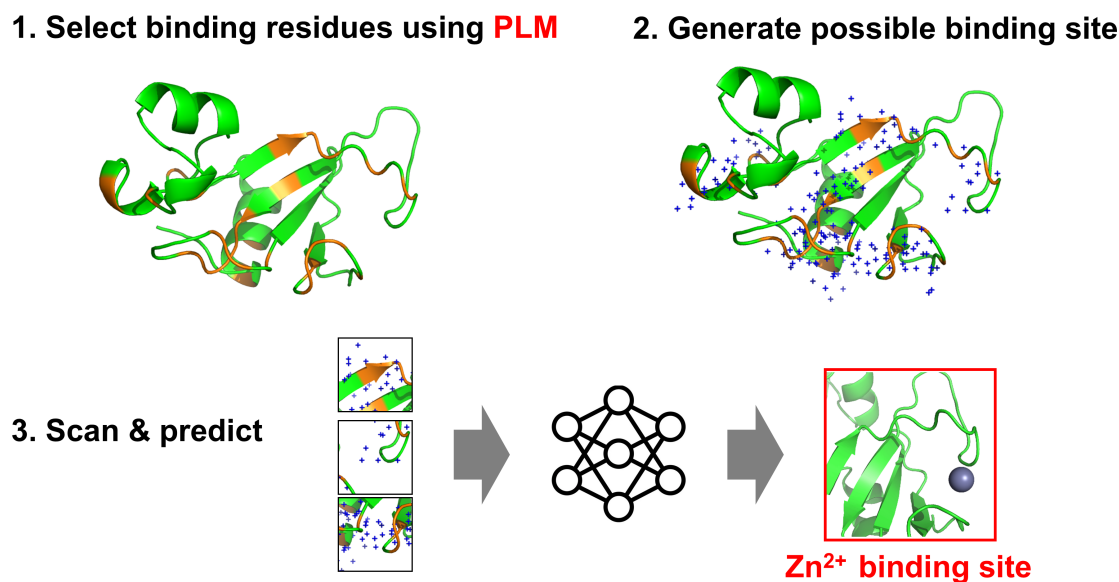


Figure 4.4: This work proposed a two-step framework combining sequence and structural information to predict protein metal-binding sites.

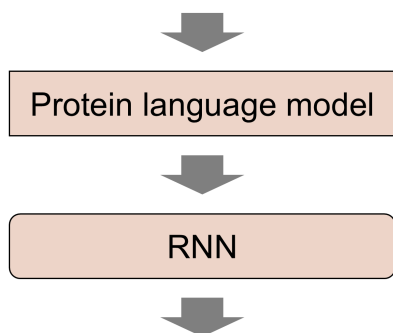
4.2 RESULTS

4.2.1 PRIME

PRIME (PRObe-based Identification of Metal binding sites of proteins using deep representations) is proposed as shown in Figure 4.5. It takes the coordinates and element type of the atoms (i.e., C/N/O/S) as input and outputs the predicted positions of the metal binding sites. First, the sequence is extracted from the structure for each chain of the protein. Missing residues are ignored and non-standard residues are replaced by the ‘X’ symbol. A protein language model (*ESM2_t33_650M_UR50D*) is used to encode the chains into high-dimensional representations. A bidirectional gated recurrent unit (BiGRU) network is used to extract the sequence features and predict the binding residues. Possible binding positions, called “probes” in this work, are generated by scanning the space around binding residues on the three-dimensional protein structures. The structures centered on the probes with a side length of 10Å are sliced and fed into a three-dimensional residual network (ResNet3D) to encode the structure features, which are then fed into two fully connected (FC) layers to predict 1) whether the probe is a binding site or not. If so, 2) the predicted positional offset is used to refine the position of the probe to obtain the final prediction. The probes are then clustered to remove the overlapping probes. The final prediction is the one with the highest probability in each cluster.

Sequence model

Input:
MMTLNLQTMSSNDMLNFGKKPAYTTSN.....



Output:
MMTLNLQTMSSNDMLNFGKKPAYTTSN.....

Structure model

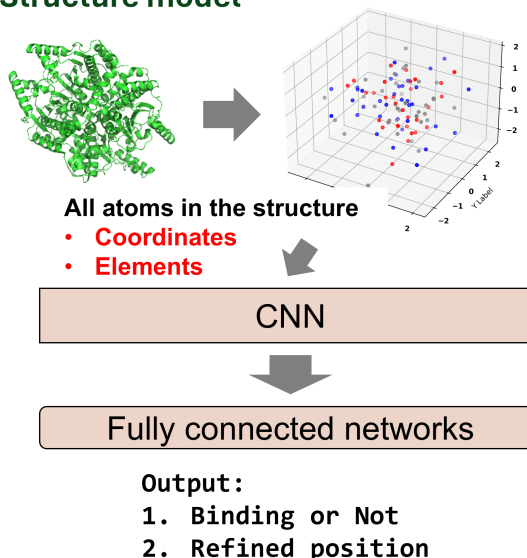


Figure 4.5: The sequence model of PRIME predicts the potentially binding residues of the input sequences. The structure model of PRIME takes the structures around the probes as input and predicts whether the probe is a binding site or not. If it is, the predicted position offset is used to refine the position of the probe to obtain the predicted positions of the metal ions.

Here we show an example of the prediction of PRIME in Figure 4.6. The predicted positions match the actual binding sites of the protein. The predicted positions are shown in red and the actual positions are shown in blue. The results suggest that PRIME can accurately predict the metal-binding sites of proteins.

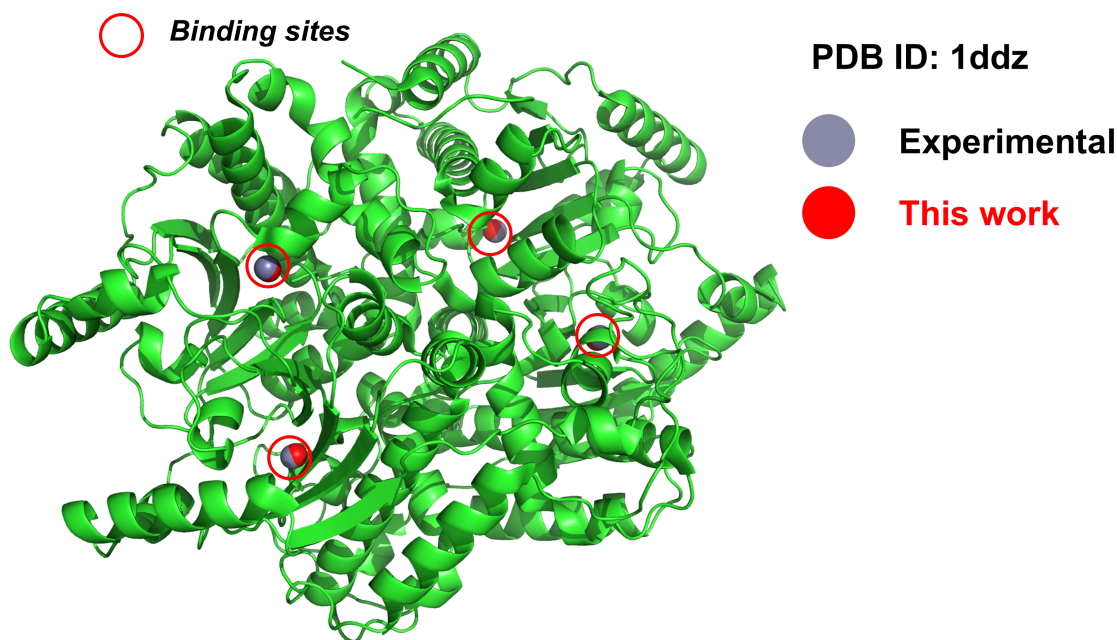


Figure 4.6: The first example of PRIME. The prediction match the actual binding sites of the protein. The predicted positions are shown in red and the actual positions are shown in blue.

4.2.2 COMPARISON TO METAL3D

We compare the performance of PRIME with the state of the art Metal3D [7] on the test set. The results are shown in the table 4.1. PRIME outperforms Metal3D in terms of area under the receiver operating characteristic curve (AUC), F1 score, Matthews correlation coefficient (MCC) and root mean square error (RMSE) for all ion types. The results suggest that PRIME is more accurate and robust than Metal3D in predicting protein metal binding sites. Note that for some proteins, zinc ions are introduced in their structures as a tool for crystallization, which are not the real binding sites in biological functions and are not considered in the evaluation.

Table 4.1: Comparison of PRIME and Metal3D on the Zn^{2+} test set. Note that we choose 3Å as the threshold to check if the prediction is correct or not.

	TP	FP	FN	Pre	Rec	F1	RMSE	Running time (s)
PRIME	167	12	26	0.9330	0.8653	0.8978	0.3393	19.93
Metal3D	174	117	11	0.5979	0.9405	0.7311	0.4677	78.51

PRIME predicts more accurately than Metal3D in terms of F1 score and RMSE. Metal3D, as a deep

learning method based on structures alone, predicts too many false positives (117 vs 12) but only similar true positives (174 vs 167) compared to PRIME. For instance, the prediction of protein (PDB ID: 1tlj) show that PRIME predicts the actual binding sites of the protein while Metal3D predicts too many false positives (Figure 4.7). Metal3D scans all residues in the protein structure with a structure-based model, which limits its efficiency. The results suggest that sequence information is important for predicting protein metal-binding sites because PRIME can effectively use the evolutionary information of the proteins encoded in the sequences, which is also very important for reducing the number of false positives.

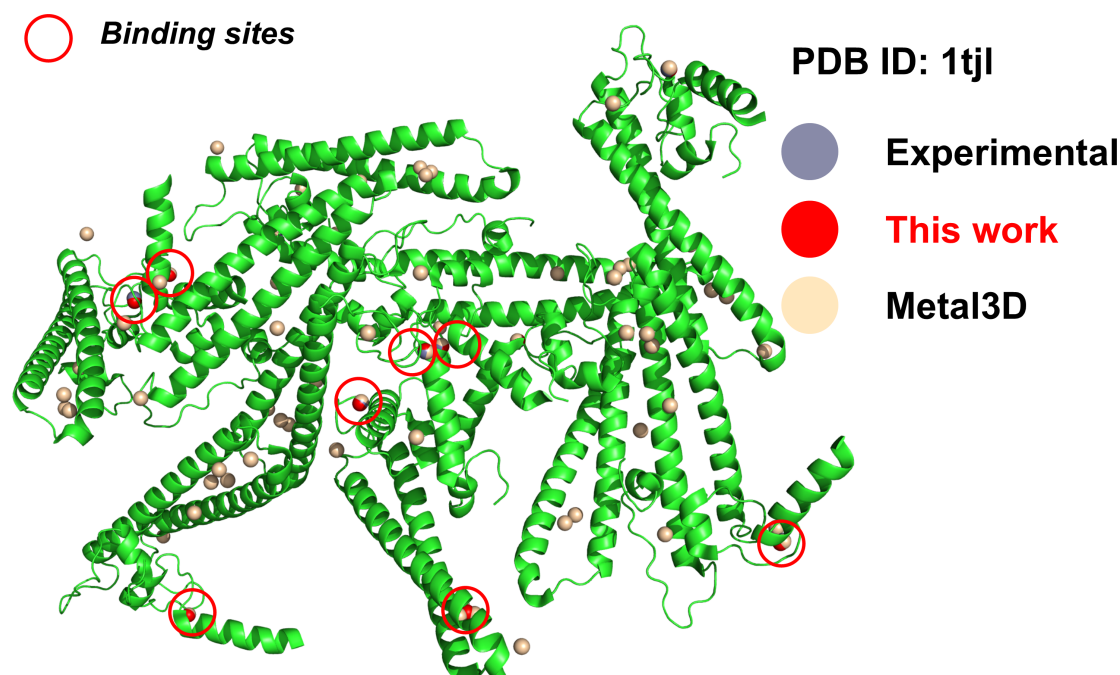


Figure 4.7: The second example of PRIME. The prediction match the actual binding sites of the protein while Metal3D predicts too many false positives. The predicted positions are shown in red and the actual positions are shown in blue.

4.3 METHOD

4.3.1 DATASET

We constructed a dataset based on the RCSB protein database [1] (downloaded 22 October 2024) and BioLip2 [14] (released 28 September 2024). All chains annotated by BioLip2 are clustered at 30% identity by mmseq2 [15] and the chains with more than 2048 residues are removed. Any two structures

have no chains with more than 30% sequence identity in the cleaned dataset. Only the single atom ligand is considered. We have removed all atoms in the PDB structure except carbon, nitrogen, oxygen and sulfur in the amino acid residues. If there were multiple models in the PDB structures, we took the first one. To train the sequence-based predictor, we used all chains regardless of ion type. The structure-based predictor was trained using metal ions of Zn^{2+} , Ca^{2+} , Mg^{2+} , Mn^{2+} , Fe^{3+} , Cu^{2+} , and Co^{2+} . The statistics of the data sets are shown in Figure [xxx]. The data set contains 6,694 chains, randomly divided into 6,000/294/400 for training, validation and testing.

For pre-training, only high-resolution (less than 2.0 Å) crystallized structures from the RCSB protein database are used. They are clustered with the same setting as above. 1,000 structures are randomly selected for validation and others for training.

4.3.2 MASKED STRUCTURE MODELING

We used a masked reconstruction method to pre-train models, which has been used successfully in various applications. The model takes the input of the surrounding atoms and predicts the masked residues.

We use a three-dimensional residue network (ResNet3D) as the backbone for structure pre-training. Although recent advances in equivariance modeling (i.e. equivariant neural networks) are promising, the huge computational cost hinders their applications and currently only residue-level equivariant neural networks are available. Instead, we used ResNet3D because they are already widely used and very successful in protein structure prediction.

4.3.3 MODEL TRAINING

We have implemented the model using *PyTorch 2.3.1*. High-dimensional representations are extracted from ESM2 models using the *transformers* package and fine-tuned with additional bidirectional LSTM layers and fully connected layers. Three-dimensional ResNet is used as the backbone for structure pre-training and fine-tuning on metalloproteins. A workstation with an AMD Ryzen 9 7950X3D 16-core processor, an NVIDIA 4090 GPU and a 3090Ti GPU is used for training, and a single 4090 GPU is used for inference. We used cross-entropy loss for classification tasks and smooth L_1 loss [16] for regression tasks, respectively. The model was trained using the AdamW optimizer [17] with a learning rate of 10^{-4} and a batch size of 32.

4.3.4 BINDING RESIDUE SCANNING

We used a sequence-based predictor to efficiently and accurately detect the possible binding residues. To obtain high recall scores, we set the threshold to 0.05 after training. Scanning is performed on each chain of the protein structure except the nucleotide acid chains. If the scanned binding residues are less than a certain number with respect to the chain length, we sort the residues by probability and select the top ones on the chain; if the number is more than 100, we select on the top 100 residues. We analyze the number of binding residues with respect to chain length in the training and validation sets (Figure 4.8). We found that the maximum number of binding residues is positively correlated with chain length and should be fit by a power function. We fit the power function $y = ax^c + b$ with unknown parameters a , c and b using the L-BFGS-B algorithm in the `SCIPY.OPTIMIZE.MINIMIZE` function.

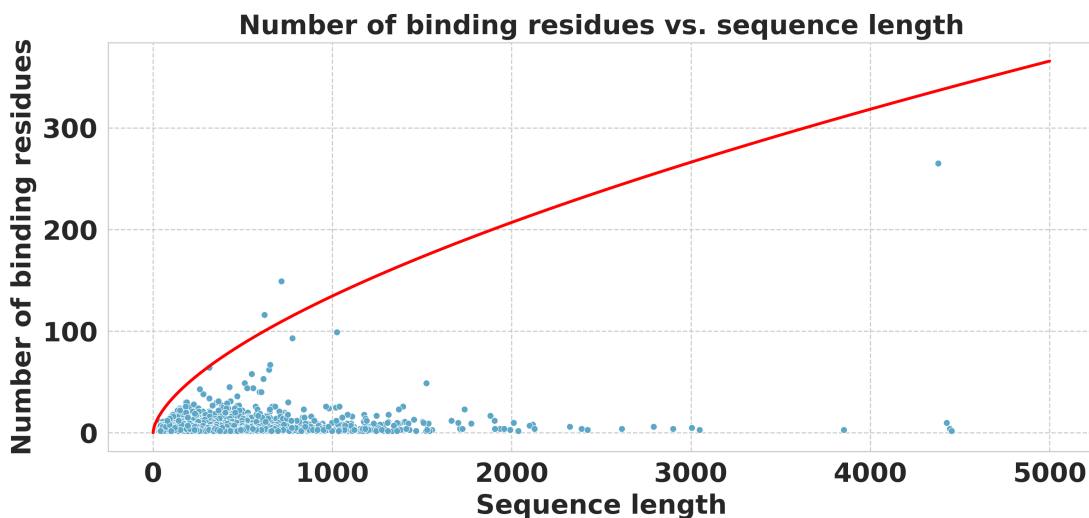


Figure 4.8: Relationship between maximum number of binding residues and chain length. The maximum number of binding residues is positively correlated with the chain length. We use a power function to fit the data (red curve): $y = 1.84 * x * 0.621 + 0.0627$.

