1 2 9 0

UNIVERSIDADE Đ
COIMBRA

Luís Pedro Baptista Silva

EXPLORING DEEP LEARNING ARCHITECTURES AND RELIABILITY OF SEVERAL DATASETS TO PREDICT PROTEIN-PROTEIN INTERACTIONS

Thesis submitted to the Faculty of Sciences and Technology of the University of Coimbra for the degree of Master in Biomedical Engineering with the specialization in Bioinformatics, supervised by Prof. Dr Joel P. Arrais and Prof. Dr. Carlos Pereira.

Julho de 2020

1 2 9 0 FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE D COIMBRA

Luís Pedro Baptista Silva

Exploring Deep Learning Architectures and Reliability of Several Datasets to Predict Protein-Protein Interactions

Thesis submitted to the
University of Coimbra for the degree of
Master in Biomedical Engineering

Supervisors:
Prof. Dr. Joel P. Arrais
Prof. Dr. Carlos Pereira

Coimbra, 2020

The work developed in this master thesis was in collaboration with:

Centre for Informatics and Systems of the University of Coimbra

CISUC

Esta cópia da tese é fornecida na condição de que quem a consulta reconhece que os direitos de autor são pertença do autor da tese e que nenhuma citação ou informação obtida a partir dela pode ser publicada sem a referência apropriada.

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognize that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

Acknowledgements

Antes de tudo gostaria de expressar os meus sinceros agradecimentos a todas as pessoas que contribuiram para tornar este trabalho realidade.

Em primeiro lugar queria agradecer ao Professor Joel P. Arrais e ao Professor Carlos Pereira, não só pela confiança que depositaram em mim e pela oportunidade que me proporcionaram, mas também pelo apoio constante. As frequentes reuniões onde ajudaram-me a organizar as minhas ideias para posteriormente as conseguir expor da maneira mais clara e concisa foram fundamentais e agradeço-lhes profundamente. Apesar disso, a liberdade que me atribuiram, foi ao mesmo tempo essencial para realmente conseguir descobrir e explorar o tema e obrigar-me a estimular a minha imaginação, porque no final de contas qualquer progresso na ciência resulta de uma nova interpretação proveniente da nossa imaginação.

Gostaria de agradecer também aos meus amigos. Aos novos amigos que tive a oportunidade de conhecer nesta etapa mais recente da minha vida, mas que desde já posso dizer que levo para a vida. Aos amigos que já vinham de trás e que apesar de termos escolhido caminhos separados nunca deixaram de ser um marco fundamental de apoio e também de insipiração. Queria agradecer-vos profundamente tenho a certeza que as nossas memórias nao se vão perder e estarei sempre pronto para criar outras novas e mais marcantes.

Por último, e muito pelo contrário não menos importante, queria agradecer profundamente à minha família. Por serem o meu porto de abrigo desde sempre quer nos bons e nos maus momentos. Pela liberdade que me atribuiram desde sempre na decisão das minhas escolhas, mas sempre apoiando independentemente do resultado final. Obrigado por tudo! Estamos a chegar ao fim de uma etapa marcante da minha vida e sei que estarão cá para a próxima que se aproxima.

Financing

This research has been funded by the Portuguese Research Agency FCT, through D4 - Deep Drug Discovery and Deployment (CENTRO-01-0145-FEDER-029266).

"Imagination is more important than knowledge."

"Condemnation without investigation is the height of ignorance."

"The important thing is to never stop questioning."

Albert Einstein

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."

Marie Curie

"The world is one big data problem."

Andrew McAfee

Resumo

As proteínas são indispensáveis para os seres vivos e são a base de quase todos os processos celulares. No entanto, estas macromoléculas raramente actuam sozinhas, formando as interações proteína-proteína. Dada a sua importância biológica não é de surpreender que a sua desregulamentação seja uma das principais causas de vários estados de doença.

A súbita onda de interesse nesta área de estudo motivou o desenvolvimento de métodos *in silico* inovadores. Apesar dos avanços óbvios nos últimos anos, a eficácia destes métodos computacionais permanece questionável. Ainda não existem evidências suficientes que apoiem o uso apenas de técnicas *in silico* para prever interações proteína-proteína ainda não determinadas experimentalmente. Está provado que uma das principais razões que leva a esta situação é a inexistência de um conjunto de dados de interações negativas padrão. Contrariamente à grande abundância de interações positivas disponíveis publicamente, os exemplos negativos são frequentemente gerados artificialmente, culminando em amostras tendenciosas.

Nesta tese de mestrado, é apresentado um novo conjunto de dados imparciais, que não restringe em demasia a distribuição das interações negativas. Além do novo conjunto de dados, são também propostos modelos distintos de aprendizagem profunda como uma ferramenta para prever se duas proteínas individuais são capazes de interagir uma com a outra, usando exclusivamente as sequências completas de aminoácidos. Os resultados obtidos indicam firmemente que os modelos propostos são realmente uma ferramenta valiosa para prever interações proteína-proteína, principalmente quando comparados com as abordagens existentes, além de destacarem ainda que existe espaço para melhorias quando implementados em conjuntos de dados imparciais.

Palavras-chave: Interação Proteína-Proteína, Conjunto de dados, Aprendizagem profunda, Rede Neuronal Convolucional, Rede Neuronal Completamente Convolucional.

Abstract

Proteins are indispensable to the living organisms and are the backbone of almost all the cellular processes. However, these macromolecules rarely act alone, forming the protein-protein interactions. Given their biological significance it should come as no surprise that their deregulation is one of the main causes to several disease states.

The sudden surge of interest in this field of study motivated the development of innovative *in silico* methods. Despite the obvious advances in recent years, the effectiveness of these computational methods remains questionable. There is still not enough evidence to support the use of just *in silico* techniques to predict protein-protein interactions not yet experimentally determined. It is proved that one of the primary reasons leading to this situation is the non-existence of a "gold-standard" negative interactions dataset. Contrary to the high abundance of publicly available positive interactions, the negative examples are often artificially generated, culminating in biased samples.

In this master thesis a new unbiased dataset is presented, that does not overly constraint the negative interactions distribution. Beyond the novel dataset, also distinct deep learning models are proposed as a tool to predict whether two individual proteins are capable of interacting with each other, using exclusively the complete raw amino acid sequences. The obtained results firmly indicate that the proposed models are actually a valuable tool to predict protein-protein interactions, principally when compared with the existing approaches, while also highlighting that there is still some room for improvement when implemented in unbiased datasets.

Keywords: Protein-Protein Interaction, Datasets, Deep Learning, Convolutional Neural Networks, Fully Convolutional Neural Networks.

Contents

List of Figures

List of Tables

List of Abbreviations

1D 1 Dimensional.
2D 2 Dimensional.
3D 3 Dimensional.

AC Auto Covariance.
ANN Artificial Neural Networks.

CNN Convolution Neural Network.

DL Deep Learning.
DNA Deoxyribonucleic Acid.

FCNN Fully Convolutional Neural Network.

GO Gene Ontology.

MD Molecular dynamics.
ML Machine Learning.
mRNA messenger Ribonucleic Acid.

PPI Protein-Protein Interaction.

RF Random Forest.
RNA Ribonucleic Acid.

SVM Support Vector Machine.

1

Introduction

1.1 Background

1.1.1 Proteins

The first appearance of the term *protein* dates back to 1839. The concept was first introduced by the chemist Gerhardus Johannes Mulder, during his studies on the composition of animal substances. However, this term was only later defined as an actual protein by Jöns Jakob Berzelius [1]. The etymology of the word is still somewhat uncertain, being ascribed to either the Latin word *primarius* or the Greek word *proteios* [2], which translate to ”primary” and ”first place”, respectively, a fitting description for the molecule that would come to be recognized as one of the most important.

Fast forwarding to the present and after immense studies conducted over the years, proteins are now acknowledged as one of the most essential macromolecules of all living systems. Their presence ranges from the simplest living beings, the bacteria and viruses, to the more complex, such as the humans. Proteins are highly elaborate structures composed by long polypeptide chains, which consist of relatively simple monomeric units, called amino acids, connected with each other by peptide bonds. The same set of these 20 simple units, arranged in a vast number of ways, is able to construct the majority of the proteins in all living systems. Every single amino acid has, in their composition, a carbon atom bonded to a carboxyl and an amino group. They differ one from the others in their side chains, which vary in their structure, electric charge and size. The side chains are the structures that grant the amino acids distinctive chemical and physical properties. Notwithstanding the 20 common amino acids, there are also two rare amino acids, found in just a few proteins, but because they are specific to some species and tremendously uncommon they are often disregarded [3].

Each individual protein is defined by an unique sequence of amino acids, also designated as the protein primary structure. The actual sequence is genetically encoded and posteriorly translated into proteins. This is achieved during the protein biosynthesis. Firstly, during the transcription phase, the Ribonucleic Acid (RNA) polymerase creates the strands of messenger Ribonucleic Acid (mRNA) using, as template, the strands of the Deoxyribonucleic Acid (DNA) encoding a protein. Once the mRNA is formed and when it reaches the ribossome, the translation phase occurs and each codon, which is a triplet of mRNA nucleotide bases, is decoded into a specific amino acid, according to the genetic code. This process is repeated until the whole protein sequence is obtained and the individual amino acids are connected with the followings. The binding process of the amino acids requires condensation reactions between the carboxyl group of the current amino acid and the amino group of the next, in which a water molecule is removed, forming the peptide bonds. Notwithstanding the primary structure vital importance, that is not their natural conformation, as they spontaneously adopt 3 Dimensional (3D) structures. The 3D conformation that a protein assumes is determined by several factors, among which stand out the specific location of each amino acid in the sequence and the non-covalent interactions triggered [4].

Proteins are crucial biomolecules due to their execution of a vast number of vital functions within the cells, from participating in cellular structure to promoting chemical reactions and even controlling immune responses. There is a tight relationship between a protein function and its structure, as a proteins adopts a spatial arrangement that reflects its biological role. However, proteins are not static structures, they are dynamic molecules, constantly undergoing subtle to dramatic conformation changes, that can also be provoked by external factors and unexpectedly may lead to denaturation and consequently loss of function.

1.1.2 Protein-Protein Interactions

Proteins play vital roles in diverse biological processes, however they rarely act alone, as they produce elaborate complexes with other proteins that allow them to perform the function they were designed to [5]. A Protein-Protein Interaction (PPI) is an intentional and non-generic physical interaction between proteins, originated from a specific biological context, since the interactions depend of a vast number of factors such as cell cycle, cell type and environmental conditions [6]. During this interaction an elaborate complex is formed, as a consequence of the association of the various polypeptide chains involved. There is not an univocal relationship between the proteins of a complex, they are able to create different complexes, when interacting with different proteins.

The PPIs are classified into different categories, according to their structural and functional properties [7]. An interaction can be classified by the lifetime of the complex formed. It is a permanent interaction, if it is stable and the complex is maintained for a long time, on the contrary, a transient interaction only occurs in nature for a brief period of time. Depending on the complex assembly they are categorized either as hetero-oligomer, when composed by distinct proteins, or as homo-oligomer, if the individuals proteins are identical. Finally, according to their stability, if its proteins are capable of existing independently as stable structures, then the complex is classified as non-obligated. On the other hand, in an obligated complex its proteins are not capable of creating an independent stable structure on their own.

Initially, it was wrongly proposed by Fischer [8] that the success and efficiency of the binding process of proteins was solely dependent on the optimal complementarity of their structures at the binding sites, resulting in a perfect fit. This concept was later refuted by the observation that individual proteins adopted dynamic structures that oscillated when exposed to, for example, different physiological temperatures. Posteriorly, it was even proved that upon complex association both proteins presented structural changes [9]. There are several physical, chemical and environmental factors that influence the PPIs, but the main contributers for a successful PPI consist on the combination of non-covalent contacts, such as hydrogen bonds, ionic bonds, and van der Waals attractions at specific domains [10]. These domains are considered to be the functional and structural unit of the proteins, ranging of a few to a hundred amino acids long. Some proteins are composed by a single domain, while, the majority, is formed by several domains.

1.2 Motivation

Characterizing and identifying the PPIs is a vital step towards the complete understanding of how different cells work, since they are essential molecular interactions comprising the proteome. Just by using simple connections between proteins the proteome is capable of representing the network of PPIs of a given organism, clearly identifying the communication between proteins. Despite the still existing challenges derived from the inherent bias and its incomplete coverage for most of the living systems [11], a complete proteome comprehension enables the extraction of interaction subnetworks that simulate disease pathways and allows for a better understanding of several disease progressions [12].

Given the PPIs biological significance, it should come as no surprise that their deregulation is one of the main explanations to several disease states. Cervical cancer, leukemia and neurodegenerative diseases [13] are some of the most reverberating examples. Recently, due to the inherent relation of aberrant PPIs and homeostasis corruption, the study of the these interactions as a target to develop novel therapies has been rising in popularity [14], but nevertheless, the identification and understanding of the whole PPIs must be accomplished before thinking about designing potential inhibitors and modulators of specific PPIs.

This field of study has been under an explosive development and new laboratory techniques have been established to try to accurately experimentally identify new PPIs [15]. Despite all the efforts, the time and cost required to perform many of these methods is unreasonable, adding to this their coverage of the whole PPI network is still minimal. The expanding availability of public databases regarding PPIs, conjugated with the existing inefficient techniques, propelled the development of robust *in silico* prediction methods, as a complement to the existing approaches, or even as independent techniques. Notwithstanding the imminent progress, there are still limitations, which leads to the continuous publication of innovative approaches.

Once the interaction process is completely unraveled, one of the main applications achievable in the short run with these techniques, consists in developing the ability to design new synthetic PPIs, that perform desirable functions, by the modification of a single protein [16]. In the longer run it is expected to create completely new proteins from scratch that interact with selected candidates [10], stimulating the use of proteins as drugs to, for example, inhibit another malfunctioning protein.

1.3 Objectives

Technology has revolutionized our world, and also how investigation is done, paving the way for an exponential development of several computational methods designed to predict PPIs. These methods base themselves in the use of distinct protein features, ranging from protein structure, domain data, protein sequence, evolutionary information and several other types of protein data. From all these types of data, the sequence is the most available and reliable one, due to its current bigger coverage of the whole proteome [17, 18].

The challenge of predicting PPIs with solely the involved protein sequences is interesting and still far from completed, so the main ambition of this masters thesis is to reliably and efficiently predict PPIs, using a deep learning model that only resorts to the primary structure of the proteins. The work involved can be broken down into the following steps:

- Assemble a reliable and not biased dataset of PPIs;
- Assess the reliability of the created dataset when compared with benchmark datasets;
- Explore deep learning algorithms, more specifically convolutional neural networks, to predict PPIs, using, exclusively, amino acids sequences;
- Analyze the influence of distinct model architectures, which contrast in the type of layers used;
- Apply a hyperparameter tuning technique to identify the best parameters combination;
- Evaluate and compare the proposed models, as well as the existing solutions for this problem.

1.4 Document Structure

The document is organized in 7 chapters. The **chapter 1** is the Introduction in which some biological background is presented to better comprehend what a PPI is and how important it is. The **chapter 2** is the State of the Art, and it displays several existing techniques that aim to solve the PPI prediction problem, as well as their advantages and disadvantages. **Chapter 3**, Data, describes the methods integrated to create the PPIs datasets and the strategies used to produce numerical representations of the proteins sequences. In **chapter 4**, Model design, a comprehensive description of the neural networks architectures considered in this work is introduced. **Chapter 5** is Hyperparameters, in which the several hyperparameters used to create the Deep Learning (DL) models are briefly described. **Chapter 6**, Results and Discussion, displays the results obtained, as well as a thorough explanatian and framing of why those results were achieved. Finally, **chapter 7** is the Conclusion, in which the main ideas of the work are accentuated and some plausible future work approaches, that go beyond the confines of this thesis are suggested.

2

State of the art

For the experimental determination of the PPIs the two most widely used high-throughput techniques are: two-hybrid screening and tandem affinity purification coupled with mass spectometry. The first approach is considered a binary method, since it determines the direct interactions between protein pairs, while the latter is regarded as a co-complex method, because it is able to identify interactions between larger groups of proteins [6]. The two-hybrid screening technique explores the fact that the transcription factors of most eukaryotic organisms are capable of being split into two fragments, the activation domain and the DNA binding domain, but still can get activated and produce a reporter gene if they are indirectly connected. Each protein is then connected to one of those fragments, in which the protein connected to the DNA binding domain is the bait and the other the prey. In case they interact the transcription of the reporter gene is verified, otherwise it will not happen [19]. The gold standard of co-complex methods, the tandem affinity purification coupled with mass spectometry, is divided into two separate phases as the name suggests. During the tandem affinity purification phase, an individual protein, tagged as *bait*, is used to fish a group of interacting proteins, the *preys*. Successively, each one of the *prey* proteins fished out is separated and posteriorly identified by mass spectometry. [20].

