DEEPKINZERO: ZERO-SHOT LEARNING FOR PREDICTING KINASE PHOSPHORYLATION SITES

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER ENGINEERING

By

Iman Deznabi

August 2018

DeepKinZero: Zero-Shot Learning for Predicting Kinase Phosphorylation Sites
By Iman Deznabi
August 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Abdullah Ercüment Çiçek(Advisor)

Öznur Taştan Okan (Co-advisor)

Erman Ayday

Ramazan Gökberk Cinbiş

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

DEEPKINZERO: ZERO-SHOT LEARNING FOR PREDICTING KINASE PHOSPHORYLATION SITES

Iman Deznabi
M.S. in Computer Engineering
Advisor: Asst. Prof. Dr. A. Ercüment Çiçek and Asst. Prof. Dr. Öznur Taştan Okan
August 2018

Protein kinases are a large family of enzymes that catalyze the phosphorylation of other proteins. By acting as molecular switches for protein activity, the phosphorylation events regulate intracellular signal transduction, thereby assuming a central role in a broad range of cellular activities. On the other hand, aberrant kinase function is implicated in many diseases. Understanding the normal and malfunctioning signaling in the cell entails the identification of phosphorylation sites and the characterization of their interactions with kinases. Recent advances in mass spectrometry enable rapid identification of phosphosites at the proteome level. Alternatively, there are many computational models that predict phosphosites in a given input protein sequence. Once a phosphosite is identified, either experimentally or computationally, knowing which kinase would catalyze the phosphorylation on this particular site becomes the next question. Although a subset of available computational methods provides kinase-specific predictions for phosphorylation sites, due to the need for training data in such supervised methods, these tools can provide predictions only for kinases for which a substantial number of the phosphosites are already known. A particular problem that has not received any attention is the prediction of new sites for kinases with few or no a priori known sites. None of the current computational methods which rely on the classical supervised learning settings can predict additional sites for this kinases. We present DeepKinZero, the first zero-shot learning approach, that can predict phosphosites for kinases with no known phosphosite information. DeepKinZero takes a peptide sequence centered at the phosphorylation site and learns the embeddings of these phosphosite sequences via a bi-directional recurrent neural network, whereas kinase embeddings are based on protein sequence vector representations and the taxonomy of kinases based on their functional properties. Through a compatibility function that associates the representations of the site

sequences and the kinases, DeepKinZero transfers knowledge from kinases with many known sites to those kinases with no known sites. Our computational experiments show that DeepKinZero achieves a 30-fold increase in accuracy compared to baseline models. DeepKinZero complements existing approaches by expanding the knowledge of kinases through mapping of the phosphorylation sites pertaining to understudied kinases with no prior information, which are increasingly investigated as novel drug targets.

Keywords: Kinase Substrate Classification, Zero-Shot Learning, Recurrent Neural Networks, RNN, LSTM.

ÖZET

DEEPKINZERO: KINAZ FOSFORILASYON YERLERININ SđFđR-ÖRNEK ÖğRENIM ILE TAHMINI

Iman Deznabi
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Yrd. Doç. Dr. A. Ercüment Çiçek and Yrd. Doç. Dr. Öznur Taştan Okan
Agustos 2018

Protein kinazlar, diğer proteinlerin fosforilasyonunu katalize eden büyük bir enzim ailesidir. Protein aktivitesi için moleküler anahtarlar olarak görev yaparlar ve fosforilasyon olayları vasıtasıyla hücre içi sinyal iletimini düzenlerler. Bu sebeple, bir çok hücresel mekanizmada, merkezi bir rol üstlenirler. Öte yandan, kinaz proteinlerinin fonksiyonel bozukluklarının da birçok hastalıkla ilişkili olduğu belirlenmiştir. Hücredeki normal ve arızalı sinyallerin anlaşılması için, fosforilasyon bölgelerinin tanımlanması ve bu bölgelerin fosforilasyonunda hangi kinazların görev aldığının belirlenmesi gerekir. Kütle spektrometresindeki son gelişmeler, fosforilasyon bölgelerinin proteom seviyesinde hızlı bir şekilde tanımlanmasını olanaklı kılmıştır. Alternatif olarak, protein dizisinde fosforilasyon yerlerini tahmin eden birçok hesaplamalı yöntem de mevcuttur. Bir fosforilasyon bölgesi, deneysel ya da hesaplamalı yöntemlerle belirlendikten sonra, bu bölgeyi hangi kinazın fosforile ettiğini belirlemek ise bir sonraki soruyu oluşturur. Fosforilasyon bölgelerini tahminleyen mevcut hesaplamalı metotların bir kısmı, kinaza-özgü tahminler sunsa da bu yöntemler konvensiyonel gözetimli öğrenme tekniklerine dayandıkları için, ancak bir çok fosforilasyon bölgesi bilinen kinazlar için yeni bölgeleri tahminleyebilirler. Bu zamana kadar üzerine eğinilmemiş bir problem ise daha önce fosforile ettiği hiç bir bölge tespit edilmemiş olan kinazlar için tahmin yapabilmektir. Klasik gözetimli tekniklere dayanan yöntemlerinden hiçbiri, bu tür kinazlar için bağlanma bölgelerini öngöremeyecektir. Bu çalışmada, fosforilasyon bilgisi olmayan kinazlar için sıfır-vuruşlu ögrenme yaklaşımına dayanan DeepKinZero'yu sunuyoruz. DeepKinZero fosforilasyon yerinin merkezde olduğu peptit dizisini girdi olarak alır ve çift yönlü tekrarlayan sinir ağı ile bu dizileri bir vektör uzayına yerleştirir. Kinazları da fonksiyonel özelliklerine ve protein dizilerine göre bir vektör uzayına yerleştirilir. Fosforilasyon bölgesinin çok boyutlu uzayda temsili

ile kinazların temsili arasında tanımlanmış bir uyumluluk fonksiyonu aracılığıyla, DeepKinZero fosforile ettiği bölgelerin bilindiği kinazlardan bu bölgeleri bilinmeyen kinazlara bilgi aktarır. Hesaplamalı deneylerimiz, DeepKinZero'nun taban modellerine göre doğrulukta 30 kata varan artış sağladığını göstermektedir. DeepKinZero'nun önceden fosforilasyon bölgeleri bilinmeyen kinazların fosforilasyon bölgelerini tahminleyerek, önemli ilaç hedefleri olan ve az çalışılmış bu kinazlar hakkındaki mevcut bilgi birikimini artırmasını bekliyoruz.

Anahtar sözcükler: Kinaz Fosforilasyon Yerleri, Sıfır-Örnek Öğrenim, RNN, LSTM.

Acknowledgement

First I would like to express my sincere gratitude to my advisor Prof. Oznur Taştan for her wisdom, guidance, support, and patience. It was a great pleasure to work under her supervision. If I ever become half the person she is I consider it a great achievement. This thesis would not have been possible without her contributions.

I would also want to thank professor Mehmet Koyutürk for supporting and guiding me throughout this project. Furthermore, I would like to thank the jury members, professor Erman Ayday and professor Gokberk Cinbiş for spending the time to read and review my thesis.

Moreover, I want to thank professor Ercüemnt Çiçek for accepting to be my supervisor in last year of my studies.

Additionally, I would like to thank my dear friends, Puria, Mohammad, Noushin, Hamed, Ehsan, Pejhman, Mina, Wiria, Zeinab and all of my other friends for their support and all the great memories. I will never forget the enjoyable time we have had together. Also, I can not forget about my dear officemates, Caner, Ali Burak, Bulent, Gencer and all others for creating a great environment in the office and providing help and support. Of course, I can not finish this part without expressing my gratitude to our department's secretary, Mrs. Ebru Ateş for her kind helps.

I would also like to express my very profound gratitude to my family, my father, mother and brother, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Most importantly, none of these could have happened without the best wife in the world, Nazanin, who offered her support and love throughout these years. This dissertation stands as a testament to your unconditional love and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Protein kinases are a large family of enzymes that catalyze the phosphorylation of other proteins. Phosphorylation is a post-translational mechanism that involves transfer of a phosphoryl group to the side chain of an amino acid residue in the substrate. The amino acid residue that receives the phosphoryl group is usually called the phosphorylation site, or briefly phosphosite. Phosphorylation events can lead to the activation or deactivation of enzymes, modification of the targets' interactions with other proteins, direct them to sub-cellular localization or target them for destruction [4]. Due to their central role in a broad range of cellular activities, aberrant kinase function is implicated in many diseases, particularly in cancer [5]. Therefore, kinases serve as critical targets for therapeutic intervention [6]. Understanding the normal and malfunctioning signaling in the cell entails the identification of phosphorylation sites and the characterization of their interactions with kinases.

Recent advances in mass spectrometry enable rapid identification of phosphosites at the proteome level [7, 8]. Alternatively, there are many computational models developed to predict phosphosites in a given input protein sequence [9–22]. Once a phosphosite is identified, either experimentally or computationally, knowing which kinase(s) act(s) on this particular site becomes the next question. Some of the aforementioned computational methods [9–16, 19–22] provide kinase-specific

predictions for phosphorylation sites. These predictive models are usually based on machine learning methods that use training data (established kinase-phosphosite associations) to model the relationship between the properties of kinases and the properties of their target phosphosites. However, since conventional supervised machine learning methods require "positively labeled" samples, these tools can provide predictions only for kinases for which a substantial number of the binding sites are already known. For example, MusiteDeep [16], reports a prediction tool that uses deep learning to predict binding sites for kinases, and it exclusively focuses on kinase families CDK, PKA, CK2, MAPK, and PKC because these families have at least 100 experimentally verified phosphosites. Another recent tool, PhosphoPredict [14], provides predictions for 12 human kinase families (ATM, CDKs, GSK-3, MAPKs, PKA, PKB, PKC and SRC) that have at least 50 known phosphorylation sites. The current predictor tools in the literature and the kinases or kinase families studied therein are outlined in Table 1.1.