4.3.5 PROBE GENERATION

The probes are generated by detecting the surrounding position at 0.5 intervals of the nitrogen, oxygen and sulfur atoms of the binding residues. The detection is based on distance. We first compute the statistics of the distances of the pairs of ions and N/O/S atoms (Figure 4.9). At present we do not consider carbon bonds because they are two weak ones. For each pair we calculate the mean distance

μ and the standard deviation δ . The lower and upper bounds of the distance for probe generation are set to $\mu - 3\delta$ and $\mu + 3\delta$ respectively. Note that for the pairs with less than 10 samples, or less than 1% in the same ion type, we ignore the statistics here and do not show them in the figure 4.9. The lower and upper bounds are set to 1.5 Å and 3.0 Å respectively. We also remove the probes that are less than 1.0 Å away from any atoms in the protein structure.

After generation, we cluster the probes with their possible coordinating atoms. For each cluster, we keep only the probes with the minimum distance to all other probes, instead of their geometric center. This is because the probes in a group often lie on the low-dimensional manifold. For example, the probes coordinating to two oxygen atoms of the same carboxyl group are often on the same plane and their geometric center would be very close to the carboxyl group, which is not the real binding site. We collect the probes with the highest coordination number and the highest probability of group, and if any probes are less than 1.5 Å away from previously collected probes, we remove them. In practice, we have observed that a protein of medium size would have about 200 probes after this step and the false negative rate is less than 0.6% for Zn^{2+} ion, which is acceptable.

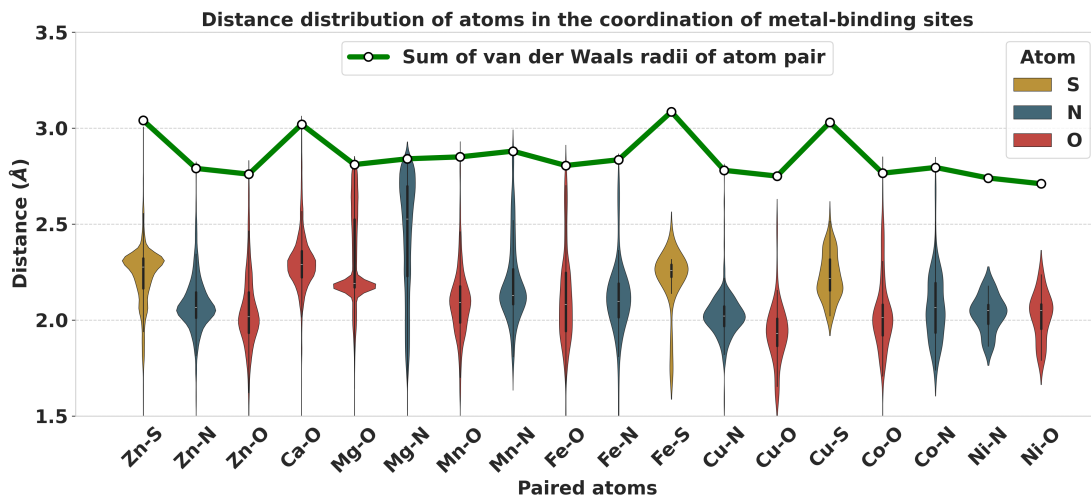


Figure 4.9: Statistics of the distances of ion pairs and N/O/S atoms. The distances are calculated from the van der Waals radii of the atoms. For example, the van der Waals radius of the nitrogen atom is $r_1 = 1.55$ and the van der Waals radius of the Zn^{2+} ion is $r_2 = 0.74$, the maximum pair distance is $r_1 + r_2 + 0.5 = 2.79$.

4.3.6 DATA AUGMENTATION

Three-dimensional CNNs are not rotation-equivalent and are sensitive to the orientation of the input coordinates. Given the small scale of metalloproteins, we used data augmentation to improve the

robustness and generalization of the model. We randomly rotated the coordinates of the atoms in the protein structure by uniformly sampling the Euler angles. Our ablation experiments showed that data augmentation significantly improved the performance of the model.

4.3.7 ABLATION STUDY OF PRE-TRAINING

We perform an ablation analysis for the influence of pre-training (Figure 4.10). We observe that on the test set, the root mean square error (RMSE) of the model pre-trained on the masked reconstruction task is significantly better than the model without pre-training for all ion types. For the majority of ion types, other metrics are similar or better than the model without pre-training. This suggests that the necessary pre-training on the masked reconstruction task is beneficial for the model to learn the structural features of metalloproteins.

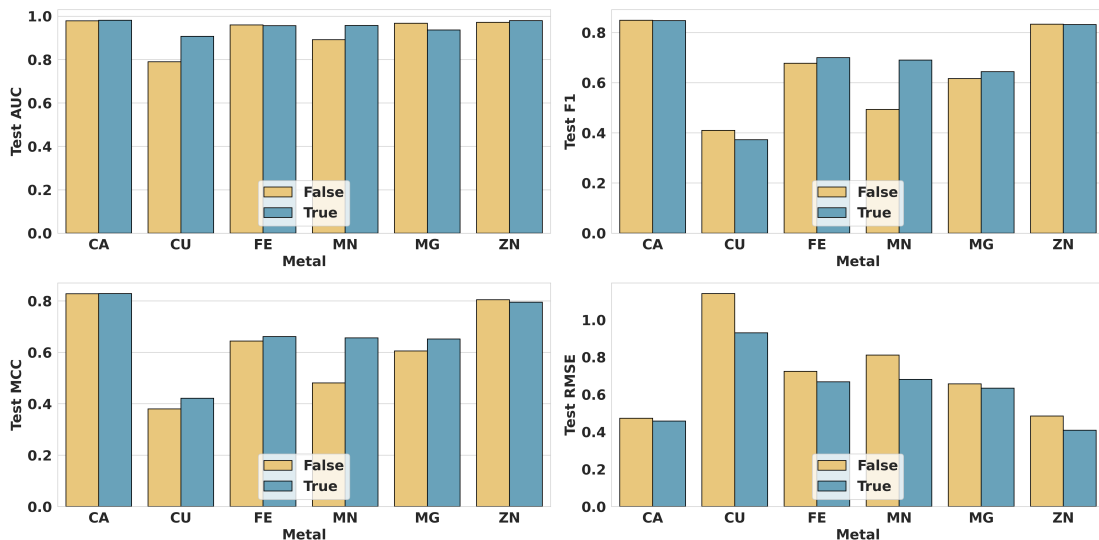


Figure 4.10: Ablation study of structure pre-training. The model pre-trained on the masked reconstruction task (blue) outperforms the model without pre-training (orange) on the test set in terms of AUC, F1, MCC and RMSE.

4.3.8 CLUSTERING OF PROBES

We used hierarchical clustering methods `SCIPY.CLUSTER.HIERARCHY.FCLUSTERDATA(X, T, CRITERION='DISTANCE')` to cluster the probes. The distance criterion t is set to 1.5 Å. In each group we keep only the probes with the highest probability.

4.3.9 EVALUATION METRICS

To comprehensively assess the efficacy of our approach, we employed the following metrics:

$$\left\{ \begin{array}{l} \text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \\ \text{SP} = \frac{\text{TN}}{\text{TN} + \text{FP}} \\ \text{SN(i.e., Rec)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{F1} = \frac{2 \cdot \text{Pre} \cdot \text{Rec}}{\text{Pre} + \text{Rec}} \\ \text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \\ \text{AUC} = \text{area under the receiver operating characteristic curve} \end{array} \right.$$

TP, FP, FN, and TN represent the true positive, false positive, false negative, and true negative, respectively. Here, the term “true” signifies the prediction that aligns with the actual outcomes, while “positive” denotes disordered states and vice versa. The Matthew’s correlation coefficient (MCC) ranges between -1 and 1 . A value of -1 indicates complete disagreement between the predictor and the actual results, while a value of 1 signifies a perfect predictor. A value of 0 corresponds to the performance of a random prediction. The F1 score lies between 0 and 1 . A value of 0 signifies complete disagreement between the predictor and the actual results, while a value of 1 represents a perfect prediction. A value of 0.5 corresponds to the performance of a random prediction. Other measures also fall within the value of 0 and 1 .

4.3.10 CODE AVAILABILITY

The corresponding reproducible source code is publicly available at <https://github.com/xu-shi-jie/PRIME>. A web server for PRIME is available at <https://onodalab.ees.hokudai.ac.jp/prime>.

4.4 CONCLUSION

In this work, we proposed a two-step approach that uses both sequence and structural information to predict the metal-binding sites of proteins. The developed method, named PRIME, can accurately predict the Zn^{2+} binding sites at high speed. Existing methods, which only use sequence or structural information, suffer from false positives and/or false negatives. Our novel methods can overcome such difficulties and significantly improve the prediction accuracy compared to the existing best methods.

We expect that this tool could be a valuable resource for the research community to study the metallo-proteins.

ACKNOWLEDGMENTS

This work was supported by Hokkaido University DX Doctoral Fellowship (JST SPRING, Grant Number JPMJSP2119) to S.X. and SATREPS Project “Recovering High-Value Bioproducts for Sustainable Fisheries in Chile (ReBiS)” funded by JST/JICA (Grant Number JPMJSA2206) and JSPS KAKENHI Grant Number JP24H02213 in Transformative Research Areas (A) JP24A202 Integrated Science of Synthesis by Chemical Structure Reprogramming (SReP) and JP24K01533 to A.O.

References

- [1] Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Henry Chao, Li Chen, Paul A Craig, Gregg V Crichlow, Kenneth Dalenberg, Jose M Duarte, et al. Rcsb protein data bank (rcsb.org): delivery of experimentally-determined pdb structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic Acids Res.*, 51(D1):D488–D508, 2023.
- [2] MA García-Sevillano, T García-Barrera, F Navarro, N Abril, C Pueyo, J López-Barea, and JL Gómez-Ariza. Use of metallomics and metabolomics to assess metal pollution in doñana national park (sw spain). *Environmental science & technology*, 48(14):7747–7755, 2014.
- [3] Shen Tang, Xiaonan Deng, Jie Jiang, Michael Kirberger, and Jenny J Yang. Design of calcium-binding proteins to sense calcium. *Molecules*, 25(9):2148, 2020.
- [4] K Pollmann, J Raff, M Merroun, K Fahmy, and S Selenska-Pobell. Metal binding by bacteria from uranium mining waste piles and its technological applications. *Biotechnol. Adv.*, 24(1): 58–68, 2006.
- [5] Jerker Porath, JAN Carlsson, Ingmar Olsson, and Greta Belfrage. Metal chelate affinity chromatography, a new approach to protein fractionation. *Nature*, 258(5536):598–599, 1975.
- [6] Amanda K Garcia, Hanon McShea, Bryan Kolaczowski, and Betül Kaçar. Reconstructing the evolutionary history of nitrogenases: Evidence for ancestral molybdenum-cofactor utilization. *Geobiology*, 18(3):394–411, 2020.
- [7] Simon L Dürr, Andrea Levy, and Ursula Rothlisberger. Metal3d: a general deep learning framework for accurate metal ion location prediction in proteins. *Nat. Commun.*, 14(1):2713, 2023.
- [8] Yu-Feng Lin, Chih-Wen Cheng, Chung-Shiuan Shih, Jenn-Kang Hwang, Chin-Sheng Yu, and Chih-Hao Lu. Mib: metal ion-binding site prediction and docking server. *J. Chem. Inf. Model.*, 56(12):2287–2291, 2016.

- [9] Chih-Hao Lu, Chih-Chieh Chen, Chin-Sheng Yu, Yen-Yi Liu, Jia-Jun Liu, Sung-Tai Wei, and Yu-Feng Lin. Mib2: metal ion-binding site prediction and modeling server. *Bioinformatics*, 38(18):4428–4429, 2022.
- [10] José-Emilio Sánchez-Aparicio, Laura Tiessler-Sala, Lorea Velasco-Carneros, Lorena Roldán-Martín, Giuseppe Sciortino, and Jean-Didier Maréchal. BioMetAll: identifying metal-binding sites in proteins from backbone preorganization. *J. Chem. Inf. Model.*, 61(1):311–323, 2020.
- [11] Qianmu Yuan, Sheng Chen, Yu Wang, Huiying Zhao, and Yuedong Yang. Alignment-free metal ion-binding site prediction from protein sequence through pretrained language model and multi-task learning. *Briefings Bioinf.*, 23(6):bbac444, 2022.
- [12] Xiuzhen Hu, Qiwen Dong, Jianyi Yang, and Yang Zhang. Recognizing metal and acid radical ion-binding sites by integrating ab initio modeling with template-based transfers. *Bioinformatics*, 32(21):3260–3269, 2016.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [14] Chengxin Zhang, Xi Zhang, Peter L Freddolino, and Yang Zhang. Biolip2: an updated structure database for biologically relevant ligand–protein interactions. *Nucleic Acids Res.*, 52(D1):D404–D412, 2024.
- [15] Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, 35(11):1026–1028, 2017.
- [16] R Girshick. Fast R-CNN. *arXiv preprint arXiv:1504.08083*, 2015.
- [17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

5

Improved evolutionary rates estimation using Bayesian optimization and parameterized sorting indices

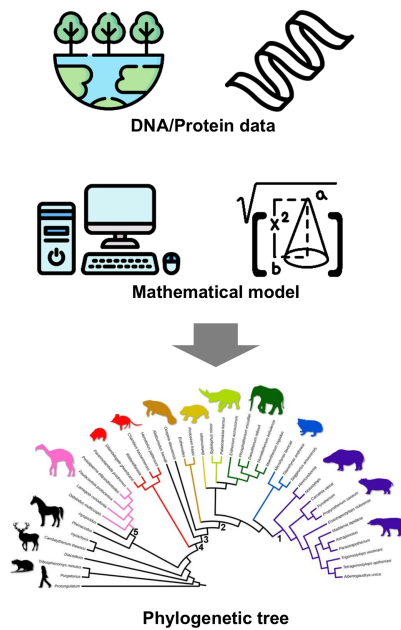
5.1 INTRODUCTION

Phylogenetics is a powerful framework for reconstructing the evolutionary history of organisms by analyzing their genetic sequences (Figure 5.1). Over the past decades, several methods have been developed to construct phylogenetic trees, including neighbor joining [1], maximum-likelihood (ML) [2], Bayesian inference (BI) [3], and graph splitting [4]. Among these, ML and BI are widely recognized as the most accurate approaches. They model evolutionary processes as continuous-time Markov processes, using specific substitution models to account for sequence changes over time [5]. The advent of high-throughput sequencing technologies has dramatically increased the size and complexity of phylogenetic datasets, leading to the emergence of phylogenomics—phylogenetic analysis on a genomic scale. To improve phylogenetic accuracy, researchers often select DNA sequences from multiple genes and concatenate them into a supermatrix [6]. It has been suggested that including more sites in such analyses enhances the reliability of inferred phylogenies [7]. However, this approach introduces challenges, particularly due to site heterogeneity in genomic data. Site heterogeneity arises from varying

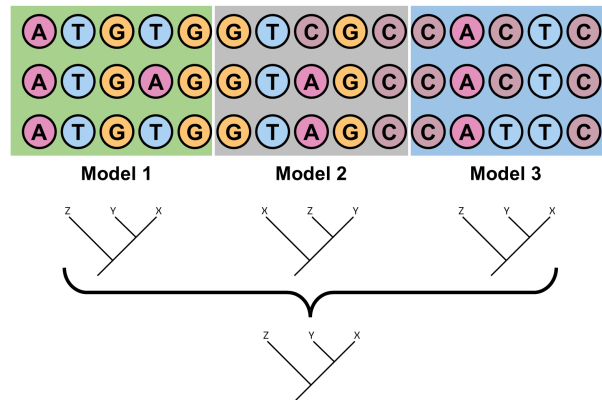
evolutionary rates among different genomic regions. For instance, the third codon position of protein-coding genes evolves faster than the first and second positions due to reduced selective pressure [8]. Similarly, heterogeneity in amino acid composition may result from differences in the secondary structures of proteins [9]. These variations make it difficult for a single homogeneous model to adequately capture the evolutionary dynamics within genomic data. Consequently, the potential accuracy gains achieved by integrating evolutionary signals from multiple genes are limited [10].

