UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Technology

Klavs Jermakovs

Deep learning based protein-protein interaction prediction using universal protein sequence representations

Bachelor's Thesis (12 ECTS)

Curriculum Science and Technology

Supervisors:
MSci. Alekszej Morgunov
Prof. Mart Loog
Prof. Gholamreza Anbarjafari

Tartu 2020

Deep learning based protein-protein interaction prediction using universal protein sequence representations

Abstract:

Protein-Protein Interactions (PPI) govern key biological events in the cell and serve as a basis for understanding disease mechanisms and developing treatments. Currently used PPI predictive methods that rely on information from multiple sequence alignments are ineffective on proteins with few known homologs. Recent advances in self-supervised learning permit extracting complex features directly from the protein sequence (sequence embeddings) for later use with predictive algorithms. In this thesis, several sequence embedding methods were used in combination with Siamese deep neural network-based classifier architecture for PPI prediction. An average AUROC score of 0.70 on C1 test set suggests that more complex embedding methods such as UniRep and PLUS-RNN are able to extract more information relevant to PPI prediction from the protein sequence. Performance of all methods dropped markedly for C2 and C3 test sets, 0.62 for UniRep and 0.58 for PLUS-RNN, suggesting that further improvements are necessary to develop models that are more general in their coverage of the protein sequence space. The results of this work confirm that using pre-trained protein representations with deep learning based classifiers is a viable approach to PPI prediction from sequence alone.

Keywords:

Protein-protein interactions, Protein Representations, Siamese Neural Network, Transfer Learning, Sequence Embeddings, Self-supervised pre-training, Deep Learning

CERCS: P176 Artificial Intelligence; B110 Bioinformatics, medical informatics, biomathematics, biometrics; P310 Proteins, enzymology

Süvaõppel põhinev proteiin-proteiini vastastiktoime ennustamine kasutades universaalseid proteiinisekventsi kujutusviise

Lühikokkuvõte:

Proteiin-proteiini vastastiktoimed (PPI) juhivad olulisi bioloogilisi etappe rakus ning on aluseks haigusmehhanismide mõistmisel ja ravimite tootmisel. Hetkel kasutusel olevad PPI ennustamise meetmed, mis sõltuvad mitme järjestuse joondamise teabest, on ebatõhusad proteiinidel, millel on vähe kaardistatud homolooge. Viimased edusammud iseenesliku õppimise vallas lubavad eraldada keerulisi eripärasusi otse proteiini sekventsist (sekventsi kodeerimine), et neid hiljem ennustavate algoritmidega rakendada. Selle lõputöö käigus kasutati mitmeid sekventsi kodeerimise meetodeid koos Siiami sügava närvivõrgu põhise PPI ennustamise algoritmiga. Keskmine AUROCi skoor 0.70 C1 testandmestikus viitab, et keerulisemad kodeerimise meetodid nagu UniRep ja PLUS-RNN, suudavad proteiini sekventsist rohkem PPI ennustamisele asjakohast informatsiooni eraldada. Kõigi meetodite täpsus langes märkimisväärselt C2 ja C3 testandmestikes, 0.62 UniRepi ja 0.58 PLUS-RNNi puhul. See näitab, et üldisema kattuvusega proteiini sekventsi mudelite arendamiseks on vaja teha edasisi täiendusi. Selle töö tulemused tõestavad, et eeltreenitud proteiini kujutiste kasutamine koos sügavõppel põhinevate klassifitseerijatega, on võimalik lähenemine PPI ennustamisele ainult sekventsi põhjal.

Võtmesõnad: Proteiin-proteiini vastastiktoime, proteiini kujutusviisid, siiami närvivõrk, siirdeõpe, sekventsi kodeerimine, iseenesliku õppimise eeltreenimine, süvaõpe

CERCS: P176 Tehisintellekt; B110 Bioinformaatika, meditsiiniinformaatika, biomatemaatika, biomeetrika; P310 Proteiinid, ensümoloogia

TABLE OF CONTENTS

TERMS, ABBREVIATIONS AND NOTATIONS

AC – Auto Covariance

AUROC – Area Under Receiver Operating Characteristic

CT – Conjoint Triad

DIP – Database of Interacting Proteins

DL - Deep Learning

DNA - Deoxyribonucleic acid

FPR – False Positive Rate

HPRD – Human Protein Reference Database

LReLU – Leaky Rectified Linear Unit

ML - Machine Learning

MSA – Multiple Sequence Alignment

NLP - Natural Language Processing

PPIs – Protein-Protein Interactions

RF - Random Forest

RNN - Recurrent Neural Network

ROC – Receiver Operating Characteristic curve

SD – Standard Deviation

SGD – Stochastic Gradient Descent

SNN - Siamese Neural Network

SVM – Support Vector Machine

TPR – True Positive Rate

INTRODUCTION

Although we may be able to describe individual components of a living cell, it is the complexity of their interactions that gives rise to an incredible diversity in form and function. At the centre of this are interactions between proteins. Through protein-protein interactions (PPIs) signals in the cell are conveyed, up-regulating or down-regulating various processes in the cell (Lim, 2014). Such important processes as cell division and immune response is mainly dependant on PPIs. Mapping such interactions helps to elucidate underlying complex cell mechanisms and serve as a schematic diagram for bioengineers to alter cell function or design new therapeutic agents (Gonzalez and Kann, 2012).

Problem definition

Over the past few decades experimental methods have mapped only 92,000 from 240,000 estimated PPIs in a well-studied Yeast organism (Ding and Kihara, 2018). Current high-throughput experimental methods for proteome-wide PPI mapping are held back by their cost and bias to errors (Mrowka et. al., 2001). Computational approaches aim to assist experimental methods by offering predictive models that can facilitate experimental research, thus improving our understanding of cellular mechanisms. State-of-the-art PPI prediction tools rely on multiple sequence alignments (MSAs) to derive phylogenetic trees for use in PPI prediction. However, deriving phylogenetic trees for proteins that share few homologs is often difficult, if not impossible process. Therefore, methods that can directly extract PPI specific information from protein sequence alone are gaining research interest in the field of bioinformatics. Several already proposed methods that directly encode sequence physiochemical properties in a feature vector have been regarded as information shallow approaches that do not capture necessary data complexity for PPI prediction tasks. Recent emergence of self-supervised machine learning approaches in computational biology that allow to extract abstract and task versatile features from sequence are superseding hand-crafted sequence encoding techniques (Kimothi et al., 2019). Such approaches coupled with standard machine learning (ML) algorithms have already succeeded in different protein related computational tasks, such as protein family, localization, and stability prediction.

Thesis aims and objectives

To facilitate PPIs identification through utilizing self-supervised sequence representations, this thesis proposes a Deep Learning (DL) approach for PPI prediction. The primary aim is to construct the Siamese Neural Network capable of predicting PPIs. The secondary aim of this thesis is to compare the performance of proposed approach for PPI prediction based on different protein sequence representations extracted from self-supervised algorithms.

1 BACKGROUND AND LITERATURE REVIEW

1.1 Protein-protein interactions

Proteins are organic macromolecules that consist of different types of amino acids linked covalently in a polypeptide sequence. Amino acids differ by their physicochemical properties, such as polarity, size and chemical composition. Genetic code in the cell determines the length and composition of a polypeptide sequence. Environmental impact and interactions between amino-acids fold the polypeptide sequence forming the protein three-dimensional structure (Alberts, 2015). Proteins interact with other proteins forming transient or permanent protein complexes of various complexities. Physical PPIs form through direct docking of molecular surfaces between two protein molecules, as shown in Figure 1.1. Docking is mainly driven by the inter-play of physicochemical and geometric features. For example, formation of hydrogen bonds in respective geometric fits (Gainza et al., 2019; Kangueane, 2018). Through these interactions such complexes mediate or are involved directly in many key cell processes such as signal transduction, cell division, transport or immune-response (Lim, 2014). Understanding and mapping such interactions is the fundamental step to study cells at the molecular level.

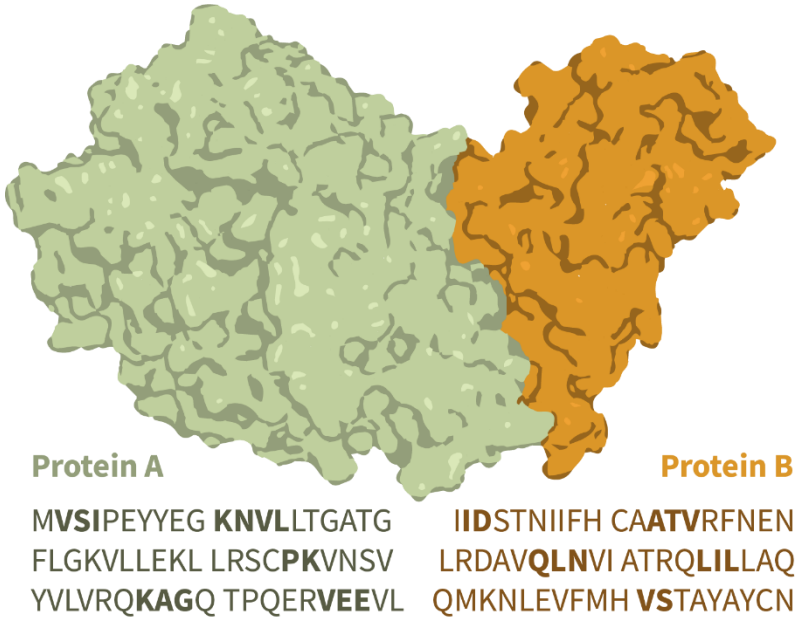

Figure 1.1. Physical interaction between two proteins. Amino acids directly participating in interaction are highlighted in respective protein sequences. Image adapted from article by Brito and Pinney (2017).