A particular problem that has been overlooked in the literature is the prediction of new sites for kinases with few or no known phosphosites. In humans, for example, more t00han 500 proteins have been annotated as kinases [2]. PhosphositePlus, a database that provides experimentally validated phosphosites, provides phosphosite annotations for 364 human kinases. The distribution of the number of phosphosites for each kinase is shown in Figure 1.1. As seen in the figure, for nearly 200 annotated kinases among a set of 364, there are at most 10 experimentally validated phosphosites. Since the number of available "positively labeled" samples is very low for most kinases, existing tools are insufficient to make reliable predictions for these kinases. Furthermore, in a possible scenario where a researcher would want to predict kinases for an identified phosphosite, well-studied kinases are likely to be favored by the existing prediction algorithms, since these algorithms see more examples of kinases with many known targets. This effect has the risk of biasing the knowledge of the human kinome toward well-studied and/or promiscuous kinases, making it difficult to comprehensively annotate the entire human kinome. For example, FAM20C is a kinase whose diminished activity causes Raine syndrome [24]. For this kinase, there are only three experimentally validated target sites. Unfortunately, existing computational methods that rely on classical supervised learning setting can not predict any

Method	Kinase Families or Individual Kinases	Criteria for inclusion	Fraction of phosphosites covered (%)
MusiteDeep [16]	Families: CDK, PKA, CK2, MAPK, PKC	Families associated with more than 100 sites	42.7
PhosphoPredict [14]	Families and individual kinases: ATM, CDKs, GSK-3, MAPKs, PKA, PKB, PKC, SRC	Families associated with at least 50 sites	51.6
KSRPred [13]	103 human kinases	Kinases associated with least 15 sites	85.9
PhosphoPICK [20]	59 human kinases	Kinases associated with more than 10 sites	64.0
Li et al. [23]	Families: ATM, CDK, CK2, GSK-3, MAPK, PKA, PKB, PKC	Families with at least 50 sites	49.9

Table 1.1: **The coverage of the phospho-proteome and kinome provided by the existing methods for predicting phosphorylation sites targeted by kinases**. For each recently published prediction method, the criteria for inclusion, is provided. Finally, the last column list what fraction of the sites that are known to be associated with a kinase is covered.Note that the sites considered here refer to the phosphosites that are listed in the Phosphosite database; the coverage of the phosphosites identified in a typical mass-spectrometry based phosphorylation screening is usually much lower than the figures reported here.

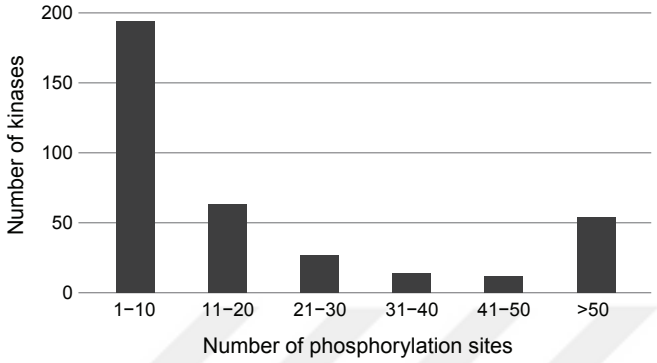

Figure 1.1: **The distribution of the number of experimentally validated target phosphosites for kinases in the human kinome.** The histogram is based on data obtained from Phosphosite database, which reports experimentally validated kinases for 364 human kinases.

additional sites for this kinase.

In this study, we introduce DeepKinZero, a zero-shot learning based approach for the prediction of the association between kinases and phosphosites. With this framework, we specifically aim at providing predictions for kinases with no or few known targets. Zero-shot learning is a machine learning paradigm that has received significant attention in recent years, as it deals with the question of how to recognize instances of classes with no training examples [25]. It has been shown to be successful in various machine learning applications, particularly those that involve computer vision [25–28]. Since there are no or few training instances for most kinases, we argue that zero-short learning could be the ideal approach to the problem of predicting kinase-phosphosite associations.

Given a potential phosphorylation site, using the local protein sequence centered at this site, DeepKinZero predicts the subset of kinases that can phosphorylate this particular site. DeepKinZero learns the site sequence representations via a bi-directional recurrent neural network, wherein the kinase representations are learned using kinase protein sequence and the taxonomy of kinases are obtained from other sources. Through a compatibility function that associates the representations of the site sequences and the kinases, DeepKinZero transfers knowledge from

kinases with many known sites to those kinases with no known sites. DeepKinZero outperforms baseline methods with a 30% margin. The important positions and the amino acids that are highlighted by the model agrees well with the current knowledge of kinases and their associated functions.

DeepKinZero offers a scalable and flexible approach for predicting sites for kinases with no prior sites. It is implemented in Python and is provided as an open source tool at `https://github.com/Tastanlab/DeepKinZero`.

The rest of the Thesis is organized as follows. In the next section, we provide background information about the DeepKinZero algorithm and the kinase proteins. In chapter 3, we describe our proposed zero-shot learning model in detail. In chapter 4, we discuss our experimental setup and results achieved. Finally, in chapter 5, we conclude the paper and discuss potential future works.

Chapter 2

Background

In this chapter, the background information is provided.

2.1 Protein Kinases

Protein kinases are a large family of enzymes that catalyze the phosphorylation of other proteins. Phosphorylation is a critical post-translational mechanism that involves transferring a phosphate group to the hydroxy group of a residue in the substrate protein. Phosphorylation is a reversible process and transiently alter protein properties. Phoshorylation can cause conformational changes and alter the acitivity states of the enzymes, modulate the targets' interactions with other proteins, direct their subcellular locazation, or target them for destruction. Deregulated protein kinase activity is associated with a variety diseases, most notably with cancer, which makes them the leading drug targets for cancer treatment [5, 6].

The protein kinases constitute one of the largest and most functionally diverse gene families. Kinases are present in a variety of species from bacteria to plants and humans. In humans there are more than 500 kinases [2]. Based on the types of amino acid residues they phosphorylate, eukaryotic protein kinases are divided

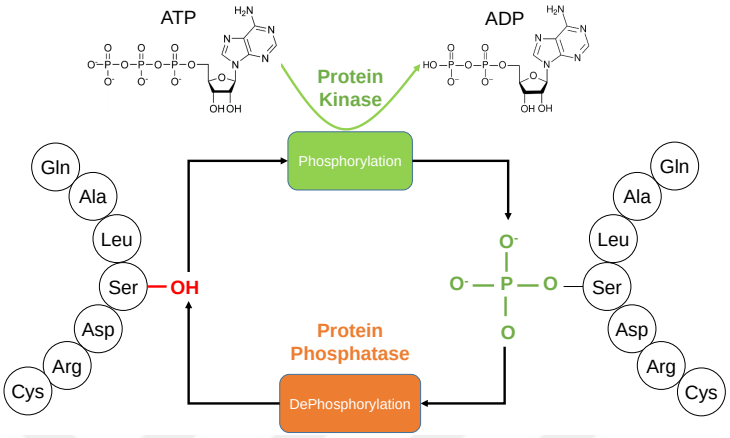

Figure 2.1: **Phosphorylation is a reversible process which involves adding a phosphate group from a nucleoside triphosphate to an amino acid**. The amino acids, Serine, Threonin and Tyrosine are the most common phostoacceptor amino acids since their side chains contain OH.

into as serine/threonine kinases and tyrosine kinases. Some protein kinases can phosphorylate both serine/threonine, as well as tyrosine residues. This group of kinases has been known as dual specificity kinases. [1] further categorizes the kinases into 43 sub-categories according to their specificity, location and function, a representation of this categorization is given in Figure 2.2. Another well-known categorization of kinases has 4 levels: groups, family, subfamily and individual kinases. This categorization is done according to a combination of sequence similarity in the kinase domain, as well as additional information from domains outside of the catalytic domain, evolutionary conservation, and known functions [2]. The corresponding representation of this hierarchy is given in Figure 2.3. In this categorization there are 10 main groups (AGC, CMGC, CAMK, CK1, Other, STE, Tyrosine Kinase(TK), Tyrosine Kinase-Like (TKL), RGC, PKL and Atypical), which further separate into 103 families.

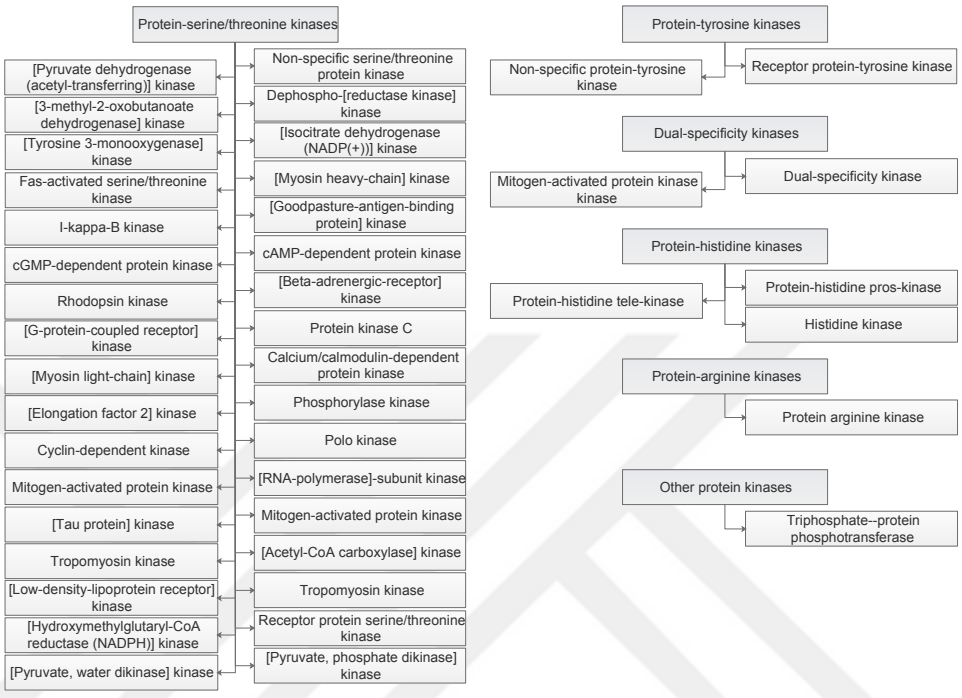

Figure 2.2: Classification of kinases according to the ENZYME database [1].