Figure 5.1: Phylogenetics uses genetic sequences and mathematical models to infer the evolutionary relationships among organisms. Partitioned models divide sites into disjoint subsets, each assigned a distinct model that may vary in substitution rates, base frequencies, and branch lengths. These models enhance the accuracy of phylogenetic inference by accounting for site heterogeneity.

Molecular phylogenetics



Partitioned models



Partitioned models have been introduced to account for site heterogeneity and enhance the accuracy of phylogenetic inference. These models work by dividing sites into several disjoint subsets, each assigned a distinct model that may vary in substitution rates, base frequencies, and branch lengths [11]. Their significance has been established in phylogenomic analyses [6], as partitioned models influence key aspects such as tree topology, branch lengths, bootstrap support, and divergence times [12, 13, 14, 15, 16, 17]. The number of partitions is a critical factor affecting the accuracy of partitioned models. Generally, increasing the number of partitions allows for a better fit to the data by incorpo-

rating more evolutionary models. However, this also raises the risk of overfitting, which can distort subsequent analyses [18]. Striking a balance between model fit and complexity is essential. To achieve this balance, information criteria such as the Akaike Information Criterion (AIC) [19], the corrected AIC (AICc) [20], and the Bayesian Information Criterion (BIC) [21] are frequently used as optimization objectives in partitioned models.

There remain critical problems in partitioning sites efficiently and properly. Firstly, the number of all possible partitioning of an n -sites alignment is the Bell number B_n satisfying the recursive definition $B_{n+1} = \sum_{k=0}^n \binom{n}{k} B_k$, which increases faster than an exponential function as n increases. Consequently, exhaustive brute-force searches for all possible partitioning schemes become infeasible even for alignments with relatively few sites [22].

Another challenge arises with partitioning methods that rely on *prior* knowledge, such as codon positions or protein secondary structures. These approaches, while intuitive, do not ensure correct site partitioning and have undergone improvements over time [12, 23, 24, 25]. Typically, such methods rely on a predefined set of partitioning candidates, requiring expert judgment and potentially introducing bias. To address these limitations, automatic partitioning methods have been developed to explore the entire partitioning space using heuristic algorithms. For instance, one study proposed a top-down greedy approach where a single initial partition is progressively divided into smaller partitions based on k -means clustering [26] of site rates estimated from reconstructed phylogenetic trees [18]. Alternatively, another method [27] adopts a bottom-up strategy, recursively merging the most similar site pairs. Despite their promising performance, these automatic methods face notable drawbacks. First, they depend on estimated site rates derived from reconstructed phylogenetic trees, which are computationally intensive to generate and may be inaccurate. Second, their reliance on greedy algorithms risks convergence to local optima. These limitations restrict the applicability of these methods, particularly for large-scale genomic datasets.

TIGER (Tree Independent Generation of Evolutionary Rates) was recently introduced as a method to estimate evolutionary rates at individual sites without relying on phylogenetic trees [28]. The primary aim of its development was to exclude rapidly evolving sites that can compromise the accurate reconstruction of phylogenies. Partitioned models based on TIGER have proven successful across numerous phylogenetic investigations [29, 30, 31, 32, 33, 34, 35, 36, 37]. To enhance TIGER, Rota *et al.* proposed incorporating a user-defined “division factor” [38], while Kim *et al.* introduced a recursive trisection approach to automate the selection of optimal partition numbers [17]. These advancements address some of the challenges associated with automatic partitioning in genomic datasets. Nonetheless, existing TIGER-based approaches still fall short in achieving optimal accuracy, underscoring the need for more accurate and efficient partitioning algorithms.

In this study, we present a novel approach for site partitioning, named PsiPartition, which leverages Parameterized Sorting Indices (PSI) and Bayesian optimization. Initially, we introduce PSI as an improved alternative to the TIGER method. Subsequently, Bayesian optimization is employed to refine PSI and determine the optimal number of partitions. Our empirical analyses demonstrate that PsiPartition significantly outperforms existing methods, achieving superior results based on the Bayesian Information Criterion (BIC) and the corrected Akaike Information Criterion (AICc). Moreover, on simulated data sets, PsiPartition reconstructs phylogenetic trees with the highest accuracy, as measured by the Robinson-Foulds distance. Notably, this method establishes a versatile Bayesian optimization-based framework for efficiently identifying the optimal number of partitions, offering potential for seamless adaptation to other partitioning strategies.

5.2 METHOD

5.2.1 DATA SETS

For phylogenetic inference benchmarking, we employ eight empirical DNA data sets from the reported work [39, 32, 16, 40, 41, 42, 43, 33]. Their statistics are listed in Table 5.1. These data sets are also employed as the benchmark data sets in previous studies [17, 38]. Alignments in the data sets have a range of base pair lengths from 4435 to 6372 and the number of taxa from 31 to 164. Each sequence in the alignments is the concatenated sequences from the mitochondrial gene (COI) and other four to seven nuclear genes commonly employed for lepidopteran phylogenetics, including CAD, EF-1 α , GAPDH, IDH, MDH, RpS5, and wingless [44].

Table 5.1: The empirical DNA data sets for benchmarking.

Name	No. of taxa	No. of loci	No. of sites
Arctiina [39]	113	8	5809
Calisto [32]	90	6	5297
Choreutidae [16]	41	8	6293
Coenonymphina [40]	69	5	4435
Geometridae [41]	164	8	5998
Morpho [42]	31	8	6372
Noctuidae [43]	78	8	6365
Pieridae [33]	110	8	6247

Six empirical protein data sets from the reported work [45] are also employed for benchmarking on the protein data. Their statistics are listed in Table 5.2. Alignments in the data sets have a range

of amino acid lengths from 385 to 811 and the number of taxa from 17 to 31. Each alignment in the data sets includes the sequences of heat shock proteins Hsp83, Hsc70-3, Hsc70-4, Hsc70-5, Hsp60, and Hsp40. These proteins are the primary molecular chaperones protecting against thermal damage. Note that they are relatively small proteins and the number of sites is smaller compared to the DNA data sets.

Table 5.2: The empirical protein data sets for benchmarking.

Name	No. of taxa	No. of sites
Nguyen_2016a	17	688
Nguyen_2016b	31	680
Nguyen_2016c	25	811
Nguyen_2016d	17	704
Nguyen_2016e	17	385
Nguyen_2016f	17	583

5.2.2 PARAMETERIZED SORTING INDICES (PSI)

We introduce the proposed Parameterized Sorting Indices (PSI) by reviewing the Tree-Independent Generation of Evolutionary Rates (TIGER) [28]. In the original paper of TIGER, the evolutionary rates of site i are estimated as

$$r_i = \frac{1}{n-1} \sum_{j \neq i} \text{pa}(i, j) \quad (5.1)$$

where n is the number of sites and

$$\text{pa}(i, j) = \frac{1}{|P_j|} \sum_{x \in P_j} a(x, P_i) \quad (5.2)$$

is the *agreement score*. Here the function $a(x, P_i)$ is defined as $a(x, P_i) = 1$ if $x \subset \mathcal{A}$ and $\mathcal{A} \in P_i$, and otherwise 0. The set P_i is the *set partition* of the site i , whose elements are disjoint subsets of $\{1, 2, \dots, m\}$ where m is the number of taxa. For each subset $\mathcal{A} \in P_i$, the character of the l -th sequence at site i has the same type for all $l \in \mathcal{A}$. For example, consider two sites $S_i = \text{CTTAA}$ and $S_j = \text{AGGGG}$ with the set partition $P_i = \{\{1\}, \{2, 3\}, \{4, 5\}\}$ and $P_j = \{\{1\}, \{2, 3, 4, 5\}\}$, respectively. We have $\text{pa}(\mathcal{A}, B) = 0.5$ because out of two subsets $\{1\}$ and $\{2, 3, 4, 5\} \in P_j$, only $\{1\} \subset \{1\} \in P_i$. $\text{pa}(B, \mathcal{A}) = 1$ because for all subsets $\{1\}, \{2, 3\}, \{4, 5\} \in P_i$, we have $\{1\} \subset \{1\} \in$

P_j , $\{2, 3\} \subset \{2, 3, 4, 5\} \in P_j$, and $\{4, 5\} \subset \{2, 3, 4, 5\} \in P_j$. The TIGER of site S_i and S_j should be $r_i = 3/3 = 1$ and $r_j = 1/3$, respectively.

PSI has two key innovations. Firstly, in the original TIGER, the function $a(x, P_i)$ is defined as binary values 1 or 0 for all characters x at site j , regardless of the characters' types. However, characters representing different base pairs or amino acids should have different properties and should be treated differently. This is similar to the idea of base frequency parameters implemented in most phylogenetic inference tools [46, 47]. We therefore introduce a group of parameters w to represent these differences among characters. For example, we adopt five parameters $w = \{w_A, w_C, w_G, w_T, w_{\text{gap}}\}$ for DNA data sets, where w_A is the parameter for character A , w_{gap} is the parameter for character gaps or missing characters, and so on. We define $a_w(x, P_i) = w_{p(x)}$ if $x \in A$ and $A \in P_i$ and otherwise 0 where $p(x)$ is the corresponding character of subset x . Similarly, there should be 21 parameters for protein data sets.

Secondly, instead of using Equation 5.1, we utilize a comparison between two sites to define PSI at site i by following

$$\text{PSI}_i = |\{j | \text{pa}'(i, j) > \text{pa}'(j, i)\}| \quad (5.3)$$

where $\text{pa}'(i, j)$ is the defined as following

$$\text{pa}'(i, j) = \frac{1}{|P_j|} \sum_{x \in P_j} a_w(x, P_i). \quad (5.4)$$

PSI_i ranging from 0 to $n - 1$ represents the number of sites that site i agrees more than it is agreed by.

After calculating the PSI for all sites, we bin the sites into k partitions based on their PSIs. Denote $\alpha = \min_i \text{PSI}_i$ and $\beta = \max_i \text{PSI}_i$, then k partitions are defined as following

$$\begin{aligned} P_1 &= \{i | \text{PSI}_i \in [\alpha, \alpha + \frac{\beta - \alpha}{k})\} \\ P_2 &= \{i | \text{PSI}_i \in [\alpha + \frac{\beta - \alpha}{k}, \alpha + 2\frac{\beta - \alpha}{k})\} \\ &\dots \\ P_k &= \{i | \text{PSI}_i \in [\alpha + (k - 1)\frac{\beta - \alpha}{k}, \beta]\} \end{aligned} \quad (5.5)$$

where k is the number of partitions. For example, if $\alpha = 0$ and $\beta = 10$, then 3 partitions are $P_1 = \{i | \text{PSI}_i \in [0, \frac{10}{3})\}$, $P_2 = \{i | \text{PSI}_i \in [\frac{10}{3}, \frac{20}{3})\}$, and $P_3 = \{i | \text{PSI}_i \in [\frac{20}{3}, 10]\}$.

Note that the parameters w and the number of partitions k described in this section are not determined and will be optimized based on the input data.

5.2.3 BAYESIAN OPTIMIZATION FOR PSI

It is important to decide the best number k of partitions in the partitioned models. When k equals the number of sites, a single-site partitioning can fit the data very well. This may cause overfitting and produce the reconstructed phylogeny different from the true one. When $k = 1$, the single partition is not suitable for describing heterogeneity among the sites. Some existing methods employ greedy algorithms to determine the number of partitions which is time-consuming for large data sets.

Bayesian optimization (BO), a black-box method, has been widely used in many fields for parameter tuning and are also adopted in this study. It has a significant advantage when the objective function is expensive to evaluate and the number of parameters is small. In our situation, the objective function is the information criterion (i.e., BIC) of the partitioned models, which is expensive to evaluate because it requires full phylogenetic inference on the given data. The optimized parameters are the number of partitions k and the introduced parameters w of PSI, which are few (i.e., 6 for DNA and 22 for proteins). BO utilizes a surrogate model such as the Gaussian process model [48] to approximate the objective function. The parameters are estimated based on the acquisition functions such as expectation improvement and upper confidence bound. Then BO evaluates the objective function on these parameters to obtain a new observation and updates the surrogate model with all observations. This process repeats until the parameter converges or the maximum number of iterations is reached.

We propose PsiPartition algorithms based on BO, as shown in Algorithm 1. The input of the algorithm is the alignment $\mathcal{A}$, the maximum number of partitions $k_{\max}$, and the maximum iterations steps n . The output is the optimal partition P^* , the optimal parameters s^* and the optimal objective value o^* . The algorithm firstly initializes an empty set S and samples the number of partitions k_1 from $\{2, 3, \dots, k_{\max}\}$ and the parameters w_1 from $[0, 1]^t$ where $t = 5$ for DNA data and $t = 21$ for amino acid data. For each iteration i , it calculates the PSI ψ_i of all sites with parameters w_i and bins the sites into k_i partitions $P_i = \{p_j\}_{j=1}^{k_i}$ based on ψ_i . It thus evaluates the phylogenetic software (e.g., IQ-TREE) with the partitioning P_i to obtain the BIC b_i . Each evaluation is recorded as an observation (P_i, s_i, b_i) where $s_i = (w_i, k_i)$, and pushed into the set S . Then it updates the surrogate model m with all observations $(P_i, s_i, b_i) \in S$ and determines the next parameters $s_{i+1} = (w_{i+1}, k_{i+1})$ by the surrogate model m with expected improvement. The iterations will be repeated for $k_{\max}$ times. Finally, the algorithm returns the partitioning P^* , the parameters s^* , and the BIC o^* that has the minimum BIC o^* in S .

To indicate the necessity of BO, we develop a variant of PsiPartition, termed PsiPartitionFast, which freezes all parameters w_i as 1 in Algorithm 1. Compared to PsiPartition, it only optimizes the number of partitions k . The search space can be reduced and the overall steps of optimization are also less than PsiPartition. A detailed description is referred to the pseudocode in 2.

Algorithm 1 PsiPartition

Require:

- $\mathcal{A}$: the alignment
- $k_{\max}$: the maximum number of partitions
- n : the maximum iterations steps

Ensure:

- P^* : the optimal partition
 - s^* : the optimal parameters
 - o^* : the optimal objective value
 - 1: Uniformly sample k_1 from $2, 3, \dots, k_{\max}$
 - 2: Uniformly sample w_1 from $[0, 1]^t$ $\triangleright t = 5$ for DNA, 21 for protein
 - 3: Let S be an empty set
 - 4: Initialize a surrogate model m
 - 5: **for** $i = 1$ to n **do**
 - 6: Let ψ_i be the PSI of all sites with parameters w_i
 - 7: Bin the sites of $\mathcal{A}$ into k_i partitions based on ψ_i as P_i
 - 8: Evaluate phylogenetic tool (e.g., IQ-TREE) with partitioning P_i and obtain BIC b_i
 - 9: $s_i \leftarrow (k_i, w_i)$
 - 10: $S \leftarrow S \cup \{(P_i, s_i, b_i)\}$
 - 11: Update m with S
 - 12: Determine the next parameters $s_{i+1} = (k_{i+1}, w_{i+1})$ by the surrogate model m with expected improvement
 - 13: **end for**
 - 14: **return** $(P^*, s^*, o^*) \in S$ that has the minimum BIC o^*
-

EVALUATION METRICS

To evaluate the performance of different partitioning methods, we employ the Bayesian information criterion (BIC) [21] and the corrected Akaike information criterion (AICc) [20]. The BIC and AICc are defined as the following

$$\text{BIC} = -2 \ln L + k \ln n$$

and

$$\text{AICc} = -2 \ln L + 2k + \frac{2k(k+1)}{n-k-1}$$

where L is the likelihood of the model, k is the number of model parameters, and n is the number of sites. BIC and AICc are both used to assess the goodness of partitioned models, which with the smallest BIC or AICc are considered the best ones. The BIC is the default criterion for model selection in IQ-TREE software [49]. We thus adopted it as the major evaluation metric and AICc as a reference. It is worth noting that the partitioning of sites is usually considered as *a priori*. Therefore the parameters w introduced in the aforementioned Bayesian optimization algorithm and the number of partition k should not be included in the calculation of BIC and AICc.

