# IMPROVING DEEPKINZERO WITH PROTEIN LANGUAGE MODELS AND TRANSDUCTIVE LEARNING

by
EMİNE AYŞE SUNAR

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of Master of Science

Sabancı University
July 2024

# ABSTRACT

IMPROVING DEEPKINZERO WITH PROTEIN LANGUAGE MODELS AND TRANSDUCTIVE LEARNING

EMİNE AYŞE SUNAR

Computer Science and Engineering M.Sc. THESIS, July 2024

Thesis Supervisor: ASSOC. PROF. OZNUR TASTAN

Keywords: Benchmark Dataset, Protein Language Models, Kinases, Phosphorylation, Zero-Shot Learning, Transductive Learning

Phosphorylation is a critical post-translational modification that regulates numerous cellular processes, including cell signaling. Kinases are the enzymes responsible for catalyzing phosphorylation events. Due to their essential roles in the cell, kinases are the major drug targets. The amino acid residue that receives the phosphate in the substrate protein is termed a phosphosite. While high-throughput experimental techniques can detect phosphosites, identifying the specific kinases that phosphorylate these sites remains challenging. Computational methods, which typically rely on supervised techniques and existing training data, fall short for understudied kinases, also known as dark kinases, due to insufficient examples for training.

Our research group previously addressed this data limitation by framing the prediction of dark kinases as a zero-shot learning problem and introduced DeepKinZero. DeepKinZero takes the phosphosite and its surrounding sequence and kinase attributes and transfers knowledge from well-studied kinases to understudied kinases to make predictions. In this thesis, we aim to enhance DeepKinZero in several aspects. Firstly, we present a new evaluation setup where the evaluation splitting strategy takes into account not only the zero-shot nature of the problem but also the kinase group memberships, and kinase sequence similarities. This benchmark dataset, DARKIN, serves as a challenging and valuable benchmark designed to accurately assess zero-shot learning performance for dark kinase-phosphosite prediction tasks.

Secondly, we improve the protein sequence representation by evaluating various protein language models in this task. As part of this study, two zero-shot models—a zero-shot k-NN model and a zero-shot bi-linear model—have been presented to benchmark the representation power of protein language models. Thirdly, we demonstrate that using kinase active sites can be as effective as using the entire kinase domain. These active sites slightly surpass the performance of the original DeepKinZero model. Additionally, we explore a transductive approach and pseudo-labeling strategies to leverage the known phosphosite sequences of the unlabeled phosphosites.

# ÖZET

PROTEİN DİL MODELLERİ VE TRANSDÜKTİF ÖĞRENME İLE DEEPKINZERO'YU İYİLEŞTİRME

EMİNE AYŞE SUNAR

Bilgisayar Bilimi ve Mühendisliği Yüksek Lisans TEZİ, Temmuz 2024

Tez Danışmanı: DOÇ. DR. ÖZNUR TAŞTAN

Anahtar Kelimeler: Denek Seti, Protein Dil Modelleri, Kinazlar, Fosforilasyon, Sıfır-Örnekli Öğrenme, Transdüktif Öğrenme

Fosforilasyon, hücre sinyalizasyonu da dahil olmak üzere birçok hücresel süreci düzenleyen kritik bir protein çevrimi sonrası değişimdir. Kinazlar, fosforilasyon olaylarını katalize eden enzimlerdir. Hücre içindeki önemli rolleri nedeniyle kinazlar başlıca ilaç hedefleridir. Sübstrat proteininde, fosfat grubunun bağlandığı amino asit fosfosit olarak adlandırılır. Yüksek verimli deneysel teknikler fosfositleri tespit edebilirken, bu bölgeleri fosforile eden spesifik kinazları tanımlamak hala zorlayıcı bir problemdir. Genel olarak denetimli öğrenme tekniklerine ve mevcut deneysel olarak ispatlanmış veri setlerine dayanan hesaplamalı yöntemler, yeterince örnek olmaması nedeniyle az çalışılmış kinazlar (karanlık kinazlar olarak da adlandırılır) için yetersiz kalmaktadır.

Araştırma grubumuz daha önceden bu veri kısıtını ele alarak karanlık kinazların tahminini sıfır örnekli öğrenme problemi olarak çerçevelemiş ve DeepKinZero modelini tanıtmıştı. DeepKinZero, fosfosit ve çevresindeki diziyi ve kinaz özelliklerini kullanarak çok çalışılmış kinazlardan az çalışılmış kinazlara bilgi aktararak tahminler yapar. Bu çalışmada, DeepKinZero'yu çeşitli yönlerden geliştirmeyi amaçlıyoruz. Öncelikle, problemin sıfır örnekli yapısına ek olarak kinaz grup üyeliklerini ve kinaz dizi benzerliklerini de ele alan yeni bir değerlendirme kurulumu sunuyoruz, başka bir ifadeyle, bu stratejileri ele alan yeni bir denek seti sunuyoruz. DARKIN ismini verdiğimiz bu denek seti, sıfır örnekli bir kurulumda karanlık kinazların kinaz-fosfosit tahminini doğru bir şekilde yapabilmek için tasarlanmış zorlayıcı ve değerli bir denek seti olarak işlev görür.

İkinci olarak, protein dizilerini temsil eden vektörleri bu kurulumda çeşitli protein dil modellerini değerlendirerek geliştiriyoruz. Çalışmamız dahilinde, protein dil modellerinin temsil gücünü kıyaslamak için sıfır örnekli k-NN modeli ve sıfır örnekli ikili doğrusal model olmak üzere iki tane sıfır örnekli model sunuyoruz. Üçüncü olarak, kinaz aktif bölgelerinin kullanılmasının, tüm kinaz alanının kullanılması kadar etkili olabileceğini gösteriyoruz. Kinaz aktif bölgeleri kullanılarak eğitilen bu modelin orjinal DeepKinZero performansını kısmen geçebildiğini gösteriyoruz. Ayrıca, etiketlenmemiş fosfositlerin bilinen fosfosit dizilerinden yararlanmak için transdüktif öğrenme ve sözde-etiketleme stratejilerini kullandığımız iki modeli DeepKinZero kurulumuna entegre ederek deneyler gerçekleştiriyoruz.

## ACKNOWLEDGEMENTS

Firstly, I would like to express my deepest gratitude to my wonderful supervisor, Assoc. Prof. Oznur Tastan, for her continuous help, guidance, support, kindness, and encouragement throughout the entire process of this thesis. Studying under her mentorship has been an exceptional and enriching experience, and I have learned invaluable lessons from her wisdom, creativity, and expertise.

I would also like to express my deepest gratitude to my co-advisor, Assoc. Prof. Ramazan Gokberk Cinbis, for his dedication and especially for his theoretical support throughout this process. His guidance, kindness, humility, and creative thoughts have enlightened me as a researcher.

Through their deep knowledge, success, and expertise, both my supervisors, Assoc. Prof. Oznur Tastan and Assoc. Prof. Ramazan Gokberk Cinbis, have been exceptional role models. Despite their remarkable achievements and extensive knowledge, they have always remained incredibly humble and kind, making them truly inspirational figures for me.

Furthermore, I would like to thank my colleagues and project mates who were also a part of the Tubitak project 122E500, Mert Pekey and Zeynep Isik, who have significantly contributed to this research study. I would like to explicitly mention their contributions to this project:

**Mert Pekey:** For converting the original DeepKinZero repository from TensorFlow to PyTorch, making it an expandable and easy-to-use repository for us. His contributions were crucial in the experimental runs, particularly for half of the results for the Bi-linear Zero-Shot model (BZSM) in Sections 6.2.2, Section 6.2.4 and Section 6.3

**Zeynep Isik:** For implementing the zero-shot k-NN model and for all the results associated with the k-NN model in Sections 6.2.2 and 6.2.3. Additionally, she conducted extensive research on protein language models (pLMs) and was responsible for the extraction of these embeddings.

I would like to express my deep gratitude to the Scientific and Technological Research Council of Turkey (TÜBİTAK) for their funding of project 122E500, which my thesis was a part of. The numerical calculations reported in this thesis were fully/partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center

(TRUBA resources).

I am also grateful for the support of the FENS Conference Travel Grant, which enabled my participation in ICLR 2024 MLGenX and contributed to the research presented in this thesis.

Additionally, I would like to express my sincere appreciation to the members of the jury, Prof. Arzucan Ozgur and Asst. Prof. Onur Varol, who dedicated their time, expertise, and valuable insights to evaluate and assess this thesis.

I am deeply grateful to my friends for their unwavering support throughout my masters journey. I want to express my heartfelt thanks to Meryem Çiçek, Alperen Ustaömer, Zeynep Işık, Neda Ahmed Gamal Mohamed, Şahd Şerif and Arghavan Sharafi for their encouragement, emotional support, insightful discussions, and heartwarming companionship. Each of you has been a source of comfort, joy, and strength, making this journey more bearable and memorable.

I also extend my heartfelt thanks to my family—my mother, Sevil Sunar; my father, Mehmet Sunar; my sisters, Rabia Merve Sunar, Zeynep Elife Sunar, and Fatma Büşra Sunar; and my grandparents, Mehtap Kılıç, Besim Kılıç and Pembe Sunar. Your endless support, encouragement, and love have been my foundation, and I am profoundly grateful for your unwavering belief in me and for always prioritizing my well-being.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# 1. INTRODUCTION

Protein phosphorylation is a key biological process that regulates various biochemical and cellular activities by serving as a switch for controlling protein function (Fischer and Krebs, 1955). These phosphorylation events are critical in regulating multiple cellular processes, including signal transduction, cell division, and metabolism (Cohen, 2000). Phosphorylation events can activate or inhibit protein function, alter protein-protein interactions, and modulate protein stability and localization. Protein phosphorylation involves adding a phosphate group from adenosine triphosphate (ATP) to specific amino acid residue locations on a protein substrate (Cohen, 2002). The substrate protein's specific amino acid residue locations that accept the phosphate are called a *phosphorylation site* or a *phosphosite*. The phosphosite residues are serine, threonine, or tyrosine amino acids, and there are also histidine residues that can be phosphorylated (Xu and Wang, 2021).

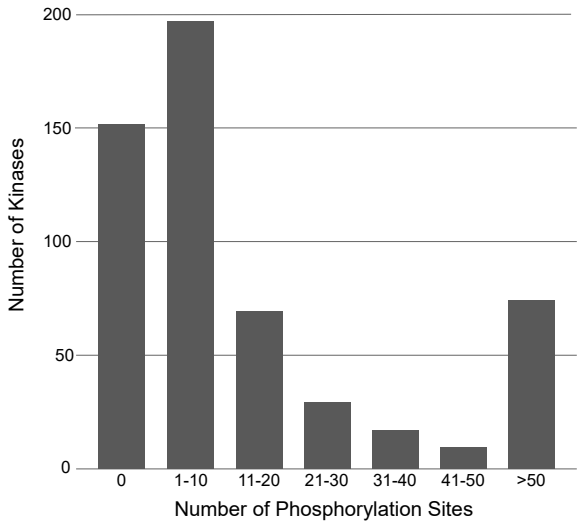
Kinases are the enzymes that mediate the phosphorylation events (Hunter, 1995). The human genome encodes over 500 kinases, collectively referred to as the *kinome*, underscoring the diversity and complexity of these enzymes. Kinases perform this process in a site-specific manner. Thus, a kinase can catalyze only a certain subset of substrate proteins. This specificity is highly dependent on the protein sequence where the site resides (Bradley and Beltrao, 2019; Safaei et al., 2011).

Malfunction in phosphorylation is frequently implicated in various diseases, including cancer, Parkinson's disease, cardiovascular diseases, and inflammatory disorders (Blume-Jensen and Hunter, 2001; Wang et al., 2012; Heineke and Molkentin, 2006; Schieven, 2005). Dysregulated kinase signaling is particularly characteristic of many cancers, prompting the development of kinase inhibitors as targeted therapies. These inhibitors aim to specifically block the aberrant kinase activity driving tumor growth and survival, offering a more precise treatment approach compared to traditional chemotherapy (Cohen et al., 2021).

Even though phosphorylation is an important process and its dysregulation is known to cause several diseases, there is still a considerable amount of information missing regarding kinase-phosphosite association data. For 95% of human phosphosites

identified, the associated cognate kinase is unknown. Around 25% of kinases have no identified phosphosite associations, and around 35% of kinases have 10 or fewer known phosphosite associations (Needham et al., 2019). Figure 1.1 shows the histogram of the phosphosite association count for human kinases (Data downloaded in May 2023, from PhosphoSitePlus) (Hornbeck et al., 2014). Approximately 150 kinases have no known phosphosite associations, and around 195 kinases have only 1-10 known phosphosite associations. This considerable knowledge gap underscores the critical need for enhanced research efforts.

Figure 1.1 Histogram of the number of experimentally validated phosphosites for all human kinases.




There has been further research to show the importance of understudied kinases. Essegian et al. (2020) developed the Clinical Kinase Index (CKI) to highlight the significance of these kinases in cancer treatment. Employing parameters such as survival analysis and clinical-pathological correlations from the Cancer Genome Atlas (TCGA) (Network et al., 2012), CKI ranks kinases based on their clinical relevance and potential. The approach is validated by demonstrating that CKI effectively identifies kinases already in clinical trials. This research underscores the importance of studying these understudied kinases, emphasizing the need for further exploration and information gathering to better understand their roles in cancer. Inspired by the CKI, Vella et al. (2022) highlighted the significance of LMTK3, a dark kinase identified as a promising therapeutic target for breast cancer treatment. Their study made substantial progress in understanding the functional roles of this understudied kinase, demonstrating its potential in cancer therapy.

As it has also been studied in the literature (Essegian et al., 2020; Vella et al., 2022), addressing kinase-related diseases is critical, yet much of the phosphoryla-

tion data relies on experimental methods that are both costly and time-consuming. Consequently, computational methods emerge to accelerate experimental efforts by assigning kinases to orphan phosphosites.

There have been several computational studies on phosphorylation data, focusing on the problems of discovering phosphosites, identifying kinase-specific phosphosites, and finding the kinase associated with a phosphosite (Blom et al., 1999; Cortes and Vapnik, 1995; Wang et al., 2022; Gao et al., 2010; Luo et al., 2019; Xue et al., 2008; Patrick et al., 2015; Xue et al., 2008; Wang et al., 2020; Chen et al., 2023; Linding et al., 2007; Zou et al., 2013; Ma et al., 2020). Most of these studies use kinase-phosphosite association data that has been previously validated through experimentation and computes position-specific amino acid preferences in the form of position-specific matrices (Altschul et al., 1997). To reliably estimate these kinase-specific binding preferences, there must be an ample amount of phosphosites for a kinase. Several other studies have experimented with supervised learning techniques, and they also require a substantial amount of training data to make accurate predictions for a kinase.

Although these methods have shown valuable results and improvements, the need for a large number of examples for a kinase limits their applicability for the understudied dark kinases, as there are no or few phosphosites known for these kinases. That is also the reason these computational methods focus solely on kinases with a large number of examples. Consequently, methods that can transfer information from well-studied kinases to understudied kinases emerge as a favorable approach, given the limitations of experimental data. The first study formulating the problem as a zero-shot learning approach was DeepKinZero, implemented by Deznabi et al. (2020), an earlier work of our group. DeepKinZero uses zero-shot learning to predict the possible kinases that phosphorylate a given phosphosite.

Zero-shot learning is a machine learning technique where the model predicts classes it has not seen during training (Xian et al., 2018; Wang et al., 2019; Palatucci et al., 2009; Zhang and Saligrama, 2015). This technique is particularly useful in the context of phosphorylation data, given that there exist kinases with no known sites and orphan sites with unknown associated kinases. In the context of kinase-phosphosite prediction, the features of well-studied kinases and phosphosites can be leveraged to transfer information to less-studied kinases, enabling the prediction of kinase-phosphosite associations without the need for direct experimental data for every possible interaction. By transferring information from well-studied kinases to less-studied ones, zero-shot learning emerges as a powerful computational tool for bridging the knowledge gaps in phosphorylation research.

This study contributes a new, reproducible dataset focused on phosphorylation data, designed for zero-shot learning.

Protein language models (pLMs) are large language models trained on extensive corpora derived from major protein databases. These models are designed to decipher the complex semantic and intrinsic properties embedded within protein sequences—details that are typically challenging to interpret manually (Rao et al., 2019; Elnaggar et al., 2021; Rives et al., 2020; Meier et al., 2021; Lin et al., 2022; Brandes et al., 2022; Ferruz et al., 2022; Geffen et al., 2022; Elnaggar et al., 2023; Su et al., 2023; Zhang and Okumura, 2024). Owing to their broad applicability and proven success across various areas of computational biology, several of these pLMs are benchmarked in this study to address the challenge of identifying the cognate kinases for orphan phosphosites. In addition to tackling this problem, the performance of these pLMs is benchmarked on this newly curated dataset.

Transductive learning is a sub-field of machine learning that focuses on making predictions on a fixed set of test samples by making use of their available features during training time (Wan et al., 2019; Lin et al., 2021b; Song et al., 2018; Xie et al., 2021; Bo et al., 2021; Ye and Guo, 2019; Li et al., 2019). Since the majority of known phosphosites are orphan sites, even though their cognate kinases are not known, their features can be used during training. As a result, transductive learning emerges as a natural solution for this task.

The first half of this thesis study aimed to create a biologically relevant phosphorylation dataset in a zero-shot learning setup. Creating this dataset in a zero-shot learning setup is particularly important since many kinases do not have any reported site associations, thus the prediction model should be robust to kinases it has not seen during training. Following the dataset creation, the second half of this study focused on developing two zero-shot models—a zero-shot k-NN model and a zero-shot bilinear model—to predict the most probable kinase to phosphorylate a given phosphosite. These models were then used to benchmark the performance of several protein language models on the DARKIN benchmark dataset. Subsequently, the best-performing protein language models were tested in the previous study of Deznabi et al. (2020) to evaluate their performance. Finally, transductive methods were applied to this setup.

The brief summary of the contributions of this thesis is as follows:

- We curated a biologically relevant, zero-shot dataset of kinase-phosphosite association data, termed DARKIN, which is publicly available[1].

[1] https://github.com/tastanlab/darkin

- We provide two simple zero-shot models, the zero-shot k-NN model and the zero-shot bi-linear model, in which we conduct a comprehensive evaluation of the protein language models to benchmark their performance on the DARKIN dataset.
- We experimented with and share the results of a transductive model (Quasi-Fully Supervised Model) and a pseudo-labeling model, aiming to make use of the orphan phosphosites.

The remainder of this thesis is organized as follows:

- Chapter 2 provides background information on the details of phosphorylation events. It then provides a comprehensive review of previously studied computational models for kinase-phosphosite association prediction. This review includes a detailed explanation of DeepKinZero, the method which this thesis expands upon. Additionally, it covers prominent protein language models and several transductive learning models.
- Chapter 3 first explains the problem formulation and provides detailed information on the DeepKinZero model. Later in this chapter, Section 3.2 explains the details of the data gathering and curation processes, as well as the description of the dataset splitting script and its strategy.
- Chapter 4 provides a detailed explanation of the protein embeddings, including both the baseline and the protein language model (pLM) embeddings. It lists all the protein representations that will be experimented with, presented in Table 4.1. Later on in this chapter, Section 4.2, explains the zero-shot adaptation of the k-NN model, and Section 4.3 discusses the implementation of the bi-linear zero-shot model used to test the effectiveness of the aforementioned protein representations.
- Chapter 5 details the transductive model, the Quasi-Fully Supervised Model (QFSM), and the semi-supervised learning approaches, specifically progressive pseudo-labeling, which were experimented with in this setup with the aim of leveraging orphan phosphosites for potential performance improvements.
- In Chapter 6, we first present the dataset statistics for the DARKIN benchmark dataset for split 1, which is the default split used in this study. Subsequently, we evaluate the protein language models tested on both the zero-shot k-NN model and the zero-shot bi-linear model, identifying the best performing pLM. In Section 6.4, the comparison between kinase domain and active site representations is presented. Later, in Section 6.5, the results of the Quasi-

Fully Supervised Model are reported. Following this, in Section 6.6, the results of the pseudo-labeling approach are discussed. Finally in Section 6.7 an error analysis on the QFSM and pseudo-labeling approach is conducted.

- We conclude our work and discuss future directions in Chapter 7.

# 2. BACKGROUND AND LITERATURE REVIEW

This chapter will review the resources for experimentally validated phosphorylation data, the computational methods used on phosphorylation data for phosphorylation data prediction, protein language models, and transductive and semi-supervised learning methods that will be applied to phosphorylation data in the problem formulation of this thesis study.

## 2.1 Background on Phosphorylation and Kinases

In this subsection, we provide background information on phosphorylation, kinases, and the available phosphorylation data databases.

### 2.1.1 Phosphorylation

Protein phosphorylation is among the important and well-studied post-translational modifications where a phosphate group is attached to a residue in the substrate. The residue that accepts the phosphate group is termed the *phosphosite* (Fischer and Krebs, 1955). There are several techniques through which phosphorylation data is collected, including immunoblotting, direct staining, isotopic labeling, and mass spectrometry (MS), each with its own applications and limitations (Delom and Chevet, 2006). Among these methods, mass spectrometry is the most widely used approach due to its high sensitivity and specificity. Additionally, mass spectrometry enables the identification of phosphorylation sites across a wide range of samples, which is essential for collecting phosphorylation data (Delom and Chevet, 2006). For this reason, mass spectrometry is the principal method for collecting experimentally validated phosphorylation data.

As mentioned previously, the phosphosite is the specific site on a protein where the phosphorylation event occurs. The protein where the phosphosite resides is also referred to as the substrate protein. The phosphosite residue, typically an amino

acid, receives a phosphate group from the catalyzing enzyme (Fischer and Krebs, 1955). In this study, the phosphosite is represented as a sequence of 15 amino acids, with the phosphosite positioned at the center. This means the phosphosite's seven neighboring amino acids are on both sides. Phosphorylation can occur in multiple regions of a protein substrate, influencing various biological processes (Tyanova et al., 2013).

Active sites on a protein are often preserved across different species through evolution (Liang et al., 2006). In their study, McDonald et al. (2018) show that phosphosites are also preserved across different species, indicating the importance of these regions on a protein. This conservation underscores the evolutionary significance of phosphosites and their potential to enlighten the understanding of cellular mechanisms and functions.

### 2.1.2 Kinases

Kinases are the enzymes that catalyze phosphorylation reactions (Hunter, 1995). There are various types of kinases, such as protein kinases and lipid kinases; however, this study will focus on protein kinases (Yang et al., 2008). Protein kinases consist of multiple domains that contribute to different functional activities of the kinase. Key domains include the kinase domain, which is essential for phosphorylation; ATP-binding sites, crucial for enzymatic activity; and protein-protein interaction domains, such as SH2 or PH domains, which facilitate interactions with other molecules (Krupa and Srinivasan, 2002; Röhm et al., 2021). Given the focus of this study on phosphorylation, the primary domain of interest is the kinase domain.

With over 500 proteins, the human kinases perform various functions to regulate the cell. Manning et al. (2002) defined a set of 518 human protein kinases and categorized this set into 10 groups and 116 families based on similarities in their kinase domains. This categorization technique has been one of the most valid and widely accepted methods. There are several other systems used for classifying kinases, one of which involves the enzyme commission numbers (EC numbers). EC numbers provide a classification based on the chemical reactions that enzymes catalyze and are retrieved from the ENZYME database (Bairoch, 2000). Another approach for classifying kinases involves the KEGG pathways they participate in, highlighting the functional roles these enzymes play in various biochemical processes (Kanehisa et al., 2016).

### 2.1.3 Available Phosphorylation Databases

There exist several phosphorylation data databases such as PhosphoSitePlus, EPSD (Eukaryotic Phosphorylation Sites Database), PHOSIDA (Phosphorylation site database), Phospho.ELM, PhosphoGRID, and PhosphoPep (Hornbeck et al., 2015; Lin et al., 2021a; Gnad et al., 2007; Diella et al., 2004; Stark et al., 2010; Bodenmiller et al., 2008). These databases differ subtly in their scope and focus; for example, Hornbeck et al. (2015) has the most extensive source of data, including both human and non-human kinases and sites, yet it restricts its coverage to eukaryotic organisms. Stark et al. (2010) focuses on *in vivo* experiments (experiments done on living organisms). Databases such as Lin et al. (2021a) and Gnad et al. (2007) also provide additional information such as the functional contexts of these sites.

## 2.2 Computational Methods on Kinase-Phosphosite Association

### Prediction

Since determining the cognate kinase of a phosphosite is time-consuming and costly, several computational methods have been developed to predict kinase-phosphosite associations. This section will introduce and explain the computational methods to predict kinase-phosphosite interactions. First, the models implemented for phosphosite prediction will be introduced. While this thesis study does not focus specifically on the problem of site prediction, it is valuable to study and explore these methods to gain a better understanding of the improvements in the field and the techniques being used. Subsequently, models that specifically predict kinases will be introduced. Finally, the zero-shot learning models in this field will be explained, which is the primary focus of this thesis study.

### 2.2.1 Phosphosite Prediction Models

In this subsection, we provide a literature review of models focused on phosphosite prediction, specifically those focused on identifying phosphosites.

#### 2.2.1.1 General Phosphosite Prediction Models

The phosphosite prediction models aim to recognize phosphosites in a given protein sequence. Thus, the input is the protein sequence, optionally accompanied by other

functional and structural information, and the output is whether a phosphosite exists in the protein sequence or not.

One of the earliest computational methods in this field was developed by Blom et al. (1999), who introduced NetPhos, an artificial neural network specifically designed for predicting phosphosites. This model leverages both protein sequence and structural features to enhance prediction accuracy. They also use varying window sizes and datasets to increase accuracy. NetPhos specifically focuses on predicting tyrosine, serine, and threonine phosphorylation sites, generating sequence logos for these three categories of phosphosites to recognize the frequent amino acid residues present at these sites.