However, a big fraction of the results obtained from these large-scale methods are often noisy and contraditory, because of the high false positive and false negative interaction rates of some techniques [21]. On top of that, their coverage of the whole proteome is still fairly limited [22], while being excessively time and cost consuming. All these drawbacks motivated the development of computational methods to bridge the gaps of experimental methods, with the ultimate goal of progressing towards a more accurate and efficient prediction of PPIs. The sudden and recent increase of interest in this field encouraged the flourishment of several successful computational approaches. The majority of the existing methods can be grouped as simulation-based or machine learning-based approaches [23].

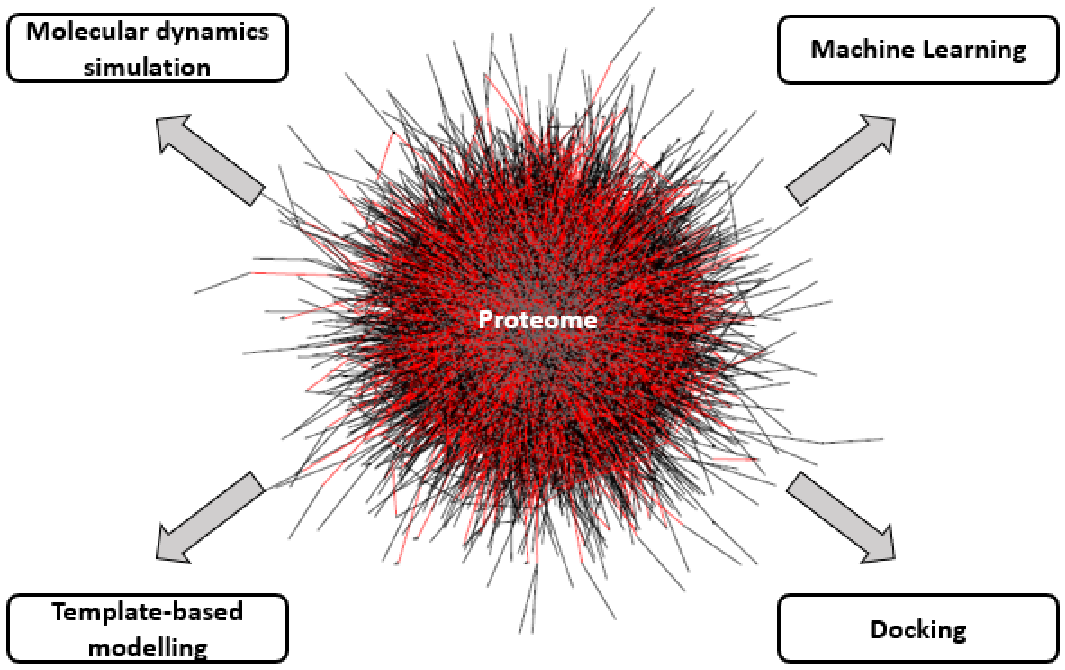


Figure 2.1: Conceptual view of the computational methods to predict PPI.

2.1 Simulation-based methods

Determining the structures of the complexes the proteins adopt when they interact with each other, along with the association mechanisms involved is fundamental. The simulation-based techniques make an effort to understand the molecular details of such interactions, formulating the kinetics of the binding process of the two proteins at the atomic level. However, the major drawbacks of these methods are their excessive computational cost and the still existing difficulty of obtaining the atomic-detailed necessary data.

2.1.1 Molecular dynamics simulation

Molecular dynamics (MD) simulation is a computational approach which aspires to portray the structure and the association dynamics of the complex formed during a PPI. Ideally, these simulations are capable of characterizing the process behind the interaction of two separate proteins, by simulating the dynamic behaviour of the proteins as a function of time. From an initial structure of the system, acquired from either experimental or comparative techniques, the simulation of the dynamics evolution of the proteins is achievable, by integrating the acting forces, also called force fields, which are responsible for the physical movements of the particles. The incorporated Newton equations calculate two essential parameters of the constantly changing position of the particles, the acceleration and velocity. From these, the movement of the particles is numerically deduced by integration. The time step used is usually very small, normally in the femtoseconds. The small time step implemented is one of the limiting factors that hinders the use of these techniques in large scale experiments, due to their necessary massive number of iterations [24]. The accuracy of these methods is strongly dependent on the precision and detail of the force fields.

Bastianelli et al. [25] proposed a theory to better comprehend the process behind a small protein of 33 amino acids long, Psalmopeotoxin I (PcFK1), and its capability to inhibit the growth of *Plasmodium falciparum* parasites. The hypothesis that PcFK1 binds and inhibits subtilisin-like serine protease PfSUB1 was examined. PfSUB1 is a protease responsible for the maturation of parasite proteins required for a successful invasion of the host erythrocytes. The premise plausibility was investigated with the help of MD simulations, more specifically with AMBER 9 software. The PcFK1 binding and inhibition capability was a critical finding, since the *Plasmodium falciparum* parasites are the main cause of malaria in humans and are becoming resistant to multiple drugs.

The MD simulation of a desired system can be accomplished with different levels of detail. An atomic representation is the one that leads to the best results, as it is the closest approximation to the actual system, however, if the systems are too complex or require long simulation times, coarse-grained representation is, usually, a viable choice [26]. This approach is characterized by using groups of atoms as its fundamental unit, consequently reducing the number of degrees of freedom, leading to a lower computational cost. On the other hand the results from this type of representation should be meticulously examined and questioned, due to its underlying simplifications. Ravikumar et al. [27] presented a coarse-grained simulation, based on available crystal structures of individual proteins, that analyzes five possible complexes and their conformations stabilities.

In spite of all the MD simulation limitations, derived from the several approximations implemented, they are one of the only tools that provide insight at an atomic level, even if limited. The detailed resolution provided by this technique, when combined with other computational approaches may produce, in the future, a powerful algorithm to predict and better understand the mechanism behind PPIs.

2.1.2 Docking strategy

Docking simulations aim to predict the most stable 3D structure of the complex formed from two unbonded interacting protein structures. The availability of accurate protein structures is a prerequisite, which can be determined by several experimental methods, such as x-ray crystallography and nuclear magnetic resonance spectroscopy.

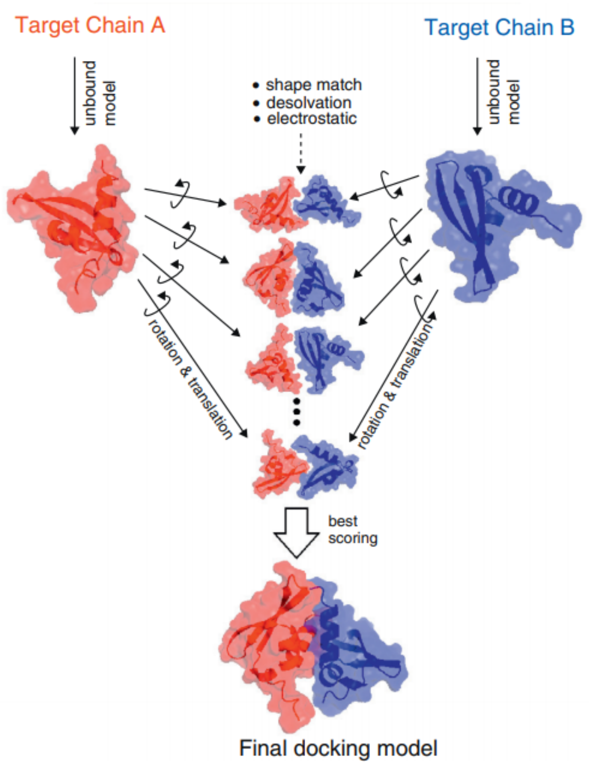


Figure 2.2: Simple illustration of the docking strategy between two proteins, highlighting both the search algorithm as well as the scoring function [28].

Docking strategies are divided into two main phases: the search algorithm and the scoring functions, similar to Figure 2.2. The search algorithm samples all possible binding orientations and conformations between the proteins. The sampling procedure considers the proteins as either rigid, semi-flexible or flexible entities [29]. For the rigid approach both proteins are static structures and only the translational and rotational variables are taken into consideration, a similar approach to the lock and key model [9], where the binding affinity is proportional to the geometric fit of both. Meanwhile, a semi-flexible approach considers one of the proteins as a flexible entity, allowing for conformational changes, while the flexible sampling recognizes both proteins as flexible structures, which is a closer representation of the biological binding process, since proteins are not static entities during binding.

The latter is clearly the most computational demanding approach, due to its higher number of degrees of freedom involved, which is the consequent price to pay for such a thorough approach. The scoring function is responsible for assigning a score to every one of the complexes instantiated during the sampling phase, enabling the discrimination between correctly and incorrectly docked proteins. The score is influenced by various factors, from the binding free energy, the force-fields, to even the shape complementarity of the individual proteins. However, in order to avoid the huge computational cost of calculating a near perfect scoring function, most of the docking algorithms are satisfied with simplifications. As far as docking algorithms goes, ZDock [30] and RosettaDock [31] are some of the most popular ones.

Wass et al. [32] revealed that the docking scores are an effective parameter to determine whether two proteins interact or not, since the scores generated by interacting pairs are distinguishable from the scores produced by non-interacting pairs. This was achievable by comparing the docking process of a protein with its previously known interactors against the docking of the same protein with non-interacting partners.

Matsuzaki et al. [33] established a prediction model that incorporates a rigid sampling docking technique and takes as input a set of 3D structures of proteins and predicts, out of all the binary combinations, the ones that integrate possible pairs, after eliminating the energetically unlikely interactions. For the docking technique two possible algorithms were explored, ZDock and MEGADOCK, since the main difference between them is the scoring function incorporated, after all ZDock uses a more complex, but more time consuming function. In this specific approach, the prediction model was tested with data from a bacterial chemotaxis signaling pathway.

Dong et al. [34] analyzed the prediction ability of a docking algorithm by evaluating all the attainable binary pairs that are created from the high quality structures available for the *Arabidopsis*. The top ranked interactions were, posteriorly analyzed in order to assess which matched the experimentally determined ones available in public data sets. The docking algorithm implemented was HEX, which incorporates a rigid sampling technique. Finally as a validation tool, yeast two-hybrid experiments were applied in randomly chosen pairs to vindicate the reliability and accuracy of the top ranked interactions.

The incorporation of structure information to predict PPIs makes a lot of sense, however the current lack of information and its excessive computational cost makes these strategies unreliable. Much more so, when integrated in large scale experiments to predict PPIs. This limits the investigators to implement simple strategies, predominantly rigid body sampling techniques, which do not perfectly replicate the biological process, leading to biased results towards proteins with perfect geometric complementarity.

2.1.3 Template-based modeling

With the intention of decreasing the computational cost of the docking strategies, template-based approaches were developed. The main difference between the two techniques is that in a docking strategy both proteins are initially unbonded and their assembled structure is predicted, through an extensive search of different orientations. Meanwhile, in a template-based approach the unknown complex is derived from related protein complexes whose structure has already been experimentally determined [28]. This technique is based on the idea that evolutionary structure information can be used to model protein-protein complexes, due to the structure and sequence conservation between complexes, as homologous proteins pairs present similar binding interfaces.

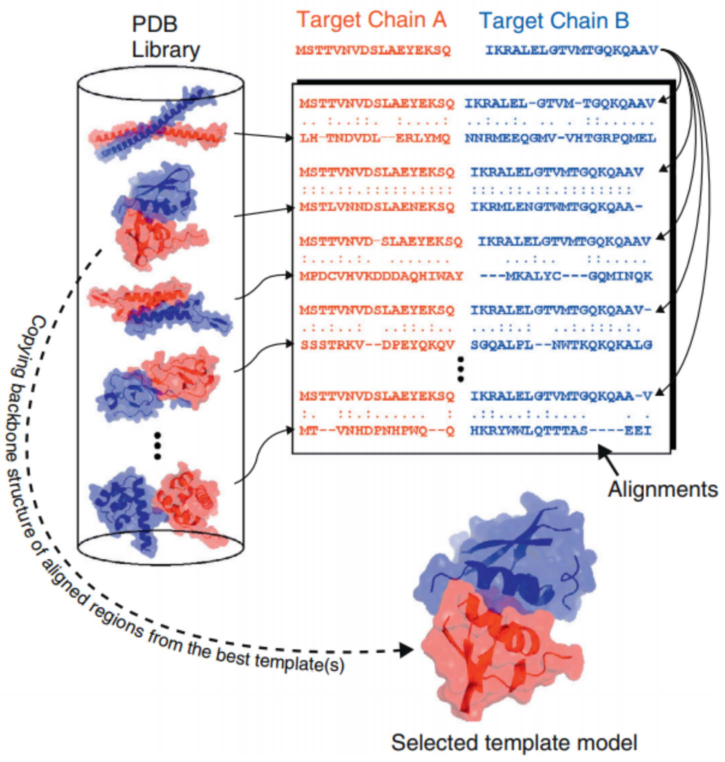

PDB Library
Target Chain A
Target Chain B
MSTTVNVDSLAEYEKSQ
IKRALELGTVMTGQKQAAV
MSTTVNVDSLAEYEKSQ IKRALEL-GTVM-TGQKQAAV
LH-TNDVDL--ERLYMQ NNRMEEQGMV-VHTGRPQMEL
MSTTVNVDSLAEYEKSQ IKRALELGTVMTGQKQAAV
MSTLVNNDSLAENEKSQ IKRMLENGTWMTGQKQAA-
MSTTVNVD-SLAEYEKSQ IKRALELGTVMTGQKQAAV
MPDCVHVKDDDAQHIWAY ---MKALYC---GQMINQK
MSTTVNVDSLAEYEKSQ IKRALELGTVMTGQKQAAV-
SSSTRKV--DPEYQKQV SGQALPL--NWTKQKQKALG
MSTTVNVDSLAEYEKSQ IKRALELGTVMTGQKQAA-V
MT--VNHDPNHPWQ--Q HKRYWWLQTTTAS----EEI
Alignments
Copying backbone structure of aligned regions from the best template(s)
Selected template model

Figure 2.3: Schematic illustration of the template-based modeling strategy between two proteins [28].

The general methodology behind this strategy consists on preparing a library of appropriate templates, aligning the target with the templates according to a similarity measure, refining and finally scoring [28]. This approach depends heavily on the available templates.

In spite of its lighter computational cost, this technique has seen little to no use when the goal is to predict PPIs. During the assembling process of this technique there is a complete disregard towards any energy and flexibility assessment [35]. The recurring lack of efficient full-length complex structure data, as well as the not straightforward and still to be unveiled correlation between sequence similarity and folding similarity, leads to suspicious results and a low implementation on large-scale studies. There are even reports that most of the PPIs are not preserved by homology, unless ridiculous values of similarity are achieved [36], because after all the interactions do depend on several factors. Template-based modeling is a very promising technique, however it is still waiting for advances concerning a more complete coverage of the proteome and for more confirmations of its reliability.

2.2 Machine learning approaches

New PPIs can be predicted using physical and chemical information of already experimentally established PPIs. The growth of PPIs data availability led to its implementation in computational algorithms, that forge mathematical models based on this data, which translate their generalization ability. These methods are able to extrapolate crucial patterns from the available data, which will assist towards the complete comprehension of the proteome of an organism, by filling the current knowledge gaps. The underlying goal of Machine Learning (ML) is to predict whether new protein pairs interact or not, based on the analysis of previously identified PPIs data and, contrary to the techniques previously mentioned, are large-scale compatible.

Machine learning techniques can be grouped into two categories: supervised and unsupervised [37]. Supervised learning learns a mathematical function by mapping an input to a labeled output, this characteristic allows it to predict the output to different new inputs. On the other hand, unsupervised learning is responsible for finding hidden patterns in unlabelled data, ideally modeling distinguishable clusters over the inputs. The data structure of the PPIs currently available, in which almost every pair of proteins is assigned a binary class that determines whether they interact or not, leads to the inevitable implementation, for the most part, of supervised learning algorithms.

The most relevant ML techniques implemented in this area can be divided, according to their mapping scheme [38], in: Support Vector Machine (SVM), Random Forest (RF) and Neural networks.

2.2.1 Random Forest

RF is a an ensemble learning classification algorithm, as it consists of a combination of individual decision trees. Each decision tree is grown, without pruning, from a random set of the training data, where several decision splits take place, based on a random set of features. These 2 random factors force a higher variation and lower correlation between decision trees. The RF class prediction is dependent on the output of the decision trees, since it combines the prediction of every single one, by using the mode, the most voted class by the trees. This technique is deeply associated with the wisdom of the crowds concept, since a single decision tree is easily prone to overfit, but a combination of those protect each other from their individual disadvantages.