The Robinson-Foulds (RF) distance [50] is adopted to evaluate the accuracy of the reconstructed phylogenetic trees. Two branches sharing a common ancestor in a tree split a part of taxa into two disjoint subsets (i.e., clades). Therefore they recursively imply a nested partitioning of the taxa. The RF distance is defined as the summation of the number of partitions implied by one tree but not by another one. It is a metric of the topological distance between two trees, and the smaller the RF distance, the more similar the two trees. In our experiments, we compute the RF distance between the true simulated trees and the reconstructed trees and compare them of different partitioning methods.

We utilize the transfer distance to compare the similarity between two partitions. Suppose X be a set, and $P = \{C_1, C_2, \dots, C_n\}$ and $P' = \{C'_1, C'_2, \dots, C'_m\}$ are two partitions of X . Without loss of generality, we assume $n = m$ because we can add empty subsets to the smaller one. We firstly define the *concordance* as following

$$c(P, P') = \max_{\sigma \in \mathcal{M}} \sum_{i=1}^n |C_i \cap C'_{(\sigma(i))}|$$

where $\mathcal{M}$ is the set of all permutations of $\{1, 2, \dots, n\}$. The *transfer distance* is defined as following

$$t(P, P') = 1 - \frac{c(P, P')}{|X|}.$$

It is obvious to see that $0 \leq t(P, P') \leq 1$ and $t(P, P') = 0$ if and only if $P = P'$. A smaller transfer

distance means more similarity between two partitions. The proof of the properties of the transfer distance is described in [51] with more details.

5.3 RESULTS

PsiPartition is implemented within Python 3.12.1. To perform the Bayesian optimization, wandb package (version 0.16.1) [52] are also used [53]. It employed normalized Gaussian process model with Matern kernel [54] and parameter $\alpha = 10^{-7}$. Expected improvement is utilized as the default acquisition function. The parameters $k_{\max}$ and n in Algorithm 1 are both set up to 30 by default unless mentioned. The partitioned models and their information criterion are evaluated by ModelFinder [49] implemented in IQ-TREE [46] (version 1.6.12). UFBoot [55] in IQ-TREE is also enabled for ultrafast bootstrap calculation of phylogenetic trees. The experiments are performed on a Linux machine with an AMD Ryzen 9 7950X3D 16-core processor and 64 GB memory. Multiple threads feature of IQ-TREE are adopted to accelerate the phylogenetic inference. Interactive Tree Of Life (ITOL) online tools [56] are used to visualize the phylogenetic trees and the bootstrap support values of branches.

5.3.1 PSI-PARTITION OUTPERFORMS OTHER PARTITIONING METHODS ON EMPIRICAL DATA SETS

To evaluate the performance of our proposed methods, PsiPartition and PsiPartitionFast, and existing state-of-the-art methods, we applied them to eight empirical DNA data sets listed in Table 5.1. The experimental results are shown in Fig 5.2. The compared methods include the default IQ-TREE using the single homogeneous model without partitioning (NP), RatePartition with parameter $\text{div_factor} = 4$ (RatePartition-4) and $\text{div_factor} = 5$ (RatePartition-5), and mPartition. We show the advantages of partitioned models compared to NP by BIC decreases defined as the following

$$\Delta\text{BIC} = \text{BIC of partitioned models} - \text{BIC of model without partitioning},$$

and ΔAICc is defined similarly.

On all data sets, our proposed methods PsiPartition and PsiPartitionFast significantly outperform other partitioned methods in terms of the ΔBIC and ΔAICc . Both of two significantly reduce the BIC and AICc values of the partitioned models, as shown in 5.4. In addition, PsiPartition is better than PsiPartitionFast, which suggests that the parameters w play a vital role for partitioning. On the data set Calisto, PsiPartitionFast is slightly better than PsiPartition, which is probably caused by the relatively

small size of the data set and the randomness in phylogenetic inference [57]. The results demonstrate that our proposed methods are more accurate than the default IQ-TREE and other existing partitioning methods, especially on the data sets with more sites and taxa.

We visualize the distribution of partitions in Fig 5.3. Considering the differences among data, the number of partitions k determined by PsiPartition for different empirical DNA data sets are distinct, ranging from 13 to 30. This suggests that our proposed methods are capable of searching for different optimal partition schemes on different data sets. Furthermore, the number of sites that fall into each partition are also plotted. Bin indices represent the binning of sites based on the PSI, i.e., the sites with lower PSI will be assigned to the smaller bin indices, and vice versa. It is clear that the distribution of the number of sites in each bin is quite imbalanced and most of the sites are assigned to the first several bins, while there are still a few sites assigned to the last several bins. This is similar to those in previous study, suggesting that our methods properly partition the sites.

PsiPartition and PsiPartitionFast produce totally different partitioning compared to existing methods. The transfer distances between the partitions generated by our proposed methods and other existing methods are shown in Fig 5.4. Here the X-axis and Y-axis represent the partitions generated by different methods and blocks with smaller values and darker colors mean more similarities between the two partitions. The results demonstrate that the partitions generated by our proposed methods, PsiPartition and PsiPartitionFast, have quite large transfer distance to the partitions generated by remaining methods. On some data sets such as Morpho and Coenonymphina, the partitions generated by PsiPartition and PsiPartitionFast are also considerable different from each other. Interestingly, two existing methods, RatePartition-4 and RatePartition-5 generated quite similar partitions to each other, because they utilized identical TIGER with only different division factors. Our PsiPartition instead optimized the PSI separately even on the same data set. These results suggest that our proposed methods are capable of searching for different partition schemes effectively and potentially finding the optimal ones.

Our proposed methods are also superior on the protein data. We evaluate the performance comparison of different partitioning methods on the empirical protein data sets, as shown in Fig 5.5. On most data sets, our proposed PsiPartition and PsiPartitionFast significantly reduce the BIC and AICc, as shown in 5.5. PsiPartition significantly outperforms other methods in terms of the Δ BIC and Δ AICc. The only exception is on the data set “Ngyuen_2016e” where the PsiPartition is slightly worse than the RatePartition-4 and RatePartition-5. This is because the Ngyuen_2016e data set has a relatively small number of sites (385 amino acids) while others have about 600 sites or more. As the number of sites increases, the heterogeneity among sites could increase. The improvements achieved by our proposed methods compared to existing methods become more significant. Additionally, the PsiPartitionFast

also achieved comparable performance to other methods. It is slightly worse than the PsiPartition because it does not adopt Bayesian optimization. The results demonstrate that our proposed PsiPartition is more accurate than the default IQ-TREE and other methods. By comparing PsiPartition and PsiPartitionFast, it is suggested that Bayesian optimization is necessary for accurate phylogenetic inference.

5.3.2 BAYESIAN OPTIMIZATION EFFICIENTLY DETERMINES THE OPTIMAL PARAMETERS OF PSIPARTITION

We further assess the efficiency of PsiPartition on different empirical DNA data sets. The tracking of the evaluated BIC and AICc during the optimization iterations is shown in Fig 5.6. The results demonstrate that the Bayesian optimization gradually decreases BIC and finally converges (or fluctuates) at the optimal BIC values. AICc can be similarly analyzed. More specifically, we list the steps of iterations to achieve the best BIC and AICc in Table 5.3. These optimizations take a range of 8 to 29 steps (17.8 on average) to achieve an optimal BIC. Similarly, they take 9 to 29 steps (19.1 on average) to achieve an optimal AICc. The results illustrate that our proposed methods can expeditiously determine the optimal parameters in a limited steps of iterations.

Table 5.3: The numbers of iterations to achieve the best BIC and AICc on different empirical DNA data sets.

Name	# Iter. to opt. BIC	# Iter. to opt. AICc
Arctiina	12	12
Calisto	8	9
Choreutidae	29	29
Coenonymphina	22	22
Geometridae	28	25
Morpho	16	23
Noctuidae	15	15
Pieridae	12	18

It is intractable to find the global minimum of BIC on all possible parameters, therefore we run PsiPartition 1000 times for the Morpho data set with uniformly sampled random parameters w from $[0, 1]^t$ and k from $\{2, 3, \dots, k_{\max}\}$, respectively. We check whether Bayesian optimization-based minimum BIC achieves the best among all randomly evaluated BICs. Considering the number of parameters of PsiPartition on DNA data sets is six, t-distributed Stochastic Neighbor Embedding (t-SNE) [58] is employed to embed parameters into the 2-dimensional plane, which will keep the Euclidean

distance between the parameters as much as possible. We normalize each parameter to normal distribution before applying t-SNE to make them comparable. The BIC values of each evaluation are shown as the sizes and colors of the spots (darker and larger spots mean smaller BIC, and vice versa). As shown in Fig 5.7, it is clear that most of the evaluations from Bayesian optimization produce sufficiently good BIC values, which are concentrated into a cluster in the left part. The optimal parameters from Bayesian optimization (circled in orange) also achieve the best BIC among all random evaluations. These results suggest that PsiPartition is capable of approaching the best parameters efficiently compared to the random search.

PSIPARTITION ACCURATELY RECONSTRUCTS SIMULATED PHYLOGENETIC TREES

To evaluate the performance of our algorithm on simulated data sets, we generate 100 simulated DNA alignments with 49 sequences and 5000 sites. We employ Seq-Gen software [59] and the General Time Reversible (GTR) model [5] with 4-category Gamma distribution to generate the alignments from 100 randomly generated trees, which are described in 3. We utilize Seq-Gen’s parameters “-r rate_matrix -f base_freq”, where rate_matrix and base_freq are 6- and 4-dimensional vectors. Their elements are randomly generated from a uniform distribution $U[0, 1]$ and represent the rate matrix and base frequencies, respectively. For base_freq, an additional L_1 normalization is performed to make them indeed base frequencies. We segmented 5,000 sites into 1 to 10 equal-length loci and applied Seq-Gen to each locus then concatenated the produced alignments of each locus into the single 5000-site alignments. This process is repeated 10 times for each number of loci and finally, we obtained 100 simulated alignments. We denoted this data set as “SimDNA-1”.

PsiPartition and PsiPartitionFast present a superior performance compared to other methods on the simulated data set SimDNA-1 in terms of the RF distance between the true trees and the reconstructed trees, as shown in Fig 5.8. The standard deviations of the RF distances are also shown. The results demonstrate that, for genomic data with heterogeneous site rates, our methods are robust and more accurate than the single homogeneous model without partitioning and existing partitioning methods. We emphasize that the differences of performance between our methods and others on the data sets with only one locus are not significant while the differences become larger in the data sets with more loci. Such differences do not increase when the number of loci continues to increase because the total lengths of sequences are always 5,000. This suggests the necessity of partitioning the sites by PsiPartition for large data sets. It is worth noting that both RatePartition-4 and RatePartition-5 failed to run on some of the simulated data sets with 87% (87/100) success rate because some sites are incorrectly partitioned and only one state is observed in some partitions thus IQ-TREE failed to start

phylogenetic inference.

We also generate 100 simulated DNA alignments with 6 to 15 loci and the lengths of single locus are always set up to 1,000. That is, we obtain alignments that have a range of lengths from 6,000 to 15,000 and each length has 10 alignments. The algorithms and configurations to generate them are the same as aforementioned. We denote this data set as “SimDNA-2”. We compare the performance of different methods in terms of RF distance, as shown in Fig 5.9. It is clear that our PsiPartition significantly outperforms other methods on most of the alignment. The only exception when the number of loci is 10 can be explained as a random bias during the phylogenetics inference because only 10 alignments are evaluated for each length. It is worth noting that some methods cannot reconstruct all phylogenies on SimDNA-2 (mPartition reconstructs 99 and RatePartition-5 reconstructs 93 out of 100 phylogenies).

5.3.3 TREE ANALYSIS ON NOCTUIDAE DATA SET

We reproduced the phylogeny in the reported work [43] by IQ-TREE, and compared it to the phylogeny inferred by PsiPartition. Overall, the results show that the reconstructed tree based on PsiPartition has more likely topology and higher bootstrap supports of the branches, as shown in 5.14, compared to the tree without partitioning in 5.15. To emphasize the accuracy of PsiPartition clearly, we compare the reconstructed topologies and bootstrap supports in detail.

As shown in Fig 5.10, in the reproduced phylogeny (tree without partitioning), the species *Arcte modesta* and *Pseudoarcte melanis* are firstly categorized as a clade with a bootstrap support of 57 and then categorized with *Paracte schneideriana* into a clade with a bootstrap support of 100. The reproduced tree has the same topology and similar bootstrap support (65) to the tree in the reported work [43]. Both of them are below the commonly used threshold of 70 and are not convinced. On the other hand, in the tree generated by PsiPartition, *Arcte modesta* and *Paracte schneideriana* are firstly categorized into a clade with a bootstrap support of 93 and then categorized with *Pseudoarcte melanis* into a clade with a bootstrap support of 100, which are more convincing, suggesting a possibly correct reconstruction of these species.

As shown in Fig 5.11, in the reproduced phylogeny (tree without partitioning), the species *Diloba caeruleocephala*, *Raphia abrupta* or *sp*, *Deltote uncula*, and *Cucullia umbratica* are nested into three clades with bootstrap supports (43/81/59). In the reported work [43], *Deltote uncula* and *Cucullia umbratica* are also categorized as a single clade with posterior probabilities of 0.61, which is very low. On the other hand, in the tree generated by PsiPartition, *Deltote uncula*, *Diloba caeruleocephala*, *Raphia abrupta* or *sp*, and *Cucullia umbratica* are nested into three clades with bootstrap supports (90/89/77). Overall, PsiPartition provides a different and possibly more accurate reconstruction of these species

compared to the original work and reproduced work and reasons could be discussed in future work.

As shown in Fig 5.12, both the reproduced phylogeny (tree without partitioning) and the tree generated by PsiPartition reconstructed the same topology for the species of *Craniophora ligustri*, *Polygrammate hebraicum*, *Comachara cadburyi*, *Harrisimemna trisignata*, and *Cerma cerintha* yet with different bootstrap supports (98/100/62/51 v.s. 100/100/82/80). In the reported work [43], *Craniophora ligustr* together with *Harrisimemna trisignata* and *Cerma cerintha* are categorized into a clade with very low posterior probabilities of 0.59. PsiPartition thus provides an accurate reconstruction of this clade with higher bootstrap supports.

These results demonstrate that, by properly partitioning sites with our proposed PsiPartition method, we can improve the accuracy of the reconstructed phylogenetic trees and the bootstrap supports of the branches.

5.4 CONCLUSION

In this study, we proposed a novel partition approach based on parameterized sorting indices (PSI) and Bayesian optimization to accurately and efficiently obtain the optimal partitioning of sites in genomic data. We applied our method to empirical data sets and achieved the best information criterion (BIC and AICc) compared to the models without partitioning or other existing state-of-the-art methods. We also show that Bayesian optimization-based PsiPartition is sufficiently efficient to determine the optimal parameters for partitioning the sites. In addition, we tested PsiPartition on the challenging simulated data sets with various site heterogeneity and demonstrated that our method outperforms other methods in terms of the average RF distance of the reconstructed phylogenetic trees to the actual trees. Finally, we show that on the empirical data set Noctuidae, our proposed method can improve the accuracy of the reconstructed phylogenetic trees, as well as the bootstrap supports of the branches. Overall, our proposed Bayesian optimization-based method achieves the best performance and provides a novel general framework to efficiently determine the optimal number of partitions, which can be easily extended to other partitioning methods. We hope its efficiency and effectiveness contribute to the growing demand for partitioning methods in phylogenetic analysis of large-scale genomic data.