2.2 Deep Learning

In the classification the primary goal is to learn a function, f, that will map the input space, $\mathcal{X}$ to the output space $\mathcal{Y}$.The information pertaining to the input and the means of transferring this input to the system is critical for the learning process. The classical machine learning approaches heavily depend on these input representations. These classical approaches learn how and which of the input features are correlated with the outcome variable of interest; however, finding the right set of features is a critical problem. Deep learning solves this critical problem by building complex representations from simpler representations during the learning process. Thus, not only the model learns the mapping from the input representations to the output space, but also the representation itself. Based on the problem and the input type, deep learning models offer a variety of different architectures to learn the representations. In our work, to learn the representations of input site sequences, we employ a bi-directional recurrent neural

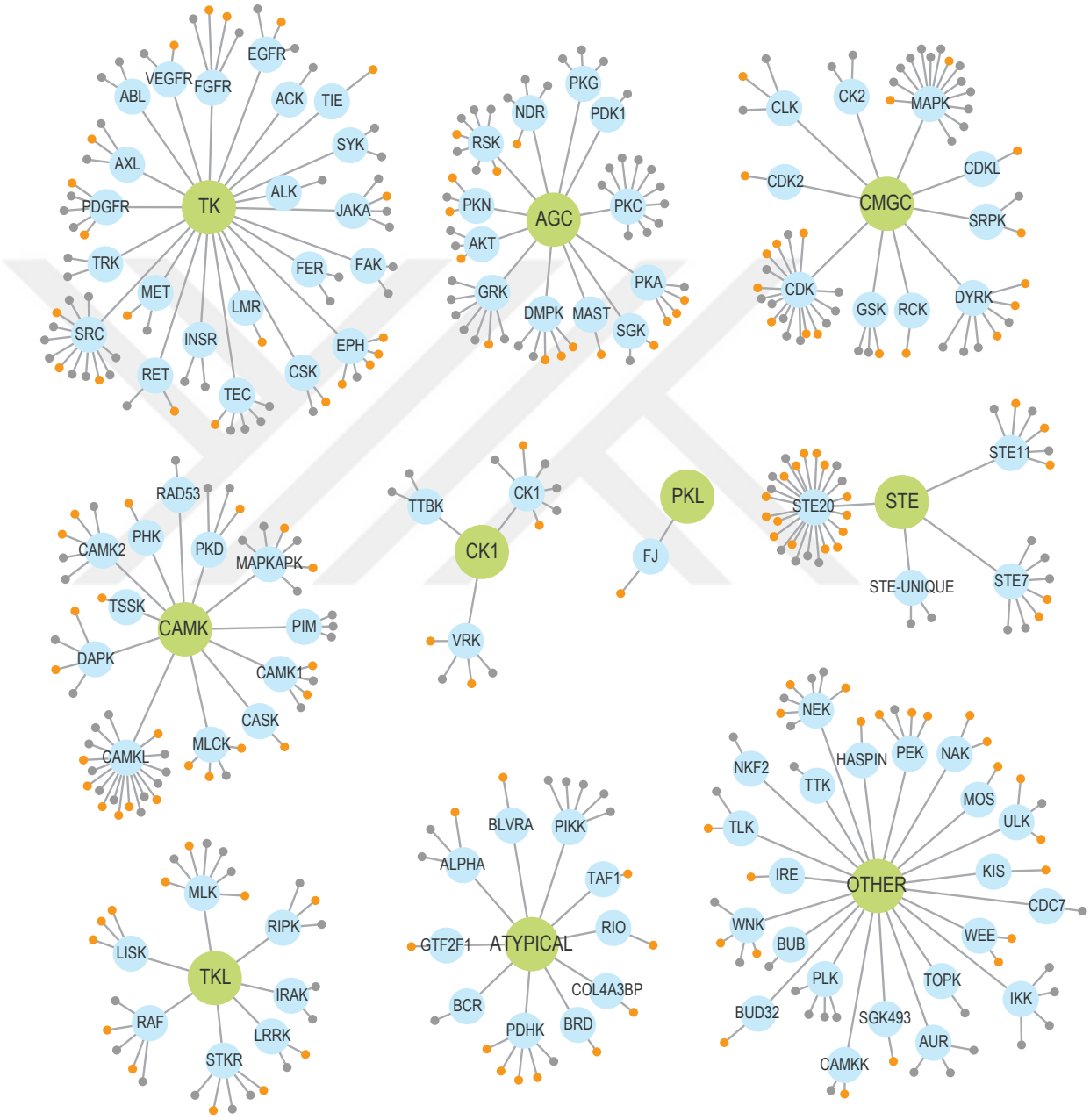

Figure 2.3: **The partitioning of kinases into families and groups as proposed in [2]**. Each network is centered on a group node shown in green. The families (blue nodes) that are listed under a group are linked with edges to the group node. The small gray nodes show kinases with many known sites; these kinases are used in the training of the models whereas the small orange nodes indicate the kinases that are unseen and used for testing.

networks. Below we provide background information on the more general class of models: recurrent neural networks and the bi-RNNs.

2.2.1 Bidirectional Recurrent Neural Networks

Recurrent neural networks(RNNs) is a class of neural networks [29] that shows state-of-the-art performances for modeling and the prediction tasks for sequential data. At each timestep, which refers to the position in the sequence, RNN accepts an input vector and updates its hidden state via non-linear activation functions to make a prediction of the target output. RNN's hidden state can store information in high-dimensional distributed representations.

RNNs include nodes that have memories which enable them to keep the past information. Unlike feed forward artificial neural networks, RNNs can use their internal memory to process input sequences. During the training, the RNN networks are unrolled (Figure 2.4) to become a strictly feed forward artificial neural network. Once they are unrolled, backpropagation can be applied through time (see Section 2.4.2). However, the resulting feed-forward neural network model will have as many layers as the length of the input sequence. Especially for long sequences, this sets a challenge for training RNNs. The performance of the gradient descent degrades due to the 'vanishing gradients' phenomenon [30], where the gradients exhibit exponential decay as they are back-propagated through time [30,31]. When the long-term gradients decay exponentially, as the total gradient is the sum of long-term and short-term influences, the long-term signals are lost and the short-term signals alone govern the gradient. Using different archiectures, different RNN models aim to handle the vanishing gradients problem differently. The common architechtures employed are Gated Recurrent Units(GRU) [32], Long-Short Term Memory (LSTM) [33] and fully recurrent [34]. In DeepKinZero, we employ LSTM structure, since its has been succesfully applied to different tasks [35–40].

Long-Short Term Memory (LSTM) [33] units can be used as building blocks for constructing RNN layers. A common LSTM unit comprises a cell, an input gate, an output gate and a forget gate. The forget gate is responsible for the scope of

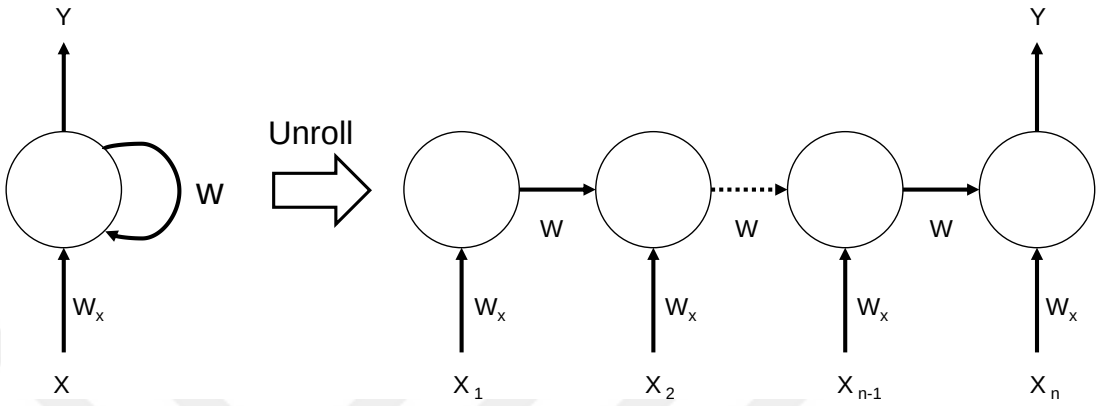

Figure 2.4: Unrolling process of RNNs to feed forward neural netwokrs

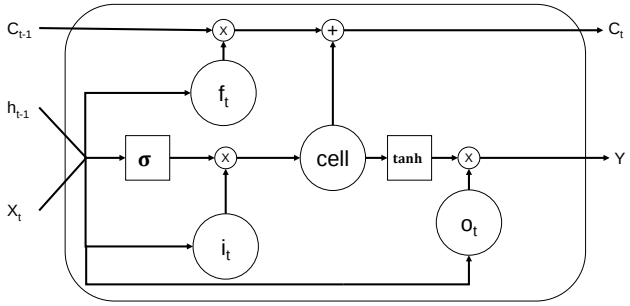

Figure 2.5: **A simple LSTM unit**, C_{t-1} is the memory from the previous cell, h_{t-1} is the output of the previous cell, X_t is the input vector, C_t is the current cell memory and h_t is the current cell output

data that goes to memory, input gate learns the amount of data to flow into the cell, and the output gate tries to learn the extent of the data to be included in the output of the gate. There are many different variations of LSTM units, we employ an implementation based on [41]. A schematic of this implementation is provided in Figure Figure 2.5. In this implementation, the input is first multiplied by the input gate and then is fed to a cell which is controlled by a forget gate. The output of this cell then is multiplied by the output gate.