PhosphoSVM, introduced by Dou et al. (2014), is a phosphorylation site prediction tool based on support vector machines (SVM) (Cortes and Vapnik, 1995). PhosphoSVM uses sequence-level features to characterize the input protein sequence: these include i) conservation of the site as measured by Shannon entropy (SE) (Shannon, 1948), relative entropy (RE) (Kullback and Leibler, 1951) of the positions, ii) structural feature predictions based on sequence, including protein secondary structure (SS) (Garnier et al., 1978), predicted protein disorder (PD) (Dunker et al., 2000), solvent accessible area (ASA) (Lee and Richards, 1971), overlapping properties (OP) (Wu and Brutlag, 1995), averaged cumulative hydrophobicity (ACH) (Sweet and Eisenberg, 1983) and k-nearest neighbor (k-NN) (Cover and Hart, 1967). PhosphoSVM demonstrated high accuracy in predicting phosphorylation sites across species it had not been trained in, reflecting its ability to generalize to diverse species. Notably, the k-NN attribute significantly enhanced prediction accuracy.

One of the recent improvements brought into the field of general phosphosite prediction is TransPhos (2022), developed by Wang et al. (2022), which employs a transformer-encoder architecture combined with a densely connected neural network for general phosphosite prediction. Through experimentation they have shown that TransPhos shows significant improvements over previously used approaches that employ LSTM, RNN, and CNN architectures (Graves and Graves, 2012; Elman, 1990; Rumelhart et al., 1986; Jordan, 1997; LeCun et al., 2015) by achieving AUC values of 0.8579 for serine, 0.8335 for threonine, and 0.6953 for tyrosine, which shows the significance of using transformer-encoder based models in identifying phosphosites.

#### 2.2.1.2 Kinase-Specific Phosphosite Prediction Models

Kinase-specific phosphosite prediction models can be further divided into two categories: models that make site predictions directly according to individual kinases,

and models that make predictions based on kinase groups and families.

### Single Kinase-Specific Phosphosite Prediction Models

An early tool developed for site prediction is Musite (2010), introduced by Gao et al. (2010). Musite aims to predict both general and kinase-specific phosphorylation sites. They approach the phosphosite prediction problem as a binary classification problem. They first collect data from six organisms from Uniprot (Consortium, 2018). These collected sites are positive samples. To train the classifier model, they also generate negative samples using the same positive site samples, but by removing the specific site location. They acknowledge that there is a risk these negative samples might turn out to be actual positive sites, however since it is a very small probability, they neglect the risk. Later on, they train separate SVM (Cortes and Vapnik, 1995) models for each organism, using amino acid frequency around the site, k-NN scores, and a protein disorder predictor as their main features. k-NN scores are calculated by finding the k most locally similar samples from both positive and negative samples, then they get the k nearest neighbors from the k positive and k negative samples combined, and the percentage of the positive closest neighbors gives the k-NN score. They take the k-NN features for several different values of k. To make predictions for query samples, they average the results of all the trained SVMs.

DeepPhos (Luo et al., 2019) uses a densely connected deep neural network for general and kinase-specific phosphosite prediction. DeepPhos specifically uses the densely connected CNN architecture (DC-CNN) (LeCun et al., 2015), where convolutional layers are connected, right after one another with intra-block concatenation, hence concatenating the output of one layer with the input of the subsequent layer. In the final prediction layer they use inter-block concatenation, where the output of any previous convolutional layer block from any layer could be concatenated with its input layer. By training this DC-CNN model, they learn high-dimensional vector representations of protein sequences, which they eventually use for phosphorylation site prediction. A key feature that differentiates DeepPhos from previous approaches is its use of transfer learning. They fine-tune DeepPhos using kinase-specific phosphosites.

### Kinase Group and Family-Specific Phosphosite Prediction Models

One of the earliest models to apply kinase hierarchy-based phosphosite prediction is the GPS 2.0 (Xue et al., 2008) model. GPS 2.0 can predict kinase-specific phosphosites for 408 kinases using a hierarchical structure with four levels: group, family, subfamily, and single PK. A prominent contribution of the GPS 2.0 model is its

ability to identify Aurora-B-specific substrates. Later, GPS 5.0 was developed, with advancements over its predecessor, GPS 2.0, in predicting kinase-specific phosphosites (Wang et al., 2020). GPS 5.0 supports predictions for a wider range of human kinases, covering 489. It also introduces novel methods such as Position Weight Determination (PWD) and Scoring Matrix Optimization (SMO), resulting in higher prediction accuracy compared to its predecessor. Recently, GPS 6.0 has been introduced, offering improvements over previous versions (GPS 2.0 and GPS 5.0). GPS 6.0 integrates novel methods that have proven to significantly boost performance, including Penalized Logistic Regression (PLR), Deep Neural Network (DNN), and Light Gradient Boosting Machine (LightGBM) algorithms (Chen et al., 2023; Park and Hastie, 2008; LeCun et al., 2015; Fan et al., 2019).

Phosphopick (2015) is a model designed to predict phosphosites for given kinase groups (Patrick et al., 2015). They integrate features such as cellular context—including both environmental and biological factors that influence protein behavior and function—along with protein-protein interactions (PPI) and variations in protein quantities throughout the cell cycle. They point out that other models, which do not use cellular context information, often fall short in identifying sites for specific kinase binding motifs, leading to high false positive rates. To overcome this issue, Phosphopick combines PPI data with cell-specific protein abundance and cellular context information.

### 2.2.2 Kinase Assignment Prediction Models

Focused on solving a similar problem, there are also models specifically designed to predict which kinase is most likely to phosphorylate a given site. This approach reverses the typical site prediction problem by focusing on identifying the kinase instead of the site itself. This subsection will provide a detailed overview of the methodologies centered on kinase assignment prediction. In this sub-section, the kinase prediction models will be categorized into two sections:

#### 2.2.2.1 Kinase Assignment Prediction Models in the Conventional Setup (Non-Zero-Shot Based Approaches)

One of the earlier efforts in identifying the cognate kinase of a given phosphosite, NetworKIN (2007), developed by Linding et al. (2007) is a computational framework that combines consensus substrate motifs, which are patterns in the substrate protein that are recognized by specific kinases and context modeling, which is the

protein-protein interaction data. They integrate data from the STRING database, which provides probabilistic interaction scores for protein-protein interactions (Szklarczyk et al., 2019). They also employ Position-Specific Scoring Matrices (PSSMs) (Altschul et al., 1997) from Scansite, which provides a mathematical representation of the likelihood of an amino acid to belong to a specific location (Obenauer et al., 2003).

A later effort for predicting kinases for experimentally validated phosphosites is PKIS (2013), developed by Zou et al. (2013). This machine learning-based approach uses the composition of monomer spectrum (CMS) encodings, an encoding strategy based on the frequency of amino acids in a sequence. They employ SVMs in their approach (Cortes and Vapnik, 1995).

A later model, KSP, combines both network-based and sequence-based architectures to predict the kinase responsible for phosphorylating a given site (Ma et al., 2020). Ma et al. (2020) highlights the large corpus of phosphosites identified by phosphoproteomic technologies, many of which are not yet associated with a cognate kinase. To address this problem, they developed a model aimed at predicting the kinases that phosphorylate these sites. The model integrates protein-protein interaction (PPI) data and substrate-kinase relationships. Given a phosphosite, KSP returns a ranked list of kinases most likely to phosphorylate the site by defining an affinity score with the site for each of the kinases.

#### 2.2.2.2 Zero-Shot Based Kinase Prediction Models

##### 2.2.2.2.1 DeepKinZero

This thesis study extends the previous model of DeepKinZero (Deznabi et al., 2020). For this reason, this section will detail and explain the DeepKinZero model and elaborate on its core objectives.

The DeepKinZero model is a deep learning model, which specifically uses a bi-directional LSTM architecture, and is built for the zero-shot learning setup. It takes phosphosite embeddings as inputs to predict the kinase class most likely to phosphorylate a given phosphosite. Thus it casts the problem as a multi-class classification task. DeepKinZero learns intrinsic knowledge from well-studied (light) kinase classes during training. It then transfers this knowledge onto the test domain, which comprises kinase classes with sparse phosphorylation data—referred to as dark kinases. In what follows, we will provide the details on DeepKinZero architecture.

**Refining Phosphosite Embeddings with LSTM** The phosphosites are represented as amino acid sequences of length 15, where the specific phosphosite residue is placed in the middle of this sequence. Initially, ProtVec embeddings for these sequences are extracted (Asgari and Mofrad, 2015a). ProtVec embeddings are generated for every trigram, in other words, three continuous residues in a protein sequence. Given that each trigram embedding is of size 100, the resulting representation for a phosphosite is 13x100. These embeddings are then fed into a Bi-LSTM (Graves and Graves, 2012) model after passing through a batch normalization layer. The LSTM layer consists of 512 LSTM cells. The final representation from the LSTM layer is batch normalized again and fed into an attention layer for better refinement and representation.

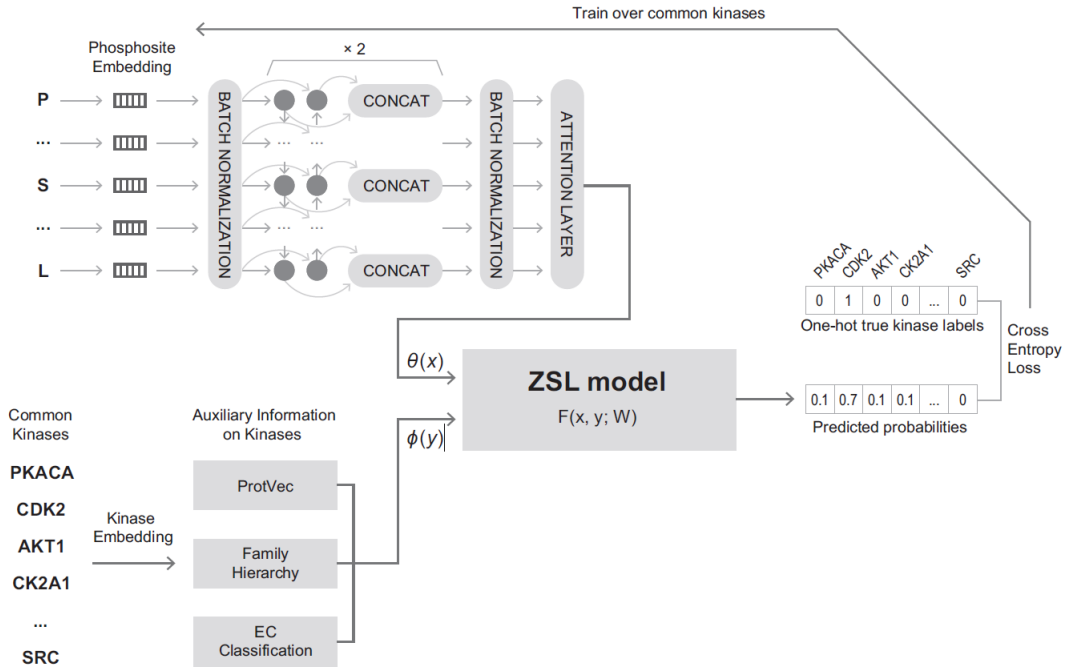
**Kinase Feature Integration to Kinase Representations** For the kinase embeddings, refinement using LSTM is not applied, as several manually curated features hold valuable information about the kinases' functionality. The features for the kinases include the ProtVec representation of the kinase domain sequences, the family and group information of the kinases, and the Enzyme Commission (EC) classification of the kinases. These three features are concatenated to form the final representation of the kinase.

**The Zero-Shot Learning Model** To achieve transfer learning between light and dark kinases, Deznabi et al. (2020) learn a compatibility matrix. In this study, they use the following bi-linear compatibility function proposed by Sumbul et al. (2017):

$$F(x,y) = \sum_{i=1}^{d}\sum_{j=1}^{m} W_{i,j}[\theta(x)]_i[\phi(y)]_j + \sum_{i=1}^{d} W_{i,m}[\theta(x)]_i + \sum_{j=1}^{m} W_{d,j}[\phi(y)]_j + b \tag{2.1}$$

They use this bi-linear function to train the compatibility matrix. The input to this bi-linear compatibility function is the phosphosite and kinase embedding pairs, shown as $\theta(x)$ and $\phi(y)$ respectively, and the output is a scalar which represents how compatible these site and kinase pairs are; in other words, it outputs a score that represents how likely a kinase is to phosphorylate the given phosphosite. Thus, the compatibility matrix is designed to learn the compatibility between the site and kinase embeddings. This score could be assumed as the probability score generated by the model for each kinase, thus the compatibility scores for each kinase is then passed to a Softmax function. As a result, the ZSL model outputs a predicted kinase, which is then evaluated by calculating the cross-entropy loss with the ground truth label for that site. The DeepKinZero architecture is depicted in Figure 2.1.

Figure 2.1 This figure presents the DeepKinZero architecture by Deznabi et al. (2020). The architecture's upper half depicts the phosphosite embeddings refinement with an LSTM layer. The phosphosite embeddings are fed into the LSTM layer, followed by an attention layer to enhance representation and focus. The use of kinase features and their concatenation is shown in the lower-left corner. Finally, the refined phosphosite and kinase embeddings are fed into the zero-shot learning model, which employs the bi-linear function by Sumbul et al. (2017) to learn a compatibility matrix between the kinase and phosphosite embeddings. The predicted kinase by the model is then evaluated using cross-entropy loss with the ground truth labels. (With permission, the figure is reproduced from Deznabi et al. (2020).)




### 2.2.2.3 Recent Developments for Predicting Dark Kinase Activity

Kinase activity prediction is a research field dealing with kinases, which slightly differs in its focus from our setup; it concentrates on determining whether a kinase would phosphorylate a site or not (Casnellie and Krebs, 1984; Casnellie, 1991; Glickman, 2012; Zhang and Daly, 2012; Wiredja et al., 2017; Cann et al., 2017). A recent study that also aims to predict kinase activity, specifically focusing on dark kinases, is Yilmaz et al. (2021). This study developed a functional network, RoKAI, by integrating data from multiple sources, including protein-protein interactions, experimentally validated kinase-substrate data, and gene expression profiles. They use a probabilistic model to make inferences on kinase interactions. This information enables them to be able to transfer knowledge and eventually allows them to transfer information from known data to missing annotations. Thus, this enables

their model to be able to make kinase activity predictions on dark kinases.

Ma et al. (2023) conducted a study focused on predicting phosphorylation sites for understudied kinases. They built a similarity network for kinases by employing STRING confidence scores (Szklarczyk et al., 2019), sequence data, functional data, and protein domains. Through this similarity network, they aimed to integrate this information and transfer knowledge from well-studied kinases to understudied ones. They used experimentally validated data as positive samples to train their predictive models, including support vector machines (SVM) and fully connected neural network-long short-term memory (FCNN-LSTM) models.

Most recently, to specify the specific region of interest for each human kinase, Zhou et al. (2024) developed Phosformer-ST, an explainable transformer model. Phosformer-ST is trained solely on 1D protein sequences. Zhou et al. (2024) use SHapley Additive exPlanation (SHAP) (Lundberg and Lee, 2017) to analyze the intrinsic findings of their transformer model. Despite not being trained in a conventional zero-shot setup, Phosformer-ST has zero-shot prediction capabilities, in other words, it could make predictions on dark kinases due to its multitask learning downstream task training and its use of ESM-2 as its pretrained model. They apply multitask learning using a shared encoder, with the embeddings directed either to a masked language model or to a classifier for kinase-specific site predictions.

### 2.3 Protein Language Models

Protein language models (pLMs) have emerged as powerful tools for vectorizing proteins, enabling the representation of proteins as multidimensional vectors that capture complex biochemical properties, thereby easing advanced computational tasks such as predicting protein interactions, structure, and function, surpassing most traditional approaches.

Alley et al. (2019) applied similar methods as in large language models to learn UniRep, a vector representation of protein sequences, making it a notable mention as one of the leading pLMs in the field. Alley et al. (2019) used advanced deep learning techniques to learn a representation from the protein sequences, capturing intrinsic and complex patterns within the protein. They specifically trained a Multiplicative LSTM (m-LSTM) network with 1900 hidden cells over 24 million UniRef50 amino acid sequences (Krause et al., 2016; Suzek et al., 2007). UniRep brought great improvements to the biological domain, signifying the success of deep learning architectures in this domain.

Among the earlier models, TAPE sets the foundation by establishing benchmarks for training and evaluating protein models (Rao et al., 2019). Rao et al. (2019) claim that there are several datasets and evaluation techniques used in biological settings, however these are not standardized. Thus with the aim of meeting this need, they build a benchmarking framework where they provide standardized tasks. These tasks are selected from broad and distinct areas of the biological domain, making sure to test different fields of studies. They specifically present five distinct semi-supervised learning tasks. Selecting these tasks from different parts of the field ensures the models to be tested and to be able to generalize to the biological domain.

Subsequently, ProtTrans introduced transformer-based architectures, training auto-regressive models and auto-encoder models on a vast corpus of protein sequences, leveraging unsupervised approaches to capture hidden biophysical features from protein sequences (Elnaggar et al., 2021).

Following these advancements, Meta AI introduced the ESM-1b model (Rives et al., 2020). One of the significant contributions of the ESM-1b architecture is its application of unsupervised learning to a large corpus of protein sequences, which enables the model to extract generalized, intrinsic knowledge relevant to a variety of protein-related tasks. Specifically, ESM-1b is trained on 250 million protein sequences, using only sequence data without additional features. The primary objective is to derive generalized information from this extensive corpus. The model treats proteins as sequences of amino acids, adopting a character-based approach to learning, given the limited number (20) of standard amino acids, rather than a word-based approach (Kim et al., 2016; Mikolov et al., 2012). To ensure effective generalization and success in contact prediction, it is crucial for the model to handle long sequences and focus on essential parts of the sequence, thus emphasizing the importance of attention mechanisms (Vaswani et al., 2017). Therefore, they employ a transformer model with 33 layers and 250 million parameters, trained on the UR50/S dataset with extensive hyperparameter optimization, now recognized as the ESM-1b transformer. As a result of this training, the representations learned by ESM-1b have been experimented on critical biological benchmarks, including secondary-structure prediction, long-range residue–residue contacts, and remote homology detection. ESM-1b represents the beginning of a larger initiative to extract knowledge from a vast corpus of protein sequences through unsupervised learning, setting the stage for the development of subsequent ESM models.

Following ESM-1b, Meta AI released the continuing series of the ESM models, including ESM-1v (2021), ESM-2 (2022) and the recently published ESM-3 (2024), each offering improvements to the previous architecture (Meier et al., 2021; Lin et al.,

2022; Hayes et al., 2024). ESM-1v is specifically designed to address the challenge of understanding variant effects in protein sequences. It operates on the prior belief that the functional properties of proteins are encoded through evolution into their sequences. By leveraging unsupervised learning on extensive corpora of protein sequences, ESM-1v aims to decode this information embedded within the sequences. The developers of ESM-1v point out the inconvenience of traditional approaches, which require training a new model for each new task. However, if a model can effectively learn sequence variation, it could eliminate the need to train separate models for different tasks. ESM-1v demonstrates that it can capture the functional effects of sequence variation without relying on experimental data, purely through unsupervised learning. The model, a transformer-based language model with 650 million parameters, is trained to predict variant effects in protein sequences using the ESM-1b architecture and a masked language modeling approach, as described by Rives et al. (2021). The ESM-2 model is trained using a BERT-style encoder-only transformer with modifications to the number of layers, attention heads, hidden size, and feed-forward hidden size (Devlin et al., 2018). It makes use of a learned positional encoding called Rotary Position Embedding (RoPE) instead of the static sinusoidal encoding used in the original transformer model (Su et al., 2024). Additionally, dropout layers in both the hidden layers and attention layers have been removed. The latest ESM architecture, ESM-3, is a generative language model capable of acting as an evolutionary simulator, predicting proteins that are evolutionary distant from known present-day proteins. It employs a transformer architecture with several enhancements: Pre-layer normalization instead of Post-layer normalization, Rotary Position Embeddings as used in ESM-2, and SwiGLU in place of ReLU (Xiong et al., 2020; Shazeer, 2020; Nair and Hinton, 2010). ESM-3 is available in three sizes: small (48 layers, 1.4B parameters), medium (96 layers, 7B parameters), and large (216 layers, 98B parameters).

Subsequently, a BERT-based model named ProteinBERT was introduced by Brandes et al. (2022). The ProteinBERT model is trained on protein sequences and GO annotations. The input to the ProteinBERT model is the corrupted version of both the protein sequence and the GO annotations, and the model tries to recover the original uncorrupted version of these protein sequences and GO annotations, using a denoising autoencoder architecture (Vincent et al., 2008). Similarly leveraging another prominent transformer model, Ferruz et al. (2022) introduced a GPT-based pLM named ProtGPT-2. ProtGPT-2 is capable of sampling proteins similar to real-life example proteins. Furthermore, ProtGPT-2 is also capable of generating protein sequence regions that have not been explored in the literature. Recently, Su et al. (2023) trained a model, SaProt, using Foldseek representations of proteins, thus

integrating and learning representations of proteins leveraging their 3D structure (van Kempen et al., 2022). SaProt is trained on 40 million protein sequences and structures and is evaluated on critical and well-known biologically relevant tasks, surpassing the performance of the ESM models. This significant improvement of SaProt highlights the benefits brought by the integration of 3D structure, emphasizing structure as a prominent feature to employ in protein language models. Most recently, a model named ProtHyena was developed, which combines transformer and recurrent neural networks (Zhang and Okumura, 2024).

## 2.4 Transductive and Semi-Supervised Learning Approaches

Transductive learning and semi-supervised learning are methods in machine learning that are specifically beneficial when there is very little labeled data or when there is a vast amount of unlabeled data whose features are available during training. Semi-supervised learning leverages both labeled and unlabeled data during training, aiming to generalize to other unseen test data (Zhu, 2005). Similarly, transductive learning also focuses on using labeled and unlabeled data during training; however, unlike semi-supervised learning, transductive learning focuses on given unlabeled data, which is generally the test data, rather than generalizing to new unseen data (Vapnik et al., 1998). These methodologies are particularly relevant in fields like bioinformatics, where obtaining comprehensive labeled datasets can be challenging, and the need for precise models is critical (Zhu, 2005).

### 2.4.1 Transductive Learning Approaches

This section will discuss transductive learning approaches relevant to zero-shot learning setups, which is the focus of this study.

The domain shift problem is a well-known issue that generally occurs in zero-shot learning setups due to the model being trained on the training domain and not being able to adapt to the test domain, resulting in a domain shift problem. Aiming to overcome the domain shift problem, Wan et al. (2019) define visual structure constraints using Chamfer distance, Bipartite matching, or Wasserstein distance (Barrow et al., 1977; Edmonds, 1965; Villani et al., 2009). Using one of these distance metrics, they aim to minimize the distance between the projected vector of the test sample and the nearest class vector in the visual space by leveraging the attribute vectors of the test data.

Song et al. (2018) introduce Quasi-Fully Supervised Learning (QFSL), which uses a transductive approach that utilizes the test samples during training. To adhere to fundamental machine learning principles where test labels are unavailable during training, they employ a novel strategy. They decrease the loss of the predictor model when test classes are predicted for test samples, while standard cross-entropy loss is computed for training samples using the actual class labels.

Dealing with document classification, Lin et al. (2021b) extract the BERT features of words in the documents to construct a graph representation (Devlin et al., 2018). This graph not only includes representations of words from training documents but also integrates words from test documents, thereby enriching the model's understanding of word connections across both training and test datasets.

### 2.4.2 Semi-Supervised Learning Approaches

The concept of semi-supervised learning consists of diverse methodologies; however, this section will specifically focus on pseudo-labeling, a specific kind of semi-supervised learning. Specifically, pseudo-labeling approaches relevant to zero-shot learning setups will be discussed.

Li et al. (2019) propose a novel pseudo-labeling method where they prioritize pseudo-labeling the classes they categorize as "hard classes". They propose two approaches for identifying these hard classes. The first strategy categorizes a class as hard if it is predicted infrequently, while the second strategy incorporates prior knowledge of class distribution to categorize a class as hard. Later, the authors implement dynamic pseudo-labeling, where hard classes are labeled proportionally to the model's epoch progression, based on more confident predictions.

Ye and Guo (2019) integrate pseudo-labeling with ensemble learning, another paradigm frequently used in machine learning which involves combining the predictions of multiple versions of the same model. In their study, they suggest that learning a single projection matrix through model training might not be sufficient. Thus, by using the same model architecture, they train K models by sampling K different subsets from the sample set, thereby learning K different projection matrices. These matrices are then used to predict test samples, employing either majority voting or averaging of predictions to determine the final pseudo-labeled outcome. Later, they add the pseudo-labeled samples into the next iteration of the training process, thus performing progressive pseudo-labeling.

# 3. PROBLEM FORMULATION AND THE DARKIN BENCHMARK DATASET CURATION

This chapter first presents the details of the problem formulation for the zero-shot prediction for dark kinases. The evaluation setup is critical for assessing the performance of the models; we will detail our efforts in creating a benchmark dataset. Subsequently, we discuss various methods experimented with in the DeepKinZero problem formulation to achieve improvements.

## 3.1 Zero-Shot Learning Problem Formulation

This thesis study builds on the previous work of Deznabi et al. (2020), where they developed a zero-shot learning model, DeepKinZero, which accepts phosphosite embeddings and outputs a kinase. Deznabi et al. (2020) achieve this by learning a compatibility matrix between the phosphosite and kinase embeddings.

Deznabi et al. (2020) employ the approach contributed by Sumbul et al. (2017), which learns a compatibility function between the input and output embeddings. This compatibility function can be represented as $F : X \times Y \to \mathbb{R}$. The DeepKinZero model takes as input $x_i$, the phosphosite embedding, and $y_j$, the kinase embedding, and outputs a compatibility score. This score indicates the likelihood of the kinase phosphorylating the phosphosite. The likelihood of a kinase phosphorylating a phosphosite can be calculated using the formula in Equation 3.1.

$$p(y_j \mid x) = \frac{\exp(F(x, y))}{\sum_{y' \in Y_{\text{te}}} \exp(F(x, y'))} \tag{3.1}$$

The compatibility function $F(x, y)$ is defined as follows:

$$F(x,y) = \begin{bmatrix} \theta(x) \\ 1 \end{bmatrix}^{\top} W \begin{bmatrix} \phi(y) \\ 1 \end{bmatrix} \tag{3.2}$$

In this representation, $\theta(x)$ and $\phi(y)$ are embeddings of the input variables $x$ and $y$, respectively. The vectors are augmented with a constant 1 to incorporate a bias term directly into the compatibility matrix $W$. This matrix $W$ projects the augmented embeddings into a scalar compatibility score, which quantitatively evaluates how well the phosphosite, $x$, and kinase, $y$, align with each other.