ACKNOWLEDGMENTS

This work was supported by Hokkaido University DX Doctoral Fellowship (JST SPRING, Grant Number JPMJSP2119) to S.X. and SATREPS Project “Recovering High-Value Bioproducts for Sustainable Fisheries in Chile (ReBiS)” funded by JST/JICA (Grant Number JPMJSA2206) and JSPS

KAKENHI Grant Number JP24Ho2213 in Transformative Research Areas (A) JP24A202 Integrated Science of Synthesis by Chemical Structure Reprogramming (SReP) and JP24Ko1533 to A.O.

Figure 5.2: The performance comparison of different partitioning methods on the empirical DNA data sets (Arctiina, Calisto, Choreutidae, Coenonymphina, Geometridae, Morpho, Noctuidae, and Pieridae). Our proposed methods (PsiPartition and PsiPartitionFast) achieves the best performance in terms of the ΔBIC and $\Delta AICc$, significantly outperforming the other methods.

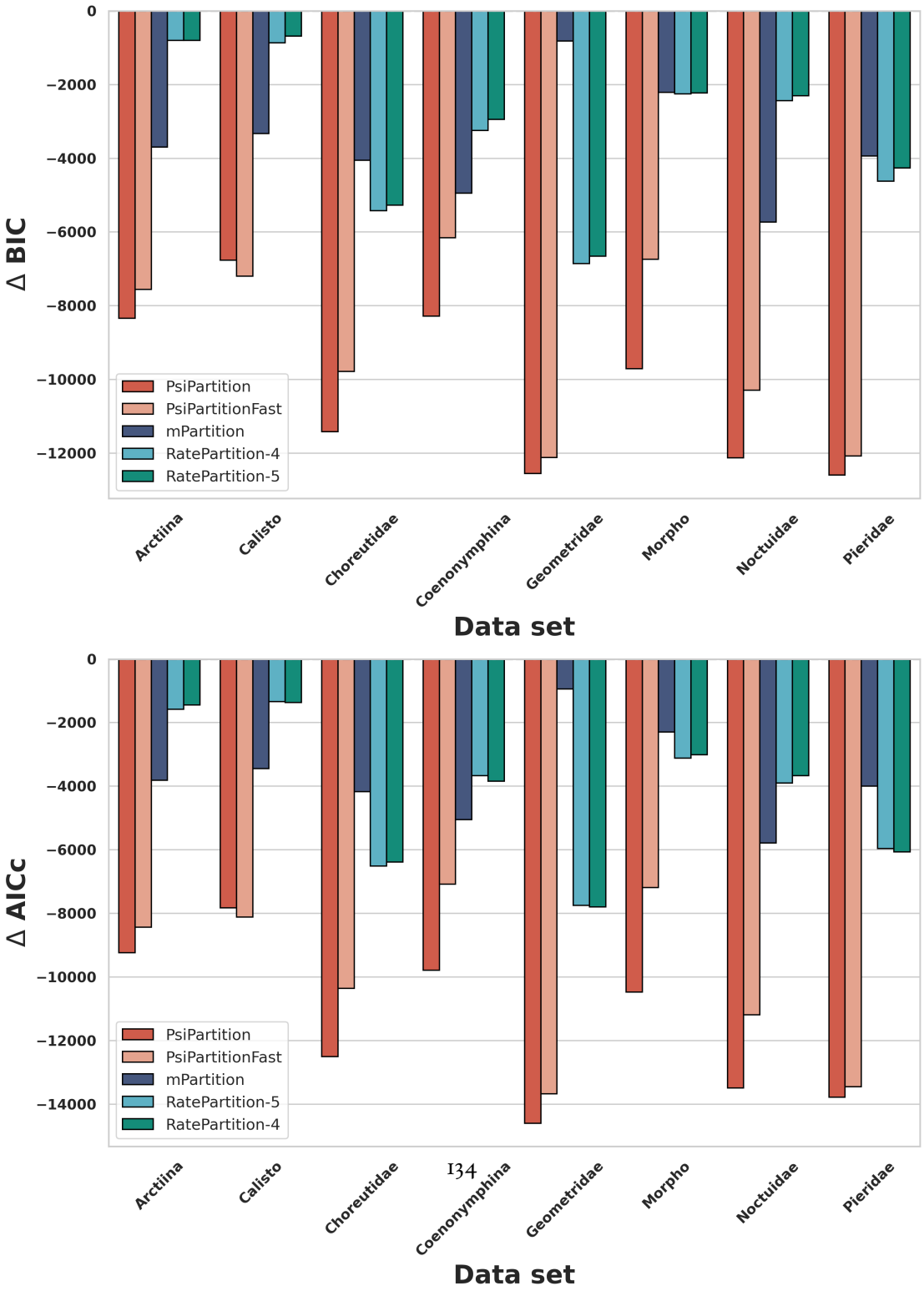


Figure 5.3: The number of sites in each bin generated by PsiPartition on the different empirical DNA data sets (Arctiina, Calisto, Choreutidae, Coenonymphina, Geometridae, Morpho, Noctuidae, and Pieridae). Different bins have different number of sites, with a long tails of bins with small number of sites. Note here a smaller bin index means a smaller PSI compared to the other bins, suggesting lower estimated evolutionary rates.

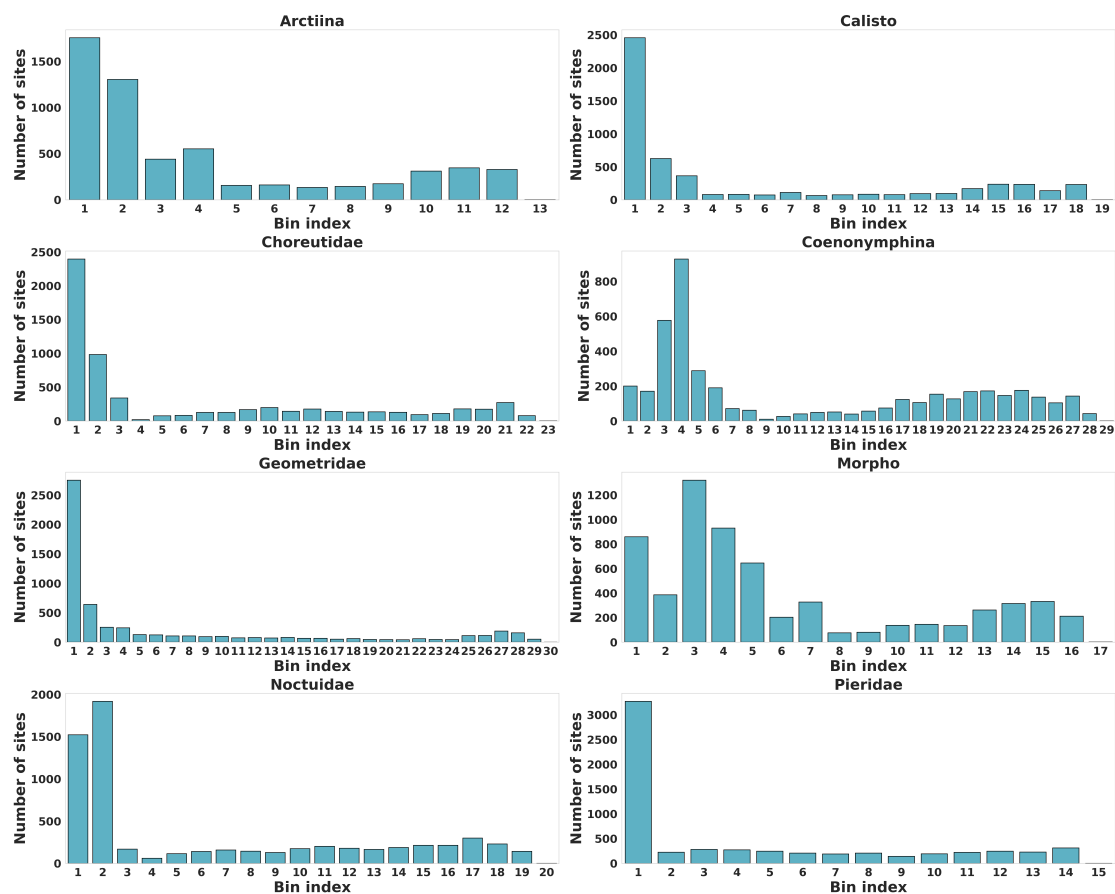


Figure 5.4: The transfer distances between the partitions generated by our proposed methods (PsiPartition and PsiPartitionFast) and other methods on different empirical DNA data sets (Arctiina, Calisto, Choreutidae, Coenonymphina, Geometridae, Morpho, Noctuidae, and Pieridae). PsiPartition and PsiPartitionFast have large transfer distances to the other methods, suggesting their partitions are significantly different from the other methods.

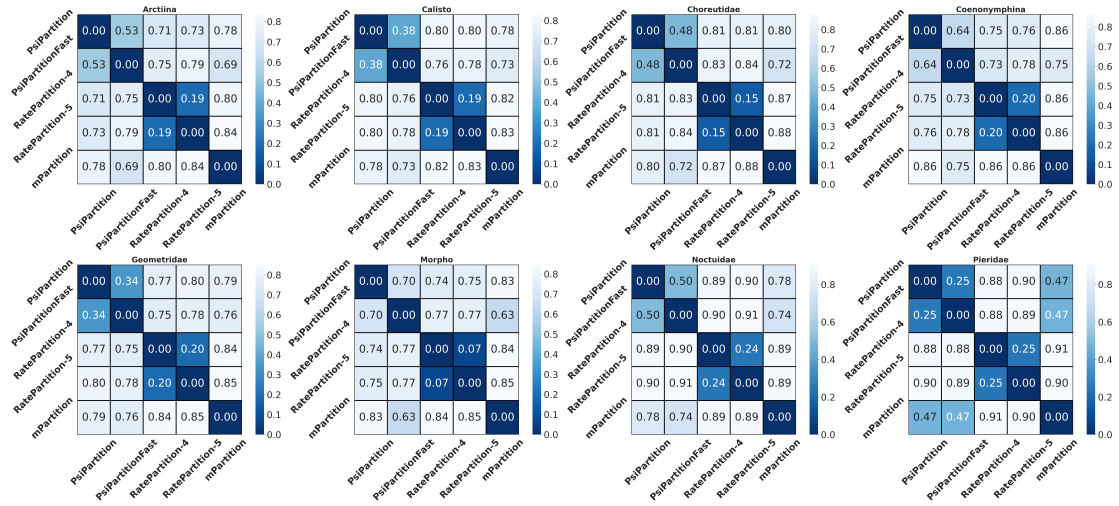


Figure 5.5: The performance comparison of different partitioning methods on the empirical protein data sets (Nguyen_2016[a-f]). Our proposed method PsiPartition achieves the best performance in terms of the ΔBIC and ΔAIC values, significantly outperforming the other methods.

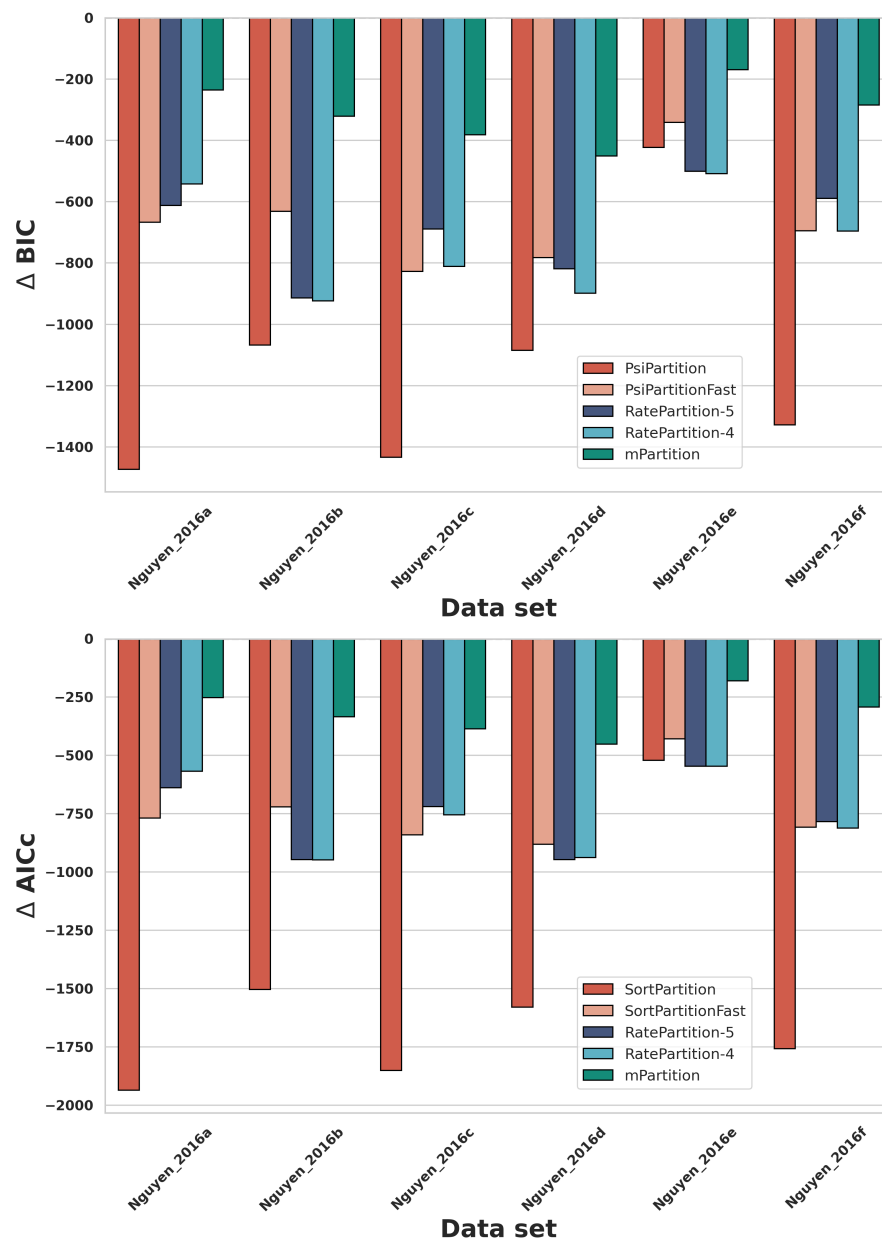


Figure 5.6: Bayesian optimization for different empirical DNA data sets. In the curves, each point (k, v) where $k = 1, 2, \dots, 30$ represent that PsiPartition achieved the minimum values of v of BIC (red) or AICc (red) in the first k iterations. The highlighted points with annotated values in the curves are the best information criterion values achieved by PsiPartition for the first time through the whole curves.

