MEMBRANE PROTEIN CLASSIFICATION WITH PROTEIN LANGUAGE MODELS

HAMED GHAZIKHANI

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE AND SOFTWARE ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

SEPTEMBER 2024

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Mr. Hamed Ghazikhani

Entitled: Membrane Protein Classification with Protein Language Models

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Chunyan Lai

_____ External Examiner
Dr. Anthony J. Kusalik

_____ Examiner
Dr. Sabine Bergler

_____ Examiner
Dr. Tristan Glatard

_____ Examiner
Dr. Ré Mansbach

_____ Supervisor
Dr. Gregory Butler

Approved _____
Chair of Department or Graduate Program Director

_____ 20 _____ _____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Membrane Protein Classification with Protein Language Models

Hamed Ghazikhani, Ph.D.
Concordia University, 2024

This thesis investigates the application of Protein Language Models (PLMs) to enhance the classification of membrane proteins, which are crucial for cellular functions and pharmacological targeting but challenging to characterize due to their context within a membrane. We employ PLMs derived from Large Language Models of natural language processing, including ProtBERT, ProtT5, ESM1b, ESM2, and Ankh. These PLMs are pretrained using self-supervised learning on extensive datasets such as UniRef50 (40 million proteins) and BFD (2 billion proteins).

Our research comprises four interconnected projects focused on discriminating membrane proteins, transport proteins, and ion channels from proteins not in those classes. We use established state-of-the-art (SOTA) tools with standard datasets for training and testing as a baseline for evaluating our work.

The first project demonstrates that fine-tuning is beneficial in classifying membrane proteins, with a fine-tuned combination of ProtBERT-BFD and logistic regression (LR) outperforming SOTA. The second project shows that Convolutional Neural Networks (CNNs) are superior to traditional classifiers when used with PLMs for membrane protein, transport protein and ion channel classification, again surpassing SOTA performance.

In the third project, we evaluate six PLMs and six downstream classifiers across three tasks, considering fine-tuned and frozen representations, dataset balance, and floating-point precision. ESM-1b emerges as the top performer across most tasks and metrics. We confirm that fine-tuning outperforms frozen representations, imbalanced datasets work best, and there is no statistically significant difference between half- and full-precision computations.

The fourth project incorporates secondary structure information into Ankh. Evaluation across multiple tasks shows little statistically significant difference between Ankh and the modified PLM with secondary structure information.

The tools developed in this research now represent the state-of-the-art in membrane protein classification. Our methodological findings provide insights into PLM applications for protein classification in general, with particular relevance to membrane proteins highly relevant to drug discovery.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Gregory Butler, for his invaluable guidance, support, and mentorship throughout my research journey. Your insights and wisdom have greatly enriched my work and academic growth.

I am deeply grateful to the members of my thesis committee, Dr. Sabine Bergler, Dr. Tristan Glatard, and Dr. Ré Mansbach, for their constructive feedback, challenging questions, and valuable suggestions that have significantly improved the quality of my research.

Special thanks go to my colleagues, including Sima Ataei, Stuart Thiel, and Steve Morse, for their assistance, collaborative spirit, and insightful discussions that have contributed to a stimulating research environment.

To my family, especially my parents and my brother, Hamid: your unwavering love, encouragement, and support have been a constant source of strength throughout my academic career. Though distance separated us, your love and support bridged that gap perfectly. A heartfelt thank you to Louis for his unwavering support throughout my PhD journey. Your commitment, patience, and assistance have been vital to both my academic work and personal well-being.

I am deeply grateful for the support of such amazing people. Thank you all for making this achievement possible.

Contents

List of Figures

List of Tables

Glossary

Amino acid The basic building blocks of proteins, consisting of a central carbon atom bonded to an amino group, a carboxyl group, a hydrogen atom, and a variable side chain

Cell membrane Biological membrane that surrounds the cytoplasm of living cells, physically separating the intracellular components from the extracellular environment.

BERT-architecture A transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google, designed to pre-train deep bidirectional representations from unlabeled text.

BLAST (Basic Local Alignment Search Tool) A sequence similarity search program that can be used to quickly search against databases of sequences and find regions of similarity between biological sequences. It is widely used in bioinformatics for identifying homologous sequences and predicting gene functions.

Convolutional Neural Network (CNN) A class of deep neural networks, most commonly applied to analyzing visual imagery, extensively used in image and video recognition, recommender systems, and classification tasks.

Cross-Validation A model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. Commonly used in settings where the goal is prediction and one wants to estimate how accurately a predictive model will perform in practice.

Dataset Imbalance Occurs when the number of samples in different classes are unevenly distributed. This often biases the training process of a classifier, leading to poorer performance on the minority class.

Downstream task A specific application or problem that uses a pre-trained model as a starting point, often requiring fine-tuning or additional training

Embedding A dense vector representation of discrete input data, such as words or amino acids, in a continuous vector space

Evolutionary Scale Modeling (ESM) A protein language model framework that utilizes evolutionary information to predict protein structure and function with high accuracy.

Feed-Forward Neural Networks (FFNN) A type of artificial neural network wherein connections between the nodes do not form a cycle. This network architecture is extensively used in pattern recognition and classification tasks.

Fine-Tuning The process of adjusting the weights of a pre-trained model to better fit the specific data or task at hand, commonly used in deep learning to adapt general models to more specialized tasks.

Frozen In machine learning, particularly when using pre-trained models, ”frozen” refers to keeping certain layers or the entire pre-trained model fixed (i.e., not updating their weights) during training on a new task. This approach treats the pre-trained model as a fixed feature extractor, with only new task-specific layers being trained.

Gene Ontology (GO) A major bioinformatics initiative to unify the representation of gene and gene product attributes across all species. It provides a controlled vocabulary of terms for describing gene product characteristics and gene product annotation data.

Gradient descent An optimization algorithm used to minimize a loss function by iteratively moving in the direction of steepest descent

Homoeostasis Property of a system in which variables are regulated so that internal conditions remain stable and relatively constant.

Homologous The existence of shared ancestry between a pair of structures, or genes, in different species.

Hydrophilic Interacting effectively with water.

Hydrophobic Not interacting effectively with water; in general, poorly soluble or insoluble in water.

Hyperparameter A parameter whose value is set before the learning process begins, distinguishing it from parameters that are learned during training

Ion Channel A pore-forming membrane protein that allows ions to pass through the channel pore passively, down their electrochemical gradient. Ion channels are essential for the electrical activity of cells and are distinct from ion transporters in that they do not require energy for ion movement.

Ion Transporter A membrane protein that actively moves ions across cellular membranes, often against their concentration gradient. Unlike ion channels, transporters require energy (often in the form of ATP) to function and can move ions in both directions. They are crucial for maintaining cellular ion balance and homeostasis.

Masked language modeling A pre-training objective where the model learns to predict masked tokens in a sequence based on the surrounding context

Masked tokens In the context of language models, these are intentionally hidden parts of the input sequence that the model is trained to predict based on the surrounding context. This technique is used in pre-training to help the model learn contextual relationships in the data.

Mathews Correlation Coefficient (MCC) A correlation coefficient between the observed and predicted binary classifications; it provides a measure of the quality of binary classifications, perfect prediction being 1.

Membrane Protein Protein that is part of, or interacts with, the biological membrane. These proteins play crucial roles in various cellular processes, including signaling, transport, and structural support.

Polypeptide A chain of amino acids linked by peptide bonds, forming the primary structure of proteins

Potts models Statistical mechanics models that describe interacting spins on a lattice, generalizing the Ising model to more than two spin states. In computational biology, particularly protein science, Potts models are used to capture higher-order dependencies between amino acid positions in protein sequences. They can represent both direct and indirect interactions between residues, making them valuable for studying protein structure, function, and evolution.

Pre-training The process of training a model on a large dataset for a general task before fine-tuning it for a specific application

ProtBERT-BFD A protein language model based on the BERT architecture, specifically fine-tuned using the Big Fantastic Database (BFD) for enhanced prediction of protein sequences and functions.

Protein Language Model (PLM) Computational models that apply concepts from natural language processing to understand and predict protein structures and functions based on their amino acid sequences.

Protein sequence The unique sequence of amino acids that characterizes a given protein.

Secondary Structure Refers to the local conformation of some part of a protein's polypeptide chain, including alpha helices and beta sheets, which form due to hydrogen bonding between backbone atoms.

Self-attention A mechanism in neural networks that allows the model to weigh the importance of different parts of the input when processing a specific element

Structural Biology A branch of molecular biology, biochemistry, and biophysics concerned with the molecular structure of biological macromolecules like proteins and nucleic acids, and how changes in structure affect function.

Tokenization The process of breaking down a sequence of text or amino acids into individual units (tokens) for processing by a machine learning model

TooT-BERT-C Tool developed to classify ion channel proteins from non-ion channel proteins using BERT-based language models.

TooT-BERT-CNN-C Classification tool combining BERT and convolutional neural networks to identify ion channel proteins.

TooT-BERT-CNN-M Method integrating BERT and convolutional neural networks for membrane protein classification.

TooT-BERT-CNN-T Hybrid model using BERT and convolutional neural networks to classify transport proteins.

TooT-BERT-M BERT-based approach for distinguishing membrane proteins from non-membrane proteins.

TooT-BERT-T BERT-derived model for classifying transport proteins from non-transport proteins.

TooT-PLM-ionCT Multi-model framework utilizing six protein language models and various classifiers to differentiate ion channels, ion transporters, and membrane proteins.

TooT-PLM-P2S Protein language model incorporating secondary structure information, evaluated on eight diverse protein-related datasets.

Transfer learning A machine learning technique where a model trained on one task is repurposed on a second related task

Abbreviations

AAC Amino Acid Composition

ANOVA Analysis of Variance

BERT Bidirectional Encoder Representations from Transformers

BFD Big Fantastic Database

BLAST Basic Local Alignment Search Tool

CNN Convolutional Neural Network

COG Clusters of Orthologous Groups

CV Cross-Validation

DS-C Dataset for Ion Channel Protein Prediction

DS-M Dataset for Membrane Protein Prediction

DS-T Dataset for Transporter Protein Prediction

EA Enrichment Analysis

ESM Evolutionary Scale Modeling

FFNN Feed-Forward Neural Network

FN False Negative

FP false positive

GO Gene Ontology

IC Ion Channel

IMP Integral Membrane Proteins

IT Ion Transporter

kNN k-Nearest Neighbor

LR Logistic Regression

MAST Motif Alignment and Search Tool

MCC Matthews Correlation Coefficient

MEME Multiple EM for Motif Elicitation

MLM Masked Language Modeling

MP Membrane Proteins

MSA Multiple Sequence Alignment

NLP Natural Language Processing

OET-kNN Optimized Evidence-Theoretic k-Nearest Neighbor

PDB Protein Data Bank

PFP Protein Function Prediction

PLM Protein Language Model

ProtT5 ProtTrans using T5 Model

PseAA Pseudo Amino Acid

Pse-PSSM Pseudo Position-Specific Scoring Matrix

PSSM Position-Specific Scoring Matrices

RF Random Forest

RoPE Rotary Position Embedding

SOTA State-of-the-Art

SSP Secondary Structure Prediction

SVM Support Vector Machine

TAPE Tasks Assessing Protein Embeddings

TN True Negative

TP True Positive

UMAP Uniform Manifold Approximation and Projection

ZOOPS Zero or One Occurrence per Sequence

Chapter 1

Introduction

Bioinformatics plays a crucial role in protein prediction, with significant implications for healthcare, drug discovery, and understanding biological processes [LGG01]. Membrane proteins (MPs), comprising approximately one-third of all cellular proteins [WH98, Qui02], are of particular interest due to their critical functions in physiological processes and their importance as pharmacological targets [YGC+07]. Despite their significance, MPs remain among the least characterized proteins [OALH06] due to their structural complexity and the experimental challenges associated with their study [MESW+14].

The challenges in characterizing MPs necessitate the development of advanced computational methods to predict and analyze their structure and function. These methods are essential for understanding MPs' roles in cellular processes and disease mechanisms, as well as for identifying new therapeutic targets and strategies for treating complex diseases. Protein informatics [Kit02] and deep learning techniques [LBH15] offer a promising approach to overcoming these limitations and gaining insights into MPs' functions and interactions.

This study aims to enhance the accuracy and efficiency of protein informatics approaches for MPs by utilizing deep learning and protein language models (PLMs) [EESE+23, EHD+21, RMS+21, RBT+19, RLV+21, AKB+19]. By combining computational techniques with biological knowledge, this research seeks to develop frameworks for MP classification and functional annotation, including ion transporters and channels. The integration of secondary structure data into PLMs represents a novel approach to advancing protein prediction methods, potentially facilitating drug discovery, disease diagnosis, and therapy [HEW+19, RMS+20].

The potential impact of improved MP prediction extends beyond bioinformatics, potentially revolutionizing drug discovery and personalized medicine approaches. By addressing current challenges in membrane protein analysis through an interdisciplinary approach, this research aims to provide new insights into the functions and interactions of MPs, contributing significantly to bioinformatics, computational biology, and biomedical sciences.

1.1 Context

1.1.1 Pre-history Review

The field of membrane protein prediction has seen significant advancements since the late 1990s. Early methods primarily focused on amino acid (AA) composition [CAPPQ97], with Chou and Elrod (1999) [CE99] pioneering protein prediction using AA composition combined with covariant discriminant analysis (CDA). To address the loss of sequence information in this approach, Chou (2001) [Cho01] introduced PseAA (Pseudo Amino Acid) composition, which incorporates sequence-order information along with traditional AA composition.

Following these initial developments, researchers began combining PseAA with various techniques. These included Support Vector Machines (SVM) [CZC03, WYL+04], a machine learning model effective for classification tasks, and Supervised Locally Linear Embedding (SLLE)

[WYXC05], a dimensionality reduction technique. The incorporation of hydrophobic-hydrophilic interactions [CC05a] and Fourier spectrum analysis [LWC05, LYW+05] further enhanced the feature extraction process. This period saw a proliferation of innovative approaches, such as the GO-PseAA method [CC05b], which combined Gene Ontology (a standardized vocabulary for gene and gene product attributes) and PseAA, and the Optimized Evidence Theoretic k-Nearest Neighbor (OET-kNN) classifier [SC05].

As the field progressed, more sophisticated methods emerged. Researchers [PGLL07] began using Position-Specific Scoring Matrices (PSSM), which capture evolutionary information by representing the probability of each amino acid occurring at each position in a sequence. They also developed web servers like MemType-2L [CS07], making these prediction tools more accessible to the scientific community. The incorporation of physiochemical properties of amino acids into models and the application of Discrete Wavelet Transform (DWT) for feature extraction further improved prediction accuracy [HKY12].

Recent years (2010-2014) saw a trend towards combining multiple feature extraction methods, employing dimensionality reduction techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [WLW+12], and utilizing ensemble classifiers [HK13]. There has also been a growing emphasis on integrating evolutionary information into prediction models [HK12].

This evolution demonstrates a clear trend from relatively simple AA composition methods to more complex, multi-feature approaches leveraging advanced machine learning techniques. These developments consistently aimed at improving the accuracy and reliability of membrane protein prediction, with particular success in classifying membrane proteins into types (e.g., transmembrane, lipid-anchored) and predicting specific functions like ion channel activity.

However, challenges persisted. Many methods struggled with limited training data, particularly for less common membrane protein types [Alb20]. The computational cost of some advanced techniques also posed limitations for large-scale applications. Additionally, while accuracy improved, interpreting the biological significance of complex feature combinations became increasingly difficult [BRK17].

These limitations, along with the exponential growth in available protein sequence data, set the stage for the transition to deep learning approaches [LBH15, SOPK18, TO19] and protein language models (PLMs) [EHD+21, LAR+23, RLV+21, EESE+23] in recent years. PLMs, inspired by natural language processing techniques [FH22], can leverage vast amounts of unlabeled sequence data, potentially capturing more nuanced patterns than traditional feature engineering approaches. This shift represents a new paradigm in the field, building upon the foundational work of these earlier methods while addressing some of their key limitations. Figure 1 shows a timeline of significant milestones of protein function prediction.

However, before delving into the revolutionary impact of PLMs, it is crucial to understand the state-of-the-art (SOTA) methods that immediately preceded them, particularly in the context of membrane (M), transporter (T), and ion channel (C) protein classification.

1.1.2 Pre-PLM SOTA for M/T/C Classification

Before the advent of protein language models, several machine learning approaches showed promising results in classifying membrane proteins and transporters. Among these, the work of Alballa et al. [AB20a, AB20b] stands out for its comprehensive analysis and methods. This section will focus on their contributions, particularly the TooT-M [AB20a] for membrane protein classification and TooT-T [AB20b] for transporter classification, and their performance compared to other state-of-the-art methods of that time.

Membrane protein prediction: TooT-M [AB20a] is an integrative approach developed for predicting membrane proteins. It combines two distinct methods: transmembrane topology prediction using TOPCONS2 [TPS+15] and a machine learning ensemble method utilizing

Pse-PSSM (Pseudo Position-Specific Scoring Matrix) encoding with OET-kNN [SC05] (Optimized Evidence-Theoretic k-Nearest Neighbor) classifiers.

The approach leverages TOPCONS2 for its high specificity in distinguishing signal peptides from transmembrane regions. The machine learning component employs an ensemble of 500 OET-kNN classifiers (OET-kNN V500) with a selective voting mechanism. This mechanism uses mRMR [PLD05] (minimum redundancy maximum relevance) to choose an optimal subset of 20 classifiers, effectively reducing noise and increasing the ensemble's distinctive power. This combination strategy proved highly effective, achieving impressive performance metrics on the training set using leave-one-out cross-validation: 91.47% sensitivity, 94.90% specificity, 93.21% accuracy, and an MCC (Matthews Correlation Coefficient) of 0.8645.

When compared to state-of-the-art methods (Table 1), TooT-M consistently outperformed both MemType-2L [CS07] and iMem-2LSAAC [AHJ18] across multiple datasets. On the DS-M dataset [AB20a], TooT-M achieved 92.46% accuracy, surpassing MemType-2L (89.44%) and iMem-2LSAAC (79.27%). It also demonstrated superior performance on the datasets DS1 and DS2 used by these competing methods, achieving 97.43% accuracy on DS1 (compared to iMem-2LSAAC's 94.61%) and 93.57% accuracy on DS2 (versus MemType-2L's 92.7%).

Table 1: TooT-M comparison on DS-M

Method	**sensitivity**	**specificity**	**accuracy**	**MCC**
MemType-2L [CS07]	88.67	90.19	89.44	0.79
iMem-2LSAAC [AHJ18]	74.52	83.9	79.27	0.59
TooT-M [AB20a]	**92.41**	**92.5**	**92.46**	**0.85**

This table compares the performance of TooT-M with other state-of-art methods on the DS-M dataset. The highest performance in each metric is highlighted in bold.

Transporter prediction: TooT-T [AB20b] is an ensemble classifier designed to distinguish transporter membrane proteins from other proteins. It combines two distinct methods: homology annotation transfer using BLAST [AGM$^+$90] against the TCDB database [STB06], and machine learning using SVM models with novel protein encodings. The project introduces a new protein encoding method called "psi-composition" (including psiAAC, psiPAAC, and psiPseAAC), which combines traditional amino acid composition with evolutionary information obtained from PSI-BLAST [AMS$^+$97] searches. This novel encoding outperforms other methods, including the commonly used PSSM.

The ensemble approach of TooT-T utilizes stacked generalization (stacking) to combine predictions from six base classifiers. These include three SVM models using the psi-composition encodings and three homology-based predictions with different thresholds. A Gradient Boosting Machine (GBM) serves as the meta-classifier to produce the final prediction.

TooT-T achieves 90.07% accuracy and 0.80 MCC in cross-validation, and 92.22% accuracy and 0.82 MCC in independent testing (Table 2). These results surpass some of the other state-of-the-art methods in the field, except the work of Li et al. [LD11]. The goal of TooT-T was to predict novel transporters by using only the protein sequence. Whereas in Li et al. [LD11] they used GO annotations as features as well.

The key advantage of TooT-T lies in its ability to exploit the low correlation between different prediction methods. It effectively balances the high accuracy of machine learning approaches with the different perspective provided by homology-based methods. This makes TooT-T particularly effective for detecting novel and unannotated transporter proteins.

Table 2: TooT-T comparison

Tool	**sensitivity**		**specificity**		**accuracy**		**MCC**	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
SCMMTP [LVY+15]	80.00	83.76	68.33	77.68	76.11	81.12	0.47	0.62
TrSSP [MCZ14]	76.67	76.67	81.67	78.46	80.00	78.99	0.57	0.58
Ho et al. [CC05b]	100.00	83.14	77.50	84.48	85.00	83.94	0.73	0.68
Li et al. [LD11]	96.67	99.50	95.83	97.44	96.11	98.33	0.91	0.97
TooT-T [AB20b]	**94.17**	**90.15**	**88.33**	**89.97**	**92.22**	**90.07**	**0.82**	**0.80**

This table presents the comparison of TooT-T with other SOTA on the DS-T dataset. Li et al. is not in bold as their high performance relies on GO annotations, while TooT-T aims to predict novel, unannotated transporters.

Ion channel prediction: MFPS_CNN [NHTO22] (Multi-filter Pattern Scanning from Position-specific Scoring Matrix with Convolutional Neural Network) represents the state-of-the-art approach for ion channel and transporter prediction. Developed by Nguyen et al. in 2022, this method leverages the power of convolutional neural networks applied to PSSMs of protein sequences.

The key innovation of MFPS_CNN lies in its use of multi-window scanning filters. Unlike traditional CNNs that use fixed-size filters, MFPS_CNN employs multiple convolutional filters of varying lengths (2, 4, 8, 16, 24, 32, and 60 amino acids) to scan the PSSM. This approach allows the model to capture motifs and patterns at different scales within the protein sequence. Following the convolutional layers, a one-max pooling operation is applied to extract the most salient features.

MFPS_CNN demonstrated superior performance compared to previous methods, including DeepIon [TO19], on the task of classifying ion channels, ion transporters, and other membrane proteins. Table 3 shows that MFPS_CNN outperforms DeepIon across all three classification tasks. For ion channel prediction, MFPS_CNN achieves an accuracy of 95.5% and an MCC of 0.63, compared to DeepIon's 86.53 accuracy and 0.37 MCC. Similar improvements are observed for ion transporter and membrane protein classification.

Table 3: MFPS_CNN and DeepIon comparison

Method	Ion Channels		Ion Transporters		Membrane Proteins	
	Acc (%)	MCC	Acc (%)	MCC	Acc (%)	MCC
DeepIon [TO19]	86.53	0.37	83.78	0.37	86.43	0.51
MFPS_CNN [NHTO22]	**95.5**	**0.63**	**92.0**	**0.56**	**92.1**	**0.69**

This table compares MFPS_CNN and DeepIon the SOTA of ion channel prediction.

1.1.3 History of Deep Learning in Protein Science

Deep learning has revolutionized the field of protein structure and function prediction in recent years [WZL+20, MLY17]. One of the most significant breakthroughs came with AlphaFold2 [JEP+21], developed by DeepMind, which achieved unprecedented accuracy in predicting 3D protein structures at the CASP13 competition [AlQ19]. The method has the accuracy of experimental techniques. AlphaFold2 utilizes deep residual networks to predict distances between amino acid residues from multiple sequence alignments, then converts these predictions into a protein-specific potential to generate 3D structures through gradient descent.

Beyond structure prediction, deep learning has found numerous applications in protein sequence analysis and functional classification. DeepLoc [AASS+17] employs convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to predict subcellular localization from protein sequences. For secondary structure prediction, methods like SPIDER3 [HPL+18] use bidirectional recurrent neural networks to achieve high accuracy. DeepEC [RKL19] applies CNNs to predict enzyme commission numbers, while DeepGOPlus [KH20] combines

CNNs with sequence similarity techniques to predict Gene Ontology terms.

These advances have been enabled by key innovations in deep learning architectures and approaches. The use of deep residual networks and attention mechanisms [VSP+17] has allowed models to capture long-range interactions in protein sequences effectively. Combining evolutionary information from multiple sequence alignments with deep learning has proven particularly powerful [RLV+21, JEP+21, LJY+21]. Many methods now aim for end-to-end learning, predicting structure and function directly from sequences [EHD+21, HEW+19].

The impact of these developments has been substantial. They have dramatically improved the accuracy of structure and function prediction from sequence data alone, enabling large-scale annotation of newly sequenced proteins. This is providing new insights into protein folding mechanisms and the determinants of protein function. However, challenges remain, particularly in predicting from very limited sequence data, accurately modeling large multi-domain proteins, and balancing the use of evolutionary information with learning from individual sequences.

As the field continues to advance rapidly, deep learning is enabling major breakthroughs in our ability to predict protein structure and function from sequence data. This is having significant impacts across structural biology and functional genomics, opening new avenues for understanding protein biology and designing proteins with novel functions.

1.1.4 History of Language Models in Protein Science

Language models (LMs) have evolved significantly over the years, starting with early statistical n-gram models for text [OBL21]. As deep learning gained prominence, neural network-based language models emerged, beginning with feed-forward networks and progressing to recurrent neural networks (RNNs) and long short-term memory (LSTM) networks. In 2017, the introduction of the Transformer [VSP+17] architecture revolutionized natural language processing with its attention mechanism. The following year, Google introduced BERT [DCLT19] (Bidirectional Encoder Representations from Transformers), which used masked language modeling to pre-train deep bidirectional representations and achieved state-of-the-art results on many NLP tasks.

Researchers soon began applying similar techniques to protein sequences, treating amino acids as ”words” in the ”language” of proteins. In 2020, ProtBERT-BFD [EHD+21] was introduced as a BERT-like model pre-trained on a large corpus of protein sequences (BFD - Big Fantastic Database) [JEP+21], showing improved performance on various protein prediction tasks. Also in 2020, ProtT5 [EHD+21] was developed, based on the T5 [RSR+20] (Text-to-Text Transfer Transformer) architecture, demonstrating state-of-the-art performance on many protein prediction tasks and outperforming previous models.

Table 4: Comparison of Protein Language Models on Various Tasks

Task	Ankh	ProtT5-XL-U50	ESM-1b	ESM-2 (3B)	ESM-2 (15B)
Structural Tasks					
Secondary Structure Prediction (CASP12)	**83.8%**	83.4%	79.6%	83.3%	83.2%
Contact Prediction (ProteinNet L/5)	**73.2%**	69.2%	50.1%	52.7%	54.7%
Fold Prediction	**61.1%**	57.6%	57.6%	60.5%	56.7%
Functional Tasks					
Embedding-based Annotation Transfer	**71.7%**	71.0%	64.5%	65.6%	65.4%
Fluorescence Prediction	**0.62**	0.58	0.50	0.48	0.55
Solubility Prediction	**76.4%**	74.4%	67.3%	74.9%	60.4%
Localization Prediction	**83.2%**	**83.2%**	80.0%	82.4%	81.8%

This table compares PLMs given from Ankh work where most tasks shown here relate to structural aspects and are primarily focused on globular proteins. Our work extends these capabilities to membrane proteins as well. Ankh consistently performs at or near the top across various tasks, demonstrating its effectiveness as a protein language model.

In 2023, the Ankh [EESE+23] model was introduced as a highly efficient protein language model. Ankh achieved comparable or better results than previous models while using significantly

fewer parameters, making it more accessible for broader research applications. This progression illustrates how techniques originally developed for text processing have been successfully adapted and applied to protein sequences, leading to significant advancements in protein structure and function prediction. Table 4 presents a comparison of the PLMs on different tasks.

1.1.5 PLM Applications to Functional Classification

Protein language models have been increasingly applied to functional classification tasks in recent years. Alley et al. [AKB+19] demonstrated that embeddings from an mLSTM model trained on UniRef50 could be used to predict protein stability, function, and other properties. Rives et al. [RMS+21] showed that embeddings from large transformer models pre-trained on up to 250 million UniRef50 sequences could be used to predict various protein properties and functions, including structure and mutational effects, often generalizing well to unseen proteins. Rao et al. [RBT+19] evaluated protein embeddings from several PLMs on tasks including secondary structure prediction, contact prediction, and remote homology detection, showcasing their versatility. Littmann et al. [LHD+21] demonstrated that protein language model embeddings could effectively predict binding residues for various ligand classes, outperforming traditional methods using multiple sequence alignments. These studies collectively highlight the growing importance and effectiveness of protein language models in various aspects of protein functional classification. Table 5 shows a comparison on different PLMs.

Table 5: Comparison of Protein Language Models

Model	Parameters	Pre-training Data	Embedding Dim	Architecture	Year
TAPE [RBT+19]	38M	UniRef50	768	BERT	2019
SeqVec [HEW+19]	93M	UniRef50	1024	ELMo	2019
ProtBERT-BFD [EHD+21]	420M	BFD	1024	BERT	2021
ProtXLNet [EHD+21]	409M	UniRef100	1024	XLNet	2021
ESM-1b [RMS+21]	650M	UniRef50	1280	BERT	2021
ProtT5-XL-U50 [EHD+21]	3B	BFD + UniRef50	1024	T5	2021
ESM-2 (650M) [LAR+23]	650M	UniRef50/BFD	1280	BERT	2022
Ankh base [EESE+23]	726M	UniRef50	768	T5	2023
Ankh large [EESE+23]	1.9B	UniRef50	1536	T5	2023

This table presents some of the SOTA of protein language models (PLMs) with their corresponding pre-training dataset, embedding dimension and the architecture. Dim refers to dimension.

1.1.6 Challenges in Protein Function Prediction

Despite the significant advancements in protein function prediction (PFP), several challenges persist [ZBC+22]. The complex nature of protein functions, including the prevalence of multifunctional proteins with multiple functional domains, continues to pose significant challenges in computational biology [HBM+23]. The intricate interactions between amino acids and their resultant structural conformations add a layer of complexity that current models struggle to interpret effectively.

One of the primary challenges in PFP is the persistent difficulty in achieving high prediction performance [TO19]. This is largely due to the multifaceted nature of protein functions, which can be influenced by subtle sequence variations, structural elements, and environmental factors. Current models, while increasingly sophisticated, still struggle to capture all these nuances effectively.

Another significant challenge is the scarcity of well-annotated data [BCF+07], particularly for rare or specialized protein functions. Deep learning techniques in PFP require extensive, high-quality datasets for training [WSY+24]. However, acquiring such datasets is challenging, especially for less common protein types or functions. This data scarcity can lead to biased or incomplete models, limiting their generalizability and real-world applicability.

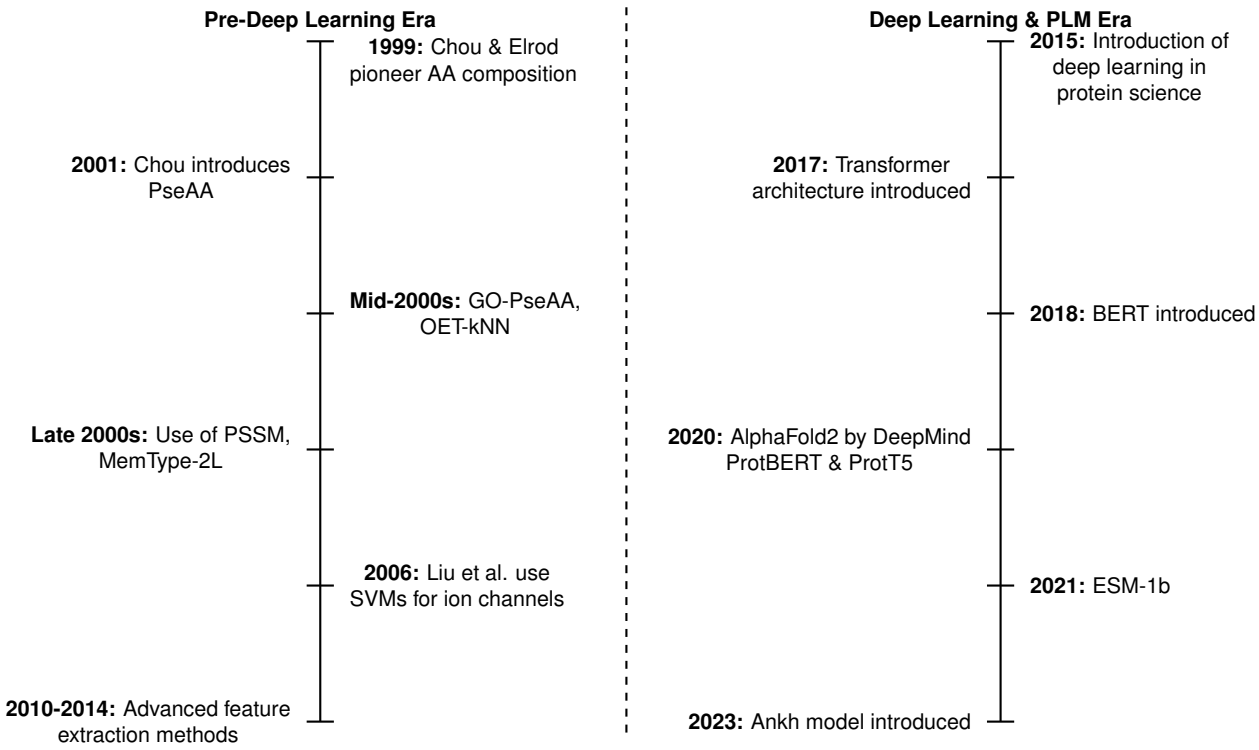

Figure 1: Timeline of significant milestones

This figure illustrates the evolution of protein functional classification methods over time, divided into two eras: pre-deep learning and the deep learning & protein language model (PLM) era.

Furthermore, the field faces challenges in interpreting and explaining the predictions made by complex models, particularly deep learning models. This ”black box” [ZHS+23] nature can limit the biological insights that can be gained from these predictions and may hinder their acceptance in some research contexts.

1.2 Process of PLMs

The development and application of PLMs typically involves two main phases: **pre-training** and downstream task application [EHD+21].

In the pre-training phase, a PLM is constructed through self-supervised learning, often using a masked language modeling (MLM) [DCLT19] approach. This process involves training on massive protein sequence datasets, often containing billions of amino acids. The ”masked” aspect of MLM refers to the technique where a portion of the input amino acids are randomly hidden or ”masked” during training. The model then learns to predict these masked amino acids based on the surrounding context. Due to the sheer scale of data and model complexity, pre-training requires high-performance computing resources. During this phase, the model learns general protein sequence features without the need for labeled data [EESE+23, EHD+21].

Once pre-trained, PLMs can be applied to specific downstream tasks through supervised learning. These tasks typically use much smaller, task-specific datasets, often in the thousands of samples, and require significantly less computational effort than pre-training, especially when using frozen representations [AKB+19].

There are two common approaches for applying PLMs to downstream tasks. The first is traditional training, where the PLM is used as a fixed feature extractor, and task-specific classifiers are trained on these **frozen** representations. The second approach is **fine-tuning**, which involves modifying the PLM itself during training on the downstream task. While fine-tuning often achieves better performance, it requires more computation than using frozen representations.

The pre-training phase allows PLMs to capture general protein sequence knowledge, while

the downstream phase adapts this knowledge to specific biological tasks [RMS+21]. This two-step process enables PLMs to leverage vast amounts of unlabeled sequence data to improve performance on specialized tasks with limited labeled data. For more detailed information on these processes, readers are referred to Chapter 2.

1.2.1 Challenges in PLM Development and Application

While PLMs have shown great promise, their development and application are not without challenges. In the pre-training phase, one of the main challenges is the **computational cost**. Training PLMs on massive protein sequence datasets requires significant computational resources, often necessitating the use of high-performance computing clusters. This can limit the accessibility of PLM development to well-resourced institutions. However, the pre-trained models can be produced at an institution with significant computational resources, then replicated and distributed to other institutions (whether or not they have significant computational resources) for downstream tasks.

Another challenge in pre-training is the potential for **bias** in the training data. If the protein sequence datasets used for pre-training are not sufficiently diverse or representative, the resulting PLM may have limited generalizability across different types of proteins or organisms.

In the downstream application phase, a key challenge is the potential for **overfitting** when fine-tuning PLMs on small, task-specific datasets. This is particularly problematic given the **scarcity** of well-annotated data for many protein functions.

There is also the challenge of **catastrophic forgetting**, where fine-tuning on a specific task can cause the model to lose some of the general knowledge it acquired during pre-training. This can limit the model's performance on other tasks or its ability to generalize to new proteins.

Moreover, the **choice between using frozen representations and fine-tuning** presents its own set of trade-offs. While frozen representations are computationally efficient, they may not capture task-specific nuances as effectively as fine-tuning. On the other hand, fine-tuning can lead to better performance but requires more computational resources and may be prone to overfitting on small datasets.

Lastly, there is the ongoing challenge of **model interpretability**. While PLMs can achieve high performance on many tasks, understanding why they make certain predictions remains difficult. This limits our ability to extract biological insights from these models and can hinder their acceptance in some research contexts.

Addressing these challenges is crucial for advancing the field of protein function prediction using PLMs. This research aims to tackle some of these issues by exploring various PLM architectures, investigating different downstream task approaches, and integrating additional structural information to enhance model performance and interpretability.

1.3 Research Questions and Objectives

This thesis aims to enhance our understanding of MPs and improve our capabilities to predict their properties and functions using PLMs. MPs are essential for many physiological functions and represent a significant fraction of pharmacological targets [OALH06, ANFS09]. Due to their complex structures and diverse functionalities, MPs present a significant challenge in bioinformatics [Alb20]. Our research objectives are addressed through the following key questions:

Q1) Can PLMs outperform state-of-the-art classifiers for membrane proteins (M), transporters (T), and ion channels (C) classification? Which PLM-based approaches are most effective for these specific tasks, and how does their performance compare to existing methods?

This question seeks to evaluate whether PLMs can outperform state-of-the-art classifiers for protein tasks, particularly for membrane protein, transporter, and ion channel classification. We will investigate various PLMs and analyze their performance in these specific tasks. Our objective is to develop methodologies to represent these protein types more accurately in computational

models by leveraging advancements in PLMs.

Q2) How can we best combine PLMs with downstream machine learning (ML) or deep learning (DL) classifiers for protein tasks? Should we use frozen representations or fine-tuning approaches, and which ML/DL classifiers are most appropriate?

This question explores optimal strategies for utilizing PLMs in downstream tasks, considering: Frozen representations vs fine-tuning approaches, selection of appropriate ML/DL classifiers, and potential issues such as catastrophic forgetting during fine-tuning. We aim to address the challenge of limited annotated data by exploring transfer learning strategies to maximize the available data.

Q3) Are the effectiveness of PLMs and their optimal utilization strategies universal across different protein classification tasks, or do they vary depending on the specific task (e.g., M, T, C classification)?

We will examine whether the effectiveness of PLMs and their optimal utilization strategies vary across different protein classification tasks, including membrane (M), transporters (T), and ion channels (C) protein classification. A key goal is to improve the predictive accuracy for these protein types, which play crucial roles in cellular processes and have significant potential as drug targets [Ash21].

Q4) What types of protein-related information are captured in PLMs, and how can we effectively incorporate additional information, such as secondary structure, to improve their performance on protein classification tasks, particularly for membrane proteins, transporters, and ion channels?

We will focus on incorporating secondary structure information, which represents a "middle ground" between the primary structure (sequence) typically used in pre-training and the tertiary structure (3D) that has been explored in some recent work [HWS$^+$24, SHZ$^+$24]. This approach aims to bridge the gap between primary sequence analysis and the complex reality of protein functionality influenced by three-dimensional structures.

1.4 Contributions

This study contributes to the field of bioinformatics, specifically in membrane protein classification using Protein Language Models (PLMs). We demonstrated that fine-tuning PLMs improves performance in membrane protein classification tasks compared to using frozen PLM embeddings.

Additionally, we developed a hybrid architecture combining PLMs with Convolutional Neural Networks (CNNs), which enhanced classification accuracy for membrane proteins compared to PLMs alone. Our experiments integrating secondary structure information into sequence-based models did not show statistically significant improvement over the baseline Ankh model [EESE$^+$23], suggesting the need for larger test sets or refined methods for incorporating structural information in future research.

Contrary to expectations, we observed that slightly imbalanced datasets produced more robust membrane protein classifiers, a finding that may inform future dataset design in protein classification tasks. To facilitate further research, we have made available a curated dataset of membrane proteins, including transporters and ion channels, with verified annotations, as well as our fine-tuned PLMs for membrane protein classification. These contributions provide methodologies, insights, and resources for advancing protein classification accuracy and elucidating the relationship between protein sequence and function.

1.5 Organization

Chapter 2 provides background on membrane proteins, protein representations, and machine learning classifiers. Chapter 3 evaluates ProtBERT-BFD in membrane protein classification tasks. Chapter 4 advances protein transport and ion channel classification using CNN classifier

integration. Chapter 5 exploits Protein Language Models for precise classification of ion channels and transporters with in-depth analysis. Chapter 6 integrates secondary structure information into Protein Language Models. Chapter 7 concludes the thesis, summarizing key findings and outlining future research directions. Appendices provide additional information, with Appendix A listing 10 publications from this doctoral study.

Chapter 2

Background

This chapter provides background information on membrane proteins, protein representations, and machine learning classifiers relevant to this thesis. It is organized into several key sections:

First, we present an overview of membrane proteins and their biological significance. The chapter then explores the BERT (Bidirectional Encoder Representations from Transformers) model, a cornerstone in the development of Protein Language Models (PLMs). We discuss its architecture, including attention mechanisms, input representations, and the processes of pre-training and fine-tuning. Special attention is given to ProtBert-BFD, a BERT-based model specifically designed for protein sequence analysis.

Following this, we examine advanced Language Models (LMs) and PLMs, including the T5 model, the ESM project, and other notable PLMs such as TAPE, MSA-Transformer, Ankh, and ProstT5.

The final section of the chapter focuses on machine learning techniques for protein sequence analysis. We provide an overview of various classifiers, including Support Vector Machines, k-Nearest Neighbors, Random Forests, Feed-Forward Neural Networks, Logistic Regression, and Convolutional Neural Networks. We also discuss evaluation metrics and statistical significance analysis methods used in this field.

2.1 Membrane Proteins

Cell membranes are fundamental structures in all living organisms, with membrane proteins comprising approximately 30% of cellular proteins [Qui02]. The cell membrane consists primarily of a lipid bilayer and associated proteins, which together regulate permeability and facilitate various biological functions [Edi03].

The lipid bilayer is composed of two layers of phospholipid molecules, with hydrophobic tails facing inward and hydrophilic heads facing outward [Yea16]. This structure provides a dynamic interface for membrane proteins, which are categorized into two main types: peripheral membrane proteins and integral membrane proteins (IMPs) [Sti16].

Peripheral proteins attach to the membrane surface through electrostatic forces or hydrogen bonds and can be dissociated under mild conditions [Sti16]. In contrast, IMPs are embedded within the hydrophobic core of the bilayer, requiring more disruptive methods for extraction due to their deep integration [WW99].

Transmembrane proteins, a significant category of IMPs, span the entire lipid bilayer. These proteins can incorporate one or multiple transmembrane segments (TMS) that typically adopt either α-helix or β-sheet conformations [ANFS09]. Surface-bound proteins, including peripheral and lipid-anchored proteins, interface with the lipid bilayer without penetrating its hydrophobic core [ANFS09].

Chou and Elrod's [CE99] classification system organizes membrane proteins into eight structural types, ranging from single-pass to multipass configurations, including lipid-anchored

and peripheral proteins. Membrane proteins can be classified into eight types based on location and structure:

1. single-pass type I
2. single-pass type II
3. single-pass type III
4. single-pass type IV
5. multipass
6. lipid-anchored
7. GPI-anchored (glycosylphosphatidylinositol-anchored)
8. peripheral membrane proteins

This structural classification aids in understanding the specific functions these proteins perform. Functionally, membrane proteins can be categorized into four main groups [ANFS09]:

1. Transporters: Regulate the selective entry and exit of ions and molecules across the cellular membrane.
2. Receptors: Detect signaling molecules and initiate intracellular signal transduction pathways.
3. Enzymes: Catalyze essential biochemical reactions on the membrane surface.
4. Anchor proteins: Maintain cellular architecture and play roles in cell adhesion and intra-cellular communication.

This structural and functional diversity of membrane proteins is crucial for maintaining cellular integrity, responding to environmental signals, and executing vital biochemical processes.

2.2 BERT

The Bidirectional Encoder Representations from Transformers (BERT) [DCLT19] model, introduced by researchers at Google AI, represents a significant breakthrough in the field of natural language processing. BERT emerged in a context of rapid progress in NLP, driven by advancements in deep learning and the availability of large datasets.

Prior to BERT, key developments in NLP included word embeddings like Word2Vec [MCCD13], which allowed words to be represented as dense vectors capturing semantic relationships. This was followed by advances in neural network architectures such as recurrent neural networks (RNNs) and long short-term memory networks (LSTMs). However, these approaches had limitations. Word embeddings were context-independent, while RNNs and LSTMs struggled with long-range dependencies in text.

The introduction of the Transformer [VSP+17] architecture in 2017 was a major breakthrough, allowing for more effective modeling of long-range dependencies through its self-attention mechanism. This paved the way for powerful language models that could be pre-trained on large corpora of unlabeled text.

BERT innovatively pre-trains deep bidirectional representations from unlabeled text by simultaneously conditioning on both left and right contexts in all layers of the model. This architecture leverages a self-attention mechanism, which allows the model to integrate context from all surrounding words in a sequence, enhancing its ability to understand language nuances [DCLT19].

Unlike traditional unidirectional models that only predict each word from the words before it, BERT processes each word in the context of all words in a sentence, both before and after it. This bidirectional approach is crucial for developing a richer understanding of language structure [DCLT19]. Figure 2 illustrates the operational stages of BERT, showcasing how the pre-trained model can be fine-tuned with just one additional output layer to perform exceptionally across a multitude of downstream tasks, such as text classification.

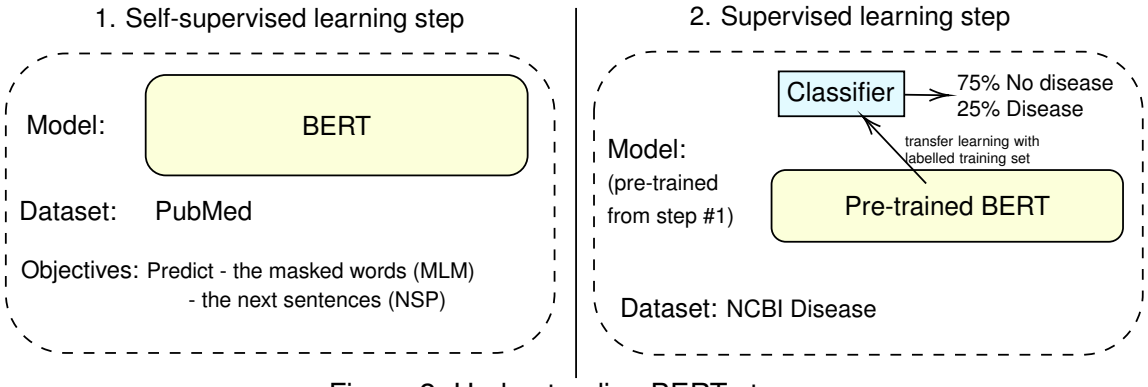

Figure 2: Understanding BERT steps

This diagram depicts two training steps for a BERT model. (1) Pre-training: Self-supervised learning on vast volumes of text (e.g. articles in PubMed). (2) Fine-tuning: Supervised training with a labeled dataset on a specified task.

2.2.1 Attention Mechanisms and Model Architecture

Attention mechanisms have become a cornerstone of modern deep learning architectures, particularly in natural language processing and, more recently, in computational biology. The concept of attention in machine learning is inspired by the cognitive process of selective focus in humans. Just as humans do not actively utilize all the information accessible from their surroundings, but rather concentrate on important subsets of data, attention mechanisms in neural networks allow the model to focus on the most relevant parts of the input when performing a task [BCB16, VSP+17].

In the context of BERT's architecture, the attention mechanism plays a crucial role in both of its pre-training objectives: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). This mechanism enables the model to dynamically focus on different parts of the input depending on the current context or task, which is particularly useful in sequence-to-sequence tasks where the relevance of input elements can vary depending on which part of the output is being generated.

The attention parameters, comprising the query, key, and value matrices, are initially randomly initialized. As the model processes vast amounts of text data across multiple epochs, these parameters are continuously updated through backpropagation. This iterative process allows the attention weights to evolve, gradually capturing increasingly meaningful relationships in the data. What begins as random noise in the attention patterns slowly transforms into specialized focus on linguistically relevant connections, such as syntactic dependencies or semantic relationships between words.

The attention mechanism in BERT works in close coordination with other key components of the model, specifically the feed-forward neural networks and layer normalization. This coordinated process enables BERT to dynamically capture and weigh relevant contextual information for each input token. The result is a set of nuanced, context-dependent representations that can be effectively adapted to a diverse array of downstream natural language processing tasks.

Attention mechanisms contribute significantly to several key benefits in transformer-based models like BERT. They improve the handling of long-range dependencies by allowing direct connections between any two positions in a sequence, enabling the model to capture relationships

between distant elements more effectively than traditional sequential models. Attention-based architectures also facilitate parallelization of computations, as they can process all input tokens simultaneously, unlike recurrent neural networks. This parallelization enables more efficient training and inference on parallel hardware. Additionally, attention mechanisms offer a degree of interpretability, as their weights can be visualized to provide insights into which parts of the input the model is focusing on for a given output, though this interpretability has limitations and requires careful analysis. The flexibility of the self-attention operation allows it to be applied to various input types and modalities, as it does not inherently assume a specific input structure. It is important to note, however, that while attention mechanisms enable or facilitate these benefits, they work in conjunction with other architectural elements and training procedures to realize the full potential of models like BERT.

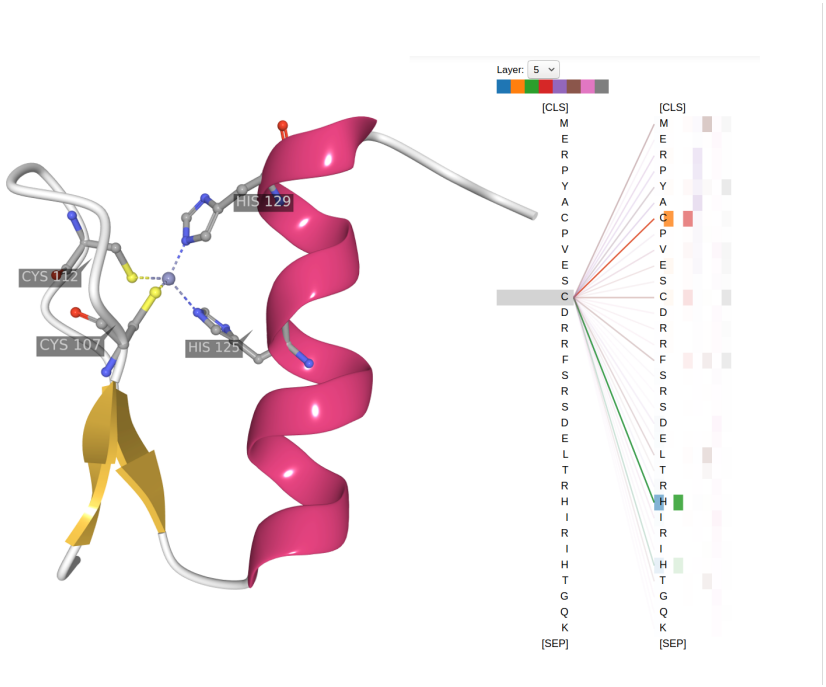

Figure 3: Self-attention visualization

This figure illustrates two key aspects of protein analysis: structural representation and attention mechanisms in protein language models. Left panel: The structure of the first 33 residues of a zinc-finger binding domain (PDB: 1A1L) is shown. Four residues crucial for coordinating zinc-binding and stabilizing the fold are highlighted: C107, C112, H125, and H129. Right panel: This displays a subset of the attention scores from one of the ProtTrans language models [EHD+21]. ”CLS” (Classification) token: A special token added at the beginning of the sequence, often used for classification tasks. Color key: The color bar at the top represents the scale of attention scores, ranging from low (blue) to high (red). Parallel tracks: Each horizontal track represents the attention patterns for a specific amino acid position or special token. The vertical alignment allows for comparison of attention patterns across different positions in the sequence. The figure is from [EHD+21].

The success of attention mechanisms in various domains led to the development of the transformer architecture, introduced by Vaswani et al. in 2017 [VSP+17]. However, it is important to note that transformers are more complex than simply attention mechanisms without recurrence or convolution. A transformer is a sophisticated neural network architecture comprising several crucial elements working in concert. At its core are multi-head self-attention mechanisms, which allow the model to attend to different aspects of the input simultaneously. These are complemented by position-wise feed-forward neural networks that process the output of the attention layers. The architecture also incorporates layer normalization to stabilize the learning process, and residual connections to facilitate gradient flow through the network. Additionally, transformers

use positional encodings to provide information about the sequence order, as the attention mechanism itself is position-agnostic. Unlike recurrent neural networks (RNNs) or convolutional neural networks (CNNs), transformers process entire sequences in parallel, relying on attention to capture dependencies between different positions in the input. This design enables efficient training on large datasets and has proven highly effective across a wide range of natural language processing tasks. This architecture has become the foundation for many state-of-the-art models in natural language processing, including BERT, and is increasingly being applied to biological sequence analysis.

Self-attention. Self-attention, a key component of the Transformer model [VSP+17], relates distinct positions of a single sequence to compute a contextual representation for each term in that sequence. As shown in Figure 3, this method allows each position in the sequence to attend to all positions, enabling the model to capture complex dependencies. During pre-training, the self-attention mechanism learns to focus on relevant parts of the input for both the MLM and NSP tasks.

The attention operation can be seen as a retrieval process that can be defined using the concepts of *queries*, *keys*, and *values*, and is motivated by the fact that it modifies word representations depending on "matching" query terms with related terms in the sequence [Agg22]. Each query vector is turned into an attention-modified vector that is a weighted average of the values (value vectors) in the sequence. Each value vector is connected with a key vector against which queries are "matched" using dot product attention [Agg22].

For an input sequence of arbitrary length N, the self-attention mechanism performs the following operations for each position i (where i ranges from 1 to N):

1. Compute the query vector q_i, the key vector k_i, both of dimension d_k, and the value vector v_i, of dimension d_v. These vectors are derived by multiplying an initial embedding $x_i \in R^{d_{model}}$ of the term i by three weight matrices $W^Q \in d_{model} \times d_k$, $W^K \in d_{model} \times d_k$, and $W^V \in d_{model} \times d_v$ learned through training the neural network (see Section 2.2.3.2):

$$\begin{aligned} q_i &= x_i W^Q \\ k_i &= x_i W^K \\ v_i &= x_i W^V \end{aligned} \tag{1}$$

   Note that the weight matrices W^Q, W^K, and W^V are learned through backpropagation during network training. Initially, they are randomly initialized. During each forward pass, these matrices are used to compute the query, key, and value vectors. After comparing the network's output to the ground truth and computing a loss function, backpropagation calculates the gradients of the loss with respect to each element of W^Q, W^K, and W^V using the chain rule. During each training iteration, the attention mechanism computes scores and outputs using the current values of W^Q, W^K, and W^V. These outputs contribute to the model's predictions, which are then compared to the true labels to compute a task-specific loss function (e.g., cross-entropy for classification tasks). An optimization algorithm, such as Stochastic Gradient Descent (SGD) or Adam [KB17], then updates W^Q, W^K, and W^V based on the gradients of this loss. This process is repeated over many batches of training data, gradually adjusting these matrices to improve the model's performance on the given task. The specific loss function depends on the downstream task for which the model is being trained.

2. Score the term i against all other terms in the sequence by multiplying its query vector q_i with all key vectors k_j:

$$s_{ij} = q_i \cdot k_j, \qquad \forall j = 1, ..., N. \tag{2}$$

3. Divide the scores of term i by the square root of the dimension of the key vector, d_k:

$$s'_{ij} = \frac{s_{ij}}{\sqrt{d_k}}, \qquad \forall j = 1, ..., N. \tag{3}$$

4. Apply a softmax function to the new term i scores to normalize them:

$$s''_{ij} = \frac{e^{s'}_{ij}}{\sum_{j=1}^{N} e^{s'}_{ij}}, \qquad \forall j = 1, ..., N. \tag{4}$$

5. Multiply each vector of values v_j by its corresponding normalized score:

$$v'_{ij} = s''_{ij} v_j, \qquad \forall j = 1, ..., N. \tag{5}$$

6. As a final output of the self-attention calculation, total the weighted value vectors:

$$z_i = \sum_{j=1}^{N} v'_{ij} \tag{6}$$

The third step of the computation seeks to solve a problem hypothesized by Vaswani et al. [VSP+17], who state that for large values of d_k, the dot products expand in magnitude, forcing the *softmax* function into regions with extremely small gradients. The division operation performed by d_k nullifies this effect [Lou20]. The vector representation of the self-attention mechanism is depicted in Figure 4.

In practice, the self-attention function is computed simultaneously on a matrix $Q \in \mathbb{R}^{N \times d_k}$ containing a set of *queries*. The keys and values are also packed into respective matrices $K \in \mathbb{R}^{N \times d_k}$ and $V \in \mathbb{R}^{N \times d_v}$. Thus, the output matrix is computed in the following manner:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{7}$$

Multi-head Attention. Instead of performing a single self-attention operation, Transformer-based models employ multi-head attention [VSP+17]. Queries, keys, and values are linearly projected h times using different learned projections. This allows for the concurrent consideration of information from distinct representation subspaces. The self-attention function is executed in parallel on these different projections, resulting in h distinct output matrices known as attention heads (Figure 5). These heads are then concatenated and projected to yield the final output:

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(Z_1, ..., Z_h)W^O \\ where\ Z_i &= Attention(QP_i^Q, KP_i^K, VP_i^V), \quad i = 1, ..., h. \end{aligned} \tag{8}$$

Note that $d_k = d_v = d_{model}/h$ is typically the case in multi-head attention [Lou20].

BERT's architecture is a bidirectional multilayer Transformer encoder [VSP+17]. In other words, BERT is comprised of L identical layers of Transformer encoders. Each encoder layer consists of two sublayer kinds. The first is a multi-head self-attention mechanism, which helps look at other words in the sequence while encoding a specific word. The second network is a position-wise, fully connected feed-forward network that is applied independently and identically to each position and consists of two linear transformations $(W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}, b_1 \in \mathbb{R}^{d_{ff}})$,

$(W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}, b_2 \in \mathbb{R}^{d_{model}})$ such that:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{9}$$

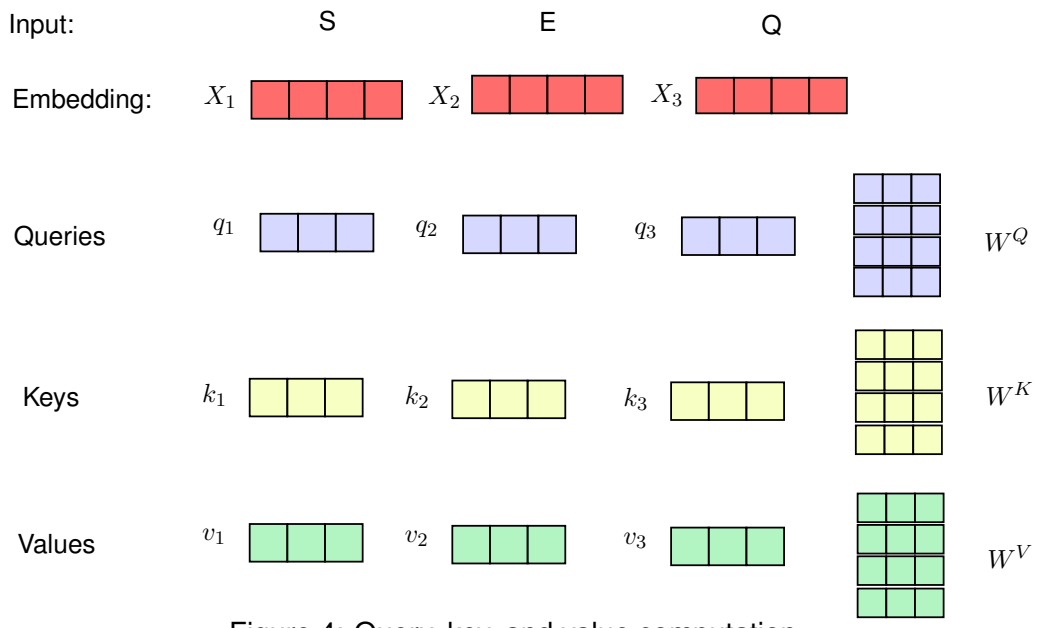

Figure 4: Query, key, and value computation

This illustration shows the diagram of the query, key, and value vectors q_i, k_i, and v_i, respectively. W^Q, W^K and W^V are weight matrices which are learned during the training phase. S, E and Q are examples of an input sequecen.

Input and output dimensions are d_{model}, while the inner layer's dimensions are $d_{ff} = 4d_{model}$. Moreover, an encoder layer applies a residual connection (by skipping some layers, residual link provides an alternate way for data to reach later layers of the neural network) [HZRS16] around each of the two sublayers, followed by layer normalization (LayerNorm) [BKH16]. Normalizing the distributions of intermediate layers allows for smoother gradients, faster training, and improved generalization accuracy [XSZ+19]. The output of each sublayer is:

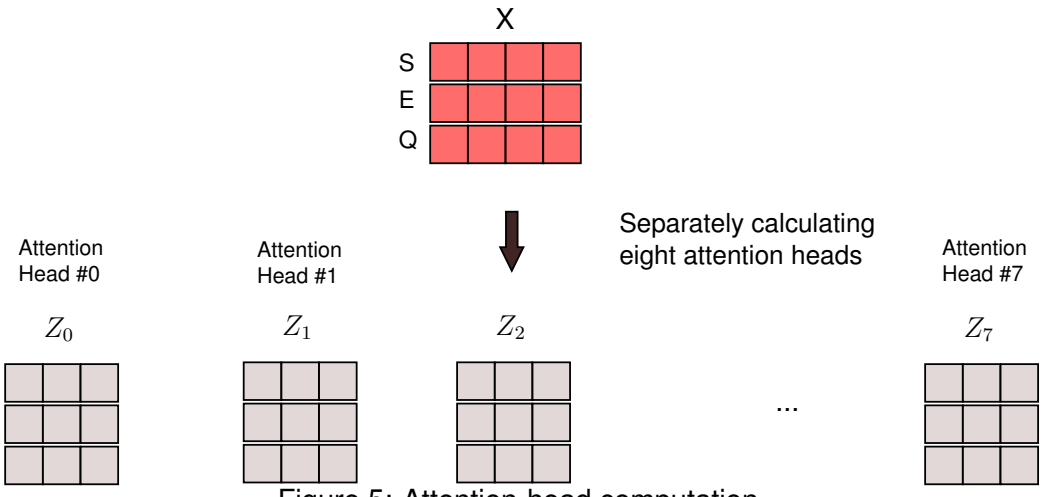

Figure 5: Attention-head computation

This graphic depicts the diagram of the attention heads $Z_i (i = 1, ..., h)$. X refers to the embedding vectors of words.

$$LayerNorm(x + Sublayer(x)) \tag{10}$$

LayerNorm stabilizes the activations, aiding in faster and easier training, where $Sublayer(x)$ is the function implemented by the sublayer itself. In order to ease these residual connections, all model sublayers generate outputs of the same dimension d_{model}. Figure 6 depicts the design of a single encoder. Note that although the linear transformations are identical across all points within the same sublayer, BERT employs distinct settings for each layer and is available in two variants:

- BERT-base: $L = 12$, $d_{model} = 768$, $h = 12$, $d_{ff} = 3072$ (110M total parameters).
- BERT-large: $L = 24$, $d_{model} = 1024$, $h = 16$, $d_{ff} = 4096$ (340M total parameters).

here, L represents the number of layers, d_{model} represents the dimensionality of each layer's input and output, h represents the number of attention heads in a self-attention sublayer, and d_{ff} represents the number of hidden units in a feed-forward sublayer.