## 3.2 Benchmark Dataset Creation

A benchmark dataset of kinase-phosphosite associations for the zero-shot prediction task of dark kinase-association predictions has been curated as part of this thesis study. This section outlines the improvements over the previous evaluation framework applied in DeepKinZero (Deznabi et al., 2020) and the steps undertaken to gather, process, and create the data for the zero-shot learning setup. The resulting dataset is called DARKIN. The scripts used to establish this dataset are publicly available in the Darkin GitHub repository[1] to foster further studies within the research community.

### 3.2.1 Enhancing the DeepKinZero Dataset: Improvements Based on Previous Dataset Analysis

In their previous work, Deznabi et al. (2020) created a dataset consisting of kinase-phosphosite pairs representing experimentally validated kinase-phosphosite associations. To address potential improvements and to use up-to-date data, this study's first stage involved developing an algorithm that employs various strategies to create randomized and reproducible kinase-phosphosite association dataset splits.

Deznabi et al. (2020) created dataset splits according to the number of phosphorylation data associated with each kinase, in other words, the number of unique kinase-phosphosite pairs associated with each kinase. In the DeepKinZero paper, they referred to dark kinases as "rare kinases" and light kinases as "common kinases".

[1] https://github.com/tastanlab/darkin

The train, validation and test splits are based on kinase-phosphosite pair associations. In the DeepKinZero dataset, kinases that have more than 5 phosphosite associations are designated as train kinases, those that have exactly 5 phosphosite associations are designated as validation kinases, and kinases that have fewer than 5 phosphosite associations are set as the test kinases. Since the aim of DeepKinZero is to predict the dark kinases that phosphorylate a given phosphosite, this setup reflects the deployment scenario, where the test set consists of the actual dark kinases. Thus, DeepKinZero performance is evaluated over phosphosite-dark kinase associations of dark kinases. However, there are very few kinase-phosphosite associations to evaluate. Thus, while the scenario is close to the real-life scenario, the performance evaluation might be limited for many kinases. In our work, as detailed in Section 3.2.4.5, we establish a specific threshold for kinases in the test set to ensure a sufficient number of samples for these kinases, in order to have a more robust evaluation. Although this method does not reflect real-life scenarios—since test kinases are those with a considerable number of phosphosite associations and are, therefore, technically not considered dark kinases—the strategy was chosen to enhance the robustness of the model's evaluation within this study.

Deznabi et al. (2020) used the kinase-phosphosite associations from PhosphoSitePlus (Hornbeck et al., 2014). However, many more kinase-phosphosite associations have been added to the PhosphoSitePlus database throughout the years. In this work, we update the dataset with the newest version.

In addition to this, Deznabi et al. (2020) used the kinase set provided by Manning et al. (2002), which is the most commonly used and oldest kinase set. A more recent study provides a curated kinase set with kinases known to show experimentally validated kinase activity (Moret et al., 2020). In this work, we use this work to form the human kinase list.

Up-to-date data have been gathered to address potential improvements, and a new and unique data-splitting strategy has been developed as part of this thesis. The details of the data gathering and splitting phases are explained in the following sections.

### 3.2.2 Data Collection

This section presents the details of the methodology and the sources which are utilized to construct the DARKIN dataset.

#### 3.2.2.1 The Human Kinase Set

The first publicly available kinase set is the 518 human kinase set by Manning et al. (2002). There have been multiple other kinase sets defined by other studies, such as Manning et al. (2023), Eid et al. (2017), UniProt Consortium (2023a), and Moret et al. (2020). Even though the definition of a kinase is clear, there is not a consensus on how to label an enzyme as a kinase. As a result, these mentioned kinase sets partially overlap. After careful analysis of the publicly available kinase sets, we decided to use the 557 human kinase set provided by Moret et al. (2020) in this study, as they provide an up-to-date, consistent, and well-curated list of kinases. This kinase list consists only of kinases that have experimentally proven phospho-transfer activity. Unlike Manning et al. (2002), this list includes all known Protein Kinase Like (PKL) kinase domains and excludes kinases with "Unrelated to Protein Kinase" (uPK) and "Unknown" protein kinase domain folds.

Moret et al. (2020) provide two human kinase sets, one consisting of an extended list of 710 human kinases and the other a further curated dataset of 557 human kinases. Their extended kinase dataset includes both kinases with experimentally proven phospho-transfer activity and kinases without experimentally proven phospho-transfer activity. On the other hand, the curated dataset consists only of kinases with known Protein Kinase Like (PKL) domains. Kinases with "Unrelated to Protein Kinase" (uPK) and "Unknown" protein kinase domain folds have been excluded from the curated kinase dataset. They have included STK19, even though it has an unknown fold because it is known that STK19 plays a role in phosphorylation (Yin et al., 2019). As the curated kinase dataset of 557 kinase domains presents a well-defined and consistent kinase list, with features such as family and group information being nearly complete, we used this kinase set as the foundation kinase set.

The collected kinase information is listed below:

- **Kinase Domain:** Kinase domains are regions within a kinase that actively participate in phosphorylation. First, we retrieve the protein sequence of the entire kinase using the UniProt API (UniProt Consortium, 2023b). Subsequently, the kinase domains were extracted from these sequences using the indices provided in the 557 human kinase set (Moret et al., 2020). Since Moret et al. (2020) also retrieved the index information from UniProt, directly using the indices provided in the dataset gave accurate and consistent results.
- **Group & Family:** In their study, Manning et al. (2002) classified kinases

into family and group hierarchies based on their catalytic domain similarity. This classification was supported by additional features, including sequence similarity, domain structure outside the catalytic domains, known biological functions, and a comparable classification of the yeast, worm, and fly kinomes. In total, 10 groups and 116 families were defined. The group and family information of the kinases, as defined by Manning et al. (2002), was present in the 557 kinase set. Thus, we directly use the family and group information provided in the 557 human kinase dataset.

- **Enzyme Commission (EC) Numbers:** Another hierarchical feature that could be used as an additional feature for kinases is Enzyme Commission (EC) categorization. EC classifies enzymes based on the chemical reactions they catalyze and are retrieved from the ENZYME database (Bairoch, 2000). The EC numbers are provided in four levels of numerical representation, separated by dots. All kinase-related categorization belongs to the same first two higher levels, which are 2.7. On the third level, there are six main kinase categories, which further divide into the fourth level of the categorization. For instance, the kinase 'O00141' belongs to the EC category 2.7.11.1, as it could be seen the two higher categories start with 2.7, which is then followed by the two lower categories which are 11 and 1. It should also be mentioned that a kinase could belong to multiple EC categories. For example, the kinase 'O00329' belongs to two EC categories: 2.7.1.137 and 2.7.1.153.

#### 3.2.2.2 Substrates

Substrates are the target proteins that undergo phosphorylation. Thus, the starting point of the substrate dataset was the set of all protein substrates in the kinase-substrate association dataset retrieved from PhosphoSitePlus (Hornbeck et al., 2014). To represent the substrates, we collected the whole amino acid sequence of the substrate proteins using the UniProt API (UniProt Consortium, 2023b). All sequences were successfully retrieved from the API except for 30 substrates. To collect the sequences for these remaining 30 substrates, we manually searched the IDs and used the ID-to-ID mapping tool on the UniProt website. Manual searches in the PhosphoSitePlus database were performed for the substrates whose sequences were still missing. The cross-references of these substrates to the UniProt database were used if any existed. All substrates whose sequences were not successfully retrieved through these steps were removed from the substrate dataset.

#### 3.2.2.3 Kinase-Phosphosite Association Data

In this study, we used experimentally validated kinase-phosphosite associations. Several publicly available databases report experimentally validated kinase-phosphosite associations, such as Hornbeck et al. (2014), Dinkel et al. (2010), Yao et al. (2012) and Ullah et al. (2016). We use the kinase-substrate dataset provided by Hornbeck et al. (2014) because it contains many human kinase phosphorylation data points and is regularly updated.

#### 3.2.2.4 Protein Structures

The one-dimensional sequence information of a protein contains useful functional information about the protein itself. However, the information that can be extracted from the sequence might fall short in some aspects when representing a protein. The 3D structure of a protein provides additional details beyond the 1D sequence to represent binding and functional properties. The structural information of both the substrates and kinases was collected.

Experimentally determined protein structures are obtained from PDBe (Protein Data Bank in Europe, 2023). AlphaFold Protein Structure Database provides prediction-based protein structures (DeepMind, 2023). The protein structures were retrieved using the AlphaFold API (DeepMind and European Bioinformatics Institute, 2023; Jumper et al., 2021b). Large protein structure predictions were not available in the AlphaFold database, so the protein structures for these proteins were downloaded from PDBe, ensuring the retrieval of the protein structure that has the largest coverage of the specific domain of interest for that particular protein. The important sections for kinases are the kinase domains, and the important sections for the substrates are the phosphosites. Thus, when retrieving these protein structures from PDBe, the structures that include the phosphosite of the substrate and the structures that include the largest coverage of the kinase domain for the kinases were retrieved.

Predictions for isoform proteins and for large proteins whose structure is not present in either PDBe or the AlphaFold database have been predicted using ColabFold (Mirdita et al., 2022). ColabFold is a tool similar to AlphaFold which predicts the 3D structures of proteins. The main difference between AlphaFold and ColabFold lies in their accessibility and prediction time, which mostly depends on how they handle the MSA search phase, a phase required for structure prediction. Colab-

Fold[2] is implemented to run both in Google Colab and locally, by installing and downloading the required packages and libraries. On the other hand, AlphaFold[3] presents a slightly less accurate version in Google Colab and a downloadable Docker version. In addition to this, AlphaFold handles its MSA search phase by searching large databases such as UniRef90 and MGnify (Consortium, 2018; Mitchell et al., 2020). On the other hand, ColabFold uses Many-against-Many sequence searching (MMseqs2), which results in a much shorter running time. The running time of this phase takes orders of hours for AlphaFold and orders of minutes for ColabFold. Thus, it was decided to use ColabFold due to its advantages in accessibility and running time.

#### 3.2.2.5 Kinase Active Sites

The active sites of kinases are the amino acid residues in the protein that are actively involved in phosphorylation. These amino acid regions in the protein correspond to the regions where the phosphoryl group directly binds. To identify kinase active sites in this study, the method developed by Born et al. (2021) is followed. Modi and Dunbrack Jr (2019) provided a structurally validated multiple sequence alignment for 497 human kinases. Using this multiple sequence alignment, amino acid residues corresponding to the 29 amino acid residues identified as the kinase active sites of Protein Kinase A (PKA) by Sheridan et al. (2009) were extracted. This process was performed for all kinases in the 557-kinase set, except for atypically defined kinases. This is because the atypical kinases either partially align or do not align at all with Protein Kinase A, so they were not included in the multiple sequence alignment prepared by Modi and Dunbrack Jr (2019). Therefore, the entire kinase domain was used for the kinase active site regions of atypical kinases.

### 3.2.3 Data Pre-Processing

This section presents the details of the pre-processing steps taken to curate the collected data (refer to 3.2.2 for data collection).

#### 3.2.3.1 Kinase Domains

[2]https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb

[3]https://colab.research.google.com/github/deepmind/alphafold/blob/main/notebooks/AlphaFold.ipynb

**Pre-processing the Kinase Domains** The kinase domains were extracted from the full protein sequence using the indices provided in the 557 kinase set by Moret et al. (2020). Only one kinase, STK19, did not have starting and ending kinase domain indices due to it having an unknown fold (uPK). Thus, the whole protein sequence was used as the kinase domain for STK19.

As mentioned in Moret et al. (2020), 13 human proteins have two defined kinase domains. 11 of these proteins, which have two kinase domains, also exist in the kinase substrate dataset downloaded from PhosphoSitePlus (Hornbeck et al., 2012). As kinases are identified by their accession IDs inside the kinase-substrate dataset, and since both kinase domains have the same accession ID, it was not possible to identify which kinase domain was responsible for the phosphorylation of that specific phosphosite. The studies identifying specific phosphorylation events did not provide information on which kinase domain specifically played a role in the phosphorylation event. They only provided general knowledge and assumptions on which kinase domain might have been responsible for the phosphorylation (Dummler et al., 2005; Tomas-Zuber et al., 2001). Some studies that investigated the phosphorylation events of these proteins claim that both kinase domains are catalytically active (Janknecht, 2003; Bignone et al., 2007; Zhao et al., 1995). Thus, as mentioned in Dummler et al. (2005), even though one of the kinase domains should be responsible for the ATP binding, the other domain and the linker region between these two kinase domains also play a regulatory role in phosphorylation. Therefore, for these 11 proteins that have two kinase domains, it was decided to define the kinase domain as the starting point of the first kinase domain until the end of the second kinase domain, including the linker region between these domains.

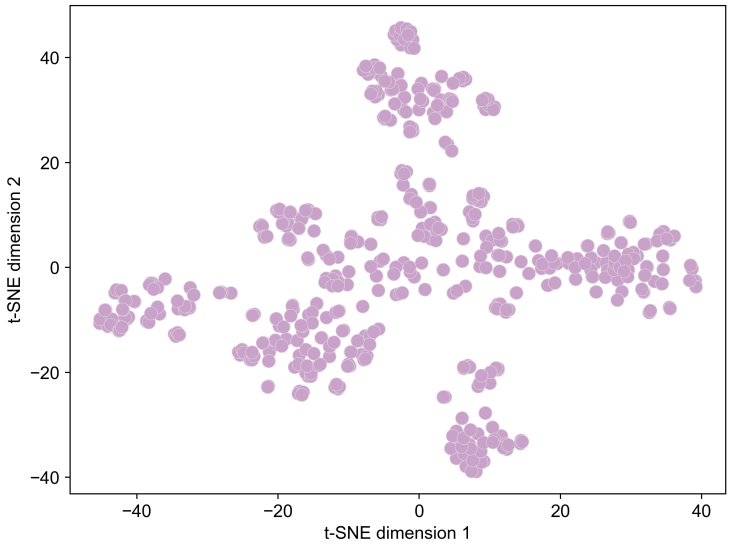
**Kinase Domain Alignments** After defining the kinase domains for each kinase, the similarity matrix of the kinases is constructed using their kinase domains. This similarity matrix is used to identify how similar a kinase domain is to every other kinase domain in the kinase set. The construction of the similarity matrix is important for the next steps where imputation is performed when kinase information is missing.

The global pairwise sequence alignment of all 544 kinase domains (557 total kinases minus 13 with two domains, resulting in 544 kinase domains) was calculated using the publicly available Biopython library (Cock et al., 2009). A gap penalty of -11 and a gap extension of -1 have been used for this calculation. Biopython's sequence alignment function provides an unnormalized similarity score, which could be misleading considering the varying sizes of the kinases. Thus, the normalized version of this score, named the identity score, has been used for the similarity score

matrix construction. The sequence identity score is calculated using a publicly available GitHub repository by JoaoRodrigues (2016).

For kinase information imputation, both the kinase similarity matrix and the visual 2D mapping of this matrix are used to identify the most similar kinases. To visually analyze the relation between the kinase domains, t-SNE (Van der Maaten and Hinton, 2008) has been used to map the similarity matrix into a 2-dimensional space. The 2D mapping of the kinase similarity matrix can be seen in Figure 3.1. Further details on how these two methods are used to impute the missing information are provided in Section 3.2.3.1.1 for imputing kinase group and family and Section 3.2.3.1.2 for imputing the EC features.

Figure 3.1 This figure shows the 2D t-SNE projection of the kinase identity score matrix, which represents the pairwise kinase domain similarities of all kinases in the human kinome as percentages. Consequently, this plot illustrates how the kinase domains of all human kinases align together in a 2D space.




#### 3.2.3.1.1 Imputing Missing Kinase Family and Group Information

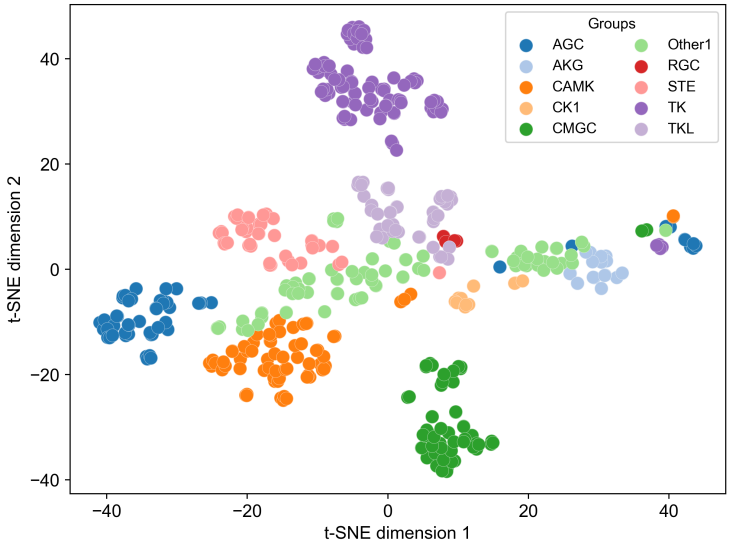
It has been shown in the prior work of Deznabi et al. (2020) that the kinase hierarchical information, which mainly consists of the kinase family and group information, greatly improves model performance. Thus, the kinase family and group information provides valuable insights regarding the kinases. Consequently, this information is collected as one of the primary features of the kinases.

Seven kinases in the kinase set had missing family information, and six of them also had missing group information. Since the kinase family and group features provide important information for the kinases, there should not be any kinase in the dataset with missing or unidentified family and group features. Therefore, the family and

group information for these seven kinases has been imputed using the kinase domain similarity score matrix and the 2D visual projection of this similarity matrix.

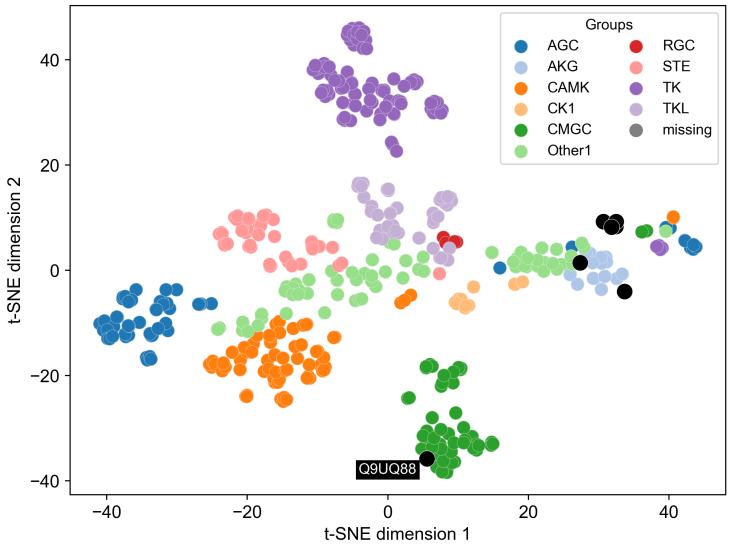
To impute the missing family and group features, the sequence-wise most similar 10 kinases' family and group features are analyzed, giving more weight to the more similar kinases. The kinase group and family features are imputed as the majority group and family if there is a clear majority group and family. If the group and family features cannot be clearly decided using this approach, then the 2D visual projection is used as a second method. To visually analyze which family and group this kinase belongs to, the kinases in the 2D visual projection are labeled with their group and family features (one projection is labeled with groups and another projection is labeled with families for clarity and easy inspection). The projection labeled with kinase groups is shown in Figure 3.2 (kinases with missing group information are excluded from this visualization). As can be seen in the visualization, kinases with the same group tend to cluster together.

Figure 3.2 Visual alignment of the 2D t-SNE projection of the kinase identity score matrix, with kinase domains colored according to their respective kinase groups. Kinases without an assigned group are excluded from this visualization.




Kinases with missing group and family features are projected into 2D space to visually analyze the kinase group clusters they are closest to. The kinases with missing group features can be seen in Figure 3.3, projected as black circles. The plots are interactive and show the UniProt ID of the kinase when hovered over. As can be seen in this figure, the black dot, close to the kinase group cluster of "CMGC" colored in dark green, shows "Q9UQ88" when hovered over.

Figure 3.3 Visual alignment of the 2D t-SNE projection of the kinase identity score matrix, with group labels. Kinases whose groups are missing and need to be imputed are colored in black.




As stated in Duong-Ly and Peterson (2013), kinases in the "Other" group are sequence-wise not similar to the groups defined in Manning's group definition (Manning et al., 2002). Following this ideology, for the kinases with missing group and family information that did not fall into a previously defined group or family by a clear margin but were still close to each other in the 2-dimensional space, an additional group named "Other2" and an additional family named "other_family" have been defined in the dataset.

#### 3.2.3.1.2 Imputing Missing Kinase EC Numbers

Four kinases' EC numbers were missing in the dataset. To impute the EC feature for these kinases, the EC numbers of the 10 closest kinases in the similarity matrix, based on pairwise identity score, were analyzed. For kinases that did not fall into a specific EC number feature with a clear margin, a zero vector was assigned as the EC number feature.

### 3.2.3.2 Substrates

The substrate with UniProt ID Q9BVL4 contained a 'U' in its amino acid sequence, which is unusual for a protein sequence. Therefore, for unexpected letters in the substrate sequences, these unusual letters were replaced with the letter 'X', as this is the convention used in the BLOSUM62 matrix.

#### 3.2.3.3 Kinase-Substrate Dataset

The kinase substrate dataset consists of experimentally validated kinase-phosphosite associations. In other words, it contains information about the kinases that phosphorylated specific phosphosites. The phosphosite information is presented with the specific site of phosphorylation and the seven neighboring residues on both sides, resulting in an amino acid sequence of length 15. The following decisions were made to clean and finalize the dataset:

- **Removal of Non-Human Kinase Associations:** Kinase-phosphosite associations related to non-human kinases were removed.
- **Inclusion of Non-Human Substrates:** No organism restriction was made on the substrates since interactions between human kinases and non-human substrates could also provide valuable information. There are preserved sequences throughout the same gene for different organisms, also known as MSA.
- **Canonical and Isoform Variations:** Substrates do not necessarily have to be the canonical form of the protein but could also be isoform variations of the canonical version. However, kinase-phosphosite associations where the kinase was not in its canonical form were removed.
- **Removal of Fusion Kinases:** Fusion kinases were removed as they might show unusual behavior.
- **Removal of Non-Existent Kinases:** Kinase-phosphosite associations where the kinase does not exist in the collected kinase set (refer to 3.2.2.1) were removed.
- **Removal of Non-Existent Substrates:** Kinase-phosphosite associations where the substrate does not exist in the collected substrate set (refer to 3.2.2.2) were removed.
- **Removal of Inconsistent Phosphosite Sequences:** Kinase-phosphosite associations where the phosphosite sequence of length 15 does not exist in the substrate's whole amino acid sequence were removed, as these kinase-phosphosite associations might introduce noise.
- **Inclusion of Pseudogenes and Pseudokinases:** Kinase-phosphosite associations containing pseudogenes or pseudokinases were kept.

### 3.2.4 DARKIN: The Zero-Shot Benchmark Dataset

In this section, detailed specifications of the DARKIN dataset and the implementation details of the script to generate the DARKIN dataset will be provided.

#### 3.2.4.1 DARKIN Dataset

To be able to make predictions for the dark kinases, we cast the problem in a zero-shot learning task. The evaluation of the zero-shot learning model is not trivial in this scenario as kinases differ in the number of examples they have and kinases have similarities to each other, which needs to be taken into account in a fair evaluation setup. We generate a procedure and implement this as a script to generate several different versions of train, validation, and test splits in the zero-shot learning setup. Below we describe these efforts.

#### 3.2.4.2 Dataset Description

After the data collection and curation steps mentioned in the previous sections, a total of 17,617 kinase-phosphosite associations remain. The description of how this data is split into train, validation, and test sets will be provided in the upcoming sections. Prior to detailing this division, the features of the dataset will be described in this subsection.

There are a total of four columns/features used in this dataset: SUB_ACC_ID, SUB_MOD_RSD, SITE_+/-7_AA and KINASE_ACC_IDS. The descriptions of these columns are provided below:

**SUB_ACC_ID**: The UniProt ID of the substrate protein in which the phosphosite resides.

**SUB_MOD_RSD**: The specific residue location of the phosphosite inside the substrate protein. For example, S267 corresponds to the $267^{th}$ amino acid inside the protein sequence, which is also a serine amino acid.

**SITE_+/-7_AA**: The amino acid sequence of length 15 that contains the phosphosite residue in the middle, or in other words, in the $8^{th}$ residue location. If the phosphosite is close to either of the terminal ends of the protein, padding is added with "_" to ensure that the phosphosite is positioned in the middle of the sequence.

**KINASE_ACC_IDS**: The kinase UniProt IDs known to phosphorylate this specific phosphosite in this given substrate protein. Since a phosphosite could be phosphorylated by multiple kinases, this column could contain multiple kinases.

A snippet of the dataset is provided in 3.1 to give a better understanding of the data.

Table 3.1 This table presents a sample snippet from the DARKIN dataset. The columns from left to right represent: 1) the substrate accession ID, 2) the residue ID of the phosphosite within the protein sequence, 3) the 15-residue sequence with the phosphosite at the center, and 4) the kinase accession IDs experimentally validated to phosphorylate this phosphosite. As shown in the table, a phosphosite can be phosphorylated by multiple kinases.

| SUB_ACC_ID | SUB_MOD_RSD | SITE_+/-7_AA | KINASE_ACC_IDS |
|---|---|---|---|
| P01106 | S267 | PPTtssDsEEEQEDE | P48729, P68400 |
| O00267 | T784 | MyGsGsrtPMyGsQt | P50613, P50750 |
| P12839 | S503 | EEPEVEKsPVKsPEA | P49840 |
| P18887 | T519 | EDPyAGstDENtDsE | P68400 |

#### 3.2.4.3 Introduction to the DARKIN Script

The DARKIN dataset splits are generated in a reproducible manner, thus the same splits could be generated when the same parameters are entered to the generation script. There are several parameters that could be adjusted accordingly. The full list of the dataset parameters can be found in Appendix A.1. Some important parameters will be touched upon in this subsection.