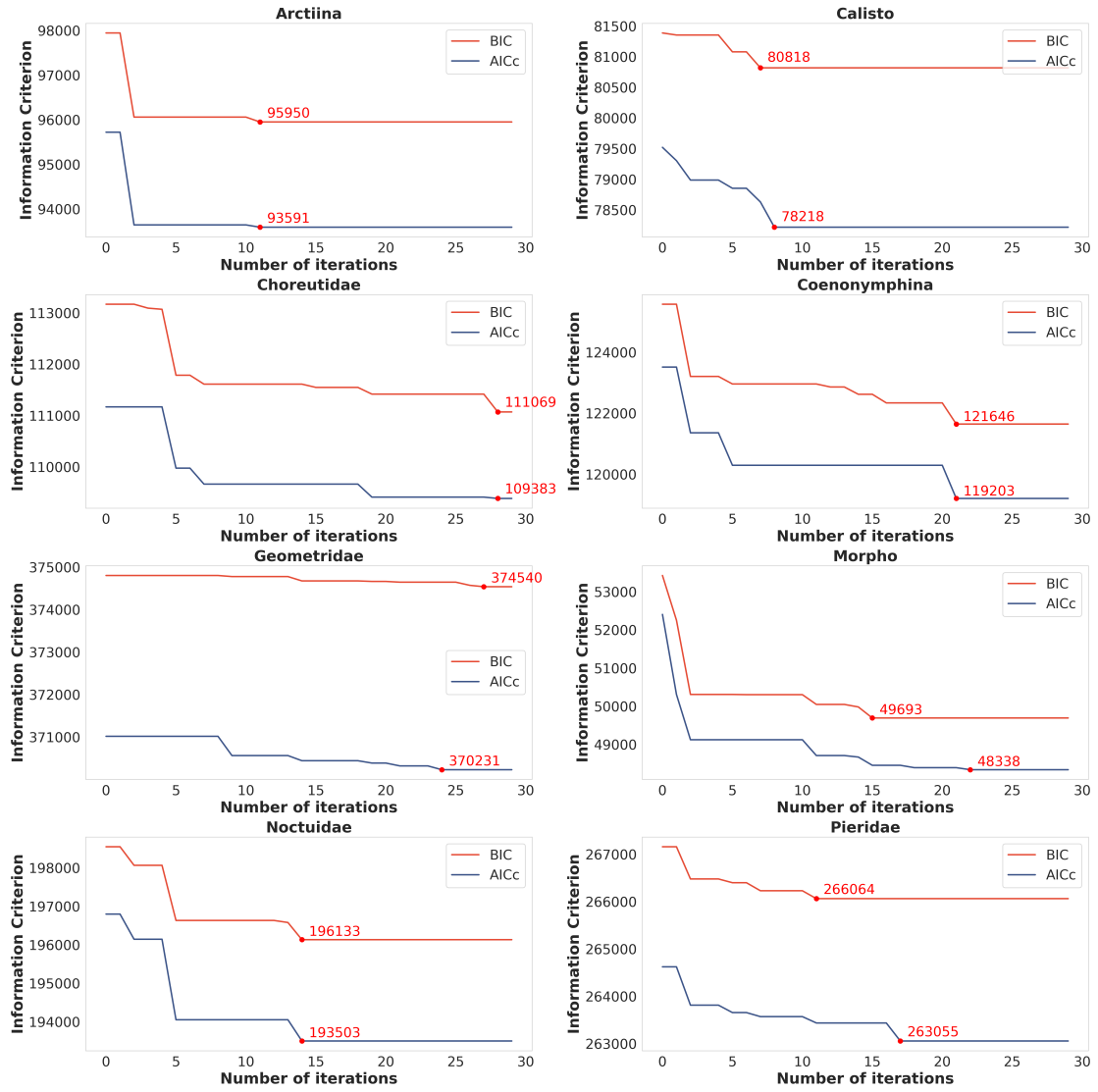


Figure 5.7: Scatter diagram of BICs produced by PsiPartition with different parameters w and k on the Morpho data set. The trajectory of Bayesian optimization is circled with black and the best Bayesian optimized BIC is the red spot circled with orange.

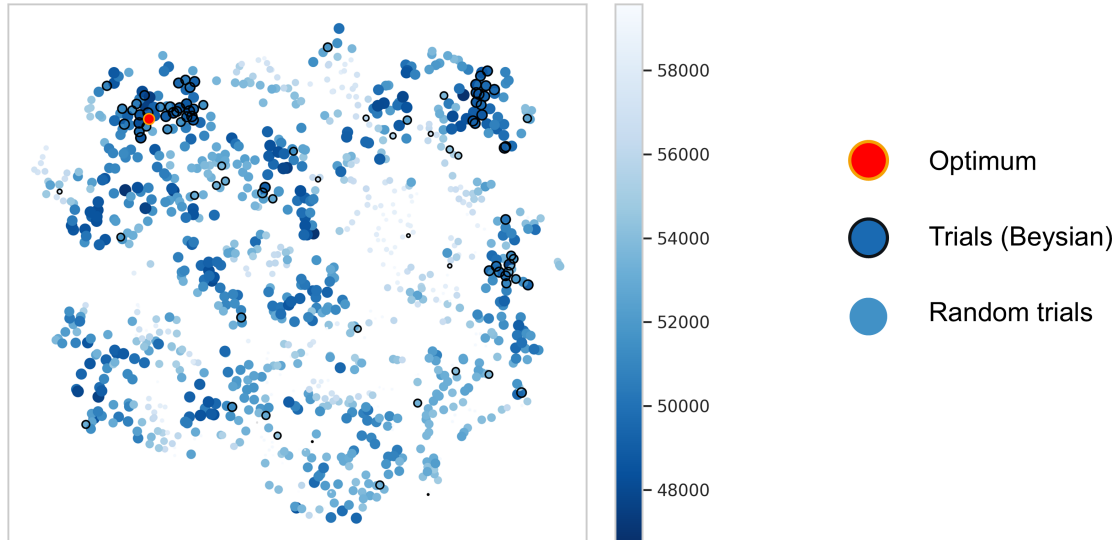


Figure 5.8: The average RF distances between the true trees and the reconstructed trees on SimDNA-1. For each different number of loci, our proposed PsiPartition (red) reconstructs the most accurate trees, with the smallest RF distances. The advantages of PsiPartition compared to other methods are significant when the number of loci is larger than 2.

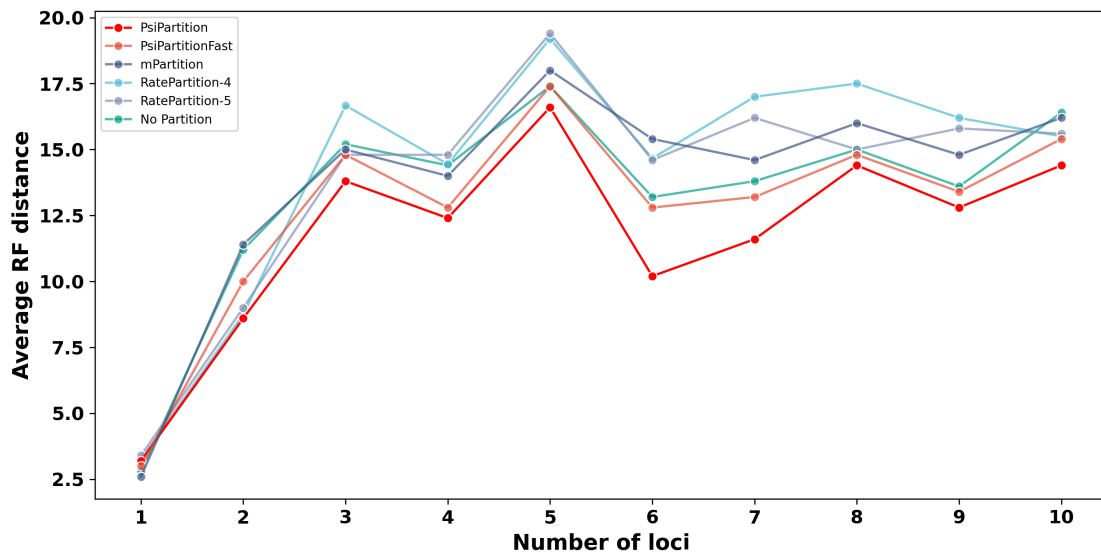


Figure 5.9: The average RF distances between the true trees and the reconstructed trees on SimDNA-2. PsiPartition (red) reconstructs the most accurate trees with the smallest RF distances for most of number of loci. The only exception when the number of loci is 10 can be explained by the randomness from the relatively small scale of the simulated data set (only 10 trees).

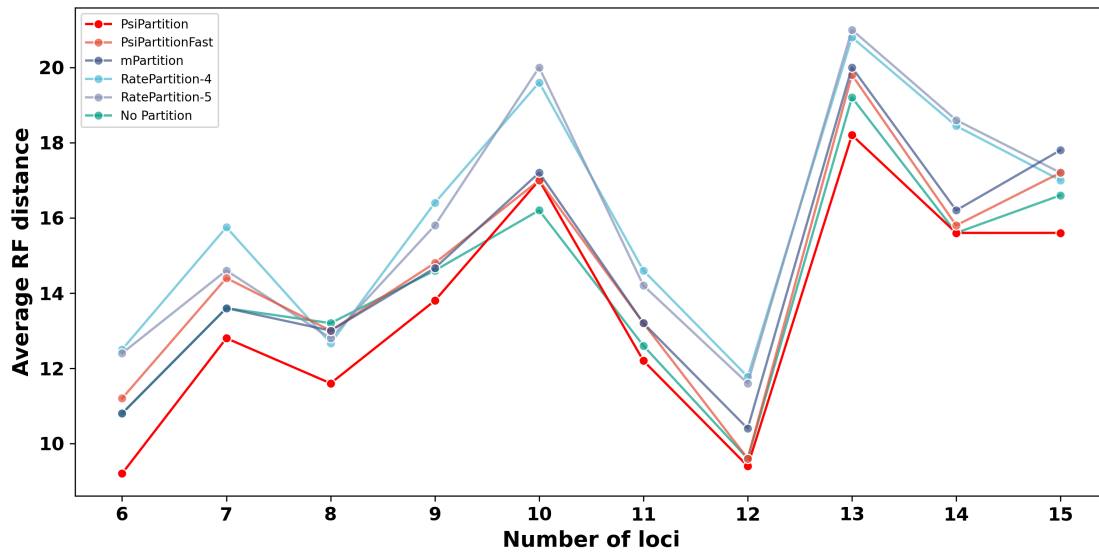


Figure 5.10: The reconstructed trees of *Arcte modesta*, *Pseudoarcte melanis*, and *Pararcte schneideriana* from phylogeny without partitioning (above) and PsiPartition (below), which shows a possible correct topology with higher bootstrap supports.

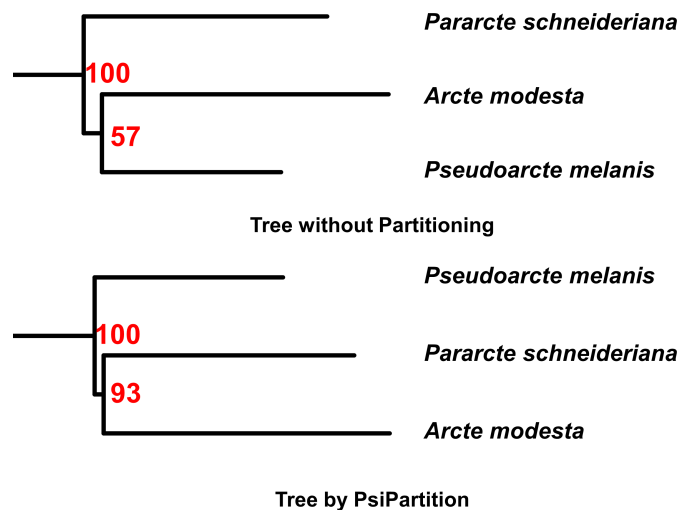


Figure 5.11: The reconstructed clades of *Diloba caeruleocephala*, *Raphia abrupta* or *sp*, *Deltote uncula*, and *Cucullia umbratica* from phylogeny without partitioning (above) and PsiPartition (below), which shows a possible correct topology with higher bootstrap supports.

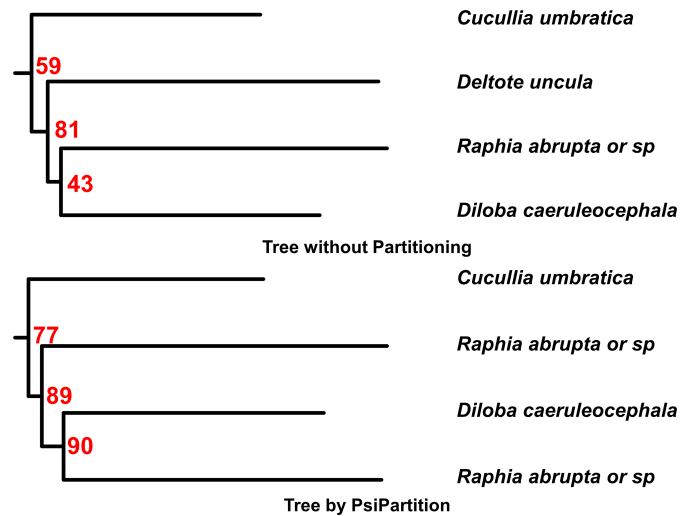
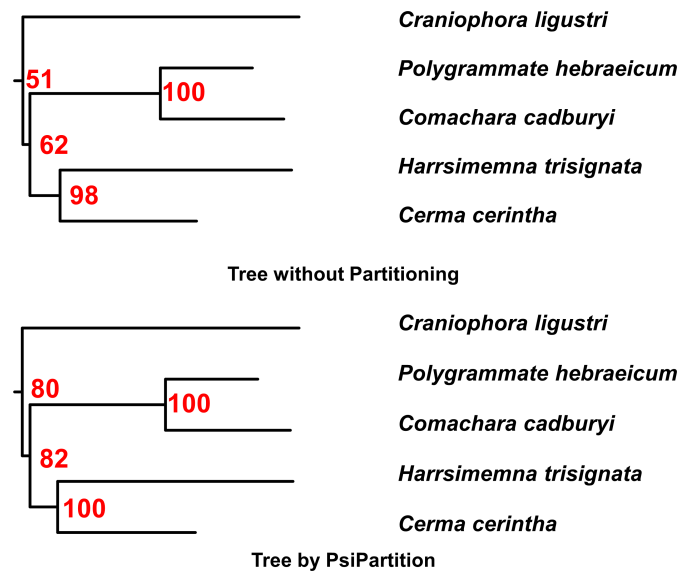


Figure 5.12: The reconstructed clades of *Craniophora ligustri*, *Polygrammate hebraeicum*, *Comachara cadburyi*, *Harrsimemna trisignata*, and *Cerma cerintha* from phylogeny without partitioning (above) and PsiPartition (below). They have the same topology yet different bootstrap supports on some branches.



EXPERIMENTAL SECTION

Algorithm 2 PsiPartitionFast

Require:

- $\mathcal{A}$: the alignment
- k_{max} : the maximum number of partitions
- n : the maximum iteration steps

Ensure:

- P^* : the optimal partition
 - s^* : the optimal parameters
 - o^* : the optimal objective value
 - 1: Uniformly sample k from $2, 3, \dots, k_{max}$
 - 2: Uniformly sample w from $[0, 1]^t$ $\triangleright t = 5$ for DNA, 21 for protein
 - 3: Let S be an empty set
 - 4: Initialize a surrogate model m
 - 5: **for** $i = 1$ to n **do**
 - 6: Let Ψ be the PSI of all sites with parameters w
 - 7: Bin the sites of $\mathcal{A}$ into k partitions based on Ψ as P_i
 - 8: Evaluate IQ-TREE with partitioning P_i and obtain BIC b_i
 - 9: $s_i \leftarrow k$
 - 10: $S \leftarrow S \cup \{(P_i, s_i, b_i)\}$
 - 11: Update m with S
 - 12: Determine the next parameters s_{i+1} by the surrogate model m with expected improvement
 - 13: **end for**
 - 14: return $(P^*, s^*, o^*) \in S$ that has the minimum BIC o^*
-

Algorithm 3 Generate binary tree

Require:

n : the number of leaves

Ensure:

T : a binary tree with n leaves

- 1: Let S be a set of n nodes
 - 2: $i \leftarrow n$
 - 3: **while** $|S| > 1$ **do**
 - 4: Randomly choose two nodes u and v from S
 - 5: Construct a tree T with u and v as its children
 - 6: Remove u and v from S and add T to S
 - 7: **end while**
 - 8: return the only tree in S
-

Table 5.4: Performance comparison of different partitioning schemes on the empirical DNA data sets.