The standard RNN architecture uses contextual information in one direction,

typically pertaining to past, which is appropriate for time-series prediction. However for sequence tasks, the surrounding context on both sides usually bring more information. Bidirectional RNNs scan the input data in both directions, backward and forwards, with two seperate recurrent layers. This provides access to all the surrounding context. Since the context for a phosphosite is fundamentally important we use bi-directional neural network.

2.3 Zero-Shot Learning

We briefly introduce zero-shot learning(ZSL) and related work here, and we will further explore ZSL and our implementation of it in Section 3.2. Zero-shot learning (ZSL) or zero-data learning is the process of classifying data into categories which we do not have any training example of them. ZSL is specially applicable in places where the data gathering for some classes are too expensive or time consuming. In ZSL our aim is to find a connection between the instances of a class and the high-level description of it. In this case if we have this connection we can find new instances of that classes merely by knowing its high-level description.

2.4 Stochastic Gradient Descent

Most of the machine learning problems involve the minimization of a cost function based on some weight parameters ($Q(w)$) where the goal is to find the weight parameter w which minimizes this function. The loss function associated for a single training pair is the loss function that measures the deviation of the predictions $\hat{y}$ from the target outputs y. The traning loss for the whole training set is the average or sum of the individual losses associated with each training example. A broad family of gradient descent methods are available for minimizing a function by repeatedly updating weights using this step:

$$w = w - \eta \nabla Q(w) \tag{2.1}$$

Where η is the step size or learning rate. Gradient descents can find local optimum of a function by taking small steps in the direction of its curve. In Equation (2.1) as $Q(w)$ is usually the sum or average of the loss over all data points ($Q(w) = \frac{\sum_{i=1}^{n} Q_i(w)}{n}$ where n is the number of data points and $Q_i(w)$ is the loss function for data point i) for finding $\nabla Q(w)$ from the data we can outline the derivation of $Q_i w$ for each data point i:

$$\nabla Q(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla Q_i(w) \tag{2.2}$$

Since the summation in Equation (2.2) can be very computationally prohibitively expensive, another extreme approach would be to calculate the derivation based on each data point and update the weights each time:

$$w = w - \eta \nabla Q_i(w) \tag{2.3}$$

Between computing the true gradient and efficiency, a meaningful compromise is done by Stochastic Gradient Descent(SGD) which tries to sample the derivations of $Q_i(w)$ for a subset of datasets and update the weights accordingly.

2.4.1 Adam Optimizer

Adam (short for adaptive moment estimation) [42] is an extension to the stochastic gradient descent, which tries to calculate adaptive learning rates per parameter by considering the running averages of both gradients and the second moments of the gradients. In Adam, the decaying averages of past and past squared gradients are calculated as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{2.4}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{2.5}$$

where m_t and v_t are the estimates of the first moment and the second moment of the gradients, respectively. β_1 and β_2 are the parameters of the optimization algorithm and g_t and g_t^2 are the first and second gradients at batch t. Furthermore,

to avoid biasing m_t and v_t towards zero, the following bias corrections to m_t and v_t are proposed:

$$\hat{m_t} = \frac{m_t}{1 - \beta_1^t} \tag{2.6}$$

$$\hat{v_t} = \frac{v_t}{1 - \beta_2^t} \tag{2.7}$$

Accordingly, Adam procedure updates the parameters as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v_t} + \epsilon}} \hat{m_t} \tag{2.8}$$

where η is the learning rate.

2.4.2 Backpropagation through time

As it can be seen in Section 2.4 training the neural networks consist of calculating the derivation of cost function $Q(w)$ and updating the weights accordingly. Since neural networks usually have several layers and the output of the model is in the last layer, we can not calculate the loss function for each layer in between since we do not know what its output should be. To solve this problem a technique called Backpropagation is used in which the loss is calculated in the last layer and then propagates backwards to nodes of each layer. Similarly in RNNs which are used for classification, the output at the last time step is known so the loss function is only known in last time step. In this case a technique called Backpropagation through time(BPTT) is used to propagate the derivation of loss function in the last time step to update the weights of nodes in each time step. Conceptually, BPTT works by unrolling the RNN (Figure 2.4), calculating the error and propagating it back to the network.

Chapter 3

Proposed Solution

In this section, we first explain the zero-shot learning formulation of the kinase prediction problem. Next, we explain the general framework, the experimental set up and the data sources that are used.

3.1 Problem setup

The local sequence surrounding the phosphorylation site is considered to encode the most relavant information for the kinase binding. For each phosphosite, we use the peptide of 15 residues centered at the phosphorylation site. We will refer to this peptide sequence as the phosphosite sequence. The prediction task is stated as follows: given the phosphosite sequence, $x \in \mathcal{X}$, predict the kinase that mediates the phosyphorylation of this site, $y \in \mathcal{Y}$. Here, $\mathcal{X}$ represents the space of phosphosite sequences and $\mathcal{Y}$ is the set of all kinases. This is a multi-class classification problem with many classes as there are many kinases. We assume that the kinases, y, are related, but for some of the kinases, there is no identified known phosphosites. We will refer the kinases with known phophosites in the training as *common* kinases, $Y_{tr} \subset \mathcal{Y}$ and those with no phoshosites information as *rare* kinases, $\mathcal{Y}_{te} \subset \mathcal{Y}$, these two class sets are disjoint $Y_{tr} \cap Y_{te} = \emptyset$. We assume that we have the identities of the common and rare kinases. During the training

we are given a dataset of kinase-phosphosite pairs, $D = \{(x_i, y_i), i = \{1 \dots N\}\}$, where $y_i \in \mathcal{Y}_{tr}$. Since during the training phase there are no positively labeled data for the rare kinases, $(y \in \mathcal{Y}_{te})$, it is not possible to directly use traditional supervised methods that can recognize these kinases. Zero-shot learning addresses this problem. In the next section, we detail this approach within the context of related work and describe how DeepKinZero adapts this framework.

3.2 Zero-Shot Learning Model

The key to making predictions for classes with no training data is to transfer knowledge obtained from seen classes, $\mathcal{Y}_{tr}$ in the training to those classes that are not seen, $\mathcal{Y}_{te}$. This learning transfer is rendered possible by knowing, for each unseen class, the relationship with the seen classes. As exemplified by Yu et al. [43], for an image classification system it is difficult to recognize an okabi when there is no images of okabi in the training set. Yet, if the visual descriptions such as – zebra-stripes, four legs, brown torso, a deer-like face – can be learned from the training classes and if the system has side information that okabis bear these attributes, it becomes possible for the algorithm to recognize an okabi with this description even without any prior exposure. Similarly, even if we do not observe any phosphosites that is associated with a rare kinase in training, the zero-shot learning framework enables us to recognize a site of this kinase by resorting to high-level descriptions of the kinases derived from its functional and sequence characteristics.

In the attribute based zero-shot learning approaches [44], each class is represented as a vector whose entries mark the presence or absence of the attributes in that class. These attributes form a multi-dimensional space that is shared between the seen and unseen classes. To classify a class with no training instances, the attributes are predicted by the classifier and by matching attribute vector with the unseen class vector. The key idea here is to use relevant side information in representing such classes so that the semantic relationship between classes is established. Early work of zero-shot learning makes use of this attribute-based

zero-shot learning [44], the recent approaches have shown that label embedding (or class embedding) framework [45] is a more successful approach.

Following this formulation, we assume that each kinase, $y \in Y$, can be represented in a multi-dimensional vector space learned from functional and sequence information on kinases. Kinase embedding function, $\phi(y)$, takes the kinase and maps it to a kinase embedding vector (class vectors). If these kinase vectors are learned correctly, "similar" kinases are close according to the Euclidean metric in the embedded space. Following the recent work of [27, 45–49], we use a bi-linear compatibility function, function $F : \mathcal{X} \times \mathcal{Y} \rightarrow R$. F takes a phosphosite - kinase pair, (x, y), returns a scalar value that is in proportion to the confidence of associating the site, x, with kinase y:

$$F(x, y; W) = \theta(x)^{\intercal} W \phi(y) \tag{3.1}$$

In the above equation, $\theta(x)$ is d-dimensional vector representation of the 15 residue long peptide sequence centered on the phoshosite. We will refer to $\theta(x)$ as site sequence embedding vector. $\phi(y)$ is the m-dimensional kinase embedding vector. W is a $d \times m$ matrix, that is to be learned in the training models. We also followed the model presented in [28] and added embedding-specific linear terms to our model. This can be handled by simply adding constant dimensions to both the input embedding and the class embedding vector as follows:

$$\theta_e(x) = [\theta(x)^{\top} 1]^{\top} \tag{3.2}$$

$$\phi_e(y) = [\phi(y)^{\top} 1]^{\top} \tag{3.3}$$

where $\theta_e(x)$ and $\phi_e(y)$ are the extended vectors of data embedding and class embedding and similarly we add bias (b) and linear weights (w_x, w_y) to W to get W_e which is the extended weight matrix:

$$W_e = \begin{bmatrix} W & w_x \\ w_y & b \end{bmatrix} \tag{3.4}$$

The new compatibility function is given below:

$$F_e(x, y) = \theta(x)^{\top} W \phi(y) + w_x^{\top} \theta(x) + w_y^{\top} \phi(y) + b \tag{3.5}$$