Information about occurring PPIs in an organism is used to build protein-protein interactome maps that represent pairwise interactions (Yu et al., 2008). Such maps, as shown in Figure 1.2, unravel complex molecular relationships in healthy or diseased organisms and are a tool to assist drug design or cell engineering (Gonzalez and Kann, 2012). Diseased organisms will contain undesired new interactions or disrupted healthy ones. Obtaining information on novel interactions and identifying any disruption in regular interactions allows researchers to identify potential drug targets (Gonzalez and Kann, 2012).

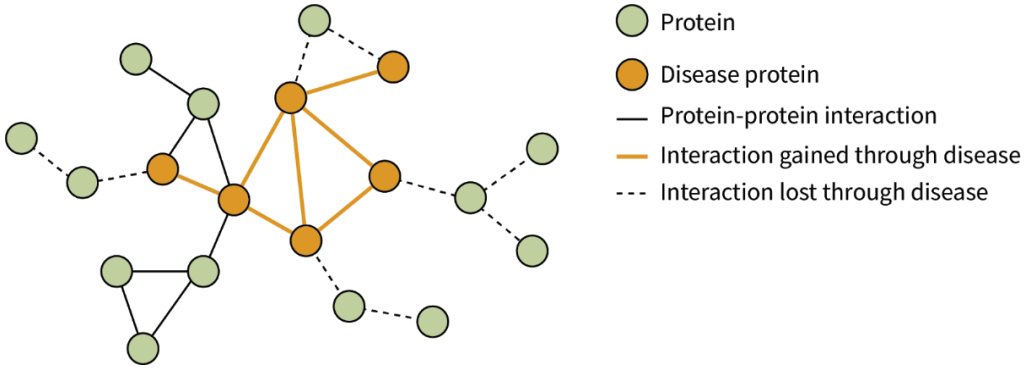

Figure 1.2. Modelled protein-protein interaction network in a diseased organism. Diseased proteins form or disrupt interactions in the cell, thus altering function. Figure was adapted from www.snap.stanford.edu.

1.2 Machine Learning

Machine learning (ML) is a subfield of Artificial Intelligence, that deals with the construction of computer algorithms and programs that are capable of learning and improving at a defined task from their own experience (Mitchell, 1997). Such algorithms learn to map certain inputs to certain outputs by constructing a function that does so. An example that is frequently used in literature is the spam filter algorithm which is capable of recognizing spam emails (Géron, 2017). This ML program has learned to recognize spam email, through comparing spam and non-spam emails.

1.2.1 Types of learning

ML algorithms can be divided into categories based on the type of supervision they get during learning. These three categories are: (i) supervised, (ii) self-supervised learning and (iii) unsupervised learning (Chollet, 2018). Indeed, there exists a fourth category known as reinforcement learning, which is not utilized within this thesis work.

During supervised learning, the algorithm learns to map input data to a certain output by comparing the predicted output with the target output. The target output in supervised learning is given together with input data, usually as a human-annotated label, e.g., spam or non-spam email. A support vector machine (SVM) is a well known supervised learning algorithm introduced by Vapnik (Cortes and Vapnik, 1995). The goal of the SVM algorithm in binary classification problems is to find an optimal decision boundary in the vector space that divides negative and positive training samples (Chollet 2018).

During unsupervised learning, the algorithm receives input data without any target outputs, since there are no target outputs, the algorithm learns by itself, trying to find patterns that characterize and describe input data the best by clustering or transforming data (Chollet, 2018). An example of an unsupervised algorithms is principal component analysis (PCA). PCA identifies the key descriptors in input data that explain most of the variance (Géron 2017).

During self-supervised learning much like in supervised learning, the algorithm learns to map input data to certain target output (Chollet 2018). However, unlike in supervised learning the target output is generated from input data itself and is not human-annotated. Example can be taken from the field of natural language processing (NLP), where an ML algorithm receives a sentence of words, masks a word, and tries to predict masked word by utilizing information about contextual words.

1.2.2 Artificial neuron

A relationship must exist between inputs x and outputs y for tasks that can be tackled by ML algorithms. The ML algorithms approximate this relationship by finding hypothesis $h(x)$ function out of all possible hypotheses that best describe this relationship relying on training data (Mitchell, 1997). An artificial neuron in the field of ML can be thought of as node that transforms **x** components $x_1, x_2, \ldots, x_i$ into some output by assigning certain weight θ to each component and taking sum (Géron, 2017). A linear hypothesis function can be modelled by by utilizing single neuron. This is expressed as:

$$h_\theta(x) = \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \ldots + \theta_i \cdot x_i \tag{1.1}$$

Or in more general form as:

$$h_\theta(x) = \theta^T x \tag{1.2}$$

Supervised algorithms internally assess how closely the hypothesis function matches the true relationship of training data by calculating the cost (also known as loss), i.e., the error between the predicted value derived from the hypothesis and the target value that is given together with training data. A common function to calculate cost is the mean squared error, which for m training samples can be written as:

$$C(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 \tag{1.3}$$

Minimal cost indicates that hypothesis fits the true relationship well, however, that is not usually the case, thus the hypothesis needs readjustment through changing the respective weights that define it. It is done by calculating the cost gradient and updating the weights against it. This process is called backpropagation and serves as the basis for supervised learning. The learning rate determines how heavily weights are updated and has the most impact on finding the cost minima (Mitchell, 1997)

In binary classification tasks the final output produced by the ML algorithm must be discrete, $y \in \{0,1\}$. For a neuron to produce output probability of inputs mapping to "0" or "1", the Sigmoid activation function is applied $\sigma(\cdot)$ on the linear value of neuron (Géron, 2017).

$$h_\theta(x) = \sigma(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{1.4}$$

The sigmoid function constraints the output values to be in the range of (0,1). Thus, by applying a step threshold function final prediction of discrete y can be obtained.

$$\hat{y}(x) = \begin{cases} 0 \; if \; h_\theta(x) < 0.5 \\ 1 \; if \; h_\theta(x) \geq 0.5 \end{cases} \tag{1.5}$$

A summary of the supervised learning process of binary classification algorithm consisting of one neuron is shown in Figure 1.3.

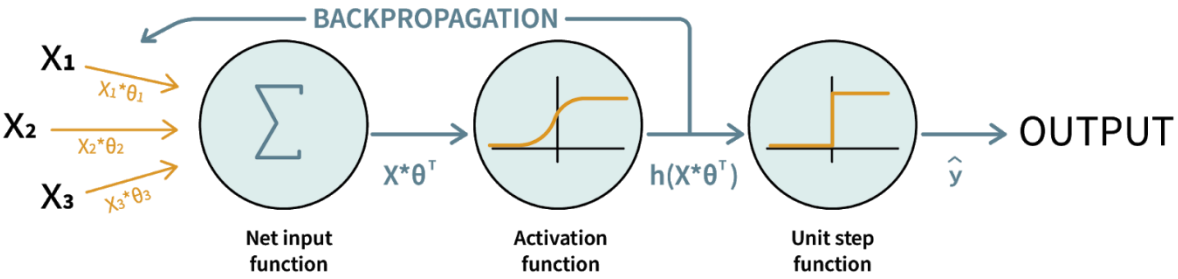

Figure 1.3. Supervised binary classification algorithm consisting of one neuron. Prediction is obtained by passing the weighted sum of input components through activation and threshold functions. By estimating predictive error, the weights are fine-tuned through back-propagation. Figure adapted from (Mitchell, 1997).

Leaky rectified linear unit (LReLU) activation function, shown in the Figure 1.4. Is type of often used rectified linear unit in multi-neuron networks to transform weighted sum of values in neuron into non-linear output for another neuron (Géron 2017).

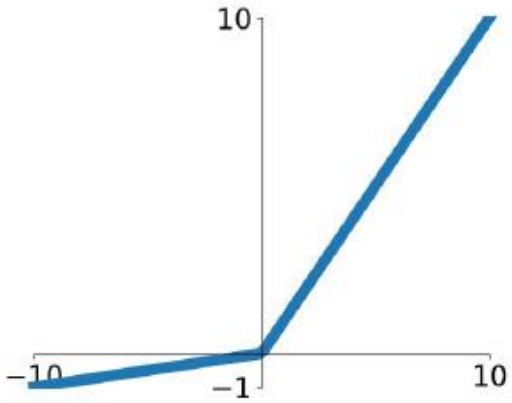

Figure 1.4. LReLU activation function. Weighted sum of neuron in horizontal axis is mapped to output values on vertical axis. Figure was taken from website www.quizlet.com.

1.2.3 Deep Feed-Forward Neural Networks

Deep Learning (DL) based algorithms deal with transforming data into increasingly more meaningful representations through forming patterns of artificial neuron activations (Chollet, 2018). In Deep Feed-Forward neural network architectures input components in input layers are sequentially propagated through several *hidden layers* into the final output layer. Such architectures, which are illustrated in Figure 1.5, require fixed-size feature vectors for input and are commonly used in supervised learning fashion for classification tasks on large quantities of complex data (LeCun et al., 2015).

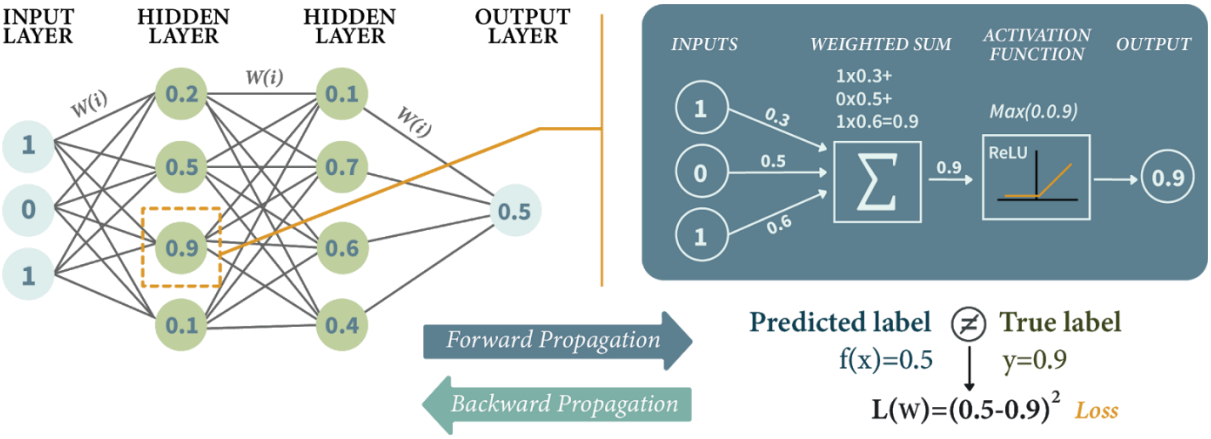

Figure 1.5. Illustrative principle of Feed-Forward neural network learning. Weighted sum of input values is produced in every neuron of hidden layer where activation function is applied. Final prediction is made by taking weighed sum of last hidden layer. Weights updated through calculating loss, i.e., error of prediction or cost. Figure adapted from paper by Du et al. (2017).