**Random Seed**: The random seed that will be set to ensure the reproducibility of the same dataset on different runs.

**Kinase Similarity Percent**: The identity similarity score of the kinase domains that will be taken into consideration when splitting the dataset. This similarity percent defines the similarity level at which the kinases are considered highly similar. (Kinase domains that have similarity equal to or above this will be placed inside the same dataset).

**Kinase Count Test Threshold**: The threshold number of phosphorylation data for a kinase to be able to enter the test dataset. In other words, kinases that have

fewer phosphorylation data than this threshold will not be candidate kinases to enter the test dataset.

**Stratify Percentage for Unseen Test Kinase**: The percentage of the dataset to be entered into the test set as unseen kinase-phosphosite data.

The DARKIN dataset is made publicly available at this GitHub link[4].

#### 3.2.4.4 Using the DARKIN Script

In this subsection, the procedure for downloading and running the DARKIN script from the GitHub repository is explained.

Listing 3.1 The step-by-step instructions for downloading and running the DARKIN script are provided in this listing.

```
# cloning the repository to the specified location
git clone git@github.com:tastanlab/darkin.git

# creating the conda environment
conda create --name darkin python=3.11.3

# activating the conda environment
conda activate darkin

# installing the pip package
conda install pip

# installing the required packages as specified in the GitHub
   repository
pip install -r requirements.txt

# or manual package installation could be done with  the following
   lines
pip install pandas
pip install numpy
pip install matplotlib

# running the DARKIN script to generate the desired split
python create_darkin_split.py
```

[4]https://github.com/tastanlab/darkin

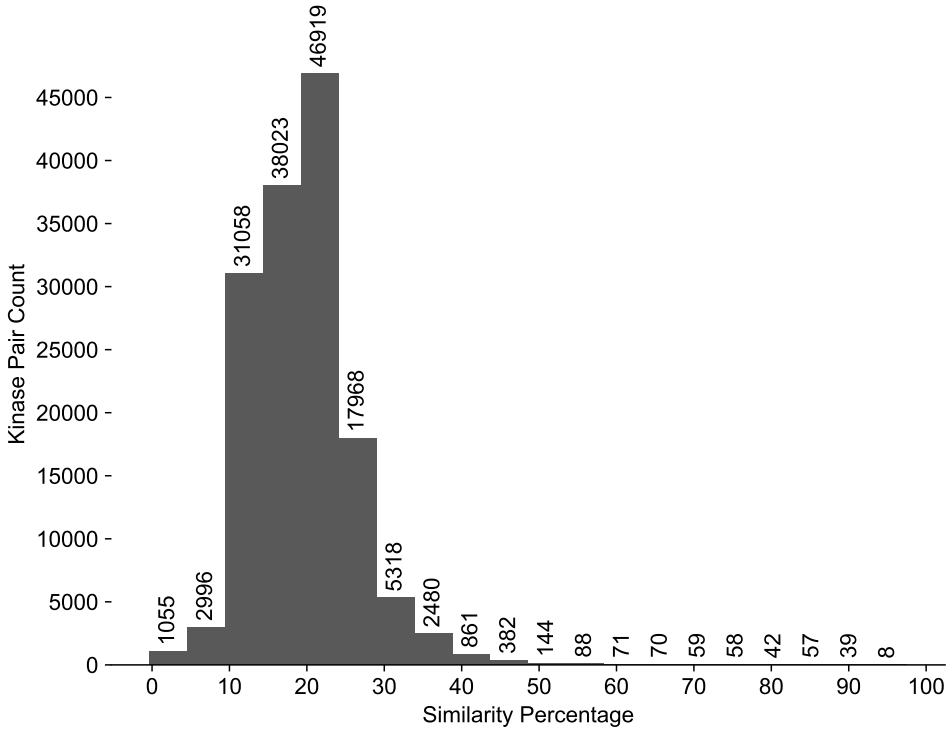
```
# parameters could be set when running the script, this line sets
    the random seed to 12 when running the script
python create_darkin_split.py --RANDOM_SEED 12
```

#### 3.2.4.5 Strategies Used in the DARKIN Generation Process

We consider several aspects of the problem and the nature of the biological problem to build good evaluation splits. These strategies are detailed below:

- **Number of phosphosites per test kinase:** The classes to be predicted are kinases, and the performance of each test kinase is reported to evaluate the performance of the applied models and methods. To have a more reliable and robust evaluation, the kinases must be evaluated on a sufficient number of phosphosites. To ensure this, a threshold for the number of kinase-phosphosite pairs related to the validation and test kinases is set.
- **Stratification with respect to kinase groups:** In the previous study by Deznabi et al. (2020), it was shown that family hierarchy information provides the most valuable insights. Additionally, the dataset contains only 392 kinases distributed across 11 kinase groups and 129 kinase families. Stratifying with respect to kinase families is not feasible since there would not be sufficient kinases from each family for each split. Since kinases within the same kinase group share evolutionary relationships and functional similarities (Manning et al., 2002), kinases are stratified according to their kinase groups to ensure equal representation of kinase groups in train, validation, and test splits.
- **Sequence similarity of kinases:** To prevent optimistic and unrealistic evaluation of test kinases, sequence-wise similar kinases are placed in the same split (train, validation, or test). The identity sequence similarity distribution of the kinase pairs can be seen in Figure 3.4. There are around 47 kinase pairs with a sequence similarity of 90 or above. These similar kinase pairs are always placed together in a randomly chosen split. This criterion is important so that the model is not trained on very similar kinases that exist in the test split.

Figure 3.4 Histogram showing the distribution of kinase pair similarity scores. This plot illustrates the number of kinase pairs whose similarity scores fall within specific ranges, providing a visual representation of the frequency of different similarity levels among kinase pairs. The histogram shows that there are 47 kinase pairs with over 90% sequence similarity.




#### 3.2.4.6 Implementation Details of the DARKIN Generation Script

The details of the zero-shot dataset splitting code will be described in this subsection step-by-step.

1. **Calculating Site Associations:** Since the dataset is generated for the zero-shot setup, the train and test kinases should be disjoint. Thus, it can be said that the splitting process will be based on the kinase classes. As the first step, the number of site associations for each kinase is calculated.

2. **Group-wise Stratification of Kinase-Phosphosite Associations:** After calculating the number of site associations for each kinase, if the parameter for splitting the dataset with respect to kinase group is set to 'True,' then the total number of kinase-phosphosite associations within each kinase group is calculated. The distribution of the total number of kinase-phosphosite associations within each kinase group can be seen in Figure 3.5. Next, a specific portion of each kinase group is calculated to be placed in train, validation, and

test sets. Thus, the number of kinase-phosphosite associations that should enter into train, validation, and test sets from each kinase group is calculated numerically. This process is illustrated in Figure 3.6.

Figure 3.5 Distribution of kinase-phosphosite association samples across kinase groups. This plot displays the number of association samples for each kinase group, highlighting the variation in sample counts among different groups.

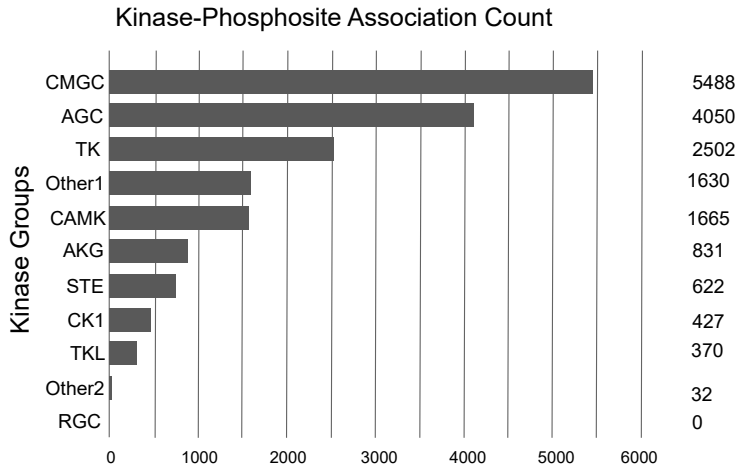
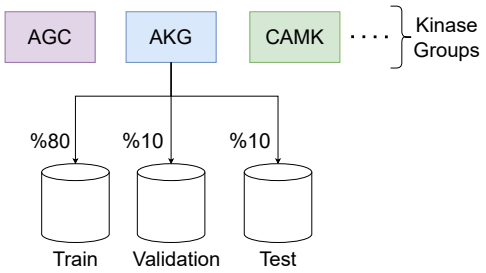



Figure 3.6 Visualization of the process of stratifying kinase-phosphosite data into train, validation, and test sets with respect to kinase groups. This plot demonstrates how the data is partitioned across the different sets, ensuring representation from each kinase group.




After all pre-calculations are done, the process of deciding how much data from each kinase will enter which set (train, validation, or test) will be calculated in the following steps.

3. **Handling Sequence-wise Highly Similar Kinases:** When placing kinase classes into their respective sets, in the first step, the kinases that have sequence similarity above a parameterized threshold are identified. These similar kinases are placed into sets due to the transitivity property (The sets where similar kinases are placed into will be named as "similarity sets" throughout this explanation). If kinase1 is similar to kinase2, and kinase2

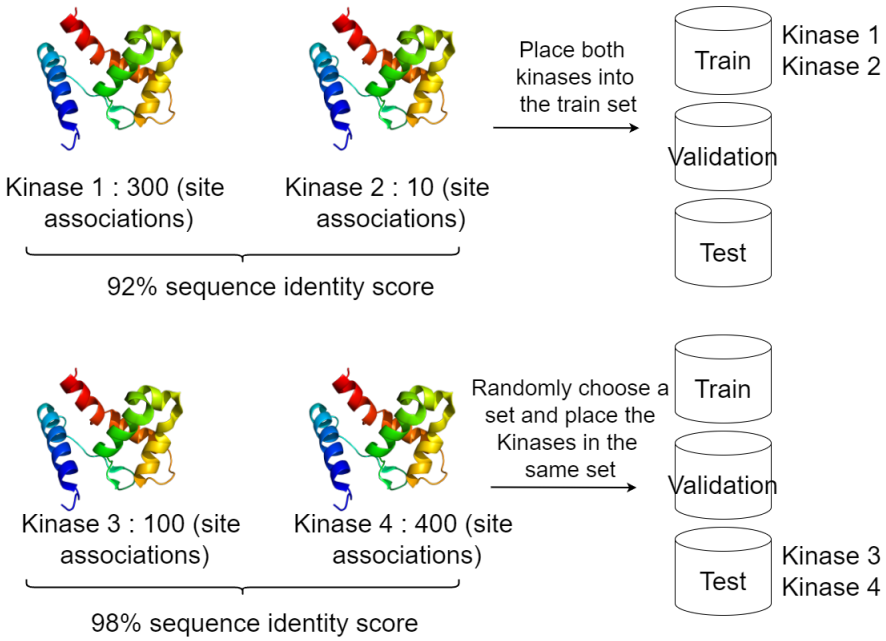
is similar to kinase3, then kinase1, kinase2, and kinase3 are all considered similar and kinase1, kinase2 and kinase3 will be in the same similarity set. These similar kinases are placed into the same sets (train, validation or test) together, abiding by the following rules:

- If any kinase in a specified similarity set has fewer site associations than the predefined test threshold, then all kinases in this set, in other words, all kinases similar to this kinase, are placed into the train set.

- For the remaining kinase similarity sets, if all kinases in the set have more site associations than the test threshold, then one of the train or test sets is randomly selected and all kinases in the aforementioned kinase similarity set are placed into this set.

This step is illustrated in Figure 3.7.

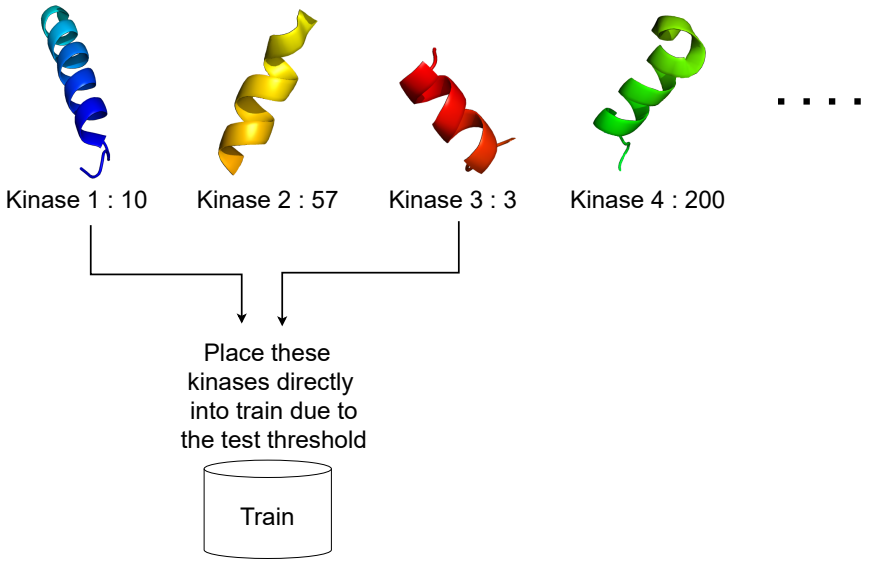
Figure 3.7 Visualization of the process for placing sequence-wise highly similar kinases into the same set, assuming a test threshold of 15 samples. The figure showcases two potential scenarios: In the upper half, if one of the similar kinases has fewer samples than the test threshold, both kinases are placed in the train set. In the lower half, if both similar kinases exceed the test threshold, a random set (train, validation, or test) is selected, and both kinases are placed in that set.




4. **Placing Low Association Kinases in Train Set:** In the next step, the

kinases that have fewer site associations than the parameterized test threshold are placed in the train set. This step is illustrated in Figure 3.8.

Figure 3.8 This figure shows the process of placing kinases with a site association count lower than the parameterized test threshold into the train set, assuming a test threshold of 15 samples. The value next to each kinase represents its site association count, formatted as Kinase_Name: Kinase_Site_Association_Count.




5. **Random Distribution of Remaining Kinases:** Kinases are selected randomly from among the remaining kinases until the target number of kinases is reached according to the target number previously calculated in step 2. Once all kinase groups have reached the target number of kinase-phosphosite associations, all remaining kinases are set as train kinases.

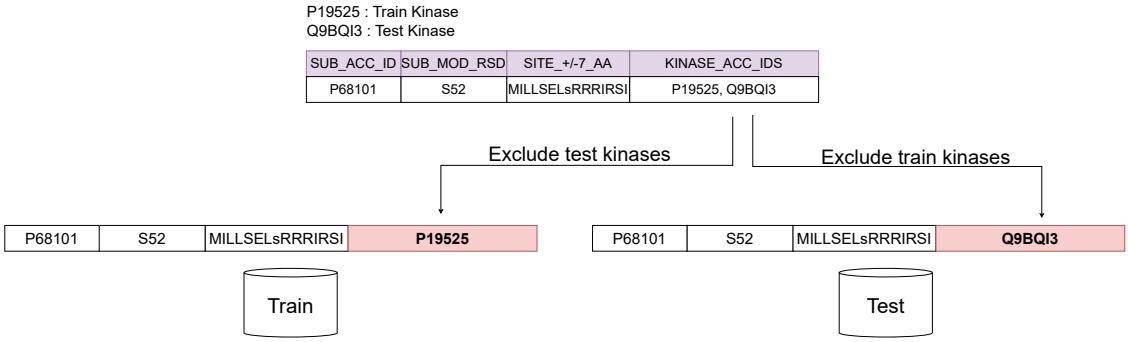
6. **Distribution of Specific kinase-phosphosite Associations:** Up until this step, kinases are set as either train or test kinases, and the number of kinase-phosphosite association counts that should be placed using this kinase into train or test is specified. However, the specific kinase-phosphosite associations are not yet distributed into the train and test sets. In this step and forward, the specific kinase-phosphosite associations will be split into train or test according to the kinases associated with that kinase-phosphosite association.

   A phosphosite could be associated with multiple kinases, and these kinases could have been specified as train or test in step 5. Thus, some phosphosites might be phosphorylated by both a train kinase and a test kinase. In these situations, it is not straightforward to decide whether these kinase-phosphosite associations should be placed into train, test, or neither. When splitting the kinase-phosphosite associations related to these kind of phosphosites into train

and test, the following rules are applied:

- If all kinases that phosphorylate the phosphosite are specified as train kinases, then this kinase-phosphosite association will be placed into the train set.
- If all kinases that phosphorylate the phosphosite are specified as test kinases, then this kinase-phosphosite association will be placed into the test set.
- If some kinases that phosphorylate a phosphosite are specified as train and some as test, then the version of the kinase-phosphosite association where the test kinase is excluded is placed into the train set, and likewise, the version of the kinase-phosphosite association where the train kinase is excluded is placed into the test set. This process is illustrated in Figure 3.9.

Figure 3.9 Phosphosites phosphorylated by both train and test kinases are added to the train set by excluding the test kinase and to the test set by excluding the train kinase.




7. **Splitting Train Set into Train and Validation:** The same procedure between steps 1-6 is repeated to split the train set into the final version of the train and validation sets using the parameter set specified for validation.

# 4. ZERO-SHOT MODELS TO BENCHMARK DARKIN AND TO EVALUATE PROTEIN LANGUAGE MODEL PERFORMANCE

To benchmark the performance of various protein language models in the DARKIN dataset setup, two simple zero-shot learning models are implemented. This section will first present the protein language models that will be experimented on and later will explain these two zero-shot models.

## 4.1 Evaluated Protein Language Models and Baseline Encodings

In this study, protein language models (pLMs) were selected based on their accessibility, reported performance in the literature, and recent development. Table 4.1 presents the pLMs experimented with in this study and their properties[1]. Processing large dimensions of protein embeddings poses a challenge. To enable more efficient processing, the column-wise average of the embeddings for all pLMs was computed, excluding the padding (PAD) token vectors. Additionally, for pLMs with a classification (CLS) token, the vectors corresponding to this token were used as an embedding summary.

## 4.2 The Baseline Model: A Zero-Shot k-NN Model

To establish baseline scores, an adaptation of the k-NN(Cover and Hart, 1967) model for the zero-shot learning setup is implemented. This model was deliberately kept simple to evaluate the representation strength of various protein language models. The process begins by identifying the $k$ most similar training sites to a test site for which a kinase prediction is to be made. Since these sites belong to training kinases, their kinase labels are known. Next, the majority train kinase among all kinases associated with these $k$ train sites is determined. Subsequently, the most similar test kinase to this majority train kinase is identified. This selected test kinase is

[1]This table has been curated by Zeynep Işık, a member of our group who is also a member of the TÜBİTAK project 122E500

Table 4.1 The Protein Language Models (pLMs) compared in this study. This table has been curated by Zeynep Işık, a member of our group who is also a member of the TÜBİTAK project 122E500

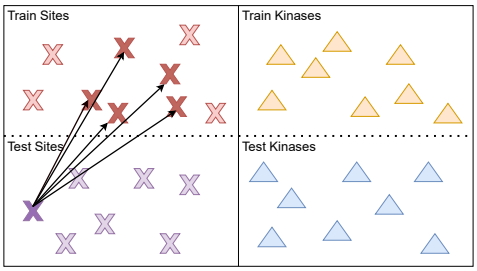
| PLM | Dataset | Vector Size | Model Size | Representation | Objective | Paper |
|---|---|---|---|---|---|---|
| TAPE | PFAM | 768 | 38M | Sequence | Sequence-based,Structural Feature Prediction | Rao et al., 2019 |
| ProtBERT | BFD100, UniRef100 | 1024 | 420M | Sequence | Sequence-based, Structural, Physicochemical Feature Prediction | Elnaggar et al., 2021 |
| ProtALBERT | UniRef100 | 4096 | 224M | Sequence | | Elnaggar et al., 2021 |
| ProtT5-XL | BFD100, UniRef50 | 1024 | 3B | Sequence | | Elnaggar et al., 2021 |
| ESM-1b | UniRef50 | 1280 | 650M | Sequence | Structural, Physicochemical Feature Prediction | Rives et al., 2020 |
| ESM-1v | UniRef90 | 1280 | 650M | Sequence | Sequence Variant Prediction | Meier et al., 2021 |
| ESM-2 | UniRef50 | 1280 | 650M | Sequence | Structural Feature,Contact Prediction | Lin et al., 2022 |
| ProteinBERT | UniRef90 | 1562 | 16M | Sequence | Sequence-based Feature, GO Annotation Prediction | Brandes et al., 2022 |
| ProtGPT2 | UniRef50 | 1280 | 738M | Subword | Protein Design and Engineering | Ferruz et al., 2022 |
| DistilProtBERT | UniRef50 | 1024 | 230M | Sequence | Sequence-based, Structural, Physicochemical Feature Prediction | Geffen et al., 2022 |
| Ankh | UniRef50 | 1536 | 1.5B | Sequence | General Purpose Modeling | Elnaggar et al., 2023 |
| SaProt | AlphaFold2, PDB | 1280 | 650M | Sequence, Structure | Structure-Aware Feature, Mutation Effect Prediction | Su et al., 2023 |

ultimately predicted as the kinase for the test site by the zero-shot k-NN model. The prediction process of the zero-shot k-NN model is illustrated in Figure 4.1.

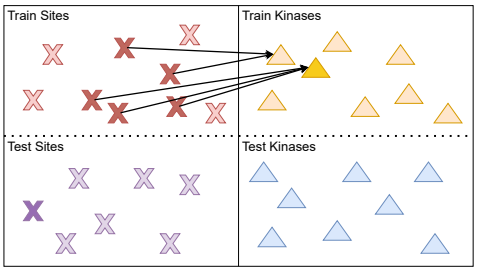
## 4.3 The Bi-Linear Zero-Shot Model

In addition to the zero-shot k-NN model, throughout this study, a second learning method, the Bi-linear Zero-Shot Model (BZSM), is also implemented. Similar to the Zero-shot k-NN model, the purpose of BZSM is to assess the strength of the protein language models. Unlike the k-NN model, the bilinear model is designed similarly to the DeepKinZero model by using the same bilinear compatibility function employed in DeepKinZero. However, the key difference is that in this model, the phosphosite embeddings are not fine-tuned using an LSTM layer. This approach aims to directly measure the performance and strengths of the protein language models within a pure bilinear compatibility function setup, without any enhancement by additional model components. The BZSM is visualized in Figure 4.2.

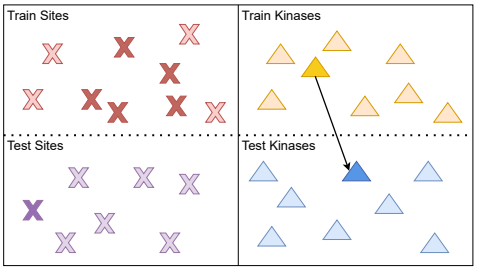
Figure 4.1 This figure depicts the step-by-step prediction process of the Zero-shot k-NN model.



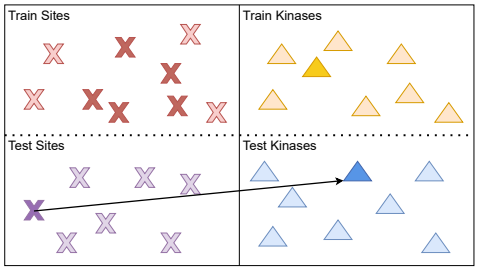

(a) Step 1: For a given test phosphosite, first the k most similar training phosphosites in the phosphosite representation space are located.




(b) Step 2: Subsequently, the most common light kinase (train kinase) among the kinases associated with the nearest neighbor phosphosites is identified. In cases where there is no majority, the kinase of the nearest neighbor is utilized.

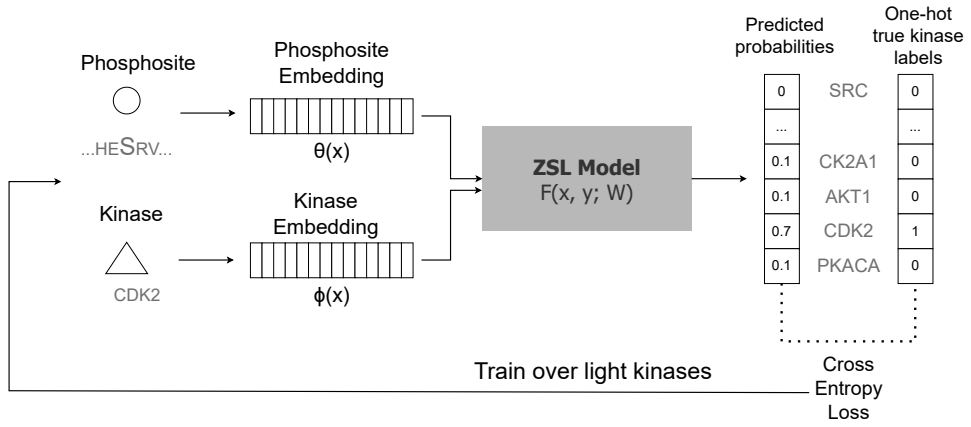



(c) Step 3: In the subsequent step, the kinase most resembling the majority train kinase in the kinase representation space is predicted as the zero-shot dark kinase.




(d) Step 4: The model finalizes its prediction by selecting the dark kinase (test kinase) most similar to the majority light kinase from the previous step.

Figure 4.2 The Visualization of the Bi-linear Zero-Shot Model (BZSM). The bilinear compatibility function $F$ takes the phosphosite and kinase embedding vectors and is trained to minimize the cross-entropy loss over light kinases.