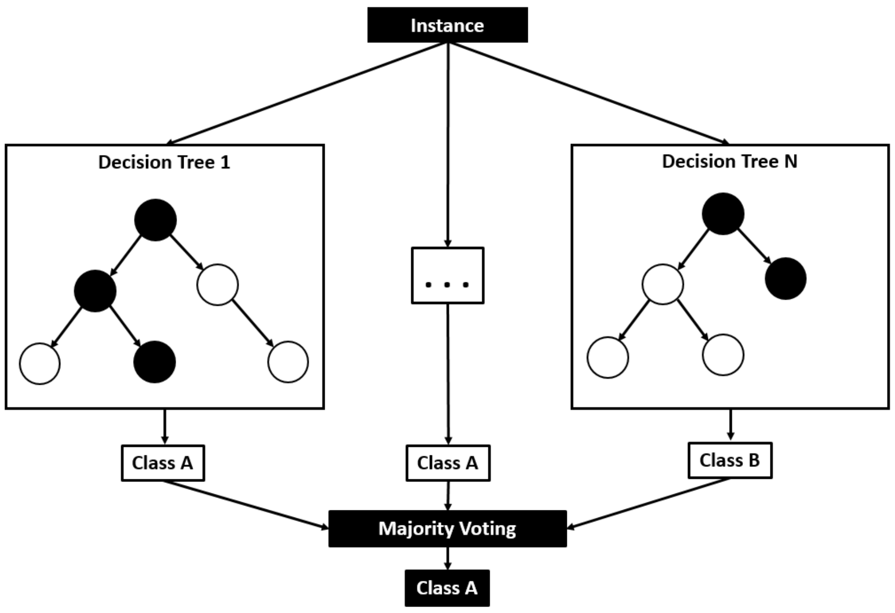


Figure 2.4: Schematic representation of the mechanism behind RF.

Chen and Liu [39] proposed a domain-based RF classsifier. In this approach each protein pair is represented as a fixed size vector with a length equal to the number of unique domains in the data set. Every single element of the vector represents 3 different possibilities: whether a specific domain does not exist in the protein pair, only one of the proteins contain it or if it is present in both. Nevertheless the RF algorithm implemented incorporates a slight twist of the random feature subspace selection. Given that each domain is not equally determinant to the pair interaction, a selection probability is assigned to each domain according to its presence on proteins of the data set considered.

You et al. [40] established a novel technique for protein representation, the Multi-scale Local Descriptor. The results of this technique were used as the input of a RF algorithm in PPI prediction problem. This technique extracts features by taking into consideration that the interactions are materialized in continuous segments with various lengths in the protein sequence.

2.2.2 Support vector machine

SVM is a classification algorithm capable of transforming the input features space into a higher-dimensional one, where a linear separating high-dimension surface is built [41]. This separating surface is the hyper-plane and maximizes the distance between data points of any class, by simultaneously minimizing the true and empirical error rate. As a golden rule, usually, the bigger the margin between data points of contrasting classes, the better is the generalization power of the model. In some cases it is impossible to linearly separate the data points, according to their classes, in the original input space, thus the high-dimensional mapping that makes the separation task easier. This mapping process is called the kernel trick, where a non-linear function, the kernel function, is implemented [42].

Computational methods based on structure information of the protein complexes are not exclusively docking and template-based algorithms, several SVM techniques have also incorporated this type of information. Hue et al. [43] proposed a large-scale compatible SVM model capable of distinguishing between interacting and non-interacting protein pairs. For every pair, the corresponding structural alignment algorithms were computed. The SVM model was compared with other eight ML methods, outperforming every single one of them. A downfall of the model established is that during its data generation phase, every protein pair referred as an interacting one with unknown structures was replaced by a proxy pair of homologs with determined structures.

As mentioned before, there are various types of protein data available that can be integrated in SVM models. Chatterjee et al. [44] presented a SVM technique that assimilates domain information to predict whether two proteins interact or not. The suggested approach is characterized by not disregarding the fact that a single protein can be composed by one or more domains, whereas most of the similar approaches only consider single-domain proteins. From all the domain combinations possible two crucial features were calculated: domain frequency value and protein pair interaction affinity value. The first one is based on the assumption that the higher the number of times that a domain appears in distinct protein pairs, the higher is the likelihood of a protein pair containing that domain to interact. The latter is based on a similar assumption, but considers the occurrence of distinct domain pairs instead of single domains, since there is a higher chance of two domains to interact if there is a high frequency of that specific combination in interacting pairs. The multiple domains considerated in this approach are a huge upgrade, as they are a closer representation of the protein reality and allow for a smoother implementation in high-throughput studies of the whole proteome.

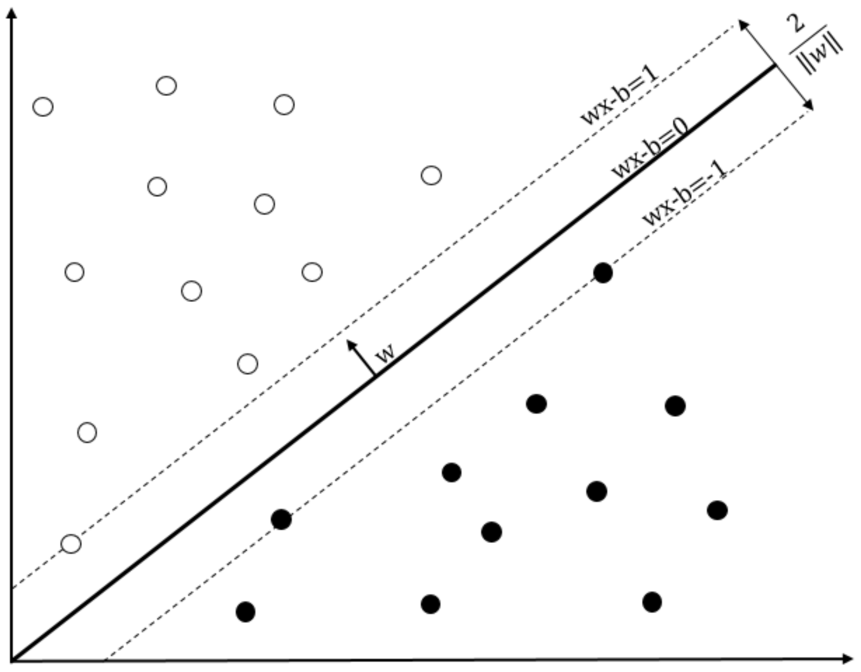


Figure 2.5: Basic demonstration of the SVM algorithm.

Bandyopadhyay and Mallick [45] established an innovative selection of features to represent a protein pair in accordance with their annotated Gene Ontology (GO) terms. GO is a vocabulary of comprehensive terms that comprises knowledge about cellular localization, associated molecular functions and participation on biological processes of a specific protein. Every single term has a different weight, according to its significance for a PPI. A vector, for every protein pair, with the novel GO based features representation was computed and provided to a SVM model to train it. The performance of the algorithm was tested in several data sets of different species.

Even evolutionary information has been integrated in PPIs prediction models. Craig and Liao [46] developed a SVM model based on phylogenetic protein profiles. A phylogenetic protein profile is a binary vector that encodes the presence or the absence of a protein in specific organisms, with, respectively, ones or zeros. Functionally linked protein tend to either be eliminated or preserved in different species. This method is based on the assumption that proteins with similar phylogenetic trees, which portrays the whole picture of the phylogenetic history of a protein in several genomes, have higher chances of interacting. Every phylogenetic tree can be converted in a distance matrix, using distance metrics and then transformed into fixed size vectors that once concatenated comprise the proteins pairs.

Most of the types of protein data incorporated in the models presented so far are scarce, as a consequence of the still incomplete coverage of the whole proteome. On the other hand, protein sequences are an exception, as a method based solely on this data may be virtually considered applicable to all proteins. Guo et al. [47] assumed that the data extracted from proteins sequences is sufficient to predict PPIs. Proteins are not displayed as continuous sequences in nature, thanks to the several levels of foldings they can undergo. Due to their folded conformation, binding sites between interacting proteins may occur in discontinuous segments of the sequence. The proposed method incorporates SVM and Auto Covariance (AC), which is a protein representation technique that takes into account information between distant amino acids in the sequence. To each protein was assigned an AC vector and the protein pair was represented by the concatenation of both. This study came to become a reference to any sequence based algorithm, due to its groundbreaking results at the time and even for its dataset, that is now considered one of the staple benchmarks.

Shen et al. [48] also presented a SVM algorithm, but now incorporated with a conjoint triad feature extraction technique that assigned every single one of the 20 amino acids into one of seven classes, which group the amino acids according to their physicochemical properties. Once the substitution took place, any three continuous amino acids were considered as a unit and its frequency a feature.

2.2.3 Neural Networks

Typical ML techniques, like the ones mentioned in Sections 2.2.1 and 2.2.2, are characterized by combining elaborate descriptors algorithms to gather crucial information, from raw data. These elaborate techniques end in simple vector representations. However, DL, which is a subfield of ML, is computationally capable of processing data in its raw form and from it discover essential patterns for the classification task [49]. DL techniques, apart from their classification capability, also incorporate an automatic features extraction tool, that induce a comprehensive big demand for training data, in order to accomplish their laborious tasks.

DL methods are based on a layered structure of Artificial Neural Networks (ANN), which is a computational algorithm inspired by the way the brain of the animals processes information and therefore presents a theoretically more capable learning process than the conventional ML methods. The brain is composed by several neurons, which are the fundamental units of the nervous system and are responsible for receiving signals, processing them and if needed signalling other neurons. The neurons are interconnected in elaborate linked networks, through synapses, which allow for the transmission of signals between neurons [50]. ANNs strive to simulate the brain organization and learn in a humanlike manner. ANNs are composed by artificial neurons and the synapses are neither more nor less than just weights that represent the relative importance and provide connections between the output of a neuron and the input of another. Similar to the human brain, where different sections of the brain are responsible for processing different parts of information, the ANNs comprise several layers that are responsible for distinct processing tasks: input, hidden and output layers.

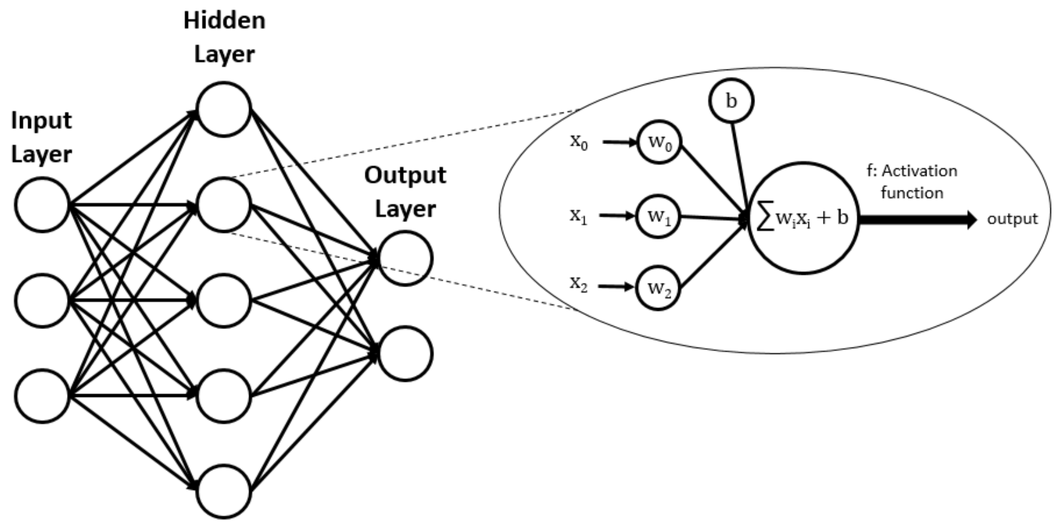


Figure 2.6: A representative scheme of an ANN and the mechanism behind a neuron.

In an ANN each element of the input is fed to a neuron of the input layer, which are then connected to the hidden layer. It is important to highlight that a given neuron may have multiple output and input connections. To each connection a weight is assigned and the input of a specific neuron can be defined as the weighted sum of the outputs given by the predecessor connected neurons, as it is demonstrated in Figure 2.6. Every single neuron has a bias term associated with it and adds it to the input calculated. Once the weighted sum is calculated and the bias added, the resulting values of different neurons may vary a lot, so it is necessary to pass the values through an activation function that is capable of limiting their range and determining whether a particular neuron is activated or not. The activated neurons establish connections with the following layer. This process is repeated and in this manner the data is propagated through the network, in an operation called forward propagation.

During the training process, along with the input, the output is also fed to the ANN and the predicted output is compared with the correct one to compute the error. The amplitude of the error indicates performance-wise how far the ANN still is and also determines the direction and the magnitude of change needed to reduce the error. This information is then propagated backwards, in what is called the back-propagation, and the weights of the connections are adjusted. A cycle of forward propagation and backpropagation is iteratively performed with various inputs until the error computed by the neural network is minimum. ANNs can include multiple hidden layers, depending on the complexity of the problem, which allows the neural network to get deeper and deeper and express more complex features. The architectures with multiple hidden layers are called deep neural networks [51].

ANNs are the foundation of DL and they motivated the development of distinct neural networks, according to the various architectures and types of hidden layers that are implemented. Wang et al. [52] established a deep neural network coupled with a novel protein representation that is capable of taking in consideration that proteins are folded structures and their binding sites might depend on spatially close, but at the same time sequentially distant amino acids in the protein sequences. They implemented a fully connected neural network and, as the name suggests, it is characterized by connecting every single neuron of a layer to every neuron of the following one. Different representations were tested to assess which produced the best results.

Zhang et al. [53] proposed a model, that, contrary to most of the existing methods, attempted to incorporate several sequence-based protein representations techniques.

For each representation technique an individual fully connected neural network was built and trained. Afterwards, the output of multiple neural networks was aggregated and fed to an ensemble fully connected neural network with 2 hidden layers.

Sun et al. [54] presented a deep neural network model, where an autoencoder is implemented to extract low-level features from protein sequences. Those features are later fed to a single output layer to predict PPIs. An autoencoder is an unsupervised learning algorithm responsible for feature reduction and reconstruction. It comprises an encoding phase, where the input is compressed into low-dimensional and low-noise features, followed by a decoding phase, where the original input is restored and the neural network architecture is symmetrical to the encoding phase. The autoencoder requires an individual training phase, where the output fed to the model is equal to the input and it tries to learn efficient ways to encode the data. Once the optimal parameters are determined and the output is as close as possible to the input, the feature reduction capability of the encoding phase is taken advantage of and is used to obtain the more relevant features.

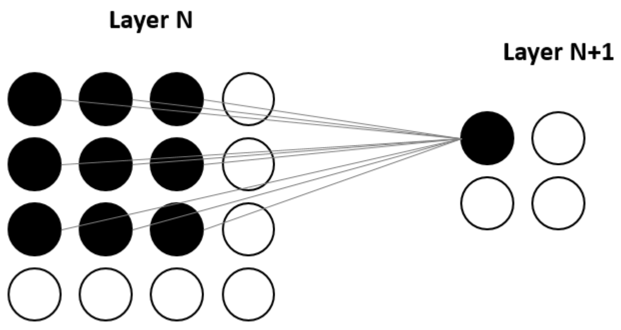


Figure 2.7: A representative scheme of the local receptive field for one neuron

Despite their inherent differences, DL and ML techniques have been combined to create methods capable of predicting PPIs. Wang et al. [55] established a model that consists of a Convolution Neural Network (CNN) as the deep feature extractor and is topped off with an ensemble classifier. a CNN is a special kind of neural network, inspired in the organization of the animal visual cortex and is able to comprehend locally correlated data. CNN takes a different approach from fully connected neural networks, as only a small region of the neurons are connected to one neuron of the following layer, as illustrated in Figure 2.7. This small region is the local receptive field for the one neuron it is connected to. The local receptive field is then slid across the entire input and connected to a different neuron of the following layer, which makes it capable of perceiving complex patterns using small regions.

3

Data

3.1 Main PPI datasets

DL techniques are increasingly cementing their place as the main classification methods to predict PPIs, due to their revolutionary results coupled with the significant improvement of computational power, since the training of such algorithms is extensive and laborious. DL scales with the amount of data available, as more data is fed to the algorithm, more accurately it learns to distinguish between the desired classes, naturally increasing its performance. The number of PPI data available is not considered an obstacle, given that a vast number of PPIs have been experimentally determined and are documented in several databases, such as IntAct [56], BioGRID [57], DIP [58] and STRING [59].