By receiving input **x** component values in input layer, multiple neurons with their respective weights act on component values and feed their outputs to consecutive neurons. Cascades of neurons build final weighted output that is merged in output layer where a prediction is estimated. By iterating several times on training data and re-adjusting weights between neurons, the neural network eventually learns to map input **x** as close as possible to target y through building patterns of neuron activations. Increasing number of hidden layers grants ability for neural network to build increasingly more patterns of neuron activations, thus allowing it to learn more complex functions.

1.2.4 Regularization

Ability of neural networks (NNs) to learn complex functions whilst having continuous strive to minimize cost while training can lead to them being prone overfitting the target task (Sarkar and Saha, 2019). Overfitting happens when the algorithm "memorizes" the training data by fitting a perfect hypothesis function that best describes the training data points, as shown in Figure 1.6. As a result, overfit model will not be as effective at mapping *unseen* inputs to correct outputs when receiving test data, i.e. it will be unable to generalize on target task. By using regularization methods NNs susceptibility to overfit the target task can be minimized resulting in a well-fit model.

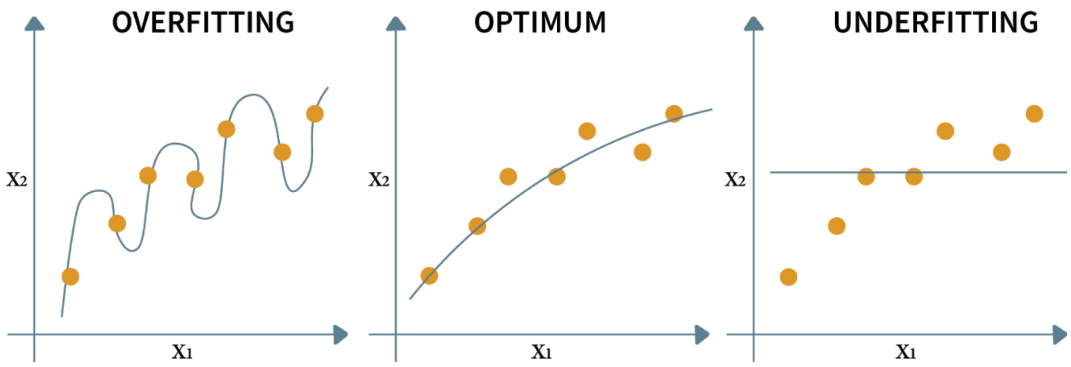

Figure 1.6. From left to right: overfit model, optimal model, underfit model. Figure adapted from www.machinelearningmedium.com.

Dropout is commonly used regularization technique for NNs. When dropout is applied to NN layers, it temporally removes several neurons and their respective connections in layer with some probability, as depicted in Figure 1.7. This approach forces NN to learn more robust patterns during training by introducing noise in layers that prevents memorization of training data (which can lead to overfitting) (Chollet, 2018).

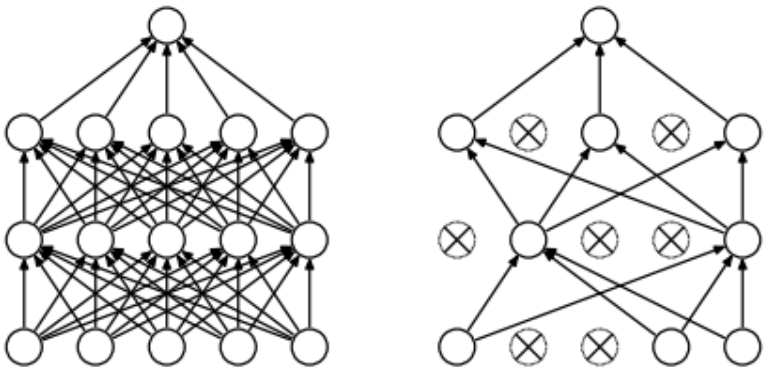

Figure 1.7. Left of the figure is NN without dropout. Right of the figure is NN with applied dropout. Figure was taken from article by Srivastava et al. (2014).

1.2.5 Siamese Neural Networks

Siamese neural networks (SNN) are utilized when the target task is learning the relationship between two or more inputs. SNNs consist of multiple identical sister neural networks that work on respective inputs and share identical weights between entire sister networks, since target task is invariant of whenever inputs are A and B or B and A (Chollet, 2018). If given two inputs, the respective sister networks will work on both and produce two outputs which are then combined into a single output before the final output layer where prediction is made. Weights of sister networks are updated simultaneously through backpropagation just as they are in a traditional NN.

1.3 Transfer learning

Transfer learning is a common method used in ML to approach tasks that have insufficient amounts of training data. Particularly in the feature-representation transfer learning approach, feature representations are constructed in either supervised or self-supervised manner on the pre-training source task, e.g. predicting missing words in the sentence. Learned feature representations are then used to fit predictive function on the target task, e.g. text sentiment prediction. Assumption in this approach is that feature representations that were learned during pre-training tasks, capture information relevant to the target task (Tan et al., 2018).

1.4 Protein Sequence Representations

Anfinsen et al. (1961 conducted experiments that proved that protein sequences in the form of amino-acids and their physicochemical properties contain all of the information needed for a protein to obtain its structural and functional properties (Anfinsen et al., 1961). Predictive performance of ML algorithms is directly dependent on how well the data given to the algorithm captures the information related to the task at hand (Domingos, 2012). Therefore, several sequence-based methods have been proposed to extract information from protein sequences that could be used together with machine learning classification methods for protein related computational tasks (Shen et al., 2007; Guo et al., 2008; Min et al., 2020; Asgari and Mofrad, 2015b; Kimothi et al., 2019; Alley et al., 2019). While there are other methods that rely on information other than sequence, e.g., structure, function of protein. These methods are outside the scope of this work since sequence is often only and most abundant information about protein (The UniProt Consortium, 2019).

1.4.1 Explicitly encoded representations

Shen et al. (2007) proposed a conjoint triad method (CT) for representing protein sequences in a PPI prediction task. The CT method divides protein sequence amino acids in seven classes depending on their properties. A sequence is then represented as triplet class frequencies. A major drawback this representation encoding method is that it captures only local triplet physicochemical properties along the sequence whilst most properties of the protein are determined by global properties of the whole sequence (Alberts, 2015).

Guo et al. (2008) improved on CT representation by utilizing auto covariance (AC) to calculate periodicity of amino-acid properties along the sequence. Improvement in PPI predictive algorithm performance was observed, since AC protein sequence representations provided algorithms with better information about global physicochemical properties of the sequence (Guo et al., 2008; Ding and Kihara, 2018).

1.4.2 Learned protein sequence representations

A major drawback of methods that hand-craft protein sequence representations, such as AC and CT, is their explicit manner of construction. Such an approach requires pre-existing knowledge (e.g. amino-acid properties) and might not capture all data needed for accurate computational task execution (Kimothi et al., 2019). Therefore, methods that do not rely on explicit feature extraction from sequence have been proposed.

Following the success of self-supervised Word2Vec (Mikolov et al., 2013) and doc2vec (Lau and Baldwin, 2016) word embedding algorithms in the field of NLP, similar algorithms have been developed for embedding biological information. ProtVec by Asgari and Mofrad (2015b) and Doc2Vec by Yang et al. (2018) respectively. In general, NLP word embedding techniques learn to embed words in Euclidean space in such way that semantic relationships between words are preserved, e.g., words with similar meaning are close. Protein sequence amino acids can be similarly thought of as words in sentences and can also be embedded in geometric space that preserves their underlying meaning in a sequence as shown in Figure 1.8.

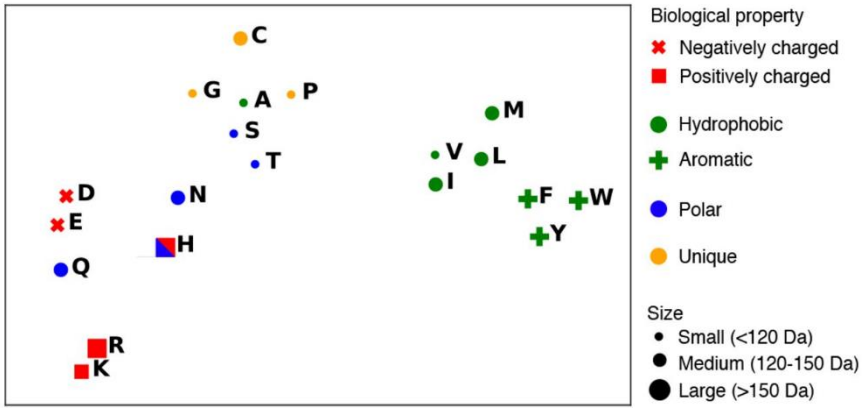

Figure 1.8. Embedded amino acids projected from high dimensional plane onto low-dimensional. Figure was taken from paper by Rives et al. (2019).