2.2.2 Input Representations in BERT

Given a sequence of words as inputs (often limited to 512 tokens), BERT conducts a first transformation to create numerical input representations for the model. In reality, these input representations are built by summing *token*, *segment*, and *positional* embeddings.

Token Embedding. Given a word in the input sequence, BERT tokenizes it using WordPiece [WSC+16] embedding. WordPiece is essentially a model that generates a fixed-size vocabulary of individual characters, subwords, and words that optimally fits a given language corpus. Under this model, before tokenizing a word, the tokenizer checks whether the entire term is present in the vocabulary. If not, it attempts to break the word into the largest feasible subwords from the lexicon, and if that fails, it breaks the word down into individual characters.

The vocabulary used by BERT in NLP domain consists of the 30,000 most common English words and subwords, as well as all English characters and three special tokens:

- $[CLS]$ is a special classification token used at the beginning of each sequence. For classification tasks, the final hidden state corresponding to this token is used as the aggregate sequence representation. It is disregarded in jobs that do not involve classification.
- The $[SEP]$ is used as a separator, when dealing with sequence (sentence) pairs compressed into a single sequence. It also always concludes the sequence.
- $[MASK]$, which is utilized for the masked language modeling (MLM) training purpose, as explained in Section 2.2.3.1.

Segment Embedding. When working with sentence pairs, a learnt *segment* embedding is appended to each token to indicate whether it belongs to sentence A or sentence B. Segment embeddings are identical to token embeddings with a simple vocabulary of size 2.

Positional Embedding. BERT, like the original Transformer, uses *positional* embeddings to inject information about the relative or absolute position of the tokens in the input sequence. These embeddings have the same dimension d_{model} as the token and segment embeddings, allowing them to be easily combined. They are calculated with sine and cosine functions of varying frequencies:

$$PE_{(pos,2i)} = \sin(\frac{pos}{10000^{2i/d_{model}}}) \tag{11}$$

$$PE_{(pos,2i+1)} = \cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{12}$$

where pos denotes position and i denotes dimension. As a result, each positional embedding dimension corresponds to a sinusoid, and the wavelengths form a geometric progression from 2π to $10000 \cdot 2\pi$. Vaswani et al. [VSP+17] theorized that this function allows the model to readily learn relative positions since PE_{pos+k} may be expressed as a linear function of PE_{pos} for any fixed offset k [Lou20].

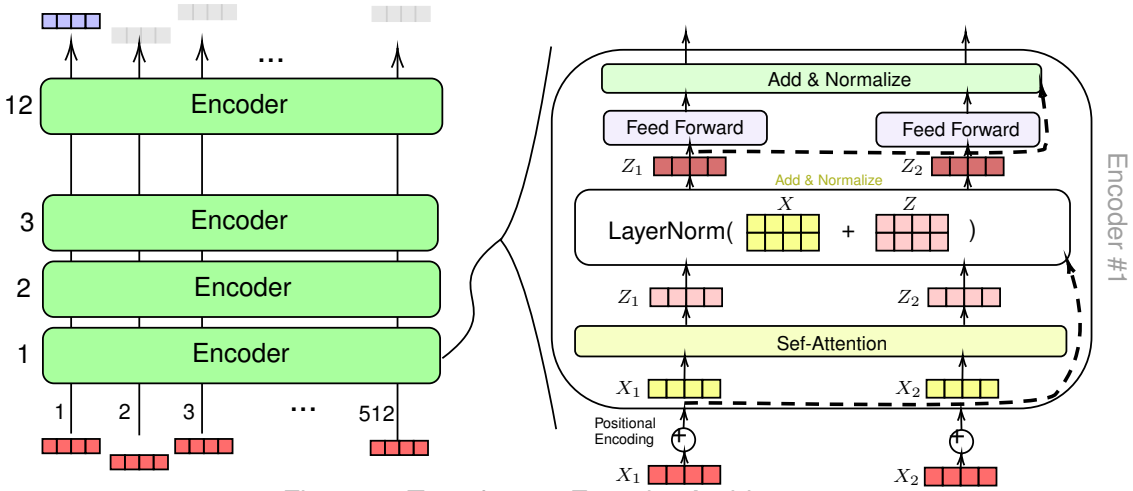

Figure 6: Transformer Encoder Architecture

This illustration depicts the interior of a Transformer encoder. [DCLT19].

2.2.3 BERT Pre-training and Fine-tuning

2.2.3.1 Pre-training Procedure

As mentioned before, BERT's pre-training procedure consists of two simultaneous tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). These tasks are designed to enable the model to learn both contextual word representations and relationships between sentences. The total pre-training loss is the sum of the mean MLM loss and the mean NSP loss.

Masked Language Modeling (MLM) MLM is a novel approach that addresses the limitations of traditional left-to-right or right-to-left language modeling when applied to bidirectional models. The key idea is to predict randomly masked tokens in the input, allowing the model to build a deep bidirectional representation.

The MLM process works as follows:

1. Random Masking: 15% of all WordPiece tokens in each training sequence are randomly selected for masking.

2. Token Replacement: For each selected token:

   - 80% of the time, replace it with the [MASK] token
   - 10% of the time, replace it with a random token
   - 10% of the time, leave it unchanged

   This strategy prevents the model from relying too heavily on the [MASK] token and helps it maintain a distribution closer to real data during fine-tuning.

3. Prediction: The model then attempts to predict the original value of the masked tokens based on the context provided by the other, non-masked, tokens in the sequence.

Formally, given an input sequence of N tokens $x = [x_1, x_2, ..., x_N]$, and a set of k masked positions $m = [m_1, ..., m_k]$, the MLM loss is calculated as:

$$\mathcal{L}MLM = \frac{1}{k} \sum i \in m - \log p(x_i | x^{masked}) \tag{13}$$

where x^{masked} is the input sequence with masks applied, and $p(x_i|x^{masked})$ is the predicted probability of the correct token at masked position i.

Next Sentence Prediction (NSP) NSP (Figure 7) is a binary classification task designed to improve the model's understanding of relationships between sentences. This is crucial for downstream tasks such as question answering and natural language inference.

The NSP process works as follows:

1. Sentence Pair Selection: For each training example, two sentences (A and B) are chosen:
   - 50% of the time, B is the actual next sentence that follows A in the corpus (labeled as IsNext)
   - 50% of the time, B is a random sentence from the corpus (labeled as NotNext)
2. Input Formatting: The sentences are tokenized and combined into a single sequence with special tokens: [CLS] Sentence A [SEP] Sentence B [SEP]
3. Prediction: The model predicts whether the second sentence follows the first in the original text, using the final hidden state of the [CLS] token.

The NSP loss is calculated using binary cross-entropy:

$$\mathcal{L}NSP = -cIsNext \log(p_{IsNext}) - (1 - c_{IsNext}) \log(1 - p_{IsNext}) \tag{14}$$

where c_{IsNext} is the true label (1 for IsNext, 0 for NotNext) and p_{IsNext} is the model's predicted probability for the IsNext class. By combining MLM and NSP, BERT learns both local (word-level) and global (sentence-level) context, resulting in a powerful pre-trained model that can be fine-tuned for a wide range of downstream tasks.

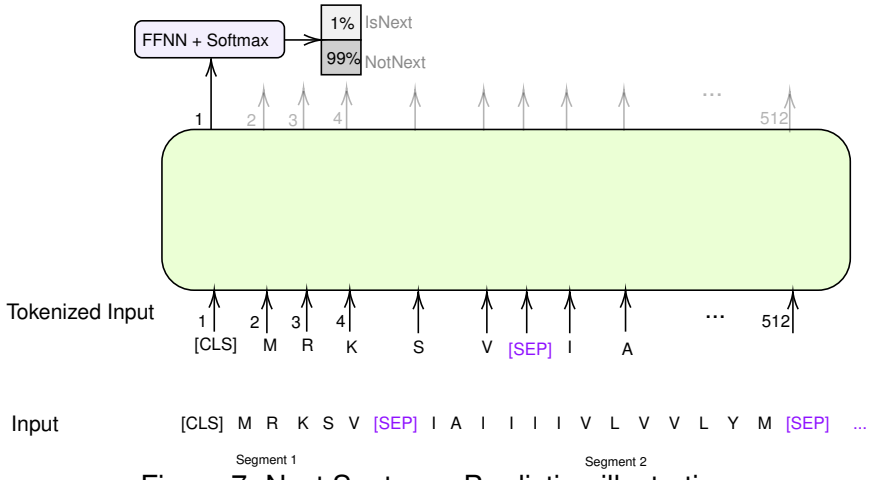

Figure 7: Next Sentence Prediction illustration

This diagram depicts the Next Sentence Prediction training objective, which predicts if two sentences follow one another. The term FFNN stands for feed-forward neural network.

2.2.3.2 Training

The training phase of a neural network is a critical process where the model learns to minimize a loss function through iterative updates of its parameters. This process involves several key components:

Loss Function: The loss function quantifies the difference between the model's predictions and the expected outputs. In BERT's case, the total loss is a combination of the Masked Language Modeling (MLM) loss (Equation 13) and the Next Sentence Prediction (NSP) loss (Equation 14).

Optimizer: An optimizer is an algorithm that adjusts the model's learnable parameters to minimize the loss function. Common optimizers include Stochastic Gradient Descent (SGD) [RM51], Adam [KB17], and AdamW [LH19] (a variant of Adam with decoupled weight decay).

Gradient Descent: The core principle of neural network training is gradient descent. This iterative process updates the model's parameters (θ) in the direction that minimizes the loss:

$$\theta_{new} = \theta_{old} - \eta \nabla \mathcal{L}(\theta) \tag{15}$$

where:

- θ represents the model's parameters (weights and biases)
- η is the learning rate, a hyperparameter that controls the step size of each update
- $\nabla \mathcal{L}(\theta)$ is the gradient of the loss function with respect to the parameters

Backpropagation: This algorithm efficiently computes the gradients of the loss with respect to each parameter in the network. It works by applying the chain rule of calculus, propagating the error backward through the network layers.

Automatic Differentiation: Modern deep learning frameworks like PyTorch [PGM+19] and TensorFlow implement automatic differentiation engines. These tools dynamically construct a computational graph of operations and automatically compute gradients, eliminating the need for manual gradient calculations. This is particularly crucial for complex models like BERT with millions of parameters.

Training Loop: The typical training process involves:

1. Forward pass: Compute the model's predictions
2. Loss calculation: Evaluate the loss function
3. Backward pass: Compute gradients using backpropagation
4. Parameter update: Apply the optimizer to update the model's parameters

Batch Processing: Training is usually done in batches to improve computational efficiency and provide a balance between update frequency and estimate accuracy of the gradient.

Regularization: Techniques like weight decay, dropout, and early stopping are often employed to prevent overfitting and improve generalization.

Learning Rate Scheduling: Many training regimes employ learning rate schedules (e.g., linear warmup followed by linear decay) to improve convergence and final model performance.

For BERT [DCLT19] specifically, the authors used the Adam optimizer with a learning rate of 1e-4, $\beta 1$ = 0.9, $\beta 2$ = 0.999, L2 weight decay of 0.01, learning rate warmup over the first 10,000 steps, and linear decay of the learning rate. The model was trained on 4 Cloud TPUs in Pod configuration (16 TPU chips total) for 1,000,000 steps with a batch size of 256 sequences (256 sequences * 512 tokens = 128,000 tokens/batch).

2.2.3.3 Downstream Tasks

Fine-tuning and frozen strategies exist for applying pre-trained language representations to subsequent tasks. On the one hand, the fine-tuning method adds minimal task-specific parameters and trains on subsequent tasks by simply fine-tuning all pre-trained parameters. The frozen approach, on the other hand, employs task-specific architectures that integrate the pre-trained representations as input features for learning the task.

Fine-tuning Approach. The two pre-training goals of BERT allow it to be applied to any single sequence and sequence-pair tasks without requiring significant task-specific architecture alterations. One need merely plug the task-specific inputs and outputs into BERT and fine-tune all parameters end-to-end for a few epochs for each task.

Frozen Approach. In addition to the fine-tuning approach in which a simple output layer is added to the pre-trained model and all parameters are jointly fine-tuned on a downstream task, BERT can also be used with a feature-based approach in which word representations are extracted from the pre-trained model and serve as inputs to other task-specific architecture.

2.2.3.4 ProtBert-BFD

ProtBert-BFD comes from ProtTrans project [EHD+21] and is based on BERT model (Section 2.2) that was pre-trained on BFD (`https://bfd.mmseqs.com`) [JEP+21], a dataset containing 2.1 billion protein sequences, using solely the Masked Language Modeling training objective (MLM) (See Section 2.2.3.1). It was presented in the GitHub repository (`https://github.com/agemagician/ProtTrans`).

One important difference between ProtBERT-BFD model and the original BERT version is the way of dealing with sequences as separate documents. This means the Next Sentence Prediction (NSP) (See Section 2.2.3.1) is not used, as each sequence is treated as a complete document. The masking follows the original BERT (Section 2.2.3.1) training with a random mask of 15% of the amino acids in the input.

Compared to the original BERT design (See Section 2.2), the pre-training ProtBERT-BFD settings are as follows: L = 30, d_{model} = 1024, h = 16, d_{ff} = 4096 (420M total parameters), where L represents the number of layers, d_{model} represents the dimensionality of each layer's input and output, h represents the number of attention heads in a self-attention sublayer, and d_{ff} represents the number of hidden units in a feed-forward sublayer. The protein sequences are uppercased and tokenized using a single space and a vocabulary size of 21 (With an amino acid 'X' to substitute unknown amino acids).

2.3 Advanced LMs and PLMs

2.3.1 T5 Language Model

The Text-to-Text Transfer Transformer (T5) language model [RSR+20], introduced by researchers at Google in 2020, extends the capabilities of previous models like BERT by adopting a unified text-to-text framework. Unlike BERT, which is primarily designed for understanding tasks, T5 converts all NLP tasks into a single text-based format where both the input and output are text strings. This approach allows T5 to be applied universally to any task that can be reformulated to fit this text-in, text-out pattern.

T5 is pre-trained on a multi-task mixture of unsupervised and supervised tasks, drawn from a diverse set of data sources, using a conditional language modeling objective similar to the traditional language modeling but involving a prefix that indicates which task is being solved [RSR+20]. This comprehensive pre-training enables T5 to generalize well across different domains and tasks.

While BERT revolutionized the field of NLP with its bidirectional training of transformers to better understand word contexts, T5 builds on and diverges from BERT in several key ways:

1. **Unified Framework:** T5 treats every problem as a text generation task, simplifying the usage of transformers across varied NLP tasks [RSR+20]. BERT, by contrast, uses task-specific heads for different downstream applications, which can complicate the integration of the model into diverse workflows.

2. **Training Objective:** BERT employs a masked language model (MLM) and next sentence prediction for training. T5, however, uses a span corruption training objective where random contiguous spans of text are replaced with a sentinel token, and the model is trained to predict these masked spans [RSR+20].

3. **Flexibility in Task Formulation:** T5's text-to-text approach inherently allows it to handle translation, summarization, text classification, and question answering all within the same model architecture by merely changing the task-specific prefix [RSR+20]. BERT, in comparison, requires different model configurations and output layers for such varying tasks.

4. **Decoder Component:** T5 includes a decoder in its architecture, which is essential for generating text. BERT, being primarily an encoder, lacks a decoder, which limits its direct use in generation tasks without modifications or additional model components.

The adaptability and comprehensive training approach of T5 have led to its incorporation into specialized models for bioinformatics, particularly in PLMs [EHD+21, EESE+23]. Models like ProtT5 [EHD+21] and Ankh [EESE+23] extend T5's architecture to understand and predict protein functions and structures, leveraging the power of T5's encoder-decoder setup to handle complex sequences typical in protein modeling [EHD+21]. This demonstrates the broad utility of T5's design, making it a valuable tool in both general language understanding and specialized fields such as proteomics.

2.3.2 ESM Project

The ESM Project comprises two main models, ESM-1b and ESM-2, which are both derived from the BERT architecture and trained for protein representation tasks.

ESM-1b [RMS+21] is a BERT-inspired model that generates 1280-dimensional vector representations. It consists of 33 layers and 650 million parameters, which is more extensive than the original BERT-Base (12 layers) and BERT-Large (24 layers) models. ESM-1b was trained on 250 million protein sequences from the UniParc database [LDB+04] and assessed on various tasks such as secondary structure prediction, remote homology, long-range interaction, and mutational effect. The model's performance was either on par with or superior to that of bidirectional LSTM or Transformer models, except for remote homology prediction.

ESM-2 [LAR+23], on the other hand, is a new family of transformer protein language models (PLMs) with sizes ranging from 8 million to 15 billion parameters. These models introduce architectural improvements, refined training parameters, and increased computational resources and data compared to ESM-1b [RMS+21]. Notably, ESM-2 outperforms its predecessor, ESM-1b, at a similar parameter count and surpasses other recent PLMs in structure prediction benchmarks [LAR+23].

ESMFold [LAR+23], a fully end-to-end single-sequence structure predictor, is developed by training a folding head classifier for ESM-2 [LAR+23]. It offers state-of-the-art structure prediction accuracy, matching the performance of AlphaFold2 [JEP+21]. As the language models scale, they learn information that enables the prediction of protein three-dimensional structures at the resolution of individual atoms. This approach results in predictions that are up to 60 times faster than the state-of-the-art while maintaining resolution and accuracy [LAR+23].

The ESM Metagenomic Atlas [LAR+23], an open science resource, makes all predicted structures available at (`https://esmatlas.com`). Structures can be accessed via bulk download, a programmatic API, or through a web resource that enables search by sequence and

structure. These tools facilitate both large-scale and focused analysis of the hundreds of millions of predicted structures.

2.3.3 ESM-1b

ESM-1b, highlighted in the work of Rives et al. [RMS+21], stands out for its profound capability to capture complex protein sequence patterns, attributed to its training on a vast corpus of 86 billion amino acids across 250 million sequences. Utilizing a deep Transformer architecture with 650 million parameters, ESM-1b sets a benchmark in protein language modeling through unsupervised learning, particularly with its masked language modeling objective, offering insights into the functional nuances of protein sequences.

Architectural Innovations Distinguished by its architectural nuances, ESM-1b employs pre-activation residual blocks and harmonic positional embeddings [HZRS16], diverging from traditional models, including its successor, ESM-2 [LAR+23]. This design choice, avoiding dropout [SHK+14], maximizes the model's potential to elucidate protein sequence intricacies. The harmonic embeddings [VSP+17], in particular, may better resonate with the periodic nature of proteins, potentially contributing to ESM-1b's superior performance in our experiments.

Training Regimen The model's training regimen is meticulously crafted, leveraging the extensive UniParc [The21] and UniRef [SWH+15] datasets to ensure a diverse and comprehensive protein sequence representation. The strategic hyperparameter tuning and the focus on a masked language modeling task, devoid of auxiliary losses, underscore a dedicated approach to optimizing predictive accuracy, setting ESM-1b apart from other PLMs, including ESM-2.

Comparative Analysis with ESM-2 Upon comparing ESM-1b with ESM-2, it is evident that the former's unique architectural features and training strategies may confer advantages in specific contexts. The use of harmonic positional embeddings and the strategic absence of dropout in ESM-1b contrast with ESM-2's methodologies, potentially offering a more stable and efficient training regime and superior handling of protein sequence complexities.

While this comparative insight provides a foundation for understanding ESM-1b's efficacious performance, it is crucial to acknowledge the limitations inherent in the absence of a comprehensive ablation study. Future research should aim to quantitatively evaluate the impact of individual model components, offering a clearer delineation of the features contributing to the observed performance differences.

2.4 Other PLMs

2.4.1 TAPE

TAPE (Tasks Assessing Protein Embeddings) [RBT+19] is a benchmark suite designed to evaluate protein language models across diverse biological tasks. It consists of five downstream tasks: secondary structure prediction, contact prediction, remote homology detection, fluorescence prediction, and stability prediction. The datasets range in size from 8,000 to 50,000 training examples, with specific test sets for each task.

The authors evaluated three model architectures (LSTM, Transformer, and ResNet) as well as two previously proposed protein-specific models. All models were pretrained on the Pfam database, containing approximately 32 million protein domains. The largest model, a 12-layer Transformer with 38 million parameters, was trained for one week on 4 NVIDIA V100 GPUs.

TAPE demonstrated that self-supervised pretraining generally improves performance across tasks. For example, on the secondary structure prediction task (CB513 dataset), the pretrained Transformer achieved 73% Q3 accuracy compared to 70% without pretraining. On the challenging remote homology detection task at the fold level, pretraining improved top-1 accuracy from 9% to 21% for the Transformer model.

However, the results also showed that no single architecture consistently outperformed others across all tasks. Additionally, traditional alignment-based features still outperformed learned embeddings on some tasks. For instance, on contact prediction, alignment features achieved 64% precision for top L/5 long-range contacts, while the best pretrained model reached only 39%.

2.4.2 MSA-Transformer

The MSA Transformer [RLV+21] is an unsupervised protein language model that operates on multiple sequence alignments (MSAs) rather than individual sequences. It contains 100 million parameters and was trained on a dataset of 26 million MSAs, with an average depth of 1,192 sequences per MSA. The model uses axial attention to efficiently process the 2D structure of MSAs, interleaving row and column attention operations.

The MSA Transformer significantly outperforms previous state-of-the-art models in unsupervised contact prediction tasks. On a test set of 14,842 proteins, it achieves a top-L long-range contact precision of 57.4%, compared to 41.1% for ESM-1b (a 650M parameter single-sequence model) and 39.3% for Potts models. Notably, the MSA Transformer maintains high performance even for proteins with shallow MSAs, where traditional coevolutionary methods struggle.

In supervised contact prediction, features from the MSA Transformer achieve 54.6% top-L long-range precision on CASP13-FM targets, surpassing the previous state-of-the-art trRosetta model (51.8%). The model demonstrates strong performance even when using small input sets, achieving better results than ESM-1b using just 16 sequences selected for diversity.

2.4.3 Ankh

Ankh is an optimized PLM that achieves state-of-the-art performance on various protein prediction tasks while using significantly fewer parameters than previous models. The model is available in two versions: Ankh base with 726 million parameters and Ankh large with 3 billion parameters. Notably, Ankh large outperforms ESM-2, which has 15 billion parameters, on most benchmarks despite having only 20% of its parameter count.

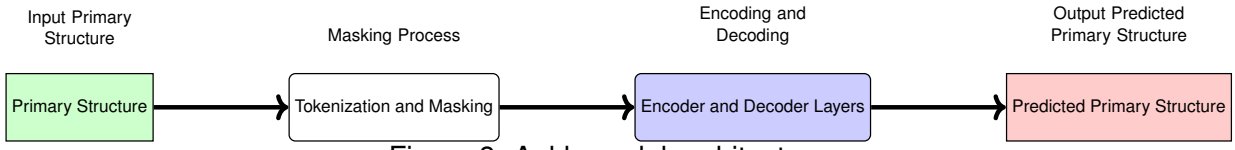

Figure 8: Ankh model architecture

The plot of the Ankh model architecture showing primary structure input, tokenization and masking, encoding-decoding process, and predicted primary structure output.

Ankh's architecture consists of 48 encoder layers and 24 decoder layers, with an embedding dimension of 1536 for the large model and 768 for the base model. It utilizes a Gated-GELU activation function and a relative positional embedding dimension of 32 with an offset of 128. The model was pre-trained on the UniRef50 dataset [SWH+15] containing 45 million protein sequences using masked language modeling, where 20% of input tokens were masked and predicted.

Training was optimized using the Adafactor optimizer and a linear scheduler, with a learning rate of 0.004 and a batch size of 24. The authors conducted extensive experiments to optimize the model architecture and training procedure, including tests of different masking strategies, model depths, and positional encodings.

Ankh demonstrates impressive performance across various downstream tasks. In secondary structure prediction using the CASP12 dataset, Ankh achieves 83.8% Q3 accuracy compared to 83.2% for ESM-2. For contact prediction using the ProteinNet dataset, Ankh reaches 49.0% L/1 precision versus 33.3% for ESM-2. Ankh also excels at protein fold classification, with 61.1% accuracy compared to 56.7% for ESM-2.

Importantly, Ankh achieves these results with significantly lower computational requirements. Feature extraction for a 1024 residue protein takes 7.1x less time with Ankh compared to ESM-2. The model was trained on 8 Google TPU v4 chips, while ESM-2 required much larger computational resources.

Ankh excels in various protein modeling tasks, including structure prediction and functional benchmarking. It is effective in generating protein variants using insights into evolutionary conservation and mutation trends, supporting complex biological research. The model's efficiency makes advanced protein modeling more accessible to researchers with limited computational resources.

The strong performance across diverse tasks suggests that Ankh learns generalizable representations of protein sequence and structure. This demonstrates that carefully optimized smaller models can outperform much larger protein language models, potentially democratizing access to advanced protein modeling tools in the research community.

2.5 ProstT5

ProstT5 [HWS+24] is a protein language model that incorporates tertiary structure information into its training process. It builds upon the ProtT5 model by fine-tuning it on a dataset of 17 million high-quality, non-redundant protein structures from AlphaFold2 [JEP+21] predictions. ProstT5 uses the 3Di alphabet introduced by Foldseek [vKKT+23] to encode 3D protein structures as 1D sequences, allowing it to translate between amino acid sequences and structural representations.

The model demonstrates improved performance on several downstream tasks compared to sequence-only models. For secondary structure prediction, ProstT5 achieves accuracy up to 90% on the NEW364 dataset. In remote homology detection using SCOPe40, ProstT5-predicted 3Di strings nearly match the performance of experimental structures while significantly outperforming traditional sequence alignment methods.

ProstT5 also shows capability in inverse folding, generating novel sequences with only 21% sequence identity to native proteins while maintaining high structural similarity (average lDDT of 72). The model consists of 3 billion parameters and was trained for 36 days on 8 NVIDIA A100 GPUs.

2.6 ML for Protein Sequence Analysis

In this study, we have employed a diverse array of classifiers to ensure a comprehensive evaluation of our approach in protein classification tasks. Each classifier was chosen for its unique characteristics and proven effectiveness in various aspects of machine learning, particularly in bioinformatics [GB23a, AB20a, AB20b]. The rationale behind the selection of each classifier, as well as their interpretability in the context of protein classification, is discussed below.

Support Vector Machine (SVM) SVM [CZC03] is a supervised learning algorithm that works by finding a decision boundary that maximally separates the training data into their corresponding classes. The decision boundary is determined by the support vectors, which are the training data points that lie on the margin of the classes. The SVM model then predicts the label of the input data based on the location of the input data relative to the decision boundary.

Given a set of input features $X = x_1, x_2, ..., x_n$ and their corresponding labels $Y = y_1, y_2, ..., y_n$, where $y_i \in 0, 1$, the SVM model learns a decision boundary by solving the following optimization problem:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^t w + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(w\phi(x_i + b)) + \xi_i - 1 \geq 0, \quad i = 1, ..., n \\ & \xi \geq 0, \quad i = 1, ..., n \end{aligned} \tag{16}$$

where w and b are the parameters of the decision boundary, ξ is the vector of slack variables, C is the penalty parameter, and $\phi(x_i)$ is the feature mapping of the input data. The decision boundary is given by the equation $w^T\phi(x) + b = 0$. The SVM model then predicts the label of the input data as:

$$\hat{y} = \text{sign}(w^T\phi(x) + b) \tag{17}$$

where sign$(z) = 1$ if $z \geq 0$ and sign$(z) = -1$ if $z < 0$.

The SVM model is effective, as it can learn complex decision boundaries that can accurately separate the training data into their corresponding classes. It is also efficient, as it only uses a small subset of the training data as support vectors to determine the decision boundary. However, the SVM model is sensitive to the choice of the kernel function, which is used to map the input data into a higher-dimensional space [BS03]. The kernel function allows the SVM model to learn non-linear decision boundaries by applying a non-linear transformation to the input data. This is known as the kernel trick, which allows the SVM model to operate as if it were working in a higher-dimensional space without explicitly computing the coordinates of the data in that space. The kernel function implicitly computes the dot products between the mapped data points, enabling the SVM to learn complex decision boundaries in the higher-dimensional space without actually transforming the input data. This approach significantly improves the SVM model's performance on non-linearly separable data while maintaining computational efficiency. By avoiding the need to explicitly calculate and store the high-dimensional representations of the input data, the kernel trick makes SVMs practical for solving complex classification problems [MKE+19].

SVMs [CV95] are known for their effectiveness in handling high-dimensional data and their ability to model complex nonlinear relationships. Their robustness in avoiding overfitting makes them suitable for protein classification tasks where the feature space can be extensive and complex. The interpretability of SVMs in this context lies in their capacity to find optimal hyperplanes that distinctly classify different protein types.

k-Nearest Neighbors (kNN) kNN [JCWJ07] is a non-parametric supervised learning algorithm that works by calculating the distance between the input data and a set of labeled training data. The distance is calculated using a distance metric, such as the Euclidean distance, which measures the difference between the feature values of the input data and the training data. The kNN model then predicts the label of the input data by finding the k training data points that are closest to the input data and averaging their labels.

Given an input data point x with features $X = x_1, x_2, ..., x_n$ and a set of training data $T = (t_1, y_1), (t_2, y_2), ..., (t_m, y_m)$, where t_i is the feature vector of the training data and y_i is its corresponding label, the kNN model predicts the label of the input data as follows:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{18}$$

where y_i is the label of the i-th nearest neighbor to the input data and k is the number of neighbors.

The kNN model is simple and easy to implement, as it does not require training to learn the relationship between the input features and the output labels. However, the model does require training to learn the distance metric that is used to determine the similarity between the input data points and the reference points. It is also effective, as it can learn complex non-linear relationships between the input data and the output labels. However, it is computationally expensive, as it requires calculating the distances between the input data and the entire training set for each prediction. In addition, the kNN model is sensitive to the choice of the distance metric and the number of neighbors, which can affect the performance of the model. The simplicity and intuitiveness of kNN [CH67] make it a valuable tool for initial exploratory analysis in protein classification. It classifies proteins based on similarity measures, providing insights into the

clustering of protein types. The interpretability of kNN is derived from its straightforward approach to classification based on proximity to known examples.

Random Forest (RF) RF [Qi12] is an ensemble model that consists of a collection of decision trees, where each tree is trained on a random subset of the input data and makes predictions based on the feature values of the input data. This allows the decision trees to make independent predictions and avoid overfitting the input data. Decision trees in a Random Forest are trained using a process called recursive binary splitting. For each tree, the algorithm first creates a bootstrap sample of the training data by randomly sampling with replacement. At each node of the tree, a random subset of features is selected. The algorithm then chooses the best split from this subset based on a criterion such as Gini impurity or information gain. This process continues recursively until a stopping condition is met, such as reaching a maximum depth or having a minimum number of samples per leaf. Once all trees are trained, the Random Forest model uses majority voting (for classification tasks) or averaging (for regression tasks) to combine the predictions of the individual decision trees. This ensemble approach reduces the variance of the predictions and improves the overall performance of the model. Given a set of input features $X = x_1, x_2, ..., x_n$ and their corresponding labels $Y = y_1, y_2, ..., y_n$, where $y_i \in 0, 1$, the RF model learns a set of decision trees to predict the labels of the input data as:

$$\hat{y}_i = \frac{1}{T} \sum_{j=1}^{T} \hat{y}_{i,j} \tag{19}$$

where $\hat{y}_i$ is the final prediction for sample i made by the random forest, T is the number of decision trees in the random forest, and $\hat{y}_{i,j}$ is the prediction of the j-th decision tree for sample i. RF [Ho98] is a powerful ensemble method known for its high accuracy and ability to handle imbalanced datasets, a common challenge in protein classification. The interpretability of RF in our study is enhanced by its feature importance measures, which provide insights into which attributes most significantly influence protein classification.

Feed-Forward Neural Network (FFNN) FFNN [Bis95, AVD22] consists of multiple layers of neurons, where each layer transforms the input data into a higher-dimensional space using a non-linear activation function. The final output of the FFNN is obtained by applying a linear activation function to the output of the last layer. Given a set of input features $X = x_1, x_2, ..., x_n$ and their corresponding labels $Y = y_1, y_2, ..., y_n$, where $y_i \in 0, 1$, the FFNN model learns a set of weights and biases to predict the labels of the input data as follows:

$$\hat{Y} = f_{\text{out}} \left(\sum_{i=1}^{n} W_i f_{\text{act}}(W_i X + b_i) + b_{\text{out}} \right) \tag{20}$$

where W_i and b_i are the weights and biases of the i-th layer, f_{act} is the non-linear activation function, and f_{out} is the linear activation function applied to the output of the last layer.

The FFNN model learns the weights and biases of each layer using a gradient-based optimization algorithm, such as *Adam* [KB17]. The optimization algorithm minimizes the loss function, which measures the difference between the predicted labels $\hat{Y}$ and the true labels Y. The loss function is used to compute the gradients of the weights and biases with respect to the loss, which are then used to update the weights and biases of the model.

The FFNN model is known for its ability to learn complex non-linear relationships between the input features and the labels. It is also flexible, as it can be easily adapted to different datasets by changing the number of layers and the number of neurons in each layer. However, the FFNN model is prone to overfitting and requires a large amount of training data to learn the weights and biases accurately [Bis95]. FFNNs [Sch15] offer flexibility and capability to model complex nonlinear relationships in high-dimensional data. In protein classification, FFNNs can capture

intricate patterns and interactions between features. However, the interpretability of FFNNs is more challenging due to their "black-box" nature, but techniques such as layer-wise relevance propagation [BBM+15] can be employed to gain insights.

Logistic Regression (LR) LR [Sto11] is a linear model that models the probability that an input sample belongs to a particular class, based on the linear combination of the input features and the model weights. Given a set of input features $X = x_1, x_2, ..., x_n$ and their corresponding labels $Y = y_1, y_2, ..., y_n$, where $y_i \in 0, 1$, the LR model learns the weights $W = w_1, w_2, ..., w_n$ that minimize the cost function, which is defined as the negative log-likelihood of the predicted labels of the input data:

$$J(W) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right] \tag{21}$$

where m is the number of samples and $\hat{y}_i$ is the predicted label of the i-th sample, which is computed as the sigmoid function of the linear combination of the input features and the model weights:

$$\hat{y}_i = \sigma(W^T \cdot x_i) = \frac{1}{1 + e^{-W^T \cdot x_i}} \tag{22}$$

The sigmoid function maps the output of the linear combination to the range $[0, 1]$, which represents the probability that the i-th sample belongs to the positive class. The weights of the LR model are learned using gradient descent, which iteratively updates the weights to minimize the cost function by computing the partial derivative of the cost function with respect to each weight.

A binary logistic regression task typically models the linear relationship between the log-odd of the positive class and the input variables (representation) [PHG+21]. The mathematical definition of the relationship is:

$$\ln(\frac{p}{1-p}) = \beta_0 + \sum_{i=1}^{N}\beta_i x_i \tag{23}$$

with p as the probability the positive class, x_i as the i^{th} element of the feature vector, β_i as the coefficient or parameter of x_i (β_0 as the intercept or bias), and N as the size of the feature vector. The objective of the machine learning algorithm of a logistic regression model is to discover parameter values that minimise the model's log-loss (a measure of how inaccurate its predictions are).

LR [TM16] provides a straightforward probabilistic approach to classification. In the context of protein classification, it offers an easily interpretable model in terms of the likelihood of a protein belonging to a particular class based on its features.

Convolutional Neural Network (CNN) A CNN [ON15, Agg18] is a type of neural network that that has been successfully applied to various tasks, including image classification and object detection [AMAZ17, Agg18] and it has also been used for protein analysis [SOPK18, NHTO22]. It consists of multiple layers of neurons, where each layer applies a convolution operation to the input data to extract local features. The convolution operation is applied using a set of filters, where each filter is a small matrix of weights that is learned by the CNN model. The final output of the CNN is obtained by applying a non-linear activation function to the output of the last layer.

The convolution operation is a mathematical operation that involves combining an input sequence with a set of filters using element-wise multiplication, sliding the filters across the input, and producing an output through element-wise summation. In the context of a CNN, this operation is used to extract features from the input sequence. The convolution operation can be defined mathematically in two ways:

1. Using the asterisk (*) as the convolution operator:

$$O = X * F \tag{24}$$

where O is the output of the convolution operation, X is the input sequence, and F is the set of filters. The output of the convolution operation is a feature map with dimensions $(l - f_l + 1) \times o$, where l is the length of the input sequence, f_l is the length of the filters, and o is the number of output channels.

2. Using a double sum:

$$O_i = f_{\text{act}} \left(\sum_{j=1}^{c} \sum_{k=1}^{f_l} F_{i,j,k} \cdot X_{i+k-1,j} + b_i \right) \tag{25}$$

where O_i is the output of the convolution operation at position i, f_{act} is the non-linear activation function, and b_i is the bias of the filter.

The CNN model, similar to FFNN, learns the filters using a gradient-based optimization algorithm. Then the loss function is used to compute the gradients of the filters with respect to the loss, which are then used to update the filters of the model.

CNNs [LBD+89] are particularly adept at capturing spatial and temporal patterns in data, making them well-suited for tasks involving sequence data such as protein sequences. Their interpretability in protein classification can be approached through techniques like visualization of filter activations to understand what features the CNN is focusing on.

Implementation and Integration of Classifiers: For the implementation of our classifiers, we utilized the scikit-learn library [Kra16] for SVM, kNN, RF, FFNN, and LR. In contrast, CNNs were implemented using PyTorch [PGM+19]. This choice of tools was guided by their widespread adoption in the field and the robustness they offer for machine learning tasks.

Utilizing these classifiers in combination with PLMs allows for a robust comparison across different machine learning approaches. This provides a comprehensive understanding of their applicability and effectiveness in the complex tasks of protein classification.

2.6.1 Evaluation Metrics

In our research, we have employed a comprehensive set of evaluation metrics to assess the performance of our models across various tasks. These metrics provide a multifaceted view of model performance, each capturing different aspects of predictive accuracy and reliability. The following metrics have been consistently used throughout our studies:

1. accuracy:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

accuracy measures the proportion of true results (both true positives and true negatives) among the total number of cases examined. It is particularly useful for balanced datasets.

2. Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision, also known as positive predictive value, calculates the fraction of relevant instances among the retrieved instances.

3. Recall (sensitivity):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall, or sensitivity, measures the ability of the model to identify all relevant instances.

4. specificity:

$$\text{specificity} = \frac{TN}{TN + FP}$$

specificity, also known as the true negative rate, quantifies the proportion of actual negative instances that are correctly identified. It complements sensitivity by providing insight into a model's ability to correctly identify negative instances, an equally important measure in many bioinformatics applications [GB23a].

5. F1 Score (F1):

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score is the harmonic mean of precision and recall, providing a single metric that balances both concerns.

6. Matthews Correlation Coefficient (MCC):

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

MCC is particularly informative for imbalanced datasets as it considers true and false positives and negatives.

7. Spearman's ρ Correlation:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Spearman's ρ measures the rank correlation between predicted and actual values, used in regression tasks to assess how well the model predicts the ordering of data points. Here, d_i represents the difference between the ranks of the i-th pair of values and n is the total number of data points.

These metrics collectively provide a comprehensive evaluation of our models' performance, enabling us to assess their effectiveness across different aspects of prediction accuracy and reliability. By consistently applying this set of metrics throughout our research, we ensure a standardized and thorough evaluation of our methodologies across various tasks and datasets.

2.6.2 Statistical Significance Analysis

Our investigation rigorously evaluated the statistical significance of observed differences employing paired Student's t-test and ANOVA, chosen for their pertinence and effectiveness in addressing our study's data structures and comparison requirements within protein sequence analysis.

Paired Student's t-test Application The utilization of the paired Student's t-test in our study was informed by its suitability for analyzing two sets of interrelated observations. This is particularly beneficial when the same dataset is subjected to different experimental conditions or methodologies. This statistical test is predicated on the assumptions of sample independence and the normal distribution of differences between paired observations. We ensured these assumptions were met by conducting a thorough examination and preprocessing of our data, thereby affirming the test's applicability to our analysis [Mow11].

One-Way ANOVA Utilization In contrast, one-way ANOVA (Analysis of Variance) was employed for its capacity to assess means across multiple distinct groups. This capability is crucial for

our study, aiming to compare model performances across various classifiers or hyperparameter configurations. The core assumptions for one-way ANOVA include the independence of groups, the normal distribution of data within each group, and the homogeneity of variances across groups, known as homoscedasticity. These assumptions were carefully verified through exploratory data analysis, ensuring their presence in our dataset [SW89].

Both statistical methods reported outcomes in terms of p-values, with values below 0.05 considered statistically significant. This conventional threshold indicates that the observed differences are unlikely to have arisen by chance, thus affirming the reliability of our findings. The application of these statistical tests in our study, drawing from the methodologies and justifications provided in relevant literature such as Arishe et al. for the paired Student's t-test [GGA+23] and Skubitz et al. for ANOVA [SBP+24], underscores our commitment to methodological precision and the validity of our conclusions in protein sequence analysis.

McNemar's Test McNemar's test [PSR20] is particularly useful in determining whether there is a significant difference in the performance of two models on the same dataset. It assesses the number of cases where one model is correct and the other is incorrect. The test calculates a chi-squared (χ^2) statistic with one degree of freedom, using the formula:

$$\chi^2 = \frac{(B - C)^2}{B + C} \tag{26}$$

where B represents the count where Model 1 is correct and Model 2 is wrong, and C is the count where Model 2 is correct and Model 1 is wrong.

The null hypothesis in McNemar's test posits that there is no difference in the proportions of correct predictions between the two models (i.e., $prop(b) \approx prop(c)$). The alternative hypothesis suggests a significant difference between the models. Decisions are made based on the p-value and a chosen significance level α. In our study, we use $\alpha = 0.05$. The interpretation is as follows:

- If p-value ¿ α, we fail to reject the null hypothesis, indicating no significant performance difference between the models.
- If p-value $\leqslant \alpha$, we reject the null hypothesis, suggesting a significant performance difference between the models.

Chapter 3

Evaluating ProtBERT-BFD in Membrane Protein Tasks

In Chapter 3 of this thesis, we investigate computational approaches for analyzing and predicting membrane proteins using protein language models and machine learning techniques [GB22,GB23b], transporters [GB23d], and ion channels [GB23c] in cellular biology. These proteins are fundamental for various life processes, such as signaling and transporting substances, and are a key focus in scientific research and medicine [Qui02].

In this study, we developed computational pipelines utilizing ProtBERT-BFD, a protein language model from the ProtTrans project, to predict and classify membrane proteins, transporters, and ion channels. Our approach aimed to enhance the accuracy and efficiency of existing bioinformatics methods for these specific protein classification tasks. These models are good at understanding the sequence of amino acids in proteins, which helps us gain better insights into their structure and function. Our research develops novel computational approaches that utilize protein language models for the prediction and classification of membrane proteins, aiming to enhance existing bioinformatics methods for protein sequence analysis.

For our analysis, we used ProtBERT-BFD (Section 2.2.3.4), a model from the ProtTrans project [EHD+21]. It is based on BERT [DCLT19] architecture (Section 2.2), a technique used in NLP, and is trained on a large number of protein sequences. This training makes it very effective in understanding the complex language of proteins.

Our research focused on three specific protein classification tasks: (1) predicting membrane proteins [GB22, GB23b] using the DS-M dataset containing 17,892 protein sequences, (2) identifying transporter proteins [GB23d] using the DS-T dataset with 1,560 sequences, and (3) classifying ion channels [GB23c] using the DS-C dataset comprising 4,564 sequences. Each task utilized a distinct dataset and was evaluated against state-of-the-art methods in their respective fields.

Our research includes three main tasks: predicting membrane proteins [GB22, GB23b], transporters [GB23d], and ion channels [GB23c]. Each task is a different aspect of studying membrane protein biology and comes with its own challenges and opportunities. We have published our findings in three separate papers [GB22, GB23d, GB23c].

3.1 Introduction

The primary objective of this research is to utilize ProtBERT-BFD, which is further detailed in Section 2.2.3.4 of this thesis. Our investigation is guided by four specific aims. First, we will evaluate ProtBERT-BFD's capability to identify the distinctive characteristics of membrane proteins, transporters, and ion channels. Second, we will assess the accuracy of Logistic Regression, a well-established machine learning technique, in classifying these proteins using comprehensive data derived from ProtBERT-BFD. Third, we intend to compare this combined

approach with conventional methods to determine if it offers a more efficient solution. Lastly, we aim to demonstrate how NLP-inspired models can be applied to protein sequence analysis, potentially paving the way for novel insights and methodologies in the field of bioinformatics.

3.2 Datasets

Our research utilizes carefully curated datasets to evaluate the performance of our computational approaches against state-of-the-art prediction models in the field of membrane protein analysis. These datasets (Table 6 and Table 7), named DS-M, DS-T, and DS-C, are crucial for our comparative analysis. We chose them because they have been widely used in previous research, providing a baseline for our experiments.

Table 6: Summary of Datasets Utilized in Experimental Comparative Analysis

Predictive Task	**Benchmarking Comparator**	**Dataset Source Reference**
Membrane Protein Prediction	TooT-M	DS-M [AB20a]
Transporter Protein Prediction	TooT-T	DS-T [AB20b]
Ion Channel Protein Prediction	DeepIon	DS-C [TO19]

The datasets corresponding to each predictive task are aligned with the benchmarking comparators from the literature.

3.2.1 Dataset for Membrane Protein Prediction (DS-M)

The DS-M dataset includes membrane and non-membrane proteins.

Source and Composition: This dataset, sourced from the Swiss-Prot database and used in the TooT-M project [AB20a], is specifically curated for membrane protein prediction. It contains a comprehensive range of membrane protein types and functions. The proteins were selected using a query for membrane location and reviewed status, ensuring high-quality data.

Preprocessing and Curation: The dataset underwent extensive preprocessing to ensure diversity and representation. This included removing sequences based on ”inferred from homology” evidence, eliminating sequences shorter than 50 amino acids, and excluding those without Gene Ontology molecular function annotation or only computational evidence annotation. Additionally, sequences with over 60% pairwise similarity were removed using the CD-HIT tool [FNZ+12] to enhance the dataset’s quality and utility.

Specifics and Relevance: The DS-M dataset consists of 17,892 protein sequences, including 8,828 membrane proteins and 9,064 non-membrane proteins, stratified into training (90%) and test (10%) sets. The membrane proteins were indeed obtained using a query specifying the location as ”membrane”. The non-membrane proteins, however, were deliberately included as negative examples to create a balanced dataset for binary classification tasks. These non-membrane proteins were carefully selected from other cellular locations (e.g., cytoplasmic, nuclear) to serve as contrasting examples, ensuring the model learns to distinguish between membrane and non-membrane proteins effectively. The functional types of membrane proteins include Transporters (25%), Receptors (13%), Enzymes (33%), and Others (29%). Structurally, the proteins are categorized as Single-pass (36%), Multi-pass (39%), Lipid-anchor (6%), GPI-anchor (3%), and Peripheral (16%), with approximately 75% being transmembrane proteins.

3.2.2 Dataset for Transporter Protein Prediction (DS-T)

The DS-T dataset includes transporters and non-transporters.

Source and Composition: This dataset, originally from the UniProt database [ABW+04] and utilized in the TrSSP project [MCZ14], consists of 10,780 initially well-characterized transporter, carrier, and channel proteins with different substrate specificity annotations.

Preprocessing and Curation: Mishra et al. [MCZ14] applied a rigorous filtering process to this dataset. They removed fragmented sequences, sequences with more than two substrate specificities, and biological function annotations based solely on sequence similarity. This resulted in a final dataset comprising 1,560 protein sequences, with 900 transporter and 660 non-transporter proteins, distributed into training and test sets.

Specifics and Relevance: The DS-T dataset is a recognized benchmark in transporter protein prediction, widely used by various predictive models.

3.2.3 Dataset for Ion Channel Protein Prediction (DS-C)

The DS-C dataset contains ion channels and other membrane proteins (non-ion channels).

Source and Composition: This dataset is sourced from the UniProt database [LDB+04] and used in previous works, such as the DeepIon [TO19] and MFPS_CNN [NHTO22] projects. It includes a total of 4,915 protein sequences, comprising a balanced mix of 301 ion channels, 351 ion transporters (not included in this experiment), and 4,263 membrane proteins.

Preprocessing and Curation: Taju and Ou [TO19] employed the BLAST algorithm [AMS+97] to exclude sequences with over 20% similarity, to guarantee the dataset's uniqueness and reduce redundancy.

Specifics and Relevance: The DS-C dataset's composition is particularly suited for ion channel prediction. It offers a distinct separation between ion channels and non-ion channel proteins (membrane proteins not classified as ion channels).

Table 7: Distribution of sequences in DS-M, DS-T, and DS-C datasets

Dataset	Class	Training	Test	Total
DS-M	Membrane protein	7,945	883	8,828
	Non-membrane protein	8,157	907	9,064
	Subtotal	*16,102*	*1,790*	*17,892*
DS-T	Transporter	780	120	900
	Non-transporter	600	60	660
	Subtotal	*1,380*	*180*	*1,560*
DS-C	Ion channel	241	60	301
	Non-ion channel	3,413	850	4,263
	Subtotal	*3,654*	*910*	*4,564*

DS-M: Membrane protein dataset (from TooT-M [AB20a]); DS-T: Transport proteins dataset (from TrSSP [MCZ14]); DS-C: Ion channel dataset (from DeepIon [TO19]).

3.3 Methodology

This section outlines the methodologies implemented in our study, focusing on employing ProtBERT-BFD and Logistic Regression to analyze membrane proteins, transporters, and ion channels.

ProtBERT-BFD: This BERT-based model is trained on a comprehensive corpus of protein sequences.

Logistic Regression: We have paired ProtBERT-BFD with Logistic Regression for classification purposes. This decision was influenced by two factors. First, Logistic Regression's simplicity allows us to evaluate the utility of a basic classifier in leveraging advanced protein language model representations. Second, its use in the ProtTrans project [EHD+21] as a baseline classifier facilitates direct comparison with other models, thus reinforcing the robustness of our study.

3.3.1 Representation Extraction

We extract sequence representations using ProtBERT-BFD's final hidden layer features. For each amino acid in a protein sequence, ProtBERT-BFD produces a 1024-dimensional vector representation. To obtain a single representation for the entire protein sequence, we apply a mean-pooling strategy. This involves calculating the average of all the amino acid representations in the sequence. Specifically, if a protein sequence contains n amino acids, we sum the 1024-dimensional vectors for all n amino acids and then divide by n. This process results in a single 1024-dimensional vector that represents the entire protein sequence, regardless of its original length. This mean-pooling approach allows us to convert variable-length protein sequences into fixed-size representations, which can be used for downstream tasks such as classification or clustering.

Two approaches are utilized for ProtBERT-BFD representations: frozen and fine-tuned. The frozen model analyzes pre-trained ProtBERT-BFD representations, while the fine-tuned approach involves adapting the model on our specific membrane protein dataset. MembraneBERT, our fine-tuned version of ProtBERT-BFD on DS-M, is specifically optimized for membrane proteins. MembraneBERT is accessible at (`https://huggingface.co/ghazikhanihamed/MembraneBERT`).

3.3.2 Fine-Tuning BERT Models

BERT models like ProtBERT-BFD and MembraneBERT are adapted (fine-tuned) to our specific classification task. This involves adding a classification layer and updating all initialized weights from the pre-training phase using our target dataset. We adopt HuggingFace's Trainer API for fine-tuning, maintaining the same hyperparameter settings as the ProtTrans project [EHD+21], except for the number of training epochs, which are experimentally determined. The fine-tuning hyperparameters are summarized in Table 8.

Table 8: Hyperparameters for Fine-Tuning ProtBERT-BFD

Hyperparameter	Value
Training Epochs	10 or 7
Training Batch Size	1
Evaluation Batch Size	32
Warmup Steps	1000
Weight Decay	0.01
Gradient Accumulation Steps	64
Random Seed	32

This employs all hyperparameters except one for the number of training epochs, align with those used in the ProtTrans project [EHD+21].

3.3.3 Experimental Design

We employed the pre-trained ProtBERT-BFD model from the ProtTrans project, available on HuggingFace [WDS+20], exploring both frozen and fine-tuned BERT representations. Our experiments involved creating representations for protein sequences and applying mean-pooling to the final hidden layer outputs. We used the speed high performance computing facility at Concordia University, equipped with NVIDIA Tesla P6 GPUs, for our computational needs. Logistic Regression, for classification, used the default hyperparameters from scikit-learn [Kra16].

3.3.4 Evaluation Strategies

In evaluating our models (TooT-BERT-M, TooT-BERT-T, and TooT-BERT-C), we employed 10-fold cross-validation (CV) and leave-one-out CV (LOOCV) for TooT-BERT-M, as well as 10-fold CV and 5-fold CV for TooT-BERT-T and TooT-BERT-C, respectively. We utilized McNemar's test

[PSR20] to compare the statistical significance of the predictive disagreements between different models. Our key evaluation metrics include sensitivity, specificity, accuracy, and Matthew's Correlation Coefficient (MCC).

3.3.5 Experimentation Overview

Our preliminary study examines the effectiveness of BERT model representations in analyzing membrane proteins. We investigated both frozen and fine-tuned models, developed MembraneBERT specifically for membrane proteins, and assessed these representations' performance.

3.4 Membrane Protein Prediction

3.4.1 Fine-tuning BERT Models Enhances Performance

The fine-tuning of ProtBERT-BFD on the DS-M dataset was aimed at determining the most effective representations for membrane proteins. Table 9 and Figure 9 present the impact of fine-tuning. Over ten training epochs, we observed a consistent increase in performance on the validation set. Starting with an MCC of 0.8049 and accuracy of 89.81% in the first epoch, it peaked at an MCC of 0.8421 and accuracy of 92.05% in the final epoch. This indicates that fine-tuned representations outperform frozen ones, despite the substantial computational resources required for fine-tuning.

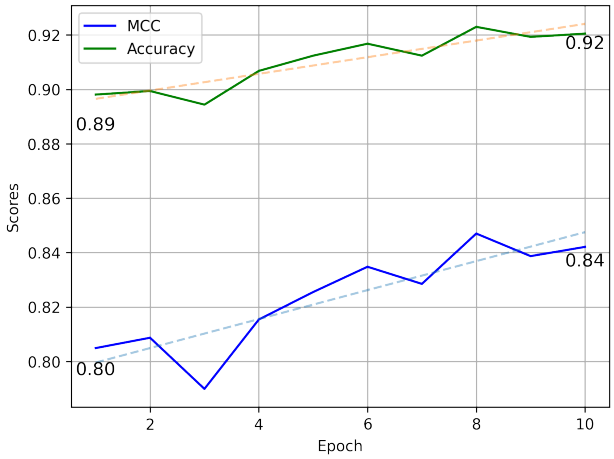

Figure 9: Enhancement Across Fine-Tuning Epochs

This figure illustrates the performance enhancement of the fine-tuned ProtBERT-BFD model across 10 training epochs, showcasing the impact of each epoch on the model's validation set accuracy and MCC.

Table 9: Frozen and Fine-tuned ProtBERT-BFD

Representation	sensitivity (%)	specificity (%)	accuracy (%)	MCC
Frozen	91.18	83.47	87.37	0.7492
Fine-tuned	91.28	93.61	92.46	0.8493

This table presents a comparative analysis of frozen and fine-tuned ProtBERT-BFD representations based on their performance metrics on the separate test set of membrane proteins.

3.4.2 Combining LR with Fine-tuned ProtBERT-BFD

We employed Logistic Regression as a classifier in conjunction with the fine-tuned ProtBERT-BFD representations. The results, including 10-fold cross-validation, leave-one-out

cross-validation, and the separate test set, are shown in Table 10. These results demonstrate that the combination of Logistic Regression and fine-tuned BERT representations effectively distinguishes membrane proteins from non-membrane proteins.

Table 10: Performance Metrics of TooT-BERT-M Using CV and Test Set

Evaluation Method	sensitivity (%)	specificity (%)	accuracy (%)	MCC
10-fold CV	98.19	98.74	98.47	0.97
Leave-One-Out CV	98.14	98.68	98.41	0.97
separate test set	91.28	93.61	92.46	0.85

The table presents the results of combining fine-tuned ProtBERT-BFD with Logistic Regression classifier across different evaluation methods.

3.4.3 Comparison with State-of-the-Art Approaches

Table 11 and Figure 10 compare TooT-BERT-M with the existing state-of-the-art membrane protein prediction methods. TooT-BERT-M outperformed both iMem-2LSAAC and MemType-2L across all evaluation measures and achieved better specificity than the existing TooT-M method, albeit with slightly lower sensitivity. This indicates that TooT-BERT-M is effective in reducing false positive predictions.

Table 11: Comparative Evaluation of Membrane Protein Prediction Methods

Method	sensitivity (%)	specificity (%)	accuracy (%)	MCC
iMem-2LSAAC	74.52	83.90	79.27	0.59
MemType-2L	88.67	90.19	89.44	0.79
TooT-M	**92.41**	92.50	**92.46**	**0.85**
TooT-BERT-M	91.28	**93.61**	**92.46**	**0.85**

This table compares the performance of different membrane protein prediction methods on an separate test set. The results for TooT-M and other methods are taken from [Alb20]. Boldface indicates the highest value in each column.

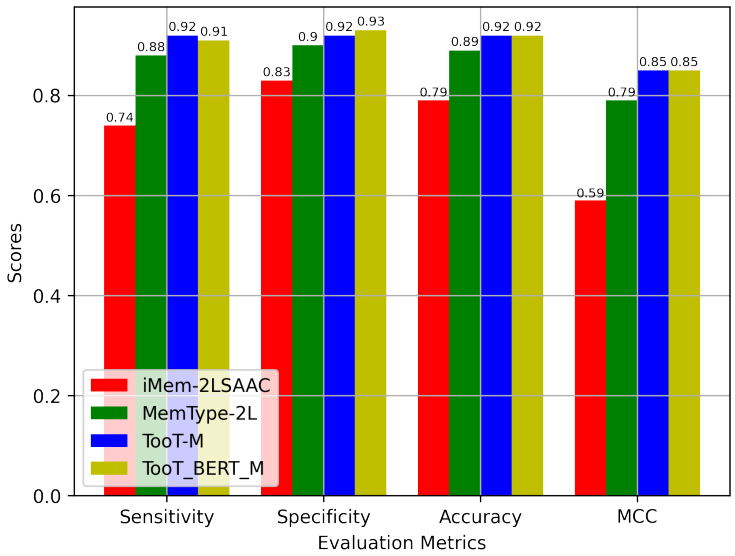

Figure 10: Comparative Analysis of Membrane Protein Prediction Methods

This figure presents a comparison of various state-of-the-art membrane protein prediction methods, highlighting the performance of TooT-BERT-M and TooT-M across different assessment measures.

3.4.4 Comparison of TooT-BERT-M and TooT-M

The performance of TooT-BERT-M and TooT-M is comparable, with TooT-BERT-M demonstrating higher specificity. This suggests that TooT-BERT-M, with its simpler architecture and single encoding approach, provides a more effective model for membrane protein prediction.

3.4.5 Statistical Analysis using McNemar's Test

McNemar's test (Table 12) was applied to compare the predictions of TooT-BERT-M and TooT-M. The obtained chi-square value of 58 and a p-value of $2e^{-20}$ indicate significant differences in the predictions of the two models, suggesting that the discrepancies are not due to random chance.

In summary, the results of TooT-BERT-M in membrane protein prediction demonstrate its effectiveness and potential as a tool for accurately classifying membrane proteins.

The predictive capabilities between TooT-BERT-M and TooT-M indicate an opportunity to employ ensemble methods that combine multiple models to capitalize on their distinct strengths for superior accuracy.

Table 12: Contingency Table for McNemar's Test

	TooT-BERT-M	
TooT-M	Correct	Wrong
Correct	1450	58
Wrong	205	77

This contingency table is used for McNemar's test to compare the prediction discrepancies between TooT-BERT-M and TooT-M.

3.5 Transporter Prediction

3.5.1 Fine-Tuning ProtBERT-BFD and MembraneBERT

We embarked on a comparative analysis of two PLM models, namely ProtBERT-BFD and MembraneBERT, utilizing the DS-T dataset.

The process of fine-tuning these models was undertaken to optimize their performance. In our study, fine-tuning ProtBERT-BFD involved updating all initialized weights from the pre-training phase using our specific protein classification datasets (DS-M, DS-T, and DS-C). We employed the HuggingFace Trainer API for this process, maintaining hyperparameters consistent with the ProtTrans project, except for the number of training epochs, which we experimentally determined (10 epochs for membrane and transporter prediction, 7 for ion channel classification). The impact of this fine-tuning process on the model performance is visually represented in Figure 11. It elucidates the progression of both models across various epochs, showcasing improvements in terms of accuracy and the MCC.

The results of this comparative study are enlightening. ProtBERT-BFD, a model trained on a more comprehensive set of protein sequences, demonstrated a remarkable improvement in its predictive capabilities with each epoch. This model's performance heightened notably from an initial MCC of zero and 56% accuracy, ultimately reaching an MCC of 0.77 and an accuracy of 87% on the validation set. These metrics indicate the effectiveness of the fine-tuning process.

In contrast, while MembraneBERT also showed improvements through fine-tuning, its performance did not match that of ProtBERT-BFD.