Several algorithms were developed to predict PPIs, which led to the publication of several independent PPIs datasets. This high number of distinct datasets hinders the comparison task between state-of-the-art algorithms, considering that different algorithms trained in different datasets, develop distinct learning abilities. The performance of a model is also heavily dependent on the quality of the PPIs datasets incorporated. If the algorithms are built upon low-quality datasets, it is only expected their predictive capabilities are somewhat debatable, even if they are capable of achieving groundbreaking results on these low-quality datasets. Park [60] highlighted some state-of-the-art algorithms that perform very differently when tested in separated datasets. Depending on its building process some of the benchmark datasets present a biased distribution of PPIs [61], which artificially inflates the performance of the models developed.

Theoretically, the perfect dataset includes experimentally determined interacting protein pairs, also called positive interaction, as well as experimentally proved non-interacting protein pairs, the negative interactions. However, databases describing experimentally determined negative interactions are practically non-existent, given that, these are usually considered failed experiments and are then left out of the public databases. Not disregarding the significance of the positive interactions, but the negative interactions are also essential, not only to guarantee that the ML algorithms are not misled during their learning phase, but also to avoid the repetition of experiments to evaluate whether two already tested proteins interact or not, which, inevitably, accelerates the prediction process by focusing resources on proteins with higher chances of interacting. Regardless, in order to create a biological representative and reliable dataset that precisely simulates the scenario of a protein in a cell and is large-scale compatible, negative interactions have then to be computationally generated.

There are different computational techniques to generate negative interactions. Trabuco et al. [62] established a technique that identifies, using viability analysis, the negative interactions that might have been tested in two-hybrid screens but not documented. The process consists on gathering positive interactions derived from two-hybrid experiments and classifying each protein as viable baits (VB), if a protein behaves as a bait in at least one interaction, and as viable preys (VP) if it behaves as a prey in at least one interaction. For the cases where a protein only acts as a bait in all the interactions it is present in, they do not stop being classified as VB but are further stratified and subclassified in viable bait only (VBO) proteins, a similar explanation is used for the viable prey only proteins (VPO). Afterwards any connection in their graph representation between a VP and a VB that is not already determined as a positive interaction is then considered a negative interaction. Links between VBO and VPO are settled as untested and are not integrated in datasets.

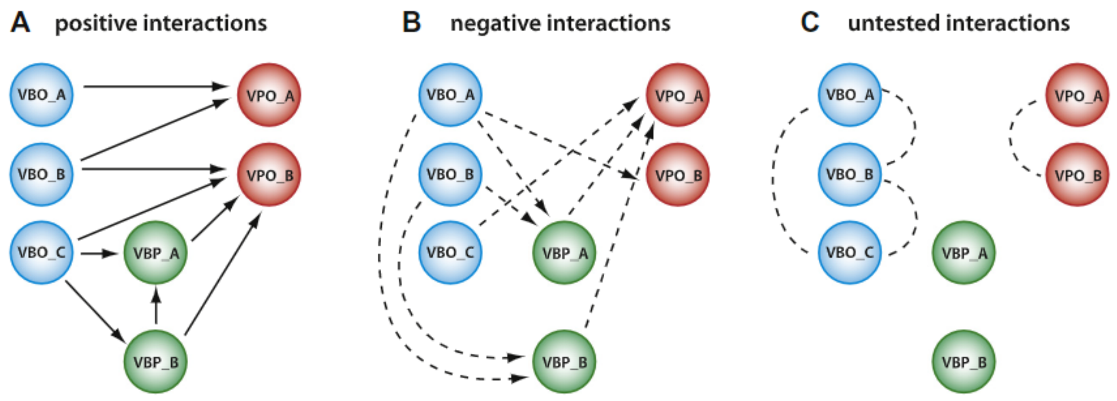


Figure 3.1: Demonstration of the positive, negative and untested interactions. The VBP acronym represents simultaneously a viable prey or a viable bait protein [62].

Although the improvements achieved with this technique, it is also faulty. First of all, because of the incompleteness of most proteomes it does not take into account several possible combinations. Additionally, in BioGrid, which will be the selected database from which the positive interactions will be collected, there are 18 distinct experimental techniques to identify physical PPIs. From all the available interactions, only around 10% are determined with two-hybrid experiments, which subsequently severely limits the generation of negative interactions.

More recently, Zhang et al. [63] proposed two techniques to computationally produce the negative interactions. The first one is based on sequence similarity, in which the most dissimilar protein pairs are selected as non-interacting pairs. The basic assumption behind this technique is that considering a validated PPI between A and B, then if a protein C is dissimilar to A it has a very low probability of interacting with B. However, sequence similarity alone is not solid enough to define a protein pair as interacting or not, because sequentially similar proteins may adopt completely different structures. There are also evidences [36] that question the viability of this approach, since PPIs are rarely conserved. The second approach defined by Zhang et al. [63] defends that the farther the proteins are in a PPI graph representation, the more likely they are to not interact. This technique is based on the triadic closure principle, a principle originated from social network analysis, which comprises neighborhood based similarity and states that proteins that share multiple interactors have a higher chance of interacting, just like individuals with multiple common friends are very likely to know each other very well and to get along. Nonetheless, a study [64] demonstrated that this principle fails for most proteins.

Most of the existing datasets incorporate an easy to understand non co-localization technique to generate the negative examples, which conceives as a negative interaction a random protein pair whose individual proteins are associated with distinct cellular localization annotations [65]. The technique is founded on the premise that proteins discovered in different sub-cellular compartments are unlikely to interact, due to the obvious physical constraints. Nevertheless, this non co-localization approach can lead to biased representations of the proteome and, consequently, overoptimistic estimates of the accuracy [66].

Some of the work developed in this master thesis included the exploration of a novel technique to generate unbiased negative examples, which allows the creation of an unbiased dataset that closely resembles the distribution of the proteome of an organism. Contrary to some of the existent approaches that are faced with inherent limitations and strongly restrict the distribution of the negative interactions on the dataset. In order to compare the reliability of the different datasets considered and their influence on the robustness of the prediction tasks, an unbiased dataset was built from scratch. Also, two benchmark datasets [67, 68] were analyzed.

3.1.1 Unbiased dataset

A careful and thorough approach was vital to create a brand new dataset, as the learning potential of a DL and ML techniques is strongly dependent on the quality of the dataset. In order to build a well grounded, biological representative dataset not only the negative interactions, but also the positive ones have to be meticulously chosen, since a fraction of the experimentally determined interactions are reported false positives [54], as a consequence of errors and bias of some experimental methods [69].

BioGRID is a database comprised of curated interactions from literature. All the human multi-validated physical PPIs available in BioGRID were selected. Physical interactions are obtained through laboratory techniques that are specialized in determining whether exists a direct physical binding between proteins or not. The interactions considered for the unbiased dataset were only the multi-validated ones, which consist of interactions simultaneously published in several publications and identified by various experimental systems. This multi-validation scan is an essential step to limit the implementation of possible false positives, since an interaction that is determined by multiple publications and experimental techniques, shows a higher coherence and a higher probability of actually being a true positive. In the end a dataset solely composed by multi-validated PPIs has a extremely low contamination potential with plausible false positives.

Considering an interacting protein pair composed by a protein A and a protein B it can simultaneously be documented in several papers, as a result of the different techniques used to detect it. However, the interaction itself can either be published as A interacts with B or B interacts with A, according to the way the experimental technique is set up. This way of documenting the interactions grants directionality to them. For the classification task the directionality is completely irrelevant and can be even considered as noise. For this reason duplicate interactions and reciprocal interactions were identified and removed from the ones gathered from BioGRID.

Beyond the already described techniques to generate negative interactions, along with the non co-localization technique, another very popular approach consists on randomly sampling negative protein pairs [66]. However, the whole proteome coverage of the human is still not complete, therefore simply randomly sampling two distinct proteins and consider them as a negative interaction if the pair is not already a positive interaction may lead to the contamination of the negative set with some potential positive interactions, that have not yet been experimentally determined. One can say that since the ratio of negative to positive interactions is expected to be disproportionate, in the hundreds to thousands ratio [70], the actual contamination is minimal.

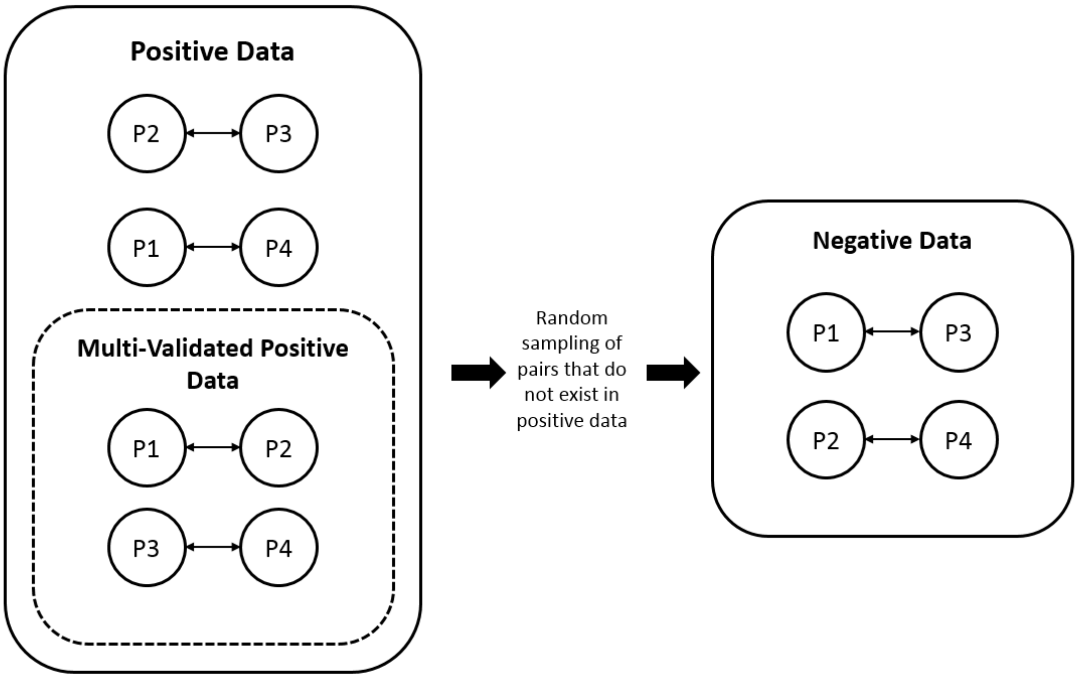

Positive Data
P2
P3
P1
P4
Multi-Validated Positive Data
P1
P2
P3
P4
Random sampling of pairs that do not exist in positive data
Negative Data
P1
P3
P2
P4

Figure 3.2: A simple simulation of the negative sampling method.

Nonetheless, the main goal behind the creation of this unbiased dataset was to build a biological representative dataset with the smallest contamination percentage possible. To achieve that, the negative random sampling method was implemented with a slight twist. The approach consists on identifying the unique proteins from the multi-validated positive interactions and put them in the set S of proteins. Afterwards, two distinct proteins were randomly sampled from S. Finally, the pair sampled was compared with the multi-validated and not multi-validated positive interactions and if it was not already labeled as one of them, it was considered a new negative interaction. By only sampling negative pairs from S, we guarantee that almost every protein considered is present in at least a positive and in a negative interaction, which contributes to a sparser distribution of the interactions. This simple twist yielded an even smaller contamination of the negative set, if even existent, as protein pairs with the slight evidence of potentially being considered positives interactions were taken into account and not constituted a negative interaction. The process is briefly illustrated in Figure 3.2. This method presents an unbiased estimate of the true distribution of the negative interactions on the whole proteome.

Deciphering the number of negative interactions that should be generated, was another concern that had to be clarified. In other words, it consisted in defying if the dataset should have been balanced or unbalanced to replicate the ratio of negative to positive interactions of the whole proteome. Notwithstanding what has just been said the negative interactions are randomly sampled and not experimentally determined, whereas the multi-validated positive interactions can be considered high-quality data. Additionally, there is still no consensus on the ratio of negative to positive interactions, then the wisest choice seemed to create a balanced dataset. In Table 3.1 is presented a synopsis of the unique proteins and PPIs of each type of interactions.

Table 3.1: Dimensions of the unbiased dataset

	Positive data	**Negative data**
Number of unique proteins	8073	7205
Number of unique PPIs	36236	36236

3.1.2 Pan et al. dataset

It is a benchmark dataset exclusively composed by human PPIs, whose positive interactions were obtained from the Human Protein References Database [71] and all the replicate interactions and self-interactions were identified and removed from it. Afterwards the negative interactions were computationally generated by the non co-localization method, where two randomly paired proteins found on distinct subcellular compartments are considered as non-interacting, if that pair is not identified in the positive data. Fistly, various human proteins were gathered from UniProt database [72] to produce the negative examples, which comprehended 6 individual cellular localizations, namely the cytoplasm, nucleus, endoplasmic reticulum, golgi apparatus, lysosome, and mitochondrion. Proteins with dubious cellular locations, as well as proteins with multiple locations were filtered out. Moreover, fragment annotated proteins and fifty or less amino acids long proteins were removed. Finally, from this set of proteins the negative interactions were created.

Table 3.2: Dimensions of the Pan et al. dataset

	Positive data	**Negative data**
Number of unique proteins	9476	2184
Number of unique PPIs	36630	36480

3.1.3 Du et al. dataset

To explore if biased datasets from different organisms actually lead to different performances a benchmark *Saccharomyces cerevisiae* PPIs dataset was examined. The positive data was gathered from the Database of Interacting Proteins [73] and was further manipulated, where interactions with at least one of the proteins fifty or less amino acids long were removed. Just as it was described in the previous dataset, in Section 3.1.2, the selection of negative interactions was also accomplished with the non co-localization method. The proteins from which the random pairs were sampled were equally obtained from UniProt, but a more recent version of it. An approximate ratio of 3:1 negative interactions to positive interactions was initially obtained with all the negative data generated, however the desired ratio mentioned by the original work was 1:1, so from the whole negative data an equal amount of pairs was randomly selected, Table 3.3.

Table 3.3: Dimensions of the Du et al. dataset

	Positive data	**Negative data**
Number of unique proteins	4382	1908
Number of unique PPIs	17257	17257

3.2 Protein data

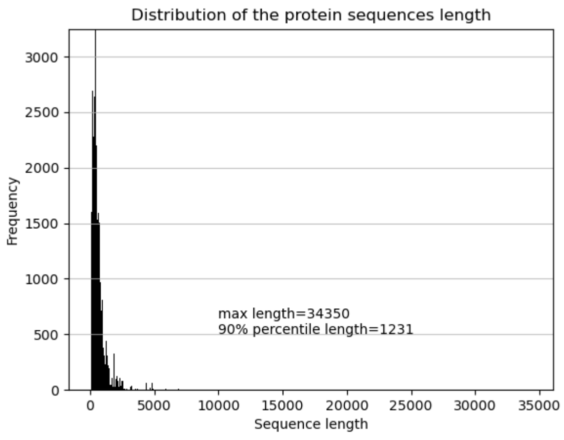
As every other sequence-based DL technique, once the protein pairs data is determined, it is fundamental to map each protein to its respective sequence. For this task the Uniprot [72] was the database of choice. In Uniprot the protein sequences are formed by different combinations of 24 characters, which correspond to the standard 20 amino acids plus 4 different characters for rare and still to determine amino acids. These special 4 characters are very rarely incorporated in sequences and to avoid any possible bias of the classifier towards the presence of these, every protein with any of these 4 characters was removed.

The main purpose of our model is to integrate the complete raw sequence of amino acids in the DL model. The DL model itself expects a fixed size input, regardless of the varying lengths of the protein sequences. A frequent work around to this problem is assigning a common length, usually the length of the longest sequence, and pad every sequence to the maximum length. From observing Figure 3.3 it is easily concluded that the sparse distribution of the sequences length is prone to create noisy vectors, considering the extreme variation between the sequence lengths. Adopting the maximum length as the common length could have even led to an excessive computational cost, as every input of the model would be unnecessarily big. In this regard, we defined the common length of the vectors as the value obtained by the 90 percentile of the distribution of the proteins length of the main datasets [74]. Any protein longer than the determined value was removed from the dataset, and the remaining ones were zero padded to the determined value. Proteins smaller than fifty amino acids long were also removed [68, 67].