Protein sequence embeddings generated by self-supervised ProtVec and Doc2Vec algorithms are low-dimensional ($1x100$ and $1x64$ respectively) and do not require any biological prior knowledge (amino-acid physicochemical properties) for acquiring, compared to hand-crafted protein sequence representations generated by CT and AC methods. Such protein sequence embeddings have proven their versatility in different computational tasks such as: protein family classification (Asgari and Mofrad, 2015a; Kimothi et al., 2016), disordered protein visualization and classification (Asgari and Mofrad, 2015a), and protein functional property prediction (K. K. Yang et al., 2018). Additionally, research done by Kimothi et al. (2019) has shown that protein sequence embeddings learned directly from sequence data have outperformed AC and CT protein sequence representations in PPI prediction task, while maintaining low-dimensionality and universal applicability to different computational tasks.

1.5 Machine Learning methods for PPI prediction

Experimental methods relying on high-throughput approaches such as yeast-two-hybrid and affinity purification-mass spectrometry are most commonly used by researchers to detect PPIs (De Las Rivas and Fontanillo, 2010), yet such methods are well known to produce large number of false positive and false negative occurrences due to noise in reporter genes or contaminant protein presence (Mrowka et al., 2001; Huang and Bader, 2009). More precise experimental methods, such as X-Ray crystallography (Smyth and Martin, 2000), are time-consuming and expensive. Consequently, most PPIs have remained undiscovered by experimental methods. The general aim of computational methods is to facilitate PPI mapping. PPI prediction algorithms allow researchers to validate experimental data and

provide data about potential targets to be selected for experimental screening (Shoemaker et al., 2006).

SVMs have been previously utilized in predicting interacting and non-interacting protein pairs using AC and CT sequence representations (Guo et al., 2008; Shen et al., 2007). Despite SVMs ability to efficiently classify complex biological data it has been regarded as a computationally inefficient method for PPI prediction, since the number of calculations needed for establishing decision boundary grows with the amount of training data and vector size (Sarkar and Saha, 2019; Guo et al., 2008).

Park and Marcotte (2012) in their research utilized a decision-tree based supervised classification method for predicting PPIs based on AC sequence representations. A decision tree algorithm during training builds a tree-like structure by selecting an input attribute that best divides input space and expands until the stopping criterion is reached. Random Forests (RF) decision-tree based algorithm have been found to perform better than SVM in the prediction of PPIs (Kimothi et al., 2019; Park and Marcotte, 2012). However, RFs in the field of ML are regarded to be sensitive to noise and are prone to overfit the data (Sarkar and Saha, 2019).

DL based supervised classification algorithms automatically extract relevant and abstract features needed for classification task at hand (LeCun, 2015). Their ability to assign lower weights to non-important features and construct abstract representations of data in their hidden layers allows learning on noisy and complex raw data (LeCun, 2015; Rolnick et al., 2018). Thus, they are considered as prospective algorithms to be used in sequence-based PPI prediction. One of the first examples of sequence-based PPI prediction by DL classification algorithms is presented in the work of Sun et al. (2017) who used the AC method to represent protein sequences in the protein pair. In their research DL algorithm performance in PPI prediction outperformed traditional ML methods. Similar results were obtained by Gui et al. (2019) utilizing DL classification algorithm on protein pair sequences represented by AC and CT methods, therefore providing with sufficient evidence that DL-based methods might lead to improved PPI predictive performance if utilized together with self-supervised sequence representation extraction techniques.

1.6 Datasets

Development of computational methods that can distinguish interacting and non-interacting protein pairs require a dataset containing such information. Information about interacting protein pairs (a positive set) is stored in experimentally and literature curated PPI databases such as Database of Interacting Protein (DIP) (Xenarios et al., 2002) containing multi-species PPIs and Human Protein Reference Database (HPRD) (Keshava Prasad et al., 2009) containing human PPIs. However, since PPI databases contain insufficient amounts of data about non-interacting proteins for developing computational methods, a dataset of non-interacting proteins (a negative set) must be constructed. One approach proposed by Martin et al. (2005) is to randomly pair protein sequences and if such pairs are not present in the positive set then they are considered as valid "non-interacting" pairs for the negative set. The major drawback of this approach is that not all possible occurring PPIs in an organism are present in a positive set, thus such approach leads to possible false negatives in a negative set. Another approach utilized by Guo et al. (2008) is to generate a negative set by randomly pairing proteins that are localized in different cell compartments, such approach is frequently used since it minimizes occurrence of false-negatives due to low probability of interaction among proteins in different cell compartments. Two key datasets constructed by the latter approach are frequently utilized in literature to train or validate computational methods on PPI prediction tasks. Guo et al. (2008) composes datasets from several species utilizing the DIP database for binary prediction of PPIs, Yeast species dataset is frequently used in the methods relevant to this work. Another dataset commonly used for benchmarking PPI algorithm predictive performance was constructed by Park and Marcotte (2012). This dataset contains Yeast and Human PPIs extracted from the DIP database. In this work dataset constructed by Park and Marcotte (2012) was utilized.

2 METHODOLOGY

PPI prediction from a protein sequence is a binary classification task, therefore, dataset is obtained enclosing labelled data about interacting and non-interacting protein pairs, and their respective sequences. In this work, five different methods are implemented to represent otherwise varying in length protein sequence as a fixed-size feature vector. Fixed-size feature vectors in this work serve as a direct inputs into SNN architecture. SNN architecture is trained with training set containing protein-pair and label indicating the existing or non-existing interaction. Then trained SNN model evaluated on a protein-pair test set. Summary of approach taken in this study is depicted in Figure 2.1.

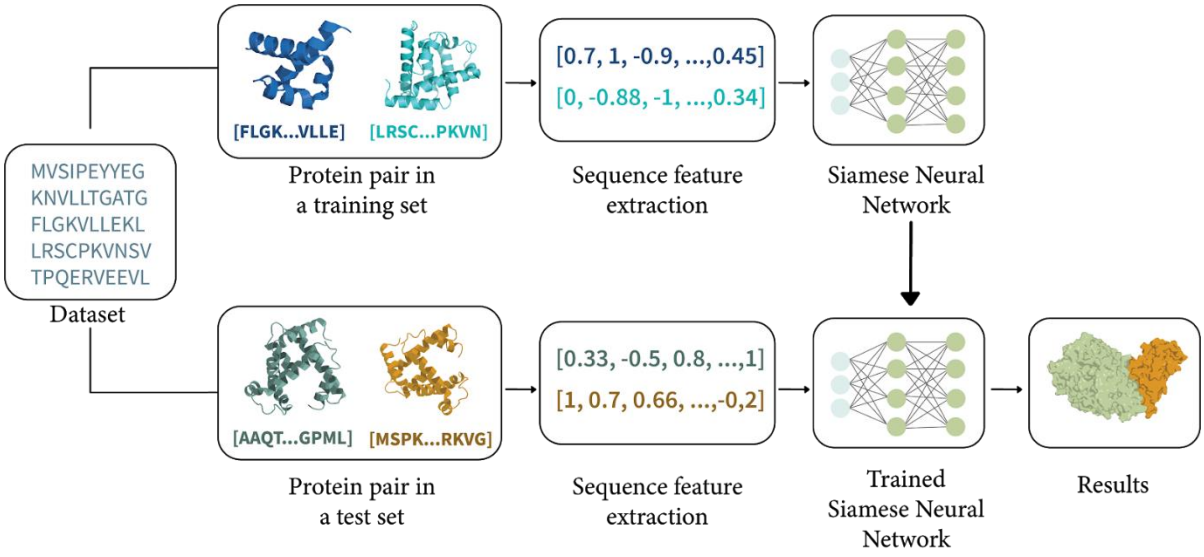

Figure 2.1. Outline of approach undertaken for PPI prediction. Sequences in protein pair are pre-processed by respective sequence feature extraction. Train set is utilized to train SNN then test set is utilized to obtain metrics of SNN performance. Obtained results are summarized in Results and Discussion section.

2.1 Dataset

Benchmark dataset constructed by Park and Marcotte (2012) enclosing data about Yeast and Human organism PPIs is utilized in this work for training and evaluating the performance of the designed SNN. Yeast organism dataset contains sequence information about 6,806 proteins and their interaction data of 14,938 PPIs. Human organism dataset contains information about 20,118 proteins and their sequences, forming in total 24,718 PPIs. Yeast and Human dataset contain sets of increasingly difficult test data C1-C3 shown in Figure 2.2. According to Park and Marcotte (2012) such division allows to fully evaluate model ability to generalize in pair-wise prediction problems involving PPIs, since test sets will contain different partitions of samples that algorithm has seen during training. C1 test class consists of protein pairs where both proteins are present in the training set, albeit in different interactions. C2 test class where only one protein in the test pair is present in the training set, and C3 where none of the proteins in the pair are found among training samples. It is important to note that according to Park and Marcotte (2012), C2 and C3 classes are particularly important to evaluate predictive model performance. Since most of the protein-pairs, in real-case scenario of implementing PPI predictive algorithm, will contain proteins that model has not trained on before.

Training set (8 data points)

Interacting protein pairs	Non-interacting protein pairs
1–2	1–4
1–3	2–3
1–5	3–5
2–4	4–5

Partitioning of the test

Test class	Property	Examples
C1	Both proteins in the test pair are found in the training set	2–5 3–4
C2	Only one protein in the test pair is found in the training set	1–7 4–8
C3	Neither protein in the test pair is found in the training set	6–7 8–9

Figure 2.2. Partitioning principle of test sets for evaluating predictive performance. Numbers symbolize unique proteins and number pairs symbolize protein-pair. Figure is adapted from paper by Park and Marcotte (2012).

In this work, 5 balanced train and test set splits from both Human and Yeast dataset were used, namely – 11,12,13,14,21.