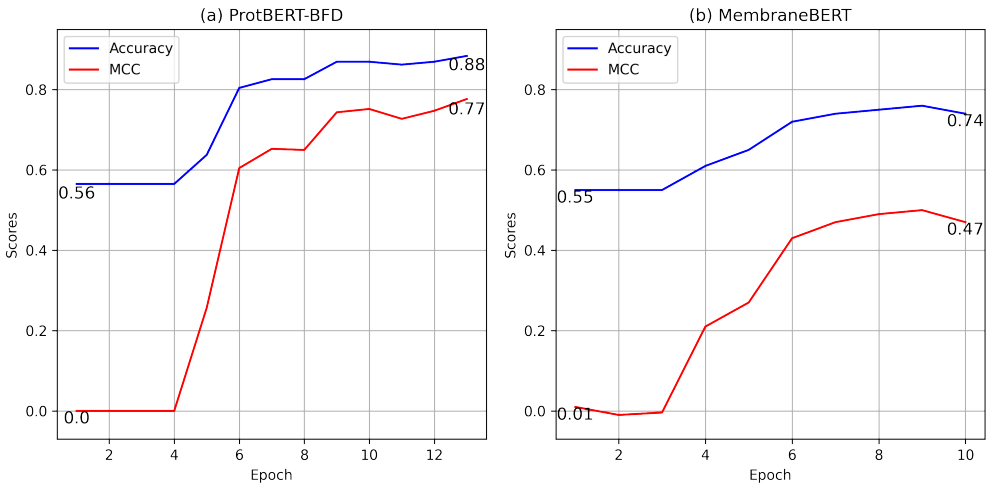

Figure 11: Impact of this fine-tuning process on the model performance

This figure depicts the results of fine-tuning the ProtBERT-BFD (left) and MembraneBERT (right) with accuracy and MCC metrics at each epoch on the validation set. The y-axis and x-axis display the scores and epochs, respectively.

3.5.2 Logistic Regression with Fine-tuned ProtBERT-BFD

The performance of logistic regression, when used in conjunction with fine-tuned representations of both ProtBERT-BFD and MembraneBERT, was extensively analyzed. This comparative study, detailed in Table 13 and Table 14, reveals that logistic regression with fine-tuned ProtBERT-BFD consistently outperformed the MembraneBERT model across various metrics on the separate test set.

Table 13: LR with ProtBERT-BFD and MembraneBERT on TooT-BERT-T

Model	Sen(%)		Spc(%)		Acc(%)		MCC	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
ProtBERT-BFD frozen	76.67	80.00 $\pm$ 3.94	**90.83**	82.69 $\pm$ 5.32	86.11	81.52 $\pm$ 4.29	0.6840	0.6262 $\pm$ 0.0854
ProtBERT-BFD fine-tuned	**95.83**	96.79 $\pm$ 4.84	90.00	**97.17 $\pm$ 4.72**	**93.89**	96.96 $\pm$ 4.68	**0.8620**	0.9387 $\pm$ 0.0945
MembraneBERT frozen	88.33	80.51 $\pm$ 3.75	68.33	77.50 $\pm$ 6.84	81.67	79.20 $\pm$ 3.78	0.5799	0.5797 $\pm$ 0.0791
MembraneBERT fine-tuned	86.67	**98.08 $\pm$ 5.36**	85.00	97.00 $\pm$ 7.92	86.11	**97.61 $\pm$ 6.46**	0.6989	**0.9512 $\pm$ 0.1318**

This table summarizes the 10-fold CV and separate test set performance of frozen/fine-tuned representations from the ProtBERT-BFD and MembraneBERT models in terms of sensitivity, specificity, accuracy, and MCC. The maximum value for each column is displayed in boldface.

When examining the specific performance metrics, the fine-tuned ProtBERT-BFD achieved a sensitivity of 95.83% and a specificity of 90.00%, culminating in an accuracy of 93.89% and an MCC of 0.8620. These figures not only surpass the performance metrics of MembraneBERT but also significantly improve upon the results achieved with the frozen ProtBERT-BFD model. This improvement highlights the critical role of fine-tuning in adapting the model more closely to the specificities of the transporter protein prediction task.

Table 14: ProtBERT-BFD and MembraneBERT models for TooT-BERT-T

Model	Sen(%)	Spc(%)	Acc(%)	MCC
ProtBERT-BFD frozen	76.67	**90.83**	86.11	0.6840
ProtBERT-BFD fine-tuned	**95.83**	90.00	**93.89**	**0.8620**
MembraneBERT frozen	88.33	68.33	81.67	0.5799
MembraneBERT fine-tuned	86.67	85.00	86.11	0.6989

This table illustrates the sensitivity, specificity, accuracy, and MCC of the frozen/fine-tuned representations from the ProtBERT-BFD and MembraneBERT models on the separate test set using a Logistic Regression classifier. Each column's maximum value is denoted in boldface.

In summary, logistic regression with fine-tuned ProtBERT-BFD emerges as a highly effective approach for transporter protein prediction.

3.5.3 Comparison of TooT-BERT-T with State-of-the-Art Models

Table 15 and Figure 12 benchmark the performance of TooT-BERT-T against other state-of-the-art models in transporter protein prediction.

Table 15: Comparative performance of TooT-BERT-T with state-of-the-art

Method	Sen(%)		Spc(%)		Acc(%)		MCC	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
SCMMTP [LVY+15]	80.00	83.76	68.33	77.68	76.11	81.12	0.47	0.62
TrSSP [MCZ14]	76.67	76.67	81.67	78.46	80.00	78.99	0.57	0.58
Nguyen et al. [NLH+19]	**100.00**	83.14	77.50	84.48	85.00	83.94	0.73	0.68
TooT-T [AB20b]	94.17	90.15	88.33	89.97	92.22	90.07	0.82	0.80
TooT-BERT-T	95.83	**96.79**	**90.00**	**97.17**	**93.89**	**96.96**	**0.86**	**0.94**

This table compares the outcomes of various techniques using sensitivity, specificity, accuracy, and MCC metrics on the CV and separate test set. Results taken from [AB20b]. The maximum value for each column is displayed in boldface.

The analysis revealed that TooT-BERT-T, using fine-tuned ProtBERT-BFD representations and a logistic regression classifier, outperformed existing methods in almost all evaluation metrics. Most notably, TooT-BERT-T demonstrated a remarkable specificity rate, which is crucial in reducing false positive predictions.

TooT-BERT-T achieved a specificity of 90.00%, a sensitivity of 95.83%, an accuracy of 93.89%, and an MCC of 0.86. These figures indicate its balanced performance in correctly identifying both positive and negative cases.

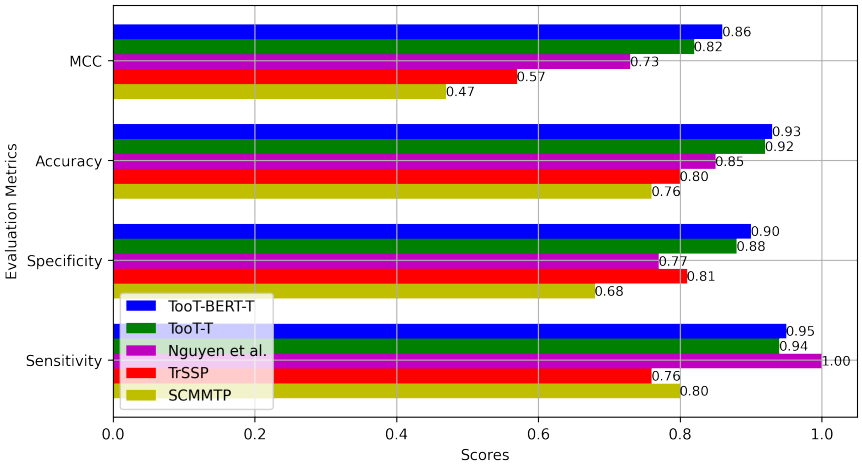

Figure 12: TooT-BERT-T comparison of methods

A comparison of methodologies is shown in this figure. Blue, green, magenta, red, and yellow denote TooT-BERT-T, TooT-T, Nguyen et al., TrSSP, and SCMMTP, respectively.

3.6 Ion Channel Classification

3.6.1 Performance Analysis of ProtBERT-BFD and MembraneBERT

We conducted a comparative study of two BERT models, ProtBERT-BFD and MembraneBERT, utilizing the DS-C dataset.

Impact of Fine-Tuning Across Epochs The process of fine-tuning both ProtBERT-BFD and MembraneBERT played a pivotal role in enhancing their performance. Figure 13 illustrates

this progression, showcasing how each epoch of fine-tuning incrementally improved the models' accuracy and MCC on the validation set.

Superior Performance of ProtBERT-BFD Figure 13 revealed a clear performance edge of ProtBERT-BFD over MembraneBERT, particularly in terms of accuracy and MCC. Upon fine-tuning, ProtBERT-BFD demonstrated remarkable progress, evolving from an initial 0 MCC and 6% accuracy to an impressive 0.90 MCC and 98% accuracy. In contrast, while MembraneBERT also showed notable improvements — its MCC increasing from 0.06 to 0.82 and accuracy from 22% to 97% — it did not reach the performance peaks of ProtBERT-BFD.

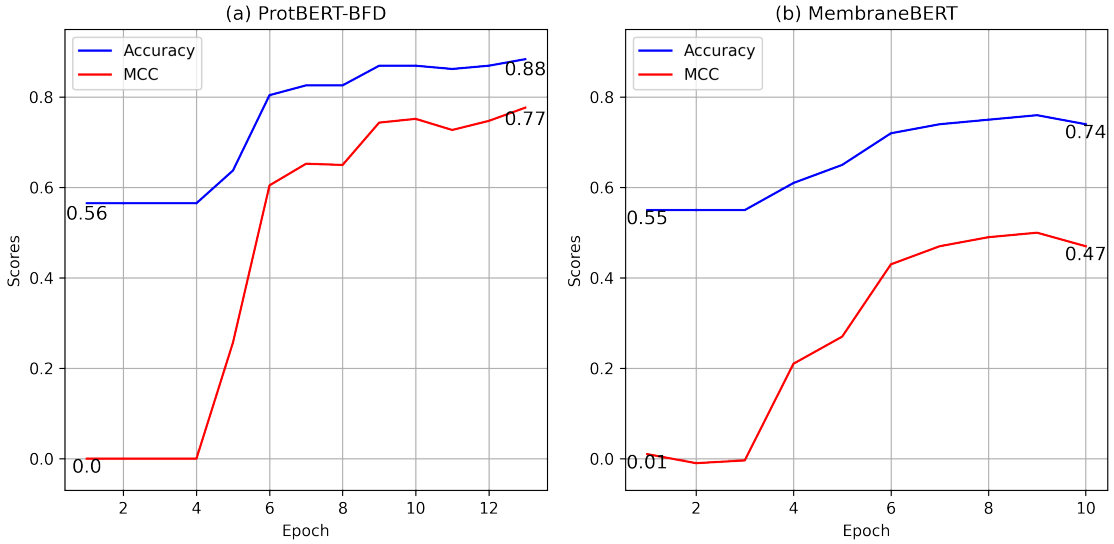

Figure 13: The effect of fine-tuning on DS-C

The outcomes of fine-tuning the ProtBERT-BFD (left) and MembraneBERT (right) using accuracy and MCC metrics at each epoch on the validation set are shown in this figure. The y-axis and x-axis, respectively, indicate the scores and epochs.

The superior performance of ProtBERT-BFD can be attributed to its training on a more extensive range of protein sequences. Additionally, we hypothesize that the process of fine-tuning ProtBERT-BFD to create MembraneBERT may have led to a phenomenon known as catastrophic forgetting. This occurs when a neural network, while learning new tasks, abruptly and severely forgets previously learned information.

3.6.2 Evaluation of Fine-tuned Representations on separate test set

Performance Assessment in Different States Table 16 shows the performance of both ProtBERT-BFD and MembraneBERT models on an separate test set.

Table 16: ProtBERT-BFD and MembraneBERT models for TooT-BERT-C

Model	Sen(%)	Spc(%)	Acc(%)	MCC
ProtBERT-BFD frozen	30.00	99.53	94.95	0.4771
ProtBERT-BFD fine-tuned	**76.67**	**99.76**	**98.24**	**0.8486**
MembraneBERT frozen	36.67	**99.76**	95.60	0.5642
MembraneBERT fine-tuned	66.67	99.41	97.25	0.7564

This table shows the sensitivity, specificity, accuracy, and MCC of the frozen/fine-tuned representations from the ProtBERT-BFD and MembraneBERT models on the separate test set using a Logistic Regression classifier. The maximum value for each column is indicated in boldface.

Fine-tuned ProtBERT-BFD's Superior Performance The fine-tuned ProtBERT-BFD model exhibited a remarkable improvement in performance compared to its frozen state and to both

states of the MembraneBERT model. In its fine-tuned form, ProtBERT-BFD achieved a sensitivity of 76.67%, a specificity of 99.76%, an outstanding accuracy of 98.24%, and an MCC of 0.8486. These results were significantly higher than those of the frozen ProtBERT-BFD model and both states of the MembraneBERT model. The fine-tuned MembraneBERT, while showing improved performance over its frozen counterpart, still lagged behind the fine-tuned ProtBERT-BFD, particularly in terms of sensitivity and MCC.

Implications for Ion Channel Protein Prediction The superior performance of the fine-tuned ProtBERT-BFD model has reinforced the importance of fine-tuning in adapting pre-trained models to specific tasks.

3.6.3 Logistic Regression Performance

Logistic Regression Performance with ProtBERT-BFD With performance of logistic regression, when combined with the fine-tuned representations of both ProtBERT-BFD and MembraneBERT, was rigorously evaluated. The assessment focused on several key metrics: sensitivity, specificity, accuracy, and Matthews Correlation Coefficient (MCC), with results detailed in Table 17. This evaluation was carried out both on an separate test set and through a 5-fold cross-validation (CV) process to ensure the robustness and reliability of the findings.

Table 17: LR with ProtBERT-BFD and MembraneBERT on TooT-BERT-C

Model	Sen(%)		Spc(%)		Acc(%)		MCC	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
ProtBERT-BFD frozen	30.00	18.70 $\pm$ 7.02	99.53	**99.97 $\pm$ 0.06**	94.95	94.61 $\pm$ 0.52	0.4771	0.4078 $\pm$ 0.0843
ProtBERT-BFD fine-tuned	**76.67**	86.71 $\pm$ 5.55	**99.76**	99.82 $\pm$ 0.11	**98.24**	98.96 $\pm$ 0.43	**0.8486**	0.9123 $\pm$ 0.0375
MembraneBERT frozen	36.67	22.83 $\pm$ 3.78	**99.76**	99.91 $\pm$ 0.12	95.60	94.83 $\pm$ 0.33	0.5642	0.4504 $\pm$ 0.0485
MembraneBERT fine-tuned	66.67	**95.43 $\pm$ 2.43**	99.41	99.79 $\pm$ 0.15	97.25	**99.51 $\pm$ 0.30**	0.7564	**0.9597 $\pm$ 0.0243**

This table shows the sensitivity, specificity, accuracy, and MCC of the 5-fold CV with the *mean* $\pm$ *sd* and an separate test set of frozen/fine-tuned representations from the ProtBERT-BFD and MembraneBERT models. Each column's maximum value is shown in boldface.

Comparative Performance Analysis With analysis revealed that logistic regression, when paired with the fine-tuned ProtBERT-BFD representation, outperformed the combination with MembraneBERT across almost all metrics on the separate test set. Notably, the fine-tuned ProtBERT-BFD representation achieved a higher accuracy and MCC, indicating a better overall performance in classifying ion channels. This trend was also observed in the 5-fold CV results, although the differences in performance metrics were less pronounced.

Implications of Logistic Regression's Effectiveness Our experiments demonstrated that logistic regression, when combined with fine-tuned ProtBERT-BFD representations, achieved high performance in ion channel classification. Specifically, on the DS-C separate test set, this approach yielded a sensitivity of 76.67%, specificity of 99.76%, accuracy of 98.24%, and MCC of 0.8486. These results surpassed the performance of existing methods such as DeepIon [TO19] and MFPS_CNN [NHTO22] in terms of specificity, accuracy, and MCC, indicating the potential of this approach for accurate ion channel prediction. It demonstrates that the integration of a simple yet robust classifier like logistic regression with a highly contextualized representation model like ProtBERT-BFD can lead to highly accurate and reliable predictions.

3.6.4 Comparative Analysis with State-of-the-Art

Table 18 and Figure 14 presents comparative analysis of TooT-BERT-C.

Evaluation Across Multiple Performance Metrics The evaluation of TooT-BERT-C spanned several critical metrics: sensitivity, specificity, accuracy, and MCC. These metrics provide a holistic view of the model's performance, capturing its ability to correctly identify ion channels (sensitivity), correctly reject non-ion channels (specificity), overall correctness (accuracy), and a

balanced measure of true positives and negatives (MCC). Our analysis revealed that TooT-BERT-C outperforms the existing methods in these metrics (except sensitivity), particularly in terms of specificity and MCC, indicating its robustness in minimizing false positives and its overall reliability.

Table 18: Comparative performance of TooT-BERT-C with state-of-the-art

Method	Sen(%)		Spc(%)		Acc(%)		MCC	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
DeepIon [TO19]	68.33	89.20	87.72	84.89	86.53	87.05	0.37	0.75
MFPS_CNN [NHTO22]	**76.70**	**95.00**	95.80	98.00	94.60	96.50	0.62	**0.93**
TooT-BERT-C	76.67	86.71	**99.76**	**99.82**	**98.24**	**98.96**	**0.85**	0.91

This table compares the performance of previous approaches on the CV and separate test set using sensitivity, specificity, accuracy, and MCC evaluation. Each column's maximum value is shown in boldface.

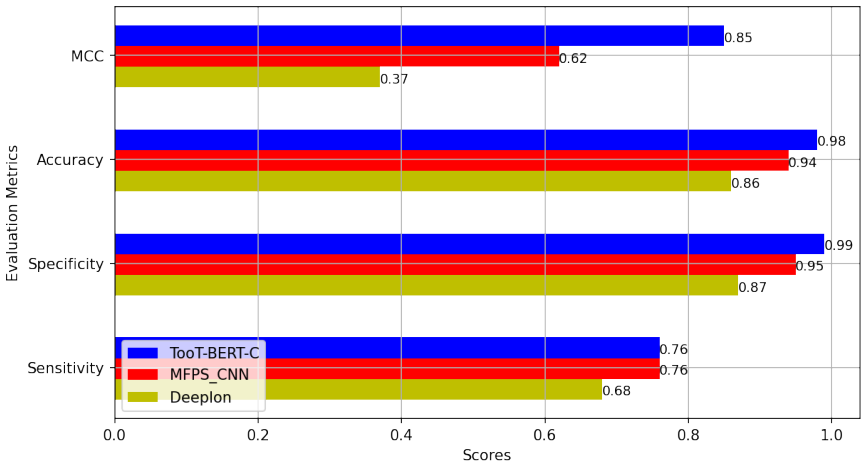

Figure 14: TooT-BERT-C comparison with other methods

A comparison of methodologies is shown in this figure. Blue, red, and yellow denote TooT-BERT-C, MFPS_CNN, and DeepIon, respectively.

Advantages and Methodological Strengths of TooT-BERT-C TooT-BERT-C leverages the BERT model's powerful contextual representation capabilities. With logistic regression as a classifier adds to its strengths. This combination of a nuanced, context-aware representation model with a straightforward, interpretable classifier results in a method that is not only highly accurate but also transparent in its decision-making process. This transparency is vital in scientific research, where understanding the "why" behind predictions is as important as the predictions themselves.

3.7 Conclusion

This study has systematically evaluated the application of ProtBERT-BFD and logistic regression in the classification of protein sequences across several critical tasks. The key insights and contributions are:

Enhancement Through Fine-Tuning: In our membrane protein prediction task using the DS-M dataset, fine-tuning ProtBERT-BFD resulted in significant performance improvements. Specifically, over ten training epochs, the model's performance on the validation set improved from an initial MCC of 0.8049 and accuracy of 89.81% to a final MCC of 0.8421 and accuracy of 92.05%. This improvement demonstrates the effectiveness of adapting the pre-trained model to our specific membrane protein classification task.

Superiority of Fine-Tuned Representations: A comparative analysis highlights that fine-tuned representations significantly outperform their frozen counterparts across multiple metrics, including sensitivity, specificity, and overall accuracy, thereby affirming the value of dynamic model adjustments in enhancing predictive precision.

Integration of Logistic Regression: The synergy between logistic regression and fine-tuned BERT models has proven exceptionally potent, particularly for distinguishing membrane proteins, which enhances both interpretability and predictive performance.

Comparative Advantages: The TooT-BERT models, including TooT-BERT-T and TooT-BERT-C, demonstrate superior performance over existing methods, particularly in specificity and MCC. This highlights their potential in reducing false positives and enhancing reliability in protein prediction tasks.

Contextual Understanding and Potential Limitations in PLMs: The architectural underpinnings of the BERT-based PLMs contextualize amino acid sequences, which fundamentally enhances the predictive accuracy of PLM-based classifiers beyond that of traditional methods. However, it is crucial to acknowledge that PLMs, including BERT-based models like ProtBERT-BFD, can potentially ”hallucinate” or generate false predictions, especially when faced with inputs significantly different from their training data.

Hallucination in PLMs refers to the model generating confident but incorrect outputs, often by extrapolating beyond its training distribution. In the context of protein prediction tasks, this could manifest as misclassifications or erroneous structural predictions, particularly for novel or rare protein sequences.

Chapter 4

Advancing Membrane Protein Classification

This chapter examines the integration of three specific PLMs - ProtBERT, ProtBERT-BFD, and MembraneBERT - with various machine learning classifiers, including k-Nearest Neighbors (kNN), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), Feed-Forward Neural Network (FFNN), and Convolutional Neural Network (CNN), for the classification of membrane proteins, transporters, and ion channels. We used three distinct datasets: DS-M (17,892 sequences) for membrane proteins, DS-T (1,560 sequences) for transporters, and DS-C (4,564 sequences) for ion channels.

Our research addresses several key questions. First, we seek to determine whether PLMs can outperform state-of-the-art classifiers for membrane protein classification. Second, we aim to identify the optimal approach for combining PLMs with downstream machine learning or deep learning classifiers for these specific protein tasks. Finally, we investigate how the performance of PLM-based methods compares across different protein classification tasks, specifically for membrane proteins.

To address these questions, we developed and evaluated three novel methodologies: TooT-BERT-CNN-M for membrane proteins discrimination, TooT-BERT-CNN-T for transporter protein classification and TooT-BERT-CNN-C for ion channel classification. These approaches integrate fine-tuned embeddings from ProtBERT, ProtBERT-BFD, and MembraneBERT with convolutional neural networks (CNNs) and traditional machine learning classifiers.

Our investigation revealed that the combination of ProtBERT-BFD with CNN, which we term TooT-BERT-CNN-M, achieved the highest accuracy (94.02%) and MCC (0.88) on the separate test set for membrane protein classification. This performance surpassed both traditional machine learning methods and previous state-of-the-art classifiers such as iMem-2LSAAC (79.27% accuracy, 0.59 MCC) and MemType-2L (89.44% accuracy, 0.79 MCC).

In our membrane protein classification task, fine-tuning ProtBERT-BFD improved performance over ten training epochs. The model's performance on the validation set increased from an initial MCC of 0.8049 and accuracy of 89.81% to a final MCC of 0.8421 and accuracy of 92.05%. This improvement demonstrates the benefit of task-specific adaptation through fine-tuning compared to using frozen representations.

4.1 Methodology

We utilize the DS-M, DS-T, and DS-C datasets for membrane proteins, transporters, and ion channels, respectively (see Section 3.2). Our approach integrates state-of-the-art PLMs with various machine learning classifiers to address challenges in protein sequence analysis.

4.1.1 Classifiers

4.1.1.1 Traditional Classifiers

We incorporated several commonly used bioinformatics and protein classifiers into our TooT-BERT framework. These include Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), k-Nearest Neighbor (kNN), and Feed-Forward Neural Network (FFNN). We implemented these classifiers using scikit-learn [Kra16]. This consistent set of classifiers was applied across our three projects: TooT-BERT-CNN-M for membrane proteins, TooT-BERT-CNN-T for transporters, and TooT-BERT-CNN-C for ion channels.

4.1.1.2 Convolutional Neural Network

Our CNN architecture consists of the following layers (Figure 15):

Convolution Layer The initial layers function as motif scanners [ON15], employing 2D convolution (Conv2D) to transform input data into new feature matrices. Each convolutional layer is followed by ReLU activation.

1-Max Pooling and Dropout Layer Following each convolutional layer is a 1-max pooling layer, which outputs the maximum value from its corresponding convolutional output. Dropout [SHK+14] is applied after pooling to prevent overfitting and enhance feature robustness.

Fully-Connected Layers The final component consists of three fully connected layers with 256, 64, and 2 neurons, respectively. The last layer corresponds to binary classification, applying classification to the flattened feature vectors derived from previous layers.

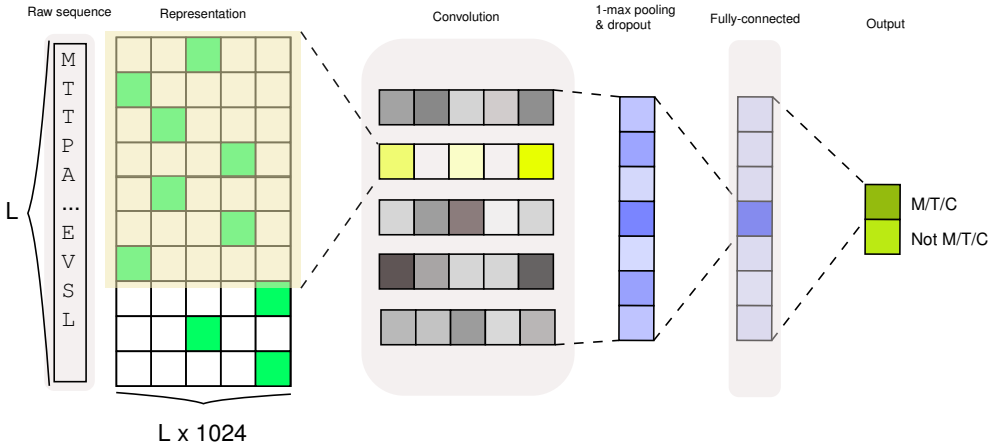

Figure 15: CNN schematic architecture

Workflow of processing sequence representations from PLMs through a CNN neural network for membrane (M), transporter (T), and ion channel (C) proteins classification. The convolution layer serves as the first layer, followed by 1D max-pooling and dropout. The final layer comprises a fully connected feed-forward neural network, which outputs the probabilities for each class, transporter or non-transporter. “L” denotes the length of the protein sequence.

4.1.2 Hyperparameters

We conducted a grid search to optimize the hyperparameters of classifiers. The random seed was set to 32 for all experiments. For the CNN, we explored different numbers of epochs, testing the model performance at 10, 20, and 30 epochs. Learning rates were varied between 0.00001, 0.0001, and 0.001 to find the optimal rate of convergence. We evaluated batch sizes of 4, 8, and 16 to balance between computational efficiency and model stability. Two configurations of convolutional filters were examined: one with 128 filters in each layer, and another with 256 filters in each layer. Kernel sizes were varied between two configurations: [5, 7, 9] and [7, 7, 7]. To

prevent overfitting, we tested dropout rates of 0.2 and 0.3. For optimization, we utilized mini-batch gradient descent with cross-entropy loss and L2 regularization. We employed AdamW [LH19], an enhanced version of Adam [KB17], as our optimizer.

For the Support Vector Machine (SVM), we evaluated C values of [0.1, 1, 10, 100], gamma values of [0.1, 1, 10], and kernels including linear, rbf, and sigmoid. Random Forest (RF) hyperparameters ranged from 50 to 200 estimators, max depths of 5, 10, or None, and minimum samples split of 2, 5, or 10. For k-Nearest Neighbors (kNN), we tested 3 to 9 neighbors, uniform and distance weights, and ball_tree, kd_tree, and brute algorithms. Logistic Regression (LR) was evaluated with l1 and l2 penalties, C values of [0.1, 1, 10, 100], and liblinear and saga solvers. The Feed-Forward Neural Network (FFNN) grid included hidden layer sizes of [(512, 256, 64), (512,), (256,)], ReLU and tanh activations, and Adam and SGD solvers.

4.1.3 Training and Evaluation

We implemented two approaches for classifier training across our three projects (membrane proteins, transporters, and ion channels): traditional classifiers with mean-pooled representations and CNN-based classification without mean-pooling.

For traditional classifiers, we augmented the PLMs with a classification layer and trained the network on the respective datasets (DS-M, DS-T, DS-C). Sequences were tokenized into amino acids and processed through the BERT model. We extracted fine-tuned representations using mean-pooling:

$$R_S = Mean(R_{s_1}^{1024}, R_{s_2}^{1024}, R_{s_3}^{1024}, ..., R_{s_n}^{1024})^{D=1024} \tag{27}$$

where $R_{s_i}^{1024}$ represents the 1024-dimensional vector of the i^{th} amino acid in a sequence of length n. These representations were then input into traditional classifiers for training and evaluation.

For the CNN-based approach (Figure 20), we directly fed the last layer representations from PLMs into the CNN's convolutional layer, bypassing mean-pooling. This architecture was consistent across all three projects. Both methods utilized Cross-Entropy loss and Adam optimizer for training, following the original BERT [DCLT19] and ProtTrans project [EHD+21] methodologies.

We evaluated model performance using four metrics: sensitivity (Sen), specificity (Spc), accuracy (Acc), and Matthews's correlation coefficient (MCC).

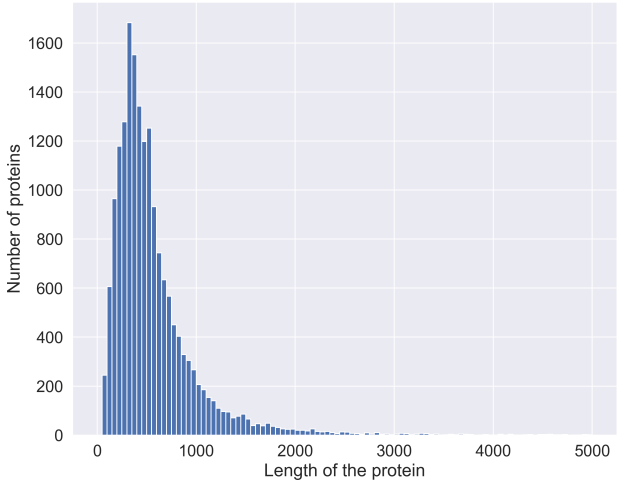

Figure 16: Membrane proteins length distribution
Histogram showing the frequency distribution of membrane protein sequence lengths.

4.1.4 Sequence Analysis

We analyzed the length distributions of protein sequences for all three projects to assess the impact of our 1024 amino acid truncation limit. Figure 16, Figure 17, and Figure 18 illustrate these

distributions for membrane proteins, transporters, and ion channels, respectively.

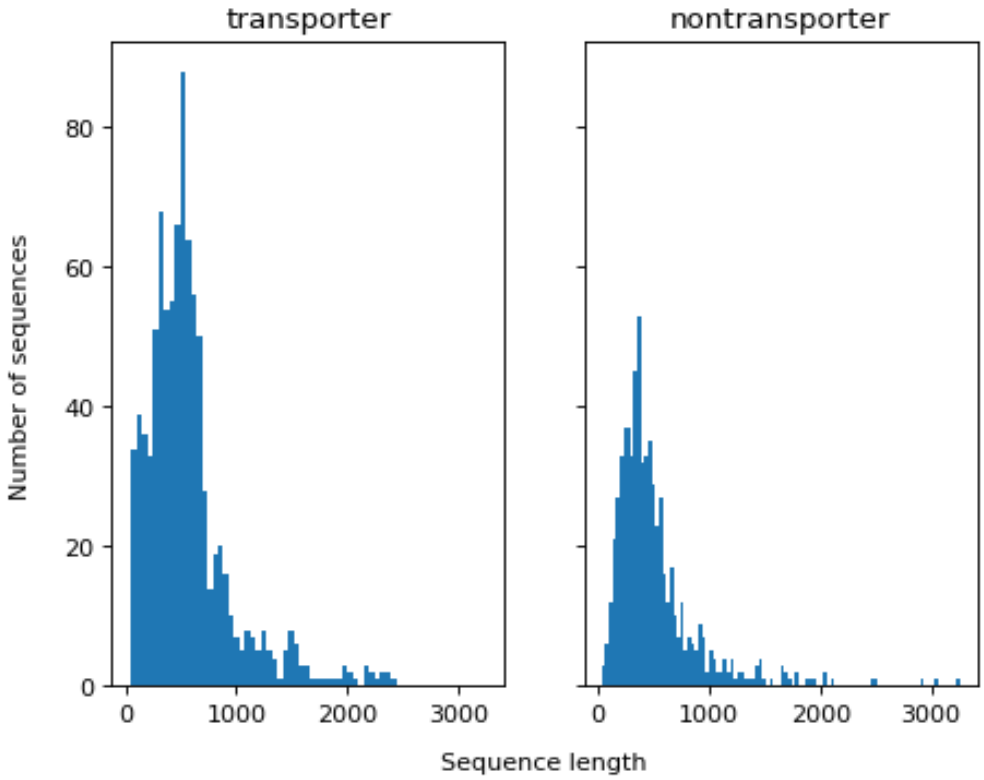

Figure 17: Sequence length distribution: Transporters
Histograms depicting the frequency distribution of protein sequence lengths for transporters (left) and non-transporters (right).

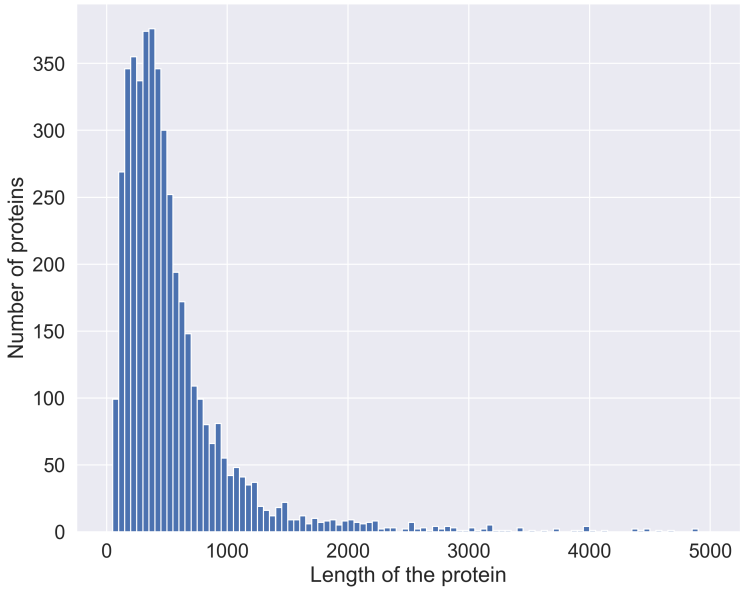

Figure 18: Distribution of protein lengths for ion channels
Histogram illustrating the frequency distribution of ion channel protein sequence lengths.

4.1.5 Experimental Setup

We implemented a consistent methodology across our three projects, integrating pre-trained PLMs (ProtBERT, ProtBERT-BFD, and MembraneBERT) with both traditional classifiers and

CNN. Figure 19 and Figure 20 illustrate the workflows for traditional classifiers and CNN-based approaches, respectively.

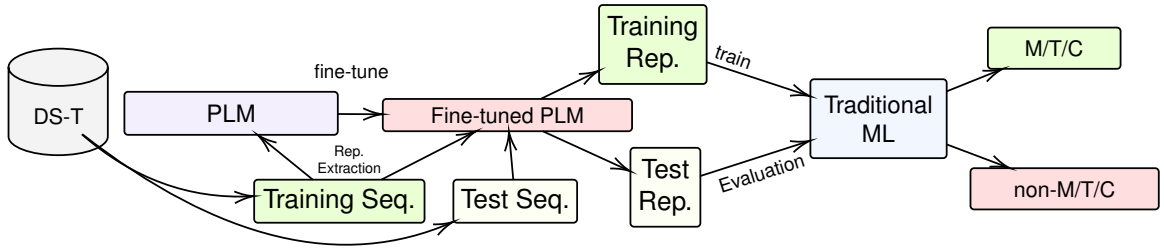

Figure 19: Proposed method of using PLMs and traditional classifiers

Schematic representation of the proposed method for membrane (M), transporter (T), and ion channel (C) proteins classification, which combines protein language models (PLMs) such as ProtBERT, ProtBERT-BFD, and MembraneBERT with traditional machine learning classifiers to distinguish transporters from non-transporters. The process entails fine-tuning the BERT-based models using the training and validation sets and subsequently extracting representations from the training and test sets to assess the performance of traditional classifiers, including kNN, RF, LR, SVM, and FFNN.

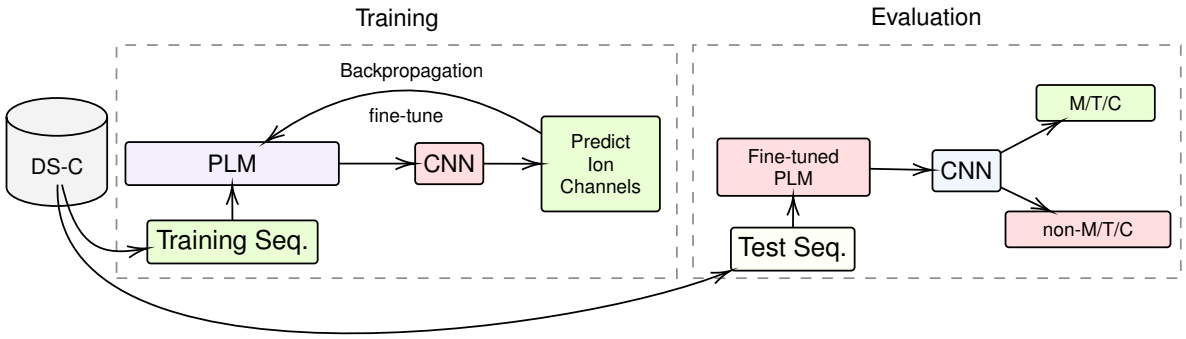

Figure 20: Proposed method for ion channel classification using CNN

This figure illustrates the proposed methodology for membrane (M), transporter (T), ion channel (C) proteins classification using a deep learning classifier, CNN. The process entails the concurrent training of CNN and fine-tuning of protein language models (PLMs), which include ProtBERT, ProtBERT-BFD, and MembraneBERT.

4.2 Results and Discussion: Membrane Proteins

4.2.1 Representation Analysis

T-SNE visualization (Figure 21) with default parameters and randome seed of zero of the ProtBERT and ProtBERT-BFD representations shows clear separation between membrane and non-membrane proteins. This indicates that these models effectively capture features distinguishing the two classes. ProtBERT-BFD appears to display more compact clustering compared to ProtBERT, suggesting potentially more consistent representations. However, we acknowledge that this observation is based on visual inspection of the dimensionality reduction plots rather than quantitative metrics.

4.2.2 Classifier Performance

We evaluated various classifiers using representations from ProtBERT and ProtBERT-BFD (Table 19). The CNN classifier consistently outperformed other methods, achieving the highest accuracy (94.02%) and MCC (0.88) on the separate test set when using ProtBERT-BFD

representations. Other classifiers, particularly Logistic Regression and Random Forest, also showed strong performance with MCC values above 0.84.

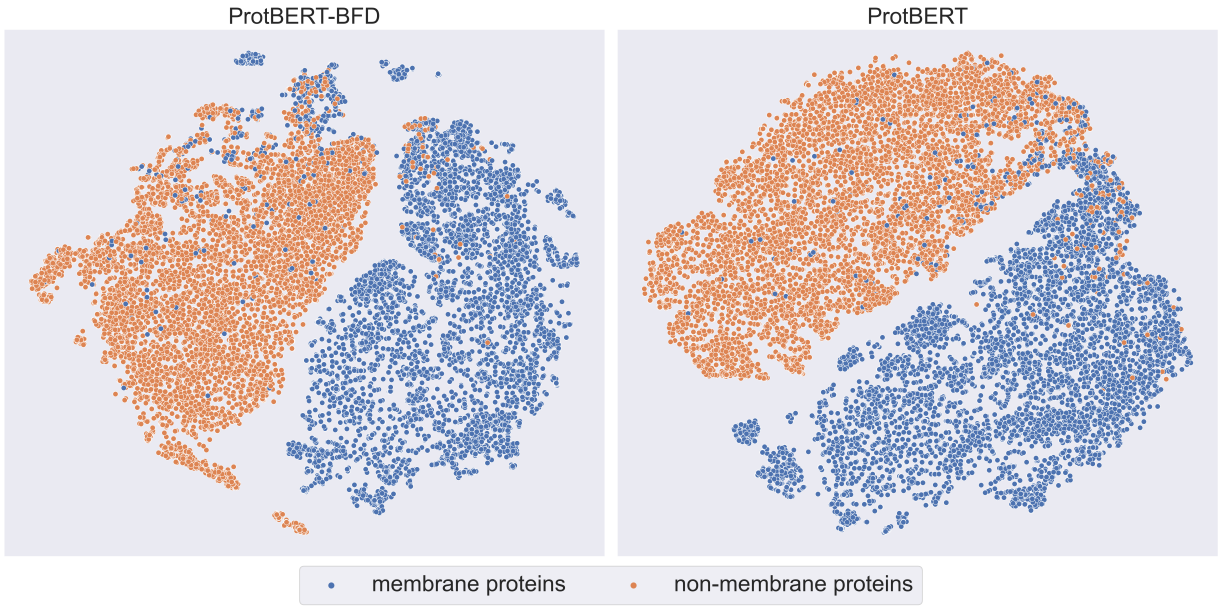

Figure 21: Membrane proteins t-SNE visualization

The t-SNE plot in this figure illustrates the ability of BERT models, specifically ProtBERT and ProtBERT-BFD, to capture important features of protein sequences that distinguish membrane proteins from non-membrane proteins. The plot shows a clear separation of the two classes, with membrane proteins plotted in blue and non-membrane proteins plotted in orange, in a lower-dimensional space. This indicates that the BERT models are able to identify fundamental differences between the two classes, such as sequence composition or structural properties, which are indicative of membrane proteins.

Table 19: Membrane proteins classification comparisons

Method	**Representation**	**Acc(%)**		**Sen(%)**		**Spc(%)**		**MCC**	
		CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.
ProtBERT	kNN	98.27 ± 1.34	92.63	98.00 ± 1.73	91.17	98.53 ± 0.96	93.94	0.96 ± 0.02	0.85
	RF	98.31 ± 1.33	93.18	97.87 ± 1.79	91.85	98.74 ± 0.89	94.27	0.96 ± 0.02	0.86
	SVM	97.72 ± 1.29	93.02	97.46 ± 1.60	92.07	97.98 ± 1.01	95.37	0.95 ± 0.02	0.86
	LR	98.17 ± 1.34	93.52	97.78 ± 1.86	91.96	98.55 ± 0.85	94.71	0.96 ± 0.02	0.87
	FFNN	97.85 ± 1.46	93.18	97.64 ± 1.77	93.09	98.19 ± 1.05	95.15	0.95 ± 0.02	0.86
	CNN	95.64 ± 2.48	94.02	93.85 ± 3.72	93.09	97.38 ± 1.46	95.15	0.91 ± 0.04	0.88
Average			93.26		92.20		94.76		0.86
ProtBERT-BFD	kNN	98.26 ± 1.38	92.01	97.93 ± 1.95	91.28	98.58 ± 0.83	93.16	0.96 ± 0.02	0.84
	RF	98.41 ± 1.46	92.12	98.08 ± 2.04	91.17	98.74 ± 0.90	93.38	0.96 ± 0.02	0.84
	SVM	98.01 ± 1.37	92.35	97.62 ± 1.94	91.28	98.38 ± 0.82	93.61	0.96 ± 0.02	0.84
	LR	98.29 ± 1.40	92.40	97.95 ± 1.95	91.17	98.62 ± 0.87	93.16	0.96 ± 0.02	0.84
	FFNN	98.03 ± 1.53	92.01	97.75 ± 2.03	91.62	98.20 ± 1.30	94.82	0.95 ± 0.03	0.84
	CNN	94.49 ± 2.75	94.02	91.12 ± 4.93	91.61	97.76 ± 0.80	96.36	0.89 ± 0.05	0.88
Average			92.48		91.36		94.08		0.84

This table presents a comparison of the performance of various classifiers and representations, grouped by PLMs, on 5-fold CV with the *mean* ± *sd* and separate test set using accuracy, sensitivity, specificity, and MCC.

4.2.3 Comparison of PLMs and Classifiers

ProtBERT generally outperformed ProtBERT-BFD in terms of accuracy, specificity, and MCC across most classifiers (Table 20). However, the CNN classifier showed consistent high performance with both representations. The CNN's superior MCC scores (0.88 for both ProtBERT and ProtBERT-BFD) indicate its reliability in membrane protein classification.

Table 20: Membrane proteins comparison of classifiers

Method	Representation	Acc(%)		Sen(%)		Spc(%)		MCC	
		CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.
kNN	ProtBERT	98.27 ± 1.34	92.63	98.00 ± 1.73	91.17	98.53 ± 0.96	93.94	0.96 ± 0.02	0.85
	ProtBERT-BFD	98.26 ± 1.38	92.01	97.93 ± 1.95	91.28	98.58 ± 0.83	93.16	0.96 ± 0.02	0.84
Average kNN			92.32		91.22		93.55		0.84
RF	ProtBERT	98.31 ± 1.33	93.18	97.87 ± 1.79	91.85	98.74 ± 0.89	94.27	0.96 ± 0.02	0.86
	ProtBERT-BFD	98.41 ± 1.46	92.12	98.08 ± 2.04	91.17	98.74 ± 0.90	93.38	0.96 ± 0.02	0.84
Average RF			92.65		91.51		93.82		0.85
SVM	ProtBERT	97.72 ± 1.29	93.02	97.46 ± 1.60	92.07	97.98 ± 1.01	95.37	0.95 ± 0.02	0.86
	ProtBERT-BFD	98.01 ± 1.37	92.35	97.62 ± 1.94	91.28	98.38 ± 0.82	93.61	0.96 ± 0.02	0.84
Average SVM			92.69		91.67		94.49		0.85
LR	ProtBERT	98.17 ± 1.34	<u>93.52</u>	97.78 ± 1.86	91.96	98.55 ± 0.85	94.71	0.96 ± 0.02	0.87
	ProtBERT-BFD	98.29 ± 1.40	92.40	97.95 ± 1.95	91.17	98.62 ± 0.87	93.16	0.96 ± 0.02	0.84
Average LR			92.96		91.56		93.94		0.85
FFNN	ProtBERT	97.85 ± 1.46	93.18	97.64 ± 1.77	**93.09**	98.19 ± 1.05	95.15	0.95 ± 0.02	0.86
	ProtBERT-BFD	98.03 ± 1.53	92.01	97.75 ± 2.03	91.62	98.20 ± 1.30	94.82	0.95 ± 0.03	0.84
Average FFNN			92.59		92.36		94.98		0.85
CNN	ProtBERT	95.64 ± 2.48	**94.02**	93.85 ± 3.72	<u>92.52</u>	97.38 ± 1.46	<u>95.47</u>	0.91 ± 0.04	**0.88**
	ProtBERT-BFD	94.49 ± 2.75	**94.02**	91.12 ± 4.93	91.61	97.76 ± 0.80	**96.36**	0.89 ± 0.05	**0.88**
Average CNN			94.02		92.06		95.91		0.88

This table presents a comprehensive comparison of the performance of various classifiers using different representations generated from ProtBERT and ProtBERT-BFD on 5-fold CV with the *mean* $\pm$ *sd* and separate test set. The performance is evaluated using four different metrics. The best performance among all classifiers and representations is highlighted in bold while the second best performance is indicated using an underline.

4.2.4 Comparison to State-of-the-Art

Our proposed method, TooT-BERT-CNN-M, outperforms previous state-of-the-art approaches in accuracy (94.02%), specificity (96.36%), and MCC (0.88) (Table 21). While TooT-M achieved the highest sensitivity (92.41%), TooT-BERT-CNN-M ranked second with 91.61%. A McNemar's test between TooT-BERT-M and TooT-BERT-CNN-M yielded a p-value of 3.15×10^{-2}, indicating a statistically significant improvement.

4.3 Results and Discussion: Transporters

We assess the output of the conventional machine learning classifiers, followed by an analysis of the outcomes of the deep neural network CNN model. The results show that TooT-BERT-CNN-T, which is the fine-tuned ProtBERT-BFD representation with CNN classifier, outperforms TooT-BERT-T [GB23d].

Table 21: Comparison of TooT-BERT-CNN-M with SOTA

Method	Acc(%)	Sen(%)	Spc(%)	MCC
iMem-2LSAAC [AHJ18]	79.27	74.52	83.90	0.59
MemType-2L [CS07]	89.44	88.67	90.19	0.79
TooT-M [AB20a]	<u>92.46</u>	**92.41**	92.50	<u>0.85</u>
TooT-BERT-M [GB22]	<u>92.46</u>	91.28	<u>93.61</u>	<u>0.85</u>
TooT-BERT-CNN-M	**94.02**	<u>91.61</u>	**96.36**	**0.88**

This table illustrates the performance of the proposed TooT-BERT-CNN-M approach in comparison to state-of-the-art methods on the task of membrane protein prediction. The performance is evaluated using various metrics including accuracy, sensitivity, specificity, and MCC. The results are presented on the separate test sets, with the highest value in each column highlighted in bold, and the second-best performance indicated by an underline.

LR and SVM surpass the other traditional classifiers Table 22 presents ProtBERT-BFD exhibits superior performance in MCC compared to alternative representations. This superior performance results from training the model on a more extensive number of sequences during pre-training. Two classifiers, LR and SVM, performed better than other traditional classifiers. The results from CNN are the best in terms of specificity, accuracy, and MCC on the separate test set and among all metrics for CV results.

Using two distinct representations of ProtBERT and ProtBERT-BFD, the performance of the SVM classifier has attained its maximum level of sensitivity, achieving a score of 100 percent on the separate test set. Also, the sensitivity of FFNN with ProtBERT representation on the test set is comparable to the one with SVM.

Regarding accuracy and MCC metrics, LR has the most outstanding values compared to other classical predictors. Moreover, the LR classifier also has the most significant values compared to other traditional classifiers within the CV results. Specifically by MembraneBERT-derived representations. This predictor's high CV values may suggest that the model has been overfitted to the training set.

Table 22: Comparison of classifiers for transporters

Classifier	Representation	CV				Independent			
		Sen	Spc	Acc	MCC	Sen	Spc	Acc	MCC
kNN	ProtBERT-BFD	97.02 ± 2.79	97.10 ± 2.78	97.06 ± 2.65	0.9405 ± 0.0537	93.33	88.33	92.20	0.8250
	ProtBERT	91.21 ± 2.37	64.25 ± 2.79	79.49 ± 1.95	0.5857 ± 0.0422	95.00	60.00	83.89	0.6265
	MembraneBERT	98.00 ± 3.54	96.79 ± 5.08	97.47 ± 4.20	0.9485 ± 0.0857	85.83	88.33	86.67	0.7172
RF	ProtBERT-BFD	95.84 ± 3.13	97.11 ± 3.04	96.38 ± 3.08	0.9276 ± 0.0619	94.17	88.33	92.22	0.8250
	ProtBERT	88.40 ± 3.38	76.91 ± 4.40	83.31 ± 2.42	0.6635 ± 0.0493	89.17	78.33	83.89	0.6750
	MembraneBERT	97.82 ± 3.68	96.88 ± 5.10	97.43 ± 4.29	0.9473 ± 0.0877	85.00	90.00	86.67	0.7073
SVM	ProtBERT-BFD	94.05 ± 2.80	86.10 ± 2.68	90.59 ± 2.50	0.7999 ± 0.0506	**100.00**	90.00	92.78	0.8369
	ProtBERT	85.69 ± 2.69	53.97 ± 2.80	71.90 ± 1.64	0.4186 ± 0.0360	**100.00**	86.67	90.00	0.7771
	MembraneBERT	97.65 ± 3.64	96.68 ± 4.81	97.23 ± 4.13	0.9439 ± 0.0838	85.00	91.67	85.00	0.6930
LR	ProtBERT-BFD	96.79 ± 3.27	97.33 ± 2.91	97.03 ± 3.05	0.9400 ± 0.0617	95.83	90.00	93.89	0.8620
	ProtBERT	90.64 ± 2.42	82.33 ± 2.95	87.03 ± 2.02	0.7358 ± 0.0410	92.50	80.00	88.33	0.7347
	MembraneBERT	98.08 ± 3.53	97.00 ± 5.18	97.61 ± 4.25	0.9513 ± 0.0866	86.67	85.00	86.11	0.6989
FFNN	ProtBERT-BFD	92.13 ± 7.08	91.79 ± 6.98	91.79 ± 6.98	0.7924 ± 0.0586	92.50	90.00	90.00	0.8043
	ProtBERT	85.95 ± 6.79	78.44 ± 7.51	82.37 ± 2.29	0.6480 ± 0.0402	**100.00**	50.00	87.22	0.7414
	MembraneBERT	95.37 ± 5.49	94.60 ± 6.73	95.43 ± 4.74	0.9073 ± 0.0936	60.00	28.33	85.00	0.6832
CNN	ProtBERT-BFD	85.64 ± 7.25	95.33 ± 3.85	89.85 ± 3.57	0.8072 ± 0.0642	95.00	**95.00**	**95.00**	**0.8894**
	ProtBERT	95.00 ± 3.58	81.16 ± 1.47	88.98 ± 4.95	0.7855 ± 0.0943	95.00	90.00	93.33	0.8500
	MembraneBERT	98.71 ± 0.90	97.83 ± 1.25	98.33 ± 0.71	0.9662 ± 0.0157	90.83	91.66	91.11	0.8070

This table illustrates the results from three different BERT-based protein representation, namely ProtBERT, ProtBERT-BFD and MembraneBERT with various classifiers for the task of transporters classification. Cross-validation (CV) and separate test set results are presented for each representation and classifier. The maximum value for each column is displayed in boldface.

As shown in Table 22, the best results on the separate test set for specificity, accuracy, and MCC were achieved by CNN. Also, the CV results from MembraneBERT using the CNN classifier are the highest values among other classifiers. However, it is noteworthy that MembraneBERT with CNN classifier exhibits significant variability between its CV and separate test set performance. This discrepancy could be attributed to the model's complexity and high capacity may lead to overfitting on the training data, resulting in optimistic CV estimates but poorer generalization to the separate test set.

CNN outperforms traditional models: transporters Table 23 and Figure 22 compares the proposed method, TooT-BERT-CNN-T, with previous methods on the separate test set. As can be observed, the performance of TooT-BERT-CNN-T is superior to that of TooT-BERT-T, TooT-T, on three metrics: specificity, accuracy, and MCC. In comparison, the performance of TooT-BERT-T in terms of the sensitivity measure is superior to that of other classifiers.

The performance of the proposed method can be attributed to the ProtBERT-BFD larger pre-training dataset. Furthermore, the convolutional filters in CNN's layers scan the entire feature matrix and perform dimensionality reduction, allowing CNN to perform well in this task. Notably, the separate test set metrics, including MCC, accuracy, and sensitivity, were observed to be higher than the CV results. This discrepancy could be due to several factors. Firstly, the separate test set may have inadvertently been more representative of the patterns learned by the model during training, leading to better performance. Secondly, CV results reflect the model's average performance across different subsets of the data, which may include more challenging or diverse examples, potentially lowering the overall scores. Additionally, the separate test set might have been smaller or less diverse than the training set, resulting in less variability and thus higher metrics. It is also possible that some degree of data leakage or selection bias occurred during the creation of the separate test set, inadvertently favoring examples that align well with the model's learned patterns.

Table 23: Comparing classifiers on test set for transporters

Classifier	Representer	Sen	Spc	Acc	MCC
TooT-T [AB20b]	Traditional*	94.17	88.33	92.22	0.8200
TooT-BERT-T [GB23d]	ProtBERT-BFD	**95.83**	90.00	93.89	0.8620
TooT-BERT-CNN-T	ProtBERT-BFD	95.00	**95.00**	**95.00**	**0.8894**

TooT-BERT-CNN-T is compared with other classifiers as well as TooT-BERT-T and TooT-T on four evaluation measurements. The maximum value for each column is displayed in boldface. * An ensemble approach of traditional vector representations such as Amino Acid Composition (AAC) and Dipeptide Composition (DPC) [AB20b].

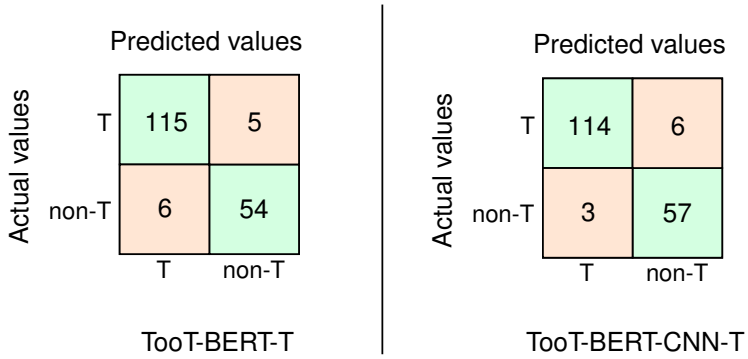

Figure 22: TooT-BERT-CNN-T and TooT-BERT-T Confusion Matrices

Confusion matrices of TooT-BERT-T and TooT-BERT-CNN-T to discriminate transporters (T) from non-transporters (non-T).

4.4 Results and Discussion: Ion Channel

4.4.1 Representation Analysis

To visualize the feature representations from ProtBERT, ProtBERT-BFD, and MembraneBERT for ion channels and non-ion channels, we employ t-SNE, or t-Distributed Stochastic Neighbor Embedding [VdMH08]. This technique serves as a powerful tool for reducing dimensionality while maintaining the relationships among high-dimensional data points. This approach is particularly useful for capturing intricate, non-linear relationships, making it widely used in areas like machine learning and data visualization.

The t-SNE plot, as shown in Figure 23, highlights the proficiency of ProtBERT, ProtBERT-BFD, and MembraneBERT in differentiating important features of ion channels from those of non-ion channels. In the plot, ion channels are marked in blue and non-ion channels in orange. The

separation between the two categories in this two-dimensional representation suggests that the models effectively capture essential distinctions between the two groups. These distinctions may encompass variations in sequence composition or structural features that serve as hallmarks for ion channel proteins.

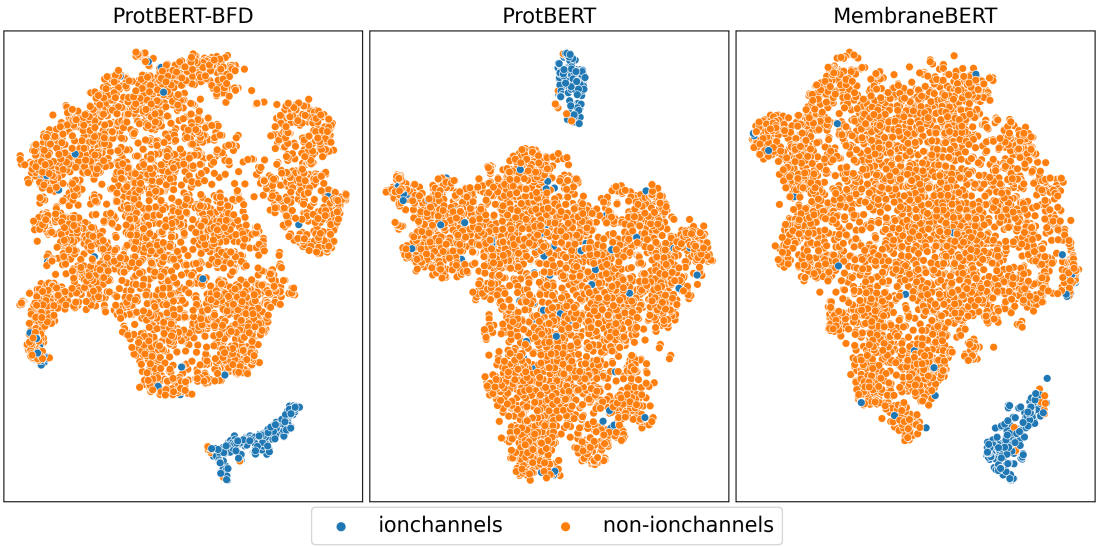

Figure 23: t-SNE plot of representations for ion channel

A t-SNE plot is shown in this figure, illustrating the two-dimensional representations of the sequences obtained from ProtBERT, ProtBERT-BFD, and MembraneBERT for the ion channel and non-ion channel classes in the dataset. The ion channel sequences are plotted in blue, while the non-ion channel sequences are plotted in orange.

4.4.2 Comparative Analysis of Classifier and PLM: Ion channel

This section discusses the performance of various PLMs and classifiers, elucidating their effectiveness in protein classification tasks. The evaluation metrics include accuracy, sensitivity, specificity, and the Matthews Correlation Coefficient (MCC). The results are derived from both CV and independent tests to offer a understanding of each model's capabilities and limitations. Detailed performance statistics can be found in Table 24 and visual representations in Figure 24. The discussion is categorized into overall performance, performance per classifier, inconsistencies in sensitivity, most stable classifiers, and individual best performances.

Overall Performance: In the comprehensive analysis of performance metrics across different PLMs and classifiers, ProtBERT-BFD emerges as a notably strong performer, particularly in the sensitivity metric. With an average sensitivity of 75.84% on the separate test sets, ProtBERT-BFD consistently demonstrates superior ability in identifying true positive cases. This is a crucial aspect in bioinformatics applications where missing a potential hit could have significant implications. In contrast, while MembraneBERT exhibits remarkable sensitivity in CV tests, its performance is somewhat diminished in the separate test sets. This discrepancy suggests that although MembraneBERT is highly sensitive to the dataset it is trained on, it may lack the generalizability exhibited by ProtBERT-BFD when subjected to unseen data.

Classifier-wise Performance: In evaluating the performance of individual classifiers across different PLMs, certain trends become evident. Notably, the kNN algorithm manifests exceptional performance when paired with ProtBERT and MembraneBERT, especially in terms of sensitivity. This suggests that kNN's instance-based learning approach synergizes well with the feature representations generated by these PLMs, particularly in identifying true positive cases. On the other hand, CNN employed with ProtBERT-BFD yield the highest MCC of 0.8584 in the separate

test sets. The high MCC value is indicative of a well-balanced performance across various classes, underscoring the classifier's ability to perform consistently in binary classification scenarios.