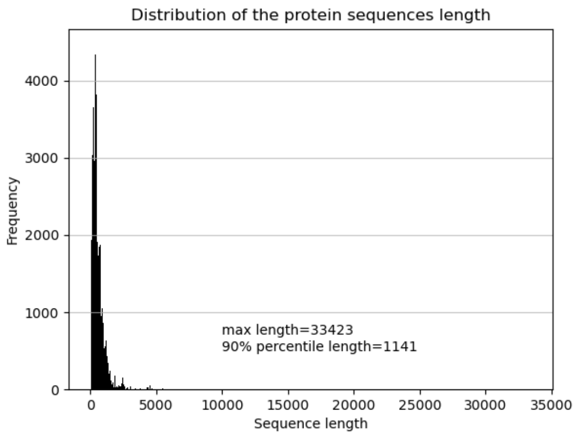
In Figure 3.3 the common length for every main dataset is demonstrated, as well as the length of the bigger protein in that specific dataset. The proteins of the unbiased dataset, Pan et al. dataset and the Du et al. dataset were zero padded to the respective lengths of 1231, 1141 and 1030.

3.3 Data representation

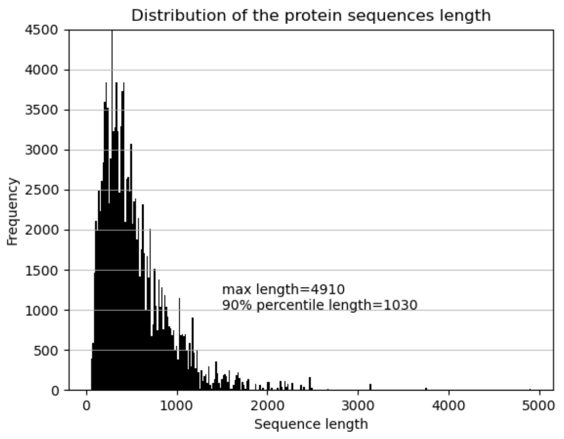
DL methods expect a fixed size vectorized numeric representation of the data, in order to be able to implement their intrinsic operations on data. Different descriptors highlight different properties of protein sequences and produce simpler representations of a protein, by encapsulating vital properties of the sequence. Due to their aggregative capability, some information is lost, as they often disregard the amino acids order in the whole sequence. Furthermore, DL techniques have incorporated features extractors in them, so feeding them the complete protein sequence, where every amino acid is treated as an individual feature, can be beneficial. They are the ones that decide, from the original complete input, which amino acids are relevant for the prediction task and which not. An encoding technique that preserves the raw input sequence and is capable of transforming categorical data in numerical data is crucial and may be a game changer. The similarities between the desire to encode complete proteins sequences and the encoding techniques used in Natural Language



(a)



(b)



(c)

Figure 3.3: Distribution of the sequence length for the three main datasets considered. (a) unbiased dataset , (b) Pan et al. and (c) Du et al. dataset.

Processing inspired the implementation of one-hot encoding.

Table 3.4: Amino acids substitution table

Integer	Amino acid	Integer	Amino acid
1	Cys (C)	11	Val (V)
2	Glu (E)	12	Met (M)
3	Leu (L)	13	His (H)
4	Thr (T)	14	Phe (F)
5	Ile (I)	15	Tyr (Y)
6	Pro (P)	16	Gln (Q)
7	Ala (A)	17	Asn (N)
8	Trp (W)	18	Asp (D)
9	Arg (R)	19	Ser (S)
10	Gly (G)	20	Lys (K)

Before applying the one-hot encoding it was necessary to produce a numerical vector from the amino acids sequence. That was possible by applying the integer encoding technique, where every single character of the sequence is transformed in a number according to Table 3.4. This representation technique alone is insufficient, as it may lead the DL model to recognize the order or hierarchy between different amino acids, given the natural ordered relationship between integers. When encoding a complete protein sequence there is no sense of superiority between amino acids, as the different amino acids are nominal variables and not ordinal variables. So, to avoid this scenario, after obtaining an integer sequence, the actual one-hot encoding technique was implemented, where every integer corresponds to a vector of zeros, except for the bit representing that amino acid that is set to 1. This encoding technique has recently been rising in popularity between protein-based DL models [75, 76, 74].

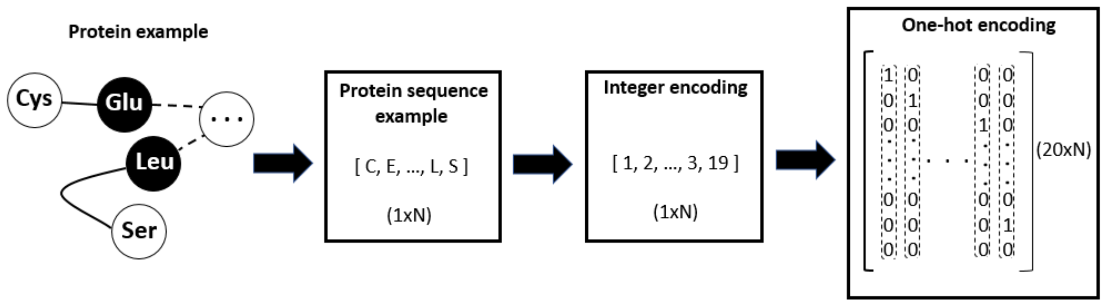


Figure 3.4: An example of the one-hot encoding technique

3.4 Datasets Splitting

DL algorithms need training data, in order to constantly update its parameters. The training data is used to fit the model and learn from it. To evaluate the final performance an uncorrelated set of data must be fed to the model, which is the testing data. If that does not happen, a model is tested in data that it already saw and defined its parameters around, which leads to a biased estimate of the performance. The ultimate goal, when developing a DL model, is to make it able to generalize and perform similarly in different sets of new data. When a model is not capable to develop its generalizability, it performs perfectly on the training data, but once new data is fed to it its performance rapidly declines, due to overfit.

After the elimination of some proteins, there was a consequent elimination of interactions where at least one of its proteins were not mapped to its respective sequence. Afterwards, all the datasets presented were split in a training set and testing set, separately. The resulting main datasets splitting dimensions are described in Table 3.5.

Table 3.5: Number of interactions of the training and testing set for the main datasets.

	Unbiased dataset	**Pan et al. dataset**	**Du et al. dataset**
Training positive data	25365	19860	9756
Training negative data	25365	21821	9744
Testing positive data	10871	8464	4173
Testing negative data	10871	9400	4185

4

Model design

4.1 Layers

The layers are the core building blocks of the neural networks and can be interpreted as the group of neurons operating at a specific depth. The first layer of a DL model is the input layer and it incorporates the input data, in which every neuron is assigned an element of the input. The last layer is the output layer and for binary classification problems it is, generally, simply composed by an individual neuron. The variability of the neural networks lies in the hidden layers and their types, which have the inherent goal of minimizing a cost function by learning more adequate representations of the input. The hidden layers considered in this work are the convolutional, pooling, fully connected and flatten layers, which will be further described in the following subsections.

4.1.1 Convolutional layers

These are the core layers of the proposed architectures and from these it is possible to detect the high-level distinguishing features, due to their ability to capture the spatial dependencies of the protein sequences. The convolutional layers were developed to handle images and are inspired by the organization of the visual cortex, in which individual neurons respond to stimulus of limited regions of the visual field. This restricted area is the local receptive field. The neurons of this layer are connected only to a local region of the neurons of the previous layer, thence the inherent local connectivity. The convolutional layers incorporate a set of learnable filters, and their dimensions define the range of the local receptive field. Every single one of these filters is slid across the width and height of the whole input and the dot product between the weights of the filter and the input is computed. The stride is the parameter that defines the size of the steps a filter is iteratively slid across the input. Each filter activates to a distinct feature and it is the systematic sliding process of one filter across the whole input that leads to the consistent detection of a specific feature anywhere in the input, this property is referred to as translation invariance. The 2 Dimensional (2D) array output, resulting from the filter translation and the convolution operation is the feature map of a specific feature. Considering that the convolutional layers integrate a set of filters, they are capable of learning multiple features in parallel.

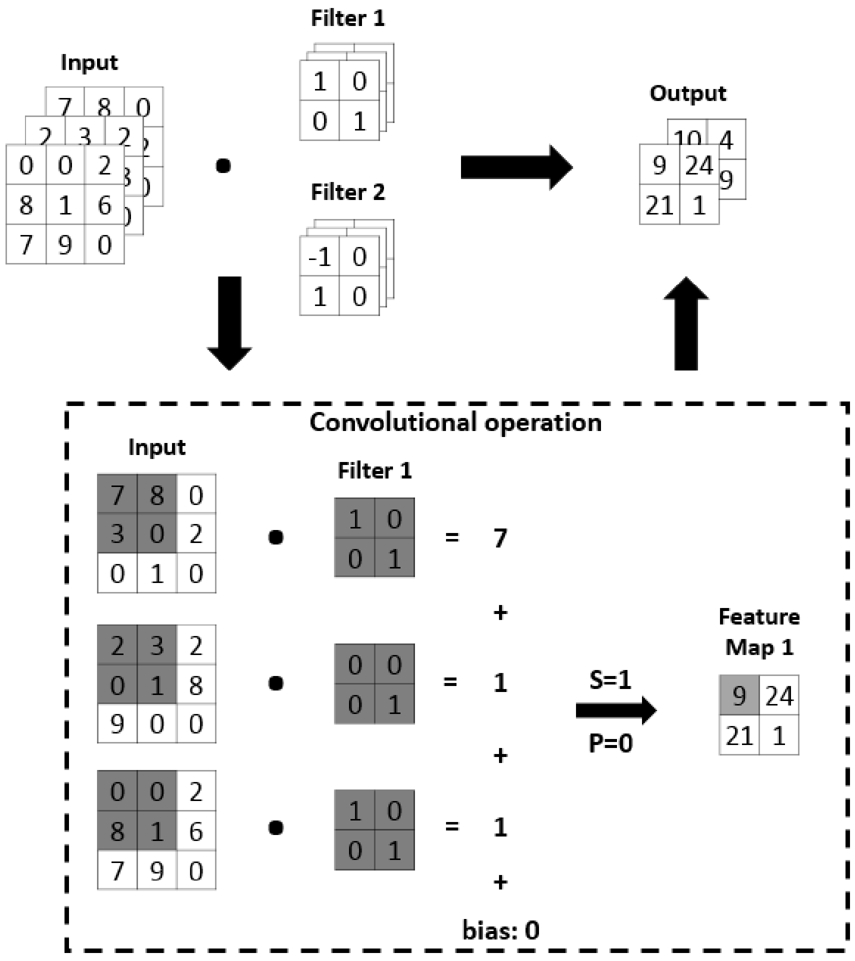


Figure 4.1: Illustration of the mechanisms behind the convolutional layer, with a specific stride, S, and padding, P. For a $3 \times 3 \times 3$ input and two filters of $2 \times 2 \times 3$ dimensions the output obtained, for the displayed stride and no padding is a $2 \times 2 \times 2$ output.

The filter and its dimensions are probably the most significant elements of the convolutional layers. Generally, a convolutional filter is referred to only by its width and height, however depending on the input the filters also have depth. Considering a 3D input of dimensions $width \times height \times depth$ a filter will have $filter\ size \times filter\ size \times depth$ as its dimensions. The filter size is defined by the user and it establishes the area of the local receptive field, on the other hand the depth of the filter always matches the depth of the input. In Figure 4.1 there is an example of the convolution operation. Every depth slice of the input convolves with the corresponding depth slice of the filter and then their output is combined in a specific feature map.

Another parameter that influences the output of a convolutional layer is the padding. The padding allows to control the size of the output, since it avoids the inherent dimensionality reduction of the convolution operation with a specific stride, by zero padding the input until it is guaranteed that the input and the output have the same dimensions.

The convolutional layers arranges their neurons in a 3D conformation, defined by its width, height and depth. The depth is defined by the number of filters that the layer has. Meanwhile the other two dimensions depend on several parameters. The output shape of a convolutional layer is computed by the following formula. Where W represents the width of the input, H the height of the input, K the size of the filter, P the padding used, S the stride and finally N the number of filters of the layer.

$$Output\ shape = \left(\frac{W - K + 2P}{S} + 1, \frac{H - K + 2P}{S} + 1, N\right) \tag{4.1}$$

According to its output shape a convolutional layer has *width* $\times$ *height* $\times$ *depth* neurons. A filter can be interpreted as the set of weights of a specific neuron of the output. Generally, in a neural network every neuron possesses a distinct set of weights, which inevitably leads to a large number of parameters produced by a layer like this. However, in a convolutional layer the neurons placed at the same output depth have a common filter throughout. This means that every neuron in the same depth slice use the same filter. This property is called parameter sharing and is based on the assumption that if a set of weights is useful to detect a specific feature of the receptive field of a neuron, the same weights are useful to detect the same feature on another neuron. This property allows for a lower number of parameters produced and consequently a lower computational cost.

The 2D convolutional layer has been the one described in this section, but as the main purpose of this work is to establish a model that works with protein sequences, 1 Dimensional (1D) convolutional layers were the ones applied. The main differences between a 1D and a 2D convolutional layer lies on the input dimensions and the fact that the first one convolves along one dimension only, while the latter convolves

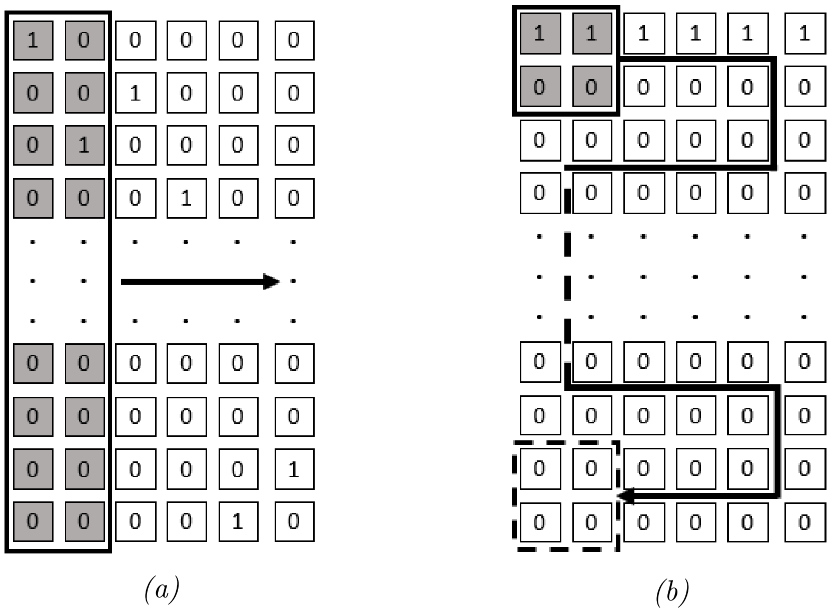


Figure 4.2: Demonstration of the differences between the 1D, (a), and 2D, (b), convolutional process

along the two dimensions of the input, as illustrated in Figure 4.2. After the one hot encoding every amino acid is represented by a full length vector, and the 1D convolution is able to recognize a complete vector as an amino acid and convolve from there, rather than convolving through fractions of the amino acids like the 2D convolutional layer.

One can easily say that the characteristic convolution process of the 1D layers is achievable by defining a filter with proper dimensions, that covers the complete amino acids, however these layers expect inputs with different dimensions. The 1D convolution requires 2D inputs, which match the dimension of the protein sequences after one hot encoding, while the 2D convolution requires 3D inputs. Given the different inputs it is to be expected that the output shapes of the 1D and a 2D convolutional layer are also different. For the 1D layer the output shape is now given by the following formula, in which the output can be interpreted as a stack of 1D data, thence its 2D conformation.

$$Output\ shape = \left(\frac{W - K + 2P}{S} + 1, N\right) \tag{4.2}$$

Apart from the output shape, the convolutional process is similar in both layers, as demonstrated in Figure 4.3. While the 2D layers are more appropriate to handle images, the 1D convolutional layers are more apt to operate with sequences, as less processing has to be done.

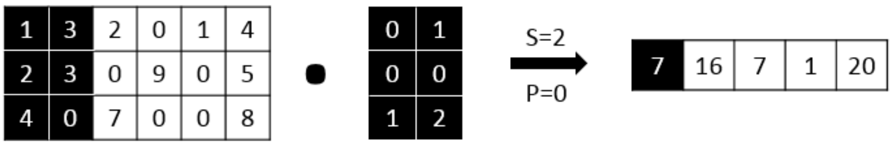


Figure 4.3: An illustration of the 1D convolutional operation.

4.1.2 Pooling layers

The pooling layers are responsible for reducing the complexity of the output of the convolutional layers and even prevent overfitting of the model. This downsamplig technique allows to reduce the dimensionality of each feature map, while preserving the most dominant features. This is an essential layer as it progressively reduces the spatial size of the feature maps, which consequently reduces the number of parameters and the amount of computation needed. Beyond the downsampling capabilities, this layer also works as a noise suppressant.