2.2 Sequence Representations

PPI databases consist of a rather small amount of labelled training data for efficiently training protein sequence-based supervised DL classification algorithms. Thus, transfer learning technique is utilized in this work by first extracting sequence feature representations utilizing pre-trained algorithms and then using such representations together with interaction labels for training and testing PPI prediction algorithm. To obtain fixed-size feature vectors that capture versatile biological information about a protein from sequence without relying on pre-determined biological features, pre-trained self-supervised algorithms were implemented. These were ProtVec and Doc2Vec algorithms, which have been previously utilized to extract protein sequence representations for PPI tasks. Furthermore, UniRep and PLUS-RNN algorithms were implemented. These algorithms which extract protein sequence representations have not yet been used for PPI prediction. Additionally, simple baseline method for constructing sequence representations is utilized. Summary about implemented feature extraction methods is given in following sections.

2.2.1 ProtVec

ProtVec is a protein sequence embedding technique developed by Asgari and Mofrad (2015b) that utilizes skip-gram neural network architecture (Mikolov et al., 2013). Skip-gram technique in the field of NLP is used to represent words with similar meanings with similar feature vectors. Models utilizing skip-gram technique in NLP learn such feature vectors by training to predict masked word neighbouring words, i.e., contextual words. Similarly, during pre-training ProtVec learned to embed k-mers of amino-acid sequences depending on the contextual k-mers. Protein sequence embeddings that are obtained using this technique have proven to capture information about protein family, level of disorder and physicochemical properties (Asgari and Mofrad, 2015b). Heinzinger et al. (2019) outlined that such k-mer embeddings capture local information about the sequence yet lack information about direction of the sequence.

In this work, a pre-trained ProtVec model from kyu999 (2020) repository was used to obtain representations of protein sequences. Each protein sequence was split into overlapping 3-mers and given to the pre-trained ProtVec model depicted in Figure 2.3. Received 3-mer vectors from model were summed into the final fixed-size sequence representation of size $1x100$.

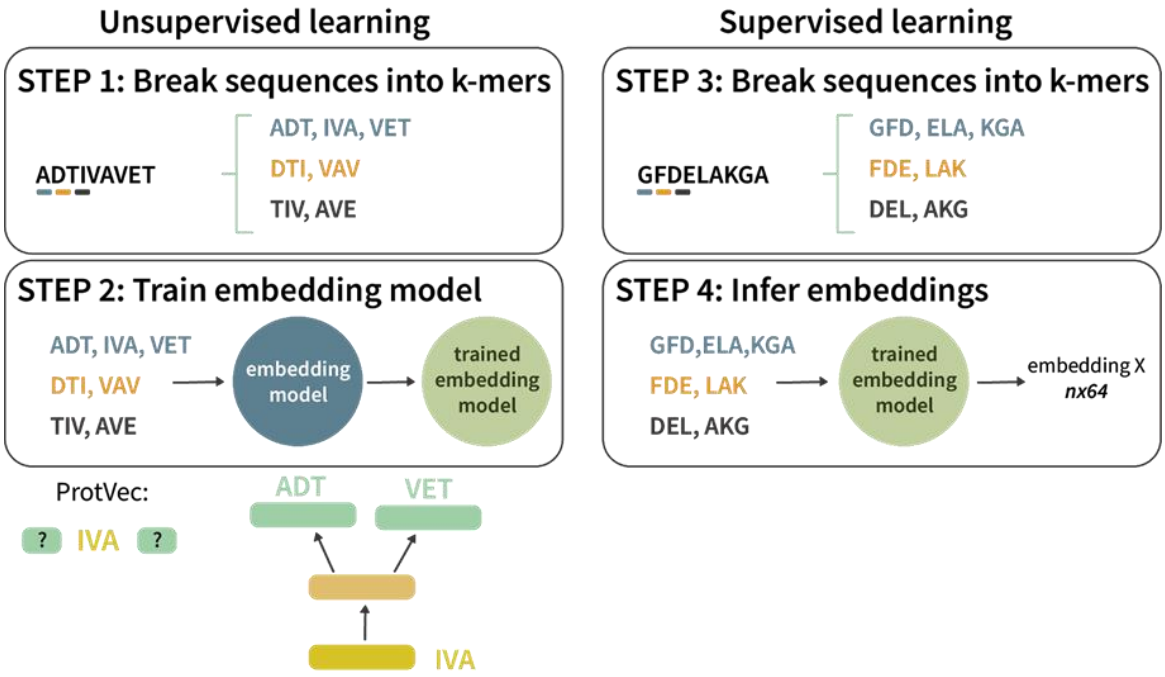

Figure 2.3. ProtVec pre-training scheme in step 1,2. Predicting context, given the k-mer. Step 3 and 4 depicts extraction of sequence feature vector. Figure adapted from paper by-Yang et al. (2018).

2.2.2 Doc2Vec

Similar to ProtVec, the Doc2Vec model by Yang et al. (2018) pre-trains by utilizing k-mers. Yet unlike the former, Doc2Vec utilizes DBOW pre-training technique (Le and Mikolov, 2014), i.e. predicting masked k-mer vector in a sentence, given contextual k-mer vectors and a context window vector, as shown in Figure 2.4. Such pre-trained protein sequence representations have proven to contain information about protein stability, localization, and global properties of amino acids sequence. Yang et al. (2020) demonstrated that Doc2Vec protein sequence representations capture PPI specific information and are applicable together with ML algorithms for human-virus PPI prediction.

In this work protein sequences were passed through pre-trained Doc2Vec implementation by the author (Fhalab/Embeddings_reproduction, 2020). Pre-trained model with k-mer length 3 and neighbouring window 7 was used. Obtained sequence representation vectors were of size $1x64$.

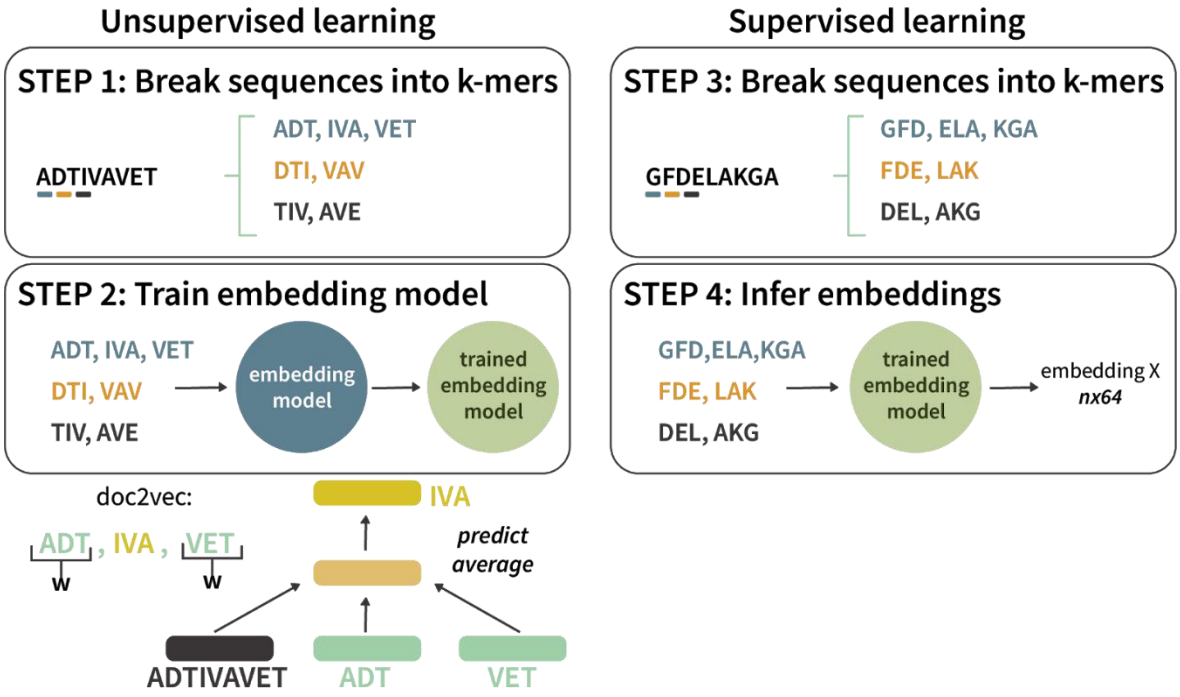

Figure 2.4. Doc2Vec pre-training and representation extraction steps. Step 1 and Step 2 depict the pre-training principle of predicting masked k-mer vector in context window by relying on neighbouring k-mer vectors and window context vectors. Step 3 and 4 show representation extraction principle from pre-trained model. Figure is adapted from paper by Yang et al. (2018).

2.2.3 UniRep

Recurrent neural networks (RNNs) are considered as one of the most powerful architectures in the field of NLP for text representation learning. Unlike NNs utilized in Word2Vec and Doc2Vec, RNNs can capture information about order of appearing words (Chollet, 2018), thus they have proven their applicability in capturing more complex information, ex. sentiment in text (Radford et al., 2017). Authors in (Alley et al., 2019) have designed the UniRep algorithm based on RNN architecture for constructing protein sequence representations. The UniRep algorithm pre-trained on protein sequence data by iterating through each amino acid in a sequence and making a prediction of next amino acid, based on information stored in RNN hidden layers about the sequence residues it has iterated through already. Therefore, learning the information to store for the most accurate prediction of consecutive amino acids (unidirectional sequence representation). Representation of a sequence can be extracted by forward passing the sequence through the model and storing the hidden layer numerical values. Alley et al. (2019) has outlined that such representations contain information about protein stability, structure and function, and that ML models using these representations have been successful at predicting these properties. This implies that such representation feature vectors might contain PPI specific information for predictive algorithms.

In this study, pre-trained 1900 hidden unit RNN was initialized according to instructions provided by authors of UniRep repository (Churchlab/UniRep, 2019). Protein sequence representations were obtained by forward passing the sequence through RNN hidden layers and taking an average of layer values resulting in vector of size $1x1900$.

2.2.4 PLUS-RNN

PLUS-RNN architecture implements bi-directional RNN (BiRNNs) (Schuster and Paliwal, 1997). BiRNNs unlike unidirectional RNNs implemented in UniRep, allow to process protein sequence from both directions. PLUS-RNN architecture pre-trains by predicting masked amino-acid in a sequence utilizing information stored in hidden layers about sequence before and after masked amino-acid. Such approach allows to construct bidirectional protein sequence representations that capture sequence directionality information from both sides. Min et al. (2020) have reported in their research that protein representations that are generated by PLUS-RNN captures global structural information about protein, thus could lead to PPI specific information.