Using the compatibility function by Sumbul et al. (2017), BZSM aims to estimate the compatibility between a given pair of phosphosite x and kinase y. The compatibility between phosphosite and kinase embeddings is learned using the formulation $F(x,y) = [\theta(x)^\top \quad 1] W [\phi(y)^\top \quad 1]^\top$ where $\theta(x) \in \mathbb{R}^d$ is the phosphosite representation, and $y \in \mathbb{R}^m$ is the kinase representation. The model is trained by minimizing the regularized cross-entropy loss:

$$\min_W - \sum_{(x,y)\in D_{tr}} \log p(y|x) + \lambda \|W\|^2 \tag{4.1}$$

where the summation runs over all kinase-phosphosite pairs available in the training set $D_{tr} = (x_i, y_i)$, and $p(y|x)$ is the softmax of $F$ over the light kinases:

$$p(y|x) = \frac{\exp F(x,y)}{\sum_{y' \in Y_{tr}} \exp F(x,y')}. \tag{4.2}$$

The $\ell_2$ regularization term in Eq. 4.1 is implemented as *weight decay* in practice. At test time, $p(y|x)$ is calculated via softmax over the test kinases.

# 5. LEVERAGING UNLABELED DATA WITH SEMI-SUPERVISED AND TRANSDUCTIVE LEARNING APPROACHES

As stated earlier, there is a vast corpus of orphan phosphosites, phosphosites whose associated kinase is not known. Even though the associated kinases of these phosphosites are unknown, the phosphosites themselves could be leveraged to learn intrinsic information from phosphorylation data, potentially improving our model's performance. We experimented with two approaches aimed at leveraging these orphan phosphosites, often referred to as unlabeled data. This section describes these two approaches, which are implemented in the DeepKinZero problem formulation: a transductive learning approach and a semi-supervised learning approach.

## 5.1 Quasi-Fully Supervised Model

Quasi-Fully Supervised Learning (QFSL) is a transductive approach in which both the train and test samples are utilized in the training process Song et al. (2018). Since both train and test samples are used in the training phase, the predictions in the training phase are made for both the train and test kinases. The unique aspect of QFSL lies in its handling of the loss function. Regular cross-entropy is applied to the training samples, where labels are known. A novel approach is used for the test samples whose labels are unavailable during training: the loss calculation includes the negative `log` sum of the predictions for these test kinases. QFSL loss is shown in Equation 5.1:

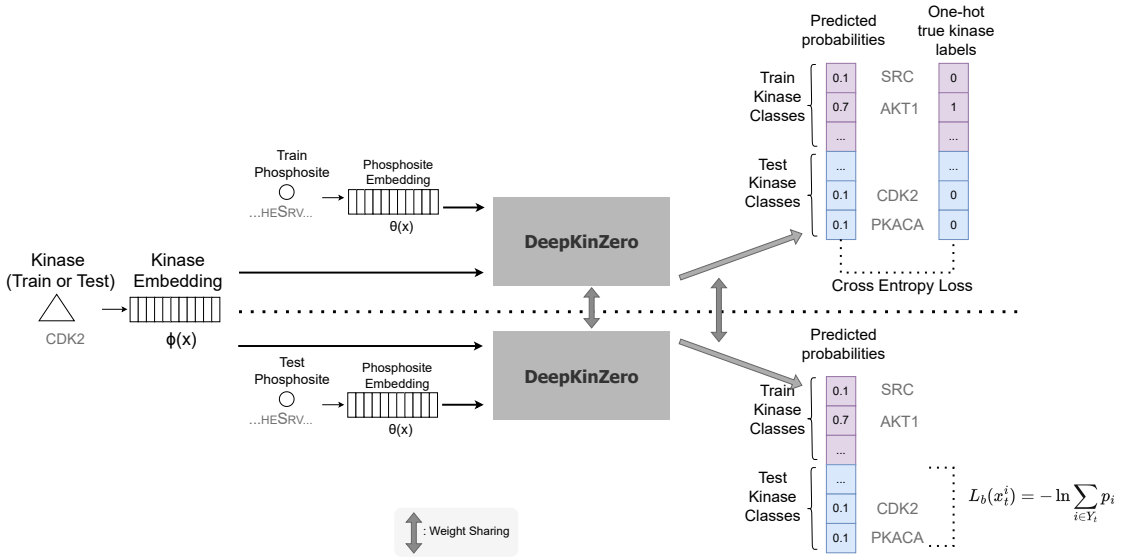
$$L = \frac{1}{N_s}\sum_{i=1}^{N_s} L_p(x_s^i) + \frac{1}{N_t}\sum_{i=1}^{N_t} \lambda L_b(x_t^i) + \gamma\Omega(W) \tag{5.1}$$

where the $L_b$ loss is defined in equation 5.2:

$$L_b(x_t^i) = -\ln \sum_{i \in Y_t} p_i \tag{5.2}$$

The loss in Equation 5.2 encourages the model to predict the test kinases for unseen test samples. The full adaptation of the QFSL loss into the DeepKinZero setup is described in Figure 5.1.

Figure 5.1 This figure illustrates the adaptation of the Quasi Fully Supervised Loss in the DeepKinZero setup, demonstrating the integration of the loss function with the model architecture. Both train and test samples are fed into the same DeepKinZero architecture, but different loss functions are applied. Cross-entropy loss is used to train phosphosites with known labels. The model's predictions on test kinase classes are summed and added to the final loss for test phosphosites with unknown labels. This encourages the model to predict test kinases for unlabeled test samples.




This model which uses QFSL, will be referred to as the Quasi-Fully Supervised Model (QFSM) throughout this study.

## 5.2 Pseudo-Labeling

To assess the integration of unlabeled data (orphan phosphosites) and test data in the training phase, a well-known method in semi-supervised learning, pseudo-labeling, is applied to the DeepKinZero model. In this section, the pseudo-labeling

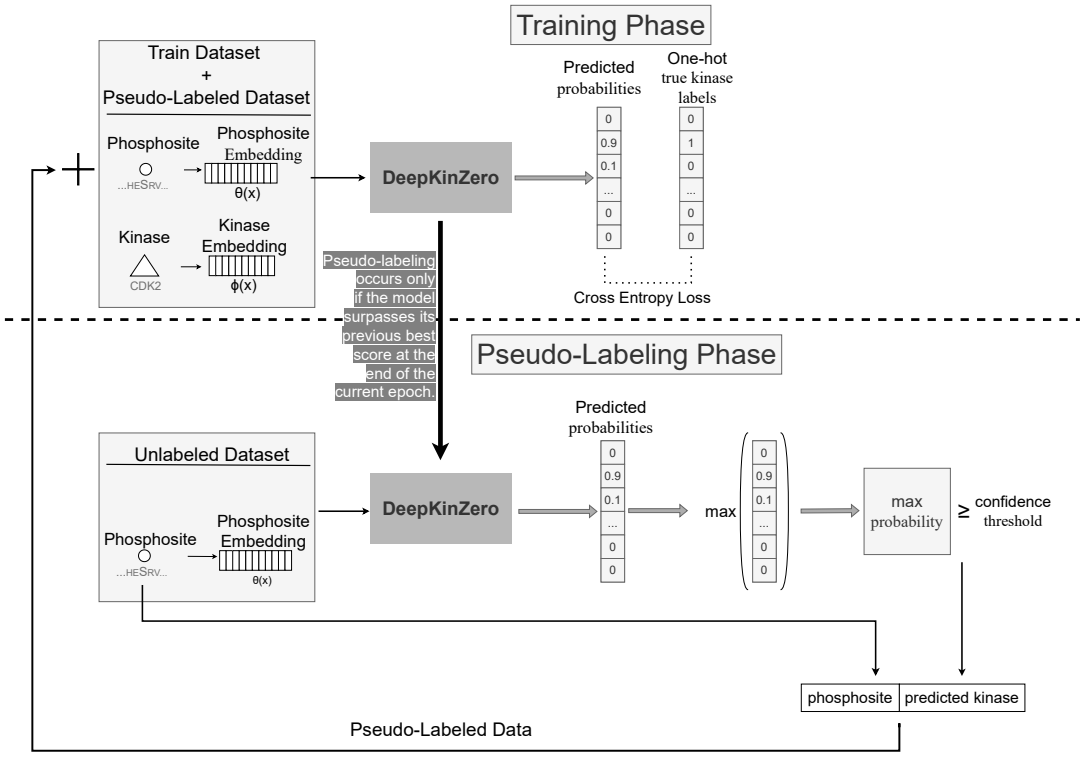
process and approach will be explained first. Next, we describe the upsampling strategies we applied to improve pseudo-labeling performance.

### 5.2.1 The Pseudo-Labeling Process

During pseudo-labeling, the vast unlabeled corpus, which consists of over 350,000 phosphosites, and the test dataset are pseudo-labeled. Initially, the model is trained on the training dataset, predicting only the train kinases. At the end of an epoch, if the model surpasses its previous highest score on the validation dataset, pseudo-labeling is applied to the unlabeled data. A sample is pseudo-labeled only if the prediction for the sample is above a specific confidence threshold. In other words, only confident predictions are added to the pseudo-labeled dataset.

Since the true labels of the orphan or test sites (the kinases that phosphorylate these sites) are not known during training, predictions are made on all train, validation, and test kinases. The pseudo-labeled set is then augmented to the labeled training set. The training set now includes associations related to both train kinases and pseudo-labeled kinases, the kinases predicted in later epochs dynamically change according to the newly defined training dataset. Since the pseudo-labeled dataset is added to the training dataset at the end of each epoch that surpasses the previous highest score, this process can be described as progressive pseudo-labeling. If the model makes predictions on the same sample in later epochs, the prediction by the latest model overwrites the previous one. The illustration of the pseudo-labeling technique applied to the DeepKinZero model is depicted in Figure 5.2.

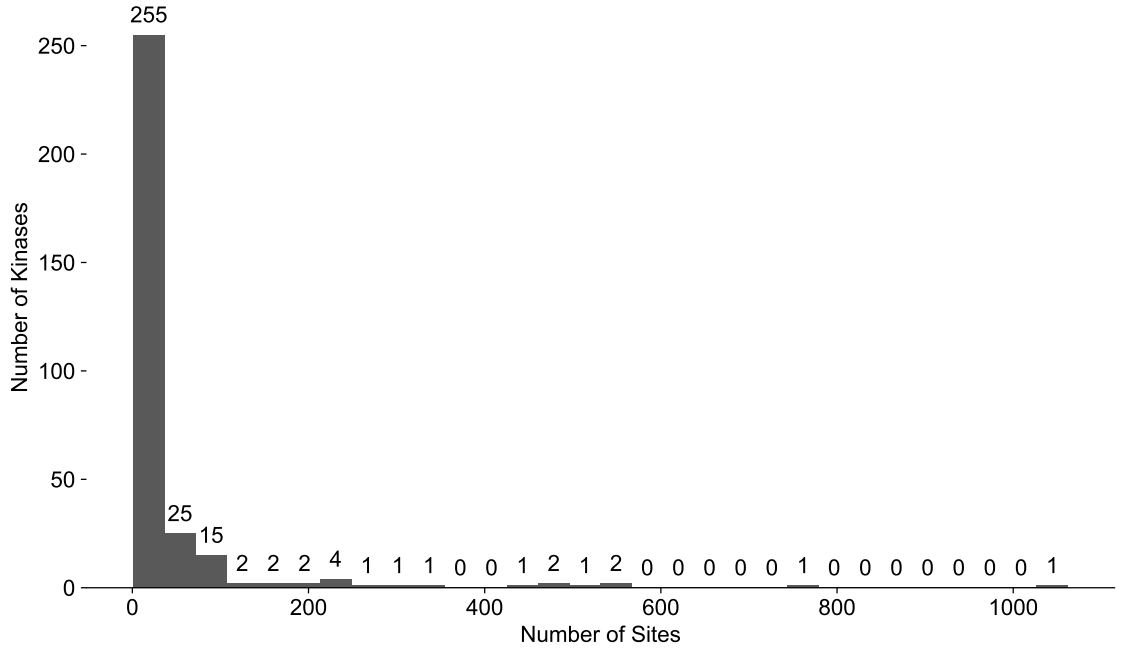
Figure 5.2 This figure illustrates the pseudo-labeling process and its integration into the DeepKinZero framework. Pseudo-labeling is applied at the end of an epoch only if the model surpasses the previous highest score. The pseudo-labeled data is then added to the training dataset to be used in subsequent epochs, hence the pseudo-labeled data is used in the training process in a progressive manner.




### 5.2.2 Upsampling in Pseudo-Labeling

As progressive pseudo-labeling is applied, it is crucial for the model to make accurate predictions during this process to avoid misleading the training phase. We implemented upsampling to address the imbalance in the training data. Due to the unbalanced data in the training set, some classes might dominate learning, leading to a drop in pseudo-labeling performance. This cumulative effect could ultimately decrease the model's overall performance. The distribution of kinase-site associations for the training kinases in the training set can be seen for the default split in Figure 5.3.

Figure 5.3 The histogram of the training kinases' phosphosite association count in the training dataset (specifically for the default split, split 1). Several kinases have 500+ site associations, while many have very few site associations, approximately 10 or less.




When upsampling, it is important to decide which kinases to upsample. Since upsampling is only applied to training kinases in training time, and since the phosphosite association counts for these train kinases are known during training, the number of phosphosite associations for each training kinase could be used for deciding which kinases to upsample. Kinases are sorted by their association counts, and a specific percentile threshold is determined to identify which kinases to upsample. Upsampling is then applied only to those kinases that fall within the lower percentile, meaning those with fewer phosphosite associations according to the predefined threshold. Additionally, we also observe that upsampling only in the first epoch and using the original dataset in the other epochs works better.

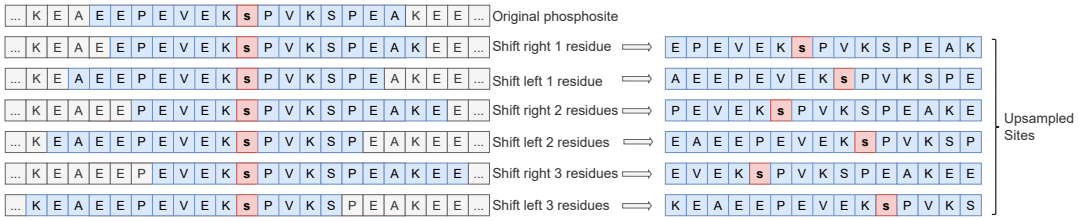
Two upsampling methods are specifically applied: duplication and shifting.

1. **Duplication:** For duplication, the original samples for the kinases that fall into the specified lower percentile's kinase-site association data are duplicated without any changes.

2. **Shifting:** Shifting is applied by sliding the phosphosite of length 15 within the protein where the phosphosite resides. Shifting is done from both the left and right, creating two samples for a single slide. The shifting process is

visualized in Figure 5.4 for better comprehension. The original phosphosite residue, known to interact directly with the kinase, is kept within the frame. Thus, shifting is applied a maximum of 7 residues to the right and left.

Figure 5.4 Illustration of the phosphosite shifting method. The 15-length phosphosite representation is shifted within the protein sequence, ensuring that the original site residue remains in the frame at all times. In this figure, the site is represented by the 's' within the red box. This site is always kept within the shifted frame.



# 6. RESULTS

This section will present and analyze the results of the zero-shot kinase-phosphosite prediction models applied to the problem formulation. Zeynep Isik obtained the results on the k-NN model, and Mert Pekey obtained the results for half of the pLMs on the bilinear model.

## 6.1 DARKIN Benchmark

Four DARKIN splits[1] are presented to ensure consistency in the experiments and to make consistent decisions on the data, particularly addressing the instability associated with zero-shot learning setups. Since the main DARKIN split used in this study was dataset split 1 (random seed 12345), the dataset statistics for this DARKIN split will be presented in this section.

We present the number of kinases, phosphosites, and kinase-phosphosite associations in the train, validation and test splits. In Figure 6.1, the sub-figure on the left shows the number of kinases in each fold. The sub-figure in the middle shows the number of phosphosites in each fold (since a site can be phosphorylated by multiple kinases, this number differs from the number of kinase-phosphosite associations). The sub-figure on the right presents the number of kinase-phosphosite associations in each fold.

[1] https://github.com/tastanlab/darkin

Figure 6.1 This figure presents three different numerical analyses of the DARKIN dataset. Left figure: Distribution of unique kinases in each set; Middle figure: Distribution of unique phosphosites in each set; Right figure: Total count of kinase-phosphorylation data in each set.

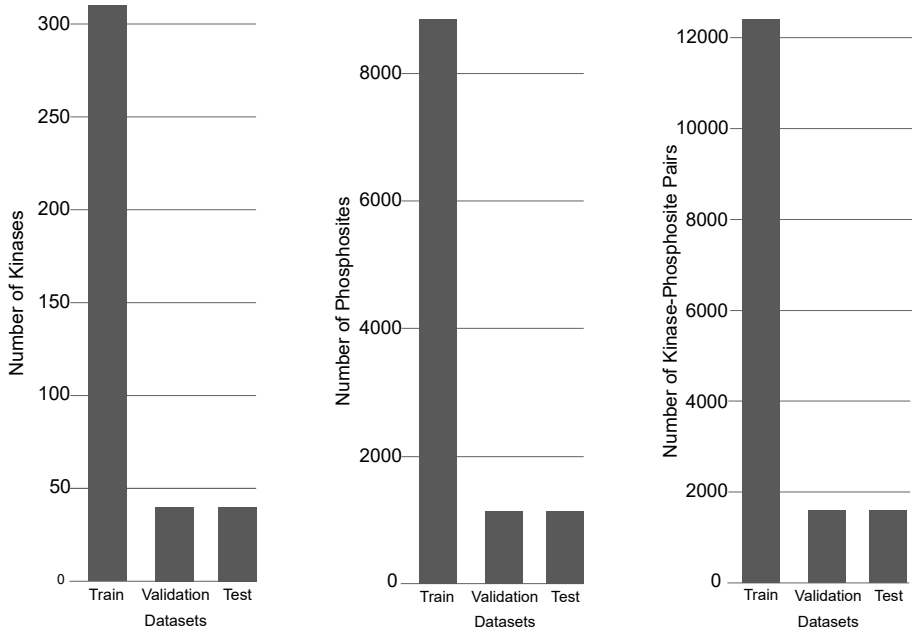
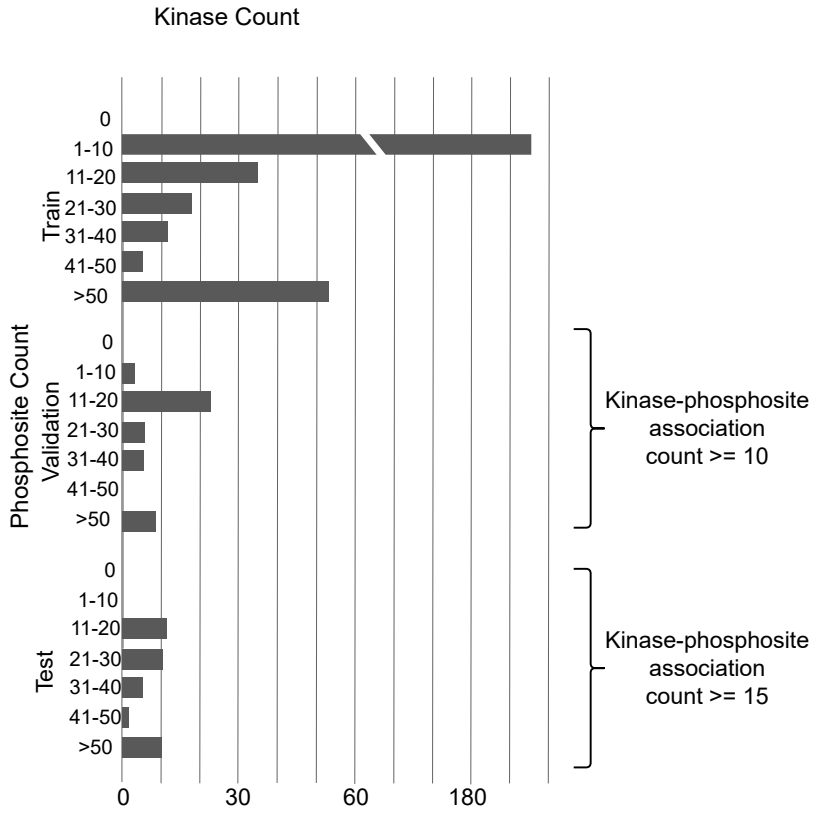


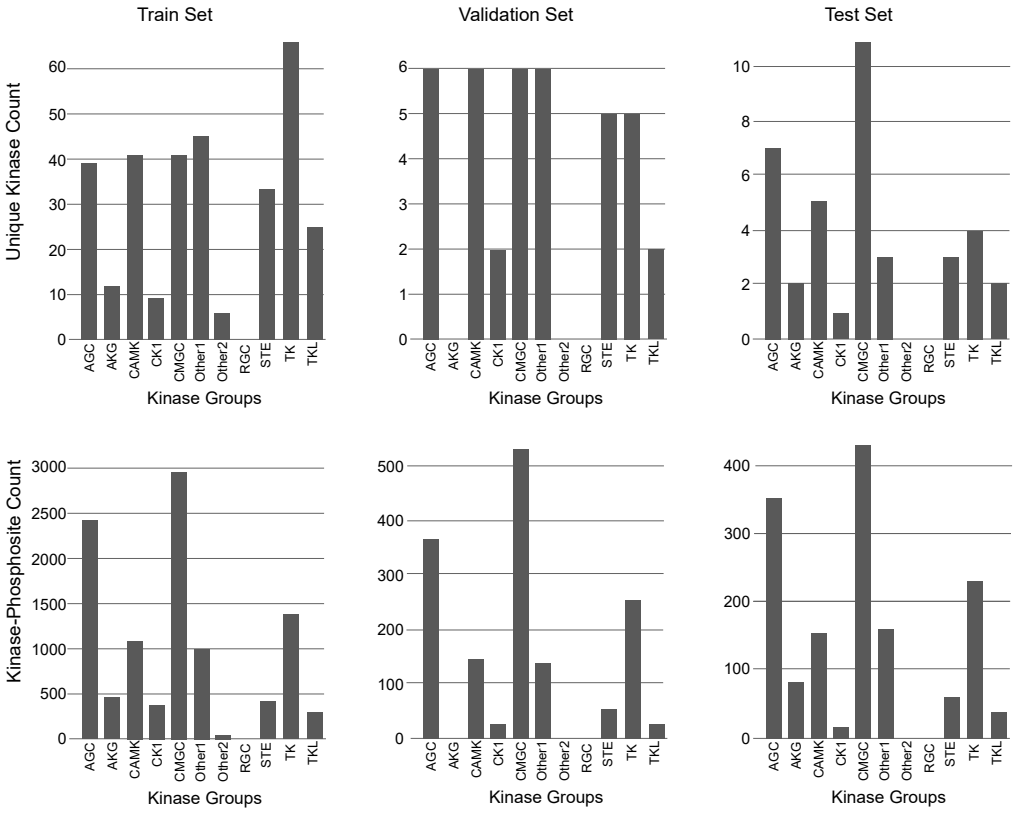

Figure 6.2 presents the distribution of site associations. This dataset has a test threshold of 15 and a validation threshold of 10; in other words, only kinases with more than 15 site associations are categorized as test kinases and only kinases with more than 10 associated sites are categorized as validation kinases. As seen in the figure, there are no kinases in the test set with fewer than 15 site associations (the bar for 1-10 site associations is of size 0).

Figure 6.2 This figure presents the histogram of the number of site associations for the kinases in each set. This dataset with this distribution has the test threshold set to 15 and the validation threshold set to 10.



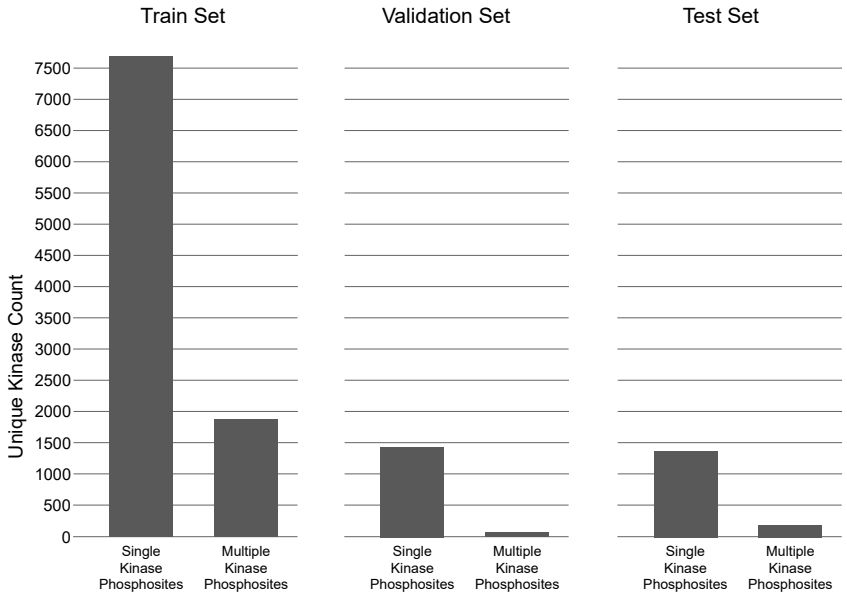

Another important statistic worth analyzing is the distribution of the associations across each kinase group in each set (train, validation, and test). This is presented in Figure 6.3. Since the data can be distributed concerning the kinase groups if the specific parameter is set to true, the kinase-site association counts are expected to be almost balanced. In situations where there were not sufficient kinases in a group, it was not possible to equally distribute the kinases and their relative site associations across the sets. For example, in the newly defined kinase group 'Other2', since there were very few kinases in this group, it was not possible to place any kinases from this group into the validation and test sets.

Figure 6.3 This figure shows the distribution of kinase counts from each kinase group in each set (upper 3 sub-figures) and the number of kinase-phosphosite associations in each set (lower 3 sub-figures).