Table 24: Comparison of classifiers and representations: ion channel

PLM	Method	Acc(%)		Sen(%)		Spc(%)		MCC	
		CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.
ProtBERT	kNN	97.80 ± 0.23	97.69	71.61 ± 3.75	70.00	99.65 ± 0.25	99.53	0.8086 ± 0.0213	0.7972
	RF	97.30 ± 0.22	97.58	60.69 ± 3.14	63.33	99.88 ± 0.15	**100.00**	0.7557 ± 0.0234	0.7749
	SVM	95.22 ± 0.27	<u>98.24</u>	37.63 ± 4.23	<u>75.00</u>	99.29 ± 0.23	**100.00**	0.4469 ± 0.0358	<u>0.8483</u>
	LR	97.57 ± 0.25	97.80	67.13 ± 3.77	68.33	99.72 ± 0.25	**100.00**	0.7852 ± 0.0234	0.8068
	FFNN	97.13 ± 0.54	98.02	67.04 ± 4.23	73.33	98.69 ± 1.42	**100.00**	0.7126 ± 0.0299	0.7848
	CNN	99.09 ± 0.82	97.80	88.90 ± 3.82	66.66	99.82 ± 0.21	**100.00**	0.9235 ± 0.0728	0.8070
Average ProtBERT			97.86		69.44		99.92		0.8032
ProtBERT-BFD	kNN	98.97 ± 0.31	97.47	88.19 ± 5.12	<u>75.00</u>	99.73 ± 0.10	98.71	0.9137 ± 0.0275	0.7848
	RF	99.03 ± 0.45	97.47	87.97 ± 5.85	**76.67**	99.80 ± 0.12	98.82	0.9187 ± 0.0390	0.7767
	SVM	98.39 ± 0.40	97.69	79.87 ± 5.49	<u>75.00</u>	99.69 ± 0.13	**100.00**	0.8450 ± 0.0353	0.8016
	LR	98.96 ± 0.43	<u>98.24</u>	86.71 ± 5.55	**76.67**	99.82 ± 0.11	<u>99.76</u>	0.9123 ± 0.0375	<u>0.8486</u>
	FFNN	98.34 ± 0.92	97.03	82.08 ± 7.42	**76.67**	99.38 ± 0.78	**100.00**	0.8557 ± 0.0584	0.8287
	CNN	99.39 ± 0.20	**98.35**	93.38 ± 2.96	<u>75.00</u>	99.82 ± 0.14	**100.00**	0.9506 ± 0.0167	**0.8584**
Average ProtBERT-BFD			97.71		75.84		99.55		0.8165
MembraneBERT	kNN	99.59 ± 0.34	96.92	96.48 ± 3.03	66.67	99.81 ± 0.18	98.82	0.9665 ± 0.0278	0.7358
	RF	99.59 ± 0.33	97.03	96.32 ± 2.84	68.33	99.82 ± 0.17	98.94	0.9670 ± 0.0274	0.7495
	SVM	99.29 ± 0.32	97.03	91.86 ± 3.13	68.33	99.81 ± 0.15	**100.00**	0.9397 ± 0.0264	0.7383
	LR	99.50 ± 0.33	97.14	95.53 ± 2.88	66.67	99.78 ± 0.17	99.29	0.9590 ± 0.0275	0.7472
	FFNN	99.18 ± 0.44	97.25	91.07 ± 4.84	68.33	99.77 ± 0.20	**100.00**	0.9159 ± 0.0491	0.7383
	CNN	97.86 ± 1.57	97.91	75.55 ± 4.55	68.33	99.44 ± 1.04	**100.00**	0.8203 ± 0.1267	0.8175
Average MembraneBERT			97.21		67.78		99.51		0.7544

The table presents a comparison of the performance of different classical and deep learning classifiers and representations, as generated from ProtBERT, ProtBERT-BFD, and MembraneBERT on CV and separate test sets for the task of ion channel prediction. The results are evaluated using various metrics, with the largest value in each column on independent test results indicated in boldface for comparison between different classifiers. The second best result in each column is indicated with an underline for further analysis and comparison.

Inconsistencies in sensitivity: The table reveals significant inconsistencies in sensitivity between the CV and separate test set, most markedly for MembraneBERT. While MembraneBERT exhibits high sensitivity scores in the CV sets—ranging from 75.55% to 96.48%—these figures drop substantially in the independent tests, with scores falling between 66.67% and 68.33%. This sharp decline could signify a couple of issues: potential overfitting of the model to the training data or a limited ability to generalize well to new, unseen data.

Most Stable Classifier: In evaluating of various classifiers across different PLMs like ProtBERT, ProtBERT-BFD, and MembraneBERT, the kNN method consistently stands out. The kNN classifier exhibits less variance in CV metrics, as evidenced by the small standard deviations across the metrics of accuracy, sensitivity, specificity, and MCC. This minimal fluctuation in performance demonstrates that kNN is robust to the nuances of different PLMs, making it a stable choice for protein classification tasks.

Best Individual Performance: In the landscape of classifiers and Protein Language Models, the combination of ProtBERT with the SVM method emerges as notably effective. It achieves an exceptional 98.24% accuracy in independent tests, underlined in the table, which surpasses most other configurations. Additionally, it exhibits a sensitivity of 75.00%, making it second-best in this metric. Notably, it also secures the highest MCC of 0.8483 among independent tests. This exceptional performance on multiple fronts indicates that the ProtBERT-SVM pairing could be a highly promising configuration that warrants further investigation for protein classification tasks.

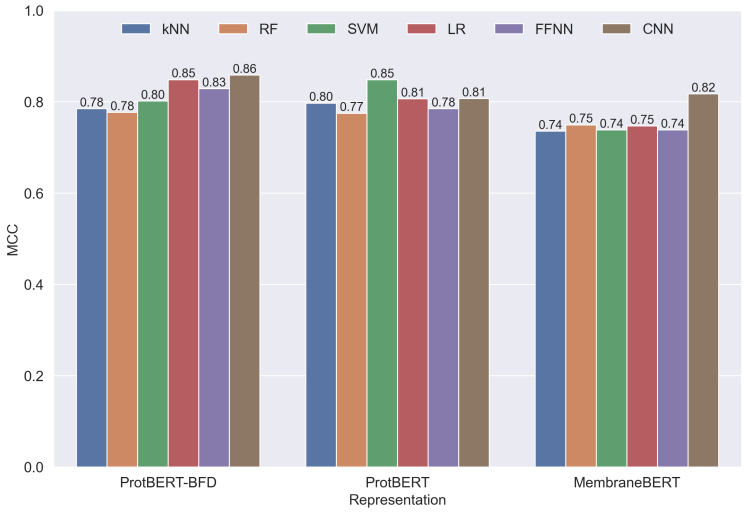

Figure 24: Comparison of classifiers for ion channel

The performance of different classical and deep learning classifiers and representations generated from ProtBERT, ProtBERT-BFD and MembraneBERT is compared on separate test sets using the MCC metric for ion channel classification.

4.4.3 Comparison to State-of-the-Art

Table 25 compares the performance of *TooT-BERT-CNN-C* with three established methodologies: DeepIon [TO19], MFPS_CNN [NHTO22], and TooT-BERT-C [GB23c]. Table 25 showcases the metric values for the independent set alongside the mean values derived from the folds in the cross-validation set.

Table 25: Comparative performance of TooT-BERT-CNN-C

Method	**Acc(%)**		**Sen(%)**		**Spc(%)**		**MCC**	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
DeepIon [TO19]	86.53	87.05	68.33	89.20	87.72	84.89	0.37	0.75
MFPS_CNN [NHTO22]	94.60	96.50	**76.70**	**95.00**	95.80	98.00	0.62	0.93
TooT-BERT-C [GB23c]	98.24	98.96	76.67	86.71	99.76	**99.82**	0.85	0.91
TooT-BERT-CNN-C	**98.35**	**99.39**	75.00	93.38	**100.00**	**99.82**	**0.86**	**0.95**

This table compares the performance of TooT-BERT-CNN-C with state-of-the-art approaches on cross-validated and separate test sets using evaluation metrics such as sensitivity, specificity, accuracy, and MCC. The maximum value in each column is highlighted in boldface.

The proposed method, *TooT-BERT-CNN-C*, consistently outperforms earlier approaches in a variety of performance metrics. Specifically, on the separate test set, it scores the highest in nearly all evaluation categories, save for sensitivity, where MFPS_CNN slightly excels. In the cross-validation set, *TooT-BERT-CNN-C* tops the charts for accuracy, specificity, and MCC. The model's strong showing in the independent set underscores its generalization capabilities, indicating its reliability for accurately predicting ion channels in new, unseen data.

Figure 25 presents confusion matrices for TooT-BERT-C and TooT-BERT-CNN-C. For TooT-BERT-C, the matrix reveals 46 TP, 848 TN, 2 FP, and 14 FN predictions. Conversely, the TooT-BERT-CNN-C matrix demonstrates 45 TP, 850 TN, 0 FP, and 15 FN predictions.

While empirical analysis reveals that *TooT-BERT-CNN-C* surpasses *TooT-BERT-C* in performance, a McNemar's test yields a p-value of 0.0625, which shows the improvement is not statistically significant.

Sequence Length Constraint One of the primary challenges faced was the constraint on sequence length. Due to computational limitations, we truncated protein sequences to a maximum

of 1024 amino acids. While most sequences in our dataset were within this limit, this truncation might have led to the loss of potentially crucial information in longer sequences.

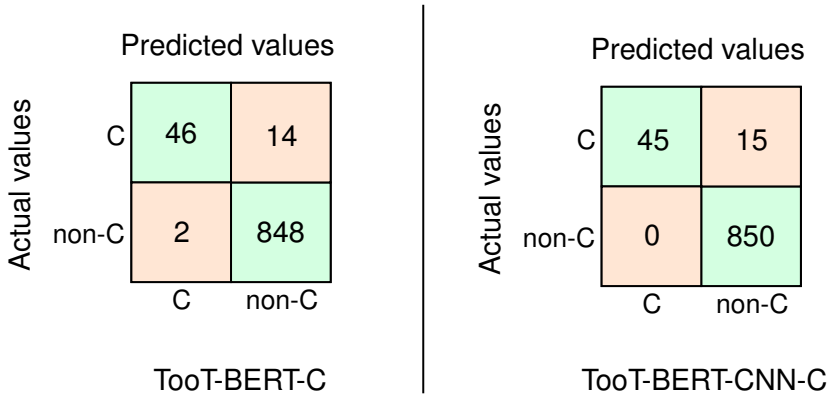

Figure 25: Confusion matrices for TooT-BERT-C and TooT-BERT-CNN-C.

This figure presents confusion matrices for two approaches, TooT-BERT-C and TooT-BERT-CNN-C, used in the task of ion channel prediction.

Computational Resource Demand The integration of CNN with PLMs, while beneficial in terms of classification performance, significantly increased computational resource demands. Training times were longer, and the process required more advanced hardware (e.g., Tesla V100 GPU). This could pose a barrier when attempting to replicate or extend this research on less powerful systems.

4.5 Conclusions

This chapter has provided an exploration of the application of a deep learning technique, particularly CNN, combined with three PLMs, for the classification of membrane proteins, transporters and ion channels.

In our experiments, the CNN classifier consistently outperformed traditional machine learning classifiers across all three classification tasks when using ProtBERT-BFD representations. For membrane protein classification, CNN achieved an accuracy of 94.02% and MCC of 0.88 on the separate test set, compared to the next best performer, Logistic Regression, which achieved 92.40% accuracy and 0.84 MCC. For transporter classification, CNN achieved 95.00% accuracy and 0.8894 MCC, surpassing Logistic Regression's 93.89% accuracy and 0.8620 MCC. For ion channel classification, CNN achieved 98.35% accuracy and 0.8584 MCC, outperforming SVM's 97.69% accuracy and 0.8016 MCC.

Our proposed method, TooT-BERT-CNN, which combines CNN with ProtBERT-BFD, outperformed state-of-the-art methods in all three classification tasks: For membrane proteins, TooT-BERT-CNN-M achieved 94.02% accuracy, 96.36% specificity, and 0.88 MCC, surpassing the previous best method, TooT-M (92.46% accuracy, 92.50% specificity, 0.85 MCC). For transporters, TooT-BERT-CNN-T achieved 95.00% accuracy, 95.00% specificity, and 0.8894 MCC, outperforming TooT-T (92.22% accuracy, 88.33% specificity, 0.8200 MCC). For ion channels, TooT-BERT-CNN-C achieved 98.35% accuracy, 100.00% specificity, and 0.86 MCC, surpassing MFPS_CNN (94.60% accuracy, 95.80% specificity, 0.62 MCC).

In our analysis of classifiers and Protein Language Models, several combinations demonstrated promising results across different protein classification tasks. For membrane proteins, the ProtBERT-CNN combination achieved the highest accuracy (94.02%) and MCC (0.88) on the separate test set. The ProtBERT-SVM pairing also performed exceptionally well for ion channel prediction, with 98.24% accuracy and an MCC of 0.8483.

Despite these strong individual performances, we ultimately selected the ProtBERT-BFD-CNN approach for our final tool in membrane protein, transporter, and ion channel prediction. This

decision was based on several key factors. For membrane protein prediction, ProtBERT-BFD-CNN matched the top accuracy (94.02%) and MCC (0.88) of ProtBERT-CNN, while potentially offering more robust representations due to its larger pre-training dataset. In transporter prediction, ProtBERT-BFD-CNN achieved the highest overall accuracy (95.00%) and MCC (0.8894) on the separate test set. For ion channel prediction, it showed the best accuracy (98.35%) and MCC (0.8584) among all methods tested.

The ProtBERT-BFD model, pre-trained on a larger dataset, offers potentially more robust and generalizable representations across all three tasks. While other combinations like ProtBERT-SVM showed high sensitivity in some cases, the CNN's balanced performance across all metrics suggested it may be more reliable across diverse datasets and use cases. This consistency across different protein classification tasks made ProtBERT-BFD-CNN a versatile choice for our unified prediction tool.

Our methodology employed a fixed-length sequence representation by truncating protein sequences to a maximum of 1024 amino acids. Despite this limitation, which potentially loses information from longer sequences, our approach achieved high performance across all three classification tasks. For example, in membrane protein classification, only 0.39% of sequences in the DS-M dataset exceeded this length limit, suggesting minimal information loss for the majority of sequences. This demonstrates that fixed-length representations can be effective for protein sequence classification, at least for the datasets and tasks examined in this study. The truncation of sequences to a manageable length for computational processing, represents a practical approach in computational bioinformatics.

Our research demonstrates the effectiveness of combining CNN with ProtBERT-BFD for classifying membrane proteins, transporters, and ion channels, achieving state-of-the-art performance in all three tasks. However, this study is limited to these specific protein classification tasks and the three PLMs we examined (ProtBERT, ProtBERT-BFD, and MembraneBERT). Future research should investigate the applicability of this approach to other protein classification tasks and explore its performance with different PLMs to establish the generalizability of our findings.

Chapter 5

Exploiting Protein Language Models for the Precise Classification of Ion Channels and Ion Transporters

This chapter undertakes an in-depth evaluation of six PLMs in conjunction with six distinct classifiers, aiming to accurately distinguish between ion channels, ion transporters, and other integral components of membrane proteins. The investigation examines key determinants of PLM efficacy in protein classification tasks, including the effects of dataset balance, nuances in model representation tuning, and the implications of computational precision in floating-point operations.

Our research has been broadened to encompass the evaluation of newly annotated data pertaining to ion channels and ion transporters. Through the incorporation of recent updates from the UniProtKB/Swiss-Prot database, we endeavor to reaffirm the scalability and generalizability of our proposed models.

5.1 Introduction

We embark on an investigation of three pivotal factors that could significantly influence the performance of PLMs in our tasks:

- The choice between using frozen or fine-tuned PLM representations.
- The influence of balanced versus imbalanced datasets on model performance.
- The implications of half-precision versus full-precision floating-point computations.

To understand the context and importance of our study, it is crucial to first examine the biological systems we aim to model. Based on the current understanding in cell biology, ion regulation across cell membranes is a fundamental process for cellular function. This process involves two main types of integral membrane proteins: ion channels (ICs) and ion transporters (ITs), which facilitate the controlled movement of ions across cellular membranes [TO19]. These MPs play pivotal roles in maintaining ion homeostasis, regulating transmembrane potential, and facilitating electrical signaling. Such functions are essential for various cellular processes, including proliferation (cell division and multiplication), migration (cells move from one location to another within an organism), and differentiation (the process by which a less specialized cell becomes a more specialized cell type) [Hil01, NC19, RADVRC10].

5.1.1 Organization

This chapter is organized as follows: Section 5.2 delineates our methodological framework, highlighting the creation and utilization of a new dataset tailored for this study, alongside the methodology for balancing the membrane proteins dataset. This section also provides an in-depth

overview of the six PLMs employed, with a particular emphasis on the ESM-1b model due to its notable performance. We delve into the architecture and training nuances of ESM-1b to elucidate its effectiveness in our tasks. Furthermore, the section outlines the classifiers used, discusses hyperparameter optimization strategies, and describes the evaluation metrics implemented to gauge model efficacy.

In Section 5.3, we analyze and interpret the outcomes of our experimental investigations. This section not only assesses the individual and collective performance of the PLMs and classifiers across the specified tasks but also examines the influence of dataset balancing, representation tuning, and computational precision on our results. Special attention is given to the application of our newly developed dataset on the TooT-PLM-ionCT system, highlighting its impact on model generalizability and performance. Comparative analyses with existing state-of-the-art approaches are also presented, providing a context for our findings.

Section 5.4 presents the key contributions of our study, distilling the essential insights derived from our research. It also proposes directions for future inquiry, pinpointing specific areas within the domain of protein classification using PLMs that warrant further exploration.

Lastly, we direct the reader's attention to Appendix B, which contains supplementary tables and figures that provide additional detail and context for the results discussed in this chapter.

5.1.2 Frozen vs Fine-tuned Representations

The concept of frozen and fine-tuned representations pertains to the degree of adaptation of pre-trained language models to a specific task. Frozen representations refer to the utilization of pre-trained models in their original state, without any further task-specific training. On the other hand, fine-tuned representations involve the additional step of task-specific training, where the pre-existing parameters of the pre-trained models are adjusted to enhance their performance on the given task.

Our research includes a comparative study of frozen and fine-tuned versions of a PLM to evaluate the impact of task-specific adaptation on membrane protein classification performance. It allows us to understand the inherent behavior of the original pre-trained models (as reflected in the frozen state) and to quantify the extent of improvement achievable through task-specific fine-tuning. This comparison can potentially expose the limitations of the pre-training process and highlight the areas where fine-tuning can yield significant benefits.

It is important to note that fine-tuning necessitates additional computational resources compared to the use of frozen models. Consequently, if the performance enhancement achieved through fine-tuning is marginal or negligible for a specific task, it might be more resource-efficient to employ the model in its frozen state.

5.1.3 Balanced vs Imbalanced Datasets

The terms "balanced" and "imbalanced" in machine learning refer to the distribution of classes within a dataset. A balanced dataset exhibits approximately equal representation of all classes, while an imbalanced dataset is characterized by unequal representation of classes. In the context of this study, these terms are used to describe the distribution of membrane protein sequences in the DS-C dataset (Table 28).

Imbalanced datasets, where certain classes are underrepresented, can significantly impact the performance of a machine learning model. The model may develop a bias towards the majority class, leading to suboptimal performance when predicting the minority class.

For instance, if the dataset contains a significantly larger number of MPs compared to ICs or ITs, the model may develop a bias towards MPs. This bias could compromise the model's ability to accurately predict ICs or ITs, underscoring the importance of considering the balance of classes in the dataset.

5.1.4 Half vs Full Precision Floating Points Calculations

Half and full precision floating-point representations pertain to the level of numerical precision employed in model computations. Full precision, typically realized through 32-bit floats, provides superior numerical accuracy. Conversely, half precision, utilizing 16-bit floats, curtails memory usage and computational demands, albeit at the expense of a reduction in numerical accuracy.

The use of half-precision computations can expedite the training process, but it may also influence model performance due to the diminished numerical precision. It is crucial to evaluate whether this reduction in precision significantly affects the model's capacity to learn and generalize effectively.

5.2 Materials and Methods

5.2.1 Methodology Overview

We studied six Protein Language Models (PLMs), including ProtBERT, ProtBERT-BFD, and ProtT5 from the ProtTrans project [EHD+21], as well as ESM-1b, ESM-2, and ESM-2_15B from the ESM project [RMS+21,LAR+23], and six classifiers, Logistic Regression (LR), k-Nearest Neighbor (kNN), Support Vector Machine (SVM), and Random Forest (RF), to more complex models like Feed-Forward Neural Network (FFNN) and Convolutional Neural Network (CNN), across three key tasks: distinguishing ion channels (ICs) from other membrane proteins (MPs), differentiating ion transporters (ITs) from other MPs, and discriminating ICs from ITs. To enhance the robustness and applicability of our findings, we have introduced a newly curated dataset alongside the existing one for a more rigorous validation of the TooT-PLM-ionCT system.

Table 26: Overview of Research Methodology.

Methodology Component	**Details**
Protein Language Models	ProtBERT, ProtBERT-BFD, ProtT5, ESM-1b, ESM-2, ESM-2_15B
New Dataset	TooT-PLM-ionCT dataset from UniProtKB
Tasks	Discrimination of ion channels vs other membrane proteins, ion transporters vs other membrane proteins, ion channels vs ion transporters
Classifiers	SVM, Logistic Regression (LR), Random Forest (RF), kNN, Feed-forward Neural Network (FFNN), CNN
Hyperparameter Optimization	Grid search using scikit-learn (for SVM, LR, RF, kNN, FFNN) and Optuna (for CNN)
Cross-Validation Technique	5-fold cross-validation
Evaluation Metrics	accuracy, MCC, sensitivity, specificity
Statistical Significance Analysis	Paired Student t-test, ANOVA
Impacts Evaluated	1) Frozen vs fine-tuned representations from PLMs, 2) Balanced vs imbalanced datasets (Downsampling of MPs dataset), 3) Half vs full precision floating-point calculations
Presentation of Results	Results include comparative analyses across dataset balance, classifier type, PLM, representation type (frozen or fine-tuned), precision type (half or full), with UMAP projections for each PLM, task, and representation type
Optimal Configuration	Identification of the best configuration for each task, with separate test set evaluation, and comparison with state-of-the-art methodologies
Limitations	Constraints in fine-tuning large PLMs like ProtT5 and ESM-2_15B due to resource limitations (GPUs, memory), leading to incomplete data in some instances.

This table encapsulates the various components of the research methodology employed in this study, including the introduction of a new dataset for enhanced validation of the TooT-PLM-ionCT system.

Our study investigates several factors potentially influencing task outcomes, including the effects of dataset balancing, the frozen versus fine-tuned PLM representations, and half versus full precision floating-point representations. Table 26 presents a summary of the research

methodology employed.

5.2.2 Dataset

Our study employed the DS-C dataset from the DeepIon [TO19] and MFPS_CNN [NHTO22] projects, sourced from the UniProt database [ABW+04]. In line with established bioinformatics protocols [WP99, AMS+97, LG06] aimed at enhancing dataset diversity and representativeness, we adopted the strategy of filtering out sequences exhibiting more than 20% similarity using the BLAST algorithm [AMS+97].

Table 28: DS-C, the ion channel and ion transporter dataset.

Class	Training	Test	Total
Ion channel (IC)	241	60	301
Ion transporter (IT)	281	70	351
Other membrane protein (MP)	3,413	850	4,263
Total	3,935	980	4,915

Distribution of sequences in the training and test sets. This dataset has been curated by Taju et al. in DeepIon project [TO19] in April 14, 2018.

This threshold is grounded in the widely accepted practice to mitigate dataset redundancy, thereby preventing model overfitting and enhancing the generalizability of predictive models [Sö05, RSS01]. The choice of a 20% similarity threshold, as applied by Taju et al. [TO19], is informed by the consensus in the field that higher levels of sequence similarity can lead to biased training and an overestimation of model performance [HSSS92, RSS01]. The DS-C dataset, thus, consisted of 4915 protein sequences, segmented into 301 ion channels, 351 ion transporters, and 4263 other membrane proteins. This dataset was subsequently divided into training and testing sets to rigorously evaluate the model's predictive capabilities across diverse protein sequences, as detailed in Table 28.

Table 29: Query parameters for DS-Cv2 collection from UniProtKB/Swiss-Prot.

Parameter	IC	IT	MP
reviewed	true	true	true
keyword	KW-0407	NOT KW-0407 AND KW-0406	NOT KW-0407 NOT KW-0406
existence	1	1	1
cc_scl_term	SL-0162	SL-0162	SL-0162
active	true	true	true
precursor	false	false	false
fragment	false	false	false
GO terms	-	-	NOT GO:0006811 NOT GO:0022857

IC, IT, and MP stand for Ion Channels, Ion Transporters, and Other Membrane Proteins, respectively. The 'reviewed' parameter ensures only manually reviewed entries are included. 'Keyword' filters use KW-0407 for ion channels and KW-0406 for ion transporters, with 'NOT' indicating exclusion. 'Existence:1' confirms the protein's existence is experimentally verified. 'cc_scl_term:SL-0162' specifies proteins located in membrane regions. The 'active' status ensures only current entries are considered. 'Precursor:false' and 'fragment:false' exclude precursor and fragmented proteins, respectively. GO terms are used to refine the search further, excluding specific ion transport functions for MPs.

To further validate the TooT-PLM-ionCT system and to incorporate recent annotations, we created a new dataset DS-Cv2 on February 11, 2024, employing the same query parameters as utilized in the DeepIon [TO19] project, from UniProtKB/Swiss-Prot [ABW+04] entries. Outlined

in Table 29, this approach ensured consistency with previous studies and allowed for a direct comparison of the results.

We employed CD-HIT [LG06] to mitigate redundancy by removing sequences exhibiting more than 20% similarity. This yielded a dataset comprising 525 ion channels, 977 ion transporters, and 11,130 other membrane proteins. Consistent with the practices of Deeplon [TO19], the dataset was partitioned into training and test sets adhering to an 80:20 ratio. The distribution of these sequences is detailed in Table 30.

Table 30: updated dataset DS-Cv2

Class	Training	Test	Total
Ion channel (IC)	420	105	525
Ion transporter (IT)	781	196	977
Other membrane protein (MP)	8,904	2,226	11,130
Total	10,105	2,527	12,632

This table displays the distribution of sequences in the newly curated datasets, separated into the training and test sets.

Dataset Balance Table 28 highlights the imbalance of membrane proteins in dataset DS-C relative to those of ion channels and ion transporters. Figure 26 visualizes this imbalance and shows the impact of downsampling the training set of membrane proteins to match the number of IT proteins. To tackle this imbalance and evaluate the efficacy of PLMs and classifiers under varied conditions, we worked with both imbalanced and balanced versions of this dataset.

Balancing involved the selection of 280 sequences from the membrane protein training set. Using distinct random seeds, we created 10 balanced datasets for intermediate evaluations. The full datasets DS-C and DS-Cv2 were reserved for final assessment.

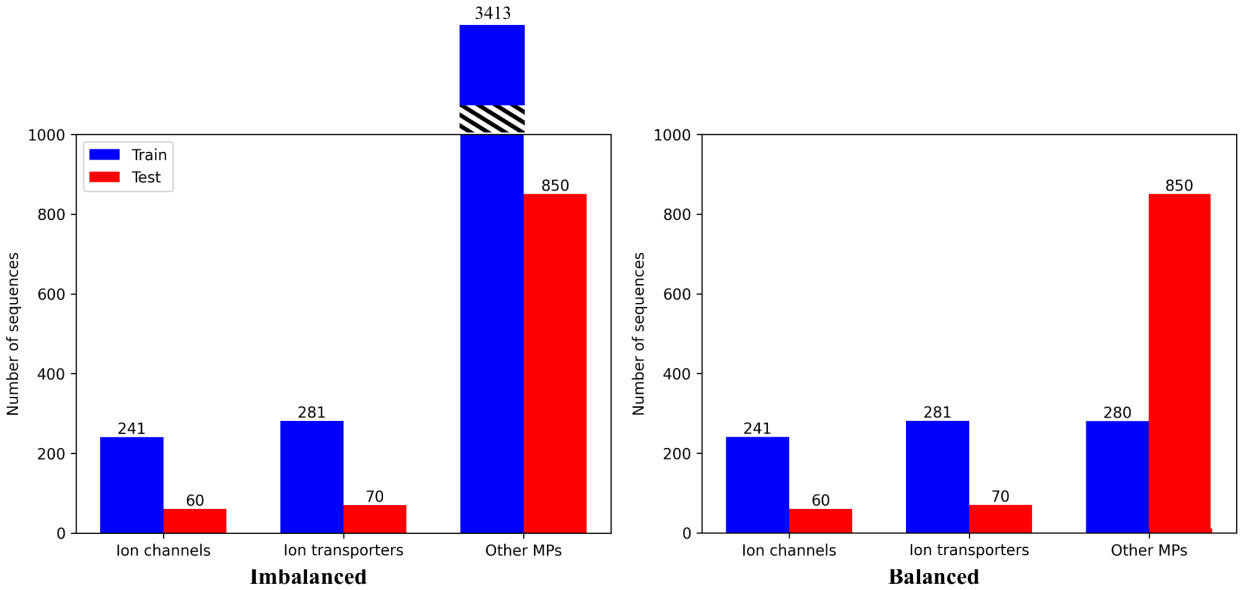

Figure 26: Visualization of membrane protein dataset balancing.

This figure presents the distribution of sequences in the dataset from Deeplon, delineated as bar plots. The training set sequences are represented by the blue bars, whereas the red bars depict the sequences in the separate test set. The left-hand figure portrays the distribution within the imbalanced dataset of additional membrane proteins (MPs). Conversely, the right-hand figure exhibits the balanced dataset, which was achieved through undersampling of MPs in the training set.

5.2.3 Protein Language Models (PLMs)

Table 31 presents the six PLMs used in this study. They are detailed in Section 2.3.

Frozen representations are feature vectors from the final layer of the PLMs, employing mean-pooling to generate a unique representation for each protein sequence. This process is consistent with the methodologies adopted in ProtTrans [EHD+21] and ESM [RMS+21, LAR+23].

For fine-tuning of the PLMs, we engage the Trainer API from the transformers library [WDS+20]. We primarily utilize the library's default hyperparameters but modify the number of epochs to 5, following the guidelines of the original BERT paper [DCLT19]. To mitigate memory constraints, we adopt a batch size of 1.

Table 31: Implementation details for PLMs.

	ProtBERT	ProtBERT-BFD	ProtT5	ESM-1b	ESM-2	ESM-2_15
Parameters	420M	420M	3B	650M	650M	15B
Dataset	UniRef100	BFD	BFD	UniRef50	UniRef50	UniRef50
Sequences	216M	2.1B	2.1B	27M	27M	27M
Embedding dim	1024	1024	1024	1280	1280	5120
Layers	30	30	24	33	33	48

This table presents the detail of PLMs used in this study, ProtBERT [EHD+21], ProtBERT-BFD [EHD+21], ProtT5 [EHD+21], ESM-1b [RMS+21], ESM-2 [LAR+23], ESM-2_15B [LAR+23].

5.2.4 Hyperparameter Optimization

In this investigation, hyperparameter optimization was performed using scikit-learn's grid search [Kra16] and Optuna [ASY+19]. Optuna is an advanced Python library specifically designed for hyperparameter optimization. The random seed was set to 32 for all experiments.

The specific ranges and values for hyperparameters were chosen based on a combination of empirical evidence from previous studies [AB20a, AB20b, GB23a], preliminary experiments, and theoretical understanding of each classifier. For instance, the cost parameter (C) in SVM was selected to cover a broad range of potential regularization strengths, acknowledging that both underfitting and overfitting are critical considerations in model performance. Similarly, the choice of kernel types was intended to explore various data transformations to understand their impact on classification boundaries. The number of trees and depth in Random Forest were selected to investigate the trade-off between model complexity and overfitting, which is especially pertinent in high-dimensional bioinformatics data. For neural network architectures like FFNN and CNN, the range of neurons, layers, and learning rates were chosen to balance model capacity with computational feasibility, while ensuring enough flexibility to capture complex patterns in the data.

With respect to conventional classifiers such as SVM, RF, kNN, LR, and FFNN, we exploited grid search—an exhaustive technique that systematically scrutinizes a pre-defined subset of hyperparameters. This process was executed utilizing the scikit-learn library [Kra16]. Each classifier was assigned a unique set of hyperparameters to investigate. The specific grids of hyperparameters tailored for each classifier were as follows:

- SVM: The investigation included cost parameters (C) of 0.1, 1, 10, and 100; kernel coefficients (gamma) of 0.1, 1, and 10; and kernel types inclusive of linear, rbf, and sigmoid.
- RF: The search encompassed the number of trees in the forest of 50, 100, and 200; the maximum tree depth of 5, 10, and None; and the minimum samples required to split an internal node of 2, 5, and 10.
- kNN: The evaluation incorporated the number of considered neighbors of 3, 5, 7, and 9; the prediction weight function of uniform and distance; and the algorithm used for calculating the nearest neighbors of ball_tree, kd_tree, and brute.

- LR: The investigation comprised various penalty types of l1 and l2; cost parameters (C) of 0.1, 1, 10, and 100; and optimization solvers of liblinear and saga.
- FFNN: The search included the number of neurons in the hidden layer of (512, 256, 64), (512,), and (256,); the activation function for the hidden layer activation of relu and tanh; and the weight optimization solver of adam and sgd.

For the evaluation of model performance for each hyperparameter combination, we employed stratified 5-fold cross-validation. The optimization scoring metric was the Matthews Correlation Coefficient (MCC).

In the case of our CNN model, we utilized Optuna [ASY+19], a Python library adept at hyperparameter optimization. Optuna leverages a variety of optimization algorithms to traverse the hyperparameter space with the goal of identifying the optimal values that enhance the model's performance. it works by using adaptive sampling algorithms, such as Tree-structured Parzen Estimators (TPE), to efficiently explore the hyperparameter space. It iteratively proposes new hyperparameter configurations based on the performance of previous trials, focusing on promising areas of the search space. This approach is particularly beneficial for CNNs, which often have numerous interconnected hyperparameters affecting network architecture, learning rates, and regularization techniques.

- Kernel Sizes: The possible combination were [3, 5, 7], [3, 7, 9], [5, 7], and [7, 7, 7].
- Output Channels: The combinations were [128, 64, 32].
- Dropout Probability: The range was set from 0.2 to 0.5.
- Optimizer: The options included Adam, RMSprop, and SGD.
- Learning Rate: The range extended from 1e-6 to 1e-2 on a logarithmic scale.

The model underwent training for 10 epochs, with performance being assessed after each epoch on a separate validation set using the Matthews Correlation Coefficient (MCC) as the performance metric. This validation set, distinct from both the training and test sets, was used to monitor the model's performance during training and to prevent overfitting through early stopping if necessary. The pruning feature of Optuna was harnessed to curtail trials early if they lacked promise, thereby conserving computational resources.

Evaluation Process and Computational Considerations Owing to the intensive computational requirements of this procedure in terms of time and memory, the optimization was carried out singularly for each task and dataset, thereby resulting in five distinct hyperparameter settings (IC-MP balanced, IC-MP imbalanced, IT-MP balanced, IT-MP imbalanced, and IC-IT). For balanced datasets, one dataset was randomly selected from a pool of 10 for consideration. The optimization procedure was executed for 100 trials, with each trial embodying a complete execution of the objective function with a distinct set of hyperparameters. The Optuna study was configured to maximize the MCC, and the optimization procedure was expedited by using a GPU for increased efficiency.

5.2.5 Limitation

Our study encountered limitations due to computational resource constraints. The fine-tuning of large PLMs such as ProtT5 (with 3 billion parameters) and ESM-2_15B (with 15 billion parameters) demands extensive computational resources and substantial GPU memory. We faced challenges given our access to only a single Nvidia GPU V100. This limitation led to the omission of some results in Section 5.3, where the full capabilities of these models could not be explored in our analysis.

The impact of these computational constraints extends to the completeness and generalizability of our findings. Specifically, the inability to fully exploit large-scale PLMs restricted our exploration to a subset of their applications. Additionally, as noted in Table 38, the absence of results for the direct comparison of ion channels versus ion transporters arises from the lack of such specific analyses in the referenced studies and the unavailability of tools for generating these comparisons. This gap highlights an area where future research could contribute, providing a more comprehensive understanding of distinctions between these protein classes.

5.3 Results and Discussion

The initial sections of our results discuss the performance of the TooT-PLM-ionCT system on the DS-C dataset. Then we transition to the extended validation on DS-Cv2.

We elucidate the performance of six distinct Protein Language Models (PLMs) as they engage with three specific tasks: differentiating ion channels (IC) from membrane proteins (MP), distinguishing ion transporters (IT) from MPs, and discerning IC from IT. We delve into the performance of six classifiers within these tasks, shedding light on three pivotal factors under investigation: the influence of frozen versus fine-tuned representations, the effect of balanced versus imbalanced datasets, and the impact of half versus full precision floating-point calculations.

Our findings are quantified using four evaluative metrics: Matthews Correlation Coefficient (MCC), accuracy, sensitivity, and specificity. We present these results as mean ± standard deviation, obtained from a 5-fold cross-validation (CV). In our attempt to provide an overarching view, we compute averages over tasks, PLMs and classifiers, yielding a high-level depiction of our results. It should be noted, however, that results compared against the state-of-the-art are derived from an separate test set, with all other evaluations conducted on the training set.

In our tables, the highest values for each column and category are highlighted in bold, facilitating immediate recognition. Where there are more than two comparable values, the second highest are underlined to illustrate the proximity between the best and second-best results. In the corresponding figures, we prioritize the MCC metric, owing to its reliability and comprehensive nature. Each bar in these figures denotes the mean MCC, with the error bar atop indicating the standard deviation from the 5-fold CV. A Δ symbol highlights the difference between pairs of bars.

To ascertain the statistical significance of our findings, we employ ANOVA [SW89], a method for comparing the means of three or more groups, and the paired t-test [Mow11], used to compare the means of two related groups. A p-value of 0.05 or smaller is typically considered evidence against the null hypothesis, suggesting that the observed difference may not be due to chance alone. However, it does not prove or guarantee a significant difference. The p-value represents the probability of obtaining results at least as extreme as those observed, assuming the null hypothesis is true. It is important to note that this section primarily discusses general findings; more detailed results can be found in Appendix B.

5.3.1 Analysis of PLM Performance in Protein Classification

Table 32 presents the comparative performance of six PLMs across three protein classification tasks: differentiating ion channels (ICs) from membrane proteins (MPs), ion transporters (ITs) from MPs, and ICs from ITs.

Table 32 highlights ESM-1b's exceptional performance across all classification tasks, outperforming other models in metrics such as MCC, accuracy, sensitivity, and specificity. Notably, ESM-1b's proficiency in distinguishing between ICs and ITs is paralleled only by the ESM-2_15B model.

While ESM-1b consistently emerges as the top performer, the second position fluctuates among tasks, with ESM-2 excelling in IC-MP and IT-MP classifications and ProtT5 showing strength in IC-IT discrimination. The statistical significance of these findings, as reflected in the p-values, indicates the superior performance of ESM-1b for most tasks, except for IC-IT where

MCC of ESM-2_15B matches ESM-1b's performance.

5.3.1.1 Dissecting ESM-1b's Superiority

The architectural and training distinctions of ESM-1b likely underpin its enhanced performance. Specifically, its use of pre-activation blocks and harmonic positional embeddings, coupled with a comprehensive and diverse training regimen, may better capture the complex spatial and functional attributes of proteins. This section posits that such architectural innovations, particularly in contrast to ESM-2 and other models, contribute to ESM-1b's effectiveness in protein sequence modeling.

Architectural and Training Distinctions The ESM-1b model incorporates specific architectural elements that distinguish it from other protein language models examined in this study. These features include pre-activation residual blocks and harmonic positional embeddings, which are hypothesized to contribute to its superior performance in protein classification tasks. These design choices, aimed at stabilizing deep network training and capturing intrinsic protein sequence periodicity, are hypothesized to be key factors in its superior classification performance.

Table 32: Performance of PLMs for protein classification tasks.

Task	PLM	MCC	accuracy	sensitivity	specificity	P-value
IC-MP	ProtBERT	0.73±0.05	90.99±1.76	76.88±4.89	91.69±2.83	1.25e-06
	ProtBERT-BFD	0.74±0.05	91.46±1.63	76.18±4.82	92.27±2.60	
	ESM-1b	**0.84±0.03**	**94.15±1.17**	**88.44±3.39**	94.33±1.91	
	ESM-2	0.83±0.04	93.89±1.27	85.66±4.43	**94.39±1.94**	
	ProtT5	0.79±0.05	93.12±1.38	79.68±4.98	94.35±1.81	
	ESM-2_15B	0.78±0.04	93.16±1.23	81.52±4.38	93.13±1.71	
IT-MP	ProtBERT	0.71±0.05	90.75±1.41	75.66±4.69	91.58±2.34	2.49e-03
	ProtBERT-BFD	0.74±0.05	91.10±1.64	78.91±4.79	92.30±2.33	
	ESM-1b	**0.82±0.04**	**93.47±1.31**	**85.09±3.46**	**94.53±2.09**	
	ESM-2	0.78±0.04	92.64±1.36	82.06±4.20	93.41±2.26	
	ProtT5	0.75±0.04	92.78±1.13	77.55±4.42	93.58±1.94	
	ESM-2_15B	0.72±0.04	91.58±1.46	76.12±4.26	91.90±2.32	
IC-IT	ProtBERT	0.79±0.03	89.33±1.67	88.92±4.38	89.62±4.46	2.14e-06
	ProtBERT-BFD	0.78±0.05	88.71±2.46	88.29±5.12	89.29±4.67	
	ESM-1b	**0.85±0.04**	**92.46±2.25**	**92.83±3.42**	92.12±4.21	
	ESM-2	0.83±0.04	91.42±2.17	91.21±3.62	91.83±4.21	
	ProtT5	0.84±0.04	91.83±1.83	91.00±2.67	92.50±3.83	
	ESM-2_15B	0.85±0.03	92.33±1.67	91.50±2.67	**92.83±3.83**	

This table presents averaged performance metrics of protein language models (PLMs) across three protein classification tasks: ion channels (IC) vs. membrane proteins (MP), ion transporters (IT) vs. MP, and IC vs. IT. Results are grouped by PLMs and presented as mean±standard deviation from 5-fold cross-validation. The p-value, calculated using ANOVA with a significance threshold of 0.05, indicates statistical significance of differences among PLMs for each task. Bold values indicate the highest performance, while underlined values show the second-highest, facilitating comparison between top-performing models.

Training Regimen and Data Utilization The training approach of ESM-1b, leveraging vast and varied protein sequences from UniParc and UniRef datasets, ensures a rich representation of the protein sequence space. This extensive pre-training, combined with meticulous hyperparameter optimization, likely equips ESM-1b with a nuanced understanding of protein structures and functions, contributing to its classification accuracy.

Implications of Positional Encoding The adoption of harmonic positional embeddings in ESM-1b, as opposed to learned embeddings or RoPE in other models, may offer a more refined

interpretation of protein sequences. This feature could provide ESM-1b with an enhanced ability to model the spatial relationships within protein structures, further explaining its classification success.

Dropout Strategy The strategic inclusion of dropout in ESM-1b, contrary to its complete omission in models like ESM-2, suggests a focused effort to prevent overfitting while retaining the model's capacity to learn complex protein sequence patterns. This regularization approach might be instrumental in ensuring the generalizability of ESM-1b's predictions across varied protein classification tasks.

In conclusion, the architectural and training nuances of ESM-1b, particularly in relation to positional encoding and dropout strategies, are believed to be pivotal in its exemplary performance in protein classification tasks. Future work should aim to further unravel these aspects, potentially offering deeper insights into the model's capabilities and guiding the development of more effective PLMs for bioinformatics applications.

5.3.1.2 Impact of Dataset Balance and Fine-Tuning

This study observes that larger models, namely ProtT5 and ESM-2_15B, despite being precluded from fine-tuning due to resource constraints, managed to equal the performance of the smaller model, ESM-1b, on the balanced IC-IT dataset. Intriguingly, even with the application of fine-tuning to ESM-1b, the frozen representations demonstrated their efficacy when the dataset is balanced, as evidenced in the IC-IT case.

This finding is substantiated by Table 58 and Figure 43, which depict superior performance with frozen representation on the balanced dataset. However, the difference was not statistically significant (with a p-value $>$ 0.05) across most of the PLMs, rendering this observation as noteworthy, though not decisive.

The observed phenomenon intriguingly suggests a potential connection between dataset balance and the concepts of frozen and fine-tuned representations. Rather than treating these concepts as mutually exclusive, our study proposes that different tasks may warrant exploration of varying combinations of these methodologies, indicating the necessity for a more nuanced approach in their application.

5.3.1.3 Size of PLMs and Performance

In the context of this study's protein classification tasks, we observed that the performance of Protein Language Models (PLMs) did not consistently improve with increasing model size. Specifically, ESM-1b, with 650 million parameters, outperformed ESM-2_15B, which has 15 billion parameters, across multiple tasks. Interestingly, we did not identify a clear linear correlation between the dimensionality of a PLM and its ensuing performance. As a case in point, ESM-1b, with its 650 million parameters, consistently outperformed ESM-2_15B, which boasts 15 billion parameters, even when dealing with frozen representations (refer to Table 54). This observation underscores the conclusion that the performance efficacy of a PLM does not hinge exclusively on its size. Instead, it is shaped by a more intricate interplay of factors, with architectural design playing a significant role.

5.3.2 Comparative Performance Analysis of Classifiers

Table 33 presents performance results grouped by various classifiers utilized for three distinct protein classification tasks: distinguishing IC from MP, differentiating IT from MP, and discerning IC from IT.

Our comprehensive investigation across distinct protein classification tasks, employing various classifiers, revealed a number of compelling insights.

5.3.2.1 Prominence of SVM and CNN Classifiers

In our comparative analysis of various classifiers for protein classification tasks, we found that Support Vector Machine (SVM) and Convolutional Neural Network (CNN) classifiers generally outperformed other tested classifiers. As shown in Table [Y], SVM and CNN achieved higher Matthews Correlation Coefficient (MCC) scores across the IC-MP, IT-MP, and IC-IT classification tasks compared to other classifiers such as Logistic Regression, k-Nearest Neighbors, and Random Forest.

These classifiers effectively navigate high-dimensional data and unravel complex patterns, contributing to their consistent performance. The CNN employs convolutional layers to identify local patterns in the representations and nonlinear relationships inherent in neural network layers, while the SVM excels at linear classification by distinguishing between classes efficiently by maximizing margins.

5.3.2.2 Comparison of Simple and Complex Models

Interestingly, a comparison of simple models, such as Logistic Regression (LR), and complex ones, like CNNs, indicated comparable performance levels. This observation counters the prevalent assumption that increasing model complexity necessarily results in superior performance. The consistent trend across all tasks and evaluation metrics suggests that in predicting IC and IT from MP, simpler models may deliver effectiveness on par with their more complex counterparts.

Table 33: Performance of classifiers across protein classification tasks

Task	Classifier	MCC	accuracy	sensitivity	specificity	P-value
IC-MP	LR	0.82±0.04	93.99±1.30	85.53±4.03	94.69±1.97	2.29e-14
	kNN	0.68±0.05	87.52±1.71	82.96±4.62	82.13±2.68	
	SVM	**0.84±0.04**	**94.51±1.13**	85.76±3.69	95.66±1.71	
	RF	0.69±0.05	92.00±1.38	63.96±4.59	**96.86±1.52**	
	FFNN	0.83±0.04	94.10±1.19	**86.66±3.93**	94.66±1.82	
	CNN	0.83±0.05	93.96±1.93	85.07±5.63	95.40±3.84	
IT-MP	LR	0.80±0.04	93.12±1.34	83.71±3.74	94.19±2.21	4.77e-11
	kNN	0.69±0.05	88.54±1.76	80.58±4.21	85.93±2.56	
	SVM	**0.81±0.04**	93.17±1.21	84.28±4.47	94.62±1.96	
	RF	0.65±0.05	90.33±1.62	64.35±4.47	93.57±2.14	
	FFNN	0.81±0.04	93.19±1.41	**84.61±4.04**	94.03±2.43	
	CNN	0.81±0.04	**93.70±1.15**	82.66±4.80	**95.23±2.14**	
IC-IT	LR	0.82±0.03	91.22±1.61	91.00±3.11	91.44±3.44	1.38e-17
	kNN	0.74±0.06	86.44±3.22	89.83±4.33	83.56±5.56	
	SVM	0.85±0.04	92.28±1.67	91.67±3.61	93.00±3.56	
	RF	0.79±0.04	89.28±2.22	86.28±6.06	91.72±6.06	
	FFNN	0.84±0.04	92.06±2.17	**92.11±3.56**	92.11±3.94	
	CNN	**0.86±0.03**	**92.67±1.67**	91.61±3.17	**93.78±3.39**	

This table presents averaged performance metrics of classifiers across three protein classification tasks: ion channels (IC) vs. membrane proteins (MP), ion transporters (IT) vs. MP, and IC vs. IT. Results are grouped by classifiers and presented as mean±standard deviation from 5-fold cross-validation. The p-value, calculated using ANOVA with a significance threshold of 0.05, indicates statistical significance of differences among classifiers for each task. Bold values indicate the highest performance, while underlined values show the second-highest, facilitating comparison between top-performing models.

5.3.2.3 Less Effective Classifiers

However, not all classifiers showcased this level of effectiveness. Classifiers such as the k-Nearest Neighbors (kNN) and Random Forest (RF) were identified as the least effective across these tasks and representations derived from PLMs. This finding suggests that these classifiers

may not align well with the specific nature of these tasks or the representations provided by the PLMs.

5.3.2.4 Performance Parallels Among Classifiers

Furthermore, our analysis disclosed an intriguing parallel in the performance metrics of LR and Feed-Forward Neural Networks (FFNN), and those of SVM and CNN. This pattern suggests that, despite inherent differences in their complexity and structure, these models can achieve similar results in these specific tasks.

5.3.2.5 Significance of Classifier Selection

Finally, the p-value analysis highlighted significant performance differences across the classifiers for all three tasks, emphasizing the crucial role of classifier selection in the outcomes of these prediction tasks. The observed variation implies that the effectiveness of a specific classifier may vary based on the unique characteristics of the task, underscoring the importance of thoughtful classifier selection.

5.3.3 Effects of Various Experimental Conditions

In this section, we delve deeper into our findings and their implications. We have conducted three distinct assessments to elucidate their impacts on the results and overall performance. The following subsections offer a comprehensive discussion on these critical areas of impact, namely, the implications of frozen vs fine-tuned representations, the influence of balanced vs imbalanced datasets, and the effects of half vs full precision floating-point computations.

5.3.3.1 Frozen vs Fine-tuned PLM Representations

Table 34 presents the impact of frozen and fine-tuned representations across the three tasks under consideration - IC-MP, IT-MP, and IC-IT. Additionally, Figure 27 underscores the performance, specifically focusing on the MCC metric across the three tasks. Note that a comprehensive analysis concerning the influence of frozen and fine-tuned representations is available in Section B.1.

Table 34: Comparison of frozen and fine-tuned representations

Task	Representation	MCC	accuracy	sensitivity	specificity	P-value
IC-MP	frozen	0.75±0.05	90.54±2.10	**90.52±4.04**	90.65±4.33	1.57e-08
	finetuned	**0.83±0.04**	**90.75±2.08**	90.33±3.92	**91.17±4.32**	
IT-MP	frozen	0.70±0.05	**93.11±1.41**	**86.71±3.93**	**93.44±2.25**	2.33e-12
	finetuned	**0.83±0.04**	92.33±1.47	77.61±4.80	93.06±2.26	
IC-IT	frozen	**0.82±0.04**	**92.81±1.37**	**88.22±3.56**	**93.24±2.16**	7.15e-01
	finetuned	0.81±0.04	91.37±1.45	73.48±4.87	92.68±2.31	

This table presents averaged performance metrics of frozen and fine-tuned representations across three protein classification tasks: ion channels (IC) vs. membrane proteins (MP), ion transporters (IT) vs. MP, and IC vs. IT. Results are grouped by representation type and presented as mean±standard deviation from 5-fold cross-validation across various PLMs. The p-value, calculated using Student's t-test with a significance threshold of 0.05, indicates statistical significance of differences between frozen and fine-tuned representations for each task. Bold values indicate metrics where the p-value is less than 0.05, showing statistical significance. Italic values indicate metrics where the p-value is above the 0.05 threshold, suggesting no statistical significance.

Our investigation has uncovered noteworthy disparities in the performance of fine-tuned and frozen representations across various tasks, underscored by their responses to task-specific conditions, dataset sizes, classifier choices, and the underlying PLM's architecture.

Task-specific Performance Variations and the Impact of Dataset Imbalances On differentiating IC from MP and IT from MP, fine-tuned representations have consistently outperformed frozen ones. This pattern, however, becomes less clear-cut in the IC-IT task. Statistical analysis further supports this pattern, revealing substantial performance discrepancies between frozen and fine-tuned representations in the IC-MP and IT-MP tasks. However, the IC-IT task showed no significant difference.

This relative performance convergence in the IC-IT task can be attributed to the balanced nature of its dataset, contrasting with potential imbalances in the MP dataset. This highlights the role of dataset balance in performance trends and suggests that evaluation metrics may capture varying aspects of model performance, particularly under conditions of dataset imbalance.

A case in point is the sensitivity metric for the IT-MP task. Here, frozen representations notably outshine their fine-tuned counterparts, contrasting with the general trend of fine-tuned superiority. This demonstrates the sensitivity metric's specific susceptibility to the effects of dataset imbalance. Whereas MCC metric, which accounts for all types of prediction errors, demonstrated equivalent performance for both representation types.

Influence of Dataset Size on Performance Our findings elucidate a noteworthy correlation between dataset size and the efficacy of fine-tuned model representations, particularly within the realm of membrane protein classification. The extensive, though imbalanced, membrane protein (MP) dataset, encompassing 3,413 sequences, facilitated the development of more nuanced fine-tuned representations compared to those derived from a balanced dataset consisting of merely 280 sequences. This phenomenon underscores the premise that larger datasets, by virtue of their volume, can significantly enhance the quality of fine-tuned representations in protein language models (PLMs).

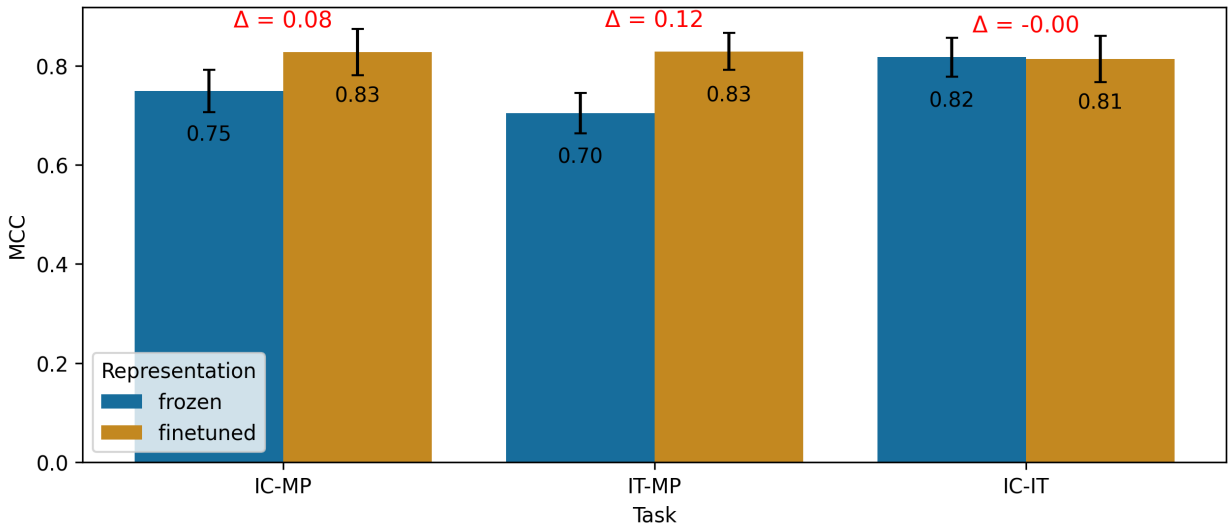

Figure 27: Graphical representation of the impact of frozen vs fine-tuned

This figure elucidates the impact of employing frozen and fine-tuned representations across a range of Protein Language Models (PLMs) for three distinct tasks: differentiating Ion Channels (IC) from Membrane Proteins (MP), distinguishing Ion Transporters (IT) from MPs, and discriminating IC from IT. The results are portrayed using the mean Matthew's Correlation Coefficient (MCC) values derived from 5-fold cross-validation. Each bar represents the mean MCC calculated across five cross-validation runs, while the error bars indicate the associated standard deviation. The symbol Δ is employed to denote the disparity between the corresponding pair of bars.

This trend posits an intriguing hypothesis regarding the untested potential of larger models, such as ProtT5 and ESM-2_15B. Given sufficient computational resources to accommodate fine-tuning, these models might demonstrate superior performance, leveraging their capacity to

absorb and reflect the intricate details present in expansive datasets.

The impact of dataset size on model performance is intrinsically linked to the biological variability inherent in membrane proteins. Membrane proteins, characterized by their diverse functions and structures, present a complex landscape for computational models. A larger dataset has the potential to capture a broader spectrum of biological diversity, potentially encompassing various sequences, structural motifs, and functional domains. However, it is important to note that increased size alone does not guarantee greater diversity, as a dataset can also grow larger due to increased redundancy or overrepresentation of certain elements. This rich representation enables models to learn more comprehensive and biologically relevant patterns, thus improving their predictive accuracy and generalizability.

In essence, the size of the dataset not only influences the depth and quality of the model's fine-tuned representations but also serves as a mirror to the biological complexity and variability of membrane proteins. By encompassing a wider array of biological features and phenomena, larger datasets equip models with a more holistic understanding of the protein universe, thereby enhancing their performance in classifying and understanding the nuanced roles of these crucial biomolecules.

Performance Across Different Classifiers A further probe into performance across all classifiers, as represented in Table 55 and Figure 40, demonstrated the consistent outperformance of fine-tuned over frozen representations. This observation reinforces the role of fine-tuning as a potent strategy to optimize PLM effectiveness across varied classifier architectures.

Performance across Diverse PLMs Our findings, as showcased in Table 54 and Figure 39, reveal that performance remains relatively stable between diverse PLM sizes when using frozen representations. However, ESM-1b, a larger model with 650M parameters, outperformed smaller-sized PLMs like ProtBERT with 420M parameters. This observation suggests that the size of the underlying PLM can exert influence on the performance of frozen representations.

Table 35: Performance of PLMs on Balanced vs Imbalanced

PLM	Dataset	MCC	accuracy	sensitivity	specificity	P-value
ProtBERT	balanced	*0.70±0.06*	*89.14±2.42*	*89.00±3.33*	*89.27±3.89*	2.52e-01
	imbalanced	*0.74±0.04*	*98.48±0.06*	*84.52±3.52*	*99.58±0.10*	
ProtBERT-BFD	balanced	*0.71±0.06*	*88.55±2.45*	*88.94±3.61*	*88.24±4.12*	1.57e-02
	imbalanced	**0.77±0.03**	**97.98±0.19**	**78.79±5.02**	**99.56±0.08**	
ESM-1b	balanced	*0.79±0.05*	*87.83±2.52*	*89.81±3.38*	*85.87±3.78*	1.38e-04
	imbalanced	**0.87±0.02**	**96.92±0.17**	**67.83±5.25**	**99.17±0.25**	
ESM-2	balanced	*0.78±0.05*	*84.82±2.94*	*85.83±4.14*	*83.87±5.04*	9.25e-03
	imbalanced	**0.83±0.03**	**96.92±0.23**	**66.71±5.44**	**99.40±0.12**	
ProtT5	balanced	*0.79±0.05*	*85.31±3.19*	*85.59±4.54*	*85.07±4.77*	4.33e-01
	imbalanced	*0.75±0.04*	*97.25±0.08*	*69.50±5.06*	*99.50±0.17*	
ESM-2_15B	balanced	*0.77±0.05*	*89.08±2.35*	*89.32±3.48*	*88.77±3.67*	6.05e-01
	imbalanced	*0.73±0.03*	*96.83±0.17*	*67.92±5.92*	*99.17±0.08*	

This comprehensive evaluation examines the performance of various Protein Language Models (PLMs) on both balanced and imbalanced datasets of membrane proteins. The results, computed using 5-fold cross-validation, are represented as mean±standard deviation for the evaluation metrics. The p-value quantifies the statistical significance of observed differences amongst the classifiers. Bold values indicate metrics where the p-value is less than 0.05, showing statistical significance. Italic values indicate metrics where the p-value is above the 0.05 threshold, suggesting no statistical significance. The PLMs are sorted based on their number of parameters.

5.3.3.2 Balanced vs Imbalanced Datasets

Table 35 and Figure 28 present the performance of the six PLMs when applied to either a balanced or imbalanced MP dataset. Our analysis suggests a profound effect of dataset balance

on the performance of different representations across PLMs, classifiers, and tasks.

Performance Across PLMs Our results, as presented in Table 35 and Figure 28, indicate that representations from imbalanced datasets outperform those from balanced datasets across six PLMs, with the exception of ProtT5 and ESM-2_15B. This inconsistency may arise from the lack of fine-tuned representations for these specific PLMs. Given the feasibility of fine-tuning, we expect that these PLMs would align with the overall trend, affirming the performance advantage of imbalanced datasets.

However, the reported p-value in Table 35 suggests no significant difference between balanced and imbalanced datasets for ProtBERT, ProtT5, and ESM-2_15B PLMs. As ProtT5 and ESM-2_15B were not fine-tuned, the observed p-value primarily reflects the impact of dataset balance on the performance of frozen representations for these PLMs.

Impact of Dataset Balance on PLM Performance The evaluation of protein language models (PLMs) on classification tasks reveals intricate dynamics between dataset balance and model performance. As delineated in Table 33 and Table 35, a nuanced pattern emerges: Our analysis of dataset balance effects revealed a nuanced pattern: for the majority of PLMs tested, imbalanced datasets resulted in higher Matthews Correlation Coefficient (MCC) and accuracy scores. However, we observed instances where balanced datasets yielded better sensitivity and specificity metrics. For example, as shown in Table [Z], the ESM-1b model achieved an MCC of 0.87 on the imbalanced dataset compared to 0.79 on the balanced dataset for the IC-MP task, while sensitivity was 89.81% for the balanced dataset versus 67.83% for the imbalanced dataset. This section delves into the possible underlying reasons for these observations, offering a scientific explanation based on the nature of the datasets and the intrinsic characteristics of PLMs.

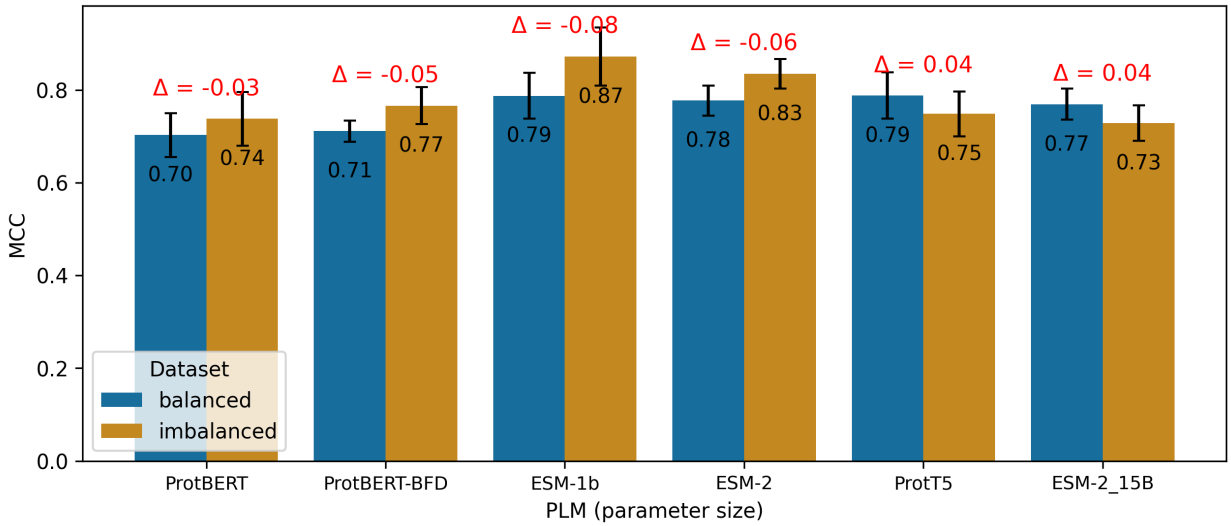

Figure 28: Evaluation of PLMs on balanced and imbalanced datasets

This figure showcases a comprehensive evaluation of various protein language models (PLMs) on both balanced and imbalanced datasets of membrane proteins. The evaluation results are depicted as the mean Matthews Correlation Coefficient (MCC) calculated over five cross-validation runs, with error bars denoting the standard deviation. The symbol Δ indicates the difference between the corresponding pair of bars, providing insights into the performance disparities across the evaluated PLMs.