Pre-trained PLUS-RNN is implemented in this study provided from authors repository (Mswzeus/PLUS, 2019). Representations are obtained by passing protein sequence through algorithm and averaging hidden layer numerical values, resulting in a vector of size $1x2048$.

2.2.5 Baseline

Baseline sequence representation containing sequence k-mer frequencies was generated to evaluate how well the predictive algorithm can perform when trained on simple sequence representation. Baseline for each protein was generated by splitting protein sequence into overlapping 3-mers and creating a fixed-size vector. Frequencies of respective k-mers were stored in the vector, as depicted in Figure 2.5. Such baseline sequence representation technique has also been used in the work of Alley et al. (2019).

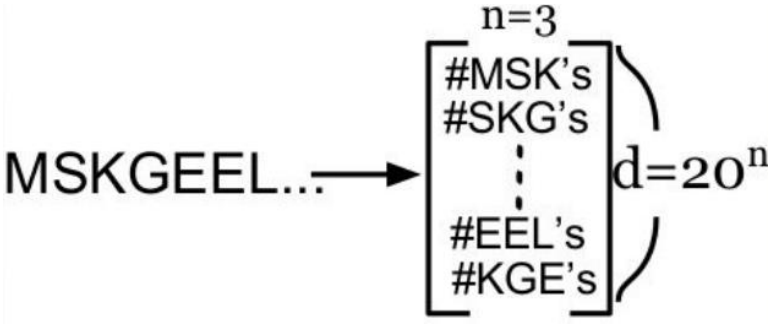

Figure 2.5. Baseline protein sequence representation vector. Protein sequence k-mer frequencies of size 3 are represented in a vector of size 20^n. Where 20 is the number of amino-acid types and n is the length of k-mer. Figure is obtained from paper by Alley et al. (2019).

2.3 Siamese Neural Network

In this study Feed-Forward type SNN was implemented for interaction prediction of given protein pair, as shown in Figure 2.6. SNN-type neural network was chosen because PPI prediction is invariant to protein-pair input order in neural network. Architecture is approximated from study performed by Du et al. (2017). Sister neural networks are built of 4 fully connected feed-forward layers with respective sizes of 1024, 512, 256, 128 neurons. Each subsequent layer width is reduced 2-fold and dropout with 10% probability is applied in order to reduce SNN capacity to overfit (Goodfellow, 2016). LReLU activation function is utilized in each layer for efficient learning process (Géron 2017). Sister neural network outputs are concatenated before two additional layers of size 256. The Sigmoid activation function with step threshold is applied in the final layer for binary prediction output.

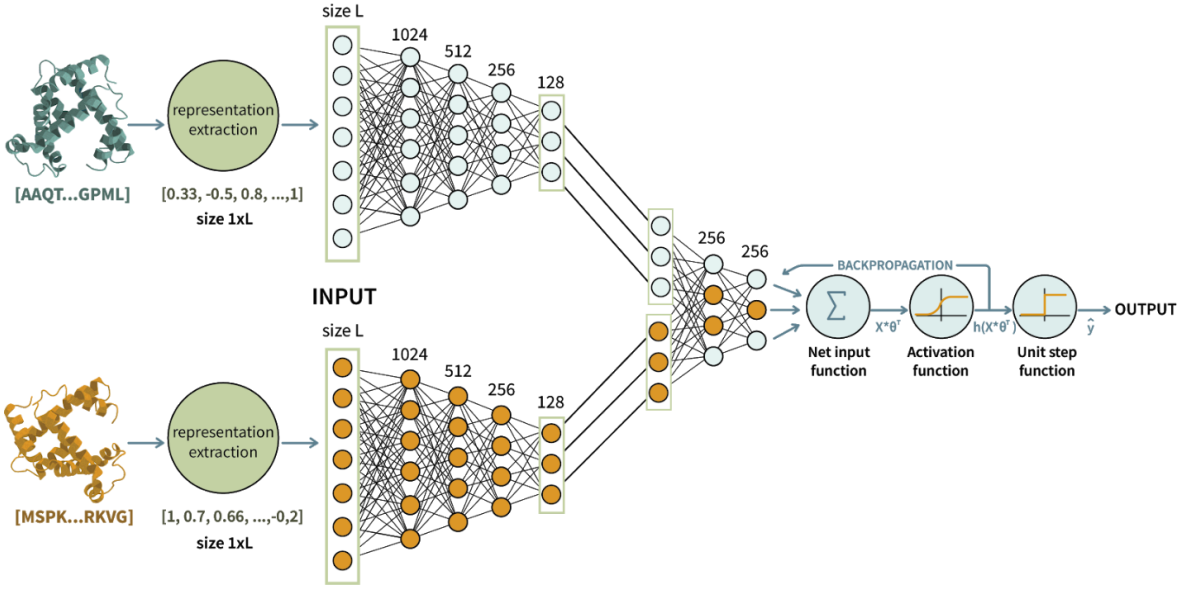

Figure 2.6. Constructed SNN architecture. Protein-pair sequences prior to being fed to predictive algorithm are represented as a feature vectors of size $1xL$ (L is the length of extracted feature vector). Two sister neural networks are concatenated before the final predictive layer.

In this work, SNN was trained with learning rate of 0.001 for 100 iterations using SGD weight optimizer (Géron 2017), with an exception being when trained on ProtVec representations (200 iterations). Several other training parameters in this work were tested. While adapting architecture few alternative configurations were explored. Setting sister network layer neuron counts to 64 resulted in SNN exhibiting high cost during training, whereas setting to 2000 resulted in no significant cost change. Increasing number of layers to 6 resulted in longer training times and no significant improvement in training cost.

2.4 Metrics

In this work Receiver Operating Characteristic (ROC) curve metrics were used to evaluate model predictive capabilities, as shown in Figure 2.7, since this metric performs The ROC curve represents a fraction of correctly predicted interacting pairs out of all interacting pairs (TPR) versus a fraction of correctly predicted non-interacting pairs out of all non-interacting pairs (FPR). FPR and TPR values on ROC curve are obtained at different classifier step function thresholds. Area under the ROC curve (AUROC) summarizes ROC curve in single numerical value between 0 and 1. It is equal to probability that randomly sampled datapoint will be classified correctly. Thus, larger value indicates better performing algorithm, yet value of 0.5 indicates random-guess predictive model.

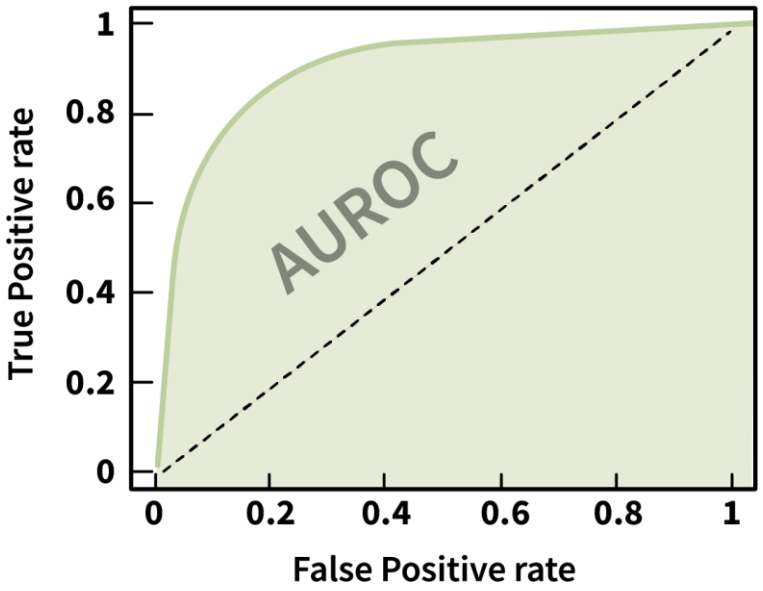

Figure 2.7. Area under the ROC curve (AUROC). Figure adapted from www.medcalc.org.

3 RESULTS AND DISCUSSION

3.1 Software and Hardware

Neural Network architecture was implemented in the Anaconda environment with Python 3.7. Front-end API Keras with Tensorflow as back-end was used to construct SNN. Development and testing was performed on Ubuntu 18.04 LTS operating system. Dual NVIDIA RTX 2080 graphics cards were used during training and testing. Additionally, 32GB of RAM and I7-8700K CPU were used[1].

3.2 Feature extraction computational time

Implemented feature extraction methods computational efficiency varied depending on feature vector size, algorithm complexity and computational resource parallelization. Summary about computational time required for obtaining database protein sequence representations is given in Table 3.1.

Table 3.1: Column on the left specifies the approach for constructing a protein sequence feature vector. Column on the middle specifies time in seconds for acquiring representations of 6,806 Yeast proteins. Column on the most right specifies time required for obtaining representations of 20,117 Human proteins.

Representation technique	Time, (s)	Time, (s)
UniRep	19459	69339
Doc2Vec	702	2338
PLUS-RNN	412	3163
Baseline	35	107
ProtVec	15	49

[1] SNN architecture designed through this work together with codes for data pre-processing, representation extraction and where to download datasets, can be accessed at GitHub repository. github.com/TheKursta/Undergraduate-thesis.

3.3 Testing Results

In this work DL based Siamese neural network PPI predictive capabilities were compared by training and evaluating its performance on five different sequence representation methods. At the initial stage SNN was trained on equal number on interacting and non-interacting protein sequence pairs which sequences were represented by one of the sequence embedding techniques discussed in section 2.2. Next, having SNN trained on respective protein pairs, its PPI predictive capabilities were evaluated by testing the model predictive performance on C1, C2, C3 respectively.