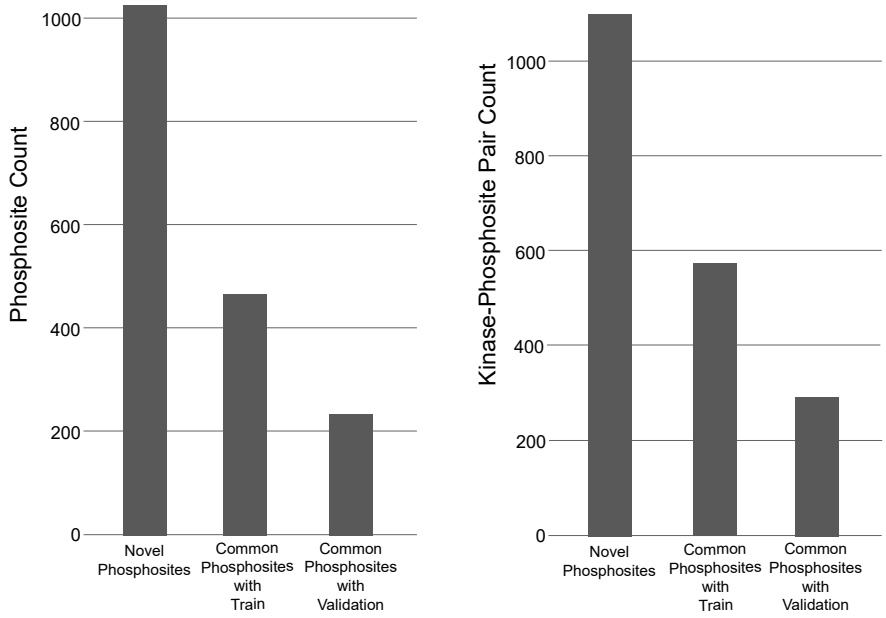

As described in previous sections, multiple kinases can phosphorylate a phosphosite. These sites can be harder to train on since the model must learn to associate the same site with several different kinases. Therefore, it is insightful to know how many sites are phosphorylated by a single kinase and how many sites are phosphorylated by several different kinases. This information can provide a sense of how challenging the splits are. This analysis can be seen in Figure 6.4.

Figure 6.4 This figure depicts the distribution of sites phosphorylated by a single kinase (Single Kinase Phosphosites) and sites phosphorylated by multiple kinases (Multiple Kinase Phosphosites).



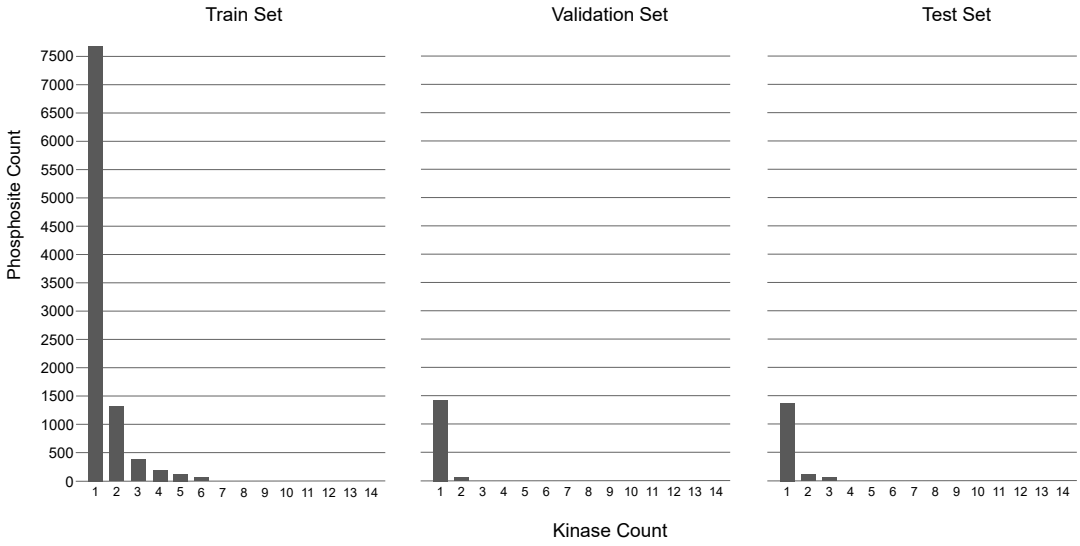

Another analysis that may provide insight into the dataset splits is the examination of the number of phosphosites observed for the first time in the test set, referred to as novel sites, as well as the common sites shared between the test and validation sets, and the common sites shared between the test and training sets. As described in the sixth step of the DARKIN dataset split generation script (refer to Section 3.2.4.6), some sites may be phosphorylated by training and test kinases. Since these sites are associated with train and test kinases, they could be assigned to both the train and test sets. When assigning these sites into the train set, test kinases are excluded as if they are not associated with this site, and when assigning these sites into the test set, similarly, train kinases are excluded as if they are not associated with this site. Consequently, some sites may appear in both the training and test sets due to this exclusion method. This method was previously explained in Section 3.2.4.6 in the figure 3.9.

Figure 6.5 The left sub-figure displays the number of phosphosites observed for the first time in the test set (Novel Phosphosites), along with the phosphosites common to both the test and training datasets and the phosphosites common to both the test and validation datasets. The right sub-figure shows the number of kinase-phosphosite associations corresponding to the novel phosphosites in the test set, the kinase-phosphosite associations corresponding to the common phosphosites between the train and test datasets, and the kinase-phosphosite associations corresponding to the common phosphosites between the validation and test datasets.




When multiple kinases phosphorylate a site, it can be more challenging to accurately predict the site's kinases, as the model must associate this site with several different kinases. Therefore, examining the distribution of sites experimentally known to be phosphorylated by a specified number of kinases may be insightful. If many sites are known to be phosphorylated by multiple kinases (e.g., six kinases), it may indicate that the split is relatively more difficult to train. The histogram of the site's kinase count associations is shown in Figure 6.6.

Figure 6.6 This plot displays the histogram of the number of sites known to be phosphorylated by the number of kinases indicated on the x-axis. For example, the plot shows that there are over 7,500 sites phosphorylated by a single kinase in the training set and approximately 1,250 sites phosphorylated by two kinases.




## 6.2 Protein Language Model Experiments on Zero-Shot Models

The experimentation results on the zero-shot k-NN model and the BZSM model, along with additional insightful experimentation results will be presented in this section.

### 6.2.1 Hyperparameter Tuning

In all cases, macro Average Precision (AP) is used on the validation set for model selection. For the k-NN based ZSL, $k$ is chosen from $3, 5, 7$. For the bilinear ZSL, hyperparameters are searched among random combinations of the parameters shown in Table 6.1. Finally, to measure the effect of initialization, unless otherwise stated, BZSM models are trained three times, and the mean and standard deviation of the macro AP values are reported.

### 6.2.2 Comparison of Protein Language Models

The effectiveness of pLM-based embeddings is initially assessed using k-NN and BZSM methods. Table 6.2 presents macro AP scores obtained through the k-NN

Table 6.1 Random search hyperparameter ranges for BZSM. This table details the parameters explored and their respective ranges.

| **Hyperparameter** | **Range** |
|---|---|
| Learning Rate | 0.000001 to 0.1 |
| Optimizer | Adam, SGD, RMSprop |
| Learning Rate Schedule | Exponential, Step, CosineAnnealing |
| Momentum | 0.95 to 0.9999 |
| Weight Decay | 0.00001 to 0.01 |

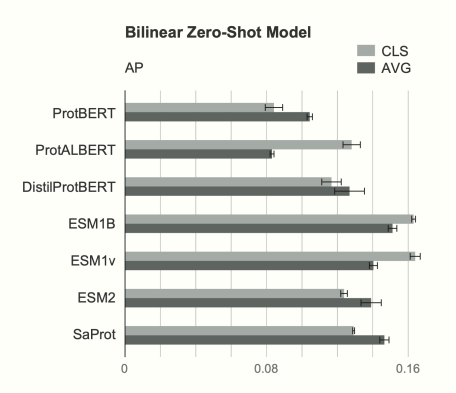
and BZSM methods when different pLM embeddings (detailed in Table 4.1) are used to represent the 15-mer around the phosphosite sequence and the kinase domain sequence. When employing pLM embeddings, embeddings sourced from the same pLM are employed for both the phosphosite and kinase sequences. To establish baseline performance, results obtained with three sequence encoding methods are also presented: one-hot encoding, BLOSUM62, and NLF encoding (Section 4.1). In both models, it is observed that most of the pLM representations perform above the baseline encodings, indicating that they capture the relevant characteristics of the protein sequences more effectively.

The TAPE embeddings perform the best among the k-NN models (0.12 AP score), while the Esm models, ProtT5-XL, are close to TAPE's results (Table 6.2). In the BZSM models, however, the TAPE embeddings fall behind ESM-1b and ESM-1v. The superior performance of TAPE in the k-NN model could be attributed to its lower dimensional vector (see Table 4.1). In BZSM, when employing the CLS token, both ESM-1b and ESM-1v exceed 0.16 macro AP. ProtT5-XL is a close third, and SaProt (CLS) also performs well.

Table 6.2 Mean macro AP of 3-NN and the BZSM using only pLM embeddings. For pLMs with CLS and average token, the best performing one is shown. The results for the 3-NN model were achieved by Zeynep Işık, while the results for the BZSM model were achieved in collaboration with Mert Pekey.

| Embedding | AP (3-NN) | AP (BZSM) |
|---|---|---|
| OneHotEnc | 0.0897 | 0.0634 ±0.0034 |
| Blosum62 | 0.0897 | 0.0327 ±0.0008 |
| NLF | 0.0902 | 0.0419 ±0.0030 |
| ProtVec | 0.0808 | 0.0959 ±0.0010 |
| ESM-1b (cls) | 0.1119 | 0.1631 ±0.0011 |
| ESM-1v (cls) | 0.1121 | **0.1640** ±0.0028 |
| ESM-2 (avg) | 0.0957 | 0.1391 ±0.0057 |
| Ankh-Large | 0.1106 | 0.0840 ±0.0012 |
| DistilProtBERT (avg) | 0.0811 | 0.1269 ±0.0084 |
| ProtBERT (avg) | 0.0540 | 0.1044 ±0.0015 |
| ProtAlbert (cls) | 0.0915 | 0.1281 ±0.0049 |
| ProteinBERT | 0.1168 | 0.1236 ±0.0023 |
| ProtGPT2 | 0.1054 | 0.1333 ±0.0020 |
| ProtT5-XL | 0.1172 | 0.1552 ±0.0011 |
| SaProt (avg) | 0.0973 | 0.1466 ±0.0026 |
| TAPE | **0.1200** | 0.1237 ±0.0018 |

Figure 6.7 Performance comparison of BZSM trained with CLS and average embedding vector for all pLMs. The results in this figure were achieved in collaboration with Mert Pekey.




### 6.2.3 CLS Token Embedding versus Averaging

Several pLMs provide a CLS token, whose embedding is commonly used as the sequence summary (Devlin et al., 2018). However, it is not clear whether the CLS token or the average of all token embeddings provides a better summary for this task. The performance differences between these two alternatives are shown in Figure 6.2.2, illustrating that (i) the results can depend on this detail, and (ii) the optimal choice varies across the pLMs.

### 6.2.4 Incorporating Additional Kinase Information

The kinase sequence embedding vectors are augmented with additional information regarding kinase family hierarchy and EC classification. One-hot encoded vectors representing this additional information are appended to the sequence embedding vectors. Only the BZSM is experimented with in this context, as it outperforms the k-NN model. Including each additional piece of information individually enhances the performance of all models (Table 6.3), with the inclusion of kinase family information yielding the most significant improvement. Using the CLS token embeddings, models based on ESM-1b, ESM-1v, and SaProt benefit the most and emerge as the top performers in this augmented setup. These findings underscore the value of

Table 6.3 This table presents the BZSM performance trained with sequence embedding and other kinase information. The mean macro APs are shown. The best-performing results of CLS and embedding averaging are shown. The results in this table were achieved in collaboration with Mert Pekey.

| Embedding | Base | + Family | + Group | + EC | + Family + Group + EC |
|---|---|---|---|---|---|
| OneHotEnc | 0.0634 | 0.1107 | 0.0832 | 0.0802 | 0.1098 |
| Blosum62 | 0.0327 | 0.0318 | 0.0310 | 0.0337 | 0.0323 |
| NLF | 0.0419 | 0.0391 | 0.0425 | 0.0400 | 0.0426 |
| ProtVec | 0.0959 | 0.1262 | 0.1129 | 0.1214 | 0.1354 |
| ProtBERT (cls) | 0.0842 | 0.1170 | 0.1077 | 0.1132 | 0.1273 |
| ProteinBERT | 0.1236 | 0.1506 | 0.1215 | 0.1367 | 0.1359 |
| ProtT5-XL | 0.1552 | 0.1701 | 0.1531 | 0.1674 | 0.1731 |
| ESM-1b (cls) | 0.1631 | **0.1740** | **0.1688** | **0.1680** | 0.1769 |
| ESM-1v (cls) | **0.1640** | 0.1737 | 0.1653 | 0.1652 | 0.1734 |
| ESM-2 (avg) | 0.1391 | 0.1588 | 0.1453 | 0.1496 | 0.1638 |
| DistilProtBERT (cls) | 0.1167 | 0.1360 | 0.1292 | 0.1287 | 0.1441 |
| ProtGPT2 | 0.1333 | 0.1476 | 0.1412 | 0.1419 | 0.1557 |
| Ankh-Large | 0.0840 | 0.1417 | 0.1135 | 0.1178 | 0.1594 |
| ProtAlbert (cls) | 0.1281 | 0.1269 | 0.1276 | 0.1285 | 0.1372 |
| SaProt (cls) | 0.1292 | 0.1696 | 0.1424 | 0.1434 | **0.1800** |
| TAPE | 0.1237 | 0.1379 | 0.1333 | 0.1310 | 0.1455 |

Table 6.4 Comparison of the two best-pLMs, ESM-1b and SaProt on four random DARKIN splits. The mean macro AP scores and their standard deviations are shown for BZSM.

| | Split 1 | Split 2 | Split 3 | Split 4 |
|---|---|---|---|---|
| ESM-1b (cls) | 0.1769 ±0.0022 | 0.1536 ±0.0020 | 0.1531 ±0.0018 | 0.1652 ±0.0020 |
| SaProt (cls) | **0.1800** ±0.0015 | **0.1599** ±0.0029 | **0.1627** ±0.0021 | **0.1690** ±0.0050 |

additional kinase categorizations that cannot be captured solely through sequence information.

### 6.2.5 Comparing the Best-Performing pLMs on Different DARKIN Splits

As ESM-1b and SaProt emerge as the two top-performing pLMs when paired with the BZSM model (Table 6.3), we further evaluated their performance on three additional random splits of the DARKIN dataset to facilitate a more comprehensive comparison between these two pLMs. While both models demonstrate competitiveness, SaProt consistently outperforms ESM-1b slightly on these four different splits (Table 6.4). The performance of SaProt underscores the added value of structural information.

## 6.3 DeepKinZero Protein Language Model Results

The DeepKinZero model represented phosphosites and protein kinases using ProtVec vectors based on word2vec (Mikolov et al., 2013). Additionally, during model training, the phosphosite embeddings were updated with the training dataset by passing

them through a long short-term memory (LSTM (Hochreiter and Schmidhuber, 1997)) neural network, which is frequently used in natural language processing. To compare the DeepKinZero model with our best-performing models, the ESM-1b (Family + Group + EC) and SaProt (Family + Group + EC) embeddings were used in a similar setup. This comparison was conducted in the four randomly partitioned DARKIN splits. The obtained results are presented in Table 6.5.

Table 6.5 In the four randomly partitioned DARKIN splits, the embeddings of ProtVec (Family + Group + EC), ESM-1b (Family + Group + EC), and SaProt (Family + Group + EC) were compared on the DeepKinZero model, both with and without LSTM. The mean macro AP scores and standard deviations for all model results are presented. The results in this table were achieved in collaboration with Mert Pekey.

| | **Split 1** | **Split 2** | **Split 3** | **Split 4** |
|---|---|---|---|---|
| ProtVec | 0.1354 ±0.0051 | 0.1342 ±0.0040 | 0.1278 ±0.0051 | 0.1511 ±0.0037 |
| ProtVec + LSTM | **0.1984** ±0.0104 | 0.1760 ±0.0034 | **0.1814** ±0.0069 | **0.2020** ±0.0066 |
| ESM-1b (cls) | 0.1769 ±0.0022 | 0.1536 ±0.0020 | 0.1531 ±0.0018 | 0.1652 ±0.0020 |
| ESM-1b (cls) + LSTM | 0.1971 ±0.0024 | **0.1810** ±0.0057 | 0.1691 ±0.0010 | 0.1967 ±0.0029 |
| SaProt (cls) | 0.1800 ±0.0015 | 0.1599 ±0.0029 | 0.1627 ±0.0021 | 0.1690 ±0.0050 |
| SaProt (cls) + LSTM | 0.1931 ±0.0041 | 0.1747 ±0.0048 | **0.1814** ±0.0057 | 0.1976 ±0.0054 |

In Table 6.5, while the ESM-1b and SaProt embeddings show significantly better performance in the BZSM method, the ProtVec embeddings generally yield better results in the DeepKinZero setup, with the contribution of the LSTM model. Table 4.1 shows that the vector dimensions for ESM-1b and SaProt embeddings are 1280, whereas the vector dimensions for ProtVec embeddings are 100 (Asgari and Mofrad, 2015b). Additionally, in ESM-1b and SaProt models, a separate vector is defined for each amino acid, whereas in ProtVec, a vector is defined for every three amino acids. Moreover, in ESM-1b and SaProt models, a CLS token is used at the beginning of a sequence, and an EOS token is used at the end. Therefore, when using ESM-1b and SaProt embeddings, the input to the LSTM is 17x1280, whereas it is 13x100 when using ProtVec embeddings. While ESM-1b and SaProt embeddings are more informative in a simpler and more straightforward model like BZSM, the larger number of tokens (13 and 17) entering the DeepKinZero model when using these embeddings compared to ProtVec might explain their slightly poorer performance in the DeepKinZero model.

## 6.4 Comparing the Performance of Kinase Domains and Active Sites

As an alternative way of representing kinase sequences, the kinase active sites have been extracted from the kinase domains (refer to Subsection 3.2.2.5). Several experiments have been conducted to evaluate whether the kinase domain or active sites better represent the kinase protein. These experiments are performed on the best-performing embeddings, ESM-1b and SaProt (refer to Table 6.3). Additionally, to introduce diversity to the embedding architectures being evaluated, results from ProtT5-XL and ProtVec are also included. ProtT5-XL is the third-best performing model after ESM-1v and does not belong to the same model family as ESM-1b, while ProtVec is the best-performing model among the classical representations.

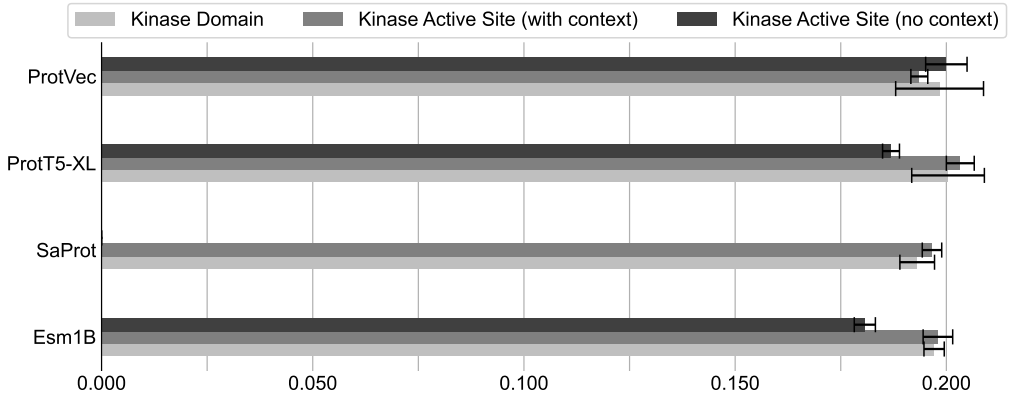
When extracting the embeddings of the active sites, two different methods were employed, referred to as 'from context' and 'not from context' representations of the active sites. The 'from context' representations involve extracting specific embeddings of each active site residue from the kinase domain embeddings and concatenating the embeddings corresponding to the active site residues from the kinase domain embedding. For the 'from context' representation of the ProtVec embeddings, all trigram vectors that include the active site amino acid were averaged.

For the 'not from context' representation of the active sites, the active site residues were concatenated in the order they appear in the multiple sequence alignment provided by Modi and Dunbrack Jr (2019), and the embedding of this sequence was extracted. The 'not from context' results for SaProt are not reported because the SaProt model requires the 3D structure of the sequence. The concatenated active site residues do not represent an actual continuous 2D protein sequence in a real-life context, as required by structure prediction models such as AlphaFold and ColabFold (Jumper et al., 2021a; Mirdita et al., 2022).

For the 'from context' representations, two specific embeddings were created: firstly, the average of all active site residue embeddings, and secondly, the average of all active site residue embeddings including the CLS token embedding. On the other hand, for the 'not from context' representations, three embeddings were generated: the first being the CLS token embedding on its own, the second being the average of all active site residue embeddings, and the third being a composite of the average of all active site residue embeddings and the CLS token embedding. These results are summarized in Figure 6.8.

The results are grouped as kinase domain, active site (from context), and active site (not from context), with only the best results within each group for each model representation being reported. The extended results are presented in the table 6.6. For each reported result, the specified model is used to represent both the kinase and phosphosite. The results are obtained by training the phosphosite embeddings with

Figure 6.8 This figure compares the performance of the kinase domains and the active sites. The presented scores are AP scores.




an LSTM model, as done in the DeepKinZero setup (Hochreiter and Schmidhuber, 1997; Deznabi et al., 2020).

As shown in Figure 6.8, in general, the kinase active site embeddings without context yield the worst results, while the active site embeddings with context provide the best results, slightly outperforming the kinase domain embeddings. The only exception to these findings is the ProtVec results, where the 'not from context' active site embeddings outperform the kinase domain embeddings while the 'from context' active site embeddings fall behind the performance of the kinase domain embeddings. The difference in performance order for ProtVec could be due to its Word2Vec-based representation, unlike the other three models (Mikolov et al., 2013).

In conclusion, the active site representations perform slightly better than the kinase domain representations in this setup. However, since the difference in performance is not significant, the kinase domain representations will still be considered, as they may provide more comprehensive information than the active sites.

## 6.5 Quasi-Fully Supervised Model (QFSM) Results

To assess the usefulness and effectiveness of QFSM, the best-performing pLMs—ESM-1b, SaProt, and ProtVec—were tested on QFSM. Since the coefficient of the quasi-fully loss affects the results (refer to Formula 5.1 for the Quasi-Fully Supervised Loss), the experimentation was conducted with three different coefficient values: 0.2, 0.5, and 1.0. Furthermore, to determine whether the results are consistent across different splits, all experiments were repeated for the four random splits of the DARKIN dataset.

The comparative results are grouped in separate tables by embedding type (SaProt, ESM-1b, and ProtVec). This presentation was chosen to demonstrate the effects of the QFSM without the influence of specific pLM performances. The results are presented in Table 6.7 for SaProt, Table 6.8 for ESM-1b and Table 6.9 for the ProtVec results.

As it could be assessed from these tables (Table 6.7 for SaProt, Table 6.8 for ESM-1b and Table 6.9 for the ProtVec), except for a few cases, the QFSM approach is generally unable to consistently surpass the original DeepKinZero results. With SaProt, the QFSM results are slightly better in two splits (the splits 3 and 4) but perform worse in the other two. For the ESM-1b results, QFSM performs slightly better in one split but worse in the remaining three splits. Finally, with the ProtVec embedding, QFSM performs worse, showing no improvements over DeepKinZero.

This may be due to the different setups required for QFSM. In the original DeepKinZero model, only the training samples enter the training phase, so the model only predicts training kinases. However, in the QFSM approach, to utilize the test samples, the test samples are also entered into the training phase. Due to the nature of this approach, the model is also expected to predict both training and test kinases. Thus, the training phase for QFSM might be causing intrinsic problems for the model when learning.

It has also been noted that QFSM performs better when using smaller coefficients, and the results decline with larger coefficient values for all three embeddings. Therefore, it could be concluded that QFSM does not benefit this problem significantly.

## 6.6 Pseudo-Labeling Results

Since up-sampling is implemented as part of pseudo-labeling, the effects of up-sampling without pseudo-labeling were examined in this section. Subsequently, the combined impact of up-sampling with pseudo-labeling was also evaluated. The detailed results of these investigations are presented in the following subsections.

### 6.6.1 Up-Sampling Results

Table 6.10 presents the effects of duplicating sites for kinases. Duplication was applied as specified in the "Duplication" column; for example, if a kinase initially has three phosphosite associations and the duplication factor is 2, then six new

duplicate samples for that kinase will be generated.

Similarly, the effects of shifting sites for kinases are presented in Table 6.11. Shifting was applied up to the specified factor in the "Shifting" column from both sides. For instance, for a single sample, if the shifting factor is 3, the site will be shifted one position to the left and right initially, then two positions left and right, and finally three positions, resulting in six new shifted site samples.

As it could be observed from Table 6.10 and Table 6.11 both duplication and shifting strategies marginally enhance the AP scores for splits 1, 2, and 3; however, these strategies do not improve performance for split 4. Therefore, it can be concluded that while up-sampling strategies are valuable for experimentation within pseudo-labeling, they should be carefully analyzed before integration into upcoming models.

### 6.6.2 Results of Pseudo-Labeling Combined with Up-Sampling

Initially, pseudo-labeling was conducted without any up-sampling. Subsequently, pseudo-labeling was performed in combination with various duplication and shifting factors. The results of these experiments can be observed in Table 6.12. The pseudo-labeling threshold was set to 0.9.

There were two potential datasets for applying pseudo-labeling: the test set and the unlabeled corpus of orphan phosphosites, whose cognate kinases are unknown. It was observed that applying pseudo-labeling solely to either dataset does not consistently improve the AP score. This inconsistency is attributed to the imbalanced nature of the DARKIN dataset. Therefore, pseudo-labeling combined with up-sampling was experimented on to address this imbalance. The results of pseudo-labeling combined with up-sampling is presented in Table 6.12. The results presented in this table align with those in the up-sampling tables (Table 6.10 and Table 6.11). Pseudo-labeling with up-sampling improved performance for splits 1, 2, and 3; however, it did not improve performance for split 4.

As a result, pseudo-labeling combined with up-sampling improves performance in severely imbalanced datasets. However, since it does not consistently enhance performance, this approach should be integrated into future models with careful analysis and inspection.