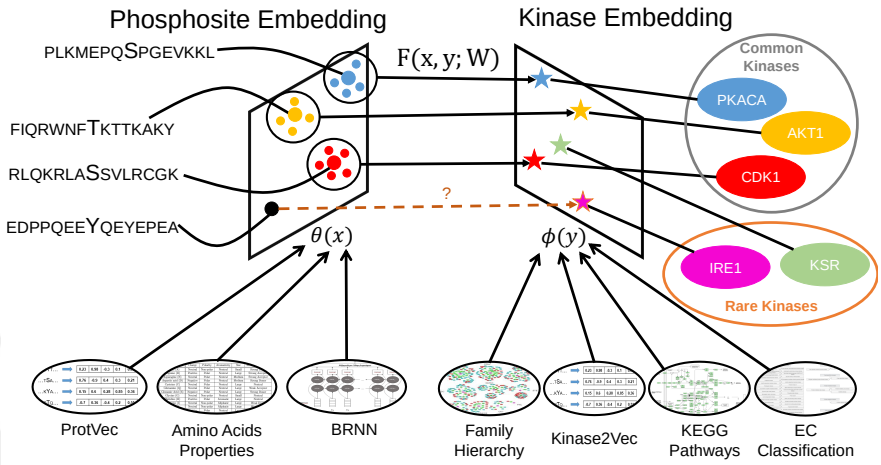

Figure 3.1: **Overview of the zero-shot learning approach**. Both the phosphorylation sites and the kinases are represented with multi-dimensional vector spaces. The phosphorylation site representations, $\theta(x)$ are based on protein sequence embeddings from a deep learning model, amino acid physiochemical properties, and Bidirectional Recurrent Neural Network (BRNN) trained on phosphorylation sites. The class representations, $\phi(x)$ are based on three level classification (superfamily, family and group information) of kinases, ProtVec representation of kinase sequences, participation of kinases in the same pathways, and Enzyme Commission classification of kinases. The function $F(x, y; W)$ is learned such that the compatibility between site embedding $\theta(x)$ and class embeddings $\phi(y)$ is maximized. $F(x, y; W)$ is then used to recognize instances of unseen classes by leveraging the classes embedding vectors.

which is equivalent to:

$$F_e(x, y) = \theta_e(x)^\top W_e \phi_e(y) \tag{3.6}$$

Once the W_e is learned, zero-shot model classifies the input phosphosite to the kinase y by:

$$y^* = \underset{y}{\operatorname{argmax}}\, F(x, y; W_e) \tag{3.7}$$

3.3 Phosphosite Sequence Embeddings

We experimented with different phosphosite sequence embeddings, ($\theta(x)$ in Equation (3.1)). The results obtained with each of these embeddings are discussed in the Section 4.2, here, we will elaborate on how each of these representations is obtained

3.3.1 Sequence as a one-hot encoded vector

One of the obvious strategies is to represent peptide sequences as one-hot vectors. In this representation, each residue of a peptide sequence is coded with a 21-dimensional vector with binary entries. An entry 1 is placed on the index corresponding to a given amino acid type and the remaining entries are all zeroes. 20 of these positions code for each of the amino acids and one extra entry is to code positions where the left part (or right part) for the phosphorylation site corresponds to N-terminal (or the C-terminal) of the protein, thus the sequence is shorter than 15. In this way, each phosphosite sequence is represented with a 15×21-length binary vector.

3.3.2 ProtVec

We employ unsupervised embedding models trained over protein sequences provided by ProtVec method [50]. Word embedding techniques, such as Word2Vec, are useful tools in natural language processing that maps words to continuous vector representations [51]. Inspired by this methods, Asgari and Mofrad proposed ProtVec representations for protein sequences [50]. ProtVec provides a continuous distributed representation for biological sequences. We used the pre-trained model of ProtVec which is trained on Uniprot-SwissProt [52] dataset. ProtVec converts each 3-mer in input sequence into a continuous vector of length 100. There are 13 3-mers in a peptide of 15 residues, thus, our ProtVec representations of each sequence are 13×100.

AA	Charge	Polarity	Aromaticity	Size	Electronic Property
A	Neutral	Non-polar	Neutral	Small	Strong Donor
R	Positive	Polar	Neutral	Large	Strong Acceptor
N	Neutral	Polar	Neutral	Medium	Strong Acceptor
D	Negative	Polar	Neutral	Medium	Strong Donor
C	Neutral	Polar	Neutral	Large	Neutral
Q	Neutral	Polar	Neutral	Large	Weak Acceptor
E	Negative	Polar	Neutral	Large	Strong Donor
G	Neutral	Non-polar	Neutral	Small	Neutral
H	Positive	Polar	Aromatic	Large	Neutral
I	Neutral	Non-polar	Aliphatic	Large	Weak Donor
L	Neutral	Non-polar	Aliphatic	Large	Weak Donor
K	Positive	Polar	Neutral	Large	Strong Acceptor
M	Neutral	Non-polar	Neutral	Large	Weak Acceptor
F	Neutral	Non-polar	Aromatic	Large	Weak Acceptor
P	Neutral	Non-polar	Neutral	Small	Strong Donor
S	Neutral	Polar	Neutral	Small	Neutral
T	Neutral	Polar	Neutral	Medium	Weak Acceptor
W	Neutral	Non-polar	Aromatic	Large	Neutral
Y	Neutral	Polar	Aromatic	Large	Weak Acceptor
V	Neutral	Non-polar	Aliphatic	Large	Weak Donor

Table 3.1: Classification of amino acids(AA) based on five different physiochemical amino acid properties as in [3].

3.3.3 Physical and Chemical Characteristics of Amino Acids

Next, we use a reduced alphabet for amino acids based on their physiochemical properties. In other words, instead of its amino acid type, each sequence is represented by the residue properties. The properties we consider are charge, polarity, aromaticity, size, and electronic-property of the residue. The categorization of each amino acid into groups based on this five properties are given in Table 3.1. To this end, we coded each sequence based on a property-based one-hot encoded vector and concatenate all the property based vectors. We use this property based vectors as input to the BRNN model (see Section 3.3.4) or concatenate them together and used directly as site sequence embedding.

3.3.4 Recurrent Neural Networks

To learn a deep representation of phosphosite sequences, we train a Bidirectional Recurrent Neural Network (BRNN) [53] model over the phosphorylation sites known to bind to the common kinases. As mentioned in Section 2.2.1 RNN's hidden state can store information as high-dimensional distributed representations. We extract the $l2$ normalized output of the last layer of BRNN as the site sequence embedding($\theta(x)$). First, we use an embedding layer for learning a deep representation of the input sequence vector when the input is not an embedding itself. In this case, the embedding layer is followed by a BRNN layer. In cases, where the input is already an embedded representation it is directly input into one layer of the BRNN. BRNN contains 500 LSTM cells [33] on each direction followed by a fully connected layer for classification (Figure 3.2). This number of cells is chosen after proving to be the best compromise between memory usage and performance accuracy on validation and training data.

In training, we minimize the cross entropy loss function using the Adam optimizer [42]. We employ drop-out regularization with a 0.5 keep probability [54], apply batch normalization [55] after embedding layer and layer normalization in LSTM cells [56].

3.4 Kinase Embedding

We use a combination of three different class embedding methods: (i) protein sequence embeddings generated using unsupervised models, and, (ii) kinase functional taxonomies derived from different sources iii) pathway participation of kinases. Finally, vectors obtained from different sources are concatenated to from the final vector, $\phi(y)$ in Equation (3.1). We experiment the utility of some of the vectors here through computational experiments and drop some of them in the final model, yet the Figure 3.3 summarizes the size of the final kinase vectors when all the sources are used. Below, we give a detailed account of such sources and the way they are deployed to arrive at these kinase embeddings.

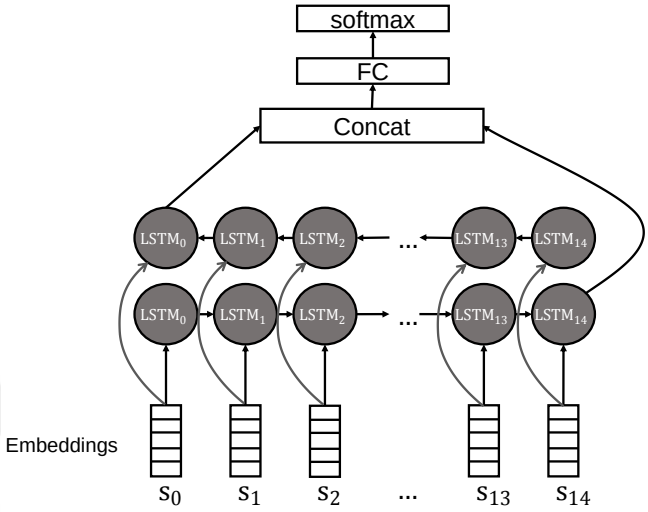

Figure 3.2: **The structure of our BRNN model**, the representation of the phosphorylation site is feed into a BRNN layer which has 500 LSTM cells on each direction, and after training, the l2-normalization of the output of this layer is used as the representation of the phosphorylation site.

3.4.1 Kinase Taxonomies

There are many different classifications for Kinases. Here we use the classification proposed by [2] which is a hybrid classification according to sequence similarity in the kinase domain that uses additional information from domains outside of the catalytic domain, evolutionary conservation, and from known functions. One can see a representation of this classification in Figure 2.3. In this classification there are 10 groups and 103 families. We convert this to a binary vector by representing families, groups and individuals as one-hot represented vectors. In this formulation if a kinase belongs to a family the representing bit in the vector shall be 1. By also adding a binary one-hotted vector representing each individual kinase (we have 364 kinases), in the end, we attain a binary vector with a size of 477.