3.3.1 Yeast dataset

Table 3.2. Summary of SNN performance on five different representation techniques. Average AUROC score and standard deviation is given for each representation technique and test set.

Representation method	AUROC		
	C1	C2	C3
UniRep	0.74±0.03	0.67±0.03	0.60±0.02
PLUS-RNN	0.71±0.07	0.61±0.04	0.54±0.07
Doc2vec	0.75±0.02	0.59±0.04	0.50±0.05
ProtVec	0.63±0.01	0.60±0.03	0.58±0.03
Baseline	0.71±0.01	0.50±0.05	0.50±0.07

SNN predictive performance on five different representation techniques is summarized in Table 3.2 where averaged AUROC scores on five Yeast dataset train-test splits and respective standard deviations (SD) are reported. The results show that proposed DL approach on average was capable of distinguishing interacting and non-interacting proteins. Its performance varied not only on the way protein sequence was represented in protein pair used for training and testing but also on which test class it was being evaluated on. General trend in summarized data can be observed in Table 3.2 – predictive performance of SNN decreases through test C class. On average across all C sets the best SNN predictive performance was achieved when trained and tested on pair sequence representations acquired from UniRep algorithm with AUROC scores ranging from 0.60 to 0.74. PLUS-RNN and Doc2Vec representation methods led SNN to perform similarly on C2 classes with AUROC score around 0.60. Doc2Vec representations resulted in great C1 class performance with AUROC of 0.75, however, resulted in 0.50 on C3 class. ProtVec representations

showed fairly good performance ranging from 0.63 to 0.58. Interestingly, SNN performs comparably well on C1 test class when trained on baseline representations. Yet this result should be taken with care since follow-up performance on C2 and C3 classes were indifferent from random guess PPI model (AUROC 0.50).

3.3.2 Human dataset

Table 3.3 Summary of SNN performance on five different representation techniques. Average AUROC score and SD is given for each representation technique and test set.

Representation method	AUROC		
	C1	C2	C3
UniRep	0.67±0.04	0.63±0.02	0.62±0.02
PLUS-RNN	0.68±0.01	0.61±0.02	0.58±0.01
Doc2vec	0.62±0.05	0.55±0.02	0.53±0.03
ProtVec	0.59±0.02	0.57±0.01	0.56±0.01
Baseline	0.52±0.02	0.49±0.02	0.51±0.01

The predictive performance of SNN algorithm was further evaluated on Human dataset with five sequence representation techniques. Average AUROC scores obtained from five train-test splits are summarized in Table 3.3. Albeit being trained on more protein pairs, general trend in results obtained on Human dataset compared to results on Yeast dataset holds – SNN performance deteriorates over C classes. Yet overall SNN performance score reliability has increased as there is less variability in terms of SD compared to results obtained on Yeast dataset in Table 3.2. When trained on UniRep protein sequence representations, SNN showed the highest PPI predictive performance across C2 and C3 classes with AUROC score ranging from 0.62 to 0.63. PLUS-RNN representations resulted in slightly lower predictive performance across than UniRep in C2, C3 test sets ranging with respective scores of 0.61 and 0.58. In comparison, training on embedding representations obtained from Doc2Vec and ProtVec yielded lower predictive performance with AUROC scores of 0.53 and 0.56 in C3 class respectively. A baseline representation technique resulted in indifferent SNN PPI predictive performance to random guess model (AUROC score of 0.50) in all test sets.

3.3.3 Discussion

Combining protein sequence embeddings with classification algorithms trained on a specific task is a promising approach to many problems in computational biology. This work makes use of the recent advances in protein sequence representation learning to train DL-based PPI predictor. The information that is encapsulated in protein sequence representation dictates predictive performance of an algorithm the most. In particular, UniRep protein sequence representations have led to the best exhibited performance of proposed DL approach for PPI prediction across two datasets in C2 and C3 test sets. The increased performance on both UniRep and PLUS-RNN representations indicates that such methods can extract more representative information for PPI predictive task compared to ProtVec and Doc2Vec methods. One possible reason could underly in ability of these methods to capture complex and diverse information from sequence, such as stability, structure, function of the protein; other underlying reason might lie in captured contextual directionality. It could be speculated that RNNs ability to capture contextual directionality might translate into them being able to capture directionality of protein folding (Ellis et al. 2010), thus consecutively leading to more representative information for PPI prediction task. Since PPIs are highly dependent on the protein structure, one can speculate that contextual directionality during pre-training is able to extract structurally relevant information in a way analogous to how folding during protein synthesis also proceeds in a directional way.

While the top performing methods achieve promising results, caution must be taken with interpreting them without reference to the k-mer baseline, which serves as a control in this study. It is important to explore is important to explore the comparatively high SNN performance on simple k-mer baseline representations in Yeast C1 class compared to respective class in Human dataset. The unusual performance of baseline given its simplicity on Yeast dataset brings question of the underlying features in sequence representation that led SNN to perform well on exact test class. Possible explanation could be inferred from conclusions of recent research performed by Eid et al. (2020) stating that biases in training data are learned by ML models if data (feature vector) does not capture task specific information. In other words, algorithms instead of learning what information in feature vector contributes to the target task it has to excel at, start learning the patterns in training data if no useful features are present, for example finding the proteins that are interacting the most often and memorizing them to later predict that the particular protein interacts. Even if such algorithm performs well due to memorizing biases in the training it defeats purpose of

creating ML models that generalize, since such models will not be able to perform well on unseen sequences. Therefore, it could be the extreme case that SNN instead of learning PPI defining features learned interacting protein recurrence in training set and utilized successfully learned information in C1 test class due to fact that proteins in C1 class re-occur in training set, thus resulting the great performance. Yet interestingly such phenomena was not observed in Human dataset, which can be speculated that Human dataset contains much less proteins that serve as interaction hubs. Yet it was out of scope of this work to study representational biases in datasets, however, such investigation is crucial for bias-free datasets that could enable creating ML models that generalize better and could be investigated more in further studies.

Another observation in results was, that performance of SNN algorithm decreased over C classes in both datasets. This can be attributed to the fact that ML algorithms are able to learn only those features describing the output which are present in the training dataset. Meaning that it learns only on features describing interacting and non-interacting protein pairs existing in training data, and in case algorithm has to predict interaction on unseen protein it lacks information describing features that describe particular protein. This trend of decreasing performance is in line with similar performance comparative analysis performed by Kimothi et al. (2019) and Park and Marcotte (2012).

One should also consider the further limitations that arise with DL algorithm implementations of this kind, nevertheless the task. In particular, DL algorithms in general are required to be trained on large sets of data to well-fit the target task. Results on larger Human PPI dataset is in line with this trend, exhibiting more stable performance, however, this idea could not be explored further due to limitations of computational hardware needed for pre-processing the protein-pair datasets. It is worth highlighting that the best performing method, UniRep, takes the longest to compute sequence feature vectors owing to its complexity, as shown in Section 3.2. With larger computing capacity or code optimization larger datasets could be used and more robust predictive models could be trained and tested, thus perhaps unlocking the full capabilities of DL.

Another limitation of this study is the transfer learning approach that was undertaken for generating representations and training SNN. Steps of generating representations and training the SNN classifier in this study were separated. Yet commonly in transfer learning approach, classifier is working in tandem with pre-trained representation generation algorithm, backpropagating with lower learning rate into pre-trained layers and fine-tuning

it to give task specific representations. Further limitation of proposed SNN approach is the vast amount of possible hyper-parameters that could have been tuned for NN as such, possibly resulting in improved performance. However, an exhaustive investigation of various NN architectures was outside the scope of this research. Having established the approach, it stands to be decided whenever further performance gains should be sought in more complex architectures. It remains to be investigated how much more performance can be gained through various architectures and whenever a simple approach of training on more data would produce better results. Other DL approaches in field have been implemented together with ProtVec representations, however, these findings have not reported performance scores on all C classes in order to establish they reliability (Alakus and Turkoglu, 2019;Yao et al., 2019).

Finally, it is worth comparing, despite the differences, the work of this thesis with a very recently released study by Kimothi et al. (2019). Best C2 and C3 scores in this study ranged between 0.56-0.61 on Yeast dataset, which was below the top result in this thesis achieved by UniRep with 0.67 on C2. Therefore, this result should further support the point that complex representations such as UniRep, when coupled with DL-based classifiers, form the basis for the most promising architectures for the PPI prediction task.

4 CONCLUSION AND FUTURE WORKS

4.1 Conclusion

The aim of this thesis was to approach computational prediction of protein-protein interactions through utilizing deep learning (DL) algorithms together with protein representations extracted from self-supervised methods. In this work DL-based Siamese Neural Network (SNN) was constructed consisting of two sister feed-forward networks. In order to train predictive algorithm several pre-trained self-supervised methods were implemented to acquire protein sequence representations.

Performance of SNN was investigated on benchmarking dataset containing data about Yeast and Human PPIs. Obtained results from training and testing SNN predictive capabilities showed that proposed architecture was capable of distinguishing interacting and non-interacting protein pairs in both datasets. When trained on novel UniRep and PLUS-RNN representations SNN exhibited relatively high performance with respective AUROC scores of 0.74 and 0.71 on Yeast C1 test set, along with 0.67 and 0.68 on Human C1 test set. The high performance of UniRep on Yeast and Human dataset C2, C3 test sets is noteworthy, with AUROC score in range of 0.60 to 0.67 implying that complex UniRep sequence representations capture PPI relevant information that allow DL algorithms to excel.

Interestingly, the research revealed possible representational bias existing in Yeast database. Since simple k-mer based method resulted in unexpected AUROC score of 0.71 in C1 test set while 0.50 in Human dataset, indicating on possible interaction hubs leading to overfitting. Another observation was the improvement of DL based method reliability when utilized on larger dataset attributing to DL algorithm nature to improve on larger datasets.

To sum up, this study has revealed the PPI predictive performance benefits that can be achieved with complex pre-training algorithms being coupled with DL-based classifiers. Thus, bringing field one step closer to PPI prediction that relies solely on sequence data and does not depend on multiple sequence alignments.