There are several types of pooling layers, depending on the dimensions desired and the operation implemented to process the reduction. In this master thesis only the 1D max pooling layers were considered. These layers have a pooling window of pre-defined dimensions that is translated across the whole input. The maximum value hovered by the pooling window at each position of the translation is preserved, as it is represented in Figure 4.4.

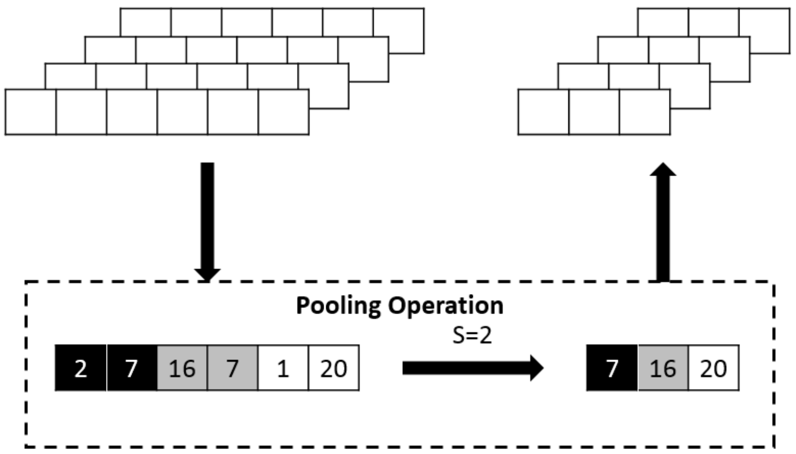


Figure 4.4: Simple illustration of the mechanisms behind the pooling layer.

When a pooling layer is implemented they manage separately the different feature maps. Every single feature map is spatially reduced, but the number of feature maps is preserved. The dimensionality reduction process of a 1D pooling layer is completed according to the following formula. In this case the K represents the size of the pooling-window, which is analogous to the size of the filter of the convolutional layers.

$$Output\ shape = \left(\frac{W-K}{S} + 1, N\right) \tag{4.3}$$

4.1.3 Fully Connected layers

As the name implies in this layers every single neuron connects to every neuron of the following one. In each neuron all the inputs are multiplied by weights that prioritize the most important features and are passed through to the following layer. These layers allow the model to combine the spatial features.

Dropout can be applied to the fully connected layers, which is a regularization technique and consists on arbitrarily deactivating neurons during training, as demonstrated in Figure 4.5, making the training process noisy, in an attempt to make the model more robust, with a stronger generalization power and consequently less prone to overfit. The droupout forces every neuron to be able to operate independently, by reducing the dependencies between neurons. In insufficient datasets the model may be able to learn the statistical noise in the training data, which evidently induces bad performances when new data is fed to it, thence the use of this technique.

4.1.4 Flatten layers

This is a simple layer, in which no elaborate operation takes place, just a basic compression of the input by removing all its dimensions except for one, producing a monodimensional vector, just as portrayed in Figure 4.6.

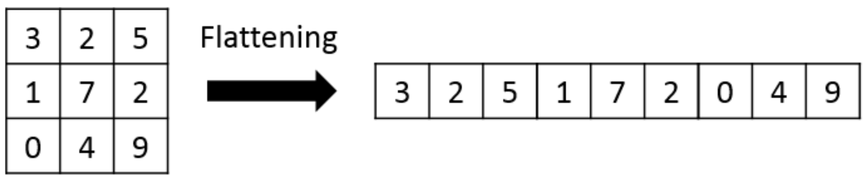


Figure 4.6: Simple illustration of the flattening process.

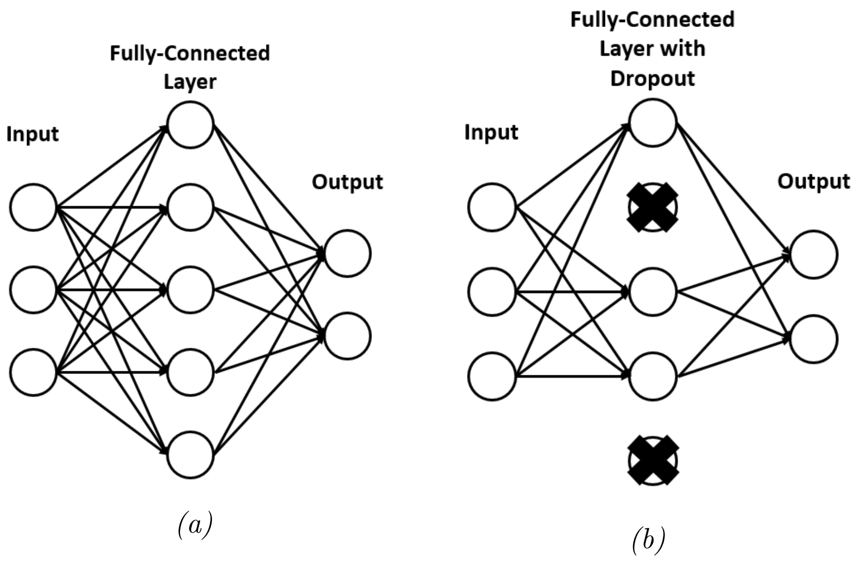


Figure 4.5: Schematic representation of the Fully-connected layer, (a), and the Fully-connected layer with dropout, (b)

4.2 Model architectures

There are different types of neural network architectures that highlight distinct DL capabilities. In this master thesis 2 unique model architectures were explored: CNN and Fully Convolutional Neural Network (FCNN).

Nevertheless the distinct architectures proposed, there is an architectural common aspect between them, which is the branch-like setup, derived from the double input of a protein pair. A PPI is composed by two proteins that are fed to the model and then it has to learn how to decide whether the proteins interact or not. A possible approach to manage the double input inherent in a PPI, and already explored in several papers [77, 78, 52, 48, 47], consists on concatenating in a single vector the representation of both proteins, before feeding it to the model. However, this single input approach may be a little limiting, as the model recognizes both proteins as being part of the same entity, unable to distinguish where a protein ends and its partner starts, due to their concatenation and representation in the same vector.

On the other hand a multiple input architecture processes the proteins of an interaction as separate entities, in which rather than concatenating the proteins, each protein representation is separately fed to a stream of layers, so that each separate network is able to better assimilate the features of a single protein. This approach has seen some use in various works [79, 68, 80, 63]. Du et al. [68] and Zhang et al. [63] even compared both techniques and concluded that implementing separated networks promoted better performances.

4.2.1 CNN

This is one of the standard architectures of neural networks and its simple architecture is inviting to use. This type of architecture was introduced by LeCun et al. [81] in 1998, but only more recently has been getting the deserved attention. This neural network has been the foundation of several other more elaborate architectures. The CNN is considered as the crown jewel for image classification and recognition [82], but it has also been applied in natural language processing [83], which works with sequences, obtaining very interesting results.

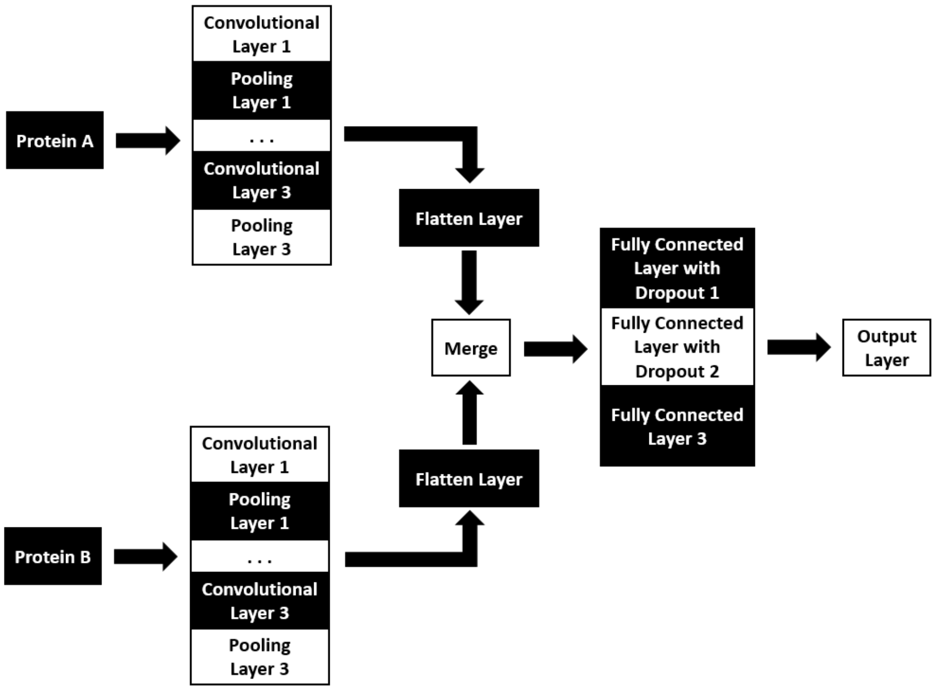


Figure 4.7: Schematic representation of the CNN model.

The overall view of the architecture of the model is illustrated in Figure 4.7. The separate networks derived from each protein are easily identified, where the three convolutional layers are interspersed with pooling layers. This first block of the whole model is responsible for extracting the high-level features of the complete protein sequences, considering that the stacking of convolutional layers hierarchically decompose the sequence, layer by layer, so that each consecutive layer learns to detect higher-level features, while at the same time preserves their spatial dependencies. Theoretically the first layers extract low-level features and, as the model gets deeper and deeper, it starts to apprehend features of higher orders more relevant for the prediction of PPIs. All the convolutional layers of this architecture were integrated with padding, so that dimensions of the input match the dimensions of the output.

At the end of the separate networks both proteins have been filtered and are represented by their relevant features. The fully connected layers require 1D inputs, due to their disregard for the spatial structure information, but the output from the 1D convolutional layers is a 2D representation. This output has then to be submitted to a flatten layer, so that the protein can be represented by a single dimension long vector, becoming compatible with the fully connected layers. These two vectors are then merged into a single one, specific to each interaction, and posteriorly fed to the final block. The final block is composed by three fully connected layers and different combinations of the features are tested so that the model learns a function, that is capable of distinguish between interacting and non-interacting pairs, from the high-level features detected.

4.2.2 FCNN

With the purpose of creating a homogeneous network, only composed by convolutional layers, Springenberg et al. [84] aspired for simplicity and developed a FCNN. This type of neural network architecture competed performance-wise with state-of-the-art models on image classification problems, while seeing no use in studies to detect relevant features in sequences, at the time of writing this work. Notwithstanding the significance of the study conducted by Springenberg et al. [84], the architecture incorporated in this master thesis does not fully match the one used in the study. They do in fact evaluate the substitution of pooling and fully connected layers with proper convolutional layers, but integrate a different technique to flatten the representations. Meanwhile, in this work, to achieve the desired homogeneous architecture of the FCNN every non-convolutional layer of the previous CNN model was replaced by a corresponding convolutional layer.

As previously described, the pooling layers perform fixed operations on the feature maps with the intention of subsampling them. The substitution of these layers for convolutional layers can be faced as an opportunity for the model to learn the pooling operation, which increases its expressiveness capacity. The main purpose of a pooling layer is to accomplish the dimensionality reduction, but this is also easily achievable just by using convolutional layers with no padding and a stride of two, instead of the standard stride of one, without any loss in accuracy, as it was detected in several image recognition benchmarks [84]. The non-existence of padding coupled with the increase of the stride forces the filter to translate, during the convolutional process, across the whole input in bigger steps, ergo the reduction of the dimensions of the feature maps produced. The implementation of convolutional layers with a higher stride even decrease the computational time as there is no need to add pooling operations to the model, since the convolutional layers are capable of subsampling, while also convolving. The original work advised to use larger stride convolutional layers once in a while to be able to reduce the size of the feature maps generated. In Figure 4.8 it is portrayed how the 1D convolutional layers with stride of 2 and no padding are capable of subsampling the input, similarly to the 1D pooling layers. On the other hand the convolutinal layers with stride and padding of 1 preserved the width of the input.

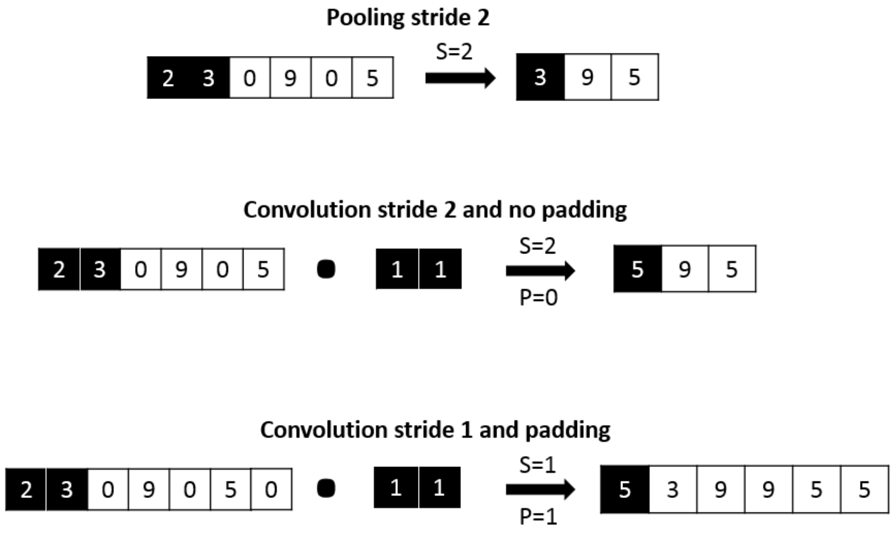


Figure 4.8: Illustration of the subsampling capacity of the convolutional layers, when an adequate stride, S, and padding, P, is integrated. For simplification reasons an input of 6×1 is represented and an output of 3×1 is achieved by the subsampling layers.

The following layers to be replaced are the flatten layers. The latter simply reduce a matrix to a single vector, then this condensing capability has to be the main focus of the replacing convolutional layers. If the size of an individual filter is the same as the size of the input and there is no padding, then, during the convolution, the filter is not translated across the whole input because it already covers it all, producing a single output neuron. The input of the replacing convolutional layers is $width \times depth$, then if the dimensions of the filters integrated match the dimensions of the input and there is no padding the output of this layer will be, according to Formula 4.3, $1 \times number\ of\ filters$, which obviously flattens the input, as illustrated in Figure 4.9. Applying convolutional layers to flatten the input is not as destructive as the flatten layer, because during the convolutional process the model attempts to create a representation that incorporates all the spatial and structural information and compresses all the knowledge of the feature maps on a single neuron, rather than simply linearly flattening the input.