## 6.7 Error Analysis

To evaluate the intrinsic benefits of various methods including QFSM, upsampling, and upsampling with pseudo-labeling, we analyzed the scatter of AP scores against the number of training samples (kinase-phosphosite associations) related to the test samples by group and family. In other words, the count of related training kinases corresponds to the number of kinase-phosphosite associations for the training kinases from the same group or family. Calculations for 'Family' and 'Group' are conducted separately in the scatter plots. This implies that when calculating train samples from the same group, they do not necessarily belong to the same family. This approach helps ascertain if the experimented methods offer any improvement. The comparative results are depicted in four subgraphs where the upper left sub-figure represents the original DeepKinZero model, the upper right shows the QFSM results, the lower left shows the up-sampling results and the lower right shows the results for when up-sampling and pseudo-labeling are combined. These results focus solely on the test set, utilizing ProtVec embeddings for phosphosites and kinases. We concentrated on analyzing the family-based scatter plot as the family feature is more distinctive. The scatter plot and error analysis can be found in Figure 6.10.

**QFSM Analysis:** The QFSM model does not improve results overall; in fact, it slightly decreases them, as depicted in the referred figures. For instance, certain kinases like those in the 'CDK' family, despite having a higher number of training samples, show a drop in performance. Additionally, the 'CLK' family kinase also exhibits a significant decrease in AP score, despite having around 800 related training samples. However, for test kinases with fewer related training kinases, the QFSM approach shows less clustering at lower AP scores, suggesting a slight improvement for kinases with fewer training associations but a decrease in performance for those with more.

**Upsampling:** Upsampling generally improves AP scores over the baseline, particularly noticeable in lower kinase count ranges. It shows higher individual kinase AP scores compared to the DeepKinZero model. However, there does not seem to be much difference in test kinases with higher related training samples, except for a slight increase in the 'CDK' family. Notably, there appears to be a drop in AP scores for the 'MAPK' family, one of the most represented groups, suggesting that upsampling less frequent kinases might inadvertently hinder the learning process for these highly represented kinases.

**Upsampling with Pseudo-labeling:** This model achieves the highest AP scores among the experimented methods, is especially beneficial at higher kinase counts, and shows results similar to the upsampling-only approach. It records the highest individual kinase scores, notably improving performance in the 'PLK' family and

slight enhancements in the 'MAPK' and 'NEK' families. There is also an observed improvement in the 'CDK' family, which has the highest number of training samples. This method results in more stable outcomes across families, with closer AP scores among kinases within the 'EGFR' and 'DYRK' families compared to those in the DeepKinZero model, indicating that while upsampling with pseudo-labeling does not drastically change overall results, it does offer promising improvements.

Figure 6.9 This figure presents a scatter plot of the average precision (AP) scores versus the number of training samples associated with kinases belonging to the same group as the test kinase. In this figure, the embedding used for both phosphosite and kinases is ProtVec.

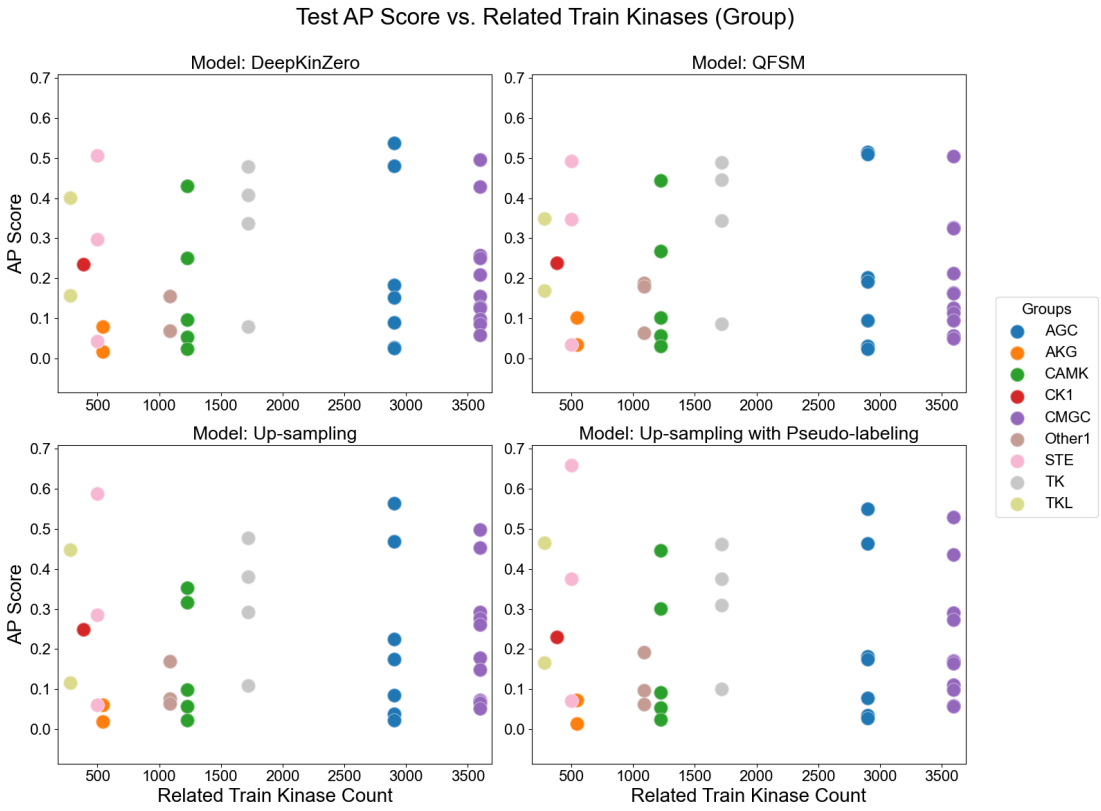


Figure 6.10 This figure presents a scatter plot of the average precision (AP) scores versus the number of training samples associated with kinases belonging to the same family as the test kinase. In this figure, the embedding used for both phosphosite and kinases is ProtVec.

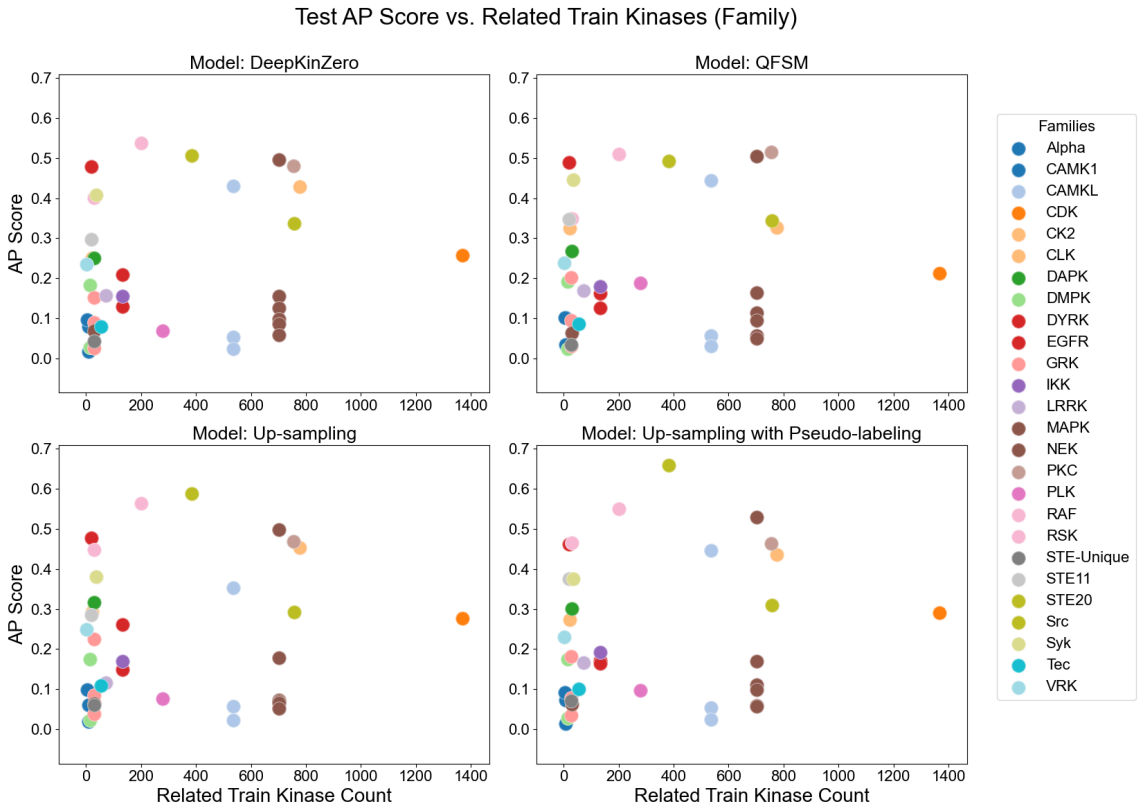


Table 6.6 The kinase domain vs. active site performance comparison in the DeepKinZero setup, where the phosphosite embeddings are trained and updated on an LSTM model. The specified pLM in the "pLM" column is used to embed both the kinase and phosphosite in each respective row.

| Split No. | pLM | Domain? | From Context? | Kinase Embedding Mode | AP Score |
|---|---|---|---|---|---|
| | ESM-1b | kinase domain | NA | cls | 0.1971 ±0.0024 |
| | ESM-1b | active site | NO | cls | 0.1797 ±0.0025 |
| | ESM-1b | active site | NO | avg with cls | 0.1807 ±0.0025 |
| | ESM-1b | active site | NO | avg | 0.1764 ±0.0009 |
| | ESM-1b | active site | YES | avg with cls | 0.1935 ±0.0020 |
| | ESM-1b | active site | YES | avg | 0.1980 ±0.0035 |
| | Protvec | kinase domain | NA | trigrams | 0.1984 ±0.0104 |
| Split 1 | Protvec | active site | NO | trigrams | 0.2000 ±0.0049 |
| | Protvec | active site | YES | trigrams | 0.1936 ±0.0020 |
| | SaProt | kinase domain | NA | cls | 0.1931 ±0.0041 |
| | SaProt | active site | YES | avg with cls | 0.1966 ±0.0023 |
| | SaProt | active site | YES | avg | 0.1961 ±0.0070 |
| | ProtT5-XL | active site | NO | avg | 0.1869 ±0.0020 |
| | ProtT5-XL | active site | YES | avg | **0.2033** ±0.0033 |
| | ProtT5-XL | kinase domain | NA | avg | 0.2004 ±0.0086 |
| | ESM-1b | kinase domain | NA | cls | 0.1810 ±0.0057 |
| | ESM-1b | active site | NO | cls | 0.1822 ±0.0029 |
| | ESM-1b | active site | NO | avg with cls | 0.1760 ±0.0027 |
| | ESM-1b | active site | NO | avg | 0.1817 ±0.0020 |
| | ESM-1b | active site | YES | avg with cls | 0.1829 ±0.0040 |
| | ESM-1b | active site | YES | avg | 0.1796 ±0.0009 |
| | Protvec | kinase domain | NA | trigrams | 0.1760 ±0.0034 |
| Split 2 | Protvec | active site | NO | trigrams | **0.1903** ±0.0051 |
| | Protvec | active site | YES | trigrams | 0.1829 ±0.0070 |
| | SaProt | kinase domain | NA | cls | 0.1748 ±0.0048 |
| | SaProt | active site | YES | avg with cls | 0.1772 ±0.0021 |
| | SaProt | active site | YES | avg | 0.1816 ±0.0086 |
| | ProtT5-XL | active site | NO | avg | 0.1866 ±0.0105 |
| | ProtT5-XL | active site | YES | avg | 0.1860 ±0.0035 |
| | ProtT5-XL | kinase domain | NA | avg | 0.1812 ±0.0078 |
| | ESM-1b | kinase domain | NA | cls | 0.1691 ±0.0010 |
| | ESM-1b | active site | NO | cls | 0.1635 ±0.0039 |
| | ESM-1b | active site | NO | avg with cls | 0.1672 ±0.0036 |
| | ESM-1b | active site | NO | avg | 0.1663 ±0.0010 |
| | ESM-1b | active site | YES | avg with cls | 0.1799 ±0.0060 |
| | ESM-1b | active site | YES | avg | 0.1792 ±0.0028 |
| | Protvec | kinase domain | NA | trigrams | 0.1814 ±0.0069 |
| Split 3 | Protvec | active site | NO | trigrams | 0.1830 ±0.0024 |
| | Protvec | active site | YES | trigrams | 0.1819 ±0.0023 |
| | SaProt | kinase domain | NA | cls | 0.1814 ±0.0057 |
| | SaProt | active site | YES | avg with cls | 0.1822 ±0.0064 |
| | SaProt | active site | YES | avg | 0.1759 ±0.0052 |
| | ProtT5-XL | active site | NO | avg | 0.1795 ±0.0009 |
| | ProtT5-XL | active site | YES | avg | **0.1849** ±0.0044 |
| | ProtT5-XL | kinase domain | NA | avg | 0.1796 ±0.0082 |
| | ESM-1b | kinase domain | NA | cls | 0.1967 ±0.0029 |
| | ESM-1b | active site | NO | cls | 0.1861 ±0.0025 |
| | ESM-1b | active site | NO | avg with cls | 0.1854 ±0.0017 |
| | ESM-1b | active site | NO | avg | 0.1851 ±0.0078 |
| | ESM-1b | active site | YES | avg with cls | 0.2004 ±0.0015 |
| | ESM-1b | active site | YES | avg | 0.2026 ±0.0056 |
| | Protvec | kinase domain | NA | trigrams | 0.2020 ±0.0066 |
| Split 4 | Protvec | active site | NO | trigrams | **0.2037** ±0.0033 |
| | Protvec | active site | YES | trigrams | 0.2030 ±0.0143 |
| | SaProt | kinase domain | NA | cls | 0.1976 ±0.0054 |
| | SaProt | active site | YES | avg with cls | 0.1842 ±0.0023 |
| | SaProt | active site | YES | avg | 0.1914 ±0.0037 |
| | ProtT5-XL | active site | NO | avg | 0.1893 ±0.0082 |
| | ProtT5-XL | active site | YES | avg | 0.1957 ±0.0064 |
| | ProtT5-XL | kinase domain | NA | avg | 0.2004 ±0.0056 |

Table 6.7 This table presents a comparative analysis of the Quasi-Fully Supervised Model (QFSM) across four distinct DARKIN splits, utilizing SaProt for both kinase and phosphosite embeddings. The results are compared with those from DeepKinZero, which uses ProtVec for its kinase and phosphosite embeddings. The QFSM has been trained using four different coefficient values: 0.2, 0.5, and 1.0.

| | **Split 1** | **Split 2** | **Split 3** | **Split 4** |
|---|---|---|---|---|
| DeepKinZero | **0.1897** ±0.0028 | **0.1585** ±0.0022 | 0.1630 ±0.0061 | 0.1950 ±0.0034 |
| QFSM (coefficient 0.2) | 0.1832 ±0.0066 | 0.1528 ±0.0022 | **0.1677** ±0.0061 | **0.1993** ±0.0028 |
| QFSM (coefficient 0.5) | 0.1613 ±0.0051 | 0.1288 ±0.004 | 0.1416 ±0.0041 | 0.1793 ±0.0036 |
| QFSM (coefficient 1.0) | 0.1502 ±0.008 | 0.1179 ±0.01 | 0.1215 ±0.0059 | 0.1572 ±0.0046 |

Table 6.8 This table presents a comparative analysis of the Quasi-Fully Supervised Model (QFSM) across four distinct DARKIN splits, utilizing ESM-1b for both kinase and phosphosite embeddings. The results are compared with those from DeepKinZero, which uses ProtVec for its kinase and phosphosite embeddings. The QFSM has been trained using four different coefficient values: 0.2, 0.5, and 1.0.

| | **Split 1** | **Split 2** | **Split 3** | **Split 4** |
|---|---|---|---|---|
| DeepKinZero | **0.1956** ±0.0051 | **0.1821** ±0.003 | 0.1678 ±0.0072 | **0.2001** ±0.0061 |
| QFSM (coefficient 0.2) | 0.1953 ±0.0002 | 0.1739 ±0.0034 | **0.1687** ±0.0033 | 0.1993 ±0.0035 |
| QFSM (coefficient 0.5) | 0.1700 ±0.0043 | 0.1742 ±0.0065 | 0.1564 ±0.0009 | 0.1893 ±0.0046 |
| QFSM (coefficient 1.0) | 0.1647 ±0.0063 | 0.1617 ±0.0068 | 0.1454 ±0.0082 | 0.1778 ±0.0013 |

Table 6.9 This table presents a comparative analysis of the Quasi-Fully Supervised Model (QFSM) across four distinct DARKIN splits, utilizing ProtVec for both kinase and phosphosite embeddings. The results are compared with those from DeepKinZero, which also uses ProtVec for its kinase and phosphosite embeddings. The QFSM has been trained using four different coefficient values: 0.2, 0.5, and 1.0.

| | **Split 1** | **Split 2** | **Split 3** | **Split 4** |
|---|---|---|---|---|
| DeepKinZero | **0.1964** ±0.0064 | **0.17635** ±0.0097 | **0.1823** ±0.0028 | **0.2087** ±0.0059 |
| QFSM (coefficient 0.2) | 0.1888 ±0.0075 | 0.1657 ±0.0058 | 0.1763 ±0.0037 | 0.2023 ±0.0011 |
| QFSM (coefficient 0.5) | 0.1805 ±0.007 | 0.1706 ±0.0065 | 0.1717 ±0.0085 | 0.2067 ±0.0014 |
| QFSM (coefficient 1.0) | 0.1723 ±0.0057 | 0.1675 ±0.007 | 0.1692 ±0.005 | 0.1972 ±0.0009 |

Table 6.10 The effects of up-sampling through duplication are depicted in the table below. All other variables were controlled, with only kinases in the 75th lower quartile being up-sampled. No shifting was applied in this analysis.

| **Model** | **Duplication** | **Split 1** | **Split 2** | **Split 3** | **Split 4** |
|---|---|---|---|---|---|
| ProtVec | NA | 0.1937 ±0.0072 | 0.1735 ±0.006 | 0.1788 ±0.0057 | **0.2108** ±0.0052 |
| ProtVec | 1 | 0.1960 ±0.0068 | 0.1670 ±0.0039 | 0.1819 ±0.0024 | 0.2076 ±0.0017 |
| ProtVec | 5 | **0.2017** ±0.0044 | 0.1683 ±0.0078 | **0.1826** ±0.0006 | 0.2095 ±0.0027 |
| ProtVec | 10 | 0.1933 ±0.0830 | **0.1762** ±0.0024 | 0.1783 ±0.0057 | 0.2027 ±0.0022 |
| ProtVec | 15 | 0.1979 ±0.0046 | 0.1723 ±0.0064 | 0.1762 ±0.0015 | 0.2061 ±0.0026 |

Table 6.11 The effects of up-sampling through shifting sites are depicted in the table below. All other variables were controlled, with only kinases in the 75th lower quartile being up-sampled. No duplication was applied in this analysis.

| **Model** | **Shifting** | **Split 1** | **Split 2** | **Split 3** | **Split 4** |
|---|---|---|---|---|---|
| ProtVec | NA | 0.1937 ±0.0072 | 0.1735 ±0.006 | 0.1788 ±0.0057 | **0.2108** ±0.0052 |
| ProtVec | 1 | 0.1938 ±0.0052 | 0.1647 ±0.0052 | 0.1813 ±0.0021 | 0.2060 ±0.0004 |
| ProtVec | 3 | 0.1968 ±0.0066 | 0.1695 ±0.0034 | 0.1818 ±0.0063 | 0.2081 ±0.003 |
| ProtVec | 5 | 0.2019 ±0.0085 | 0.1711 ±0.0004 | 0.1816 ±0.0037 | 0.2013 ±0.0062 |
| ProtVec | 7 | **0.2030** ±0.002 | **0.177** ±0.0068 | **0.1846** ±0.0042 | 0.2059 ±0.0087 |

Table 6.12 This table summarizes the pseudo-labeling results. The 'Model' column indicates the embedding used for the phosphosite and kinase in the pseudo-labeling setup. The 'Pseudo-labeled set' column specifies which set is used for pseudo-labeling ('test' refers to the test set, and 'unlabeled data' refers to the corpus of orphan sites whose cognate kinase is missing in the literature). The 'Up-sampling' column shows the combination of up-sampling techniques applied to the training set.

| **Model** | **Pseudo-labeled set** | **Up-sampling** | **Split 1** | **Split 2** | **Split 3** | **Split 4** |
|---|---|---|---|---|---|---|
| ProtVec | NA | None | 0.1937 ±0.0072 | 0.1735 ±0.006 | 0.1788 ±0.0057 | **0.2108** ±0.0052 |
| ProtVec | test set | None | 0.1926 ±0.0056 | 0.1642 ±0.0016 | 0.1758 ±0.0051 | 0.2021 ±0.002 |
| ProtVec | unlabeled data | None | 0.1959 ±0.0011 | 0.1726 ±0.0013 | 0.1761 ±0.0019 | 0.2086 ±0.0082 |
| ProtVec | test set | Duplicate: 5<br>Shifting: 5 | **0.2087** ±0.0023 | 0.1765 ±0.0052 | 0.1860 ±0.0034 | 0.2067 ±0.0098 |
| ProtVec | test set | Duplicate: 10<br>Shifting: 0 | 0.1962 ±0.0028 | 0.1686 ±0.0015 | 0.1831 ±0.0029 | 0.1922 ±0.0018 |
| ProtVec | test set | Duplicate: 6<br>Shifting: 3 | 0.1955 ±0.0046 | 0.1749 ±0.0004 | **0.1866** ±0.0049 | 0.2037 ±0.0051 |
| ProtVec | test set | Duplicate: 20<br>Shifting: 6 | 0.1981 ±0.0037 | **0.1797** ±0.005 | 0.1854 ±0.0029 | 0.1959 ±0.0028 |

# 7. CONCLUSION & FUTURE WORK

Phosphorylation is a key post-translational modification that regulates protein function and is thus a crucial biological process. This modification is catalyzed by kinases, which bind a phosphate group to a substrate protein at a specific amino acid residue termed the phosphosite. Dysfunctions in phosphorylation are known to cause numerous diseases, highlighting the importance of understanding which kinases prefer specific substrates and residues. As kinases can serve as significant drug targets in treating various diseases, research into kinase-phosphosite associations is critical.

Despite advances in phosphoproteomic studies identifying numerous phosphosites, pinpointing the cognate kinases for these sites remains challenging. Furthermore, numerous orphan sites with unidentified associated kinases still exist. Given that laboratory experiments are costly and time-consuming, computational methods can expedite the progress in this field. Extensive research has aimed at developing computational approaches to predict phosphosites in a given sequence and determine the specific kinases responsible for their phosphorylation. However, most of these studies employ supervised learning methods or rely on experimentally validated data, which is not available for a large portion of kinases. A large portion of kinases remains understudied. These are referred to as "dark kinases". Zero-shot learning approaches, in which a model can predict classes it has not encountered during training, are particularly applicable to this challenge. This thesis builds upon the previously developed zero-shot model, DeepKinZero, to advance the field of kinase-phosphosite prediction. It specifically aims to identify the cognate kinases for numerous orphan sites.

This thesis study presents several contributions, with each section specifically structured to focus on a separate contribution. The first contribution is the curation of a zero-shot learning benchmark dataset. We curate a zero-shot learning dataset named DARKIN, which is a reproducible dataset. Different random splits over this dataset can be generated by setting various parameter values. The splitting considers the phosphosite associations of each kinase, as well as the kinase group

membership and sequence similarity. DARKIN has been made publicly available[1]. Additionally, we provide a script that facilitates numerous statistical analyses on the generated dataset splits. This script allows for a deeper understanding of the nature of the data, such as assessing imbalances and examining the distribution of kinase groups.

The second contribution of this thesis study is benchmarking the performances of several protein language models on the DARKIN dataset. We introduce two zero-shot models: a training-free k-NN model and a bidirectional zero-shot model (BZSM). These models are deliberately kept simple to evaluate the performance of various protein language models (pLMs) using the DARKIN benchmark dataset. Extensive experiments with these pLMs have demonstrated the superior performance of the ESM-1b and SaProt models. ProtT5-XL also performs respectably, though it does not reach the same level of efficacy but still shows notable results. Our experiments indicate that the effectiveness of using the CLS token (the summary token of the protein) versus the average token varies from one pLM to another. To determine the best-performing pLM, we conducted additional experiments on different random splits of the DARKIN dataset. These experiments showed that SaProt consistently outperforms the ESM-1b model slightly. This finding underscores the importance of utilizing 3D structures in pLMs, as SaProt incorporates the 3D structure of proteins. Additionally, incorporating features such as kinase family and group and kinase EC number enhances performance.

We also observed improvements in model performance when fine-tuning the phosphosite embeddings with an LSTM model. However, within the DeepKinZero framework, the best-performing pLMs could not surpass the original DeepKinZero model, which uses ProtVec. We also re-evaluated kinase embeddings using the active sites of the kinases, where active sites are defined as the 29 residue locations known to play a significant role in phosphorylation. Comparing the active site representations to kinase domain representations across various pLMs with different architectures, we found that active sites generally perform slightly better than the kinase domain representations. This is a notable finding, as using just 29 residues holds the same or even slightly greater representational power than the kinase domain, which is, on average, 289 residues, with a median sequence length of 264 residues.

The third contribution of this study involves experimentation on transductive models to leverage orphan (unlabeled) phosphosites and to explore how much information can be utilized from sites whose cognate kinase is not identified. Transductive learning, a sub-field of machine learning, incorporates test samples during the training

[1] https://github.com/tastanlab/darkin

phase. Initially, a simple transductive model, the Quasi-Fully Supervised Model (QFSM), was integrated into the DeepKinZero setup. This model incorporates test samples into the training phase and encourages the model to predict test classes for test samples by reducing the loss whenever any test kinase is predicted for a test sample. It should also be noted that the ground truth labels of the test samples are not used in the training phase; the model only decreases the loss if any test kinase is predicted for a test sample, but not necessarily the correct ground truth kinase. However, the QFSM has not improved over the original DeepKinZero results, which could be due to the nature of the QFSM model. In the original DeepKinZero model, only training kinases are predicted during training, but the QFSM setup also requires predicting test kinases during training, potentially complicating the training phase. Thus, this model has proven to be ineffective for the DeepKinZero setup.