3.4.2 EC Classification of Kinases

As an alternative source of kinase categorization, we use the enzyme classification provided by the ENZYME database [57]. According to this classification scheme, kinases are grouped into 6 main categories based on their functions. The two largest categories of kinases are the tyrosine-specific protein kinases and serine/threonine kinases. The main categories are further divided into subcategories (as shown in Figure 2.2). To capture this information, for each kinase, we encode a binary vector with a size of 43, wherein 43 represents the total number of categories and subcategories.

3.4.3 Kinase2Vec

As kinases can be related through their kinase domain sequences, we use a ProtVec representation of kinase domain sequences just as we do for the input phosphosite sequence (see Section 3.3.2). Since ProtVec vector legths are defined by the number of 3-mers in the sequence and since for each kinase, the kinase domains can be of different lengths, we average the ProtVec vectors generated for each 3-mer amino acids into one vector whose size is 100.

3.4.4 KEGG Pathway

To capture the relatedness of kinases in the biological functional space, we create kinase vectors based on the pathways in which the kinases participate. Cumulatively, there are a 190 KEGG pathways in which at least one of the kinases participate. Each kinase vector is coded as a 190 element binary vector based on its participation in each of the cellular pathways.

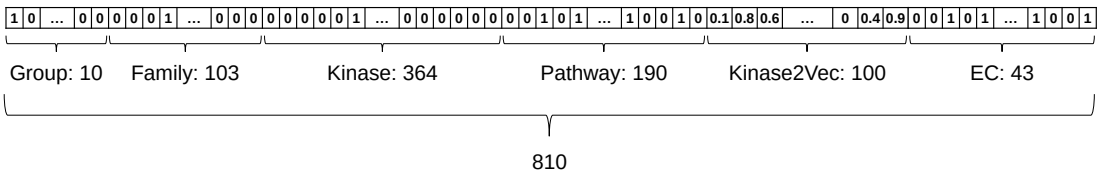

Figure 3.3: **The one-hot encoding representation of the class embeddings**. Vectors from different sources are concatenated to form the class embedding vector. The numbers in the parenthesis state the size of the vectors culled from each data source. The number 810 is the total size of the class embedding vector.

3.5 Data Sets

The dataset of the experimentally identified kinases- phosyphorylation sites pairs, the phosphorylation site sequences and family information on kinases are obtained from PhosphositePlus [58] (downloaded in March 2018). The dataset include information on the experimentally identified 13,426 phosyphorylation sites and their cognate human kinases. There are 364 kinases used in our experiments. Kinase group and family membership is obtained from [2] (downloaded in April 2018). The pathways information is obtained from KEGG database [59–61] (downloaded on April 2018). Enzyme Comission classifications of the kinases are from ENZYME database [1](downloaded in June 2018).

Chapter 4

Results

In this section, we present the experimental tests of our approach, present relevant findings and discuss the results. First, we explain our experimental setup which we use throughout our experiments, then we compare different methods targeted towards the supervised classification of frequent kinases and finally we present zero-shot learning results.

4.1 Supervised classification

Before presenting our zero-shot learning results, we present the results on supervised set to demonstrate the level of difficulty of the multi-class classification problem. We compare our BRNN model with classical classification techniques. We use support vector machine, logistic regression and k-NN. To this end, we report micro- and macro-averaged accuracies, which are calculated for a 3-fold cross validation over the training set (train set in Figure 4.1). We report the micro and macro-accuracies for each model in Table 4.1. 5% of the common kinases dataset (the training set in ZSL model) is set aside for validation set for the supervised setting. We perform 3-fold cross-validation to train the models. Each model is trained on the supervised training-set. For logistic regression, l1-regularization is applied , for SVM, a linear kernel is used and the selected k for

Model	Micro-Accuracy	Macro-Accuracy
Random guess	0.45	0.45
Most frequent class	1.92	0.67
Logistic regression (l1)	22.69	6.35
SVM	21.80	2.21
KNN	20.84	2.95
BRNN	**23.73**	**8.97**

Table 4.1: **The micro- and macro-average accuracies of common classes with different classifiers**. The one-hot representation is used for representing phosphorylation site sequences as vectors.

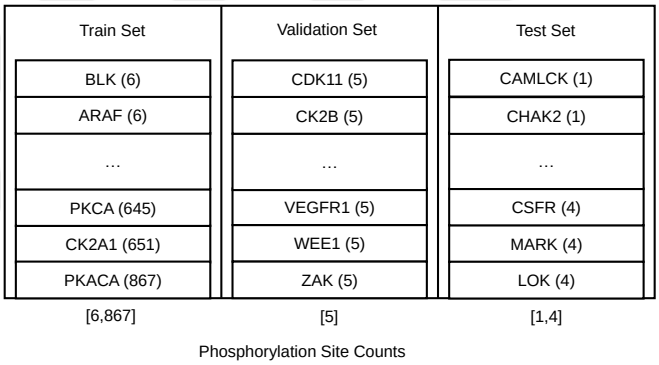

Figure 4.1: **For the zero-shot learning model, the kinases are partitioned into train/validation and test settings based on the number of sites that they target (shown in the parenthesis)**. Kinases with 1 to 4 phosphorylation sites are used as test instances, those with 5 for validation set and kinases with equal or more than 6 phosphorylation sites are used for training the models.

k-NN is 70. These hyper-parameters are tuned on the supervised set validation data. A major hurdle in this task is the presence of 218 classes, only a fraction of which possess enough positively labeled instances. The results demonstrate the inherent challenges of multi-class classification.

4.2 Zero-Shot Learning Results

We train and evaluate our models on the experimentally validated kinase-phosphosite associations obtained from PhosphoSite database. Following the evaluation protocol suggested in [46], we split the data into training, validation and test data based on classes and how many sites are associated with each of these classes. Kinases with more than 5 sites are considered as training classes. The BRNN model and zero-shot learning models are trained on this set, which contains 12.999 phosphorylation sites associated with a total of 218 kinases. The validation set includes the kinase-phosphosite associations of kinases for which there were 5 phosphorylation sites. The validation set includes 95 phosphorylation sites interacting with 19 kinases. The remaining kinases with less than 5 positively labeled examples constitute the test classes. The test data includes 297 phosphorylation sites that belongs to 127 classes. For tuning BRNN hyper-parameters we also split the training data of common kinases into two sets; we use 5% of phosphorylation sites in this set as validation set.

In this section, we report the model performance of DeepKinZero. In these experiments the models are trained on the train clases (common kinases), hyper-parameters are tuned using accuracy over the validation classes and performance is assessed on test classes (rare kinases). To assess the overall performance we use hit@k accuracy. This metric evaluates accuracy considering the top k predicted classes. If the true class is within the top-k predicted classes, it is considered a true positive prediction. We report results pertaining to k=1,3 and 5. For the zero-shot learning model we initialize the W matrix randomly from a uniform distribution and train the model using Adam optimizer [42].

The representation of the input site sequence and the kinase classes is critical in the models performance. To thoroughly asses the efficacy of using different embeddings, DeepKinZero is trained with three different input representations: One-Hot, Amino Acid Properties (AA Prop) and ProtVec with and without using BRNN. When a BRNN is employed, the BRNN is trained with the specified site sequence embeddings and the final layer of the BRNN is used as the final sequence embedding and input to the zero shot classifier.

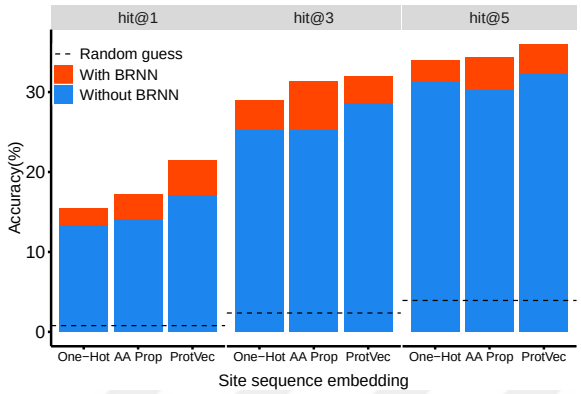

Figure 4.2: **Performance comparison of the models with different representation of the site sequence embedding with and without using a BRNN**. Performance is measured by having a hit in the top 1, top 3 and top 5 predicted kinase classes. Classifiers are trained with three different representations of the input sequence (site sequence embeddings): One-Hot, Amino Acid Properties (AA Prop) and ProtVec. When a BRNN is employed, the BRNN is trained with the specified site sequence embedding and the final layer of the BRNN is used as the final sequence embedding and input to zero shot classifier. The red bar indicates the performance improvement when a BRNN is used.

Figure 4.2 summarizes the results of using different site sequence embeddings. In these experiment the kinase embeddings is the best possible combination of the four kinase sources as detailed below. As shown in Figure 4.2, the site sequence embeddings obtained using a BRNN coupled with ProtVec vectors perform best with respect to almost all hit@k metrics. The model without BRNN embeddings that uses One-Hot sequence embedding as input, only predicts the right class as the top prediction in 13.46% of the test cases while the model with BRNN and ProtVec site embeddings predict the right class with 21.54% accuracy. As there are 127 test classes, the random guess will achieve only 0.78 % accuracy. Additionally, we observe that regardless of the input to the model, the use of BRNN model significantly improves the hit@1 accuracies. This observation also holds for the hit@3 and hit@5. We also observe that using ProtVec representation yields better results compared to the One-hot and AA Prop encodings regardless of their being coupled with BRNN or not, in all metrics except for the case of hit@5. We also inspect the value of using BRNN embeddings visually using t-SNE maps [62]. Figure 4.3 shows that the BRNN can separate the examples in the

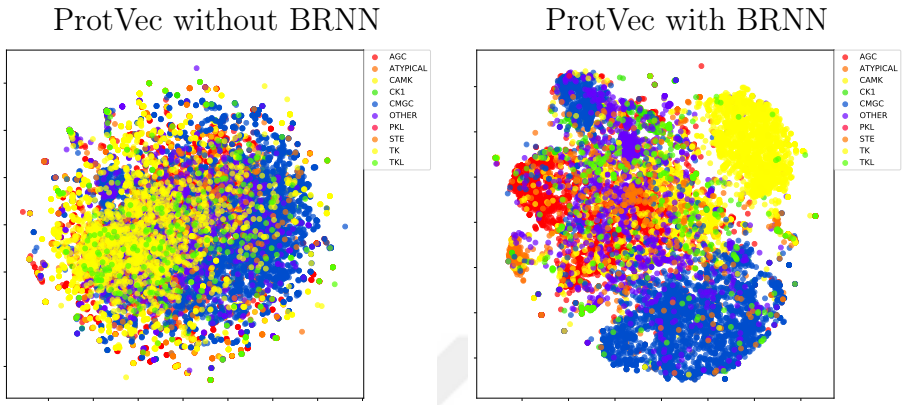

Figure 4.3: **TSNE visualization of the generated embedding with and without BRNN on ProtVec phosphorylation site representation**. The colors show different kinase groups, as you can see BRNN learned to represent the data in a way that fairly separates the groups while it is only trained on individual kinases

case of kinase groups better than the ProtVec representations, hinting that that it successfully captures additional critical information about kinases.