Data set	LogL	AICc	BIC	AIC	# params	Method
Arctiina	-45624	92066	94559	92012	382	PsiPartition
Arctiina	-50345	101408	103619	101366	338	RatePartition-4
Arctiina	-51181	102857	104417	102837	237	No Partition
Arctiina	-46811	94421	96863	94370	374	PsiPartitionFast
Arctiina	-50254	101274	103620	101226	359	RatePartition-5
Arctiina	-49255	99044	100721	99021	255	mPartition
Calisto	-42277	85122	86868	85093	270	RatePartition-4
Calisto	-43048	86492	87734	86478	191	No Partition
Calisto	-38722	78178	80386	78131	343	PsiPartition
Calisto	-38808	78334	80497	78288	336	PsiPartitionFast
Calisto	-41304	83045	84408	83027	210	mPartition
Calisto	-42266	85155	87053	85120	294	RatePartition-5
Choreutidae	-57403	115381	117211	115356	275	RatePartition-5
Choreutidae	-60854	121889	122487	121886	89	No Partition
Choreutidae	-58749	117716	118434	117712	107	mPartition
Choreutidae	-55548	111493	112776	111480	192	PsiPartitionFast
Choreutidae	-54369	109263	110942	109241	252	PsiPartition
Choreutidae	-57509	115505	117065	115486	234	RatePartition-4
Coenonymphina	-62374	125319	126999	125285	268	RatePartition-5
Coenonymphina	-59465	119639	121685	119587	328	PsiPartition
Coenonymphina	-62311	125148	126705	125119	248	RatePartition-4
Coenonymphina	-61796	123941	125002	123927	168	mPartition
Coenonymphina	-64339	128991	129946	128980	151	No Partition
Coenonymphina	-60564	121834	123874	121782	327	PsiPartitionFast
Geometridae	-188014	377098	380300	377010	491	RatePartition-4
Geometridae	-184715	370950	375294	370779	674	PsiPartitionFast
Geometridae	-191587	383950	386341	383903	364	mPartition
Geometridae	-192079	384890	387159	384848	345	No Partition
Geometridae	-183521	368406	372366	368266	612	PsiPartition
Geometridae	-188009	377145	380498	377048	515	RatePartition-5
Morpho	-28203	56577	57143	56575	84	mPartition
Morpho	-22908	46236	47594	46222	203	PsiPartition
Morpho	-27741	55862	57095	55851	184	RatePartition-4
Morpho	-29365	58873	59351	58871	71	No Partition
Morpho	-25681	51657	52624	51651	144	PsiPartitionFast
Morpho	-27667	55756	57122	55743	204	RatePartition-5
Noctuidae	-96573	193837	196018	193802	328	PsiPartition
Noctuidae	-100391	201148	202340	201137	178	mPartition
Noctuidae	-101258	203271	205635	203229	356	RatePartition-4
Noctuidae	-101074	203032	205766	202974	413	RatePartition-5
Noctuidae	-103295	206937	208070	206927	169	No Partition
Noctuidae	-97610	195747	197436	195726	253	PsiPartitionFast
Pieridae	-135110	271296	274547	271210	495	RatePartition-5
Pieridae	-131490	263880	266650	263819	420	PsiPartitionFast
Pieridae	-135102	271187	274182	271115	455	RatePartition-4
Pieridae	-138384	277253	278805	277235	233	No Partition
Pieridae	-136377	273258	274870	273239	242	mPartition
Pieridae	-130590	262319	265742	262224	522	PsiPartition

Table 5.5: Performance comparison of different partitioning schemes on the empirical protein data sets.

Data set	LogL	AICc	BIC	AIC	# params	Method
Nguyen_2016a	-3979	8041	8213	8037	39	RatePartition-4
Nguyen_2016a	-3138	6674	7282	6584	154	PsiPartition
Nguyen_2016a	-3944	7971	8143	7966	39	RatePartition-5
Nguyen_2016a	-4270	8609	8755	8606	33	No Partition
Nguyen_2016a	-3858	7841	8089	7830	57	PsiPartitionFast
Nguyen_2016a	-4139	8357	8520	8352	37	mPartition
Nguyen_2016b	-5370	10882	11158	10868	64	mPartition
Nguyen_2016b	-5541	11216	11479	11203	61	No Partition
Nguyen_2016b	-5153	10496	10848	10472	83	PsiPartitionFast
Nguyen_2016b	-5059	10268	10556	10253	67	RatePartition-4
Nguyen_2016b	-4599	9712	10412	9571	186	PsiPartition
Nguyen_2016b	-5057	10269	10565	10253	69	RatePartition-5
Nguyen_2016c	-5874	11909	12238	11895	73	PsiPartitionFast
Nguyen_2016c	-6298	12749	13065	12736	70	No Partition
Nguyen_2016c	-5223	10899	11631	10799	177	PsiPartition
Nguyen_2016c	-5936	11995	12254	11986	57	RatePartition-4
Nguyen_2016c	-5930	12030	12375	12014	77	RatePartition-5
Nguyen_2016c	-6104	12363	12683	12349	71	mPartition
Nguyen_2016d	-5845	11817	12071	11807	58	PsiPartitionFast
Nguyen_2016d	-5343	11119	11769	11017	165	PsiPartition
Nguyen_2016d	-5804	11752	12034	11738	65	RatePartition-5
Nguyen_2016d	-5833	11760	11955	11754	44	RatePartition-4
Nguyen_2016d	-6087	12247	12403	12243	35	mPartition
Nguyen_2016d	-6312	12698	12854	12694	35	No Partition
Nguyen_2016e	-3388	6921	7139	6897	61	PsiPartition
Nguyen_2016e	-3399	6895	7054	6884	43	RatePartition-4
Nguyen_2016e	-3592	7262	7393	7255	35	mPartition
Nguyen_2016e	-3438	7013	7221	6992	58	PsiPartitionFast
Nguyen_2016e	-3686	7441	7562	7435	32	No Partition
Nguyen_2016e	-3397	6896	7061	6883	45	RatePartition-5
Nguyen_2016f	-4000	8137	8389	8122	61	PsiPartitionFast
Nguyen_2016f	-3983	8160	8495	8132	83	RatePartition-5
Nguyen_2016f	-3997	8133	8388	8118	62	RatePartition-4
Nguyen_2016f	-3378	7186	7755	7069	157	PsiPartition
Nguyen_2016f	-4437	8944	9084	8940	33	No Partition
Nguyen_2016f	-4288	8651	8800	8647	35	mPartition

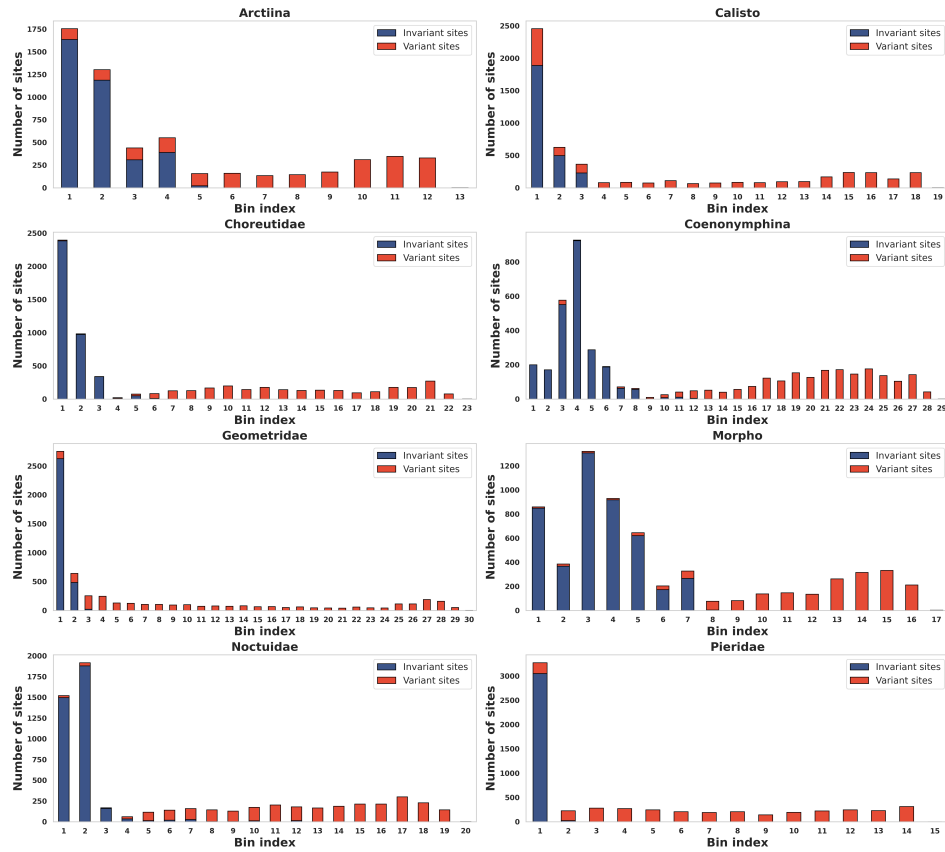
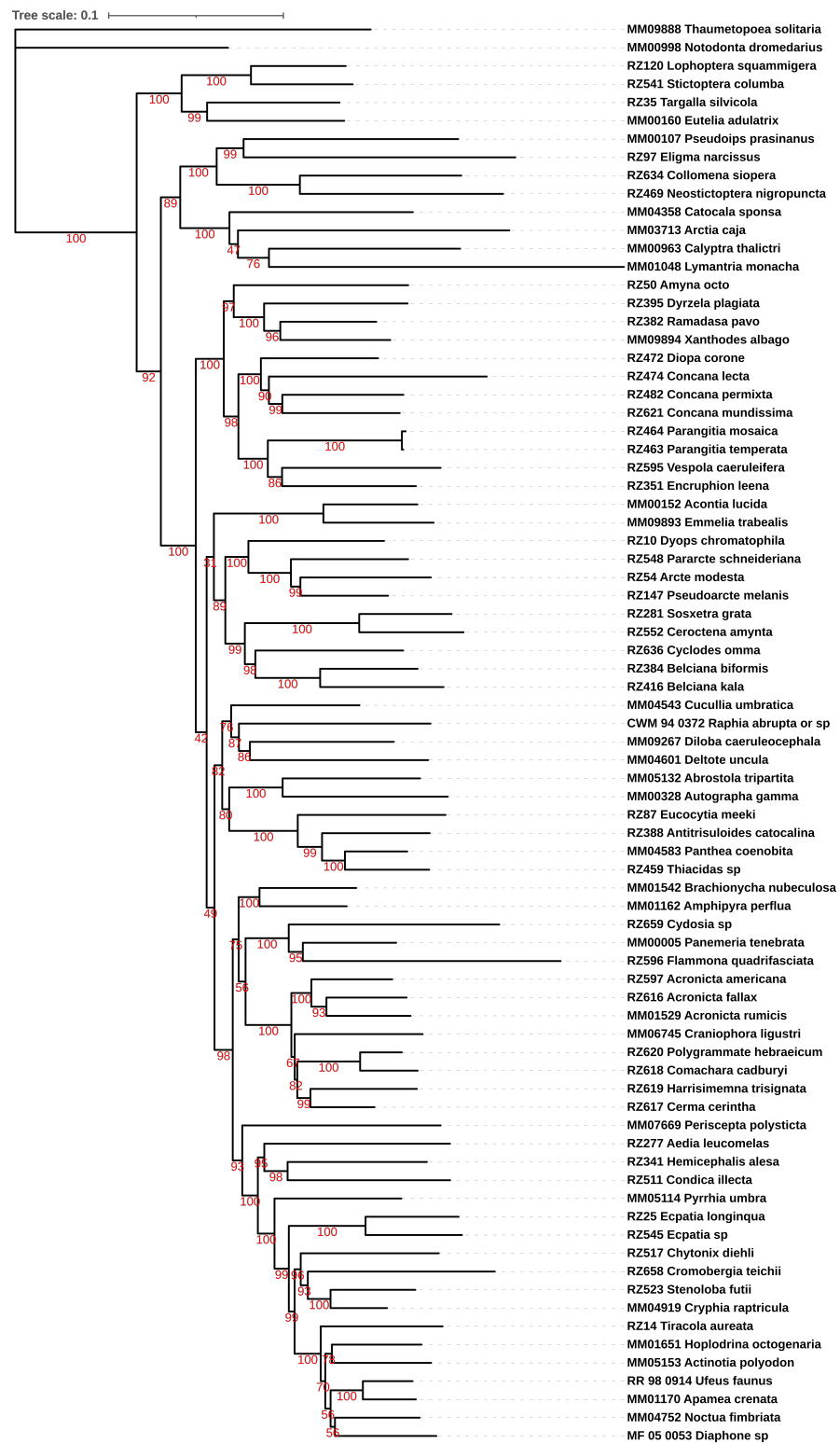


Figure 5.13: The number of sites in each bin generated by PsiPartition on the different empirical DNA data sets (Arctiina, Calisto, Choreutidae, Coenonymphina, Geometridae, Morpho, Noctuidae, Pieridae). Different bins have different number of sites and invariant and variant sites are in different colors. Note Pieridae has the first bin with the most invariant sites, while the other data sets have the last bin with the most invariant sites.



147
Figure 5.14: The reconstructed Noctuidae tree based on PsiPartition.

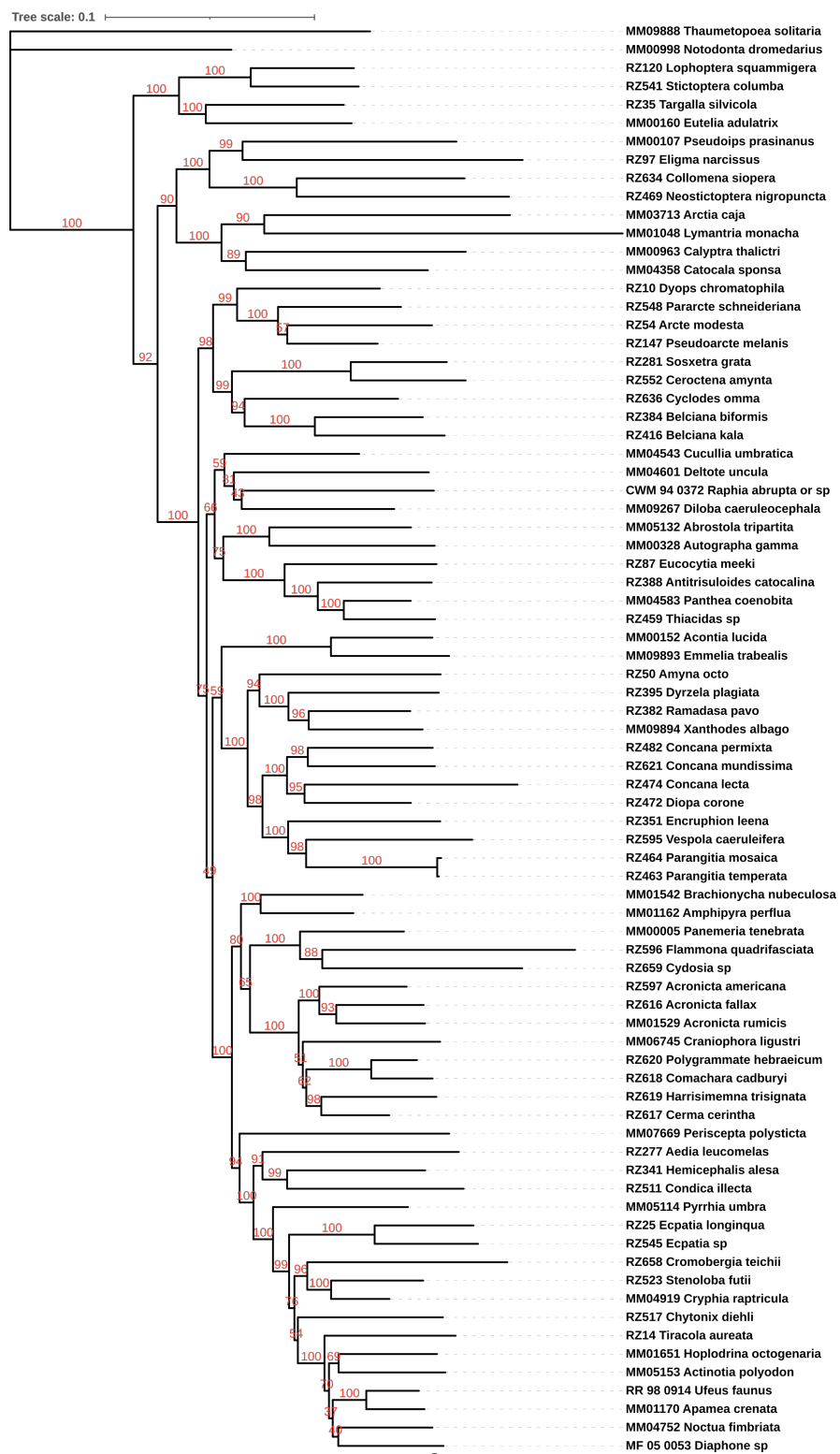


Figure 5.15: The reconstructed Noctuidae tree without partitioning.