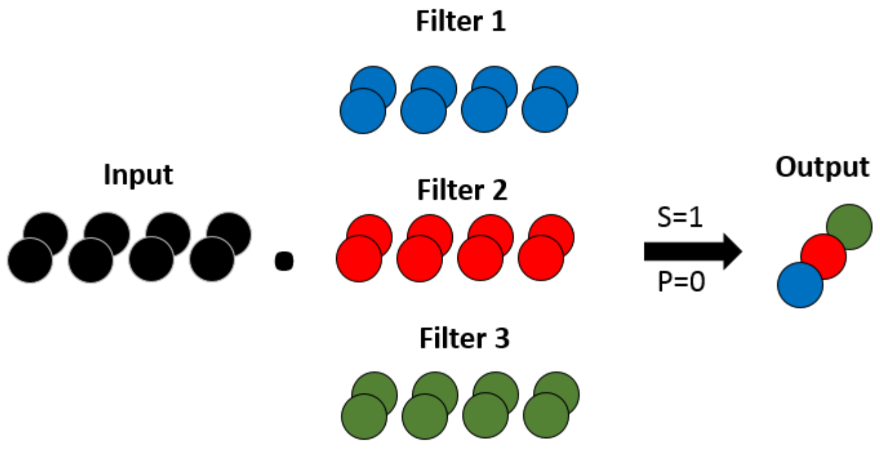


Figure 4.9: Simple schematic representation of the flattening potential of the convolutional layers. For 1D stacked 4×1 input and three matching filters of 4×1 the convolutional process, with the proper stride, S, and padding, P, obtains an output of 1×3,

Once the pooling and the flatten layers are replaced, only the fully connected layers remain to change. It is no surprise that the fully connected and the convolutional layers are not that disparate, as their main difference lies on the connectivity of the neurons between layers. In the first layer the neurons are connected to every neuron of the previous layer, while in convolutional layers the neurons are connected only to a small region of the output of the previous layer, determined by the dimensions of the filters defined. Nevertheless, in the end both layers compute dot products between the input and their weights, which result in similar operations. Consequently, using a convolutional layer with filters with proper dimensions is in essence the same as using fully connected layers. The dimensions of the filter just have to match the dimensions of the output of the previous layers. Since the representations have already been flattened by the proper convolutional layer the input of the convolutional layers replacing the fully connected is $1 \times depth$. Incorporating filters of size 1 attains an output of $1 \times number\ of\ filters$, similar to the output of a fully connected layer. With this approach every feature map, derived from each filter, is composed by a single neuron. Due to parameter sharing every produced neuron as

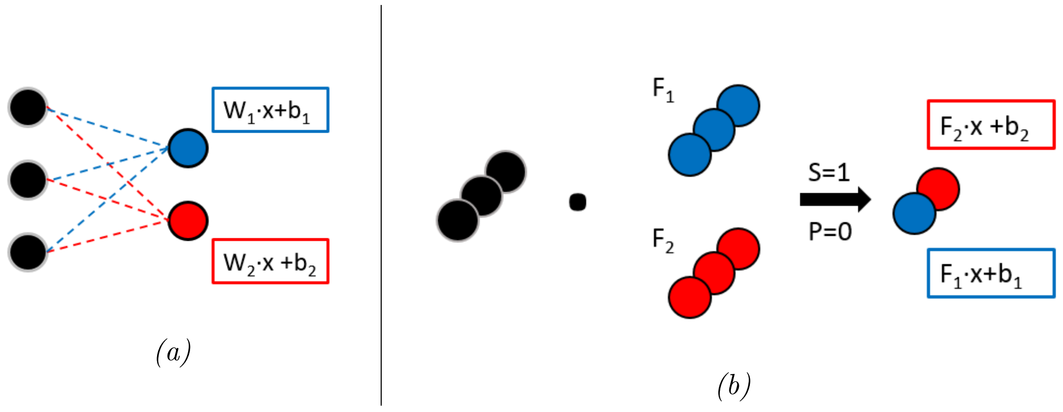


Figure 4.10: Demonstration of the similarities between a fully connected layer , (a), and a convolutional layer (b). If on the flattened input with 1×3 dimensions, two filters with the same dimensions are applied, with the proper stride, S, and padding, P, then an output of 1×2 is obtained, by a process similar to the one incorporated on the fully connected layers.

a different set of weights, because there is only one neuron per depth slice of the output. Every filter is then correctly associated to the weights of the connections of the fully connected layers.

As demonstrated in Figure 4.10, for the already flattened input the proper filters can in fact reproduce the mechanism behind the fully connected layers. The elements of the filters, represented by the colored circles, are equivalent to the the weights of the connections of the fully connected layers, and, if they are equal, the output obtained by both layers is similar.

The overall view of the FCNN model is exemplified in Figure 4.11. The separate networks, derived from the two proteins are again easily identified. Similarly to the CNN model the convolutional layers were stacked one after the other, in order to extract the high-level features of the separate protein sequences. In this initial block the main discrepancy lies in the non-existence of pooling layers. The dimensionality reduction is achieved with the convolutional layers with stride of two. The representations obtained in this block are then flattened by the suitable convolutional layers, just like they are in the CNN model by the flatten layers. The number of filters of this specific layer is the same of the previous layer, so that it only compresses the representations of each feature map and does not create new ones.

Following the merging process, the representations are subsequently fed to the final block of convolutional layers. These convolutional layers are capable of simulating the fully connected layers of the CNN model, by applying the adequate filters. In the whole model every convolutional layer, except for the ones with stride of 1 placed in the first block, are implemented with no padding, so that the desired dimensions can be obtained.

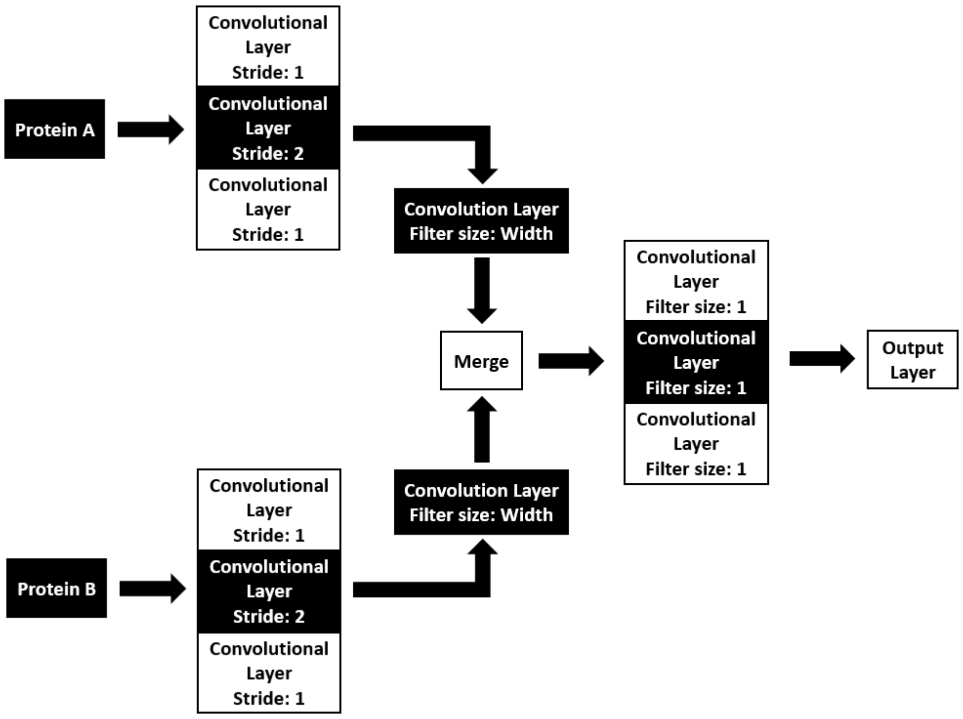

Protein A
Convolutional Layer Stride: 1
Convolutional Layer Stride: 2
Convolutional Layer Stride: 1
Convolution Layer Filter size: Width
Merge
Convolutional Layer Filter size: 1
Convolutional Layer Filter size: 1
Convolutional Layer Filter size: 1
Output Layer
Convolution Layer Filter size: Width
Protein B
Convolutional Layer Stride: 1
Convolutional Layer Stride: 2
Convolutional Layer Stride: 1

Figure 4.11: Schematic representation of the FCNN model.

5

Hyperparameters

5.1 Hyperparameters tuning

A fundamental task when using neural networks consists on describing the reasoning behind the implementation of a specific neural network architecture, trying to contextualize it in the problem. Once this is achieved it is essential to give a meticulous rundown of the various hyperparameters integrated, since the DL models are structures much more complex than they seem, considering that the tweaking of various embedded hyperparameters can lead to drastic changes in the performance. Notwithstanding the large number of parameters of a neural network and the even larger number of possible combinations of those, only some parameters were optimized, which were the number of filters of the convolutional layers, the number of neurons of the fully connected layers, the optimizers and their learning rate, the activation functions and, finally, the operation behind the merging process.

The number of layers of each of the blocks of the models, is without a doubt the simplest hyperparameter to understand. Although, there is no standard approach to define the actual number. Generally, the bigger the number, the bigger the computational cost to train the whole model and it does not necessarily result in better performances. Despite the uncertainty behind the number of layers, every block of both architectures was established with three convolutional or fully connected layers, adopting a similar approach to several works [68, 80, 63, 78, 52].

Another essential task consisted on determining the combination of number of filters implemented in the convolutional layers, as well as the combination of number of neurons integrated in the fully connected layers. Both these hyperparameters are essential towards the definition of a model architecture with a reliable distinguishing power. Several values were given to each and various combinations were tested.

The CNN architecture, contrary to the the FCNN architecture, has pooling layers.

In the pooling layers the size of the pooling window was set to 2, as well as the pooling stride that was also fixed at 2. Generally, these are the standard values, as larger windows are too destructive.

The aim of the DL models, while training, consists on reducing the difference between the predicted output and the actual output. This difference is computed by the loss function. Ideally the minimum difference between the outputs is achieved when the minimum of the loss function is discovered. The prediction of PPIs in this work is considered a binary classification problem, in which the model has to decide between two different classes to classify an input. The standard loss function for these type of problems is the binary cross entropy [63]. It is presented in the upcoming formula, in which y represents the actual labels and p the predicted ones.

$$L(y,p) = -\frac{1}{n}\sum_{i=1}^{n}(y(i) * log(p(i)) + (1 - y(i)) * log(1 - p(i))) \tag{5.1}$$

Now that the loss function is defined, it is necessary to find the values for the weights of the network that minimize the loss function value, while assuring the generalization capability of the model. The techniques used to search for the optimal set of weights are gradient descent optimization algorithms, which aim to compute the gradient of the loss function, repeatedly evaluate it and update the set of weight accordingly. These optimizers use the loss function as guidance, to tell themselves if they are going through the right path to reach the loss function local minimum. Since the choice of an optimizer is usually dependent on the model and the dataset, two distinct optimizers were considered during the model conception, the Adam [85] and RMSprop. The learning rate of the learning process, is an embedded hyperparamenter of the optimizers. This variable determines how quickly it updates its weights during the backpropagation of the training phase, as a response to the computed gradient. Small values, extend the training time, while larger values shorten it, but do not ensure an optimal convergence of the model performance.

Two different activation functions were also explored, the ReLU [86] and Swish functions [87]. The activation function is an essential element of the convolutional and the fully connected layers, as it is able to normalize the output, and determine which data gets through to the next layer, by activating or deactivating the output according to the limit implemented by the function. Both the activations considered are non-linear, which help the model to fit to the complex data. ReLU is the most widely used activation function, because of its simple computation and ability to obtain good performances on state-of-the-art models.

$$ReLU(x) = max(0,x) \tag{5.2}$$

However, its basic conformation maps all the negative output values of a neuron to zero, which is a limitation during the backward phase of the learning process, as a negative value close to zero is treated the same way as a negative value distant from zero. Once some neurons achieve the zero value, their gradient also collapses to zero and no further update of its weights is verified and the learning phase becomes purposeless, as if the model died. Swish is an activation that aims to solve the dying ReLU problem [88], due to its non-monotic conformation. In Figure 5.1 the conformations of both activation functions are graphically illustrated.

$$Swish(x) = x \times \frac{1}{(1 + e^{-x})} \tag{5.3}$$

Since the activation is problem dependent both functions were evaluated to determine which contributed the most towards the best performing model.

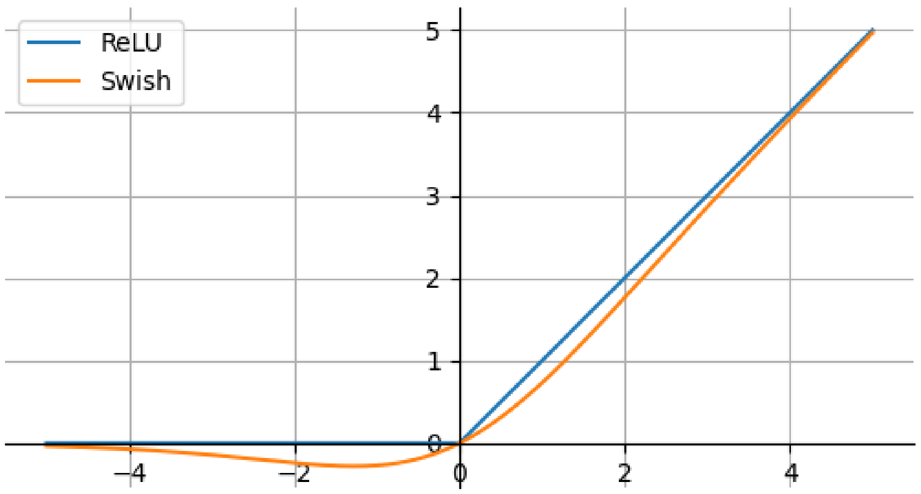


Figure 5.1: Graphic illustration of the ReLU and the Swish activation function.

When came the time to merge the high-level features vectors of each of the proteins it was investigated whether an element-wise multiplication was beneficial, due to its potential to highlight contrasts between vectors, or if the standard concatenation of the features [68, 63, 78] is the right course of action. The process behind each technique is illustrated in Figure 5.2, in which it is easily identified that, contrary to the concatenation, which produces a vector with double the size of the input, the element-wise multiplication is capable of preserving the input dimensions, which consequently leads to a smaller computational burden.

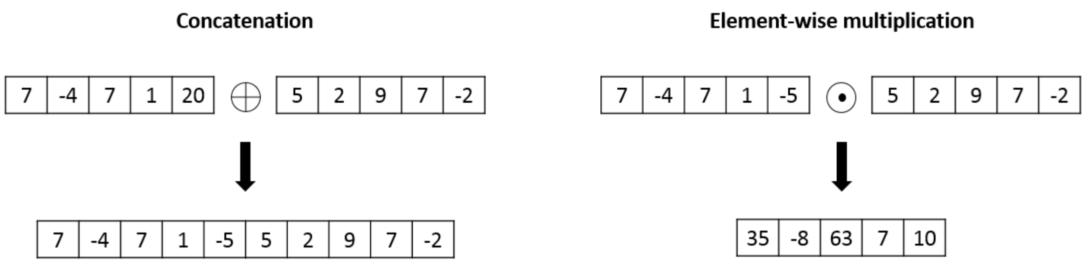


Figure 5.2: Basic representation of the concatenation and element-wise multiplication merging processes

At the end of both models an output layer was inserted. In the CNN model it was a fully-connected layer with just one neuron, while in the FCNN model it was a convolutional layer with one filter. Both layers were incorporated with a sigmoid activation function in order to obtain a probability between 1 and 0, which is easily translated to whether the input proteins interact or not, respectively.

In spite of all the progress made developing DL algorithms, there is no single model capable of correctly classifying all data given to it, partly due to the incoherences between datasets. As mentioned before three distinct datasets, with obvious differences between them, were considered for this work. Thanks to their contrasting individual characteristics, they represent distinct interactions domains and the properties of the data are uncorrelated, so, evidently, a model trained on these datasets will fit distinct functions to the data, that emphasize distinct patterns. Also, the architecture and the parameters of a model highly influence its predictive potential. Therefore each of the datasets served as a starting point for the two neural networks architectures considered. As a consequence each architecture presents three models, one for each dataset, totalizing six different models established and tuned.

During the training phase the hyperparameters that induced the highest accuracy values in the validation set were noted and are represented in Tables 5.1 and 5.2. A grid search technique was integrated to explore the combination of parameters that contributed the most for the optimal model. However, before applying this technique several combinations of three distinct number of filters and neurons were randomly chosen, in which each value is integrated in one of the three layers of every block. It was from these random combinations and a range of values of the others hyperparameters that the grid search took place. The implementation was set up in python 3.7 and Keras [89], a high-level API for developing deep learning models of TensorFlow [90].

Table 5.1: Parameters of the best performing CNN models

	Independent dataset	**Pan et al. dataset**	**Du et al. dataset dataset**
Filters	[128,256,512]	[64,256,128]	[64,256,128]
Filter size	2	2	2
Neurons	[1024,128,256]	[256,512,1024]	[256,512,1024]
Drop rate	0.1	0.1	0.1
Learning rate	0.001	0.001	0.001
Optimizer	Adam	Adam	RMSprop
Activation function	Swish	Swish	Swish
Merging	Concatenation	Concatenation	Concatenation

Table 5.2: Parameters of the best performing FCNN models

	Independent dataset	**Pan et al. dataset**	**Du et al. dataset dataset**
Filters block 1	[256, 512, 128]	[128, 512, 64]	[128, 256, 64]
Filters block 2	[64, 128, 512]	[512, 64, 128]	[256, 128, 64]
Filter size	2	2	2
Learning rate	0.001	0.001	0.001
Optimizer	RMSprop	Adam	RMSprop
Activation function	ReLU	Swish	Swish
Merging	Concatenation	Concatenation	Concatenation

6

Results and discussion

6.1 Evaluation metrics

The evaluation metrics are highly dependent on the problem and the properties that the investigators desire to highlight. The widely used metrics, already implemented in many others studies of this field were adopted to evaluate the performance of both architectures on the main datasets. Firstly, the accuracy was assessed and it indicates the proportion of correct predictions made by the model. The sensitivity and specificity metrics were also explored, in order to express the proportion of correctly predicted positives and negative cases, respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{6.2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{6.3}$$

The TP refers to the true positives and TN to the true negatives. While, the FP represents the false positives, and FN the false negatives.