4.2 Future works

Current immediate possibilities to enhance predictive performance of proposed SNN is to expand on its input representations. By effectively combining several sequence representation methods in such a way that they complement each other, information gaps in sequence representation could be filled, thus most likely improving the immediate performance. Yet as such approach might grow in unnecessary complexity, nascent Transformer architecture from field of NLP could be adapted. Transformer architectures have shown state-of-the-art performance in language understanding tasks, thus similarly could be used to understand language of protein sequence and therefore generate PPI task beneficial representations.

Another possibility to expand on task specific representations could be sought in multi-task learning. Multi-task learning builds on assumption that learning on one task could help or even unlock learning on another. Possible workflow might be using pre-trained UniRep model to predict subcellular localization of protein. In such way UniRep representations would build in diversity of encapsulated information. Therefore, possibly containing even richer information for PPI prediction task.

To take this study step further in topic of PPI, the next ambitious goal would be to investigate the regions of protein sequence that play the key role in formation of protein pair. By deriving which values in sequence representation vector play key role in producing neuron activations dictat interaction exists or not, self-supervised algorithms could be probed to infer the part of sequence that resulted in particular feature value.

REFERENCES

Alakus, T. B., & Turkoglu, I. (2019). Prediction of Protein-Protein Interactions with LSTM Deep Learning Model. *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 1–5.

2015 a, B. (2015). *Molecular biology of the cell* (Sixth edition). Garland Science, Taylor and Francis Group.

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., & Church, G. M. (2019). *Unified rational protein engineering with sequence-only deep representation learning* [Preprint]. Synthetic Biology.

Asgari, E., & Mofrad, M. R. K. (2015a). Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLOS ONE*, *10*(11), e0141287.

Asgari, E., & Mofrad, M. R. K. (2015b). ProtVec: A Continuous Distributed Representation of Biological Sequences. *PLOS ONE*, *10*(11), e0141287.

Brito, A. F., & Pinney, J. W. (2017). Protein–Protein Interactions in Virus–Host Systems. *Frontiers in Microbiology*, *8*.

Chollet, F. (2018). *Deep learning with Python*. Manning Publications Co.

Cong, Q., Anishchenko, I., Ovchinnikov, S., & Baker, D. (2019). *Protein interaction networks revealed by proteome coevolution*. 6.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

De Las Rivas, J., & Fontanillo, C. (2010). Protein–Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks. *PLoS Computational Biology*, *6*(6), e1000807.

Ding, Z., & Kihara, D. (2018). Computational Methods for Predicting Protein-Protein Interactions Using Various Protein Features. *Current Protocols in Protein Science*, *93*(1), e62.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87.

Du, X., Sun, S., Hu, C., Yao, Y., Yan, Y., & Zhang, Y. (2017). DeepPPI: Boosting Prediction of Protein–Protein Interactions with Deep Neural Networks. *Journal of Chemical Information and Modeling*, *57*(6), 1499–1510.

Eid, F.-E., Elmarakeby, H., Chan, Y. A., Martins, N. F., ElHefnawi, M., Allen, E. V., Heath, L. S., & Lage, K. (2020). Systematic auditing is essential to debiasing machine learning in biology. *BioRxiv*, 2020.05.08.085183.

Ellis, J. J., Huard, F. P., Deane, C. M., Srivastava, S., & Wood, G. R. (2010). Directionality in protein fold prediction. *BMC Bioinformatics*, *11*(1), 172.

Fhalab/embeddings_reproduction. (2020). [Jupyter Notebook]. Arnold Lab. https://github.com/fhalab/embeddings_reproduction (Original work published 2018)

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (First edition). O'Reilly Media.

Gonzalez, M. W., & Kann, M. G. (2012). Chapter 4: Protein Interactions and Disease. *PLoS Computational Biology*, *8*(12), e1002819.

Gui, Y., Wang, R., Wei, Y., & Wang, X. (2019). Dnn-ppi: A large-scale prediction of protein–protein interactions based on deep neural networks. *Journal of Biological Systems*, *27*(01), 1–18.

Guo, Y., Yu, L., Wen, Z., & Li, M. (2008). Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research*, *36*(9), 3025–3030.

Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., & Rost, B. (2019). *Modeling the language of life – Deep Learning Protein Sequences* [Preprint]. Bioinformatics.

Huang, H., & Bader, J. S. (2009). Precision and recall estimates for two-hybrid screens. *Bioinformatics*, *25*(3), 372–378.

Keshava Prasad, T. S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A., Balakrishnan, L., Marimuthu, A., Banerjee, S., Somanathan, D. S., Sebastian, A., Rani, S., Ray, S., Harrys Kishore, C. J., Kanth, S., … Pandey, A. (2009). Human Protein Reference Database—2009 update. *Nucleic Acids Research*, *37*, D767–D772.

Kimothi, D., Biyani, P., & Hogan, J. M. (2019). *Sequence representations and their utility for predicting protein-protein interactions* [Preprint]. Bioinformatics.

Kimothi, D., Soni, A., Biyani, P., & Hogan, J. M. (2016). Distributed Representations for Biological Sequence Analysis. *ArXiv:1608.05949 [Cs, q-Bio]*.

kyu999. (2020). *Kyu999/biovec* [Python]. https://github.com/kyu999/biovec (Original work published 2016)

Lau, J. H., & Baldwin, T. (2016). An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. *ArXiv:1607.05368 [Cs]*.

Le, Q. V., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *ArXiv:1405.4053 [Cs]*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lim, W. (2014). *Cell Signaling*. Routledge; 1 edition.

Martin, S., Roe, D., & Faulon, J.-L. (2005). Predicting protein-protein interactions using signature products. *Bioinformatics*, *21*(2), 218–226.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 3111–3119). Curran Associates, Inc.

Min, S., Park, S., Kim, S., Choi, H.-S., & Yoon, S. (2020). Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information. *ArXiv:1912.05625 [Cs, q-Bio, Stat]*.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Mrowka, R., Patzak, A., & Herzel, H. (2001). Is there a bias in proteome research? *Genome Research*, *11*(12), 1971–1973.

Mswzeus/PLUS: Protein sequence representations Learned Using Structural information (https://arxiv.org/abs/1912.05625). (n.d.). Retrieved May 13, 2020, from https://github.com/mswzeus/PLUS/

Park, Y., & Marcotte, E. M. (2012). Flaws in evaluation schemes for pair-input computational predictions. *Nature Methods*, *9*(12), 1134–1136.

Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2019). *Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences* [Preprint]. Synthetic Biology.

Rolnick, D., Veit, A., Belongie, S., & Shavit, N. (2018). Deep Learning is Robust to Massive Label Noise. *ArXiv:1705.10694 [Cs]*.

Sarkar, D., & Saha, S. (2019). Machine-learning techniques for the prediction of protein–protein interactions. *Journal of Biosciences*, *44*(4), 104.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y., & Jiang, H. (2007). Predicting protein-protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences*, *104*(11), 4337–4341.

Shoemaker, B. A., Panchenko, A. R., & Bryant, S. H. (2006). Finding biologically relevant protein domain interactions: Conserved binding mode analysis. *Protein Science: A Publication of the Protein Society*, *15*(2), 352–361.

Smyth, M. S., & Martin, J. H. J. (2000). X Ray crystallography. *Molecular Pathology*, *53*(1), 8–14.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sun, T., Zhou, B., Lai, L., & Pei, J. (2017). Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinformatics*, *18*(1), 277.

Tan, C., Yang, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A Survey on Deep Transfer Learning. *ArXiv:1808.01974 [Cs, Stat]*.

Team, K. (n.d.). *Keras documentation: Optimizers*. Retrieved May 13, 2020, from https://keras.io/api/optimizers/

The UniProt Consortium. (2019). UniProt: A worldwide hub of protein knowledge. *Nucleic Acids Research*, *47*(D1), D506–D515.

Xenarios, I., Salwínski, L., Duan, X. J., Higney, P., Kim, S.-M., & Eisenberg, D. (2002). DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, *30*(1), 303–305.

Yang, K. K., Wu, Z., Bedbrook, C. N., & Arnold, F. H. (2018). Learned protein embeddings for machine learning. *Bioinformatics*, *34*(15), 2642–2648.

Yang, X., Yang, S., Li, Q., Wuchty, S., & Zhang, Z. (2020). Prediction of human-virus protein-protein interactions through a sequence embedding-based machine learning method. *Computational and Structural Biotechnology Journal*, *18*, 153–161.

Yao, Y., Du, X., Diao, Y., & Zhu, H. (2019). An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*, *7*, e7126.

Yu, H., Braun, P., Yildirim, M. A., Lemmens, I., Venkatesan, K., Sahalie, J., Hirozane-Kishikawa, T., Gebreab, F., Li, N., Simonis, N., Hao, T., Rual, J.-F., Dricot, A., Vazquez, A., Murray, R. R., Simon, C., Tardivo, L., Tam, S., Svrzikapa, N., … Vidal, M. (2008). High-quality binary protein interaction map of the yeast interactome network. *Science (New York, N.Y.)*, *322*(5898), 104–110.

Acknowledgments

I would like to thank...

- My supervisors. MSci. Alekszej Morgunov, Prof. Gholamreza Anbarjafari and Prof. Mart Loog. For inspiring me to explore the unknown depths of closely inter-connected fields.
- Rain Eric Haamer. For giving me the last few pushes needed to release my work into the public.
- Frida Matiyevskaya. For encouragement when the times were rough. For sunny smiles and for bringing colors in this work and in every aspect besides it.
- My family. For support and motivation.

NON-EXCLUSIVE LICENCE TO REPRODUCE THESIS AND MAKE THESIS PUBLIC

I, Klavs Jermakovs

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Deep learning based protein-protein interaction prediction using universal protein sequence representations

supervised by MSci. Alekszej Morgunov, Prof. Mart Loog, Prof. Gholamreza Anbarjafari

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Klavs Jermakovs

20/05/2020