Influence of Imbalanced Datasets Imbalanced datasets, characterized by a disproportionate representation of classes, often result in PLMs achieving higher MCC and accuracy. This phenomenon can be attributed to the models' tendency to better learn the features of the dominant class, thereby improving overall predictive performance on the more frequently represented class. In the context of protein classification, where certain types of proteins might be over-represented,

PLMs like ESM-1b and SVM tend to excel in overall accuracy and MCC due to their robust feature extraction capabilities which are honed on the prevalent class.

Advantages of Balanced Datasets Conversely, balanced datasets provide an equal representation of classes, facilitating a more equitable learning environment for PLMs. This balance allows models to equally learn features from all classes, often resulting in improved sensitivity and specificity. sensitivity (true positive rate) benefits from a balanced dataset as the model is equally exposed to all classes, improving its ability to correctly identify true positives across the board. Similarly, specificity (true negative rate) is enhanced as the model learns to accurately reject non-members of each class due to equal exposure to negative examples for each class.

Scientific Rationale The observed dichotomy in performance metrics between balanced and imbalanced datasets can be scientifically rationalized through the lens of learning biases and the nature of the classification tasks. In imbalanced datasets, PLMs may develop a bias towards the majority class, leading to higher overall accuracy and MCC, as these metrics are influenced by the model's ability to correctly predict the dominant class. However, this bias might come at the cost of decreased sensitivity to minority classes, which is less of an issue in balanced datasets.

Balanced datasets, by providing an equal representation of all classes, mitigate this bias, enabling PLMs to achieve a more harmonious sensitivity to all classes, as reflected in sensitivity and specificity metrics. This equilibrium in class representation fosters a more comprehensive learning of the protein sequence space, allowing PLMs to develop a more nuanced understanding of the features distinguishing each class.

Task-specific Performance Variations Evidence from Table 56 and Figure 41 indicates a superior performance of imbalanced datasets in the IC-MP and IT-MP tasks. These findings underscore the impact of dataset balance on model performance across these specific tasks.

Performance Across Different Classifiers The comparison of classifier performances presented in Table 59 and Figure 44 suggests that imbalanced datasets outshine balanced datasets across all classifiers, except for the RF classifier. This exception implies a particular sensitivity of the RF classifier to dataset balance, potentially explaining its performance divergence from the other classifiers.

Table 36: Performance of half vs full precision floating-point

Classifier	Precision	MCC	accuracy	sensitivity	specificity	P-value
LR	half	*0.82±0.04*	*93.56±1.62*	*85.54±5.08*	*94.99±3.04*	9.69e-01
	full	*0.81±0.04*	*93.62±1.53*	*85.32±4.58*	*95.02±3.10*	
kNN	half	*0.69±0.05*	*93.20±1.50*	*86.95±3.90*	*93.78±2.56*	9.01e-01
	full	*0.70±0.05*	*93.43±1.45*	*86.92±3.90*	*93.99±2.44*	
SVM	half	*0.83±0.04*	*92.93±1.42*	*85.91±3.80*	*93.65±2.44*	9.22e-01
	full	*0.83±0.04*	*93.22±1.35*	*85.88±3.68*	*94.00±2.29*	
RF	half	*0.69±0.05*	*90.81±1.73*	*70.20±5.09*	*94.31±2.76*	9.64e-01
	full	*0.70±0.05*	*90.77±1.58*	*67.29±4.63*	*94.68±2.61*	
FFNN	half	*0.82±0.04*	*93.40±1.28*	*86.41±3.98*	*94.59±2.24*	9.27e-01
	full	*0.82±0.04*	*93.63±1.26*	*86.30±3.99*	*94.81±2.13*	
CNN	half	*0.83±0.04*	*87.85±2.04*	*83.29±4.37*	*84.35±3.19*	8.09e-01
	full	*0.83±0.04*	*87.60±2.03*	*83.46±4.42*	*83.60±3.22*	

This table presents averaged performance metrics of classifiers using half and full precision floating-point calculations across protein classification tasks. Results are grouped by classifier and precision type, presented as mean±standard deviation from 5-fold cross-validation. The p-value, calculated using Student's t-test with a significance threshold of 0.05, indicates statistical significance of differences between half and full precision for each classifier. Italic values indicate metrics where the p-value is above the 0.05 threshold, suggesting no statistical significance.

Fine-Tuned vs Frozen Representations The performance patterns as seen in Table 57 and Figure 42 demonstrate that imbalanced datasets exhibit superior performance when employing fine-tuned representations across all fine-tuned PLMs. In contrast, balanced datasets perform better when using frozen representations, except for ProtBERT, where the p-value of 8.66e-02 indicates a statistically significant difference. These findings emphasize the significant impact of dataset balance on model performance, dependent on the choice of representation type (fine-tuned or frozen).

5.3.3.3 Half vs Full Precision Floating Point Calculations

Table 36 and Figure 29 present the outcomes obtained from employing half and full precision floating-point calculations across the classifiers. Our analysis explores the influence of numerical precision—specifically half versus full precision floating-point calculations—on the performance of different tasks, classifiers, and PLMs.

Performance Across Different Classifiers As evidenced by the results presented in Table 36 and Figure 29, the performance remains consistent across all classifiers, irrespective of whether half or full precision floating-point calculations are employed. This suggests that the level of numerical precision does not significantly affect classifier performance in the evaluated tasks.

Task-specific Performance Variations Performance consistency extends to specific tasks as well. As shown in Table 60 and Figure 45, the IC-MP, IT-MP, and IC-IT tasks exhibit comparable performance levels regardless of the employed floating-point precision. These findings reinforce the notion that the numerical precision choice for the floating-point calculations does not materially affect model performance across these tasks.

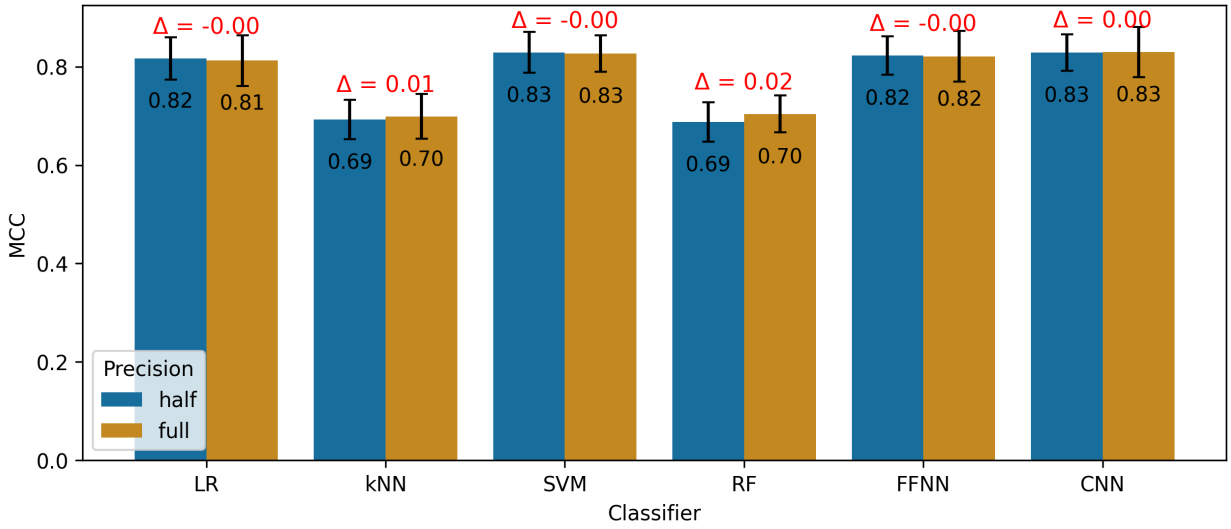

Figure 29: Half vs full precision evaluation across classifiers.

This evaluation compares the performance of different protein language models (PLMs) using both half and full precision floating-point calculations. The results are presented as the mean Matthews Correlation Coefficient (MCC) calculated across five cross-validation runs, with error bars indicating the standard deviation. The symbol Δ represents the difference between the corresponding pair of bars, providing insights into the impact of numerical precision on classifier performance.

Performance Across PLMs The performance comparison among the six PLMs, as displayed in Table 61 and Figure 46, reveals minor performance variations when using both half and full precision floating-point calculations. This observation implies that the selection of floating-point precision has minimal impact on the performance of the evaluated PLMs.

Influence on Evaluation Metrics and Statistical Significance An overarching analysis of evaluation metrics and p-values reveals no statistically significant differences between the usage of half and full precision floating-point calculations across varied tasks, classifiers, and PLMs. These findings underscore that the choice of floating-point precision does not exert a considerable influence on the outcomes of the prediction tasks assessed in this study.

5.3.4 Visualization of Representations: Insights and Implications

The UMAP projection matrix of representations derived from the ESM-1b PLM, presented in Figure 30, provides a compelling visualization of both frozen and fine-tuned representations for balanced and imbalanced datasets within the context of the IC-MP task on the training set. It is crucial to note that the representation shown for the balanced dataset is randomly selected from one of the ten available balanced datasets.

5.3.4.1 Fine-tuned Representations in Imbalanced Dataset

The Figure 30 visualization underscores the distinct clusters and patterns within the fine-tuned representations for the imbalanced dataset. The evident separation between ion channels and membrane proteins signifies the highly discriminative capability of fine-tuned representations, demonstrating their efficacy in this task. This insight underscores the prowess of fine-tuned representations in capturing the unique and distinguishable characteristics of ion channels, fostering precise classification and analysis.

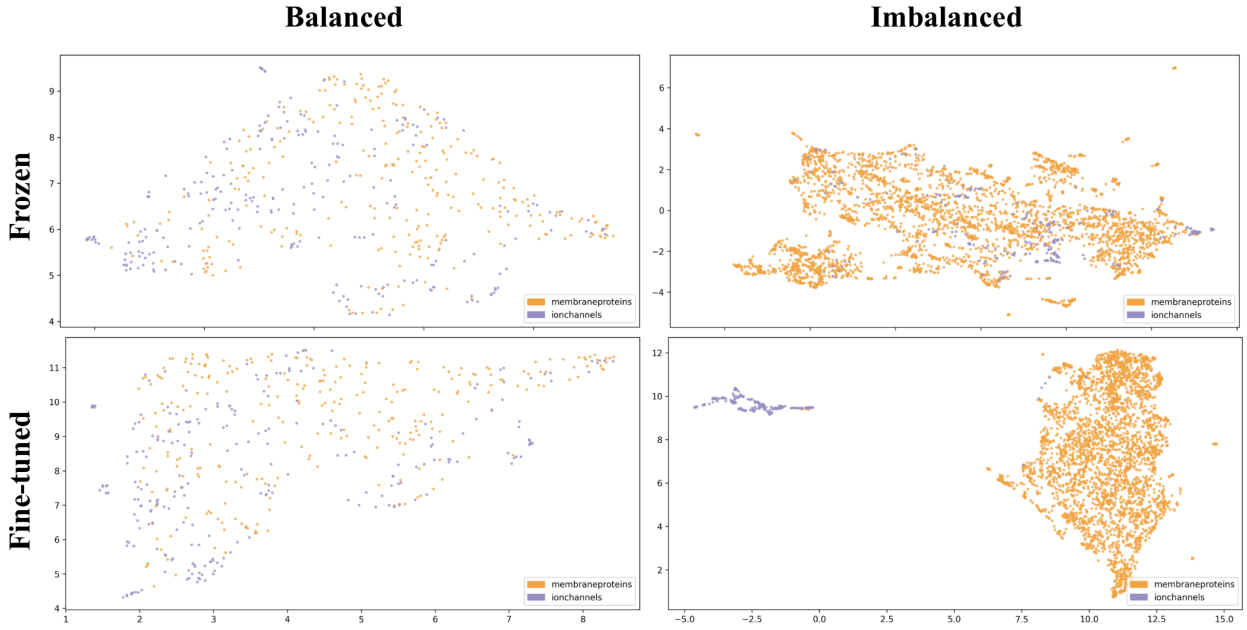

Figure 30: UMAP projection of representations from top PLMs

The figure showcases a UMAP projection of representations derived from ESM-1b, the highest-performing Protein Language Model (PLM) in the task of discriminating ion channels (IC) from membrane proteins (MP). The representations are visualized in four variations: frozen and fine-tuned representation types, along with balanced and imbalanced datasets. In the visualization, membrane protein representations are depicted in yellow, while ion channel protein representations are depicted in blue.

5.3.4.2 Frozen Representations in Imbalanced Dataset

Notably, the visualization also indicates that the next best level of clarity is achieved using frozen representations with the imbalanced dataset. This suggests that the imbalanced dataset, enriched with a broader spectrum of other membrane proteins, enhances the performance of the frozen representations. This may be due to the diversity and complexity of the other membrane

proteins, requiring a larger dataset for effective representation and discrimination. Hence, this highlights the advantage of employing imbalanced datasets with frozen representations for capturing the intricacies of diverse membrane protein structures.

5.3.4.3 Impact of Undersampling on Classification Task

Our results accentuate the potential adverse consequences of undersampling the dataset on the classification task performance. Undersampling, which reduces the dataset size, can impair the model's ability to classify proteins accurately, underscoring the need for a sufficiently large dataset to ensure effective protein classification. A substantial dataset ensures the model's exposure to diverse and representative examples, facilitating the learning of robust, discriminative patterns that generalize well to unseen data. Consequently, securing a substantial dataset is of paramount importance for achieving optimal performance in protein classification tasks.

5.3.4.4 Implications for Balanced Dataset Representations

Examining the visualization of frozen and fine-tuned representations with balanced datasets, we find a lack of clear patterns. This signifies a less distinct characterization of ion channels compared to other membrane proteins, suggesting these representations may not effectively differentiate ion channels from other membrane proteins. This lack of clear patterns implies that the representations derived from balanced datasets may fail to capture unique features or discriminative information vital for robust ion channel classification. Hence, alternative representation strategies or dataset balancing techniques may warrant consideration to enhance model effectiveness.

Table 37: Top cross-validation results for each task

Task	Representation	Representer	Dataset	Classifier	MCC	
					CV	Independent
IC-MP	finetuned	ESM-1b	Imbalanced	SVM	0.99±0.01	**0.85**
				RF	0.98±0.01	0.84
				kNN	0.99±0.01	0.83
				LR	**1.00±0.00**	**0.85**
				FFNN	**1.00±0.01**	**0.85**
				CNN	0.99±0.01	**0.85**
IT-MP	finetuned	ESM-1b	Imbalanced	SVM	**1.00±0.00**	0.68
				RF	0.99±0.01	0.67
				kNN	0.99±0.01	**0.70**
				LR	**1.00±0.00**	0.69
				FFNN	**1.00±0.01**	0.67
				CNN	0.99±0.01	0.69
IC-IT	frozen	ESM-2_15B	Balanced	SVM	0.88±0.03	**0.88**
	finetuned	ESM-1b		RF	0.84±0.03	0.79
	frozen	ProtT5		kNN	0.81±0.03	0.75
	finetuned	ESM-1b		LR	0.88±0.05	0.79
	frozen	ESM-2		FFNN	0.88±0.05	0.74
	finetuned	ESM-2		CNN	**0.89±0.03**	0.87

This table presents the best 5-fold cross-validation (CV) results for each task and classifier, as well as the corresponding results on the separate test set for comparison purposes. The tasks include discriminating ion channels (IC) from other membrane proteins (MP), ion transporters (IT) from MP, and IC against IT. The table displays the mean and standard deviation of the 5-fold CV results for each metric. The results for the IC-MP and IT-MP tasks are obtained from imbalanced datasets, while the dataset for the IC-IT task remains balanced. The best values for each task are shown in bold, and the second-best values are underlined. It is important to note that the separate test set results are provided solely for evaluating the models based on the CV results and not for selecting the best model, as the best models are chosen based on the CV results.

5.3.4.5 Comprehensive Visualization of PLMs

The representation visualizations for all six PLMs, including both frozen and fine-tuned representations for the IC-MP, IT-MP, and IC-IT tasks, are provided in Section B.3. As shown in Figure 47, Figure 48, and Figure 49, these visualizations offer a holistic view of the performance and discriminative abilities of various PLMs and representations for these tasks. These comprehensive visualizations allow for an in-depth understanding of how different PLMs capture the characteristics and separability of ion channels and other membrane proteins, illuminating their respective strengths and weaknesses.

5.3.5 Overview of Top Cross-Validation Results

The top results obtained from the 5-fold cross-validation (CV) for each task are detailed in Table 37. Results are stratified by classifier and presented in the CV column, showing the mean and standard deviation over the five folds. While separate test set results are provided for comparative purposes, they do not contribute to the selection of the best model, ensuring a robust and unbiased evaluation of classifier performance.

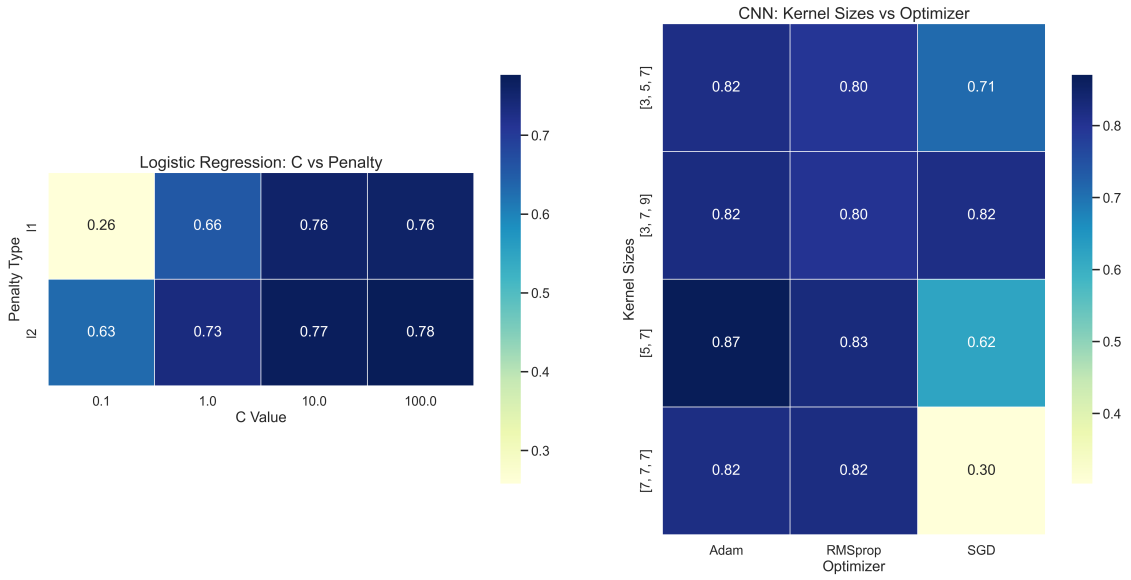

Figure 31: Hyperparameter Impact on Model Performance for LR and CNN.

This figure presents a comparative analysis of the impacts of hyperparameters on two significant models: Logistic Regression (LR) and Convolutional Neural Network (CNN). The left subplot illustrates LR's performance across varying regularization strengths ('C' values on the x-axis) and penalty types ('l1' and 'l2' on the y-axis), while the right subplot displays CNN's performance, mapping kernel sizes to optimizer configurations on the y-axis and x-axis, respectively. The heatmaps represent Matthews Correlation Coefficient (MCC) scores, where higher values indicate better model performance. Each cell in the heatmaps denotes the average MCC score for each hyperparameter combination, visually demonstrating their influence on the classifiers' efficacy.

5.3.5.1 Superior Performance of ESM-1b PLM in IC-MP and IT-MP Tasks

As outlined in Table 37, the ESM-1b PLM, combined with fine-tuned representations and an imbalanced dataset, exhibits superior performance in the IC-MP and IT-MP tasks. The LR and FFNN classifiers, in particular, achieve a perfect MCC of 1.00, indicating flawless prediction on 5-fold CV. Other classifiers also present highly competitive results, with MCC values reaching 0.99, thereby emphasizing the exceptional efficacy of the ESM-1b PLM with fine-tuning and an imbalanced dataset.

5.3.5.2 Results from Multiple PLMs in IC-IT Task

The IC-IT task, employing a balanced dataset, sees a range of PLMs delivering notable results. The top-performing classifier, CNN, leverages the ESM-2 PLM with fine-tuned representations, yielding an impressive MCC of 0.89. Notably, larger PLMs like ProtT5 and ESM-2_15B produce comparable results to their smaller counterparts such as ESM-1b and ESM-2. This suggests that the size of the PLM does not necessarily influence performance enhancement for the IC-IT task.

5.3.5.3 Comparative Performance of Classifiers for IC-IT Task

While the CNN classifier utilizing the ESM-2 PLM's fine-tuned representations achieves the top result for the IC-IT task, other classifiers also demonstrate comparable performances. The SVM classifier with frozen representations from ESM-2_15B, the LR classifier with fine-tuned representations from ESM-1b, and the FFNN classifier with frozen representations from ESM-2 deliver similar results to the CNN classifier. This suggests that a diverse set of classifiers can deliver equivalent performance levels, depending on the selected PLM and representation type.

5.3.5.4 Comprehensive Analysis of Results

A detailed examination of the results for each task - IC-MP, IT-MP, and IC-IT - is provided in Section B.3.1. Here, the evaluation metrics are delineated in detail across various tables for each task. This thorough breakdown offers an exhaustive and nuanced understanding of the performance of the employed models, classifiers, and representations. Delving into the evaluation metrics' specifics enables readers to gain deeper insights into the results, providing valuable information for future research in the prediction of ion channels and ion transporters from other membrane proteins.

For the comparison between ion channels and other membrane proteins, refer to Table 62, Table 63, Table 64, and Table 65. The analysis of ion transporters versus other membrane proteins can be found in Table 66, Table 67, Table 68, and Table 69. Lastly, the comparison between ion channels and ion transporters is detailed in Table 70, Table 71, Table 72, and Table 73.

5.3.6 Performance Comparison with State-of-the-Art Projects

Our comprehensive analysis juxtaposes the performance of TooT-PLM-ionCT with leading-edge methodologies, including DeepIon, MFPS_CNN, and TooT-BERT-C, across three critical classification challenges: distinguishing ion channels (ICs) from membrane proteins (MPs), differentiating ion transporters (ITs) from MPs, and discerning ICs from ITs. This evaluation, delineated in Tables and Figures referenced as Table 38 and Figure 32, offers an exhaustive comparison, highlighting TooT-PLM-ionCT's relative performance.

The data illustrate that TooT-PLM-ionCT either surpasses or competes closely with contemporary projects in the IT-MP and IC-IT classification tasks. Particularly, it matches the performance of TooT-BERT-C in the IC-MP task, showcasing its adeptness at accurately predicting ICs and ITs among other membrane proteins. This comparison underscores the efficacy of TooT-PLM-ionCT, not only in terms of computational performance but also in its contribution to the nuanced understanding of membrane protein function and classification.

Notably, the absence of specific results for the IC-IT task in studies such as DeepIon and MFPS_CNN highlights TooT-PLM-ionCT's novel contribution to this area. By venturing into the IC-IT classification, TooT-PLM-ionCT provides pivotal insights into the intricate distinctions between ion channels and transporters, thereby enriching our comprehension of their unique roles within biological systems.

Table 38: Comparative performance of TooT-PLM-ionCT

Task	Project	Encoder	Classifier	accuracy	MCC
IC-MP	DeepIon [TO19]	PSSM	CNN	86.53	0.37
	MFPS_CNN [NHTO22]	PSSM	CNN	<u>94.60</u>	<u>0.62</u>
	TooT-BERT-C [GB23c]	ProtBERT-BFD	LR	**98.24**	**0.85**
	TooT-PLM-ionCT	ESM-1b	LR	**98.24**	**0.85**
IT-MP	DeepIon [TO19]	PSSM	CNN	83.78	0.37
	MFPS_CNN [NHTO22]	PSSM	CNN	93.30	0.59
	TooT-BERT-C [GB23c]	ProtBERT-BFD	LR	<u>95.43</u>	<u>0.64</u>
	TooT-PLM-ionCT	ESM-1b	LR	**95.98**	**0.69**
IC-IT	DeepIon [TO19]	-	-	-	-
	MFPS_CNN [NHTO22]	-	-	-	-
	TooT-BERT-C [GB23c]	ProtBERT-BFD	LR	<u>85.38</u>	<u>0.71</u>
	TooT-PLM-ionCT	ESM-2	CNN	**93.07**	**0.87**

This table provides a comparative analysis of the performance of TooT-PLM-ionCT with the state-of-the-art methods on the separate test set. The performance is evaluated for classifying membrane proteins (MP), ion channels (IC), and ion transporters (IT). The absence of results is denoted by a "-" when corresponding studies and tools do not report ion channel and ion transporter classification against each other. The boldface highlights the highest values in the accuracy and Matthews Correlation Coefficient (MCC) columns, while the underline indicates the second-highest values.

5.3.6.1 Comparative Analysis of Hyperparameters in LR and CNN

In this section we delve into the impact of two crucial hyperparameters on the efficacy of two prominent classifiers: Logistic Regression (LR) and Convolutional Neural Networks (CNN).

For LR, the exploration was centered around the regularization strength, denoted by the parameter 'C', and the type of penalty applied, either 'l1' or 'l2'. The heatmap in Figure 31 (left) elucidates the influence of these parameters on the model's performance, measured by the Matthews Correlation Coefficient (MCC). It is evident from the heatmap that increasing the regularization strength ('C') generally leads to an improvement in performance for both 'l1' and 'l2' penalties, with 'l2' regularization slightly outperforming 'l1' across various 'C' values.

In contrast, the hyperparameter investigation for CNN focused on the kernel sizes and optimizer choices. The right subplot of Figure 31 showcases the performance landscape across different combinations of kernel sizes and optimizers, namely Adam, RMSprop, and SGD. The MCC scores depicted in the heatmap highlight the significant impact of these hyperparameters, with larger kernel sizes and the Adam optimizer generally yielding better performance.

This comparative analysis underscores the significance of hyperparameter tuning in optimizing model performance. The interplay between different hyperparameters and their configurations can markedly influence the outcome, thereby necessitating a meticulous approach to hyperparameter selection.

5.3.6.2 Model Selection Process

The selection of models for inclusion in our comparative analysis was informed by their performance in our experimental evaluations, as detailed in Table 37. In scenarios where multiple classifiers demonstrated equivalent Matthews Correlation Coefficient (MCC) scores, we opted for models that blend simplicity with effectiveness, particularly for the IC-MP and IT-MP tasks. However, the CNN classifier was chosen for the IC-IT task over the SVM, despite the latter's marginally superior performance on an separate test set, due to the CNN's enhanced performance in cross-validation (CV) results. This strategic choice underscores our commitment to balancing model complexity with performance, tailored to the specific requirements of each classification

task.

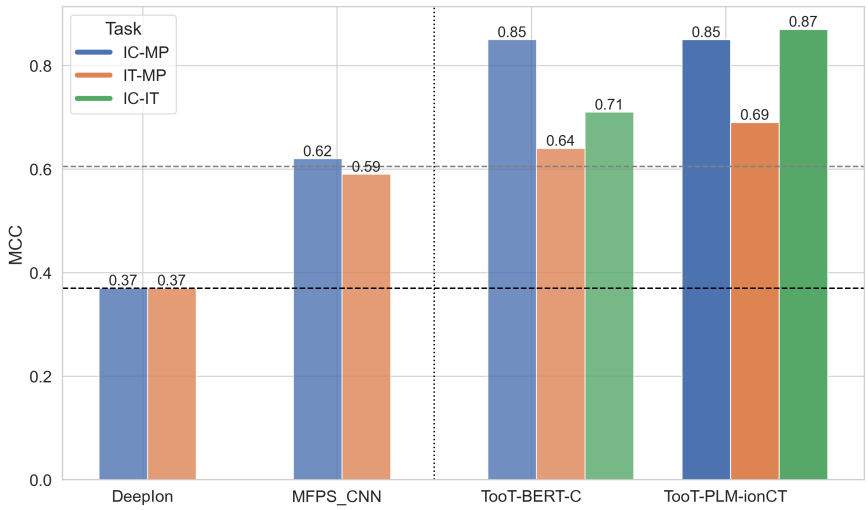

Figure 32: Comparative performance with state-of-the-art.

This figure presents the comparative performance of TooT-PLM-ionCT on the separate test set, showcasing the classification results for membrane proteins (MP), ion channels (IC), and ion transporters (IT). The absence of bars indicates studies that focused on classifying ion channels and ion transporters against membrane proteins, rather than against each other, resulting in no available results from either publications or tools. The horizontal dashed lines represent two baselines, while the vertical dashed line distinguishes between traditional and PLM-based representations.

TooT-PLM-ionCT's comparative success, especially in tasks involving IT-MP and IC-IT distinctions, signifies a substantial leap forward in our understanding of membrane protein functionality and classification. By leveraging advanced PLM technologies and comprehensive datasets, TooT-PLM-ionCT not only achieves high accuracy in classification tasks but also sheds light on the subtle biological distinctions between different types of membrane proteins. This advancement paves the way for more informed biological interpretations and applications, bridging the gap between computational predictions and biological insights.

5.3.7 Validation of TooT-PLM-ionCTv2

In this section, we delve into the extended validation of the TooT-PLM-ionCT system, emphasizing its adaptability and generalization capabilities when faced with newly annotated data from UniProtKB/Swiss-Prot. This validation is crucial for demonstrating the model's robustness and its potential applicability to evolving biological datasets. The analysis is structured into two distinct subsections, each addressing a separate aspect of the validation process.

5.3.7.1 Evaluation with the Model Trained on the Original Dataset

Experimental Setup In our pursuit to evaluate the TooT-PLM-ionCT system's adaptability, we utilized the model initially trained on the dataset curated by Taju et al. This was done to examine how well the system could perform on a novel dataset that was meticulously compiled from UniProtKB/Swiss-Prot, comprising sequences absent from the original training corpus. This new dataset, as delineated in Section 5.2.2, poses a modern challenge, testing the model's ability to generalize and adapt to newly annotated sequences. Our approach was consistent with the methodologies described in Section 5.2, maintaining the original training mechanisms and model configurations to ensure a direct and fair comparison of the model's generalization capabilities across both datasets.

Table 39: Comparison of Sequence Distribution in the Test Sets

Category / Task	**Novel Dataset Test Set**	**Taju et al. Dataset Test Set**
Per Label:		
Ion Channels (IC)	245	60
Ion Transporters (IT)	657	70
Membrane Proteins (MP)	7,334	850
Per Task:		
IC-MP	7,579	910
IT-MP	7,991	920
IC-IT	902	130
Total Sequences in Test Set:	8,236	980

This table contrasts the number of sequences across different categories (Ion Channels, Membrane Proteins, and Ion Transporters) and classification tasks (IC-MP, IT-MP, and IC-IT) in the test sets of the novel dataset and the Taju et al. dataset. The novel sequences were specifically curated to exclude any sequences present in the Taju et al. dataset, ensuring the model's evaluation on entirely unseen data. This juxtaposition highlights the expanded scope and diversity of the novel dataset for assessing the model's generalization capability.

The composition of the novel dataset is notably extensive, encompassing a total of 8,236 sequences. These sequences are distributed across various categories and classification tasks, as detailed in Table 39. This table juxtaposes the sequence distribution within the test sets of both the novel and the original Taju et al. datasets, providing a clear comparison of the datasets' scope and diversity. Specifically, the novel dataset includes 245 Ion Channels (IC), 657 Ion Transporters (IT), and a significant 7,334 Membrane Proteins (MP), showcasing an expanded range of sequences for the IC-MP, IT-MP, and IC-IT classification tasks.

This comprehensive dataset not only allows for a rigorous assessment of the TooT-PLM-ionCT system's generalization abilities but also underscores the model's potential to remain effective and relevant amidst the rapidly evolving landscape of bioinformatics data. The expanded diversity and the inclusion of entirely unseen sequences in the novel dataset are pivotal for testing the robustness and adaptability of our model, ensuring its applicability to contemporary and future bioinformatics challenges.

Results and Analysis In this section, we delve into the comparative analysis of the TooT-PLM-ionCT system's performance, evaluating its adaptability and generalization capabilities across both the novel dataset and the baseline Taju et al. dataset. Table 40 showcases the system's performance metrics, including accuracy and Matthews Correlation Coefficient (MCC), across three critical classification tasks: IC-MP, IT-MP, and IC-IT.

The IC-MP task witnessed an appreciable increase in accuracy, moving from 98.20% in the baseline dataset to 99.40% in the novel dataset, with a concurrent rise in MCC from 0.85 to 0.90. This improvement underscores the system's refined ability to discern Ion Channels from Membrane Proteins, even when introduced to previously unseen sequences, highlighting the robust classification capabilities of the Logistic Regression (LR) model and ESM-1b encoder.

Conversely, the IT-MP task experienced a slight dip in accuracy from 96.00% in the baseline dataset to 94.97% in the novel dataset. The minor decrease in MCC from 0.69 to 0.68 suggests a negligible impact on the model's overall predictive quality. This minor variance may indicate the model encountering more complex Ion Transporter sequences in the novel dataset, thus slightly challenging the established classification boundaries.

Table 40: Comparative Performance of TooT-PLM-ionCT

	Novel Dataset			Baseline Dataset		
Task	**accuracy (%)**	**MCC**	**Model**	**accuracy (%)**	**MCC**	**Model**
IC-MP	99.40	0.90	LR (ESM-1b)	98.20	0.85	LR (ESM-1b)
IT-MP	94.97	0.68	LR (ESM-1b)	96.00	0.69	LR (ESM-1b)
IC-IT	94.78	0.86	CNN (ESM-2)	93.10	0.87	CNN (ESM-2)

This table presents the comparative performance of TooT-PLM-ionCT on novel and baseline datasets.

The IC-IT task, leveraging the Convolutional Neural Network (CNN) model and the ESM-2 encoder, demonstrated a minor uptick in accuracy from 93.10% in the baseline dataset to 94.78% in the novel dataset, albeit with a slight decrease in MCC from 0.87 to 0.86. This indicates a marginal trade-off between precision and recall as the model adapts to new data variations, a common phenomenon in machine learning models.

Figure 33 presents confusion matrices derived from our study, providing a granular view of the system's performance across the aforementioned tasks and datasets.

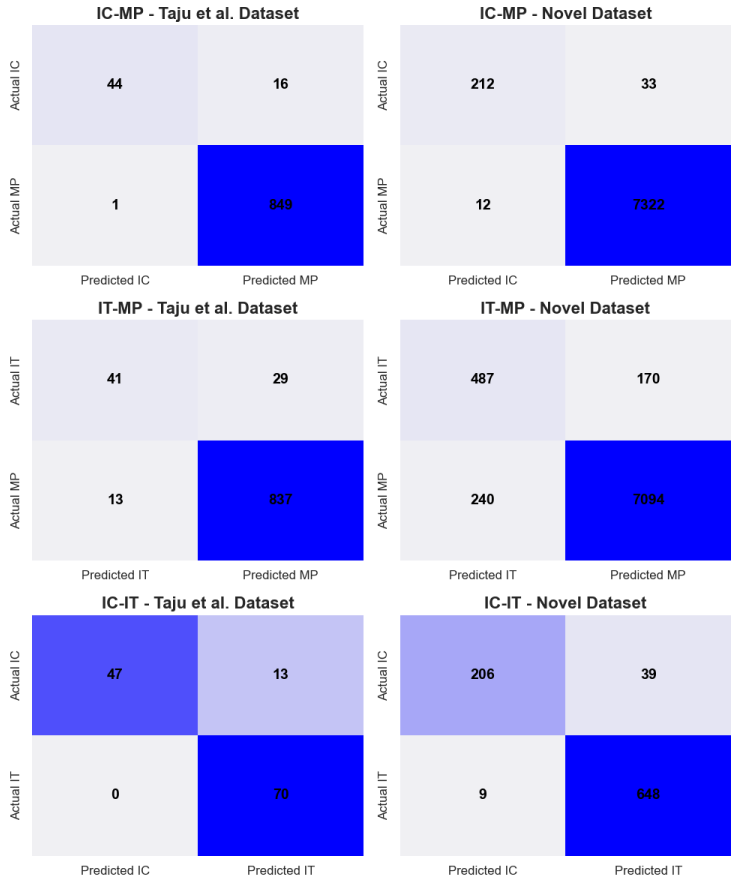

Figure 33: Comparative confusion matrices for protein classification tasks

Comparative confusion matrices for protein classification tasks: IC-MP, IT-MP, and IC-IT, across 'Novel Dataset' and 'Taju et al. Dataset'. Each matrix annotates True Positives (TP), False Negatives (FN), false positives (FP), and True Negatives (TN), illustrating the model's predictive accuracy and specificity for each class within the tasks. Darker shades of blue signify higher counts, indicating the model's performance concentration.

The confusion matrices elucidate the nuanced performance differences across tasks and datasets, with darker shades indicating higher values, reflecting the concentration of the model's predictive performance. Notably, the matrices reveal the system's enhanced sensitivity in detecting

Ion Channels and Transporters within the novel dataset, despite the increase in false positives in the IT-MP task, suggesting a broader diversity of Transporter sequences within the novel dataset.

In summary, the TooT-PLM-ionCT system exhibits commendable generalization capabilities across all tasks, with slight performance fluctuations signaling areas for potential optimization. The insights derived from both the comparative performance table and the confusion matrices not only attest to the system's robustness and reliability but also highlight the importance of continuous model evaluation against diverse and evolving datasets to ensure sustained relevance and efficacy in bioinformatics research. The expanded diversity of the novel dataset serves as a rigorous benchmark, confirming the TooT-PLM-ionCT system's potential to adapt and maintain high performance amidst the evolving landscape of bioinformatics data.

5.3.7.2 Evaluation with the Model Trained on the New Dataset

Following the validation of the TooT-PLM-ionCT system with a model trained on the established dataset by Taju et al., we embarked on a subsequent phase of validation. This phase involved retraining the system using a newly curated dataset, which is representative of the most recent annotations and encompasses a diverse array of protein sequences. The objective was to assess the adaptability and learning efficiency of the TooT-PLM-ionCT system when exposed to novel data.

Model training Employing the same methodology outlined in Section 5.2, we trained the TooT-PLM-ionCT system, maintaining consistency in the training mechanisms and parameters to ensure a fair comparison. The training process was applied to the entire suite of Protein Language Models (PLMs) and classifiers within our system, including ESM-1b and ESM-2 for their respective tasks, and logistic regression and CNN classifiers for IC-MP/IT-MP and IC-IT tasks, respectively.

Results and Analysis The trained model's performance, as shown in Table 41, illustrates significant improvements across key metrics such as Matthews Correlation Coefficient (MCC), accuracy, sensitivity, and specificity. These results underscore the system's capability to effectively learn from and adapt to the new dataset, enhancing its predictive accuracy and generalization potential.

Table 41: Extended validation of TooT-PLM-ionCT

Task	MCC		accuracy		sensitivity		specificity	
	CV	Test	CV	Test	CV	Test	CV	Test
IC-MP	0.99±0.01	0.94	1.00±0.00	0.99	0.98±0.02	0.92	1.00±0.00	0.998
IT-MP	0.99±0.01	0.90	1.00±0.00	0.99	0.99±0.01	0.89	1.00±0.00	0.994
IC-IT	0.87±0.07	0.90	0.94±0.04	0.95	0.95±0.02	0.99	0.93±0.07	0.934

Extended validation of TooT-PLM-ionCT for generalization on newly annotated data. The table presents the model's performance across various metrics including MCC (Matthews Correlation Coefficient), accuracy, sensitivity, and specificity. Results are shown as mean±standard deviation from 5-fold cross-validation (CV) alongside values from the separate test set for tasks differentiating Ion Channels (IC) from Membrane Proteins (MP), segregating Ion Transporters (IT) from MPs, and discriminating IC from IT.

A comparative analysis, detailed in Table 42 and illustrated in Figure 34, reveals the system's enhanced efficacy on the updated dataset. Notably, the improvements in MCC and accuracy highlight the system's refined predictive power and its ability to offer balanced, reliable predictions across diverse protein sequences.

Discussion The findings from this phase of validation demonstrate the TooT-PLM-ionCT system's robustness and its potential for ongoing adaptation to emerging scientific data. The consistent performance enhancement across various tasks reflects the system's flexibility and its capability to handle the complexity inherent in different protein classification challenges.

These insights not only affirm the system's generalization capabilities but also pave the way for future refinements, aiming at optimizing the system's discriminative power further, particularly in distinguishing between closely related protein families.

Table 42: Comparative Performance on Test Sets

Task	accuracy (%)		MCC	
	Taju et al.	New Dataset	Taju et al.	New Dataset
IC-MP	98.24	99.49	0.85	**0.94**
IT-MP	95.98	98.55	0.69	**0.90**
IC-IT	93.07	95.35	0.87	**0.90**

This table showcases the differences in performance metrics, specifically accuracy and MCC (Matthews Correlation Coefficient), for the tasks of differentiating Ion Channels (IC) from Membrane Proteins (MP), segregating Ion Transporters (IT) from MPs, and discriminating IC from IT, across the two datasets.

In conclusion, this phase of extended validation presents compelling evidence of the TooT-PLM-ionCT system's improved performance and adaptability when trained on a newly annotated dataset. The results validate the system's applicability to contemporary bioinformatics challenges and highlight its potential for continuous improvement in the face of evolving datasets.

5.4 Conclusions

This research introduced and extensively evaluated the TooT-PLM-ionCT system, tailored for the nuanced classification of ion channels (ICs) and ion transporters (ITs) from other membrane proteins (MPs), alongside distinguishing ICs from ITs. Utilizing a suite of six Protein Language Models (PLMs) – including ProtBERT, ProtBERT-BFD, ProtT5, ESM-1b, ESM-2, and ESM-2 (15B parameters) – in conjunction with various classifiers, the study offered a comprehensive exploration into the efficacy of these models across different classification tasks.

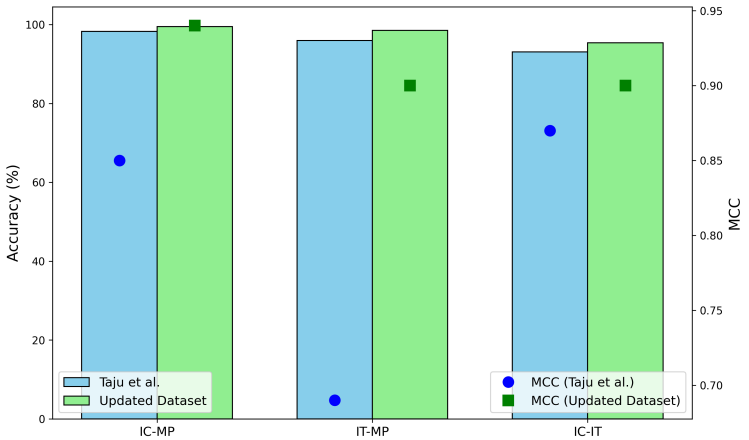

Figure 34: Comparative performance of TooT-PLM-ionCT

This figure illustrates the comparative performance of the TooT-PLM-ionCT system across three distinct classification tasks: distinguishing Ion Channels (IC) from Membrane Proteins (MP), segregating Ion Transporters (IT) from MPs, and differentiating IC from IT. Performance metrics include accuracy (presented as bar plots) and Matthews Correlation Coefficient (MCC, indicated by markers) for both the reference dataset (originally used by Taju et al.) and the updated dataset. The sky blue bars and blue circles represent the accuracy and MCC, respectively, for the reference dataset, while the light green bars and green squares denote the same metrics for the updated dataset. The juxtaposition of these datasets provides insights into the generalization capability and performance enhancements of the TooT-PLM-ionCT system, showcasing its robustness and adaptability in the face of new and diverse data.

A pivotal aspect of this study was the introduction of a new dataset, aimed at assessing the generalization capability of the TooT-PLM-ionCT system on contemporary, diverse protein sequences. This extended validation not only underscored the system's robust performance but also illuminated potential pathways for future enhancements. Key takeaways from our investigation include:

- **Superior Performance of ESM-1b:** ESM-1b emerged as the predominant PLM across most tasks, with its performance particularly pronounced in the extended validation with the new dataset. This reaffirms the model's robustness and adaptability to diverse and contemporary protein sequence data.
- **Enhanced Generalization with New Dataset:** The inclusion of the new dataset facilitated a deeper understanding of the system's generalization abilities. The observed improvements in key metrics like MCC and accuracy highlight the system's capability to adapt and maintain high performance across varied datasets.
- **Dataset Balance and Classifier Performance:** The nuanced analysis of balanced versus imbalanced datasets revealed complex dynamics influencing model performance. While imbalanced datasets generally yielded better outcomes, balanced datasets showed advantages in specific metrics, indicating the potential need for tailored dataset preparation strategies depending on the classification task.
- **Implications for Future Research:** The insights garnered from the extended validation point towards several avenues for future exploration, including the potential integration of additional knowledge sources and the investigation of more advanced sequence representation techniques. Furthermore, the exploration of larger and more diverse datasets stands as a crucial next step to validate and possibly expand the applicability of the TooT-PLM-ionCT framework.

In conclusion, the TooT-PLM-ionCT system demonstrates promising capabilities in the classification of integral membrane proteins, with the extended validation offering valuable insights into its generalization potential. The learnings from this study not only contribute to the ongoing evolution of the system but also inform broader efforts in computational bioinformatics to develop tools that are both powerful and adaptable to the ever-expanding landscape of biological data.

5.5 Availability of the TooT-PLM-ionCT System and Dataset

The TooT-PLM-ionCT system is designed to serve the academic and research community by providing a robust platform for the classification of integral membrane proteins, including ion channels and ion transporters. To facilitate widespread use and further development, both the system and the meticulously curated dataset are made publicly accessible.

System Access: Researchers interested in exploring the system's functionalities or employing it for their studies can access it via the web portal provided below:

`https://tootsuite.encs.concordia.ca/service/TooT-PLM-ionCT/`

Dataset Access: The comprehensive dataset, pivotal in the training and validation of the TooT-PLM-ionCT system, offers an extensive collection of protein sequences. This dataset is invaluable for researchers aiming to investigate the system's generalization capabilities or to benchmark its performance against other models. The dataset is available for download at:

`https://huggingface.co/datasets/ghazikhanihamed/TooT-PLM-ionCT_DB`

Chapter 6

Incorporating Secondary Structure Information into Protein Language Models

The integration of structural information into PLMs represents a significant advancement in computational biology and bioinformatics. Recent research has demonstrated the potential of incorporating three-dimensional (3D) and secondary structure data to enhance the predictive capabilities of PLMs. While models incorporating structure-aware vocabularies and adapter modules have shown promise, the full potential of integrating secondary structure information remains underexplored. Our research aims to address this gap by developing a PLM that explicitly integrates secondary structure information into its training and prediction processes, with the goal of enhancing the model's predictive accuracy and providing a more comprehensive understanding of protein functionalities, particularly for tasks involving membrane proteins and other complex protein structures.

Recent advancements in PLMs have focused on integrating structural information to enhance their predictive capabilities. This integration aims to provide a more comprehensive understanding of protein functionalities, particularly for complex structures such as membrane proteins.

Several studies have explored the integration of three-dimensional (3D) structural data into PLMs to improve predictive performance. One notable example is ProstT5 [HWS+24]. Another significant contribution is S-PLM [WPA+24], introduced by Wang et al., which employs multi-view contrastive learning to align sequence and 3D structural data. By coordinating sequence and structural information within a shared latent space, S-PLM achieves improved performance in clustering and classification tasks. These approaches demonstrate the potential of integrating 3D structural data to provide additional context beyond primary amino acid sequences, leading to more accurate predictions of protein functions and behaviors.

Despite the progress in incorporating 3D structural data, the integration of secondary structure information into PLMs remains a critical area for improvement. Secondary structures, such as alpha-helices and beta-sheets, are local folded shapes within polypeptides that play a crucial role in determining the overall three-dimensional conformation and functional properties of proteins [RFB22, Bue15, Gro10]. The incorporation of secondary structure information offers several potential benefits, including enhanced prediction accuracy for protein folding, improved identification of functional sites, and more precise modeling of interaction interfaces. However, many existing models primarily focus on the primary sequence of amino acids, overlooking the structural context provided by secondary structures. This limitation may restrict the models' predictive accuracy and their ability to generalize across diverse protein datasets.

Recent efforts have begun to address the integration of structural information, including secondary structures, into PLMs. Models such as SES-Adapter and SaProt have incorporated

structure-aware vocabularies and adapter modules to enhance PLM performance [TLZ+24, SHZ+24]. While these models represent progress in integrating structural information, the full potential of incorporating secondary structure data remains underexplored. Future research directions may include developing PLMs that explicitly integrate secondary structure information into training and prediction processes, exploring novel architectures that can effectively leverage both sequence and structural data, and investigating the impact of secondary structure integration on specific tasks, such as predicting membrane protein functions. By addressing these challenges, future PLMs may achieve enhanced predictive accuracy and provide a more comprehensive understanding of protein functionalities across a wide range of protein types and structures.

The objective of this research is to evaluate the effectiveness of integrating secondary structure information into a pretrained PLM. Specifically, we aim to determine whether this integration enhances the model's performance across various protein-related tasks. The integration was achieved using an encoder-decoder architecture based on the Ankh model [EESE+23], where the primary amino acid sequence is input to the model, and the model generates the corresponding secondary structure. This approach implicitly integrates secondary structure knowledge into the pretrained model, which has been extensively trained on primary sequence data.

By incorporating secondary structure data, we hypothesize that the model will achieve a more comprehensive understanding of protein behavior and interactions. The enhanced model, named TooT-PLM-P2S, was assessed across multiple tasks using diverse datasets. This evaluation includes fluorescence prediction, solubility prediction, sub-cellular localization prediction, ion channel classification, transporter classification, membrane protein classification, and secondary structure prediction. We also conducted a detailed analysis of prediction error cases to understand the underlying reasons for errors, employing three distinct bioinformatics-based methods: Multiple Sequence Alignment (T-Coffee), Orthologous Groups (eggNOG) Analysis, and Motif Alignment and Search Tool (MEME Suite).

The primary objectives of this study are as follows:

1. Integration of Secondary Structure Information: Develop an enhanced version of the Ankh model by incorporating secondary structure information into the training process.

2. Evaluation of Classification accuracy: Rigorously evaluate the classification performance of TooT-PLM-P2S in comparison to the baseline Ankh model. This involves diverse tasks and datasets.

3. Analysis of Failure Cases: Conduct a detailed analysis of wrongly predict instances where both the baseline and TooT-PLM-P2S model protein sequences. This analysis will use tools such as Multiple Sequence Alignment (T-Coffee), Orthologous Groups (eggNOG) Analysis, and Motif Alignment and Search Tool (MEME Suite) to uncover patterns or specific characteristics that lead to prediction errors.

Organization This chapter is organized into several sections. Section 6.1 outlines the methodological framework used in the study, detailing the model architecture, integration of secondary structure knowledge, and the evaluation approach. Section 6.2 presents the findings from the evaluation of the TooT-PLM-P2S model. Section 6.3 discusses the implications of these results. Section 6.4 summarizes the key outcomes of the study, highlighting the contributions and potential future directions for research in protein language modeling.

6.1 Materials and Methods

6.1.1 Model Architecture: TooT-PLM-P2S

To enhance protein language models (PLMs), we developed the TooT-PLM-P2S model by continuing the training of the Ankh model using a secondary structure dataset. This process

does not modify the Ankh architecture; it simply extends its training to include secondary structure information. The Ankh model remains intact throughout this process. For downstream tasks, we used ConvBERT with frozen representations from these models without further fine-tuning, following methodologies from recent studies [EHD+21, DHB22, SDHR21]. This approach allows us to assess the predictive capabilities of the representations learned by TooT-PLM-P2S in various protein-related tasks.

6.1.2 Integration of Secondary Structure Knowledge

To develop TooT-PLM-P2S, we extended the Ankh model by incorporating secondary structures like alpha-helices and beta-sheets. This integration aims to improve the model's predictive accuracy and understanding of protein functions.

The integration leverages the Ankh model's sequence-to-sequence architecture, specifically adopting its 48 encoder layers and 24 decoder layers. We used the initial weights from the pre-trained Ankh model, which had been pre-trained on 45 million primary protein sequences. This provided a strong foundation of learned representations.

Primary protein sequences are input into the model, which predicts their corresponding secondary structures. The encoder processes the primary sequence and creates a contextual representation of each amino acid. This representation is passed to the decoder, which predicts the secondary structure for each amino acid. A cross-entropy loss function measures the model's performance by quantifying the difference between the predicted and actual distributions of secondary structure types at each position.

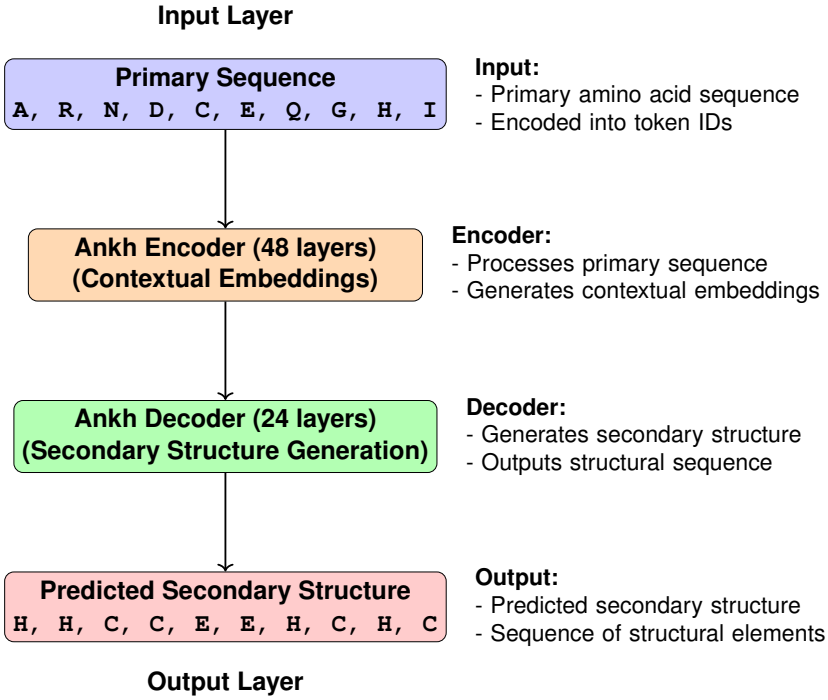

Figure 35: Schematic of the TooT-PLM-P2S Model Architecture

The model integrates secondary structure information to enhance the predictive capabilities of the protein language model. For the training the three-state of secondary structure has been used.

To create the TooT-PLM-P2S model (Figure 35), we further trained the Ankh model using a dataset that pairs primary protein sequences with their annotated secondary structures. Training was performed over 10 epochs, determined through hyperparameter optimization with Optuna, to minimize the discrepancy between predicted and actual secondary structures. The random seed was set to 32 for all experiments.

Optuna was used to fine-tune the following parameters: learning rate (0.0003), training epochs

(10), warmup ratio (0.2), weight decay (0.09), gradient accumulation steps (4), and batch size (1). The training dataset was sourced from NetSurfP-2.0 [KJN+19], designed for secondary structure prediction. It includes 11,361 protein sequences, with 8,678 training samples, 2,170 validation samples, and 513 test samples. This dataset provides classifications at both 3-class and 8-class levels.

6.1.3 Downstream Tasks

This section details the prediction tasks used to evaluate TooT-PLM-P2S, examining various aspects of the model's capabilities. These tasks employ diverse datasets, reflecting real-world challenges and benchmarks used by PEER [XZL+22] and xTrimoPGLM [CCG+23], as well as our prior projects.

We selected these tasks to span a wide range of protein functionalities and structures, ensuring comprehensive evaluation. This includes datasets focusing on membrane proteins, ion channels, and transporters, emphasizing the importance of these categories.

Table 43: Summary of Downstream Tasks for Evaluation

Task	Methodology	Data Source	#Proteins	#Classes	#Prot. Per Class(Pos./Neg.)	#Train/Valid./Test
Non-SSP Tasks						
FluP	Regression	Sarkisyan et al. [SBM+16]	54,025	-	-	21,446/5,362/27,217
SolP	Classification	DeepSol [KRK+18]	71,421	2	29,972/41,449	62,478/6,942/1,999
LocP	Classification	DeepLoc [AASS+17]	13,961	10	-	8,945/2,248/2,768
IonP	Classification	DeepIon [TO19]	4,564	2	301/4,263	3,289/365/910
TranP	Classification	TrSSP [MCZ14]	1,560	2	900/660	1,242/138/180
MemP	Classification	TooT-M [AB20a]	17,892	2	8,825/9,064	14,492/1,610/1,790
SSP Tasks						
SSP3	Classification	NetSurfP-2.0 [KJN+19]	11,361	3	-	8,678/2,170/513
SSP8	Classification	NetSurfP-2.0 [KJN+19]	11,361	8	-	8,678/2,170/513

The table summarizes the downstream tasks used to evaluate the TooT-PLM-P2S model. Each row details a specific task, including methodology, data source, number of protein sequences, and dataset splits for training, validation, and testing. Non-SSP tasks include Fluorescence Prediction (FluP), Solubility Prediction (SolP), Sub-cellular Localization Prediction (LocP), Ion Channel Prediction (IonP), Transporter Prediction (TranP), and Membrane Protein Prediction (MemP). SSP tasks involve Secondary Structure Prediction for three states (SSP3) and eight states (SSP8), reflecting the model's performance in these areas.

Fluorescence Prediction (FluP) This regression task assesses the fluorescence intensity of green fluorescent protein mutants, which is crucial for tracking proteins in live cells and organisms. The dataset, annotated by Sarkisyan et al. [SBM+16], includes mutants with up to three mutations for training and evaluation, and mutants with four or more mutations for testing. This setup tests the model's ability to generalize from lower-order to higher-order mutations.

Solubility Prediction (SolP) Solubility prediction is vital for designing effective pharmaceuticals, as soluble proteins are more likely to be functional and usable in drug formulations. This binary classification task uses the DeepSol dataset [KRK+18], ensuring no protein in the training and evaluation sets shares more than 30% sequence identity with any protein in the test set to prevent information leakage.

Sub-cellular Localization Prediction (LocP) Predicting protein localization within the cell is important for understanding protein function and interactions, especially in disease research. The DeepLoc dataset [AASS+17] classifies proteins into 10 sub-cellular localizations. This dataset is critical for evaluating the model's capability to capture functional context from protein sequences.

Ion Channels Prediction (IonP) Differentiating ion channels from other membrane proteins is essential for understanding various physiological processes and drug target identification. The

dataset used is from the DeepIon project [TO19], compiled from UniProt and refined to reduce sequence similarity below 20%. It includes ion channels, ion transporters, and other membrane proteins, with ion transporters excluded for consistency with previous methodologies.

Transporters Prediction (TranP) Identifying transporters is crucial for studying how substances move across cellular membranes, which is fundamental in cellular biology and pharmacology. This task uses a dataset from the TrSSP project [MCZ14], consisting of well-characterized transporter, carrier, and channel proteins from SwissProt. Sequences with fragmented or ambiguous annotations are excluded to ensure data quality.

Membrane Proteins Prediction (MemP) Differentiating membrane proteins from other proteins helps in understanding cellular processes and the role of membrane-bound proteins in signaling and transport. This dataset is the same as that used in the TooT-M project [AB20a], derived from Swiss-Prot. It filters for sequence quality and diversity, excluding sequences with inferred homology, less than 50 amino acids, lacking molecular function annotations, or exhibiting over 60% pairwise similarity.

SSP3 (3-State Secondary Structure Prediction) SSP3 involves predicting three states of secondary structure (alpha-helices, beta-sheets, and coils), which are fundamental for understanding protein folding and function. This task uses the NetSurfP-2.0 [KJN+19] dataset. Additional testing sets include CB513 [YGW+18], TS115 [CB99], CASP12 [ATM+18], and CASP14 [KST+21].

SSP8 (8-State Secondary Structure Prediction) SSP8 involves predicting eight distinct states of secondary structure, providing a more detailed and nuanced understanding of protein folding patterns. This task also uses the NetSurfP-2.0 dataset with the same additional testing sets (CB513, TS115, CASP12, and CASP14) to ensure robustness.

6.1.3.1 Overview of ConvBERT Use

ConvBERT [JYZ+20] (Figure 36) is employed as the downstream model in our architecture for protein structure prediction tasks due to its effectiveness in managing complex sequences. ConvBERT integrates convolutional layers with self-attention mechanisms, capturing both local and global sequence features. For each prediction task, the pre-trained TooT-PLM-P2S model generates frozen embeddings, which are then fed into ConvBERT for final predictions. This approach retains the benefits of the pre-trained models without altering their parameters, ensuring consistency across tasks.

The ConvBERT configuration used includes an embedding dimension that matches the pre-trained model, a feed-forward network dimension set to half the embedding dimension, four attention heads, a dropout rate of 0.2, a convolutional kernel size of 7, and the GatedGELU activation function [HG23]. The ConvBERT classifier comprises a ConvBERT layer followed by linear layers tailored to the task type, with no activation for regression tasks, sigmoid for binary classification, and softmax for multi-class classification. A global max pooling layer aggregates features from the convolutional layer outputs for classification.

Hyperparameter tuning for ConvBERT is conducted using Optuna [ASY+19], focusing on optimizing parameters such as learning rate, weight decay, warmup ratio, and gradient accumulation steps. The goal is to maximize the Matthews Correlation Coefficient (MCC) on the validation set. MCC is chosen for its robustness in evaluating classifier performance, particularly with imbalanced datasets [CJ20].

6.1.4 Enrichment Analysis of prediction errors

To gain deeper insights into the prediction errors made by our models, we employed several analytical tools focusing on sequence alignment, functional annotation, and motif identification. Prediction errors offer insights into the limitations of a model and highlight areas for iterative

improvement, ensuring the model's applicability across a wide range of proteins with varying complexities. In this context, an Enrichment Analysis (EA) [STM+05, RIV+19] was conducted on the wrongly predicted sequences by both the baseline Ankh model and the newly developed TooT-PLM-P2S model.