Next, we evaluate different sources for class embeddings. Table 4.2, in these results we used the best site sequence embedding technique (BRNN trained on ProtVec) and compare different combinations of class embedding features with each other. The first row shows the accuracies attained using a simple random guess. The next row is the zero-shot learning model when no kinase embeddings are used, here we simply input one-hot vector of kinases as kinase embeddings. As shown in the table, the performance of this model is even worse than random guess. The next 4 rows in the table, show the results of the models trained with kinase embedding vectors of individual data sources Thus they portray the strength of each source in isolation of others. Among the four possible kinase embeddings, the family hierarchy (see Section 3.4.1) of kinases contribute the most to the accuracy of the model which gets a 18.51% accuracy when used as the sole auxiliary information on kinases. As this hierarchy reflects the functional and sequence similarities of the kinases, it is expected that they carry valuable information about kinase similarities. When used in isolation of other sources, EC classification is found to be the least valuable source.

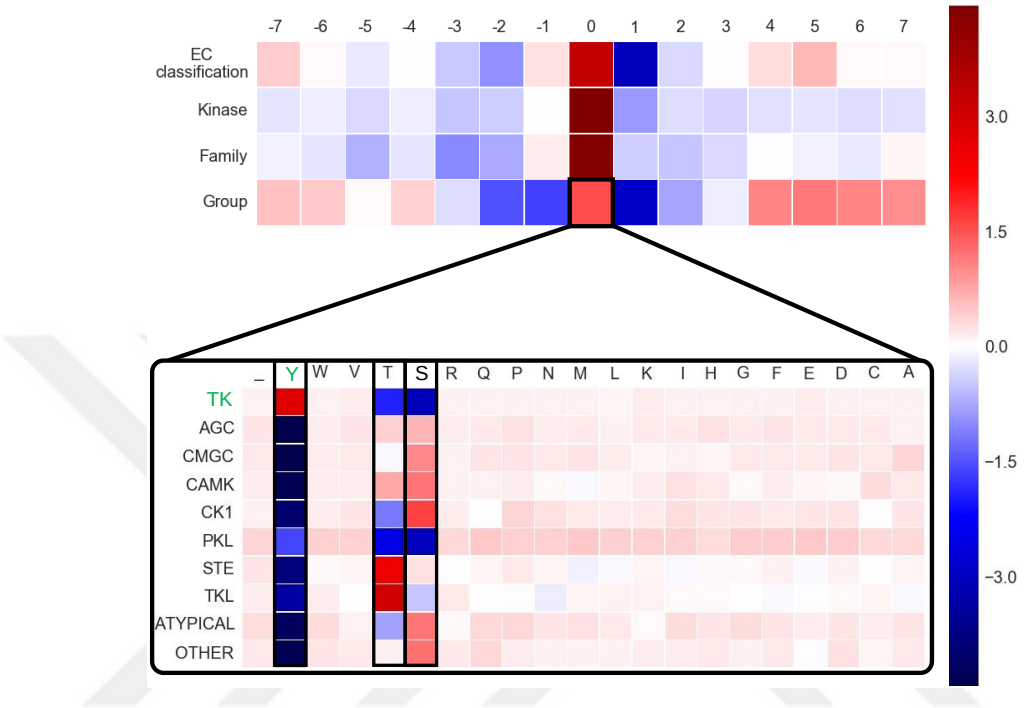

Figure 4.4: The visualization of Zero Shot learning weight (W) when the one-hot sequence directly fed to ZSL model.

Combining family hierarchy with one of the other class embeddings improves the model's ability and increases its accuracy. The model achieves 18.51% hit@1 accuracy by combining family hierarchy with EC classification and 19.19% with Kin2Vec. Furthermore, combining family hierarchy with EC classification or Kin2Vec vectors increases hit@5 accuracy from 32.99% to 33.33% and 36.02% respectively. Overall, the best accuracy is achieved by using Hierarchy, EC classification and Kinase2Vec vectors which achieves 21.54% on hit@1 accuracy, 31.98% on hit@3 and 34.68% on hit @5. Adding Pathway vectors into this combination degrades the accuracy significantly, although the using of pathways alone is the second best (fifth row) when used individually as an embedding. We suspect that this could be a manifestation of increased dimensionality.

We further inspect the learned weights to gain more quantitative insight into the models' mode of action. Figure 4.4.a shows the sum of weights assigned to each position in the input sequence. We analyze the weights for each of the class

embeddings. The center residue emerges as the most important residue, thus the model correctly learns to assign more weight to the center, where the phosphosite is located at. We also show the weights assigned to different amino acid types in each group of kinases. The weights assigned to different type of amino acids align well with the biological knowledge of the kinases. For example, the TK family which exclusively works on thyrosine residue (Y) puts a very large weight on tyrosine while other families do not. Similarly, the groups which work on threonine(T) and serine(S) are rightly assigned a large weight in that position and others have S as the most weighted amino acid. Only for the PKL group, incorrect amino acid type receives the higher weight. this is because there only 3 sites belonging to kinases in these groups and all of them are in the test set. Thus, the algorithm did not observe enough data for this group and did not learn very well how to handle this group. In this case the model relies on other sources of kinase embedding to handle these kinases.

Kinase Taxonomies	EC Classification	Pathways	Kin2Vec	hit@1	hit@3	hit@5
Random Guess				0.78	2.38	4.00
				0.67	1.35	2.69
✓				18.51	28.61	32.99
	✓			3.36	14.47	18.51
		✓		6.39	12.79	15.82
			✓	4.71	9.09	13.13
✓	✓			18.51	28.95	33.33
✓		✓		12.12	24.57	29.62
✓			✓	19.19	29.96	36.02
	✓	✓		10.10	16.83	24.57
	✓		✓	8.75	16.49	20.87
		✓	✓	7.40	13.80	18.51
✓	✓	✓		14.47	26.26	32.65
✓	✓		✓	**21.54**	**31.98**	**34.68**
✓		✓	✓	15.82	25.92	31.31
	✓	✓	✓	10.43	18.85	24.91
✓	✓	✓	✓	19.53	27.61	34.01

Table 4.2: **Comparison between the performance (in %) of models using different combinations of class embedding features**. We used the best phosphorylation sequence embedding here which is ProtVec, and we used one layer of BRNN with 500 node which had the best performance.

Chapter 5

Conclusion and Future Work

Deep learning has been successful in a wide range of machine learning tasks [63–66]. There has also been several successful applications of deep learning to computational biology tasks [16, 66–68]. Among these, MusiteDeep [16] demonstrates that a deep learning-based approach for recognizing phosphorylation sites exhibit significant performance improvements over the conventional feature engineering-based machine learning models. In this thesis, we present DeepKinZero as a novel method for kinase-specific phosphorylation site predictions that makes use of deep representations of the kinase and phosphosite sequences in a Zero-Shot learning framework. DeepKinZero, unlike conventional supervised methods can offer predictions for kinases which do not have any known phosphosites or have very few known phosphosites.

The zero-shot learning framework transfers knowledge from common kinases to rare kinases and this way it renders the predictions for classes that were never observed in the training phase possible. The ability to transfer learning between classes is based on the ability to learn a good class embedding function. To this end, we represent kinase classes, as multi-dimensional vectors. These kinase vectors define the characteristics of the kinases and is derived from auxiliary information on kinases, such as taxonomies of kinases or deep representation of their kinase domain sequences (as detailed in Section 3.4). In this work we also

explore various kinase representations as vectors and evaluate the contribution of each one to the model's accuracy. In future work, we will also explore the utility of using other class embeddings such as kinase cellular localizations or Gene Ontology annotations [69]. Similarly, we evaluate different phosphosite embeddings, and our computational experiments show that using BRNN with ProtVec vectors significantly improves the performance compared to the other methods. As a follow-up study, vectors generated by methods like PhosContext2vec [70] can be explored for use in phosphosite or kinase embeddings.

DeepKinZero complements the other kinase classification methods that focus on the kinases with abundant site information. DeepKinZero can be used as a second predictor on more challenging sites for which the current methods fail to assign a kinase with confidence. Additionally, DeepKinZero can also be used to transfer knowledge across species.

The work can be further extended in different dimensions. The zero-shot learning assumes that the testing instances are only classified into the candidate unseen classes, we too assume that the candidate classes at the time of testing all belong to the rare kinases. The generalized zero-shot learning is a more open setting where all the classes (seen and unseen) are available as candidates for the classifier at the testing phase [71]. This is a much harder problem because not only number of classes increases during testing, but also the classifier tends to classify objects to classes it had never been exposed to before in training. This problem needs more specific methods and careful tuning. In future work, we plan to extend this framework to this generalized setting.