The last contribution is an experimentation with pseudolabeling in the transductive setup. Given the imbalanced nature of the DARKIN dataset splits, the effects of upsampling were also experimented with. Two upsampling methods were employed: one uses multiple copies of the sites of the kinases, and another shifts the site representation within the substrate protein, ensuring that the original phosphosite always stays in the frame. Experiments on upsampling have shown that upsampling generally improves performance for some splits. However, it does not enhance performance across all random splits, suggesting its use should be limited to cases where it is proven beneficial. Further experiments with pseudolabeling and upsampling revealed that pseudolabeling alone does not yield performance improvements. However, the splits that benefited from upsampling also saw improvements when pseudolabeling was applied to the upsampled data.

Identifying the cognate kinase of an identified phosphosite remains an important challenge. The experiments conducted in this thesis study have provided insights into the problem, resulting in some improvements, but pointed out that there is still significant potential for further exploration. Many aspects of this issue remain unaddressed. For future research, strategies to overcome the data imbalance in the DARKIN setup should be considered, as upsampling has been shown to enhance the results of pseudolabeling. Different pseudolabeling strategies could also be implemented, as the features of orphan sites hold considerable potential knowledge to be utilized. Furthermore, additional features such as protein-protein interactions and protein domains could improve the representation power of kinases and phosphosites, thus the inclusion of these features in phosphosite and kinase representation will be considered.

# BIBLIOGRAPHY


Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. (2019). Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.

Asgari, E. and Mofrad, M. R. (2015a). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, 10(11):e0141287.

Asgari, E. and Mofrad, M. R. K. (2015b). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE*, 10(11):e0141287.

Bairoch, A. (2000). The enzyme database in 2000. *Nucleic acids research*, 28(1):304–305.

Barrow, H. G., Tenenbaum, J. M., Bolles, R. C., and Wolf, H. C. (1977). Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop*, pages 21–27. Science Applications, Inc.

Bignone, P. A., Lee, K., Liu, Y., Emilion, G., Finch, J., Soosay, A., Charnock, F., Beck, S., Dunham, I., Mungall, A., et al. (2007). Rps6ka2, a putative tumour suppressor gene at 6q27 in sporadic epithelial ovarian cancer. *Oncogene*, 26(5):683–700.

Blom, N., Gammeltoft, S., and Brunak, S. (1999). Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. *Journal of molecular biology*, 294(5):1351–1362.

Blume-Jensen, P. and Hunter, T. (2001). Oncogenic kinase signalling. *Nature*, 411(6835):355.

Bo, L., Dong, Q., and Hu, Z. (2021). Hardness sampling for self-training based transductive zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16499–16508.

Bodenmiller, B., Campbell, D., Gerrits, B., Lam, H., Jovanovic, M., Picotti, P., Schlapbach, R., and Aebersold, R. (2008). Phosphopep—a database of protein phosphorylation sites in model organisms. *Nature biotechnology*, 26(12):1339–1340.

Born, J., Huynh, T., Stroobants, A., Cornell, W. D., and Manica, M. (2021). Active site sequence representations of human kinases outperform full sequence representations for affinity prediction and inhibitor generation: 3d effects in a 1d model. *Journal of Chemical Information and Modeling*, 62(2):240–257.

Bradley, D. and Beltrao, P. (2019). Evolution of protein kinase substrate recognition at the active site. *PLoS biology*, 17(6):e3000341.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. (2022). ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110.

Cann, M. L., McDonald, I. M., East, M. P., Johnson, G. L., and Graves, L. M. (2017). Measuring kinase activity—a global challenge. *Journal of Cellular Biochemistry*, 118(11):3595–3606.

Casnellie, J. E. (1991). [9] assay of protein kinases using peptides with basic residues for phosphocellulose binding. In *Methods in enzymology*, volume 200, pages 115–120. Elsevier.

Casnellie, J. E. and Krebs, E. G. (1984). The use of synthetic peptides for defining the specificity of typrosine protein kinases. *Advances in Enzyme Regulation*, 22:501–515.

Chen, M., Zhang, W., Gou, Y., Xu, D., Wei, Y., Liu, D., Han, C., Huang, X., Li, C., Ning, W., et al. (2023). Gps 6.0: an updated server for prediction of kinase-specific phosphorylation sites in proteins. *Nucleic acids research*, 51(W1):W243–W250.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., et al. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423.

Cohen, P. (2000). The regulation of protein function by multisite phosphorylation–a 25 year update. *Trends in biochemical sciences*, 25(12):596–601.

Cohen, P. (2002). The origins of protein phosphorylation. *Nature cell biology*, 4(5):E127–E130.

Cohen, P., Cross, D., and Jänne, P. A. (2021). Kinase drug discovery 20 years after imatinib: progress and future directions. *Nature reviews drug discovery*, 20(7):551–569.

Consortium, U. (2018). Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 46(D1):D158–D169.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

DeepMind (2023). AlphaFold Protein Structure Database. https://alphafold.ebi.ac.uk/. [Online; accessed 2023-11-09].

DeepMind and European Bioinformatics Institute (2023). AlphaFold API. https://alphafold.ebi.ac.uk/api-docs. Accessed: 2023-10-24.

Delom, F. and Chevet, E. (2006). Phosphoprotein analysis: from proteins to proteomes. *Proteome science*, 4:1–12.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Deznabi, I., Arabaci, B., Koyutürk, M., and Tastan, O. (2020). Deepkinzero: zero-shot learning for predicting kinase–phosphosite associations involving understudied kinases. *Bioinformatics*, 36(12):3652–3661.

Diella, F., Cameron, S., Gemünd, C., Linding, R., Via, A., Kuster, B., Sicheritz-Pontén, T., Blom, N., and Gibson, T. J. (2004). Phospho. elm: a database of experimentally verified phosphorylation sites in eukaryotic proteins. *BMC bioinformatics*, 5:1–5.

Dinkel, H., Chica, C., Via, A., Gould, C. M., Jensen, L. J., Gibson, T. J., and Diella, F. (2010). Phospho. elm: a database of phosphorylation sites—update 2011. *Nucleic acids research*, 39(suppl_1):D261–D267.

Dou, Y., Yao, B., and Zhang, C. (2014). Phosphosvm: prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine. *Amino acids*, 46:1459–1469.

Dummler, B. A., Hauge, C., Silber, J., Yntema, H. G., Kruse, L. S., Kofoed, B., Hemmings, B. A., Alessi, D. R., and Frodin, M. (2005). Functional characterization of human rsk4, a new 90-kda ribosomal s6 kinase, reveals constitutive activation in most cell types. *Journal of Biological Chemistry*, 280(14):13304–13314.

Dunker, A. K., Romero, P., Obradovic, Z., Garner, E. C., and Brown, C. J. (2000). Intrinsic protein disorder in complete genomes. *Genome informatics*, 11:161–171.

Duong-Ly, K. C. and Peterson, J. R. (2013). The human kinome and kinase inhibition. *Current protocols in pharmacology*, 60(1):2–9.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467.

Eid, S., Turk, S., Volkamer, A., Rippmann, F., and Fulle, S. (2017). Kinmap: a web-based tool for interactive navigation through human kinome data. *BMC Bioinformatics*, 18:16.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Elnaggar, A., Essam, H., Salah-Eldin, W., Moustafa, W., Elkerdawy, M., Rochereau, C., and Rost, B. (2023). Ankh: Optimized protein language model unlocks general-purpose modelling.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Yu, W., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. (2021). Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Essegian, D., Khurana, R., Stathias, V., and Schürer, S. C. (2020). The clinical kinase index: a method to prioritize understudied kinases as drug targets for the treatment of cancer. *Cell Reports Medicine*, 1(7).

Fan, J., Ma, X., Wu, L., Zhang, F., Yu, X., and Zeng, W. (2019). Light gradient boosting machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. *Agricultural water management*, 225:105758.

Ferruz, N., Schmidt, S., and Höcker, B. (2022). A deep unsupervised language model for protein design. *bioRxiv*.

Fischer, E. H. and Krebs, E. G. (1955). Conversion of phosphorylase b to phosphorylase a in muscle extracts. *Journal of Biological Chemistry*, 216(1):121–132.

Gao, J., Thelen, J. J., Dunker, A. K., and Xu, D. (2010). Musite, a tool for global prediction of general and kinase-specific phosphorylation sites. *Molecular & Cellular Proteomics*, 9(12):2586–2600.

Garnier, J., Osguthorpe, D. J., and Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of molecular biology*, 120(1):97–120.

Geffen, Y., Ofran, Y., and Unger, R. (2022). Distilprotbert: a distilled protein language model used to distinguish between real proteins and their randomly shuffled counterparts. *Bioinformatics*, 38(Supplement_2):ii95–ii98.

Glickman, J. F. (2012). Assay development for protein kinase enzymes. *Assay Guidance Manual [Internet]*.

Gnad, F., Ren, S., Cox, J., Olsen, J. V., Macek, B., Oroshi, M., and Mann, M. (2007). Phosida (phosphorylation site database): management, structural and evolutionary investigation, and prediction of phosphosites. *Genome biology*, 8:1–13.

Graves, A. and Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., et al. (2024). Simulating 500 million years of evolution with a language model. *bioRxiv*, pages 2024–07.

Heineke, J. and Molkentin, J. D. (2006). Regulation of cardiac hypertrophy by intracellular signalling pathways. *Nature reviews Molecular cell biology*, 7(8):589–600.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hornbeck, P. V., Kornhauser, J. M., Tkachev, S., Zhang, B., Skrzypek, E., Murray, B., Latham, V., and Sullivan, M. (2012). Phosphositeplus: a comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse. *Nucleic acids research*, 40(D1):D261–D270.

Hornbeck, P. V., Zhang, B., Murray, B., Kornhauser, J. M., Latham, V., and Skrzypek, E. (2014). Phosphositeplus, 2014: mutations, ptms and recalibrations. *Nucleic acids research*, 43(D1):D512–D520.

Hornbeck, P. V., Zhang, B., Murray, B., Kornhauser, J. M., Latham, V., and Skrzypek, E. (2015). Phosphositeplus, 2014: mutations, ptms and recalibrations. *Nucleic Acids Research*, 43:D512–D520.

Hunter, T. (1995). Protein kinases and phosphatases: the yin and yang of protein phosphorylation and signaling. *Cell*, 80(2):225–236.

Janknecht, R. (2003). Regulation of the er81 transcription factor and its coactivators by mitogen-and stress-activated protein kinase 1 (msk1). *Oncogene*, 22(5):746–755.

JoaoRodrigues (2016). seq_align. https://gist.github.com/JoaoRodrigues/8c2f7d2fc5ae38fc9cb2.

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology*, volume 121, pages 471–495. Elsevier.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021a). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

Jumper, J. M., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D. A., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021b). Highly accurate protein structure prediction with alphafold. *Nature*, 596:583 – 589.

Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., and Tanabe, M. (2016). Kegg as a reference resource for gene and protein annotation. *Nucleic acids research*, 44(D1):D457–D462.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. (2016). Character-aware neural language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Krause, B., Lu, L., Murray, I., and Renals, S. (2016). Multiplicative lstm for sequence modelling. *arXiv preprint arXiv:1609.07959*.

Krupa, A. and Srinivasan, N. (2002). The repertoire of protein kinases encoded in the draft version of the human genome: atypical variations and uncommon domain combinations. *Genome biology*, 3:1–14.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lee, B. and Richards, F. M. (1971). The interpretation of protein structures: estimation of static accessibility. *Journal of molecular biology*, 55(3):379–IN4.

Li, K., Min, M. R., and Fu, Y. (2019). Rethinking zero-shot learning: A conditional visual classification perspective. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3583–3592.

Liang, Z., Xu, M., Teng, M., and Niu, L. (2006). Comparison of protein interaction networks reveals species conservation and divergence. *BMC bioinformatics*, 7:1–14.

Lin, S., Wang, C., Zhou, J., Shi, Y., Ruan, C., Tu, Y., Yao, L., Peng, D., and Xue, Y. (2021a). Epsd: a well-annotated data resource of protein phosphorylation sites in eukaryotes. *Briefings in Bioinformatics*, 22(1):298–307.

Lin, Y., Meng, Y., Sun, X., Han, Q., Kuang, K., Li, J., and Wu, F. (2021b). Bertgcn: Transductive text classification by combining gcn and bert. *arXiv preprint arXiv:2105.05727*.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. (2022). Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*.

Linding, R., Jensen, L. J., Pasculescu, A., Olhovsky, M., Colwill, K., Bork, P., Yaffe, M. B., and Pawson, T. (2007). Networkin: a resource for exploring cellular phosphorylation networks. *Nucleic acids research*, 36(suppl_1):D695–D699.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Luo, F., Wang, M., Liu, Y., Zhao, X.-M., and Li, A. (2019). Deepphos: prediction of protein phosphorylation sites with deep learning. *Bioinformatics*, 35(16):2766–2773.

Ma, H., Li, G., and Su, Z. (2020). Ksp: An integrated method for predicting catalyzing kinases of phosphorylation sites in proteins. *BMC genomics*, 21:1–10.

Ma, R., Li, S., Parisi, L., Li, W., Huang, H.-D., and Lee, T.-Y. (2023). Holistic similarity-based prediction of phosphorylation sites for understudied kinases. *Briefings in Bioinformatics*, 24(2):bbac624.

Manning, G., Whyte, D. B., Martinez, R., Hunter, T., and Sudarsanam, S. (2002). The protein kinase complement of the human genome. *Science*, 298(5600):1912–1934.

Manning, G., Whyte, D. B., Martinez, R., Hunter, T., and Sudarsanam, S. (2023). Kinome tables. Accessed on 2023-12-14.

McDonald, M., Trost, B., and Napper, S. (2018). Conservation of kinase-phosphorylation site pairings: Evidence for an evolutionarily dynamic phosphoproteome. *Plos one*, 13(8):e0202036.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Deoras, A., Le, H.-S., Kombrink, S., and Cernocky, J. (2012). Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*, 8(67).

Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., and Steinegger, M. (2022). Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682.

Mitchell, A. L., Almeida, A., Beracochea, M., Boland, M., Burgin, J., Cochrane, G., Crusoe, M. R., Kale, V., Potter, S. C., Richardson, L. J., Sakharova, E., Scheremetjew, M., Korobeynikov, A., Shlemov, A., Kunyavskaya, O., Lapidus, A., and Finn, R. D. (2020). Mgnify: the microbiome analysis resource in 2020. *Nucleic Acids Research*, 48(D1):D570–D578.

Modi, V. and Dunbrack Jr, R. L. (2019). A structurally-validated multiple sequence alignment of 497 human protein kinase domains. *Scientific reports*, 9(1):19790.

Moret, N., Liu, C., Gyori, B. M., Bachman, J. A., Steppi, A., Hug, C., Taujale, R., Huang, L.-C., Berginski, M. E., Gomez, S. M., et al. (2020). A resource for exploring the understudied human kinome for research and therapeutic opportunities. *BioRxiv*, pages 2020–04.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Needham, E. J., Parker, B. L., Burykin, T., James, D. E., and Humphrey, S. J. (2019). Illuminating the dark phosphoproteome. *Sci. Signal.*, 12(565):eaau8645.

Network, C. G. A. et al. (2012). Comprehensive molecular characterization of human colon and rectal cancer. *Nature*, 487(7407):330.

Obenauer, J. C., Cantley, L. C., and Yaffe, M. B. (2003). Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs. *Nucleic acids research*, 31(13):3635–3641.

Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). Zero-shot learning with semantic output codes. *Advances in neural information processing systems*, 22.

Park, M. Y. and Hastie, T. (2008). Penalized logistic regression for detecting gene interactions. *Biostatistics*, 9(1):30–50.

Patrick, R., Lê Cao, K.-A., Kobe, B., and Bodén, M. (2015). Phosphopick: modelling cellular context to map kinase-substrate phosphorylation events. *Bioinformatics*, 31(3):382–389.

Protein Data Bank in Europe (2023). Pdbe. https://www.ebi.ac.uk/pdbe/pdbe-kb/. Accessed: 2023-11-09.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. (2019). Evaluating protein transfer learning with tape.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. (2020). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*.

Röhm, S., Krämer, A., and Knapp, S. (2021). Function, structure and topology of protein kinases.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71(599-607):6.

Safaei, J., Maňuch, J., Gupta, A., Stacho, L., and Pelech, S. (2011). Prediction of 492 human protein kinase substrate specificities. In *Proteome science*, volume 9, pages 1–13. Springer.

Schieven, G. L. (2005). The biology of p38 kinase: a central role in inflammation. *Current topics in medicinal chemistry*, 5(10):921–928.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Shazeer, N. (2020). Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Sheridan, R. P., Nam, K., Maiorov, V. N., McMasters, D. R., and Cornell, W. D. (2009). Qsar models for predicting the similarity in binding profiles for pairs of protein kinases and the variation of models between experimental data sets. *Journal of chemical information and modeling*, 49(8):1974–1985.

Song, J., Shen, C., Yang, Y., Liu, Y., and Song, M. (2018). Transductive unbiased embedding for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1024–1033.

Stark, C., Su, T.-C., Breitkreutz, A., Lourenco, P., Dahabieh, M., Breitkreutz, B.-J., Tyers, M., and Sadowski, I. (2010). Phosphogrid: a database of experimentally verified in vivo protein phosphorylation sites from the budding yeast saccharomyces cerevisiae. *Database*, 2010:bap026.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. (2024). Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. (2023). Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*.

Sumbul, G., Cinbis, R. G., and Aksoy, S. (2017). Fine-grained object recognition and zero-shot learning in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):770–779.

Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. (2007). Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288.

Sweet, R. M. and Eisenberg, D. (1983). Correlation of sequence hydrophobicities measures similarity in three-dimensional protein structure. *Journal of molecular biology*, 171(4):479–488.

Szklarczyk, D., Gable, A. L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N. T., Morris, J. H., Bork, P., et al. (2019). String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 47(D1):D607–D613.

Tomas-Zuber, M., Mary, J.-L., Lamour, F., Bur, D., and Lesslauer, W. (2001). C-terminal elements control location, activation threshold, and p38 docking of ribosomal s6 kinase b (rskb). *Journal of Biological Chemistry*, 276(8):5892–5899.

Tyanova, S., Cox, J., Olsen, J., Mann, M., and Frishman, D. (2013). Phosphorylation variation during the cell cycle scales with structural propensities of proteins. *PLoS computational biology*, 9(1):e1002842.

Ullah, S., Lin, S., Xu, Y., Deng, W., Ma, L., Zhang, Y., Liu, Z., and Xue, Y. (2016). dbpaf: an integrative database of protein phosphorylation in animals and fungi. *Scientific reports*, 6(1):23534.

UniProt Consortium (2023a). https://www.uniprot.org/uniprotkb?query=%28taxonomy_id%3A9606%29+AND+%28ft_domain%3Akinase%29+AND+%28organism_id%3A9606%29&facets=reviewed%3Atrue. [Online; accessed 14-December-2023].

UniProt Consortium (2023b). Uniprot api. https://www.uniprot.org/help/api. Accessed: 2023-12-29.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Gilchrist, C. L., Söding, J., and Steinegger, M. (2022). Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02.

Vapnik, V. N., Vapnik, V., et al. (1998). Statistical learning theory.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vella, V., Giamas, G., and Ditsiou, A. (2022). Diving into the dark kinome: lessons learned from lmtk3. *Cancer Gene Therapy*, 29(8):1077–1079.

Villani, C. et al. (2009). *Optimal transport: old and new*, volume 338. Springer.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Wan, Z., Chen, D., Li, Y., Yan, X., Zhang, J., Yu, Y., and Liao, J. (2019). Transductive zero-shot learning with visual structure constraint. *Advances in neural information processing systems*, 32.

Wang, C., Xu, H., Lin, S., Deng, W., Zhou, J., Zhang, Y., Shi, Y., Peng, D., and Xue, Y. (2020). Gps 5.0: an update on the prediction of kinase-specific phosphorylation sites in proteins. *Genomics, Proteomics and Bioinformatics*, 18(1):72–80.

Wang, W., Zheng, V. W., Yu, H., and Miao, C. (2019). A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–37.

Wang, X., Zhang, Z., Zhang, C., Meng, X., Shi, X., and Qu, P. (2022). Transphos: A deep-learning model for general phosphorylation site prediction based on transformer-encoder architecture. *International Journal of Molecular Sciences*, 23(8):4263.

Wang, Y., Shi, M., Chung, K. A., Zabetian, C. P., Leverenz, J. B., Berg, D., Srulijes, K., Trojanowski, J. Q., Lee, V. M.-Y., Siderowf, A. D., et al. (2012). Phosphorylated $\alpha$-synuclein in parkinson's disease. *Science translational medicine*, 4(121):121ra20–121ra20.

Wiredja, D. D., Koyutürk, M., and Chance, M. R. (2017). The ksea app: a web-based tool for kinase activity inference from quantitative phosphoproteomics. *Bioinformatics*, 33(21):3489–3491.

Wu, T. D. and Brutlag, D. L. (1995). Identification of protein motifs using conserved amino acid properties and partitioning techniques. In *ISMB*, pages 402–410.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. (2018). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265.

Xie, G.-S., Zhang, X.-Y., Yao, Y., Zhang, Z., Zhao, F., and Shao, L. (2021). Vman: A virtual mainstay alignment network for transductive zero-shot learning. *IEEE Transactions on Image Processing*, 30:4316–4329.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. (2020). On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.

Xu, W. and Wang, Y. (2021). Post-translational modifications of serine/threonine and histidine kinases and their roles in signal transductions in synechocystis sp. pcc 6803. *Applied Biochemistry and Biotechnology*, 193(3):687–716.

Xue, Y., Ren, J., Gao, X., Jin, C., Wen, L., and Yao, X. (2008). Gps 2.0, a tool to predict kinase-specific phosphorylation sites in hierarchy. *Molecular & cellular proteomics*, 7(9):1598–1608.

Yang, C.-Y., Chang, C.-H., Yu, Y.-L., Lin, T.-C. E., Lee, S.-A., Yen, C.-C., Yang, J.-M., Lai, J.-M., Hong, Y.-R., Tseng, T.-L., et al. (2008). Phosphopoint: a comprehensive human kinase interactome and phospho-protein database. *Bioinformatics*, 24(16):i14–i20.

Yao, Q., Bollinger, C., Gao, J., Xu, D., and Thelen, J. J. (2012). P3db: an integrated database for plant protein phosphorylation. *Frontiers in plant science*, 3:28672.

Ye, M. and Guo, Y. (2019). Progressive ensemble networks for zero-shot recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11728–11736.

Yılmaz, S., Ayati, M., Schlatzer, D., Çiçek, A. E., Chance, M. R., and Koyutürk, M. (2021). Robust inference of kinase activity using functional networks. *Nature communications*, 12(1):1177.

Yin, C., Zhu, B., Zhang, T., Liu, T., Chen, S., Liu, Y., Li, X., Miao, X., Li, S., Mi, X., et al. (2019). Pharmacological targeting of stk19 inhibits oncogenic nras-driven melanomagenesis. *Cell*, 176(5):1113–1127.

Zhang, L. and Daly, R. J. (2012). Targeting the human kinome for cancer therapy: current perspectives. *Critical Reviews™ in Oncogenesis*, 17(2).

Zhang, Y. and Okumura, M. (2024). Prothyena: A fast and efficient foundation protein language model at single amino acid resolution. *bioRxiv*.

Zhang, Z. and Saligrama, V. (2015). Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*, pages 4166–4174.

Zhao, Y., Bjørbæk, C., Weremowicz, S., Morton, C. C., and Moller, D. E. (1995). Rsk3 encodes a novel pp90 rsk isoform with a unique n-terminal sequence: growth factor-stimulated kinase function and nuclear translocation. *Molecular and cellular biology*, 15(8):4353–4363.

Zhou, Z., Yeung, W., Soleymani, S., Gravel, N., Salcedo, M., Li, S., and Kannan, N. (2024). Using explainable machine learning to uncover the kinase–substrate interaction landscape. *Bioinformatics*, 40(2):btae033.

Zhu, X. J. (2005). Semi-supervised learning literature survey.

Zou, L., Wang, M., Shen, Y., Liao, J., Li, A., and Wang, M. (2013). Pkis: computational identification of protein kinases for experimentally discovered protein phosphorylation sites. *BMC bioinformatics*, 14:1–8.

# APPENDIX A

## Parameters for the DARKIN Split Generation Script

Table A.1 All adjustable parameters that can be modified in the DARKIN dataset creation script are presented in this table.

| **Parameter** | **Description** |
|---|---|
| Random Seed | The random seed that will be set to ensure reproducibility of the same dataset on different runs. |
| Kinase Similarity Percent | The identity similarity score of the kinase domains that will be taken into consideration when splitting the dataset. This similarity percent defines at what similarity level we define the kinases as highly similar (Kinase domains which have similarity equal to or above this will be placed inside the same dataset). |
| Kinase Count Test Threshold | The threshold number of phosphorylation data for a kinase to be able to enter the test dataset. In other words, kinases which have fewer phosphorylation data than this threshold will not be candidate kinases to enter the test dataset. |
| Stratify Percentage for Unseen Test Kinase | The percentage of the dataset to be entered into the test set as unseen kinase-phosphosite data. |
| Kinase Count Validation Threshold | The threshold number of phosphorylation data for a kinase to be able to enter the validation dataset. In other words, kinases which have fewer phosphorylation data than this threshold will not be candidate kinases to enter the validation dataset. |
| Stratify Percentage for Unseen Validation Kinase | The percentage of the dataset to be entered into the validation set as unseen kinase-phosphosite data. |
| Take Sequence Similarity into Consideration | Determines whether to consider kinase domain sequence similarity when splitting datasets. If True, kinases with sequence similarity at or above KINASE_SIMILARITY_RATE are placed in the same dataset. |
| Divide wrt Group | Defines whether to stratify kinases according to kinase groups. If False, the dataset is split without considering kinase group information, possibly leading to imbalanced kinase groups. |