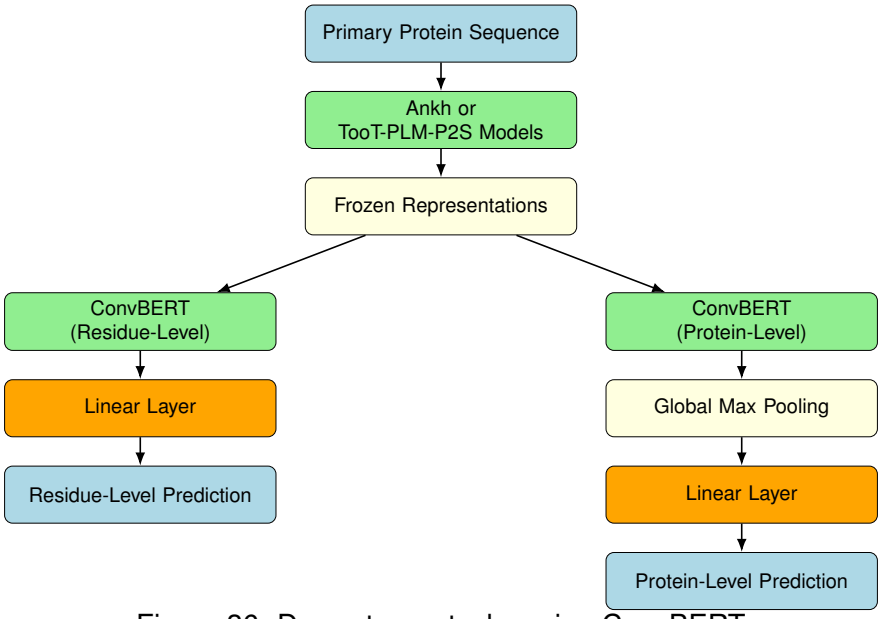

Figure 36: Downstream tasks using ConvBERT

This plot illustrates the process of applying the TooT-PLM-P2S or Ankh models for various downstream tasks.

This analysis utilized a suite of bioinformatics tools—T-Coffee, eggNOG, and the Motif Alignment and Search Tool (MAST)—each providing unique perspectives on the wrongly predicted sequences. Understanding the root causes of prediction errors is critical for improving model accuracy and reliability. Previous work [STM+05, RIV+19] has shown that analyzing wrongly predicted instances can uncover underlying patterns and biases in models.

6.1.4.1 Sequence Alignment: T-Coffee

We employed T-Coffee [NHH00] for multiple sequence alignment (MSA) to compare wrongly predicted sequences with correctly predicted ones, aiming to identify sequence regions or motifs problematic for our models. Our input consisted of FASTA files containing protein sequences for each task, separated into wrongly predicted and correctly predicted sets, which we parsed using Biopython [CAC+09].

For each task, we randomly selected five wrongly predicted sequences and five correctly predicted sequences. We then combined one wrongly predicted sequence with all five correctly predicted sequences into a temporary FASTA file. T-Coffee was executed on this combined file using the command: "t_coffee -in combined_file -outfile msa_output_file -output fasta". We repeated this process for each of the five wrongly predicted sequences, resulting in five alignment files per task.

To analyze these alignments, we calculated sequence identity by comparing amino acid residues at each position, computing the proportion of matches over the alignment length. For each wrongly predicted sequence, we calculated pairwise sequence identities with all correctly predicted sequences in the same alignment file.

We computed the average sequence identity for each alignment file and then calculated task-specific average identities by aggregating the results from the five alignment files per task. Additionally, we determined a global average identity across all tasks.

6.1.4.2 Functional Annotation: eggNOG

We utilized eggNOG [HCSH+19] to explore the evolutionary relationships and functional contexts of wrongly predicted sequences. For each task, we had separate files for wrongly predicted and correctly predicted sequences. These files were processed through eggNOG-mapper to generate comprehensive annotation files.

The algorithm began by parsing the eggNOG-mapper output files, extracting key information such as Clusters of Orthologous Groups (COG) categories. We then extracted and counted the frequency of COG categories for both wrongly predicted and correctly predicted sequences. We counted COG category across all tasks and classification types. This allowed us to identify the most prevalent COG categories in wrongly predicted sequences.

Our analysis produced COG category frequencies across tasks and classification types. Results include the frequencies of COG categories in wrongly predicted and correctly predicted sequences across different tasks.

6.1.4.3 Motif Analysis: MEME Suite

We utilized the MEME Suite [BJGN15] to identify recurring amino acid patterns, known as motifs, within the protein sequences that our model predicted correctly and incorrectly. A motif is a short, conserved sequence of amino acids that may correspond to functional or structural elements within proteins. Our analysis focused on the Zero or One Occurrence per Sequence (ZOOPS) model, which assumes each sequence may contain at most one occurrence of each motif. The input data consisted of two sets of sequences for each of the five tasks: one set of wrongly predicted sequences and another of correctly predicted sequences, provided to MEME in FASTA format.

For each task and classification type, MEME was run to discover the top three motifs enriched in the sequence set. We then parsed the MEME output files to extract motif information, including motif ID and number of occurrences. The script compiled this information, calculating the total number of motifs and their occurrences for each task and classification type.

Results include the number of occurrences for each motif in wrongly predicted and correctly predicted sequences, and a comparative analysis of motif prevalence between wrongly predicted and correctly predicted sequences.

6.2 Results

6.2.1 Overview of Results

This section presents an overview of the comparative performance of the TooT-PLM-P2S model and the Ankh model. Then we provide performance analysis categorized by secondary structure prediction (SSP) and non-SSP tasks. The task-specific performance evaluation is presented in Appendix C.

Table 44 presents the cross-validation results in summary. While Table 45 and Figure 37 presents the separate test set results in summary. Then, we discuss the results from the separate test set to validate the model's generalizability and real-world applicability.

In our comparative analysis of the Ankh and TooT-PLM-P2S models across various protein prediction tasks, we found that several performance differences did not reach statistical significance at the p¡0.05 level. Specifically, for the fluorescence task (p=0.48), solubility task (p=0.80), and membrane proteins prediction task (p=0.10), the observed differences in performance metrics between the two models could not be conclusively attributed to model differences rather than random variation.

For the fluorescence intensity prediction task, a regression problem, the TooT-PLM-P2S model achieved a higher mean Spearman's ρ (0.6482 ± 0.0219) compared to the Ankh model (0.6360 ± 0.0157). However, this difference was not statistically significant (p=0.48), suggesting that we cannot confidently conclude that TooT-PLM-P2S outperforms Ankh for this specific task based

on our current evidence. For the solubility task, evaluated using the MCC, both models showed comparable performance, with a p-value of 0.80, also not statistically significant.

In the protein subcellular localization prediction task, the Ankh model demonstrated superior performance with a mean Matthews Correlation Coefficient (MCC) of 0.735 ± 0.009, compared to the TooT-PLM-P2S model's MCC of 0.677 ± 0.014. This difference was statistically significant (p=0.004), providing strong evidence that the Ankh model is more effective for this particular classification task. In the ion channels prediction task, the Ankh model showed superior performance with a p-value of 0.072, close to the significance threshold. For the transporters prediction task, the Ankh model outperformed the TooT-PLM-P2S model with a statistically significant p-value of 0.0077.

In the membrane proteins prediction task, the Ankh model showed better performance with a p-value of 0.10, not statistically significant. In the secondary structure prediction tasks, both three-state (SSP3) and eight-state (SSP8) tasks showed comparable results between the models. For SSP3, the p-value was 0.0012, and for SSP8, it was 0.00021, both indicating statistically significant differences.

Table 44: Comparative Performance Overview of PLMs on Cross-Validation

Task	Model	Metric	Mean ± Sd	P-Value
fluorescence	Ankh TooT-PLM-P2S	Spearman's ρ	0.6360 ± 0.0157 0.6482 ± 0.0219	4.8e-01
solubility	Ankh TooT-PLM-P2S	MCC	0.510 ± 0.039 0.506 ± 0.006	8.0e-01
localization	Ankh TooT-PLM-P2S	MCC	**0.735 ± 0.009** 0.677 ± 0.014	4.4e-03
ionchannels	Ankh TooT-PLM-P2S	MCC	**0.91 ± 0.03** 0.86 ± 0.06	7.2e-02
transporters	Ankh TooT-PLM-P2S	MCC	**0.78 ± 0.03** 0.69 ± 0.05	7.7e-03
mp	Ankh TooT-PLM-P2S	MCC	0.851 ± 0.009 0.842 ± 0.008	1.0e-01
ssp3	Ankh TooT-PLM-P2S	Q3	0.85 ± 0.01 0.85 ± 0.01	1.2e-03
ssp8	Ankh TooT-PLM-P2S	Q8	0.75 ± 0.01 0.74 ± 0.01	2.1e-04

This table provides an overview of the comparative performance between the Ankh and TooT-PLM-P2S models across various prediction tasks based on cross-validation datasets. For the Fluorescence Intensity Prediction Task, Spearman's Correlation Coefficient (ρ) is used as the performance metric. The Matthews Correlation Coefficient (MCC) is employed for protein function prediction tasks, while Q3 and Q8 accuracy metrics are utilized for secondary structure prediction (SSP) tasks, indicating the proportion of correctly predicted secondary structure states. Notably, values in boldface within the table signify the better performance differences between the models. The p-value shows the statistical significance of the comparison between the two models. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant.

The results on the separate test set align with those from the cross-validation analysis. Specifically, in the fluorescence task, the TooT-PLM-P2S model performed better on the Spearman's ρ metric. However, for solubility, localization, ion channels, transporters, and membrane proteins tasks, the Ankh model demonstrated superior performance.

In the SSP3 task, both models showed comparable accuracy, while in the SSP8 task, the Ankh model had a slight advantage. For the ion channels task, the Ankh model also outperformed the TooT-PLM-P2S model slightly.

Overall, the separate test set results confirm the cross-validation findings, with the Ankh model generally excelling across most tasks, while the TooT-PLM-P2S model shows potential in the fluorescence task. Detailed analysis of non-SSP and SSP tasks follows in subsequent sections for a clearer understanding of model performance.

6.2.2 Performance By Category

In this section, we categorize performance results into non-SSP (non-secondary structure prediction) tasks and SSP (secondary structure prediction) tasks.

Non-SSP tasks include classification tasks such as solubility, localization, ion channels, transporters, and membrane proteins. We evaluated the Ankh and TooT-PLM-P2S models using several metrics: MCC, F1 score, recall, precision, and accuracy. We computed the average performance for each metric across all non-SSP tasks to facilitate a clear comparison of the overall effectiveness of the two models on these classification tasks. Table 46 compares the performance of the Ankh model and the TooT-PLM-P2S model across these various tasks.

Table 45: Comparative performance of PLMs on test set

Task	Model	Metric	Value
fluorescence	TooT-PLM-P2S Ankh	Spearman's ρ	**0.6257** 0.6064
solubility	TooT-PLM-P2S Ankh	MCC	0.494 **0.517**
localization	TooT-PLM-P2S Ankh	MCC	0.661 **0.766**
ionchannels	TooT-PLM-P2S Ankh	MCC	0.82 **0.83**
transporters	TooT-PLM-P2S Ankh	MCC	0.63 **0.82**
mp	TooT-PLM-P2S Ankh	MCC	**0.865** 0.845
ssp3	TooT-PLM-P2S Ankh	Q3	0.81 0.81
ssp8	TooT-PLM-P2S Ankh	Q8	0.68 0.68

This table presents a comparison of the performance of the Ankh model and the TooT-PLM-P2S model across various prediction tasks using an separate test set. For the fluorescence intensity prediction task, Spearman's correlation coefficient (ρ) is used as the performance metric. Matthews correlation coefficient (MCC) is employed for protein function prediction tasks. Q3 and Q8 accuracy metrics are utilized for secondary structure prediction (SSP) tasks, indicating the proportion of correctly predicted secondary structure states. Values in boldface within the table indicate superior performance between the models.

Overall, the Ankh model outperformed the TooT-PLM-P2S model on several metrics during cross-validation, specifically MCC, F1 score, recall, and accuracy. However, for the precision metric, both models demonstrated comparable performance. These cross-validation results indicate that the Ankh model generally provides superior performance compared to the TooT-PLM-P2S model.

The p-values associated with these results show statistical significance for the F1 score and recall metrics, as they are less than 0.05. However, for MCC, precision, and accuracy, the p-values are not less than 0.05, indicating that the differences in these metrics are not statistically significant.

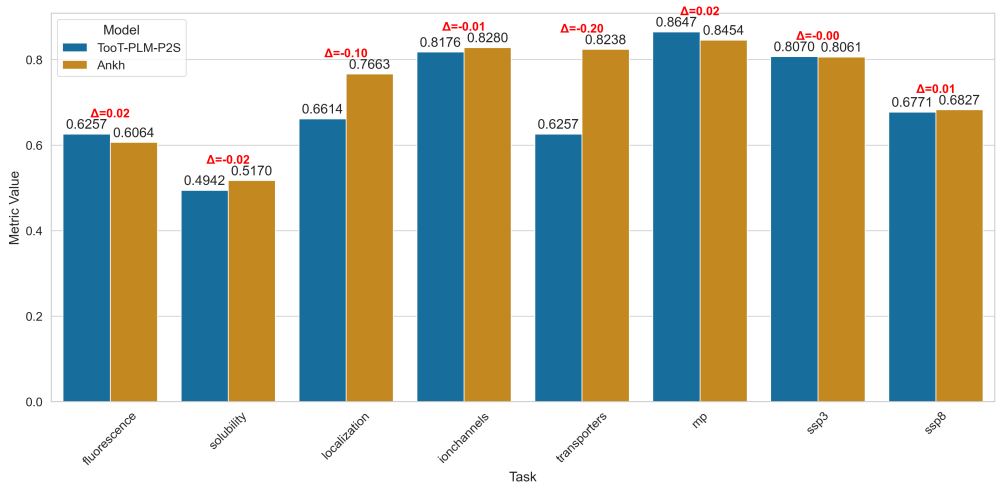

Figure 37: Comparative performance overview on the test set

This figure provides a side-by-side comparative analysis of the Ankh model and the TooT-PLM-P2S model across a range of prediction tasks on the separate test set. The x-axis enumerates the distinct tasks evaluated, while the y-axis quantifies the performance metrics. Each task features a pair of bars, distinguished by colors as denoted in the figure's legend, corresponding to the performance of the Ankh and TooT-PLM-P2S models. The visual representation highlights the performance differences, making it easier to identify which model performs better on each task. See corresponding Table 45.

Additionally, the performance of the two models was compared on the test set for the non-SSP tasks. Consistent with the cross-validation results, the Ankh model demonstrated better performance than the TooT-PLM-P2S model on the test set.

Table 47 presents the evaluation metrics for the SSP tasks, SSP3 and SSP8, including F1 score, recall, precision, and accuracy, computed as averages across cross-validation and test sets. The p-values are also included to assess the statistical significance of the results.

Table 47 shows that Ankh outperformed the TooT-PLM-P2S model where differences were statistically significant.

Table 46: Non-SSP tasks overview of PLMs comparison

Metric	Model	Cross-Validation	Test Set	P-Value
MCC	Ankh	0.757 ± 0.138	0.756	0.0566
	TooT-PLM-P2S	0.714 ± 0.128	0.693	
F1	Ankh	0.844 ± 0.081	0.838	0.0165
	TooT-PLM-P2S	0.807 ± 0.083	0.799	
Recall	Ankh	0.862 ± 0.069	0.821	0.0392
	TooT-PLM-P2S	0.791 ± 0.102	0.812	
Precision	Ankh	0.837 ± 0.092	0.866	0.9780
	TooT-PLM-P2S	0.838 ± 0.093	0.797	
accuracy	Ankh	0.867 ± 0.090	0.878	0.2482
	TooT-PLM-P2S	0.850 ± 0.093	0.841	

This table presents a comparison of non-SSP tasks between the Ankh and TooT-PLM-P2S models on both the cross-validation and test sets. The p-values indicate the statistical significance of the results. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. MCC stands for Matthews correlation coefficient. Data aggregation was performed by grouping results by model, task type, evaluation metric, and data partition (cross-validation or test set) to compute summary statistics.

The statistical significance analysis indicated by p-values reveals significant differences in F1 score, recall, and accuracy between the models, with p-values less than 0.05. For precision,

where the TooT-PLM-P2S model outperformed the Ankh model, the p-value is greater than 0.05, indicating the result is not statistically significant.

Overall, these findings suggest that while the Ankh model excels in F1 score and recall for SSP tasks, both models perform similarly in terms of accuracy. The TooT-PLM-P2S model shows an edge in precision, although this result is not statistically significant.

6.3 Enrichment Analysis of Failure Cases

Understanding prediction errors by machine learning models is crucial for enhancing their performance and reliability in protein classification. prediction errors reveal model limitations and guide iterative improvements. This Enrichment Analysis (EA) of wrongly predicted sequences by the Ankh and TooT-PLM-P2S models utilized T-Coffee, eggNOG, and the Motif Alignment and Search Tool (MAST) to investigate alignment patterns, evolutionary contexts, and motif characteristics. These findings provide a thorough examination of prediction errors, informing future model refinements. Table 48 presents the number of common correctly predicted and wrongly predicted sequences for both models analyzed in this section.

6.3.1 T-Coffee Sequence Alignment

Our T-Coffee sequence alignment analysis, as illustrated in Figure 38, revealed task-specific differences in average sequence identities between misclassified and correctly classified protein sequences. For instance, the transporters task showed the highest average identity (0.51), while the solubility task exhibited the lowest (0.32). These variations suggest that the difficulty of accurate prediction may be influenced by the degree of sequence similarity within each task's dataset.

Table 47: SSP tasks overview of the PLMs comparison

Metric	Model	Cross-Validation	Test Set	P-Value
F1	Ankh	0.692 ± 0.169	0.621	0.0067
	TooT-PLM-P2S	0.684 ± 0.176	0.616	
Recall	Ankh	0.673 ± 0.186	0.613	0.0020
	TooT-PLM-P2S	0.665 ± 0.190	0.608	
Precision	Ankh	0.759 ± 0.102	0.664	0.4850
	TooT-PLM-P2S	0.760 ± 0.103	0.678	
accuracy	Ankh	0.800 ± 0.056	0.744	0.0001
	TooT-PLM-P2S	0.797 ± 0.058	0.742	

This table presents a comparison of SSP tasks between the Ankh and TooT-PLM-P2S models, displaying averages of F1 score, recall, precision, and accuracy for both SSP3 and SSP8 across cross-validation and test sets. The p-values indicate the statistical significance of these results. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. Data aggregation was performed by grouping results by model, task type, evaluation metric, and data partition (cross-validation or test set) to compute summary statistics.

For the transporters task, the average sequence identity is approximately 0.51, indicating substantial similarity between wrongly predicted and correctly predicted sequences. This suggests that prediction errors may arise from subtle differences not captured by current model features, possibly in critical protein regions affecting function. In the solubility task, the average identity is about 0.32, reflecting greater divergence and suggesting complex factors influencing solubility that are not apparent in primary sequences alone. Tasks such as localization and ion channels show intermediate identities (around 0.41 and 0.34), indicating moderate similarity and highlighting the models' difficulty with weak or confounded functional signals. The overall average identity of 0.39 across all tasks underscores the need for models to address both conserved and variable sequence regions.

Table 48: Common Correctly Predicted and Wrongly Predicted Sequences

Task	wrongly predicted Sequences	correctly predicted Sequences	Total Sequences
Solubility	294	1293	1587
Localization	215	1206	1421
Ionchannels	13	888	901
Transporters	9	146	155
MP	93	1619	1712
Total	**624**	**5152**	**5776**

This table presents the number of common correctly predicted and wrongly predicted sequences for both the Ankh and TooT-PLM-P2S models across different tasks. The tasks include solubility, localization, ion channels, transporters, and membrane proteins (MP). The table shows the number of sequences wrongly predicted and correctly predicted in each task, along with the total number of sequences analyzed per task. The total row at the end summarizes the overall counts for all tasks combined.

6.3.2 Functional Annotation with eggNOG

Table 49 presents the eggNOG analysis of wrongly predicted sequences provides data on the distribution of correctly predicted and wrongly predicted sequences across various COG categories. In our eggNOG analysis of misclassified sequences (Table 49), we observed that the 'S' category, which represents proteins with unknown functions, had the highest absolute number of misclassifications (158 out of 1238 total sequences in this category, or 12.76%). This finding suggests that both the Ankh and TooT-PLM-P2S models face challenges in accurately classifying proteins without well-defined functional annotations, potentially due to the lack of clear feature patterns associated with known protein functions.

Table 49 indicates areas for model enhancement. prediction error rates in the 'K' (Transcription) and 'T' (Signal transduction mechanisms) categories suggest a need for improved feature engineering to better distinguish these proteins. Additionally, high prediction error in the 'S' (Function unknown) category indicates a requirement for expanded training data and better annotations for proteins with unknown functions.

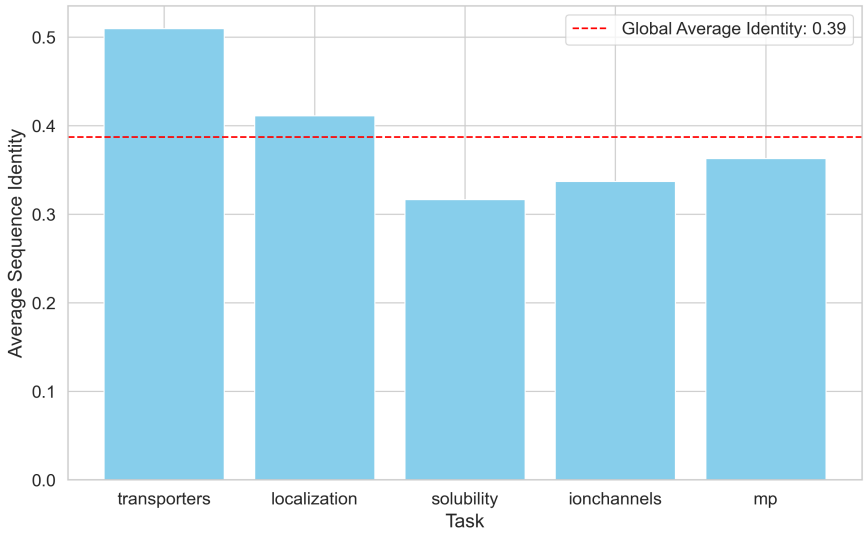

Figure 38: Average Sequence Identity for Sequences

This bar plot illustrates the average sequence identity between misclassified sequences and correctly predicted sequences across five protein classification tasks: transporters, localization, solubility, ion channels, and membrane proteins (MP). For each task, five multiple sequence alignments were analyzed. The sequence identity was calculated by comparing the single misclassified sequence to each correctly predicted sequence within each alignment, then averaging these values. The task-specific averages were computed from the five alignments per task. The overall global average identity, represented by the dashed line, is the mean of these task-specific averages, reflecting the general trend across all analyzed alignments in the dataset.

6.3.3 Motif Analysis with MEME Suite

The motif analysis using MEME Suite identified patterns in motif occurrences between correctly predicted and wrongly predicted sequences across various tasks. The data presented in Table 50 presents the occurrences of specific motifs in correctly predicted and wrongly predicted sequences across the tasks, including solubility, localization, ion channels, transporters, and membrane proteins (mp).

In our motif analysis using the MEME Suite (Table 50), we identified a specific motif 'KIHHHHHH' (#1 and #4) that occurred 144 times in misclassified sequences and 597 times in correctly classified sequences for the solubility prediction task. The higher prevalence of this motif in correctly classified sequences (80.6% of total occurrences) suggests a potential association between the presence of this motif and accurate solubility prediction. However, further statistical analysis and experimental validation would be necessary to confirm the significance and causal relationship of this observed pattern. The data indicates that this motif occurs significantly more often in sequences that were correctly predicted as soluble, compared to those that were wrongly predicted. This implies that the motif #1 and #4 'KIHHHHHH' could be an important feature that the model uses to correctly identify solubility, and it might be contributing to the model's ability to make accurate predictions regarding the solubility of proteins. However, this is an observation that would require further analysis to confirm its significance and to understand the underlying biological reasons for this correlation.

Table 49: EggNOG Analysis of wrongly predicted Sequences

COG Category	Description	correctly predicted	wrongly predicted	Percentage wrongly predicted
A	RNA processing and modification	130	21	13.89%
E	Amino acid transport and metabolism	132	21	13.73%
G	Carbohydrate transport and metabolism	196	22	10.09%
J	Translation, ribosomal structure and biogenesis	208	26	11.11%
K	Transcription	469	79	14.42%
L	Replication, recombination and repair	144	26	15.29%
O	Post-translational modification, protein turnover, chaperones	304	29	8.71%
P	Inorganic ion transport and metabolism	243	23	8.64%
S	Function unknown	1080	158	12.76%
T	Signal transduction mechanisms	525	31	5.58%

This table presents the results of the eggNOG analysis, detailing the distribution of correctly predicted and wrongly predicted protein sequences across various Clusters of Orthologous Groups (COG) categories. Each row represents a specific COG category, providing insights into RNA processing, amino acid transport, carbohydrate metabolism, translation, transcription, replication, post-translational modification, inorganic ion transport, and signal transduction mechanisms.

Other motifs such as #2 'SSGLVPRGSHM' and #3 'SSGRENLYFQGHMNP' were found less frequently in wrongly predicted sequences, while motifs like #5 'PIKHSYTPHISHTHTHTHTHTHRSDVIKLQPLPVMAPWGLPAHPHEPASM' and #6 'RPVPRRPSTWSRRSRRLHGMHLSRKRRWSCTTNFLNLSNWTRMTGPRPRQ' were rare in correctly predicted sequences, indicating specific motifs' varying impact on classification accuracy.

In the localization task, motifs such as #7 'QQQQQQQQQQQ' and #8 'CQYCDKAFSRLENLKIHERSHTGEKPYKC' were more frequent in wrongly predicted sequences, with 16 and 6 occurrences respectively, while the motif #11 'SNNNNNNNNNNNNNNSNNN' appeared 13 times in correctly predicted sequences.

This pattern is consistent across other tasks, such as ion channels and transporters, where correctly predicted sequences generally exhibited higher occurrences of certain motifs, like #16 'KWKYVTALYFAFTVJTTIGFGNVSPNTDSGKIFCIIYMLJGSLL' in ion channels (11 occurrences) and #22 'FSSYPDALYFAVVTMTTVGYGDVVPKTDSGK' in transporters (9 occurrences). The motif occurrences for membrane proteins also highlighted distinct patterns, with motifs such as #30 'HQRTHTGEKPYKCEECGKAFSRKSNLKRHQRT' appearing 43 times in correctly predicted sequences, indicating a significant motif presence correlating with accurate membrane protein

classification.

Table 50: Motif occurrences in correctly and wrongly predicted sequences

Task	No.	Motif	Classification	Occurrences
solubility	1	KIHHHHHH		144
	2	SSGLVPRGSHM	wrongly predicted	5
	3	SSGRENLYFQGHMNP		3
	4	KIHHHHHH		597
	5	PIKHSYTPHISHTHTHTHTHTHRSDVIKLQPLPVMAPWGLPAHPHEPASM	correctly predicted	2
	6	RPVPRRPSTWSRRSRRLHGMHLSRKRRWSCTTNFLNLSNWTRMTGPRPRQ		2
localization	7	QQQQQQQQQQQ		16
	8	CQYCDKAFSRLENLKIHERSHTGEKPYKC	wrongly predicted	6
	9	RYENQKRRDWNTFCQYLRNHRPPLSLPRCSGAHVLEFLRYLDQFGKTKVH		2
	10	QQQQQQQQLQQQQRAQLZQQQPQAQJQQQRQQQQQQQQQQQQQQQQQHIPQ		3
	11	SNNNNNNNNNNNNNNSNNN	correctly predicted	13
	12	WPWQVSLRYEGEHLCGGAIIAENWIVTAASCVYDRKHPKVW		5
ionchannels	13	RHRHHKQ		2
	14	DHHAPW	wrongly predicted	3
	15	WNCCGP		2
	16	KWKYVTALYFAFTVJTTIGFGNVSPNTDSGKIFCIIYMLJGSLL		11
	17	HDNYRNNPFHNFRHCFCVAQMMYSMVWLCGLQEKFSQMDILILMTAAICH	correctly predicted	2
	18	KMENFDYSNEEHLTLLKMILIKCCDISNEVRPMEVAEPWVDCLLEEYFMQ		2
transporters	19	KKKPRRCNGFKMF		2
	20	YCNEECNCEECQW	wrongly predicted	2
	21	NEYFDNLLPKCGFCQ		2
	22	FSSYPDALYFAVVTMTTVGYGDVVPKTDSGK		9
	23	IIDNFNQQKKKFGGQDIFMTEEQKKYYNAMKKLGSKKPQKPIPRPANKFQ	correctly predicted	2
	24	LRVLRAFRVLRVFKLARHWPGLRILGKTJRASVGELGLLILFLAIGVFIF		5
mp	25	EGYNGCIFAYGQTGSGKTYTMTG		3
	26	PCVDGWVYDQSVFLSTAVTEWDLVCGRQ	wrongly predicted	3
	27	MWVGNKKRTVAATNMNEESSRSHAVFTIK		3
	28	PLVCEVNGTWYLVGIVSWGEGCGRPNKPGVYTRVTSYLDWI		5
	29	HQRTHTGEKPYACDECGKAFTQQSHLEKHMKV	correctly predicted	8
	30	HQRTHTGEKPYKCEECGKAFSRKSNLKRHQRT		43

This table shows the complete list occurrences of specific motifs in sequences that were correctly predicted and wrongly predicted across various tasks. Each row details the task, the motif sequence, the classification status (whether the motif was found in correctly predicted or wrongly predicted sequences), and the number of occurrences of the motif. Note that ”mp” stands for membrane proteins.

6.4 Conclusion

This study aimed to enhance Protein Language Models (PLMs) by integrating secondary structure information, specifically alpha-helices and beta-sheets, into the TooT-PLM-P2S model. We evaluated TooT-PLM-P2S across several protein-related tasks, including fluorescence prediction, solubility prediction, sub-cellular localization, ion channel classification, transporter classification, membrane protein classification, and secondary structure prediction.

Our experiments demonstrated that incorporating secondary structure information into PLMs did not provide evidence of improvement over Ankh. The Ankh model outperformed TooT-PLM-P2S in three out of eight tasks and this is not statistically significant. For the remaining five tasks, there were no statistically significant differences between the models, suggesting that any observed improvements in TooT-PLM-P2S could be due to random variations rather than true model enhancements. Notably, the only statistically significant improvement for TooT-PLM-P2S was in the precision metric for membrane protein classification, aligning with recent findings from other studies [HWS$^+$24] when integrating 3D structural data.

These findings indicate that achieving statistical significance in future studies may require larger test sets or improved methods for integrating secondary structure information. Additionally, understanding how Ankh captures secondary structure information [CBB$^+$22, LAR$^+$22, VMV$^+$21], particularly the roles of sequence identity and evolutionary conservation, is essential for refining these models. Future research should focus on expanding datasets and refining integration methods to enhance the robustness and accuracy of protein classification models.

Chapter 7

Conclusion

This thesis addressed the challenge of accurately predicting and characterizing membrane proteins, which are crucial for cellular functions and drug discovery but remain poorly understood due to their structural complexity. The research aimed to leveraging Protein Language Models (PLMs) and deep learning techniques.

The study sought to develop improved methodologies for representing membrane proteins in computational models using advanced PLMs. It explored the potential enhancement of PLMs by incorporating secondary structure information, aiming to bridge the gap between primary sequence analysis and structural protein functionality. The research also focused on improving predictive accuracy for ion transporters and channels, which are significant potential drug targets. To address the challenge of limited annotated data in protein function prediction, the study employed transfer learning offered by PLM.

To be precise, the four specific research questions were:

Q1) Can PLMs outperform state-of-the-art classifiers for membrane proteins (M), transporters (T), and ion channels (C) classification? Which PLM-based approaches are most effective for these specific tasks, and how does their performance compare to existing methods?

Q2) How can we best combine PLMs with downstream machine learning (ML) or deep learning (DL) classifiers for protein tasks? Should we use frozen representations or fine-tuning approaches, and which ML/DL classifiers are most appropriate?

Q3) Are the effectiveness of PLMs and their optimal utilization strategies universal across different protein classification tasks, or do they vary depending on the specific task (e.g., M, T, C classification)?

Q4) What types of protein-related information are captured in PLMs, and how can we effectively incorporate additional information, such as secondary structure, to improve their performance on protein classification tasks, particularly for membrane proteins, transporters, and ion channels?

These research objectives were pursued through four interconnected projects, each exploring various aspects of PLM application and enhancement in membrane protein prediction and classification. These projects collectively contributed to advancing our understanding and predictive capabilities in the field of membrane protein research.

7.1 Improvement in Classifications

To answer Q1 we presented a comparative analysis of SOTA methodologies and our contributions for three key tasks in membrane protein bioinformatics: membrane protein classification (M), transporter classification (T), and ion channel classification (C). Our research has led to the development of new tools that utilize protein language models and various machine learning approaches for feature extraction and classification.

7.1.1 Membrane Proteins

The state-of-the-art for membrane protein classification prior to this work was represented by TooT-M. Our research produced two new tools to address this classification task.

The first, TooT-BERT-M (Section 3.4), utilizes the ProtBERT-BFD protein language model for feature extraction. It achieved performance comparable to TooT-M with 92.46% accuracy and an MCC of 0.85 on the separate test set, with a slight improvement in specificity (93.61% vs 92.50%). The second tool, TooT-BERT-CNN-M (Section 4.2), combines ProtBERT-BFD features with a CNN classifier. This approach demonstrated improved performance with 94.02% accuracy and an MCC of 0.88 on the separate test set. Table 51 presents a comparison of these methods, allowing for a comprehensive view of their relative performance.

Table 51: Summary of Membrane Protein Classification Methods

Category	Method	sensitivity (%)	specificity (%)	accuracy (%)	MCC
SOTA	iMem-2LSAAC	74.52	83.90	79.27	0.59
	MemType-2L	88.67	90.19	89.44	0.79
	TooT-M	**92.41**	92.50	92.46	0.85
Our Tools	TooT-BERT-M	91.28	93.61	92.46	0.85
	TooT-BERT-CNN-M	91.61	**96.36**	**94.02**	**0.88**

All results are from the separate test sets of DS-M. Bold indicates the best overall performance for each metric.

7.1.2 Transporters

The state-of-the-art for transporter classification was TooT-T. Our research developed two new methods to address this classification task.

The first method, TooT-BERT-T (Section 3.5), uses ProtBERT-BFD for feature extraction and achieved 93.89% accuracy and an MCC of 0.86 on the separate test set. The second approach, TooT-BERT-CNN-T (Section 4.3), combines ProtBERT-BFD features with a CNN classifier and achieved 95.00% accuracy and an MCC of 0.89 on the separate test set. Table 52 presents a comparison of these methods with previous approaches.

Table 52: Summary of Transporter Classification Methods

Category	Method	sensitivity (%)	specificity (%)	accuracy (%)	MCC
SOTA	SCMMTP	80.00	68.33	76.11	0.47
	TrSSP	76.67	81.67	80.00	0.57
	Nguyen et al.	**100.00**	77.50	85.00	0.73
	TooT-T	94.17	88.33	92.22	0.82
Our Tools	TooT-BERT-T	95.83	90.00	93.89	0.86
	TooT-BERT-CNN-T	95.00	**95.00**	**95.00**	**0.89**

All results are from the separate test sets of DS-T. Bold indicates the best performance for each metric.

7.1.3 Ion Channels

Recent advancements in ion channel classification methods prior to this work included DeepIon and MFPS_CNN.

Our research developed several new methods to further improve ion channel classification. The first method, TooT-BERT-C (Section 3.6), uses ProtBERT-BFD for feature extraction and a logistic regression classifier, achieving 98.24% accuracy and an MCC of 0.85 on the IC-MP task. The second approach, TooT-BERT-CNN-C (Section 4.4), combines ProtBERT-BFD features with a CNN classifier, achieving 98.35% accuracy and an MCC of 0.86 on the IC-MP task.

We also developed TooT-PLM-ionCT, a tool that addresses multiple classification tasks (ion channel (IC)-membrane proteins (MP), ion transporter (IT)-MP, and IC-IT) using various

protein language models (ESM-1b, ESM-2) and classifiers (LR, CNN). Finally, we evaluated TooT-PLM-ionCTv2 (Section 5.3.7), which uses the same model trained on DS-C dataset as TooT-PLM-ionCT but was tested on an updated and expanded dataset (DS-Cv2), approximately 8 times larger than the original DS-C dataset. On this larger dataset, it achieved 99.00% accuracy and an MCC of 0.94 for the IC-MP task, 99.00% accuracy and an MCC of 0.90 for the IT-MP task, and 95.00% accuracy and an MCC of 0.90 for the IC-IT task. Table 53 presents a comprehensive comparison of these methods.

Table 53: Summary of Ion Channel Classification Methods

Category	Method	Task[a]	Model Specifications[b]	accuracy (%)	MCC
SOTA	DeepIon	IC-MP	PSSM, CNN	86.53	0.37
	MFPS_CNN	IC-MP	PSSM, CNN	94.60	0.62
Our Tools	TooT-BERT-C	IC-MP	ProtBERT-BFD, LR	98.24	0.85
	TooT-BERT-CNN-C	IC-MP	ProtBERT-BFD, CNN	98.35	0.86
	TooT-PLM-ionCT	IC-MP	ESM-1b, FT, IB, LR, H	98.24	0.85
		IT-MP	ESM-1b, FT, IB, LR, H	95.98	0.69
		IC-IT	ESM-2, FT, B, CNN, H	93.07	0.87
	TooT-PLM-ionCTv2	IC-MP	ESM-1b, FT, IB, LR, H	**99.00**	**0.94**
		IT-MP	ESM-1b, FT, IB, LR, H	**99.00**	**0.90**
		IC-IT	ESM-2, FT, B, CNN, H	**95.00**	**0.90**

All results are from separate test sets. Results for TooT-PLM-ionCTv2 are based on the larger DS-Cv2 dataset, while other results use the original DS-C dataset. [a] IC: Ion Channels, MP: Membrane Proteins, IT: Ion Transporters [b] FT: Finetuned, IB: Imbalanced, B: Balanced, LR: Logistic Regression, H: Half precision

7.2 Other Contributions

This research has made several key contributions to the field of protein classification using PLMs. This section addresses three research questions (Q2, Q3, and Q4).

Q2: Fine-tuned representations typically yield superior performance compared to frozen representations, particularly for tasks with extensive datasets. This trend is evident in IC-MP and IT-MP tasks (Section 5.3.3.1 and Table 34). However, frozen representations remain competitive or advantageous for balanced datasets or tasks with limited data, as observed in IC-IT task results (also Section 5.3.3.1 and Table 34). Our analysis indicates that fine-tuning large-scale pre-trained Protein Language Models, specifically ProtBERT-BFD, coupled with task-specific Convolutional Neural Networks, provides an effective approach for various protein classification tasks (Section 4.2, Section 4.3 and Section 4.4). This method efficiently utilizes transfer learning to address limited annotated data while maintaining high performance across membrane proteins, transporters, and ion channels.

The choice of classifier depends on the specific task and dataset characteristics. From the results presented in Table 33, Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) consistently delivered superior performance across all tasks. However, simpler models like Logistic Regression (LR) showed comparable performance in many cases, particularly for IC-MP and IT-MP tasks (Table 38).

As discussed in Section 5.3.3.2, imbalanced datasets often led to better performance in terms of MCC and accuracy, while balanced datasets occasionally exhibited superior sensitivity and specificity. This suggests that the choice between balanced and imbalanced datasets should be task-specific and guided by the particular performance metrics of interest.

Larger models do not always guarantee better performance. As noted in Section 5.3, ESM-1b, with 650 million parameters, consistently outperformed larger models like ESM-2_15B (15 billion parameters) in many tasks.

The success of fine-tuned representations, especially in tasks with larger datasets (IC-MP and IT-MP in Table 34), demonstrates the effectiveness of transfer learning in addressing limited annotated data. This is particularly evident in the performance improvements seen when fine-tuning ESM-1b for these tasks (Table 34).

Q3: The effectiveness of PLMs and their optimal utilization strategies do vary across different protein classification tasks. While PLMs generally improve performance across tasks, the specific strategies for their optimal use differ depending on the classification task at hand.

The performance of PLMs varies across different tasks (membrane proteins, transporters, and ion channels). For instance, as shown in Table 32, ESM-1b consistently outperformed other models across all tasks, but the margin of improvement varied. It showed the highest performance gain in the IC-MP task (MCC of 0.84±0.03) compared to the IT-MP task (MCC of 0.82±0.04).

The optimal choice between frozen and fine-tuned representations depends on the specific task. As evidenced in Table 34, fine-tuned representations outperformed frozen ones for IC-MP and IT-MP tasks, but this difference was not significant for the IC-IT task. This suggests that the effectiveness of fine-tuning varies depending on the classification task.

Q4: PLMs primarily capture information from primary amino acid sequences. However, our study found that incorporating additional secondary structure information did not provide statistically significant improvements in most protein classification tasks, particularly for membrane proteins, transporters, and ion channels.

Specifically, as shown in Table 44 and Table 45, the TooT-PLM-P2S model, which integrated secondary structure information, did not consistently outperform the baseline Ankh model across various tasks. For instance, in the ion channels prediction task, the Ankh model showed superior performance with a p-value of 0.072, close to but not reaching statistical significance (Section 6.2). The only statistically significant improvement for TooT-PLM-P2S was observed in the precision metric for membrane protein classification, as indicated in Table 46.

Our analysis suggests that PLMs may already implicitly capture some degree of secondary structure information from primary sequences. The Enrichment Analysis of failure cases (Section 6.3) revealed that certain motifs and sequence patterns correlate with correct predictions, indicating that PLMs learn to recognize relevant sequence features.

The 10 publications resulting from these works are listed in Appendix A. Our contributions advance the understanding of how language models can be applied to biological sequences, potentially leading to new paradigms in computational biology and bioinformatics. The developed models and methodologies provide new tools for researchers in bioinformatics and related fields, with potential applications in drug discovery and personalized medicine approaches.

7.3 Limitations and Challenges

This research faced several limitations that affect the scope and generalizability of the results. The main constraints were computational resources, data limitations, and model interpretability.

Computational Resource Constraints The use of a single Tesla V100 GPU for model training and evaluation limited the scale of training data, model complexity, and hyperparameter tuning. This constraint influenced decisions on batch sizes, training duration, and model design.

Data Limitations The reliance on existing datasets for secondary structure information introduced inconsistencies, variable data quality, and potential biases. These issues affected the model's training process and may limit its ability to generalize across diverse protein families.

Model Interpretability The complexity of advanced PLMs, particularly with integrated secondary structure information, posed challenges for interpretability. Understanding the decision-making mechanisms of these models remains difficult, highlighting the need for improved interpretability methods.

Generalizability Challenges The computational and data limitations raise questions about the generalizability of the findings. Further research with enhanced resources is needed to fully explore the potential of PLMs in protein classification. The challenges in model interpretability emphasize the importance of developing models that are both accurate and transparent.

7.4 Future Research Directions

This dissertation on protein language models for membrane proteins suggests several areas for future research:

Enhancing Computational Resources Future studies should utilize advanced hardware, such as multiple GPUs or specialized TPUs, to train and evaluate more complex PLM architectures. This could enable deeper exploration of advanced models like ProtT5 (3 billion parameters) and ESM-2 (15 billion parameters), potentially yielding new insights into protein classification and prediction.

Improving Data Quality Research should focus on performing repeat masking before analyzing the differences between correct and incorrect predictions. This preprocessing step is crucial for ensuring that the analysis of prediction accuracy is not confounded by repetitive elements in the sequences, which could potentially skew the results. By removing or masking these repeats, we can more accurately assess the true performance of the prediction models and identify genuine differences between correctly and incorrectly predicted sequences. This may involve developing new data collection and preprocessing methods to ensure diverse representation of protein structures and functions, minimizing biases and improving PLM generalization across protein families.

Exploring Model Interpretability Future work should develop techniques to improve PLM transparency and explainability. Integrating explainable AI methods could clarify how these models, particularly those incorporating structural information, make predictions. This would increase trust and adoption of PLM-based solutions in fields where interpretability is crucial.

Expanding PLM Applications Future studies could broaden PLM applications in many areas such as predicting protein-protein interactions, drug-target affinity, and effects of genetic variations on protein function. This expansion could advance understanding of complex biological systems, with implications for drug discovery and personalized medicine.

7.5 Final Reflections

This section critically reflects on the key experiences, challenges, broader impact, and place of this research within the field of bioinformatics.

Key Learning Experiences The research provided insights into computational biology and bioinformatics, emphasizing the importance of interdisciplinary approaches. Integrating secondary structure information into PLMs revealed the complexities of protein functions and interactions, demonstrating the need to combine concepts from molecular biology, machine learning, and data science. This integration, while promising, also highlighted the current limitations in our understanding of the relationship between protein sequence and structure.

Challenges and Solutions The research faced computational resource constraints and data limitations. These challenges necessitated careful model design, strategic data curation, and analysis approaches to maintain accuracy and generalizability. Addressing these issues led to the development of novel methodologies. However, these constraints also underscored the ongoing need for more efficient algorithms and larger, more diverse datasets in the field of protein modeling.

Critical Evaluation of Research Impact While this research has contributed to advancing protein language modeling, its impact should be viewed in the context of the rapidly evolving field of bioinformatics. The improvements in prediction accuracy, while statistically significant, require further validation on larger, more diverse datasets to establish their robustness and

generalizability. The integration of secondary structure information, although innovative, yielded mixed results, indicating that our understanding of how to effectively incorporate structural data into sequence-based models remains incomplete.

Place within the Broader Field This work sits at the intersection of deep learning and protein biology, contributing to the growing body of research applying natural language processing techniques to biological sequences. However, it is important to note that while PLMs show promise, they are not a panacea for all protein prediction challenges. Other approaches, such as physics-based modeling and experimental methods, remain crucial for a comprehensive understanding of protein function and structure.

Future Directions and Open Questions Key questions remain about the optimal way to integrate structural information into sequence-based models, the interpretability of complex PLMs, and their applicability to a wider range of protein families. Additionally, the field would benefit from more systematic comparisons between PLM-based approaches and other state-of-the-art methods in protein prediction.

In conclusion, while this research has contributed to advancing the field of protein language modeling, it also highlights the complexities and challenges that remain in accurately predicting and understanding protein structure and function. The journey of scientific inquiry continues, with each step forward revealing new questions and avenues for exploration.

Bibliography

[AASS+17] José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. DeepLoc: Prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, November 2017.

[AB20a] Munira Alballa and Gregory Butler. Integrative approach for detecting membrane proteins. *BMC Bioinformatics*, 21(19):575, December 2020.

[AB20b] Munira Alballa and Gregory Butler. TooT-T: Discrimination of transport proteins from non-transport proteins. *BMC Bioinformatics*, 21(3):25, April 2020.

[ABW+04] Rolf Apweiler, Amos Bairoch, Cathy H. Wu, Winona C. Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, Maria J. Martin, Darren A. Natale, Claire O'Donovan, Nicole Redaschi, and Lai-Su L. Yeh. UniProt: The universal protein knowledgebase. *Nucleic Acids Research*, 32:D115–D119, January 2004.

[Agg18] Charu C. Aggarwal. Convolutional Neural Networks. In Charu C. Aggarwal, editor, *Neural Networks and Deep Learning: A Textbook*, pages 315–371. Springer International Publishing, Cham, 2018.

[Agg22] Charu C. Aggarwal. Attention Mechanisms and Transformers. In Charu C. Aggarwal, editor, *Machine Learning for Text*, pages 369–391. Springer International Publishing, Cham, 2022.

[AGM+90] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, October 1990.

[AHJ18] Muhammad Arif, Maqsood Hayat, and Zahoor Jan. iMem-2LSAAC: A two-level model for discrimination of membrane proteins and their types by extending the notion of SAAC into Chou's pseudo amino acid composition. *Journal of Theoretical Biology*, 442:11–21, April 2018.

[AKB+19] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, December 2019.

[Alb20] Munira Alballa. *Predicting Transporter Proteins and Their Substrate Specificity*. PhD Thesis, Concordia University, April 2020.

[AlQ19] Mohammed AlQuraishi. AlphaFold at CASP13. *Bioinformatics*, 35(22):4862–4865, November 2019.

[AMAZ17] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, August 2017.

[AMS+97] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, September 1997.

[ANFS09] Markus Sällman Almén, Karl JV Nordström, Robert Fredriksson, and Helgi B. Schiöth. Mapping the human membrane proteome: A majority of the human membrane proteins can be classified according to function and evolutionary origin. *BMC Biology*, 7(1):50, August 2009.

[Ash21] Md Ashrafuzzaman. Artificial intelligence, machine learning and deep learning in ion channel bioinformatics. *Membranes*, 11(9):672, September 2021.

[ASY+19] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA, July 2019. Association for Computing Machinery.

[ATM+18] Luciano A. Abriata, Giorgio E. Tamò, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Matteo Dal Peraro. Assessment of hard target modeling in CASP12 reveals an emerging role of alignment-based contact prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 86(S1):97–112, 2018.

[AVD22] Dimitrios Amanatidis, Konstantina Vaitsi, and Michael Dossis. Deep Neural Network Applications for Bioinformatics. In *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–9, September 2022.

[BBM+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):e0130140, July 2015.

[BCB16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, May 2016. arXiv:1409.0473 [cs, stat].

[BCF+07] William A. Baumgartner, K. Bretonnel Cohen, Lynne M. Fox, George Acquaah-Mensah, and Lawrence Hunter. Manual curation is not sufficient for annotation of genomic databases. *Bioinformatics (Oxford, England)*, 23(13):i41–i48, July 2007.

[Bis95] Christopher M. Bishop. Neural Networks for Pattern Recognition. In *Pattern Recognition and Machine Learning*, pages 225–290. Oxford University Press, 1995. Chapter 5.

[BJGN15] Timothy L. Bailey, James Johnson, Charles E. Grant, and William S. Noble. The MEME Suite. *Nucleic Acids Research*, 43(W1):W39–W49, July 2015.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, July 2016. arXiv:1607.06450 [cs, stat].

[BRK17] Ahmad Hassan Butt, Nouman Rasool, and Yaser Daanial Khan. A treatise to computational approaches towards prediction of membrane protein and its subtypes. *The Journal of Membrane Biology*, 250(1):55–76, February 2017.

[BS03] Evgeny Byvatov and Gisbert Schneider. Support vector machine applications in bioinformatics. *Applied Bioinformatics*, 2(2):67–77, January 2003.

[Bue15] Lukas Buehler. The Structure of Membrane Proteins. In *Cell Membranes*. Garland Science, 2015. Section: 3.

[CAC+09] Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, June 2009.

[CAPPQ97] Juan Cedano, Patrick Aloy, Josep A. Pérez-Pons, and Enrique Querol. Relation between amino acid composition and cellular location of proteins. *Journal of Molecular Biology*, 266(3):594–600, February 1997.

[CB99] James A. Cuff and Geoffrey J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519, 1999.

[CBB+22] Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Christina Floristean, Anant Kharkar, Koushik Roy, Charlotte Rochereau, Gustaf Ahdritz, Joanna Zhang, George M. Church, Peter K. Sorger, and Mohammed AlQuraishi. Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology*, 40(11):1617–1623, November 2022. Publisher: Nature Publishing Group.

[CC05a] Kuo-Chen Chou and Yu-Dong Cai. Prediction of membrane protein types by incorporating amphipathic effects. *Journal of Chemical Information and Modeling*, 45(2):407–413, March 2005. Publisher: American Chemical Society.

[CC05b] Kuo-Chen Chou and Yu-Dong Cai. Using GO-PseAA predictor to identify membrane proteins and their types. *Biochemical and Biophysical Research Communications*, 327(3):845–847, February 2005.

[CCG+23] Bo Chen, Xingyi Cheng, Li-ao Gengyang, Shen Li, Xin Zeng, Boyan Wang, Gong Jing, Chiming Liu, Aohan Zeng, Yuxiao Dong, Jie Tang, and Le Song. xTrimoPGLM: Unified 100B-scale pre-trained transformer for deciphering the language of protein, July 2023. Pages: 2023.07.05.547496 Section: New Results.

[CE99] Kuo-Chen Chou and David W. Elrod. Prediction of membrane protein types and subcellular locations. *Proteins: Structure, Function, and Bioinformatics*, 34(1):137–153, 1999.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967.

[Cho01] Kuo-Chen Chou. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, 43(3):246–255, 2001. Publisher: Wiley Online Library.

[CJ20] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews Correlation Coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, January 2020.

[CS07] Kuo-Chen Chou and Hong-Bin Shen. MemType-2L: A web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochemical and Biophysical Research Communications*, 360(2):339–345, August 2007.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[CZC03] Yu-Dong Cai, Guo-Ping Zhou, and Kuo-Chen Chou. Support vector machines for predicting membrane protein types by using functional domain composition. *Biophysical Journal*, 84(5):3257–3263, May 2003. Publisher: Elsevier.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805 [cs]*, May 2019.

[DHB22] Nicki Skafte Detlefsen, Søren Hauberg, and Wouter Boomsma. Learning meaningful representations of protein sequences. *Nature Communications*, 13(1):1914, April 2022.

[Edi03] Michael Edidin. Lipids on the frontier: a century of cell-membrane bilayers. *Nature Reviews Molecular Cell Biology*, 4(5):414–418, May 2003. Publisher: Nature Publishing Group.

[EESE+23] Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling, January 2023. arXiv:2301.06568 [cs, q-bio].

[EHD+21] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards cracking the language of life code through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, February 2021.

[FH22] Noelia Ferruz and Birte Höcker. Towards controllable protein design with conditional transformers. *arXiv:2201.07338 [q-bio]*, January 2022.

[FNZ+12] 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, December 2012.

[GB22] Hamed Ghazikhani and Gregory Butler. TooT-BERT-M: Discriminating membrane proteins from non-membrane proteins using a BERT representation of protein primary sequences. In *2022 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–8, August 2022.

[GB23a] Hamed Ghazikhani and Gregory Butler. Enhanced identification of membrane transport proteins: a hybrid approach combining ProtBERT-BFD and convolutional neural networks. *Journal of Integrative Bioinformatics*, 20(2), June 2023. Publisher: De Gruyter.

[GB23b] Hamed Ghazikhani and Gregory Butler. A study on the application of protein language models in the analysis of membrane proteins. In José Manuel Machado, Pablo Chamoso, Guillermo Hernández, Grzegorz Bocewicz, Roussanka Loukanova, Esteban Jove, Angel Martin del Rey, and Michela Ricca, editors, *Distributed Computing and Artificial Intelligence, Special Sessions, 19th International Conference*, Lecture Notes in Networks and Systems, pages 147–152, Cham, 2023. Springer International Publishing.

[GB23c] Hamed Ghazikhani and Gregory Butler. TooT-BERT-C: A study on discriminating ion channels from membrane proteins based on the primary sequence's contextual representation from BERT models. In *Proceedings of the 9th International Conference on Bioinformatics Research and Applications*, ICBRA '22, pages 23–29, Berlin, Germany, 2023. Association for Computing Machinery.

[GB23d] Hamed Ghazikhani and Gregory Butler. TooT-BERT-T: A BERT Approach on Discriminating Transport Proteins from Non-transport Proteins. In Florentino Fdez-Riverola, Miguel Rocha, Mohd Saberi Mohamad, Simona Caraiman, and Ana Belén Gil-González, editors, *Practical Applications of Computational Biology and Bioinformatics, 16th International Conference (PACBB 2022)*, Lecture Notes in Networks and Systems, pages 1–11, Cham, 2023. Springer International Publishing.

[GGA+23] A Garnto, D Garnto, O Arishe, S Wilczynski, R Dos Anjos Moraes, J Pratt, C Webb, and F Priviero. (228) Polo-like kinase 1 modulates vascular and cavernosal reactivity in wild type mice. *The Journal of Sexual Medicine*, 20(Supplement_1):qdad060.217, May 2023.

[Gro10] M. Michael Gromiha. Chapter 2 - Protein Sequence Analysis. In M. Michael Gromiha, editor, *Protein Bioinformatics*, pages 29–62. Academic Press, Singapore, January 2010.

[HBM+23] Tymor Hamamsy, Meet Barot, James T. Morton, Martin Steinegger, Richard Bonneau, and Kyunghyun Cho. Learning sequence, structure, and function representations of proteins with language models, November 2023. Pages: 2023.11.26.568742; bioRxiv.

[HCSH+19] Jaime Huerta-Cepas, Damian Szklarczyk, Davide Heller, Ana Hernández-Plaza, Sofia K Forslund, Helen Cook, Daniel R Mende, Ivica Letunic, Thomas Rattei, Lars J Jensen, Christian von Mering, and Peer Bork. eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. *Nucleic Acids Research*, 47(D1):D309–D314, January 2019.

[HEW+19] 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 Bioinformatics*, 20(1):723, December 2019.

[HG23] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs), June 2023. arXiv:1606.08415 [cs].

[Hil01] Bertil Hille. *Ionic Channels of Excitable Membranes*, volume 21. Springer, 3 edition, 2001.

[HK12] Maqsood Hayat and Asifullah Khan. Mem-PHybrid: Hybrid features-based prediction system for classifying membrane protein types. *Analytical Biochemistry*, 424(1):35–44, May 2012.

[HK13] Maqsood Hayat and Asifullah Khan. Prediction of membrane protein types using pseudo-amino acid composition and ensemble classification. *International Journal of Computer and Electrical Engineering*, 5(5):456, 2013.

[HKY12] Maqsood Hayat, Asifullah Khan, and Mohammed Yeasin. Prediction of membrane proteins using split amino acid and ensemble classification. *Amino Acids*, 42(6):2447–2460, June 2012.

[Ho98] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, August 1998.

[HPL+18] Rhys Heffernan, Kuldip Paliwal, James Lyons, Jaswinder Singh, Yuedong Yang, and Yaoqi Zhou. Single-sequence-based prediction of protein secondary structures and solvent accessibility by deep whole-sequence learning. *Journal of Computational Chemistry*, 39(26):2210–2216, 2018.

[HSSS92] Uwe Hobohm, Michael Scharf, Reinhard Schneider, and Chris Sander. Selection of representative protein data sets. *Protein Science*, 1(3):409–417, 1992.

[HWS+24] Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Milot Mirdita, Martin Steinegger, and Burkhard Rost. Bilingual language model for protein sequence and structure, March 2024. Pages: 2023.07.23.550085 Section: New Results.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 630–645, Cham, 2016. Springer International Publishing.

[JCWJ07] Liangxiao Jiang, Zhihua Cai, Dianhong Wang, and Siwei Jiang. Survey of Improving k-Nearest-Neighbor for Classification. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, volume 1, pages 679–683, August 2007.

[JEP+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. Number: 7873 Publisher: Nature Publishing Group.

[JYZ+20] Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. ConvBERT: Improving BERT with span-based dynamic convolution. In *Advances in Neural Information Processing Systems*, volume 33, pages 12837–12848. Curran Associates, Inc., 2020.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980 [cs]*, January 2017.

[KH20] Maxat Kulmanov and Robert Hoehndorf. DeepGOPlus: Improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, January 2020.

[Kit02] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–210, November 2002. Publisher: Nature Publishing Group.

[KJN+19] Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjærgaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Sønderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, and Paolo Marcatili. NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87:520–527, 2019.

[Kra16] Oliver Kramer. Scikit-Learn. In Oliver Kramer, editor, *Machine Learning for Evolution Strategies*, pages 45–53. Springer International Publishing, Cham, 2016.

[KRK+18] Sameer Khurana, Reda Rawi, Khalid Kunji, Gwo-Yu Chuang, Halima Bensmail, and Raghvendra Mall. DeepSol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 34(15):2605–2613, August 2018.

[KST+21] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (CASP)—Round XIV. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1607–1617, 2021.

[LAR+22] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, and Alexander Rives. Language models of protein sequences at the scale of evolution enable accurate structure prediction, July 2022. bioRxiv; Pages: 2022.07.20.500902 Section: New Results.

[LAR+23] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023. Publisher: American Association for the Advancement of Science.

[LBD+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, December 1989. Conference Name: Neural Computation.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. Publisher: Nature Publishing Group.

[LD11] Hao Lin and Hui Ding. Predicting ion channels and their types by the dipeptide mode of pseudo amino acid composition. *Journal of Theoretical Biology*, 269(1):64–69, January 2011.

[LDB+04] Rasko Leinonen, Federico Garcia Diez, David Binns, Wolfgang Fleischmann, Rodrigo Lopez, and Rolf Apweiler. UniProt archive. *Bioinformatics*, 20(17):3236–3237, November 2004.

[LG06] Weizhong Li and Adam Godzik. CD-HIT: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, July 2006.

[LGG01] N. M. Luscombe, D. Greenbaum, and M. Gerstein. What is Bioinformatics? A proposed definition and overview of the field. *Methods of Information in Medicine*, 40(4):346–358, 2001. Publisher: Schattauer GmbH.

[LH19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, January 2019. arXiv:1711.05101 [cs, math].

[LHD+21] Maria Littmann, Michael Heinzinger, Christian Dallago, Konstantin Weissenow, and Burkhard Rost. Protein embeddings and deep learning predict binding residues for various ligand classes. *Scientific Reports*, 11(1):23916, December 2021. Publisher: Nature Publishing Group.

[LJY+21] Yunan Luo, Guangde Jiang, Tianhao Yu, Yang Liu, Lam Vo, Hantian Ding, Yufeng Su, Wesley Wei Qian, Huimin Zhao, and Jian Peng. ECNet is an evolutionary context-integrated deep learning framework for protein engineering. *Nature Communications*, 12(1):5743, September 2021. Number: 1 Publisher: Nature Publishing Group.

[Lou20] Antoine Louis. *NetBERT: A Pre-trained Language Representation Model for Computer Networking*. PhD thesis, Université de Liège, Liège, Belgique, 2020. Backup Publisher: Louppe, Gilles.

[LVY+15] Yi-Fan Liou, Tamara Vasylenko, Chia-Lun Yeh, Wei-Chun Lin, Shih-Hsiang Chiu, Phasit Charoenkwan, Li-Sun Shu, Shinn-Ying Ho, and Hui-Ling Huang. SCMMTP: Identifying and characterizing membrane transport proteins using propensity scores of dipeptides. *BMC Genomics*, 16(12):S6, December 2015.

[LWC05] Hui Liu, Meng Wang, and Kuo-Chen Chou. Low-frequency Fourier spectrum for predicting membrane protein types. *Biochemical and Biophysical Research Communications*, 336(3):737–739, October 2005.

[LYW+05] Hui Liu, Jie Yang, Meng Wang, Li Xue, and Kuo-Chen Chou. Using Fourier spectrum analysis and pseudo amino acid composition for prediction of membrane protein types. *The Protein Journal*, 24(6):385–389, August 2005.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. Technical Report arXiv:1301.3781, arXiv, September 2013. arXiv:1301.3781 [cs] type: article.