Bibliography

[1] A. Bairoch, "The enzyme database in 2000," *Nucleic acids research*, vol. 28, no. 1, pp. 304–305, 2000.

[2] G. Manning, D. B. Whyte, R. Martinez, T. Hunter, and S. Sudarsanam, "The protein kinase complement of the human genome," *Science*, vol. 298, no. 5600, pp. 1912–1934, 2002.

[3] M. Ganapathiraju, N. Balakrishnan, R. Reddy, and J. Klein-Seetharaman, "Transmembrane helix prediction using amino acid property features and latent semantic analysis," in *Bmc Bioinformatics*, vol. 9, p. S4, BioMed Central, 2008.

[4] T. Pawson and J. D. Scott, "Protein phosphorylation in signaling–50 years and counting," *Trends in biochemical sciences*, vol. 30, no. 6, pp. 286–290, 2005.

[5] S. Müller, A. Chaikuad, N. S. Gray, and S. Knapp, "The ins and outs of selective kinase inhibitor development," *Nature chemical biology*, vol. 11, no. 11, p. 818, 2015.

[6] F. M. Ferguson and N. S. Gray, "Kinase inhibitors: the road ahead," *Nature Reviews Drug Discovery*, vol. 17, no. 5, p. 353, 2018.

[7] M. Mann, S.-E. Ong, M. Grønborg, H. Steen, O. N. Jensen, and A. Pandey, "Analysis of protein phosphorylation using mass spectrometry: deciphering the phosphoproteome," *Trends in biotechnology*, vol. 20, no. 6, pp. 261–268, 2002.

[8] A. Lundby, A. Secher, K. Lage, N. B. Nordsborg, A. Dmytriyev, C. Lundby, and J. V. Olsen, "Quantitative maps of protein phosphorylation sites across 14 different rat organs and tissues," *Nature communications*, vol. 3, p. 876, 2012.

[9] N. Blom, T. Sicheritz-Pontén, R. Gupta, S. Gammeltoft, and S. Brunak, "Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence," *Proteomics*, vol. 4, no. 6, pp. 1633–1649, 2004.

[10] H.-D. Huang, T.-Y. Lee, S.-W. Tzeng, and J.-T. Horng, "Kinasephos: a web tool for identifying protein kinase-specific phosphorylation sites," *Nucleic acids research*, vol. 33, no. suppl_2, pp. W226–W229, 2005.

[11] Y.-H. Wong, T.-Y. Lee, H.-K. Liang, C.-M. Huang, T.-Y. Wang, Y.-H. Yang, C.-H. Chu, H.-D. Huang, M.-T. Ko, and J.-K. Hwang, "Kinasephos 2.0: a web server for identifying protein kinase-specific phosphorylation sites based on sequences and coupling patterns," *Nucleic acids research*, vol. 35, no. suppl_2, pp. W588–W594, 2007.

[12] Y. Xue, Z. Liu, J. Cao, Q. Ma, X. Gao, Q. Wang, C. Jin, Y. Zhou, L. Wen, and J. Ren, "Gps 2.1: enhanced prediction of kinase-specific phosphorylation sites with an algorithm of motif length selection," *Protein Engineering, Design & Selection*, vol. 24, no. 3, pp. 255–260, 2010.

[13] M. Wang, T. Wang, B. Wang, Y. Liu, and A. Li, "A novel phosphorylation site-kinase network-based method for the accurate prediction of kinase-substrate relationships," *BioMed research international*, vol. 2017, 2017.

[14] J. Song, H. Wang, J. Wang, A. Leier, T. Marquez-Lago, B. Yang, Z. Zhang, T. Akutsu, G. I. Webb, and R. J. Daly, "Phosphopredict: A bioinformatics tool for prediction of human kinase-specific phosphorylation substrates and sites by integrating heterogeneous feature selection," *Scientific Reports*, vol. 7, no. 1, p. 6862, 2017.

[15] J. Gao, J. J. Thelen, A. K. Dunker, and D. Xu, "Musite: a tool for global prediction of general and kinase-specific phosphorylation sites," *Molecular & Cellular Proteomics*, pp. mcp–M110, 2010.

[16] D. Wang, S. Zeng, C. Xu, W. Qiu, Y. Liang, T. Joshi, and D. Xu, "Musitedeep: a deep-learning framework for general and kinase-specific phosphorylation site prediction," *Bioinformatics*, vol. 33, no. 24, pp. 3909–3916, 2017.

[17] Y. Dou, B. Yao, and C. Zhang, "Phosphosvm: prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine," *Amino acids*, vol. 46, no. 6, pp. 1459–1469, 2014.

[18] H. D. Ismail, A. Jones, J. H. Kim, R. H. Newman, and D. B. Kc, "Rf-phos: a novel general phosphorylation site prediction tool based on random forest," *BioMed research international*, vol. 2016, 2016.

[19] N. Blom, S. Gammeltoft, and S. Brunak, "Sequence and structure-based prediction of eukaryotic protein phosphorylation sites1," *Journal of molecular biology*, vol. 294, no. 5, pp. 1351–1362, 1999.

[20] R. Patrick, K.-A. Lê Cao, B. Kobe, and M. Bodén, "Phosphopick: modelling cellular context to map kinase-substrate phosphorylation events," *Bioinformatics*, vol. 31, no. 3, pp. 382–389, 2014.

[21] H. Horn, E. M. Schoof, J. Kim, X. Robin, M. L. Miller, F. Diella, A. Palma, G. Cesareni, L. J. Jensen, and R. Linding, "Kinomexplorer: an integrated platform for kinome biology studies," *Nature methods*, vol. 11, no. 6, p. 603, 2014.

[22] N. F. Saunders, R. I. Brinkworth, T. Huber, B. E. Kemp, and B. Kobe, "Predikin and predikindb: a computational framework for the prediction of protein kinase peptide specificity and an associated database of phosphorylation sites," *BMC bioinformatics*, vol. 9, no. 1, p. 245, 2008.

[23] T. Li, P. Du, and N. Xu, "Identifying human kinase-specific protein phosphorylation sites by integrating heterogeneous information from various sources," *PloS one*, vol. 5, no. 11, p. e15411, 2010.

[24] M. Simpson, R. Hsu, L. Keir, J. Hao, G. Sivapalan, L. Ernst, E. Zackai, L. Al-Gazali, G. Hulskamp, H. Kingston, *et al.*, "Mutations in fam20c are associated with lethal osteosclerotic bone dysplasia (raine syndrome), highlighting a crucial molecule in bone development," *The American Journal of Human Genetics*, vol. 81, no. 5, pp. 906–912, 2007.

[25] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Advances in neural information processing systems*, pp. 1410–1418, 2009.

[26] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in neural information processing systems*, pp. 935–943, 2013.

[27] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, pp. 2152–2161, 2015.

[28] G. Sumbul, R. G. Cinbis, and S. Aksoy, "Fine-grained object recognition and zero-shot learning in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 770–779, 2018.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[30] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[31] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[32] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[35] S. Fernández, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *International Conference on Artificial Neural Networks*, pp. 220–229, Springer, 2007.

[36] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[37] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in neural information processing systems*, pp. 545–552, 2009.

[38] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[39] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint arXiv:1602.02410*, 2016.

[40] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[41] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*.

[43] Y. Yu, Z. Ji, J. Guo, *et al.*, "Zero-shot learning via latent space encoding," *arXiv preprint arXiv:1712.09300*, 2017.

[44] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2014.

[45] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1425–1438, 2016.

[46] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *arXiv preprint arXiv:1707.00600*, 2017.

[47] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, *et al.*, "Devise: A deep visual-semantic embedding model," in *Advances in neural information processing systems*, pp. 2121–2129, 2013.

[48] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2927–2936, 2015.

[49] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," *arXiv preprint arXiv:1704.08345*, 2017.

[50] E. Asgari and M. R. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PloS one*, vol. 10, no. 11, p. e0141287, 2015.

[51] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[52] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, *et al.*, "The universal protein resource (uniprot)," *Nucleic acids research*, vol. 33, no. suppl_1, pp. D154–D159, 2005.

[53] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*.

[56] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*.

[57] S. Higashiyama, H. Iwabuki, C. Morimoto, M. Hieda, H. Inoue, and N. Matsushita, "Membrane-anchored growth factors, the epidermal growth factor family: Beyond receptor ligands," *Cancer science*, vol. 99, no. 2, pp. 214–220, 2008.

[58] P. V. Hornbeck, B. Zhang, B. Murray, J. M. Kornhauser, V. Latham, and E. Skrzypek, "Phosphositeplus, 2014: mutations, ptms and recalibrations," *Nucleic acids research*, vol. 43, no. D1, pp. D512–D520, 2014.

[59] M. Kanehisa and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.

[60] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "Kegg as a reference resource for gene and protein annotation," *Nucleic acids research*, vol. 44, no. D1, pp. D457–D462, 2015.

[61] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima, "Kegg: new perspectives on genomes, pathways, diseases and drugs," *Nucleic acids research*, vol. 45, no. D1, pp. D353–D361, 2016.

[62] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[63] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[64] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[65] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[66] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, "Deep learning for computational biology," *Molecular systems biology*, vol. 12, no. 7, p. 878, 2016.

[67] D. Quang and X. Xie, "Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences," *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107, 2016.

[68] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning," *Nature biotechnology*, vol. 33, no. 8, p. 831, 2015.

[69] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, p. 25, 2000.

[70] Y. Xu, J. Song, C. Wilson, and J. C. Whisstock, "Phoscontext2vec: a distributed representation of residue-level sequence contexts and its application to general and kinase-specific phosphorylation site prediction," *Scientific reports*, vol. 8, 2018.

[71] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *European Conference on Computer Vision*, pp. 52–68, Springer, 2016.