References

- [1] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–425, 1987.
- [2] Joseph Felsenstein. Evolutionary trees from dna sequences: a maximum likelihood approach. *J. Mol. Evol.*, 17:368–376, 1981.
- [3] Bruce Rannala and Ziheng Yang. Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. *J. Mol. Evol.*, 43:304–311, 1996.
- [4] Motomu Matsui and Wataru Iwasaki. Graph splitting: a graph-based approach for superfamily-scale phylogenetic tree reconstruction. *Syst. Biol.*, 69(2):265–279, 2020.
- [5] Ziheng Yang. *Computational molecular evolution*. OUP Oxford, 2006.
- [6] Olga Chernomor, Arndt Von Haeseler, and Bui Quang Minh. Terrace aware data structure for phylogenomic inference from supermatrices. *Syst. Biol.*, 65(6):997–1008, 2016.
- [7] Michael S Rosenberg and Sudhir Kumar. Taxon sampling, bioinformatics, and phylogenomics. *Syst. Biol.*, 52(1):119, 2003.
- [8] Lee Bofkin and Nick Goldman. Variation in evolutionary processes at different codon positions. *Mol. Biol. Evol.*, 24(2):513–521, 2007.
- [9] Si Quang Le and Olivier Gascuel. Accounting for solvent accessibility and secondary structure in protein phylogenetics is clearly beneficial. *Syst. Biol.*, 59(3):277–287, 2010.
- [10] David Kainer and Robert Lanfear. The effects of partitioning on phylogenetic inference. *Mol. Biol. Evol.*, 32(6):1611–1627, 2015.
- [11] Robert Lanfear, Paul B Frandsen, April M Wright, Tereza Senfeld, and Brett Calcott. Partitionfinder 2: new methods for selecting partitioned models of evolution for molecular and morphological phylogenetic analyses. *Mol. Biol. Evol.*, 34(3):772–773, 2017.

- [12] Matthew C Brandley, Andreas Schmitz, and Tod W Reeder. Partitioned bayesian analyses, partition choice, and the phylogenetic relationships of scincid lizards. *Syst. Biol.*, 54(3):373–390, 2005.
- [13] Simon YW Ho and Robert Lanfear. Improved characterisation of among-lineage rate variation in cetacean mitogenomes using codon-partitioned relaxed clocks. *Mitochondrial DNA*, 21(3-4):138–146, 2010.
- [14] James R Leavitt, Kevin D Hiatt, Michael F Whiting, and Hojun Song. Searching for the optimal data partitioning strategy in mitochondrial phylogenomics: a phylogeny of acridoidea (insecta: Orthoptera: Caelifera) as a case study. *Mol. Phylogenet. Evol.*, 67(2):494–508, 2013.
- [15] Céline Poux, Ole Madsen, Julian Glos, Wilfried W De Jong, and Miguel Vences. Molecular phylogeny and divergence times of malagasy tenrecs: influence of data partitioning and taxon sampling on dating analyses. *BMC Evol. Biol.*, 8(1):1–16, 2008.
- [16] Jadranka Rota and Niklas Wahlberg. Exploration of data partitioning in an eight-gene data set: phylogeny of metalmark moths (lepidoptera, choreutidae). *Zool. Scr.*, 41(5):536–546, 2012.
- [17] Thu Le Kim and Vinh Le Sy. mpartition: a model-based method for partitioning alignments. *J. Mol. Evol.*, 88(8-9):641–652, 2020.
- [18] Paul B Frandsen, Brett Calcott, Christoph Mayer, and Robert Lanfear. Automatic selection of partitioning schemes for phylogenetic analyses using iterative k-means clustering of site rates. *BMC Evol. Biol.*, 15(1):1–17, 2015.
- [19] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Trans. Autom. Control*, 19(6):716–723, 1974.
- [20] Nariaki Sugiura. Further analysis of the data by akaike’s information criterion and the finite corrections: further analysis of the data by akaike’s. *Commun. Stat.- Theory Methods*, 7(1): 13–26, 1978.
- [21] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.
- [22] Robert Lanfear, Brett Calcott, Simon YW Ho, and Stephane Guindon. Partitionfinder: combined selection of partitioning schemes and substitution models for phylogenetic analyses. *Mol. Biol. Evol.*, 29(6):1695–1701, 2012.

- [23] Jeremy M Brown and Alan R Lemmon. The importance of data partitioning and the utility of bayes factors in bayesian phylogenetics. *Syst. Biol.*, 56(4):643–655, 2007.
- [24] Jimmy A McGuire, Christopher C Witt, Douglas L Altshuler, and JV Remsen. Phylogenetic systematics and biogeography of hummingbirds: Bayesian and maximum likelihood analyses of partitioned data and selection of an appropriate partitioning strategy. *Syst. Biol.*, 56(5):837–856, 2007.
- [25] Chenhong Li, Guoqing Lu, and Guillermo Ortí. Optimal data partitioning and a test case for ray-finned fishes (actinopterygii) based on ten nuclear loci. *Syst. Biol.*, 57(4):519–539, 2008.
- [26] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.
- [27] Robert Lanfear, Brett Calcott, David Kainer, Christoph Mayer, and Alexandros Stamatakis. Selecting optimal partitioning schemes for phylogenomic datasets. *BMC Evol. Biol.*, 14:1–14, 2014.
- [28] Carla A Cummins and James O McInerney. A method for inferring the rate of evolution of homologous characters that can potentially improve phylogenetic inference, resolve deep divergence and correct systematic biases. *Syst. Biol.*, 60(6):833–844, 2011.
- [29] Lauri Kaila, Marc E Epstein, Maria Heikkilä, and Marko Mutanen. The assignment of prodiactidae to hyblaeoidea, with remarks on thyridoidea (lepidoptera). *Zootaxa*, 3682(3):485–494, 2013.
- [30] Jadranka Rota and Scott E Miller. A new genus of metalmark moths (lepidoptera, choreutidae) with afrotropical and australasian distribution. *ZooKeys*, (355):29, 2013.
- [31] Maria Heikkilä, Marko Mutanen, Mari Kekkonen, and Lauri Kaila. Morphology reinforces proposed molecular phylogenetic affinities: a revised classification for gelechioidea (lepidoptera). *Cladistics*, 30(6):563–589, 2014.
- [32] Pável Matos-Maraví, Rayner Núñez Águila, Carlos Peña, Jacqueline Y Miller, Andrei Sourakov, and Niklas Wahlberg. Causes of endemic radiation in the caribbean: evidence from the historical biogeography and diversification of the butterfly genus calisto (nymphalidae: Satyrinae: Satyrini). *BMC Evol. Biol.*, 14(1):1–18, 2014.

- [33] Niklas Wahlberg, Jadranka Rota, Michael F Braby, Naomi E Pierce, and Christopher W Wheat. Revised systematics and higher classification of pierid butterflies (lepidoptera: Pieridae) based on molecular data. *Zool. Scr.*, 43(6):641–650, 2014.
- [34] Patrick P Edger, Hanna M Heide-Fischer, Michaël Bekaert, Jadranka Rota, Gernot Glöckner, Adrian E Platts, David G Heckel, Joshua P Der, Eric K Wafula, Michelle Tang, et al. The butterfly plant arms-race escalated by gene and genome duplications. *Proc. Natl. Acad. Sci.*, 112(27):8362–8366, 2015.
- [35] Niels P Kristensen, Douglas J Hilton, Axel Kallies, Liz Milla, Jadranka Rota, Niklas Wahlberg, Stephen A Wilcox, Richard V Glatz, David A Young, Glenn Cocking, et al. A new extant family of primitive moths from kangaroo island, australia, and its significance for understanding early lepidoptera evolution. *Syst. Entomol.*, 40(1):5–16, 2015.
- [36] Hossein Rajaei, Carola Greve, Harald Letsch, Dieter Stüning, Niklas Wahlberg, Joël Minet, and Bernhard Misof. Advances in geometridae phylogeny, with characterization of a new family based on pseudobiston pinratanae (lepidoptera, glossata). *Zool. Scr.*, 44(4):418–436, 2015.
- [37] Erki Ounap, Jaan Viidalepp, and Andro Truuverk. Phylogeny of the subfamily lerentiinae (lepidoptera: Geometridae): Integrating molecular data and traditional classifications. *Syst. Entomol.*, 41(4):824–843, 2016.
- [38] Jadranka Rota, Tobias Malm, Nicolas Chazot, Carlos Peña, and Niklas Wahlberg. A simple method for data partitioning based on relative evolutionary rates. *PeerJ*, 6:e5498, 2018.
- [39] Katja Rönkä, Johanna Mappes, Lauri Kaila, and Niklas Wahlberg. Putting parasemia in its phylogenetic place: a molecular analysis of the subtribe arctiina (lepidoptera). *Syst. Entomol.*, 41(4):844–853, 2016.
- [40] Ullasa Kodandaramaiah, Carlos Peña, Michael F Braby, Roger Grund, Chris J Müller, Sören Nylin, and Niklas Wahlberg. Phylogenetics of coenonymphina (nymphalidae: Satyrinae) and the problem of rooting rapid radiations. *Mol. Phylogenet. Evol.*, 54(2):386–394, 2010.
- [41] Pasi Sihvonen, Marko Mutanen, Lauri Kaila, Gunnar Brehm, Axel Hausmann, and Hermann S Staude. Comprehensive molecular sampling yields a robust phylogeny for geometrid moths (lepidoptera: Geometridae). *PLoS One*, 6(6):e20356, 2011.

- [42] Carla M Penz, Philip J Devries, and Niklas Wahlberg. Diversification of morpho butterflies (lepidoptera, nymphalidae): a re-evaluation of morphological characters and new insight from dna sequence data. *Syst. Entomol.*, 37(4):670–685, 2012.
- [43] Reza Zahiri, Donald Lafontaine, Christian Schmidt, Jeremy D Holloway, Ian J Kitching, Marko Mutanen, and Niklas Wahlberg. Relationships among the basal lineages of noctuidae (lepidoptera, noctuoidea) based on eight gene regions. *Zool. Scr.*, 42(5):488–507, 2013.
- [44] Niklas Wahlberg and Christopher West Wheat. Genomic outposts serve the phylogenomic pioneers: designing novel nuclear markers for genomic dna extractions of lepidoptera. *Syst. Biol.*, 57(2):231–242, 2008.
- [45] Andrew D Nguyen, Nicholas J Gotelli, and Sara Helms Cahan. The evolution of heat shock protein sequences, cis-regulatory elements, and expression profiles in the eusocial hymenoptera. *BMC Evol. Biol.*, 16:1–13, 2016.
- [46] Lam-Tung Nguyen, Heiko A Schmidt, Arndt Von Haeseler, and Bui Quang Minh. Iq-tree: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol. Biol. Evol.*, 32(1):268–274, 2015.
- [47] Sebastian Höhna, Michael J Landis, Tracy A Heath, Bastien Boussau, Nicolas Lartillot, Brian R Moore, John P Huelsenbeck, and Fredrik Ronquist. Revbayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Syst. Biol.*, 65(4):726–736, 2016.
- [48] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. *Advances in neural information processing systems*, 8, 1995.
- [49] Subha Kalyanamoorthy, Bui Quang Minh, Thomas KF Wong, Arndt Von Haeseler, and Lars S Jermiin. Modelfinder: fast model selection for accurate phylogenetic estimates. *Nat. Methods*, 14(6):587–589, 2017.
- [50] David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Math. Biosci.*, 53(1-2):131–147, 1981.
- [51] William HE Day. The complexity of computing metric distances between partitions. *Math. Social Sci.*, 1(3):269–287, 1981.

- [52] Lukas Biewald et al. Experiment tracking with weights and biases. *Software available from wandb. com*, 2:233, 2020.
- [53] Wandb. https://github.com/wandb/sweeps/blob/master/src/sweeps/bayes_search.py, 2024. [Online; accessed 1-April-2024].
- [54] Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006.
- [55] Bui Quang Minh, Minh Anh Thi Nguyen, and Arndt Von Haeseler. Ultrafast approximation for phylogenetic bootstrap. *Mol. Biol. Evol.*, 30(5):1188–1195, 2013.
- [56] Ivica Letunic and Peer Bork. Interactive tree of life (itol) v5: an online tool for phylogenetic tree display and annotation. *Nucleic Acids Res.*, 49(W1):W293–W296, 2021.
- [57] Sudhir Kumar, Qiqing Tao, Alessandra P Lamarca, and Koichiro Tamura. Computational reproducibility of molecular phylogenies. *Mol. Biol. Evol.*, 40(7):msad165, 2023.
- [58] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9(11), 2008.
- [59] Andrew Rambaut and Nicholas C Grass. Seq-gen: an application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. *Bioinformatics*, 13(3):235–238, 1997.

Conclusions and Future Work

Protein structure prediction has been a long-standing challenge in protein science due to its immense theoretical and practical value. Recent advances in protein structure prediction using deep learning have achieved remarkable results, with AlphaFold2 serving as a significant milestone. In the post-AlphaFold era, many open problems remain, some of which may be even more challenging. This thesis focuses on Protein Language Models (PLMs), which are the counterparts of pre-trained Large Language Models (LLMs) in protein science. The work presented here employs PLMs to address challenges in protein structure and property prediction that are difficult to solve using traditional methods.

In Chapter 2, a novel approach to intrinsically disordered protein prediction is proposed, leveraging the high-dimensional representations of PLMs. For the first time, protein language models are used to predict the intrinsic disorder of proteins. Instead of relying on traditional sequence features, these high-dimensional representations enable the model to capture essential information from single sequences, achieving state-of-the-art performance in terms of both accuracy and speed for the IDP prediction task. In Chapter 3, a novel pK_a predictor based on single-residue representations from PLMs is introduced. The topic of pK_a prediction has long suffered from a lack of data, and existing methods are insufficiently accurate. PLMs help address this issue by providing high-quality representations of residues. The proposed method achieves state-of-the-art performance in pK_a prediction and is several orders of magnitude faster than existing methods. This work demonstrates the structure-awareness of PLMs and their potential in protein property prediction tasks. In Chapter 4, a novel two-step metal-binding site prediction method is proposed. The PLM-based sequence predictor accurately and efficiently scans binding residues, while a pre-trained structure predictor removes false-positive binding sites and refines binding positions. Compared to existing methods, this approach achieves state-of-the-art performance in the metal-binding site prediction task. These findings suggest that structural information remains indispensable for certain protein property prediction tasks, and the integration of sequence and structure information is a promising direction for future research. In Chapter 5, a novel partitioning method based on parameterized sorting indices (PSI) is proposed to improve phylogeny reconstruction. PSI is a new approach to represent the heterogeneity between different sites in biological sequence data. The proposed method significantly improves the accuracy of phylogeny

reconstruction compared to existing methods.

In conclusion, the work presented in this thesis highlights the immense potential of PLMs in various protein structure prediction tasks. As powerful tools for accurate and efficient protein modeling, PLMs are poised to play a pivotal role in addressing challenges in protein science, including protein structure prediction, property prediction, and function prediction.

List of Publications

1. Shijie Xu and Akira Onoda*. “Accurate and Fast Prediction of Intrinsically Disordered Protein by Multiple Protein Language Models and Ensemble Learning.” *Journal of Chemical Information and Modeling* 64.7 (2023): 2901-2911. <https://doi.org/10.1021/acs.jcim.3c01202>
2. Shijie Xu and Akira Onoda*. “PsiPartition: Improved Site Partitioning for Genomic Data by Parameterized Sorting Indices and Bayesian Optimization.” *Journal of Molecular Evolution* 92: 874-890. <https://doi.org/10.1007/s00239-024-10215-7>
3. Shijie Xu and Akira Onoda*. “Accurate and Rapid Prediction of Protein pK_a : Protein Language Models Reveal the Sequence- pK_a Relationship.” Under review.
4. Shijie Xu and Akira Onoda*. PRIME: Probe-Based Identification of Metal Binding Sites of Proteins Using Deep Representations. To be submitted.

THIS THESIS WAS TYPESET using L^AT_EX, originally developed by Leslie Lamport and based on Donald Knuth's T_EX. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. A template that can be used to format a PhD dissertation with this look & feel has been released under the permissive AGPL license, and can be found online at github.com/suchow/Dissertate or from its lead author, Jordan Suchow, at suchow@post.harvard.edu. The template used in this dissertation is created based on Harvard Template, which was previously used in the dissertation of Hongjie Zhai, who is a former student of Hokkaido University. The logo of Hokkaido University is used under the [University Logo Guideline](#).