[MCZ14] Nitish K. Mishra, Junil Chang, and Patrick X. Zhao. Prediction of membrane transport proteins and their substrate specificities using primary sequence information. *PLOS ONE*, 9(6):e100278, June 2014.

[MESW+14] Isabel Moraes, Gwyndaf Evans, Juan Sanchez-Weatherby, Simon Newstead, and Patrick D. Shaw Stewart. Membrane protein structure determination — The next generation. *Biochimica et Biophysica Acta (BBA) - Biomembranes*, 1838(1, Part A):78–87, January 2014.

[MKE+19] Vinicius Gonçalves Maltarollo, Thales Kronenberger, Gabriel Zarzana Espinoza, Patricia Rufino Oliveira, and Kathia Maria Honorio. Advances with support vector machines for novel drug discovery. *Expert Opinion on Drug Discovery*, 14:23–33, January 2019.

[MLY17] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in Bioinformatics*, 18(5):851–869, September 2017.

[Mow11] Bernice D. Mowery. The paired t-test. *Pediatric Nursing*, 37(6):320–322, November 2011. Publisher: Jannetti Publications, Inc.

[NC19] Juan J. Nogueira and Ben Corry. Ion Channel Permeation and Selectivity. In Arin Bhattacharjee, editor, *The Oxford Handbook of Neuronal Ion Channels*. Oxford University Press, April 2019.

[NHH00] Cédric Notredame, Desmond G Higgins, and Jaap Heringa. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205–217, September 2000.

[NHTO22] Trinh-Trung-Duong Nguyen, Quang-Thai Ho, Yu-Chun Tarn, and Yu-Yen Ou. MFPS_CNN: Multi-filter pattern scanning from position-specific scoring matrix with convolutional neural network for efficient prediction of ion transporters. *Molecular Informatics*, page e2100271, March 2022.

[NLH+19] Trinh-Trung-Duong Nguyen, Nguyen-Quoc-Khanh Le, Quang-Thai Ho, Dinh-Van Phan, and Yu-Yen Ou. Using word embedding technique to efficiently represent protein sequences for identifying substrate specificities of transporters. *Analytical Biochemistry*, 577:73–81, July 2019.

[OALH06] John P. Overington, Bissan Al-Lazikani, and Andrew L. Hopkins. How many drug targets are there? *Nature Reviews Drug Discovery*, 5(12):993–996, December 2006. Publisher: Nature Publishing Group.

[OBL21] Dan Ofer, Nadav Brandes, and Michal Linial. The language of proteins: NLP, machine learning & protein sequences. *Computational and Structural Biotechnology Journal*, 19:1750–1758, January 2021.

[ON15] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, December 2015. arXiv:1511.08458 [cs].

[PGLL07] Xian Pu, Jian Guo, Howard Leung, and Yuanlie Lin. Prediction of membrane protein types from sequences and position-specific scoring matrices. *Journal of Theoretical Biology*, 247(2):259–265, July 2007.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[PHG+21] Vivitri Dewi Prasasty, Rory Anthony Hutagalung, Reinhart Gunadi, Dewi Yustika Sofia, Rosmalena Rosmalena, Fatmawaty Yazid, and Ernawati Sinaga. Prediction of human-*Streptococcus pneumoniae protein-protein interactions using logistic regression.* Computational Biology and Chemistry*, 92:107492, June 2021.*

[PLD05] Hanchuan Peng, Fuhui Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, August 2005. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[PSR20] Matilda Q. R. Pembury Smith and Graeme D. Ruxton. Effective use of the McNemar test. Behavioral Ecology and Sociobiology*, 74(11):133, October 2020.*

[Qi12] Yanjun Qi. Random Forest for Bioinformatics. In Cha Zhang and Yunqian Ma, editors, Ensemble Machine Learning: Methods and Applications*, pages 307–323. Springer US, Boston, MA, 2012.*

[Qui02] Michael W. Quick, editor. Transmembrane Transporters*. Receptor biochemistry and methodology. Wiley-Liss, New York, 2002.*

[RADVRC10] Iván Restrepo-Angulo, Andrea De Vizcaya-Ruiz, and Javier Camacho. Ion channels in toxicology. Journal of Applied Toxicology*, 30(6):497–512, 2010.*

[RBT+19] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with TAPE. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.*

[RFB22] Ibraheem Rehman, Mustafa Farooq, and Salome Botelho. Biochemistry, Secondary Protein Structure. In StatPearls*. StatPearls Publishing, Treasure Island (FL), 2022.*

[RIV+19] Jüri Reimand, Ruth Isserlin, Veronique Voisin, Mike Kucera, Christian Tannus-Lopes, Asha Rostamianfar, Lina Wadi, Mona Meyer, Jeff Wong, Changjiang Xu, Daniele Merico, and Gary D. Bader. Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap. Nature Protocols*, 14(2):482–517, February 2019. Publisher: Nature Publishing Group.*

[RKL19] Jae Yong Ryu, Hyun Uk Kim, and Sang Yup Lee. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. Proceedings of the National Academy of Sciences*, 116(28):13996–14001, July 2019. Publisher: National Academy of Sciences Section: Biological Sciences.*

[RLV+21] Roshan M. Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA transformer. In Proceedings of the 38th International Conference on Machine Learning*, pages 8844–8856. PMLR, July 2021. ISSN: 2640-3498.*

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. The Annals of Mathematical Statistics*, 22(3):400–407, 1951. Publisher: Institute of Mathematical Statistics.*

[RMS+20] *Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners, December 2020. Pages: 2020.12.15.422761;bioRxiv.*

[RMS+21] *Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences.* Proceedings of the National Academy of Sciences*, 118(15):e2016239118, April 2021. Publisher: Proceedings of the National Academy of Sciences.*

[RSR+20] *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, July 2020. arXiv:1910.10683 [cs, stat].*

[RSS01] *Maido Remm, Christian E. V. Storm, and Erik L. L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons.* Journal of Molecular Biology*, 314(5):1041–1052, December 2001.*

[SBM+16] *Karen S. Sarkisyan, Dmitry A. Bolotin, Margarita V. Meer, Dinara R. Usmanova, Alexander S. Mishin, George V. Sharonov, Dmitry N. Ivankov, Nina G. Bozhanova, Mikhail S. Baranov, Onuralp Soylemez, Natalya S. Bogatyreva, Peter K. Vlasov, Evgeny S. Egorov, Maria D. Logacheva, Alexey S. Kondrashov, Dmitry M. Chudakov, Ekaterina V. Putintseva, Ilgar Z. Mamedov, Dan S. Tawfik, Konstantin A. Lukyanov, and Fyodor A. Kondrashov. Local fitness landscape of the green fluorescent protein.* Nature*, 533(7603):397–401, May 2016. Publisher: Nature Publishing Group.*

[SBP+24] *Amy P.N. Skubitz, Kristin L.M. Boylan, Ashley J. Petersen, Joshua R. Hansen, Tao Liu, Yuqian Gao, Yi-Ting Wang, Karin D. Rodland, Melissa A. Geller, Peter A. Argenta, Samantha L. Hoffman, and Paul D. Piehowski. Abstract B039: Identification of ovarian cancer protein biomarkers in liquid-based Pap tests.* Cancer Research*, 84(5_Supplement_2):B039, March 2024.*

[SC05] *Hongbin Shen and Kuo-Chen Chou. Using optimized evidence-theoretic k-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types.* Biochemical and Biophysical Research Communications*, 334(1):288–292, August 2005.*

[Sch15] *Jürgen Schmidhuber. Deep learning in neural networks: An overview.* Neural Networks*, 61:85–117, January 2015.*

[SDHR21] *Hannes Stärk, Christian Dallago, Michael Heinzinger, and Burkhard Rost. Light attention predicts protein location from the language of life.* Bioinformatics Advances*, 1(1):vbab035, January 2021.*

[SHK+14] *Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting.* The Journal of Machine Learning Research*, 15(1):1929–1958, January 2014.*

[SHZ+24] *Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein language modeling with structure-aware vocabulary, March 2024. Pages: 2023.10.01.560349; bioRxiv.*

[SOPK18] *Seokjun Seo, Minsik Oh, Youngjune Park, and Sun Kim. DeepFam: Deep learning based alignment-free method for protein family modeling and prediction.* Bioinformatics*, 34(13):i254–i262, July 2018.*

[STB06] *Milton H. Saier, Jr, Can V. Tran, and Ravi D. Barabote. TCDB: The transporter classification database for membrane transport protein analyses and information.* Nucleic Acids Research*, 34(suppl_1):D181–D186, January 2006.*

[Sti16] *William Stillwell. Chapter 6 - Membrane Proteins. In William Stillwell, editor,* An Introduction to Biological Membranes (Second Edition)*, pages 89–110. Elsevier, January 2016.*

[STM+05] *Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles.* Proceedings of the National Academy of Sciences*, 102(43):15545–15550, October 2005. Publisher: Proceedings of the National Academy of Sciences.*

[Sto11] *Jill C. Stoltzfus. Logistic regression: A brief primer.* Academic Emergency Medicine*, 18(10):1099–1104, 2011.*

[SW89] *Lars Sthle and Svante Wold. Analysis of variance (ANOVA).* Chemometrics and Intelligent Laboratory Systems*, 6(4):259–272, November 1989.*

[SWH+15] *Baris E. Suzek, Yuqi Wang, Hongzhan Huang, Peter B. McGarvey, Cathy H. Wu, and the UniProt Consortium. UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches.* Bioinformatics*, 31:926–932, March 2015.*

[Sö05] *Johannes Söding. Protein homology detection by HMM–HMM comparison.* Bioinformatics*, 21(7):951–960, April 2005.*

[The21] *The UniProt Consortium. UniProt: The universal protein knowledgebase in 2021.* Nucleic Acids Research*, 49(D1):D480–D489, January 2021.*

[TLZ+24] *Yang Tan, Mingchen Li, Bingxin Zhou, Bozitao Zhong, Lirong Zheng, Pan Tan, Ziyi Zhou, Huiqun Yu, Guisheng Fan, and Liang Hong. Simple, efficient and scalable structure-aware adapter boosts protein language models, April 2024. arXiv:2404.14850 [cs, q-bio].*

[TM16] *Juliana Tolles and William J. Meurer. Logistic regression: relating patient characteristics to outcomes.* JAMA*, 316(5):533–534, August 2016.*

[TO19] *Semmy Wellem Taju and Yu-Yen Ou. Deeplon: Deep learning approach for classifying ion transporters and ion channels from membrane proteins.* Journal of Computational Chemistry*, 40(15):1521–1529, 2019.*

[TPS+15] *Konstantinos D. Tsirigos, Christoph Peters, Nanjiang Shu, Lukas Käll, and Arne Elofsson. The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides.* Nucleic Acids Research*, 43(W1):W401–W407, July 2015.*

[VdMH08] *Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE.* Journal of Machine Learning Research*, 9(11), 2008.*

[vKKT+23] *Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek.* Nature Biotechnology*, pages 1–4, May 2023. Publisher: Nature Publishing Group.*

[VMV+21] *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:2006.15222 [cs, q-bio]*, March 2021.*

[VSP+17] *Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need.* arXiv*, December 2017.*

[WDS+20] *Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's transformers: State-of-the-art natural language processing.* arXiv*, July 2020.*

[WH98] *Erik Wallin and Gunnar Von Heijne. Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms.* Protein Science*, 7(4):1029–1038, 1998.*

[WLW+12] *Jingyan Wang, Yongping Li, Quanquan Wang, Xinge You, Jiaju Man, Chao Wang, and Xin Gao. ProClusEnsem: Predicting membrane protein types by fusing different modes of pseudo amino acid composition.* Computers in Biology and Medicine*, 42(5):564–574, May 2012.*

[WP99] *Todd C Wood and William R Pearson. Evolution of protein sequences and structures.* Journal of Molecular Biology*, 291(4):977–995, August 1999.*

[WPA+24] *Duolin Wang, Mahdi Pourmirzaei, Usman L. Abbas, Shuai Zeng, Negin Manshour, Farzaneh Esmaili, Biplab Poudel, Yuexu Jiang, Qing Shao, Jin Chen, and Dong Xu. S-PLM: Structure-aware protein language model via contrastive learning between sequence and structure, May 2024. Pages: 2023.08.06.552203; bioRxiv.*

[WSC+16] *Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, October 2016. arXiv:1609.08144 [cs].*

[WSY+24] *Wenkang Wang, Yunyan Shuai, Qiurong Yang, Fuhao Zhang, Min Zeng, and Min Li. A comprehensive computational benchmark for evaluating deep learning-based protein function prediction approaches.* Briefings in Bioinformatics*, 25(2):bbae050, March 2024.*

[WW99] *Stephen H. White and William C. Wimley. Membrane protein folding and stability: Physical Principles.* Annual Review of Biophysics*, 28(Volume 28, 1999):319–365, June 1999. Publisher: Annual Reviews.*

[WYL+04] *Meng Wang, Jie Yang, Guo-Ping Liu, Zhi-Jie Xu, and Kuo-Chen Chou. Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition.* Protein Engineering, Design and Selection*, 17(6):509–516, June 2004.*

[WYXC05] *Meng Wang, Jie Yang, Zhi-Jie Xu, and Kuo-Chen Chou. SLLE for predicting membrane protein types.* Journal of Theoretical Biology*, 232(1):7–15, January 2005.*

[WZL+20] *Bo Wen, Wen-Feng Zeng, Yuxing Liao, Zhiao Shi, Sara R. Savage, Wen Jiang, and Bing Zhang. Deep learning in proteomics.* Proteomics*, 20(21-22):1900335, 2020.*

[XSZ+19] *Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and Improving Layer Normalization. In* Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.*

[XZL+22] *Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Chang Ma, Runcheng Liu, and Jian Tang. PEER: A comprehensive and multi-task benchmark for protein sequence understanding, September 2022. arXiv:2206.02096 [cs].*

[Yea16] *Philip L. Yeagle. Chapter 10 - Membrane Proteins. In Philip L. Yeagle, editor,* The Membranes of Cells (Third Edition)*, pages 219–268. Academic Press, Boston, January 2016.*

[YGC+07] *Muhammed A. Yıldırım, Kwang-Il Goh, Michael E. Cusick, Albert-László Barabási, and Marc Vidal. Drug—target network.* Nature Biotechnology*, 25(10):1119–1126, October 2007. Publisher: Nature Publishing Group.*

[YGW+18] *Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch?* Briefings in Bioinformatics*, 19(3):482–494, May 2018.*

[ZBC+22] *Yue Zhang, Wenzheng Bao, Yi Cao, Hanhan Cong, Baitong Chen, and Yuehui Chen. A survey on protein–DNA-binding sites in computational biology.* Briefings in Functional Genomics*, page elac009, June 2022.*

[ZHS+23] *Zhongliang Zhou, Mengxuan Hu, Mariah Salcedo, Nathan Gravel, Wayland Yeung, Aarya Venkat, Dongliang Guo, Jielu Zhang, Natarajan Kannan, and Sheng Li. XAI meets biology: A comprehensive review of explainable AI in bioinformatics applications, December 2023. arXiv:2312.06082 [cs, q-bio].*

Appendices

Appendix A

Research Outputs and Publications During Doctoral Studies

1. Ghazikhani, H., & Butler, G. (2022). TooT-BERT-M: Discriminating Membrane Proteins from Non-Membrane Proteins using a BERT Representation of Protein Primary Sequences. 2022 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Ottawa, Canada, 1–8.

2. Ghazikhani, H., & Butler, G. (2023). TooT-BERT-T: A BERT Approach on Discriminating Transport Proteins from Non-transport Proteins. In F. Fdez-Riverola, M. Rocha, M. S. Mohamad, S. Caraiman, & A. B. Gil-González (Eds.), Practical Applications of Computational Biology and Bioinformatics, 16th International Conference (PACBB 2022) (pp. 1–11). Springer International Publishing. L'Aquila, Italy.

3. Ghazikhani, H., & Butler, G. (2023). TooT-BERT-C: A study on discriminating ion channels from membrane proteins based on the primary sequence's contextual representation from BERT models. Proceedings of the 9th International Conference on Bioinformatics Research and Applications, 23–29. Berlin, Germany.

4. Ghazikhani, H., & Butler, G. (2023). A Study on the Application of Protein Language Models in the Analysis of Membrane Proteins. In J. M. Machado, P. Chamoso, G. Hernández, G. Bocewicz, R. Loukanova, E. Jove, A. M. del Rey, & M. Ricca (Eds.), Distributed Computing and Artificial Intelligence, Special Sessions, 19th International Conference (pp. 147–152). Springer International Publishing. L'Aquila, Italy.

5. Ghazikhani, H., & Butler, G. (2023). Enhanced identification of membrane transport proteins: A hybrid approach combining ProtBERT-BFD and convolutional neural networks. Journal of Integrative Bioinformatics, 20(2).

6. Ghazikhani, H., & Butler, G. (2024). Ion Channel Classification Through Machine Learning and Protein Language Model Embeddings. Journal of Integrative Bioinformatics, (under review).

7. Ghazikhani, H., & Butler, G. (2024). Enhancing Membrane Protein Identification with TooT-BERT-CNN-M: Leveraging ProtBERT and Convolutional Neural Networks. Journal of Integrative Bioinformatics, (under review).

8. Ghazikhani, H., & Butler, G. (2024). Exploiting protein language models for the precise classification of ion channels and ion transporters. Proteins: Structure, Function, and Bioinformatics, 92(8), 998–1055.

9. Ghazikhani, H., & Butler, G. (2024). Integrating Secondary Structure into Protein Language Models for Enhanced General and Membrane Protein Predictions. 32nd Intelligent Systems For Molecular Biology (ISMB 2024) - Poster.

10. Ghazikhani, H., & Butler, G. (2024). TooT-PLM-P2S: Incorporating Secondary Structure Information into Protein Language Models. Journal of Cellular Biochemistry, (under review).

Appendix B

Exploiting Protein Language Models for the Precise Classification of Ion Channels and Ion Transporters

B.1 Frozen vs Fine-tuned Representations

Table 54: Frozen vs fine-tuned representations across protein language models.

PLM	Representation	MCC	Accuracy	Sensitivity	Specificity	P-value
ProtBERT	frozen	0.69±0.05	94.14±1.48	93.80±2.65	94.17±2.42	2.78e-06
	finetuned	0.78±0.04	92.94±1.41	82.16±4.20	93.76±2.46	
ProtBERT-BFD	frozen	0.70±0.05	93.35±1.40	90.32±3.55	93.62±2.46	1.40e-06
	finetuned	0.79±0.05	92.44±1.57	80.35±4.81	93.36±2.58	
ESM-1b	frozen	0.79±0.04	92.36±1.41	81.36±3.99	92.58±2.38	5.31e-07
	finetuned	0.88±0.03	91.09±1.51	83.56±4.47	91.45±2.84	
ESM-2	frozen	0.77±0.05	90.04±1.69	74.04±4.94	91.01±3.08	5.40e-07
	finetuned	0.85±0.04	91.34±1.74	84.63±4.56	91.97±2.80	
ProtT5	frozen	0.78±0.04	90.19±1.86	74.76±5.17	91.41±3.02	None
ESM-2_15B	frozen	0.77±0.04	92.73±1.37	81.09±4.29	93.67±2.27	None

This table presents a comparison and evaluation of frozen versus fine-tuned representations across a range of protein language models (PLMs). The assessment is based on four evaluation metrics computed using a 5-fold cross-validation procedure and is presented as the mean ± standard deviation. Statistical significance of observed discrepancies among the models is denoted by the provided p-value. Please note, instances of 'None' indicate that due to resource constraints, we were unable to fine-tune larger PLMs such as ProtT5 and ESM-2 with 15 billion parameters.

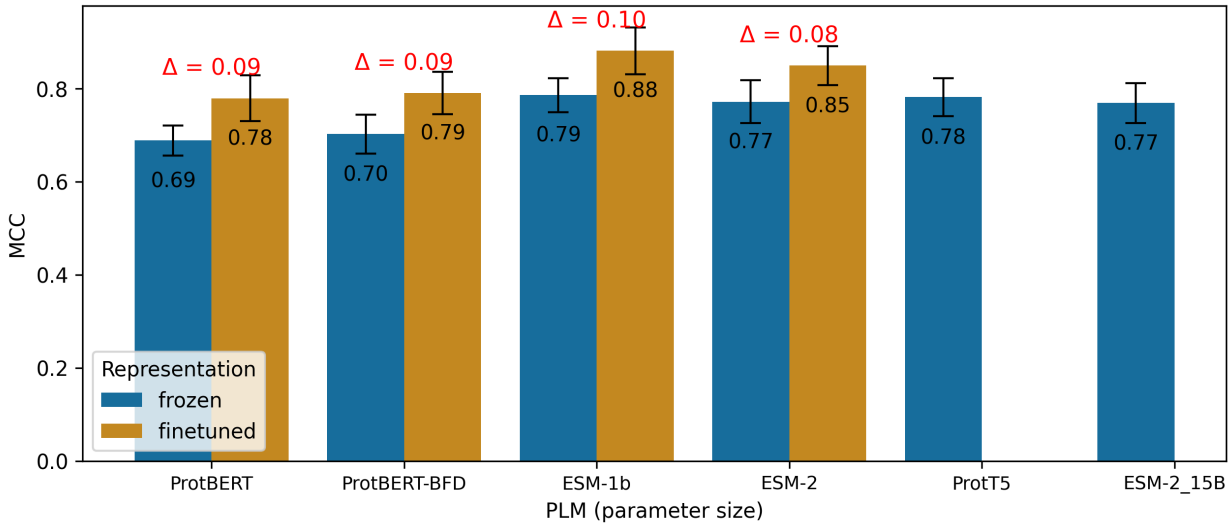

Figure 39: Differential impact of frozen and fine-tuned on various PLMs

This figure provides a graphical display of the differential impact of employing frozen and fine-tuned representations across various Protein Language Models (PLMs). The comparison is made using the mean Matthew's Correlation Coefficient (MCC) values, as determined from 5-fold cross-validation. Each bar signifies the mean MCC obtained across the cross-validation sets, with error bars representing the standard deviation. The delta symbol (Δ) illustrates the difference between the associated pair of bars. Absent bars denote the inability to fine-tune large PLMs such as ProtT5 and ESM-2, each containing 15 billion parameters, due to resource limitations.

Table 55: Frozen vs fine-tuned representations across classifiers.

Classifier	Representation	MCC	Accuracy	Sensitivity	Specificity	P-value
LR	frozen	0.79±0.04	93.94±1.53	89.42±3.84	95.08±2.97	2.56e-05
	finetuned	0.84±0.04	93.32±1.60	82.21±5.58	94.95±3.16	
kNN	frozen	0.65±0.05	93.62±1.47	89.93±3.49	94.01±2.49	1.13e-05
	finetuned	0.75±0.05	93.09±1.48	84.53±4.24	93.81±2.49	
SVM	frozen	0.81±0.04	93.35±1.36	89.28±3.35	93.94±2.34	2.67e-05
	finetuned	0.85±0.03	92.88±1.39	83.18±4.03	93.77±2.38	
RF	frozen	0.61±0.05	91.74±1.61	82.22±4.70	94.43±2.66	3.02e-06
	finetuned	0.80±0.04	90.03±1.67	57.67±4.94	94.58±2.69	
FFNN	frozen	0.80±0.04	93.73±1.25	89.67±3.48	94.69±2.21	3.97e-05
	finetuned	0.85±0.04	93.36±1.28	83.69±4.39	94.74±2.15	
CNN	frozen	0.81±0.04	88.47±1.97	87.94±3.98	84.67±3.11	4.68e-06
	finetuned	0.86±0.04	87.10±2.09	79.74±4.74	83.35±3.28	

This table presents a comparison and evaluation of frozen versus fine-tuned representations across a range of classifiers. The assessment is based on four evaluation metrics computed using a 5-fold cross-validation procedure and is presented as the mean ± standard deviation. Statistical significance of observed discrepancies among the models is denoted by the provided p-value.

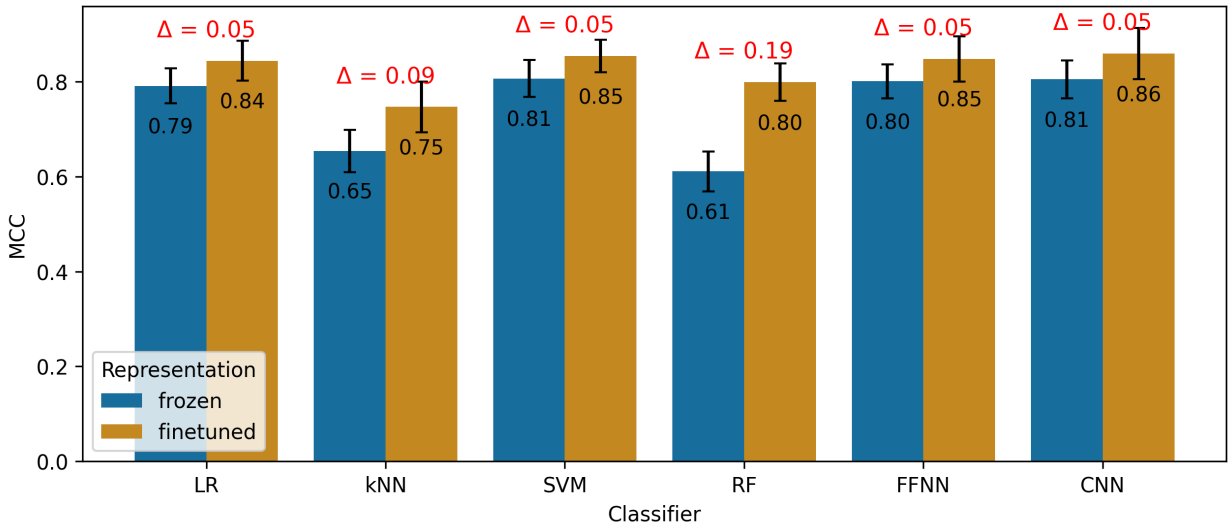

Figure 40: Differential impact of frozen and fine-tuned on various classifiers

This figure provides a graphical display of the differential impact of employing frozen and fine-tuned representations across various classifiers. The comparison is made using the mean Matthew's Correlation Coefficient (MCC) values, as determined from 5-fold cross-validation. Each bar signifies the mean MCC obtained across the cross-validation sets, with error bars representing the standard deviation. The delta symbol (Δ) illustrates the difference between the associated pair of bars.

B.2 Balanced vs Imbalanced Datasets

Table 56: Balanced vs imbalanced dataset performance across tasks.

Task	Dataset	MCC	Accuracy	Sensitivity	Specificity	P-value
IC-MP	balanced	0.76±0.05	87.47±2.73	88.10±4.03	86.88±4.42	5.50e-04
	imbalanced	0.81±0.03	97.89±0.16	75.21±4.80	99.58±0.09	
IT-MP	balanced	0.74±0.05	86.77±2.70	87.07±3.67	86.50±4.32	1.44e-02
	imbalanced	0.78±0.03	97.25±0.13	72.99±4.91	99.36±0.16	

This table presents a comparison and evaluation of balanced versus imbalanced dataset peformance across the tasks of ion channels vs other membrane proteins (MP) and ion transporters vs MP. The assessment is based on four evaluation metrics computed using a 5-fold cross-validation procedure and is presented as the mean ± standard deviation. Statistical significance of observed discrepancies among the models is denoted by the provided p-value.

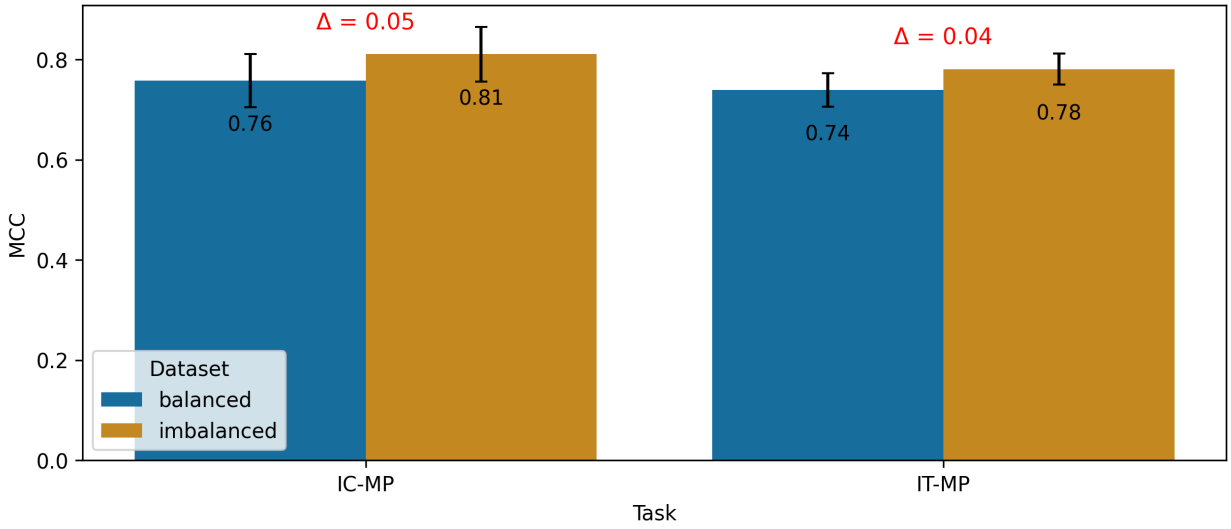

Figure 41: Differential impact of balanced and imbalanced dataset

This figure provides a graphical display of the differential impact of employing balanced and imbalanced dataset across various tasks of ion channels (IC) vs other membrane proteins (MP) and ion transporters (IT) vs MP. The comparison is made using the mean Matthew's Correlation Coefficient (MCC) values, as determined from 5-fold cross-validation. Each bar signifies the mean MCC obtained across the cross-validation sets, with error bars representing the standard deviation. The delta symbol (Δ) illustrates the difference between the associated pair of bars.

Table 57: Balanced vs imbalanced dataset across fine-tuned PLMs

PLM	Dataset	MCC	Accuracy	Sensitivity	Specificity	P-value
ProtBERT	balanced	0.71±0.06	89.21±2.40	89.00±3.33	89.39±3.89	1.55e-08
	imbalanced	**0.85±0.03**	**100.00±0.00**	**99.12±1.29**	**100.00±0.00**	
ProtBERT-BFD	balanced	0.71±0.06	88.57±2.42	88.96±3.66	88.26±4.03	3.27e-10
	imbalanced	**0.87±0.03**	**99.04±0.04**	**91.17±3.50**	**99.88±0.04**	
ESM-1b	balanced	0.79±0.05	84.96±2.86	**85.95±4.00**	84.05±4.93	1.43e-14
	imbalanced	**0.99±0.01**	**98.00±0.21**	78.29±5.17	**99.83±0.04**	
ESM-2	balanced	0.78±0.05	85.48±3.18	**85.66±4.57**	85.41±4.66	6.96e-10
	imbalanced	**0.93±0.02**	**98.42±0.00**	81.58±4.46	**99.92±0.04**	

This table presents a comparison and evaluation of balanced versus imbalanced dataset peformance across the fine-tuned protein language models. The assessment is based on four evaluation metrics computed using a 5-fold cross-validation procedure and is presented as the mean ± standard deviation. Statistical significance of observed discrepancies among the models is denoted by the provided p-value. Bold values indicate metrics where the p-value is less than 0.05, showing statistical significance.

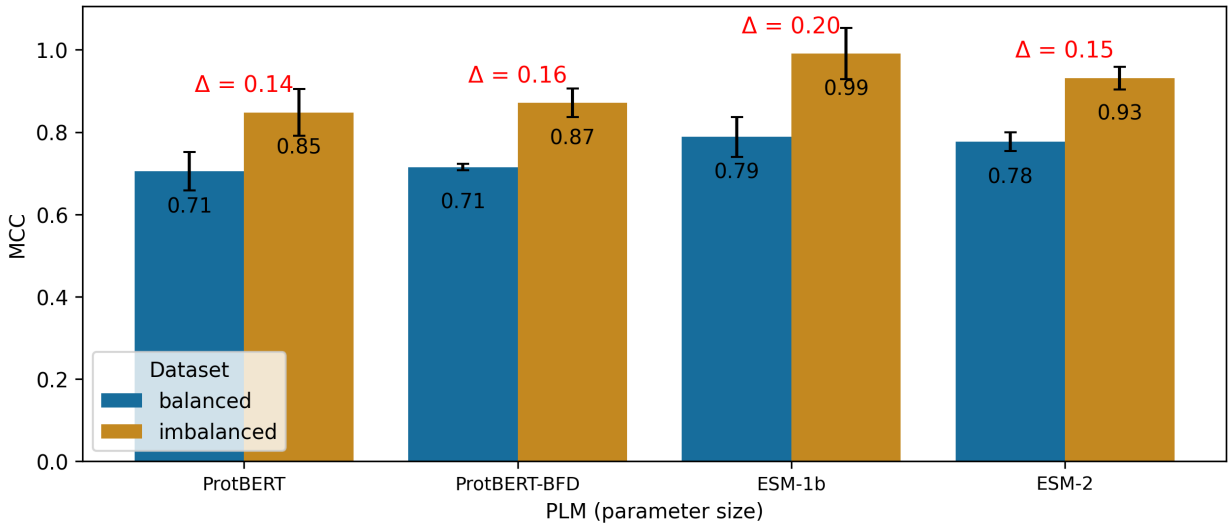

Figure 42: Balanced vs imbalanced dataset performance across fine-tuned PLMs.

This figure provides a graphical display of the differential impact of employing balanced and imbalanced datasets across various fine-tuned Protein Language Models (PLMs). The comparison is made using the mean Matthew's Correlation Coefficient (MCC) values, as determined from 5-fold cross-validation. Each bar signifies the mean MCC obtained across the cross-validation sets, with error bars representing the standard deviation. The delta symbol (Δ) illustrates the difference between the associated pair of bars.

Table 58: Balanced vs imbalanced dataset across frozen PLMs

PLM	Dataset	MCC	Accuracy	Sensitivity	Specificity	P-value
ProtBERT	balanced	0.70±0.06	89.07±2.45	89.01±3.34	89.15±3.89	8.66e-02
	imbalanced	0.63±0.04	96.96±0.12	69.92±5.75	99.17±0.21	
ProtBERT-BFD	balanced	0.71±0.06	88.52±2.47	88.91±3.56	88.22±4.20	1.34e-01
	imbalanced	0.66±0.04	96.92±0.33	66.42±6.54	99.25±0.12	
ESM-1b	balanced	0.79±0.05	87.83±2.52	89.81±3.38	85.87±3.78	2.34e-01
	imbalanced	0.75±0.04	96.92±0.17	67.83±5.25	99.17±0.25	
ESM-2	balanced	0.78±0.05	84.67±3.02	85.71±4.28	83.70±5.16	2.46e-01
	imbalanced	0.74±0.04	95.83±0.25	55.12±5.71	98.96±0.21	
ProtT5	balanced	0.79±0.05	85.14±3.20	85.52±4.52	84.73±4.87	4.33e-01
	imbalanced	0.75±0.04	96.08±0.17	57.42±5.67	99.08±0.29	
ESM-2_15B	balanced	0.77±0.05	89.08±2.35	89.32±3.48	88.77±3.67	6.05e-01
	imbalanced	0.73±0.03	96.83±0.17	67.92±5.92	99.17±0.08	

This table presents a comparison and evaluation of balanced versus imbalanced dataset peformance across the frozen protein language models. The assessment is based on four evaluation metrics computed using a 5-fold cross-validation procedure and is presented as the mean ± standard deviation. Statistical significance of observed discrepancies among the models is denoted by the provided p-value.

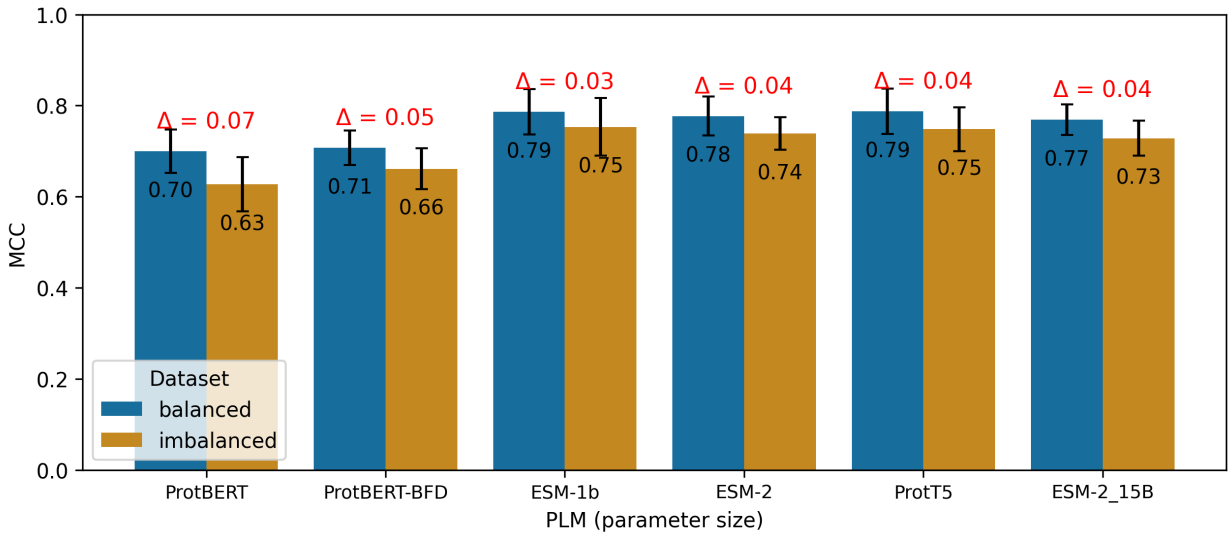

Figure 43: Balanced vs imbalanced dataset performance across frozen PLMs.

This figure provides a graphical display of the differential impact of employing balanced and imbalanced dataset across various frozen Protein Language Models (PLMs). The comparison is made using the mean Matthew's Correlation Coefficient (MCC) values, as determined from 5-fold cross-validation. Each bar signifies the mean MCC obtained across the cross-validation sets, with error bars representing the standard deviation. The delta symbol (Δ) illustrates the difference between the associated pair of bars.

Table 59: Balanced vs imbalanced dataset performance across classifiers.

Classifier	Dataset	MCC	Accuracy	Sensitivity	Specificity	P-value
LR	balanced	0.78±0.05	89.49±2.81	87.85±4.74	91.07±5.68	5.91e-04
	imbalanced	0.84±0.03	98.17±0.28	79.89±5.69	99.56±0.31	
kNN	balanced	0.60±0.06	89.32±2.55	89.38±3.31	89.27±4.17	1.99e-07
	imbalanced	0.77±0.03	97.97±0.06	81.89±4.67	99.42±0.08	
SVM	balanced	0.79±0.05	89.14±2.53	88.71±3.35	89.47±4.18	1.83e-04
	imbalanced	0.85±0.03	97.97±0.11	80.53±4.42	99.42±0.00	
RF	balanced	0.73±0.06	86.14±2.92	81.34±4.81	90.43±3.66	1.13e-02
	imbalanced	0.62±0.04	96.19±0.08	46.97±4.25	100.00±0.00	
FFNN	balanced	0.79±0.05	89.60±2.31	88.34±3.21	90.76±3.56	1.28e-04
	imbalanced	0.85±0.03	98.08±0.03	81.69±4.94	99.53±0.11	
CNN	balanced	0.80±0.05	79.03±3.17	89.90±3.69	69.14±4.99	8.20e-04
	imbalanced	0.85±0.03	97.03±0.31	73.64±5.14	98.92±0.25	

This table presents a comparison and evaluation of balanced versus imbalanced dataset peformance across classifiers. The assessment is based on four evaluation metrics computed using a 5-fold cross-validation procedure and is presented as the mean ± standard deviation. Statistical significance of observed discrepancies among the models is denoted by the provided p-value.

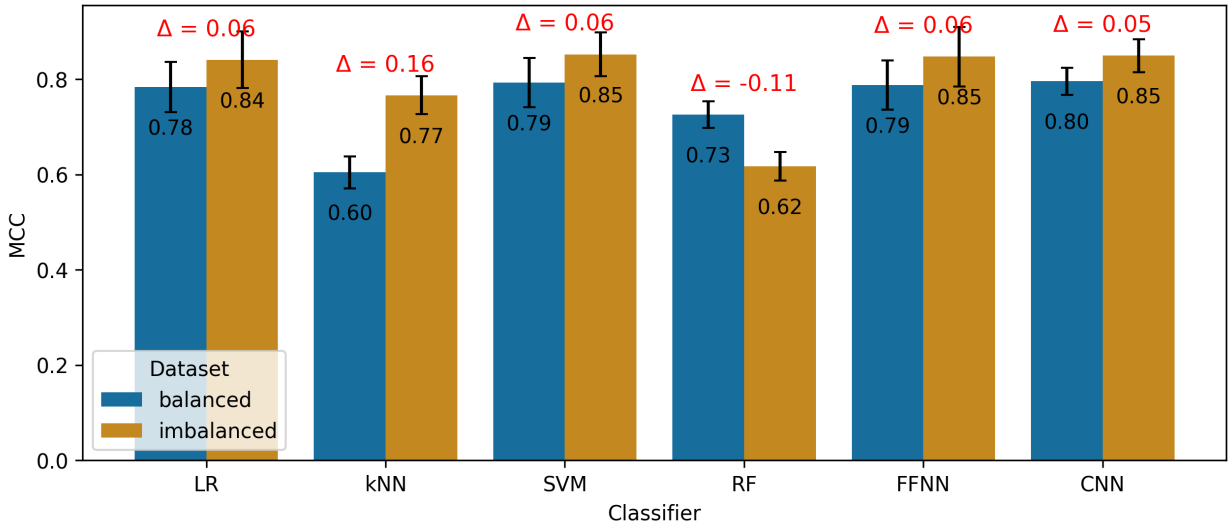

Figure 44: Balanced vs imbalanced dataset performance across classifiers.

This figure provides a graphical display of the differential impact of employing balanced and imbalanced dataset across various classifiers. The comparison is made using the mean Matthew's Correlation Coefficient (MCC) values, as determined from 5-fold cross-validation. Each bar signifies the mean MCC obtained across the cross-validation sets, with error bars representing the standard deviation. The delta symbol (Δ) illustrates the difference between the associated pair of bars.

B.3 Half vs Full Precision Floating Point Calculations

Table 60: Half vs full precision floating point calculations across tasks.

Task	Precision	MCC	Accuracy	Sensitivity	Specificity	P-value
IC-MP	half	*0.78±0.04*	*90.46±2.17*	*90.21±4.19*	*90.73±4.44*	9.75e-01
	full	*0.78±0.04*	*90.82±2.03*	*90.58±3.80*	*91.10±4.23*	
IT-MP	half	*0.76±0.04*	*92.65±1.46*	*81.89±4.43*	*93.18±2.29*	7.48e-01
	full	*0.76±0.04*	*92.71±1.42*	*81.47±4.40*	*93.27±2.23*	
IC-IT	half	*0.82±0.04*	*92.03±1.45*	*80.62±4.40*	*92.99±2.25*	9.34e-01
	full	*0.81±0.04*	*92.00±1.39*	*79.56±4.20*	*92.88±2.23*	

This table presents a comparison and evaluation of half versus full precision floating-point across the tasks of ion channels (IC) vs other membrane proteins (MP), ion transporter (IT) vs MP, and IC vs IT. The assessment is based on four evaluation metrics computed using a 5-fold cross-validation procedure and is presented as the mean ± standard deviation. Statistical significance of observed discrepancies among the models is denoted by the provided p-value. Italic values indicate metrics where the p-value is above the 0.05 threshold, suggesting no statistical significance.

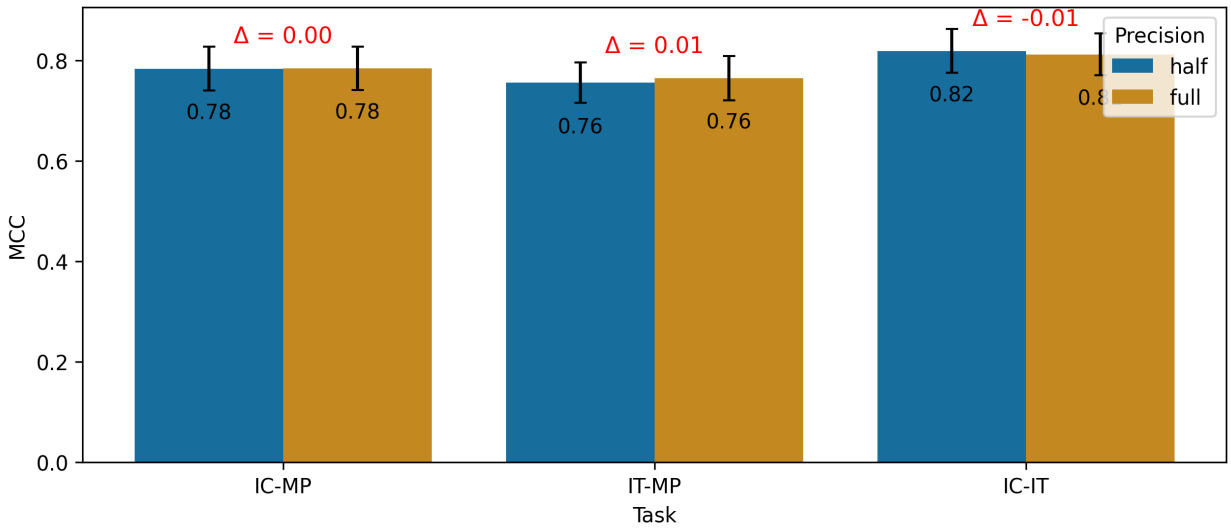

Figure 45: Half vs full precision floating point calculations across tasks.

This figure provides a graphical display of the differential impact of employing half and full precision floating-point calculation across various tasks of ion channels (IC) vs other membrane proteins (MP), ion transporters (IT) vs MP and IC vs IT. The comparison is made using the mean Matthew's Correlation Coefficient (MCC) values, as determined from 5-fold cross-validation. Each bar signifies the mean MCC obtained across the cross-validation sets, with error bars representing the standard deviation. The delta symbol (Δ) illustrates the difference between the associated pair of bars.

Table 61: Half vs full precision floating point calculations across PLMs

PLM	Precision	MCC	Accuracy	Sensitivity	Specificity	P-value
ProtBERT	half	*0.73±0.04*	*93.52±1.45*	*87.91±3.57*	*93.98±2.46*	7.41e-01
	full	*0.74±0.05*	*93.56±1.44*	*88.05±3.28*	*93.96±2.42*	
ProtBERT-BFD	half	*0.75±0.05*	*92.94±1.48*	*85.44±4.09*	*93.55±2.46*	9.59e-01
	full	*0.75±0.05*	*92.85±1.49*	*85.23±4.26*	*93.43±2.58*	
ESM-1b	half	*0.83±0.04*	*92.36±1.41*	*81.36±3.99*	*92.58±2.38*	9.13e-01
	full	*0.83±0.04*	*90.60±1.65*	*79.20±4.87*	*91.23±2.98*	
ESM-2	half	*0.81±0.04*	*90.52±1.55*	*78.39±4.54*	*91.24±2.94*	8.09e-01
	full	*0.81±0.04*	*90.78±1.80*	*79.65±4.95*	*91.71±2.92*	
ProtT5	half	0.78±0.04	90.75±1.80	79.74±4.78	91.67±2.89	None
ESM-2_15B	half	0.77±0.04	92.73±1.37	81.09±4.29	93.67±2.27	None

This table presents a comparison and evaluation of half versus full precision floating-point across protein language models (PLMs). The assessment is based on four evaluation metrics computed using a 5-fold cross-validation procedure and is presented as the mean ± standard deviation. Statistical significance of observed discrepancies among the models is denoted by the provided p-value. Please note, instances of 'None' indicate that due to resource constraints, we were unable to fine-tune larger PLMs such as ProtT5 and ESM-2 with 15 billion parameters. Italic values indicate metrics where the p-value is above the 0.05 threshold, suggesting no statistical significance.

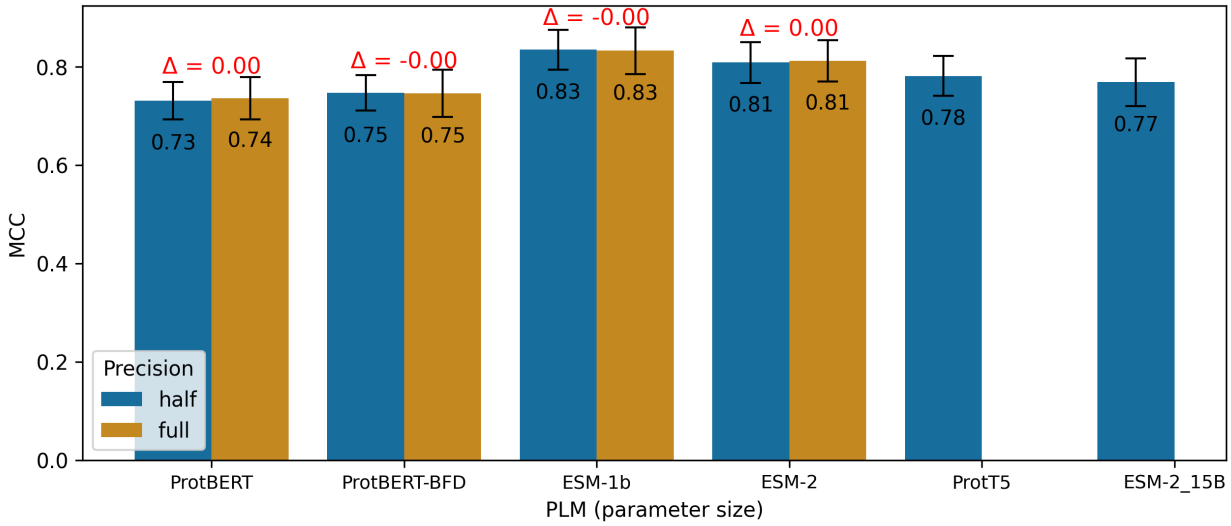

Figure 46: Half vs full precision floating point calculations across PLMs.

This figure provides a graphical display of the differential impact of employing half and full precision floating-point calculation across various Protein Language Models (PLMs). The comparison is made using the mean Matthew's Correlation Coefficient (MCC) values, as determined from 5-fold cross-validation. Each bar signifies the mean MCC obtained across the cross-validation sets, with error bars representing the standard deviation. The delta symbol (Δ) illustrates the difference between the associated pair of bars. Absent bars denote the inability to fine-tune large PLMs such as ProtT5 and ESM-2, each containing 15 billion parameters, due to resource limitations.

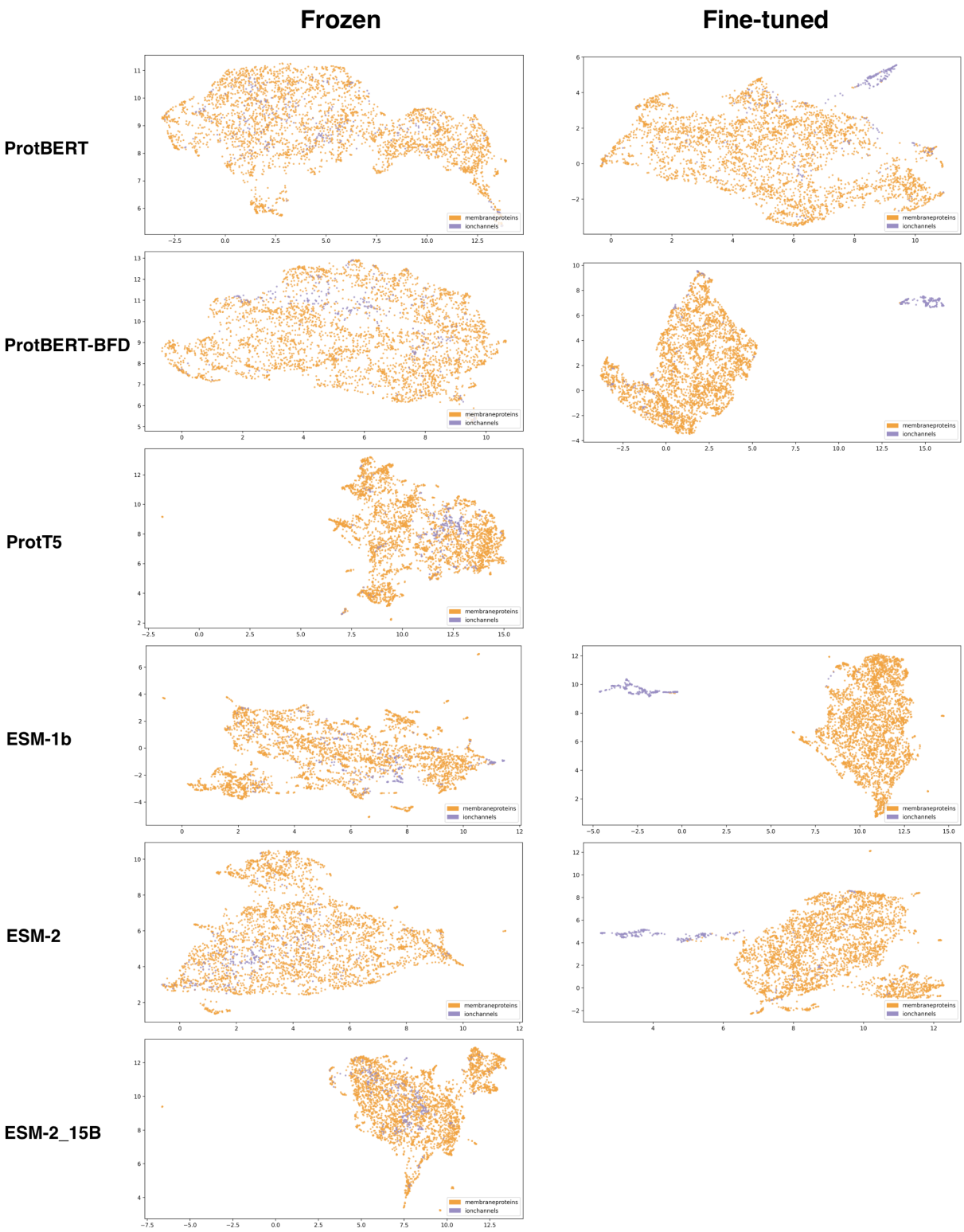

Figure 47: UMAP projection visualizing IC-MP

This figure illustrates a UMAP projection visualizing the separation of ion channels and an imbalanced dataset of other membrane proteins. The visualization encompasses all six protein language models and includes both frozen and fine-tuned representation types. Membrane proteins are represented by yellow points, while ion channels are depicted in blue. The x and y axes correspond to the first and second UMAP dimensions, respectively.

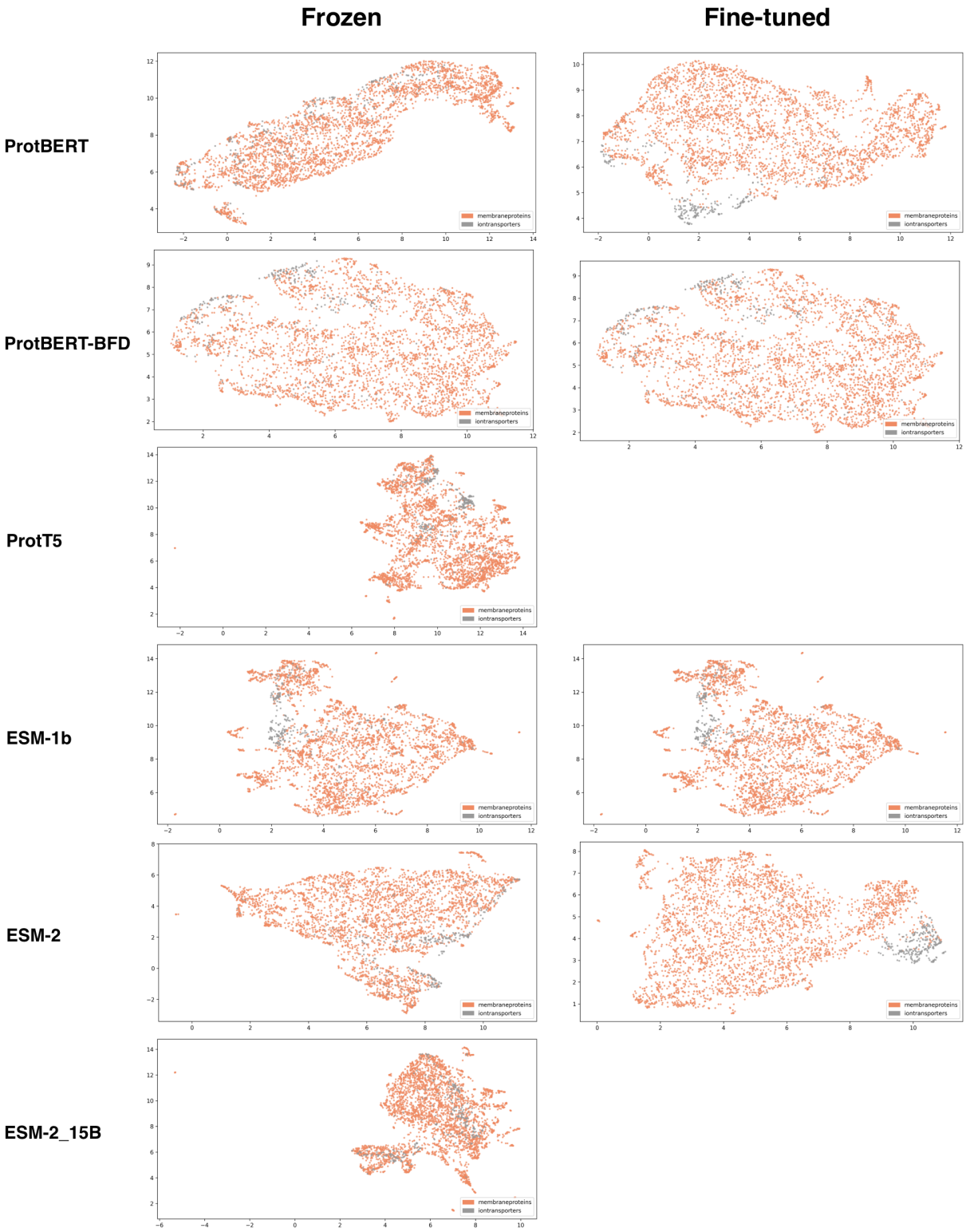

Figure 48: UMAP projection visualizing IT-MP

This figure illustrates a UMAP projection visualizing the separation of ion transporters and an imbalanced dataset of other membrane proteins. The visualization encompasses all six protein language models and includes both frozen and fine-tuned representation types. Membrane proteins are represented by red points, while ion transporters are depicted in grey. The x and y axes correspond to the first and second UMAP dimensions, respectively.

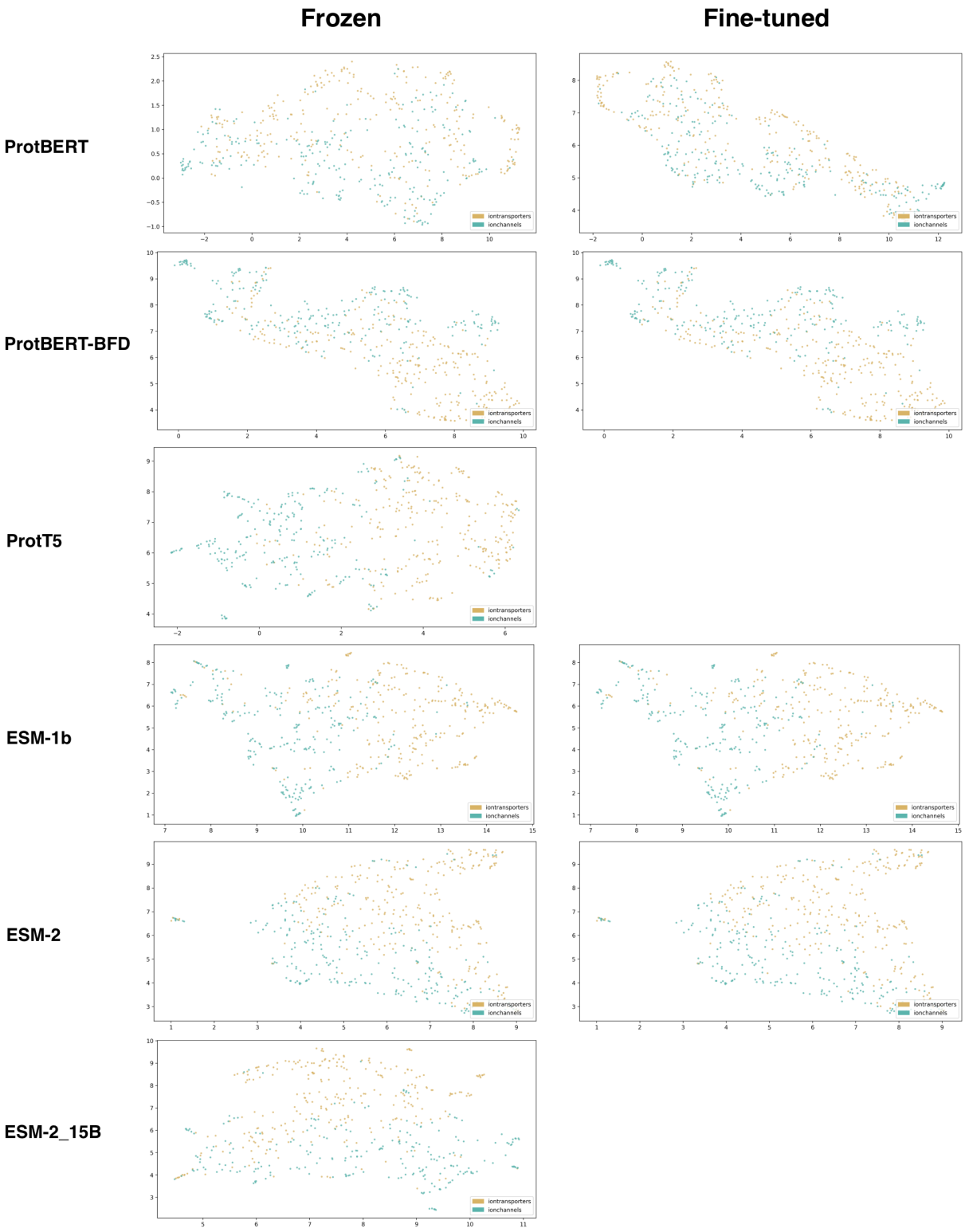

Figure 49: UMAP projection visualizing IC-IT

This figure illustrates a UMAP projection visualizing the separation of ion channels and ion transporters. The visualization encompasses all six protein language models and includes both frozen and fine-tuned representation types. Ion channels are represented by yellow points, while ion transporters are depicted in green. The x and y axes correspond to the first and second UMAP dimensions, respectively.

Table 62: accuracy Comparison of representations for IC-MP

Representer	Representation	Dataset	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-2_15B	frozen	imbalanced	half	99.00±0.00	99.00±1.00	95.00±0.00	96.00±0.00	99.00±0.00	99.00±0.00
		balanced	half	92.20±2.20	93.10±1.90	89.80±2.30	68.90±3.20	93.50±2.00	93.40±2.20
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ProtBERT	frozen	imbalanced	half	98.00±0.00	97.00±0.00	94.00±0.00	95.00±1.00	98.00±0.00	97.00±0.00
		balanced	half	86.60±4.60	87.80±3.10	86.20±2.40	72.50±4.10	86.70±3.40	87.60±2.70
		imbalanced	full	98.00±1.00	97.00±0.00	94.00±0.00	95.00±1.00	98.00±0.00	97.00±0.00
		balanced	full	86.78±3.56	87.70±3.10	86.30±2.70	72.50±4.20	86.70±3.40	87.50±2.60
	finetuned	imbalanced	half	98.00±0.00	98.00±0.00	97.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00
		balanced	half	86.90±3.60	87.70±2.90	86.50±2.60	73.20±3.90	87.30±2.80	87.70±2.50
		imbalanced	full	99.00±1.00	99.00±0.00	98.00±1.00	98.00±0.00	98.00±1.00	98.00±1.00
		balanced	full	86.50±3.80	87.80±3.20	86.10±2.50	72.90±4.30	87.30±2.80	87.70±2.70
ESM-2	frozen	imbalanced	half	99.00±1.00	99.00±0.00	95.00±0.00	97.00±1.00	98.00±0.00	98.00±0.00
		balanced	half	91.00±3.40	92.40±1.70	88.10±3.00	80.50±2.90	91.80±2.00	91.90±2.00
		imbalanced	full	99.00±1.00	99.00±0.00	95.00±0.00	97.00±1.00	98.00±0.00	98.00±0.00
		balanced	full	91.90±3.00	92.40±1.70	88.00±2.90	80.30±2.80	91.90±2.00	92.00±1.90
	finetuned	imbalanced	half	100.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	half	91.60±2.80	92.30±2.00	88.00±3.00	80.40±2.40	91.80±2.00	91.70±2.00
		imbalanced	full	100.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	full	91.90±2.50	92.30±1.80	88.20±2.90	80.50±2.60	91.90±2.00	92.00±1.80
ESM-1b	frozen	imbalanced	half	98.00±1.00	99.00±0.00	96.00±0.00	97.00±0.00	98.00±0.00	98.00±0.00
		balanced	half	90.40±4.00	92.80±1.80	88.50±2.70	80.70±2.70	92.00±1.70	91.90±1.80
		imbalanced	full	98.00±0.00	99.00±0.00	96.00±0.00	97.00±0.00	98.00±0.00	98.00±0.00
		balanced	full	91.00±2.30	92.80±1.80	88.70±2.60	80.70±2.70	91.90±1.80	91.90±1.80
	finetuned	imbalanced	half	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	half	90.70±3.90	92.80±1.70	88.30±2.70	81.20±2.80	91.80±1.70	91.70±1.80
		imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	91.20±2.50	92.80±1.50	88.50±2.50	81.40±2.70	91.80±1.80	91.70±1.90
ProtT5	frozen	imbalanced	half	98.00±1.00	98.00±0.00	95.00±0.00	97.00±1.00	98.00±0.00	98.00±0.00
		balanced	half	91.00±2.90	92.00±2.20	88.80±2.30	80.10±3.10	90.70±1.80	90.90±2.30
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ProtBERT-BFD	frozen	imbalanced	half	97.00±1.00	97.00±0.00	94.00±0.00	96.00±0.00	97.00±0.00	97.00±0.00
		balanced	half	87.50±3.80	88.30±2.20	86.30±2.90	77.60±3.20	86.40±3.60	87.40±2.90
		imbalanced	full	97.00±1.00	97.00±0.00	94.00±0.00	96.00±0.00	97.00±0.00	97.00±0.00
		balanced	full	86.20±4.30	88.30±2.20	86.30±3.30	77.60±3.10	86.70±3.50	87.50±3.10
	finetuned	imbalanced	half	98.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00
		balanced	half	87.40±4.40	88.60±2.50	86.20±2.50	78.30±2.90	87.20±3.70	88.10±2.80
		imbalanced	full	98.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00
		balanced	full	87.67±4.00	88.60±2.40	86.20±2.90	78.30±3.10	87.30±3.70	88.00±3.20

Comparison of representations and classifiers performance for discriminating ion channels from membrane proteins on accuracy metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 63: MCC comparison of representations for IC-MP

Representer	Representation	Dataset	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-1b	finetuned	imbalanced	half	0.99±0.01	0.99±0.01	0.98±0.01	0.99±0.01	1.00±0.00	1.00±0.01
		balanced	half	0.82±0.07	0.85±0.04	0.77±0.05	0.66±0.05	0.84±0.04	0.83±0.04
		imbalanced	full	0.99±0.01	0.99±0.01	0.97±0.01	0.98±0.01	0.99±0.01	0.99±0.01
		balanced	full	0.83±0.04	0.85±0.04	0.77±0.05	0.66±0.05	0.84±0.04	0.83±0.04
	frozen	imbalanced	half	0.83±0.07	0.88±0.03	0.58±0.03	0.78±0.03	0.83±0.04	0.85±0.04
		balanced	half	0.81±0.07	0.85±0.04	0.78±0.05	0.65±0.05	0.84±0.04	0.84±0.04
		imbalanced	full	0.87±0.04	0.88±0.03	0.59±0.04	0.78±0.03	0.83±0.04	0.85±0.04
		balanced	full	0.82±0.04	0.85±0.04	0.78±0.05	0.65±0.05	0.84±0.04	0.84±0.04
ESM-2	finetuned	imbalanced	half	0.97±0.02	0.95±0.03	0.90±0.01	0.90±0.03	0.95±0.02	0.95±0.02
		balanced	half	0.84±0.05	0.85±0.04	0.76±0.05	0.64±0.05	0.83±0.04	0.84±0.04
		imbalanced	full	0.97±0.01	0.95±0.01	0.91±0.02	0.90±0.02	0.95±0.02	0.95±0.03
		balanced	full	0.84±0.05	0.84±0.03	0.77±0.06	0.64±0.05	0.83±0.04	0.84±0.04
	frozen	imbalanced	half	0.88±0.05	0.88±0.03	0.51±0.05	0.75±0.05	0.87±0.04	0.86±0.04
		balanced	half	0.83±0.06	0.85±0.04	0.76±0.06	0.64±0.06	0.84±0.04	0.84±0.04
		imbalanced	full	0.87±0.05	0.88±0.03	0.52±0.06	0.75±0.05	0.87±0.04	0.86±0.04
		balanced	full	0.84±0.05	0.85±0.04	0.77±0.06	0.63±0.06	0.84±0.04	0.84±0.04
ESM-2_15B	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	frozen	imbalanced	half	0.88±0.03	0.88±0.05	0.40±0.03	0.72±0.03	0.89±0.03	0.88±0.03
		balanced	half	0.85±0.04	0.86±0.04	0.80±0.05	0.47±0.06	0.87±0.04	0.87±0.04
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ProtBERT	finetuned	imbalanced	half	0.87±0.02	0.86±0.03	0.73±0.02	0.79±0.03	0.83±0.03	0.84±0.03
		balanced	half	0.75±0.06	0.76±0.06	0.73±0.06	0.51±0.07	0.75±0.06	0.75±0.05
		imbalanced	full	0.88±0.05	0.88±0.04	0.80±0.06	0.84±0.04	0.85±0.05	0.87±0.05
		balanced	full	0.74±0.07	0.76±0.06	0.72±0.05	0.50±0.08	0.74±0.06	0.75±0.05
	frozen	imbalanced	half	0.81±0.02	0.78±0.03	0.31±0.05	0.54±0.05	0.79±0.03	0.79±0.03
		balanced	half	0.75±0.08	0.75±0.06	0.72±0.05	0.49±0.08	0.73±0.07	0.75±0.05
		imbalanced	full	0.81±0.05	0.78±0.03	0.30±0.04	0.54±0.06	0.79±0.03	0.78±0.04
		balanced	full	0.75±0.06	0.76±0.06	0.73±0.06	0.49±0.08	0.74±0.07	0.75±0.05
ProtT5	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	frozen	imbalanced	half	0.87±0.05	0.86±0.03	0.54±0.07	0.76±0.06	0.82±0.04	0.84±0.03
		balanced	half	0.83±0.06	0.84±0.04	0.78±0.05	0.64±0.06	0.82±0.04	0.82±0.05
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ProtBERT-BFD	finetuned	imbalanced	half	0.82±0.02	0.87±0.04	0.86±0.03	0.84±0.03	0.86±0.04	0.85±0.04
		balanced	half	0.76±0.08	0.77±0.05	0.72±0.05	0.60±0.05	0.75±0.08	0.76±0.05
		imbalanced	full	0.82±0.03	0.83±0.03	0.82±0.05	0.81±0.04	0.83±0.04	0.82±0.04
		balanced	full	0.77±0.07	0.77±0.04	0.73±0.06	0.59±0.06	0.75±0.07	0.76±0.06
	frozen	imbalanced	half	0.78±0.05	0.75±0.04	0.34±0.04	0.63±0.03	0.72±0.03	0.74±0.03
		balanced	half	0.76±0.07	0.77±0.04	0.73±0.06	0.58±0.07	0.73±0.07	0.75±0.06
		imbalanced	full	0.80±0.04	0.75±0.04	0.33±0.07	0.63±0.03	0.72±0.03	0.74±0.01
		balanced	full	0.74±0.07	0.77±0.04	0.72±0.06	0.58±0.06	0.73±0.07	0.75±0.06

Comparison of representations and classifiers performance for discriminating ion channels from membrane proteins on MCC metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 64: sensitivity comparison of representations for IC-MP

Representer	Representation	Dataset	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-1b	finetuned	imbalanced	half	100.00±1.00	99.00±2.00	98.00±2.00	98.00±2.00	100.00±1.00	100.00±1.00
		balanced	half	89.50±3.60	90.50±2.70	80.70±4.80	95.00±3.20	91.20±2.40	92.00±2.60
		imbalanced	full	99.00±2.00	99.00±2.00	98.00±2.00	98.00±2.00	100.00±1.00	100.00±1.00
		balanced	full	89.30±4.20	90.50±2.70	80.90±4.40	94.80±3.10	91.20±2.80	92.10±2.50
	frozen	imbalanced	half	82.00±7.00	81.00±6.00	36.00±4.00	82.00±6.00	84.00±5.00	84.00±5.00
		balanced	half	89.30±5.10	90.70±2.70	80.20±4.50	95.20±2.80	91.90±2.20	92.10±2.60
		imbalanced	full	82.00±7.00	81.00±6.00	36.00±5.00	82.00±6.00	84.00±5.00	85.00±5.00
		balanced	full	88.80±3.90	90.70±2.70	81.10±4.00	95.30±2.70	91.80±2.20	92.20±2.40
ESM-2	finetuned	imbalanced	half	97.00±3.00	93.00±4.00	83.00±2.00	85.00±6.00	93.00±3.00	93.00±3.00
		balanced	half	90.20±4.50	91.40±2.60	81.10±6.20	92.20±4.20	91.60±2.60	92.00±3.00
		imbalanced	full	96.00±3.00	94.00±3.00	85.00±3.00	86.00±4.00	94.00±4.00	94.00±4.00
		balanced	full	89.90±4.90	91.30±2.80	81.60±6.20	92.00±4.10	91.30±2.70	92.40±2.60
	frozen	imbalanced	half	81.00±8.00	83.00±6.00	28.00±6.00	71.00±6.00	84.00±6.00	85.00±6.00
		balanced	half	89.00±6.50	91.40±2.40	81.60±5.90	92.40±4.40	91.40±2.70	92.20±2.80
		imbalanced	full	81.00±8.00	83.00±6.00	29.00±6.00	71.00±6.00	84.00±6.00	85.00±7.00
		balanced	full	90.10±4.70	91.40±2.40	81.60±5.80	92.30±4.30	91.40±2.60	92.10±2.80
ESM-2_15B	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	frozen	imbalanced	half	85.00±6.00	87.00±6.00	17.00±3.00	78.00±7.00	85.00±5.00	84.00±5.00
		balanced	half	88.10±4.90	92.00±2.60	80.80±4.40	95.90±2.80	92.80±2.90	92.70±2.90
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ProtT5	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	frozen	imbalanced	half	84.00±6.00	79.00±6.00	31.00±7.00	73.00±9.00	79.00±5.00	80.00±4.00
		balanced	half	87.90±6.80	88.30±4.00	79.40±3.90	93.70±2.80	90.00±2.40	90.90±2.80
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ProtBERT-BFD	finetuned	imbalanced	half	76.00±2.00	80.00±3.00	76.00±4.00	75.00±5.00	78.00±6.00	80.00±7.00
		balanced	half	85.20±9.00	87.40±3.20	79.80±4.60	90.90±3.20	87.20±4.80	88.40±3.30
		imbalanced	full	70.00±3.00	74.00±5.00	72.00±5.00	70.00±7.00	74.00±5.00	73.00±5.00
		balanced	full	87.67±7.67	87.90±3.20	80.10±5.00	90.40±3.20	87.00±4.80	88.50±3.30
	frozen	imbalanced	half	71.00±9.00	67.00±3.00	13.00±2.00	53.00±6.00	64.00±5.00	70.00±6.00
		balanced	half	86.00±8.90	87.40±2.70	80.60±5.90	90.70±3.60	86.70±4.30	88.40±3.40
		imbalanced	full	75.00±8.00	67.00±3.00	13.00±5.00	53.00±6.00	64.00±5.00	69.00±4.00
		balanced	full	86.00±7.90	87.30±2.70	80.50±6.30	90.70±3.20	86.70±4.50	88.00±3.60
ProtBERT	finetuned	imbalanced	half	81.00±2.00	81.00±7.00	56.00±4.00	68.00±5.00	77.00±5.00	80.00±5.00
		balanced	half	85.80±7.00	87.20±3.40	78.00±3.80	89.50±3.80	87.00±3.80	87.50±3.70
		imbalanced	full	85.00±6.00	84.00±4.00	68.00±9.00	80.00±5.00	80.00±8.00	83.00±6.00
		balanced	full	82.90±8.60	87.20±3.40	77.50±4.40	89.10±3.70	87.00±3.50	87.70±3.70
	frozen	imbalanced	half	68.00±4.00	74.00±4.00	11.00±4.00	52.00±8.00	73.00±5.00	77.00±7.00
		balanced	half	83.90±8.70	87.40±3.10	78.40±4.20	89.10±3.70	86.40±4.30	87.40±3.70
		imbalanced	full	76.00±2.00	74.00±4.00	10.00±3.00	53.00±8.00	73.00±5.00	74.00±5.00
		balanced	full	84.11±8.89	87.50±3.50	78.70±5.00	89.30±3.60	86.50±4.40	87.30±3.90

Comparison of representations and classifiers performance for discriminating ion channels from membrane proteins on sensitivity metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 65: specificity Comparison of representations for IC-MP

Representer	Representation	Dataset	Precision	CNN	SVM	RF	kNN	LR	FFNN
ProtBERT-BFD	frozen	imbalanced	half	99.00±1.00	99.00±0.00	100.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	half	88.80±8.70	89.00±3.80	90.90±3.80	66.20±4.90	86.20±4.60	86.70±4.40
		imbalanced	full	99.00±1.00	99.00±0.00	100.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	full	86.50±10.30	89.10±3.90	90.70±3.90	66.40±5.10	86.30±4.70	87.10±4.30
	finetuned	imbalanced	half	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	half	89.20±8.70	89.70±4.10	91.50±3.70	67.70±4.70	87.30±4.90	87.80±4.20
		imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	88.11±8.56	89.30±4.30	91.50±3.60	67.80±4.70	87.60±4.60	87.40±4.40
ESM-1b	frozen	imbalanced	half	99.00±1.00	100.00±0.00	100.00±0.00	98.00±0.00	99.00±0.00	99.00±0.00
		balanced	half	91.10±6.60	94.40±2.30	95.60±2.70	68.40±4.60	91.80±3.50	91.40±3.20
		imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	98.00±0.00	99.00±0.00	99.00±0.00
		balanced	full	93.00±5.20	94.40±2.30	95.20±2.70	68.30±4.60	91.80±3.60	91.60±3.20
	finetuned	imbalanced	half	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	half	91.80±7.50	94.60±2.50	94.90±2.70	69.10±4.90	92.30±3.40	91.40±3.30
		imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	92.50±5.10	94.50±2.50	95.40±2.70	69.80±4.60	92.20±3.40	91.20±3.60
ProtBERT	frozen	imbalanced	half	100.00±0.00	99.00±1.00	100.00±0.00	98.00±1.00	99.00±0.00	99.00±0.00
		balanced	half	89.20±10.00	87.80±4.50	92.70±3.10	58.20±6.50	87.00±5.20	87.80±4.10
		imbalanced	full	99.00±1.00	99.00±1.00	100.00±0.00	98.00±1.00	99.00±0.00	99.00±0.00
		balanced	full	89.22±9.56	87.80±4.50	92.80±3.30	57.80±6.40	87.10±5.30	87.70±4.20
	finetuned	imbalanced	half	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	99.00±0.00
		balanced	half	88.00±9.50	88.20±5.00	93.50±3.20	59.10±6.10	87.50±4.30	87.50±4.10
		imbalanced	full	100.00±1.00	100.00±0.00	100.00±0.00	99.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	89.70±8.00	88.20±4.90	93.10±2.90	58.90±6.90	87.60±4.50	87.80±4.00
ProtT5	frozen	imbalanced	half	99.00±0.00	100.00±0.00	100.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	half	93.80±4.90	95.20±2.40	96.70±2.50	68.30±5.70	91.40±3.10	90.80±3.10
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ESM-2	frozen	imbalanced	half	100.00±0.00	100.00±0.00	100.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	half	92.50±7.80	93.30±2.60	93.70±2.90	70.20±4.90	91.90±3.40	91.80±3.00
		imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	full	93.50±6.50	93.30±2.50	93.70±3.30	70.00±4.60	92.00±3.40	92.00±3.00
	finetuned	imbalanced	half	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	half	93.10±6.30	93.10±2.60	93.90±3.50	70.50±4.60	91.90±3.10	91.80±3.40
		imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	93.70±5.40	93.10±2.60	93.80±2.70	70.40±4.50	92.00±3.20	91.70±3.20
ESM-2_15B	frozen	imbalanced	half	100.00±0.00	99.00±0.00	100.00±0.00	98.00±1.00	100.00±0.00	100.00±0.00
		balanced	half	95.50±4.80	93.90±2.30	97.40±1.60	45.50±5.10	94.10±2.80	94.20±2.90
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-

Comparison of representations and classifiers performance for discriminating ion channels from membrane proteins on specificity metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 66: accuracy comparison of representations for IT-MP

Representer	Representation	Dataset	Precision	CNN	SVM	RF	kNN	LR	FFNN
ProtBERT-BFD	finetuned	imbalanced	full	99.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	full	87.70±2.90	86.70±3.00	82.80±4.00	80.60±3.30	86.40±3.10	86.30±3.00
		imbalanced	half	98.00±0.00	99.00±0.00	99.00±0.00	98.00±0.00	99.00±0.00	99.00±0.00
		balanced	half	87.60±2.80	86.40±2.90	82.60±3.90	80.10±3.20	86.20±3.20	86.30±2.90
	frozen	imbalanced	full	97.00±0.00	96.00±0.00	94.00±0.00	95.00±1.00	96.00±0.00	97.00±0.00
		balanced	full	87.40±3.10	86.30±2.70	82.50±3.90	79.90±3.40	86.30±3.20	86.40±2.90
		imbalanced	half	96.00±0.00	97.00±0.00	94.00±0.00	95.00±1.00	96.00±0.00	97.00±0.00
		balanced	half	87.50±2.90	86.40±2.60	82.10±4.40	79.70±3.30	86.40±3.30	86.30±2.90
ESM-2	finetuned	imbalanced	full	100.00±0.00	99.00±0.00	98.00±0.00	99.00±0.00	99.00±0.00	99.00±0.00
		balanced	full	91.11±1.78	89.60±2.20	85.80±2.70	80.60±3.70	89.80±1.90	89.90±2.80
		imbalanced	half	100.00±0.00	99.00±0.00	98.00±0.00	98.00±1.00	99.00±0.00	99.00±0.00
		balanced	half	91.30±2.00	89.40±2.20	85.60±2.90	80.60±3.50	89.70±1.90	89.70±2.80
	frozen	imbalanced	full	97.00±0.00	97.00±0.00	94.00±1.00	95.00±0.00	97.00±1.00	97.00±1.00
		balanced	full	91.20±1.70	89.30±2.40	85.70±2.50	80.80±3.40	89.60±2.00	89.70±3.00
		imbalanced	half	97.00±0.00	97.00±0.00	94.00±0.00	95.00±0.00	97.00±1.00	97.00±0.00
		balanced	half	91.10±1.90	89.40±2.30	85.70±2.60	80.70±3.40	89.50±2.00	89.70±2.70
ProtBERT	finetuned	imbalanced	full	99.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00
		balanced	full	88.20±1.90	86.90±2.30	82.80±3.10	77.90±3.30	87.50±2.70	87.50±2.30
		imbalanced	half	98.00±0.00	98.00±0.00	97.00±0.00	97.00±0.00	98.00±1.00	98.00±0.00
		balanced	half	88.20±2.10	86.90±2.20	82.90±3.40	78.00±3.30	87.10±2.10	87.60±2.30
	frozen	imbalanced	full	96.00±1.00	96.00±0.00	93.00±0.00	94.00±1.00	96.00±0.00	96.00±0.00
		balanced	full	88.10±2.00	86.50±2.60	82.30±3.50	77.50±3.20	87.10±2.40	87.20±2.80
		imbalanced	half	96.00±0.00	96.00±0.00	93.00±0.00	94.00±1.00	96.00±0.00	96.00±0.00
		balanced	half	88.20±1.70	86.50±2.50	82.30±3.80	77.30±2.90	87.10±2.30	87.20±2.80
ESM-1b	finetuned	imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	90.80±2.20	90.70±2.10	87.40±2.50	84.50±2.90	90.00±2.60	89.90±2.90
		imbalanced	half	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	half	90.80±1.90	90.80±2.10	87.60±2.40	84.50±2.80	90.10±2.60	90.10±3.00
	frozen	imbalanced	full	96.00±1.00	97.00±0.00	94.00±1.00	96.00±0.00	97.00±0.00	97.00±0.00
		balanced	full	90.90±2.10	90.40±2.10	87.00±2.70	84.20±2.90	89.90±2.70	90.00±2.40
		imbalanced	half	97.00±0.00	97.00±0.00	94.00±0.00	96.00±0.00	97.00±0.00	97.00±0.00
		balanced	half	90.56±2.67	90.40±2.20	86.90±3.10	84.20±2.80	89.90±2.60	90.00±2.80
ESM-2_15B	finetuned	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
	frozen	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	97.00±0.00	97.00±0.00	93.00±0.00	95.00±1.00	97.00±0.00	97.00±0.00
		balanced	half	90.80±2.10	90.70±2.70	86.00±2.70	74.00±3.30	91.00±2.80	90.50±2.90
ProtT5	finetuned	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
	frozen	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	97.00±0.00	97.00±0.00	94.00±0.00	96.00±0.00	97.00±0.00	97.00±0.00
		balanced	half	91.80±1.70	91.70±2.40	88.00±2.30	82.40±2.70	90.80±2.00	90.70±2.50

Comparison of representations and classifiers performance for discriminating ion transporters from membrane proteins on accuracy metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 67: MCC comparison of representations for IT-MP

Representer	Representation	Dataset	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-1b	finetuned	imbalanced	full	0.99±0.01	1.00±0.00	0.99±0.01	0.99±0.01	1.00±0.00	1.00±0.01
		balanced	full	0.82±0.04	0.82±0.04	0.75±0.05	0.70±0.05	0.80±0.05	0.80±0.06
		imbalanced	half	0.99±0.01	0.99±0.01	0.98±0.01	0.99±0.01	1.00±0.00	1.00±0.00
		balanced	half	0.82±0.04	0.82±0.04	0.75±0.05	0.69±0.05	0.80±0.05	0.80±0.06
	frozen	imbalanced	full	0.74±0.04	0.77±0.04	0.42±0.10	0.74±0.02	0.77±0.03	0.79±0.03
		balanced	full	0.82±0.04	0.81±0.04	0.74±0.06	0.69±0.06	0.80±0.05	0.80±0.05
		imbalanced	half	0.77±0.03	0.77±0.04	0.45±0.03	0.74±0.02	0.78±0.04	0.79±0.03
		balanced	half	0.82±0.05	0.81±0.04	0.74±0.06	0.69±0.05	0.80±0.05	0.80±0.05
ESM-2	finetuned	imbalanced	full	0.98±0.01	0.95±0.01	0.86±0.04	0.90±0.03	0.95±0.02	0.94±0.02
		balanced	full	0.83±0.04	0.80±0.04	0.72±0.06	0.62±0.07	0.80±0.04	0.80±0.06
		imbalanced	half	0.97±0.01	0.94±0.03	0.88±0.04	0.87±0.03	0.93±0.03	0.93±0.03
		balanced	half	0.83±0.04	0.79±0.04	0.72±0.06	0.62±0.07	0.80±0.04	0.79±0.06
	frozen	imbalanced	full	0.77±0.02	0.77±0.04	0.44±0.07	0.65±0.03	0.75±0.04	0.76±0.04
		balanced	full	0.83±0.04	0.79±0.05	0.72±0.05	0.63±0.07	0.80±0.04	0.80±0.06
		imbalanced	half	0.74±0.04	0.77±0.04	0.43±0.07	0.65±0.03	0.74±0.05	0.76±0.03
		balanced	half	0.82±0.04	0.79±0.04	0.72±0.05	0.63±0.07	0.79±0.04	0.80±0.06
ProtBERT	finetuned	imbalanced	full	0.92±0.03	0.89±0.04	0.81±0.03	0.86±0.04	0.89±0.03	0.88±0.03
		balanced	full	0.76±0.04	0.74±0.04	0.66±0.06	0.57±0.07	0.75±0.05	0.75±0.05
		imbalanced	half	0.88±0.02	0.87±0.03	0.76±0.02	0.81±0.04	0.87±0.04	0.87±0.03
		balanced	half	0.77±0.04	0.74±0.05	0.66±0.07	0.57±0.07	0.74±0.04	0.75±0.05
	frozen	imbalanced	full	0.64±0.14	0.72±0.04	0.22±0.07	0.51±0.06	0.68±0.03	0.71±0.04
		balanced	full	0.76±0.04	0.73±0.05	0.65±0.07	0.56±0.06	0.74±0.05	0.75±0.06
		imbalanced	half	0.69±0.03	0.72±0.04	0.23±0.05	0.52±0.05	0.68±0.03	0.71±0.03
		balanced	half	0.77±0.04	0.73±0.05	0.65±0.07	0.56±0.06	0.74±0.05	0.75±0.06
ProtBERT-BFD	finetuned	imbalanced	full	0.90±0.03	0.92±0.03	0.92±0.03	0.92±0.02	0.92±0.02	0.92±0.02
		balanced	full	0.76±0.06	0.73±0.06	0.66±0.08	0.61±0.07	0.73±0.06	0.73±0.06
		imbalanced	half	0.89±0.03	0.90±0.02	0.90±0.01	0.88±0.01	0.90±0.01	0.90±0.01
		balanced	half	0.75±0.06	0.73±0.05	0.66±0.08	0.61±0.07	0.73±0.06	0.73±0.06
	frozen	imbalanced	full	0.74±0.03	0.74±0.04	0.41±0.03	0.62±0.06	0.71±0.02	0.75±0.01
		balanced	full	0.75±0.06	0.73±0.06	0.65±0.08	0.60±0.07	0.72±0.06	0.73±0.06
		imbalanced	half	0.71±0.05	0.74±0.04	0.43±0.05	0.62±0.06	0.72±0.01	0.75±0.02
		balanced	half	0.75±0.06	0.73±0.05	0.65±0.09	0.60±0.07	0.73±0.07	0.73±0.06
ProtT5	finetuned	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
	frozen	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	0.76±0.03	0.80±0.01	0.42±0.05	0.73±0.03	0.79±0.02	0.79±0.04
		balanced	half	0.83±0.03	0.83±0.05	0.76±0.05	0.66±0.06	0.82±0.04	0.82±0.05
ESM-2_15B	finetuned	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
	frozen	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	0.79±0.03	0.78±0.03	0.27±0.03	0.66±0.06	0.80±0.03	0.79±0.02
		balanced	half	0.82±0.04	0.82±0.05	0.72±0.06	0.53±0.06	0.81±0.06	0.81±0.06

Comparison of representations and classifiers performance for discriminating ion transporters from membrane proteins on MCC metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 68: sensitivity comparison of representations for IT-MP

Representer	Representation	Dataset	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-1b	finetuned	imbalanced	half	99.00±1.00	99.00±1.00	98.00±2.00	99.00±1.00	99.00±1.00	100.00±1.00
		balanced	half	88.50±3.70	88.20±3.50	85.50±3.10	87.60±4.00	89.30±2.90	90.00±2.90
		imbalanced	full	99.00±1.00	100.00±1.00	99.00±1.00	99.00±1.00	100.00±0.00	99.00±1.00
		balanced	full	88.50±2.90	88.00±3.50	85.70±4.00	87.60±4.20	89.40±3.10	89.90±3.10
	frozen	imbalanced	half	72.00±6.00	76.00±7.00	23.00±4.00	69.00±4.00	74.00±5.00	77.00±5.00
		balanced	half	89.22±3.67	88.20±3.60	84.40±4.40	87.40±3.90	89.20±2.80	90.10±3.00
		imbalanced	full	72.00±11.00	76.00±7.00	21.00±8.00	69.00±4.00	74.00±5.00	76.00±5.00
		balanced	full	89.00±3.40	88.20±3.50	84.70±4.30	87.40±3.90	89.10±2.80	90.10±3.00
ESM-2	finetuned	imbalanced	half	98.00±2.00	95.00±3.00	81.00±6.00	89.00±3.00	93.00±4.00	93.00±3.00
		balanced	half	89.00±3.20	89.10±2.90	82.50±4.30	88.60±3.90	89.30±3.20	90.20±3.40
		imbalanced	full	98.00±2.00	95.00±3.00	77.00±7.00	89.00±4.00	94.00±2.00	93.00±3.00
		balanced	full	89.00±2.67	89.20±2.70	82.60±4.40	88.80±4.30	89.30±3.10	90.40±3.30
	frozen	imbalanced	half	64.00±7.00	72.00±7.00	22.00±6.00	61.00±7.00	69.00±6.00	72.00±7.00
		balanced	half	89.30±2.90	88.90±2.80	82.70±3.70	88.80±3.90	88.90±2.90	89.90±3.60
		imbalanced	full	71.00±5.00	72.00±7.00	24.00±7.00	61.00±7.00	69.00±6.00	72.00±8.00
		balanced	full	89.80±2.50	88.80±2.90	82.50±2.80	88.80±3.80	89.00±2.90	89.60±3.40
ESM-2_15B	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	frozen	imbalanced	half	71.00±8.00	77.00±5.00	8.00±2.00	70.00±7.00	77.00±4.00	75.00±5.00
		balanced	half	87.80±2.80	90.00±3.40	83.40±4.10	94.80±2.60	89.50±3.60	89.90±3.60
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ProtT5	finetuned	imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
	frozen	imbalanced	half	69.00±8.00	74.00±4.00	19.00±5.00	75.00±6.00	76.00±5.00	76.00±6.00
		balanced	half	91.60±2.30	90.30±3.40	85.40±3.50	92.30±3.60	91.00±2.80	91.00±3.50
		imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
ProtBERT-BFD	finetuned	imbalanced	half	87.00±4.00	87.00±5.00	86.00±2.00	86.00±4.00	88.00±3.00	87.00±2.00
		balanced	half	86.10±3.80	84.70±4.30	78.70±6.80	85.10±4.40	85.90±4.10	85.70±3.80
		imbalanced	full	85.00±7.00	91.00±5.00	91.00±5.00	91.00±4.00	90.00±5.00	91.00±4.00
		balanced	full	87.00±3.50	85.40±4.50	79.40±6.80	85.20±4.20	86.20±3.90	86.00±4.30
	frozen	imbalanced	half	60.00±11.00	71.00±10.00	21.00±5.00	61.00±5.00	67.00±3.00	69.00±4.00
		balanced	half	86.00±3.20	85.30±3.80	78.60±6.40	84.70±4.40	86.00±4.00	85.90±4.10
		imbalanced	full	64.00±8.00	70.00±11.00	20.00±3.00	61.00±5.00	66.00±4.00	69.00±5.00
		balanced	full	85.90±3.30	85.40±4.00	78.90±5.60	84.70±4.70	86.10±3.80	86.00±4.10
ProtBERT	finetuned	imbalanced	half	83.00±5.00	85.00±5.00	61.00±3.00	75.00±6.00	82.00±5.00	82.00±6.00
		balanced	half	88.50±2.90	85.80±3.50	83.10±4.90	86.90±4.30	86.60±3.50	87.70±3.60
		imbalanced	full	88.00±5.00	85.00±5.00	68.00±5.00	78.00±4.00	85.00±4.00	84.00±5.00
		balanced	full	87.90±2.50	86.20±3.10	82.40±4.30	86.70±3.40	86.90±3.90	87.70±3.30
	frozen	imbalanced	half	59.00±8.00	68.00±6.00	6.00±2.00	45.00±4.00	63.00±6.00	68.00±6.00
		balanced	half	88.00±2.70	85.60±3.80	82.40±5.20	86.20±3.70	86.30±3.70	87.00±3.70
		imbalanced	full	48.00±19.00	68.00±6.00	6.00±4.00	45.00±5.00	63.00±6.00	69.00±6.00
		balanced	full	87.80±2.80	85.60±3.70	82.60±5.30	86.40±3.40	86.40±3.80	86.80±3.90

Comparison of representations and classifiers performance for discriminating ion transporters from membrane proteins on sensitivity metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 69: specificity comparison of representations for IT-MP

Representer	Representation	Dataset	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-2	frozen	imbalanced	full	99.00±1.00	99.00±0.00	100.00±0.00	98.00±0.00	99.00±0.00	99.00±1.00
		balanced	full	92.90±3.50	90.20±4.20	89.40±4.30	72.80±4.60	90.30±4.30	89.60±5.40
		imbalanced	half	99.00±0.00	99.00±0.00	100.00±0.00	98.00±0.00	99.00±0.00	99.00±1.00
		balanced	half	92.80±4.50	90.10±4.10	88.90±4.10	72.80±4.70	90.10±4.20	89.50±5.00
	finetuned	imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	99.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	93.33±3.33	90.20±4.20	88.90±3.90	72.40±4.80	90.10±4.30	89.40±5.10
		imbalanced	half	100.00±0.00	99.00±0.00	100.00±0.00	99.00±1.00	100.00±0.00	100.00±0.00
		balanced	half	93.80±3.60	90.30±4.20	88.80±4.10	72.50±5.00	90.10±4.00	89.30±5.10
ProtBERT-BFD	frozen	imbalanced	full	99.00±0.00	99.00±1.00	100.00±0.00	98.00±1.00	99.00±0.00	99.00±0.00
		balanced	full	89.10±4.60	87.20±4.00	85.80±4.90	74.80±4.30	86.30±4.90	86.80±4.70
		imbalanced	half	99.00±1.00	99.00±1.00	100.00±0.00	98.00±1.00	99.00±0.00	99.00±0.00
		balanced	half	88.80±4.90	87.30±3.80	85.80±4.20	74.70±4.20	86.50±5.10	86.50±4.90
	finetuned	imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	89.00±4.50	88.00±3.80	86.20±4.40	75.70±4.30	86.70±4.60	86.90±4.80
		imbalanced	half	99.00±1.00	100.00±0.00	100.00±0.00	99.00±0.00	100.00±0.00	100.00±0.00
		balanced	half	89.20±4.40	87.90±3.70	86.80±3.90	75.30±4.00	86.30±4.60	87.00±4.40
ESM-1b	frozen	imbalanced	full	98.00±1.00	99.00±0.00	100.00±0.00	99.00±1.00	99.00±0.00	99.00±0.00
		balanced	full	93.00±3.80	92.90±2.80	89.20±4.60	81.00±4.70	90.70±4.20	89.80±3.80
		imbalanced	half	99.00±1.00	99.00±0.00	100.00±0.00	99.00±1.00	99.00±0.00	99.00±0.00
		balanced	half	92.22±4.67	92.90±2.60	89.30±4.50	81.10±4.70	90.70±4.20	89.70±4.40
	finetuned	imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	93.40±3.40	93.40±2.70	89.20±3.90	81.60±4.70	90.50±4.30	90.00±5.00
		imbalanced	half	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	half	92.90±3.50	93.30±2.40	89.50±3.80	81.30±4.40	90.60±4.30	90.00±4.80
ESM-2_15B	frozen	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	99.00±0.00	99.00±0.00	100.00±0.00	97.00±1.00	99.00±0.00	99.00±1.00
		balanced	half	93.90±3.10	91.30±4.10	88.80±4.10	53.00±6.30	91.70±3.90	91.10±4.40
	finetuned	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
ProtT5	frozen	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	99.00±1.00	99.00±0.00	100.00±0.00	98.00±0.00	99.00±0.00	99.00±0.00
		balanced	half	91.90±3.40	92.90±3.60	90.60±3.60	72.50±4.40	90.80±3.40	90.30±3.90
	finetuned	imbalanced	full	-	-	-	-	-	-
		balanced	full	-	-	-	-	-	-
		imbalanced	half	-	-	-	-	-	-
		balanced	half	-	-	-	-	-	-
ProtBERT	frozen	imbalanced	full	100.00±0.00	99.00±0.00	100.00±0.00	98.00±0.00	98.00±0.00	98.00±0.00
		balanced	full	88.40±4.00	87.40±4.70	82.40±4.40	68.10±5.40	87.80±4.90	87.60±4.80
		imbalanced	half	99.00±0.00	99.00±0.00	100.00±0.00	98.00±0.00	98.00±0.00	99.00±0.00
		balanced	half	88.30±4.00	87.50±4.60	82.60±5.20	68.20±5.30	87.90±5.10	87.40±4.80
	finetuned	imbalanced	full	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
		balanced	full	88.40±3.90	87.70±4.70	83.30±4.80	69.10±5.40	88.10±4.60	87.60±4.70
		imbalanced	half	100.00±0.00	99.00±0.00	100.00±0.00	99.00±0.00	100.00±0.00	100.00±0.00
		balanced	half	88.00±4.10	88.00±4.30	83.00±4.40	69.50±4.90	87.70±4.60	87.60±4.50

Comparison of representations and classifiers performance for discriminating ion transporters from membrane proteins on specificity metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

B.3.1 Ion channels vs ion transporters

Table 70: accuracy comparison of representations for IC-IT

Representer	Representation	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-1b	finetuned	half	93.00±1.00	93.00±2.00	92.00±2.00	89.00±3.00	94.00±2.00	94.00±3.00
		full	91.00±5.00	93.00±2.00	91.00±3.00	89.00±3.00	94.00±2.00	94.00±3.00
	frozen	half	93.00±2.00	94.00±2.00	92.00±2.00	90.00±2.00	94.00±2.00	93.00±2.00
		full	93.00±1.00	94.00±2.00	92.00±2.00	90.00±2.00	94.00±2.00	93.00±2.00
ESM-2	finetuned	half	93.00±1.00	93.00±2.00	90.00±1.00	87.00±5.00	92.00±1.00	94.00±2.00
		full	94.00±1.00	93.00±2.00	90.00±2.00	87.00±4.00	92.00±1.00	93.00±3.00
	frozen	half	92.00±2.00	93.00±2.00	89.00±2.00	87.00±5.00	92.00±1.00	94.00±2.00
		full	94.00±1.00	93.00±2.00	89.00±2.00	87.00±5.00	92.00±1.00	94.00±2.00
ESM-2_15B	finetuned	half	-	-	-	-	-	-
		full	-	-	-	-	-	-
	frozen	half	94.00±1.00	94.00±1.00	90.00±1.00	89.00±4.00	94.00±1.00	93.00±2.00
		full	-	-	-	-	-	-
ProtT5	finetuned	half	-	-	-	-	-	-
		full	-	-	-	-	-	-
	frozen	half	93.00±1.00	93.00±2.00	89.00±2.00	90.00±2.00	93.00±2.00	93.00±2.00
		full	-	-	-	-	-	-
ProtBERT	finetuned	half	93.00±1.00	92.00±0.00	89.00±2.00	82.00±4.00	90.00±1.00	91.00±1.00
		full	93.00±0.00	92.00±0.00	89.00±2.00	82.00±4.00	90.00±1.00	91.00±1.00
	frozen	half	92.00±0.00	92.00±1.00	88.00±3.00	82.00±3.00	90.00±2.00	91.00±2.00
		full	92.00±1.00	92.00±1.00	88.00±3.00	82.00±3.00	90.00±2.00	91.00±2.00
ProtBERT-BFD	finetuned	half	92.00±3.00	90.00±2.00	87.00±3.00	86.00±2.00	89.00±2.00	90.00±2.00
		full	92.00±3.00	90.00±3.00	88.00±2.00	85.00±2.00	88.00±2.00	90.00±2.00
	frozen	half	92.00±3.00	90.00±2.00	87.00±3.00	86.00±2.00	87.00±2.00	89.00±4.00
		full	92.00±3.00	90.00±2.00	87.00±3.00	86.00±3.00	87.00±2.00	89.00±2.00

Comparison of representations and classifiers performance for discriminating ion channels from ion transporters on accuracy metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 71: MCC comparison of representations for IC-IT

Representer	Representation	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-2	finetuned	full	0.89±0.03	0.87±0.04	0.80±0.03	0.74±0.08	0.84±0.03	0.86±0.05
		half	0.86±0.03	0.87±0.04	0.80±0.01	0.75±0.09	0.84±0.01	0.87±0.05
	frozen	full	0.88±0.02	0.87±0.04	0.78±0.03	0.74±0.09	0.84±0.02	0.88±0.05
		half	0.85±0.04	0.87±0.04	0.77±0.03	0.74±0.09	0.85±0.02	0.87±0.04
ESM-2_15B	finetuned	full	-	-	-	-	-	-
		half	-	-	-	-	-	-
	frozen	full	-	-	-	-	-	-
		half	0.89±0.02	0.88±0.03	0.80±0.03	0.79±0.07	0.87±0.03	0.87±0.03
ESM-1b	finetuned	full	0.83±0.08	0.86±0.05	0.83±0.06	0.79±0.05	0.88±0.05	0.87±0.06
		half	0.87±0.02	0.87±0.05	0.84±0.03	0.80±0.05	0.88±0.04	0.87±0.06
	frozen	full	0.85±0.02	0.87±0.04	0.83±0.05	0.80±0.05	0.88±0.03	0.87±0.04
		half	0.87±0.03	0.87±0.04	0.84±0.03	0.80±0.05	0.88±0.03	0.87±0.04
ProtT5	finetuned	full	-	-	-	-	-	-
		half	-	-	-	-	-	-
	frozen	full	-	-	-	-	-	-
		half	0.85±0.03	0.86±0.04	0.79±0.05	0.81±0.03	0.86±0.03	0.85±0.04
ProtBERT	finetuned	full	0.86±0.01	0.84±0.01	0.78±0.04	0.66±0.08	0.80±0.02	0.81±0.02
		half	0.86±0.02	0.84±0.01	0.78±0.05	0.66±0.08	0.80±0.02	0.81±0.02
	frozen	full	0.85±0.02	0.84±0.02	0.77±0.06	0.65±0.06	0.81±0.03	0.82±0.04
		half	0.84±0.00	0.84±0.02	0.77±0.05	0.65±0.06	0.80±0.03	0.82±0.04
ProtBERT-BFD	finetuned	full	0.84±0.06	0.81±0.05	0.76±0.05	0.71±0.04	0.76±0.04	0.81±0.05
		half	0.84±0.06	0.81±0.04	0.75±0.06	0.71±0.05	0.77±0.03	0.81±0.04
	frozen	full	0.84±0.07	0.81±0.04	0.75±0.05	0.72±0.05	0.74±0.05	0.79±0.05
		half	0.84±0.06	0.81±0.04	0.75±0.06	0.72±0.05	0.75±0.04	0.78±0.08

Comparison of representations and classifiers performance for discriminating ion channels from ion transporters on MCC metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 72: sensitivity comparison of representations for IC-IT

Representer	Representation	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-1b	frozen	full	91.00±1.00	93.00±2.00	89.00±6.00	95.00±3.00	93.00±3.00	95.00±2.00
		half	93.00±1.00	93.00±2.00	90.00±6.00	95.00±3.00	93.00±3.00	95.00±2.00
	finetuned	full	88.00±13.00	94.00±3.00	88.00±5.00	95.00±3.00	95.00±2.00	94.00±3.00
		half	93.00±2.00	94.00±3.00	88.00±6.00	95.00±3.00	95.00±2.00	94.00±3.00
ESM-2	frozen	full	93.00±2.00	93.00±2.00	85.00±4.00	90.00±7.00	92.00±3.00	93.00±3.00
		half	93.00±3.00	93.00±2.00	85.00±7.00	90.00±7.00	93.00±3.00	93.00±3.00
	finetuned	full	92.00±2.00	93.00±2.00	87.00±4.00	91.00±6.00	92.00±2.00	93.00±3.00
		half	93.00±3.00	93.00±2.00	87.00±5.00	91.00±6.00	90.00±2.00	94.00±4.00
ESM-2_15B	frozen	full	-	-	-	-	-	-
		half	94.00±2.00	92.00±2.00	85.00±5.00	92.00±3.00	93.00±2.00	93.00±2.00
	finetuned	full	-	-	-	-	-	-
		half	-	-	-	-	-	-
ProtT5	frozen	full	-	-	-	-	-	-
		half	91.00±2.00	90.00±4.00	86.00±4.00	94.00±1.00	92.00±2.00	93.00±3.00
	finetuned	full	-	-	-	-	-	-
		half	-	-	-	-	-	-
ProtBERT-BFD	frozen	full	92.00±4.00	88.00±8.00	85.00±7.00	85.00±4.00	88.00±5.00	90.00±5.00
		half	91.00±3.00	88.00±8.00	86.00±8.00	85.00±3.00	88.00±5.00	89.00±6.00
	finetuned	full	91.00±3.00	88.00±7.00	87.00±7.00	87.00±3.00	88.00±4.00	90.00±5.00
		half	91.00±2.00	90.00±6.00	86.00±8.00	86.00±4.00	88.00±4.00	92.00±4.00
ProtBERT	frozen	full	91.00±5.00	92.00±3.00	85.00±7.00	85.00±6.00	89.00±4.00	90.00±4.00
		half	89.00±4.00	92.00±3.00	85.00±7.00	85.00±6.00	89.00±4.00	90.00±4.00
	finetuned	full	91.00±2.00	92.00±3.00	84.00±7.00	88.00±5.00	90.00±3.00	90.00±4.00
		half	92.00±3.00	92.00±3.00	85.00±6.00	88.00±5.00	90.00±3.00	90.00±4.00

Comparison of representations and classifiers performance for discriminating ion channels from ion transporters on sensitivity metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Table 73: specificity comparison of representations for IC-IT

Representer	Representation	Precision	CNN	SVM	RF	kNN	LR	FFNN
ESM-2	finetuned	full	96.00±2.00	94.00±4.00	92.00±5.00	83.00±7.00	92.00±4.00	94.00±4.00
		half	93.00±3.00	94.00±4.00	92.00±5.00	84.00±7.00	94.00±3.00	94.00±2.00
	frozen	full	96.00±2.00	94.00±3.00	91.00±5.00	84.00±7.00	93.00±3.00	95.00±3.00
		half	92.00±6.00	94.00±3.00	92.00±5.00	84.00±7.00	92.00±4.00	95.00±3.00
ESM-1b	finetuned	full	93.00±6.00	92.00±5.00	95.00±6.00	84.00±5.00	93.00±4.00	94.00±4.00
		half	94.00±3.00	93.00±4.00	95.00±3.00	85.00±5.00	93.00±3.00	94.00±4.00
	frozen	full	94.00±2.00	94.00±3.00	94.00±5.00	85.00±5.00	94.00±5.00	92.00±4.00
		half	94.00±3.00	94.00±3.00	94.00±5.00	85.00±5.00	94.00±5.00	92.00±4.00
ESM-2_15B	finetuned	full	-	-	-	-	-	-
		half	-	-	-	-	-	-
	frozen	full	-	-	-	-	-	-
		half	94.00±3.00	95.00±3.00	94.00±5.00	86.00±7.00	94.00±2.00	94.00±3.00
ProtT5	finetuned	full	-	-	-	-	-	-
		half	-	-	-	-	-	-
	frozen	full	-	-	-	-	-	-
		half	94.00±2.00	95.00±3.00	93.00±6.00	87.00±4.00	94.00±3.00	92.00±5.00
ProtBERT	finetuned	full	94.00±3.00	92.00±3.00	92.00±7.00	78.00±7.00	90.00±3.00	91.00±3.00
		half	94.00±2.00	92.00±3.00	92.00±7.00	78.00±7.00	90.00±3.00	91.00±3.00
	frozen	full	93.00±4.00	91.00±4.00	91.00±7.00	80.00±7.00	91.00±2.00	91.00±4.00
		half	95.00±3.00	91.00±4.00	91.00±8.00	80.00±7.00	91.00±3.00	92.00±3.00
ProtBERT-BFD	finetuned	full	93.00±4.00	93.00±3.00	88.00±7.00	84.00±4.00	88.00±4.00	90.00±6.00
		half	93.00±4.00	90.00±4.00	88.00±8.00	85.00±3.00	89.00±3.00	89.00±5.00
	frozen	full	93.00±4.00	93.00±4.00	89.00±8.00	86.00±3.00	87.00±4.00	89.00±5.00
		half	93.00±5.00	93.00±4.00	88.00±7.00	86.00±3.00	87.00±4.00	89.00±6.00

Comparison of representations and classifiers performance for discriminating ion channels from ion transporters on specificity metric as m±d, where m is the mean and d is the standard deviation across the five runs of the cross-validation. The symbol "-" indicates that results are unavailable due to the extensive computational resources needed for fine-tuning large PLMs, which could not be accommodated by our limited resources.

Appendix C

Incorporating Secondary Structure Information into Protein Language Models

C.1 Task-Specific Results

This section provides detailed observations from seven different tasks: fluorescence prediction, solubility, subcellular localization, ion channel classification, transporter classification, membrane protein classification, and secondary structure prediction. Examining each task individually offers a comprehensive understanding of the performance differences between the Ankh and TooT-PLM-P2S models. This analysis highlights specific strengths and weaknesses of each model across various biological prediction tasks.

C.1.1 Fluorescence Prediction

We compared the performance of the Ankh model and the TooT-PLM-P2S model for fluorescence prediction using Spearman's correlation coefficient (ρ). The TooT-PLM-P2S model showed better performance in terms of mean, standard deviation, maximum, and median ρ values. The mean ρ for TooT-PLM-P2S was higher, and the standard deviation was also greater. The maximum ρ achieved by TooT-PLM-P2S was higher, and the median ρ was better for TooT-PLM-P2S. However, the differences in performance between the two models were not statistically significant, with a p-value of 0.48. The detailed comparison metrics are presented in Table 74.

Table 74: Fluorescence prediction comparison

Model	Mean ± Sd	Max	Min	Median	P-Value
Ankh	0.6360 ± 0.0157	0.6533	0.6160	0.6375	4.8e-01
TooT-PLM-P2S	0.6482 ± 0.0219	0.6742	0.6139	0.6523	

This table presents the comparison of fluorescence prediction performance between the TooT-PLM-P2S and Ankh models, using Spearman's correlation coefficient (ρ) on cross-validation. The table includes mean, standard deviation, maximum, minimum, and median values from the cross-validation results, along with the p-value indicating the statistical significance of the comparison. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. Boldface values denote higher performance in the comparison between the two models.

C.1.2 Solubility Prediction

We compared the Ankh and TooT-PLM-P2S models for solubility prediction using accuracy, precision, recall, F1 score, and Matthews correlation coefficient (MCC). The results, shown in

Table 75, highlight the performance metrics for each model in this task.

Table 75: Solubility Prediction: Performance Comparison

	accuracy	Precision	Recall	F1	MCC
Ankh	0.745 ± 0.048	0.687 ± 0.088	0.779 ± 0.117	**0.719 ± 0.010**	0.510 ± 0.039
TooT-PLM-P2S	0.762 ± 0.003	0.754 ± 0.019	0.641 ± 0.037	0.692 ± 0.014	0.506 ± 0.006
P-Value	4.50e-01	2.30e-01	1.10e-01	4.60e-03	8.00e-01

This table presents the performance comparison of the Ankh and TooT-PLM-P2S models for solubility prediction, using cross-validation. The metrics reported include accuracy, precision, recall, F1 score, and Matthews correlation coefficient (MCC). Each metric's p-value is provided to indicate statistical significance. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. Boldface values denote the higher performance between the models.

The TooT-PLM-P2S model achieved higher accuracy and precision than the Ankh model. However, the Ankh model showed higher recall and F1 score. Both models had comparable MCC values. The p-values indicate that the differences in accuracy, precision, recall, and MCC are not statistically significant (p-value ¿ 0.05). Only the difference in F1 score is statistically significant (p-value ¡ 0.05).

C.1.3 Sub-cellular Localization Prediction

In this subsection, we compare the performance of the Ankh model and the TooT-PLM-P2S model for subcellular localization prediction. Table 76 summarizes the evaluation metrics, including accuracy, precision, recall, F1 score, and MCC. Each metric is accompanied by its respective p-value to assess statistical significance.

Table 76: Subcellular Localization Prediction: Performance Comparison

	accuracy	Precision	Recall	F1	MCC
Ankh	**0.786 ± 0.007**	0.792 ± 0.012	**0.786 ± 0.007**	**0.781 ± 0.009**	**0.735 ± 0.009**
TooT-PLM-P2S	0.739 ± 0.010	0.753 ± 0.022	0.739 ± 0.010	0.735 ± 0.009	0.677 ± 0.014
P-Value	2.90e-03	5.70e-02	2.90e-03	3.70e-03	4.40e-03

This table presents the performance comparison of the Ankh and TooT-PLM-P2S models for subcellular localization prediction, using cross-validation. The metrics reported include accuracy, precision, recall, F1 score, and Matthews correlation coefficient (MCC). Each metric's p-value is provided to indicate statistical significance. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. Boldface values denote the higher performance between the models.

According to Table 76, the Ankh model outperformed the TooT-PLM-P2S model across all evaluation metrics. The p-values for each metric are less than 0.05, indicating statistically significant differences in performance between the two models.

C.1.4 Ion Channel Classification

This section compares the performance of the Ankh and TooT-PLM-P2S models for ion channel classification. The evaluation metrics, including accuracy, precision, recall, F1 score, and Matthews correlation coefficient (MCC), are summarized in Table 77.

Table 77: Ion Channel Classification: Performance Comparison

	accuracy	Precision	Recall	F1	MCC
Ankh	0.99 ± 0.003	0.95 ± 0.037	**0.88 ± 0.035**	0.92 ± 0.026	0.91 ± 0.028
TooT-PLM-P2S	0.98 ± 0.007	0.98 ± 0.023	0.77 ± 0.098	0.86 ± 0.064	0.86 ± 0.062
P-Value	7.7e-02	3.0e-01	2.0e-02	6.0e-02	7.2e-02

This table presents the performance comparison of the Ankh and TooT-PLM-P2S models for ion channel classification, using cross-validation. The metrics reported include accuracy, precision, recall, F1 score, and Matthews correlation coefficient (MCC). Each metric's p-value is provided to indicate statistical significance. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. Boldface values denote the higher performance between the models.

The Ankh and TooT-PLM-P2S models show comparable accuracy, with no statistically significant difference. The TooT-PLM-P2S model achieves higher precision, but this difference is also not statistically significant. The Ankh model has higher recall, though the difference is not statistically significant. The Ankh model outperforms the TooT-PLM-P2S model in F1 score and MCC, with both differences being statistically significant.

C.1.5 Transporter Classification

This section compares the performance of the Ankh and TooT-PLM-P2S models for transporter classification. The evaluation metrics, including accuracy, precision, recall, F1 score, and MCC, are summarized in Table 78.

Table 78: Transporter Classification: Performance Comparison

	accuracy	Precision	Recall	F1	MCC
Ankh	**0.89 ± 0.02**	**0.86 ± 0.03**	0.90 ± 0.01	**0.88 ± 0.02**	**0.78 ± 0.03**
TooT-PLM-P2S	0.84 ± 0.03	0.79 ± 0.05	0.88 ± 0.03	0.83 ± 0.03	0.69 ± 0.05
P-Value	1.1e-02	2.6e-02	8.1e-02	5.8e-03	7.7e-03

This table presents the performance comparison of the Ankh and TooT-PLM-P2S models for transporter classification, using cross-validation. The metrics reported include accuracy, precision, recall, F1 score, and Matthews correlation coefficient (MCC). Each metric's p-value is provided to indicate statistical significance. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. Boldface values denote the higher performance between the models.

The Ankh model demonstrates higher accuracy than the TooT-PLM-P2S model, but this difference is not statistically significant. Precision is also higher for the Ankh model, yet this difference is not statistically significant either. However, the Ankh model significantly outperforms the TooT-PLM-P2S model in recall, F1 score, and MCC, as indicated by the statistically significant p-values.

C.1.6 Membrane Protein Classification

In this section, we compare the performance of the Ankh and TooT-PLM-P2S models for membrane protein classification. The evaluation metrics, including accuracy, precision, recall, F1 score, and MCC, along with their respective p-values to assess statistical significance, are presented in Table 79.

Table 79: Membrane Protein Classification: Performance Comparison

	accuracy	Precision	Recall	F1	MCC
Ankh	0.924 ± 0.005	0.900 ± 0.016	**0.958 ± 0.012**	0.928 ± 0.004	0.851 ± 0.009
TooT-PLM-P2S	0.921 ± 0.004	**0.919 ± 0.013**	0.926 ± 0.016	0.922 ± 0.004	0.842 ± 0.008
P-Value	1.90e-01	9.30e-03	4.40e-02	7.60e-02	1.00e-01

This table presents the performance comparison of the Ankh and TooT-PLM-P2S models for membrane protein classification. The evaluation metrics include accuracy, precision, recall, F1 score, and MCC, with corresponding p-values indicating statistical significance. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. Boldface values highlight the higher performance between the models.

The accuracy of the Ankh and TooT-PLM-P2S models is comparable, with a p-value greater than 0.05, indicating no statistically significant difference. The F1 scores of the two models are also comparable, but the p-value for the F1 score is less than 0.05, suggesting statistical significance. The Ankh model shows better performance in recall and MCC compared to the TooT-PLM-P2S model, although the p-values for these metrics are greater than 0.05, indicating that the differences are not statistically significant. The TooT-PLM-P2S model outperforms the Ankh model in precision, with a p-value less than 0.05, indicating statistical significance.

C.1.7 Secondary Structure Prediction

We analyzed the performance of the Ankh and TooT-PLM-P2S models across various non-SSP tasks and now evaluate their performance on Secondary Structure Prediction (SSP) tasks. This section compares the models' ability to predict secondary structure elements of proteins, focusing on SSP-3 and SSP-8 prediction tasks.

SSP-3 prediction involves categorizing protein residues into three secondary structure classes: alpha-helix, beta-strand, and coil. SSP-8 prediction entails a more granular classification into eight distinct secondary structure states.

We present the results for both SSP-3 and SSP-8 predictions, showcasing the evaluation metrics including F1 score, recall, precision, and accuracy for each model. These metrics provide insights into the models' performance in predicting secondary structures, a critical task in bioinformatics.

SSP3 Prediction: We evaluate the performance of the Ankh and TooT-PLM-P2S models on SSP-3 prediction tasks. Table 80 provides a comparison of the models using four evaluation metrics: accuracy, precision, recall, and F1 score. Each metric is accompanied by its respective p-value to assess statistical significance.

Table 80: SSP-3 Prediction: Performance Comparison between the models

	accuracy	Precision	Recall	F1
Ankh	0.83 ± 0.04	0.83 ± 0.05	0.82 ± 0.04	0.83 ± 0.05
TooT-PLM-P2S	0.83 ± 0.04	0.83 ± 0.05	0.82 ± 0.04	0.82 ± 0.05
P-Value	8.9e-01	7.2e-01	2.7e-01	6.0e-01

This table presents the performance comparison of the Ankh and TooT-PLM-P2S models for SSP-3 prediction. The evaluation metrics include accuracy, precision, recall, and F1 score, each with their respective p-values to indicate statistical significance. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant.

According to Table 80, there is no discernible difference between the Ankh and TooT-PLM-P2S models across all evaluation metrics. Both models consistently achieve a value of 82% for accuracy, precision, recall, and F1 score. The p-values for each metric are greater than 0.05, indicating that these differences are not statistically significant. This demonstrates that the

performance of the Ankh and TooT-PLM-P2S models is equivalent on SSP-3 prediction tasks, with no significant variation observed in any of the metrics.

SSP8 Prediction: We evaluate the performance of the Ankh and TooT-PLM-P2S models on SSP-8 prediction tasks. Table 81 provides a comparison of the models using four evaluation metrics: accuracy, precision, recall, and F1 score, with corresponding p-values to assess statistical significance.

Table 81: SSP-8 Prediction: Performance Comparison between the models

	accuracy	**Precision**	**Recall**	**F1**
Ankh	**0.71 ± 0.05**	0.59 ± 0.11	**0.46 ± 0.06**	**0.49 ± 0.07**
TooT-PLM-P2S	0.71 ± 0.05	0.61 ± 0.10	0.45 ± 0.05	0.48 ± 0.07
P-Value	1.8e-05	2.3e-01	2.1e-05	1.0e-04

This table presents the performance comparison of the Ankh and TooT-PLM-P2S models for SSP-8 prediction. The evaluation metrics include accuracy, precision, recall, and F1 score, each with their respective p-values to indicate statistical significance. The threshold for statistical significance is set at a p-value of 0.05, meaning that p-values less than 0.05 are considered statistically significant. Boldface values highlight the higher performance between the models.

According to Table 81, the Ankh model outperforms the TooT-PLM-P2S model in all evaluation metrics except precision. Specifically, the Ankh model achieves higher values in accuracy, recall, and F1 score. The p-values for these metrics are less than 0.05, indicating that the differences are statistically significant. This demonstrates the superior performance of the Ankh model over the TooT-PLM-P2S model in terms of accuracy, recall, and F1 score for SSP-8 prediction.

Conversely, the TooT-PLM-P2S model outperforms the Ankh model in precision. However, the p-value for precision is greater than 0.05, indicating that this difference is not statistically significant. Therefore, while the Ankh model shows statistically significant improvements in accuracy, recall, and F1 score, the advantage of the TooT-PLM-P2S model in precision is not statistically significant.