Beyond the evaluation metrics considered, the confusion matrix and the Area Under the ROC Curve (AUC) are also computed. The confusion matrix portrays the actual values of the TP, TN, FP and FN, which allows for a more detailed analysis of the results. On the other hand, the AUC evaluates how a model is capable of distinguishing between classes, in which a perfect classifier has a maximum value of 1.

6.2 Results

6.2.1 Main datasets

To assess the predictive potential of the proposed architectures the FCNN was compared with the more conventional CNN. Nevertheless its high correlation with image classification problems, the CNN architecture has seen little to no use in PPIs prediction problems. Therefore both architectures are pioneers, as they explore different approaches to handle the unsolved problem of PPI prediction.

The models that achieved the highest results on the main datasets for each architecture are demonstrated in Table 6.1. From those the confusion matrix of the architectures that achieved the highest accuracy for each dataset are illustrated in Figure 6.1.

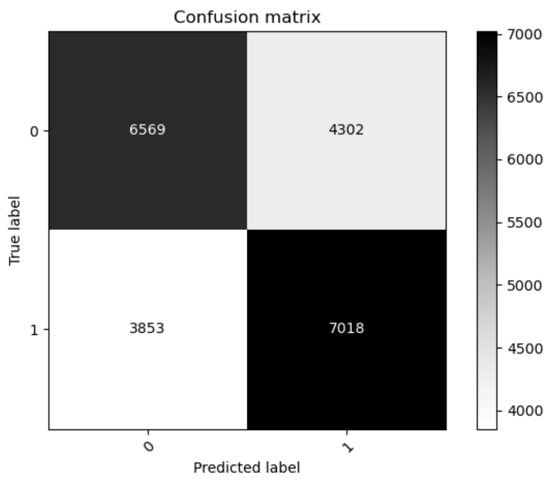
Table 6.1: Results of the best models. The dataset refers to where the model was trained and tested, according to their training and testing set.

	Metrics	**Unbiased dataset**	**Pan et al. dataset**	**Du et al. dataset**
CNN model	**Accuracy**	62.5%	98.2%	90.3%
	Sensitivity	64.5%	97.6%	89.0%
	Specificity	60.4%	98.8%	91.7%
	AUC	62.5%	98.2%	90.3%
FCNN model	**Accuracy**	60.7%	98.8%	90.8%
	Sensitivity	67.3%	98.7%	89.5%
	Specificity	53.9%	98.9%	92.2%
	AUC	60.7%	98.8%	90.8%

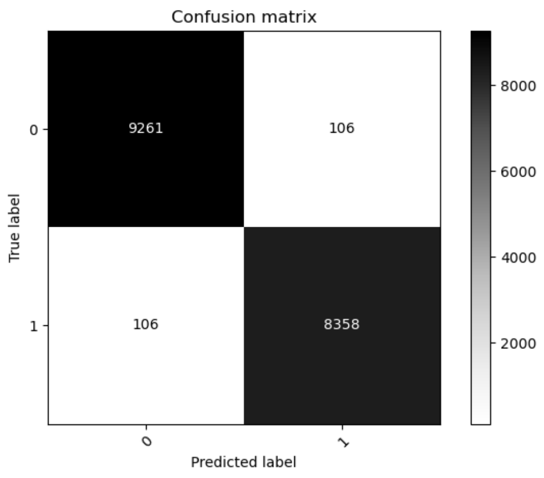
6.2.2 External datasets

Apart from the main three datasets, another four external datasets were used to assess the pratical prediction ability of the best models, when integrated in completely new datasets. This procedure allows to estimate their generalization capability and judge whether the models are ready to be implemented in large-scale experiments or not.

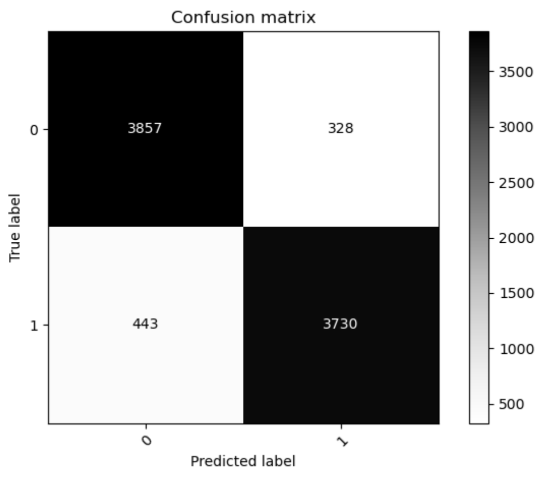
The four datasets included are from: *Escherichia coli*, *Helicobacter pylori*, *Mus musculus* and *Mammalian*. Repectively, they have 6944 positive interactions, 1399 positive interactions, 311 positives interactions, and 2049 manually curated from literature non-interacting proteins pairs in the *Mammalian* dataset.



(a)



(b)



(c)

Figure 6.1: Confusion matrix of the best models on the testing set of the considered datasets. The CNN model in the unbiased dataset (a), FCNN model in the Pan et al. dataset (b) and FCNN model in the Du et al. dataset (c)

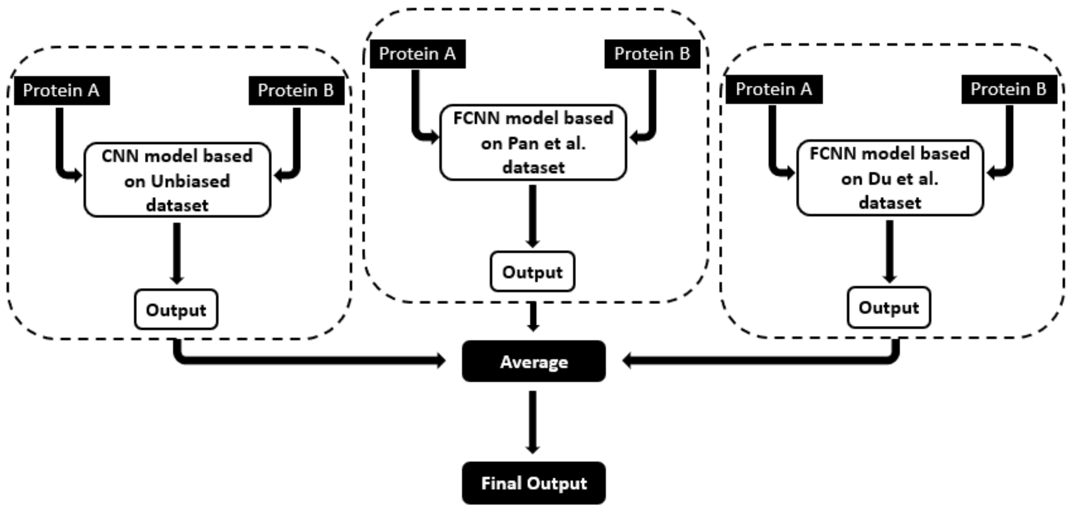

Protein A
Protein B
CNN model based on Unbiased dataset
Output
Protein A
Protein B
FCNN model based on Pan et al. dataset
Output
Protein A
Protein B
FCNN model based on Du et al. dataset
Output
Average
Final Output

Figure 6.2: Basic scheme of the ensemble algorithm.

Distinct datasets have different learning models. The three datasets considered are all part of the PPI prediction problem, but at the same time describe diverse regions of the PPI domain. Therefore the three best models when presented to new interactions behave differently, classifying each interaction according to the knowledge they gathered during the training phase, in which each model was trained on one of the main datasets. An ensemble algorithm that is able to combine the output of every model might be an interesting alternative, as it integrates all the information into a single decision. An ensemble classification algorithm is capable of taking advantage of several models that possess different regions of competence and producing an optimal predictive algorithm, avoiding the variance of the single models. Figure 6.2 portrays a simple scheme of the ensemble technique used.

Table 6.2: Accuracy results of the best models on the external datasets

	Ensemble	**CNN on Unbiased dataset**	**FCNN on Pan et al. dataset**	**FCNN on Du et al. dataset**
E. coli	91.5%	52.7%	98.5%	80.8%
H. pylo	91.9%	52.6%	99.6%	81.8%
M. musc	92.9%	58.5%	99.0%	83.3%
Mammalian	25.8%	48.7%	33.6%	18.9%

The best models trained on the main datasets correspond to the CNN model for the unbiased dataset and the FCNN model for the Pan et al. and the Du et al. dataset. They were tested on the four external datasets, and then their inputs were combined through the average, which originated the ensemble classifier. In Table 6.2 the results of the three proposed models and the ensemble model are illustrated.

6.3 Discussion

Even though benchmark datasets were used as the starting point, the subsequent elimination of some interactions, attributable to the padding constraints, results in the creation of slightly smaller datasets. Nevertheless, the PPIs representation and distribution of each dataset is similar to its respective benchmark dataset, as the larger part of the dataset remains the same. Just by itself this simple fact does not completely invalidate the comparison of the proposed models with the state of the art techniques, built upon the respective benchmark datasets. It simply hinders the comparison task, as is is not possible to clearly claim that the models proposed in this work surpass the already established ones, it is only justifiable to assess whether their performances are at a similar level or not, which just corroborates the reliability of the models and validates their predictive potential.

Comparison with existing techniques

The meticulous analysis of the Table 6.3 and Table 6.1 recognizes that the individual models established in this work are on par with the state of the art algorithms, as both techniques achieve high results in the benchmark datasets. It is clear that in some cases the CNN and FCNN models even surpass the state of the art alternatives, but again, those differences may be a consequence of the small variances of the datasets, and not from a performance standpoint. With the results obtained it is only fair to declare that the models are in fact valid alternatives. These performances legitimize the convolutional layers capabilities to automatically extract crucial features from the whole sequence and further analyse of them, contradicting some of the available state of the art algorithms that resort to simple ML techniques.

These findings dispute several studies that solely rely on protein descriptors to obtain valid protein representations, as, it is easily concluded, that working with the complete raw sequence of amino acids of the proteins is a logical technique and might even be an interesting replacement to further explore as a substitute to those. The aggregative capability of the descriptors is their main asset, but at the same time is also considered as their main disadvantage, as crucial information like the sequence order is completely overlooked.

The FCNN model produces the better results on benchmark datasets, when com-

Table 6.3: Accuracy results of some state of the art models built upon one of the benchmark datasets.

	Pan et al. dataset	**Du et al. dataset**
Gui et al. [91]	97.1%	-
Wang et al. [92]	97.1%	-
Sun et al. [54]	98.1%	-
Gui et al. [78]	97.7%	-
Wang et al. [93]	98.5%	-
Pan et al. [67]	97.9%	-
Wang et al. [55]	97.8%	-
Zhang et al. [53]	-	95.2%
Du et al. [68]	-	92.5%
You et al. [40]	-	89.2%
Zhou et al. [65]	-	88.8%

pared with the CNN model. Every single metric of the FCNN models on the benchmark datasets excels the respective metrics of the CNN model. However, when analysing the unbiased dataset, which theoretically should have been a harder dataset to evaluate, the latter surpasses the FCNN, except for the sensitivity aspect. The results approve the FCNN architecture as it yields, for the most part, better performances than the CNN. After all, the homogeneous take on the architectures is at least intriguing and deserves further examination, as the non-convolutional layers when replaced may in fact produce more relevant representations of the data. Nonetheless these results enhance the complexity behind the decision of determining which model is the best overall, not allowing to define one as the clear best, as further investigation is required

Another compelling fact derived from the consistent inclusion of the Swish activation function on the best performing models, as it is portrayed in Table 5.1 and Table 5.2. Notwithstanding further validation, this might be an an indication that in the near future, neural networks should start to progressively pivot towards the implementation of this activation function, as a way to contest some of the ReLU limitations.

The algorithms built upon the Du et al. dataset, produce slightly deviated results, but nothing too concerning that completely invalidates the established models capacities. As mentioned in Section 3.1.3, during the conception of this dataset the number of negative examples generated with the non co-localization technique are almost three times more than the selected positive interactions. From those negative interactions an amount equal to the number of positive interactions was randomly selected by the authors. All the negative examples were made available by Du et al. and considering that different papers desire to compare their algorithms with the original work, they also implemented a random selection from all the negative interactions to obtain a similar dataset. However, the random factor inserted during the selection phase produces somewhat distinct datasets, thence the variance between the results.

Impact of the main datasets

Despite the overall excellent results on the benchmark datasets, the same is not verified in the models trained on the unbiased dataset, as their results are much lower. The models are fundamentally the same, so, the inconsistency has to come from the datasets and the way they are built. The main difference between the unbiased and the benchmark datasets lies in the generation of the negative examples and the selection of the positive interactions. The unbiased dataset was created from scratch, with the purpose of creating reliable and unbiased data, whereas the latters, fall back on the standard non co-localization technique to generate the negative examples. The non co-localization technique oversimplifies the datasets and subsequently the classification task.

The inherent simplifications embedded in the non co-localization technique do not take into consideration various possible and recurrent scenarios. Firstly, there are still a large number of proteins with unknown localization, which evidently limits this approach [94]. Secondly, two proteins may coexist in the same cellular localization but not interact, since the determining factors that promote a interaction are much more complex than the simple co-localization and this technique does not take into account this type of negative interactions. Additionally, strong evidences support the fact that there are, not yet experimentally determined, interacting protein pairs with different cellular localization annotations, caused by miss-annotations of their cellular localization, as some methods are based on sequence similarity and not on experimental evidence, or even because the possible localizations of those proteins are still not completely experimentally determined [95]. Finally, fully described hub proteins, which have many interacting partners, are confined in different cellular compartments in different moments of time, leading to presenting multiple cellular localization annotations [96]. Since they are virtually present in multiple sub-cellular compartment, following this non-interacting protein pair generation procedure is severely limiting as it is almost impossible to generate a negative example with one of these hub proteins. This fact simplifies a lot the succeeding classification task,

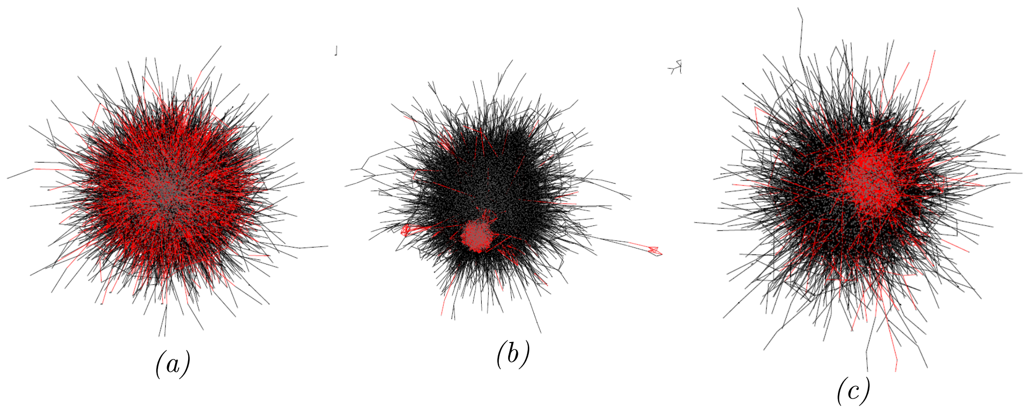


Figure 6.3: Distribution of the interactions in each of the PPIs domain of the three evaluated datasets. (a) unbiased dataset, (b) Pan et al. dataset, (c) Du et al. dataset. Proteins are represented by the grey nodes, the positive interactions by the black edges and the negative interactions by the red edges.

given that the classifier learns that pairs where a hub protein is present are almost exclusively interacting pairs. To add insult to injury in the benchmark datasets, they explicitly say that the proteins with multiple localization annotations were removed from the set of proteins from which the negative interactions were sampled.

The complete PPI domain is characterized by presenting a massive number of negative and some positive interactions. When a domain is so vast and complex a restricted set of samples is only capable of portraying some of its properties. Hence, it comes as no surprise that some of the PPIs datasets are biased and only suitable for the examples they incorporate and not for the complete PPI domain. The obervation of Figure 6.3 easily illustrates that the benchmark PPIs datasets are in fact biased. This figure was created with Cytoscape [97], which is an open source platform to visualize elaborate networks. The proteins are represented by the grey dots, whereas the interactions are illustrated by the edges between the grey dots. These edges are either black lines, if the respective interaction amongst the two connected proteins is a positive interaction or red lines if the connection composed by the attached proteins is a negative interaction. The network was produced with a Force-Directed Layout, which is an information visualization algorithm that places the nodes in a 2D space, according to attractive and repulsive forces. The attractive forces enhances the connected adjacent nodes, while the repulsive forces tend to draw non-connected nodes further apart. These forces are iteratively applied until the graph achieves a mechanical equilibrium state. The resulting network conformation is capable of exposing its clustered structure with a well-balanced distribution of the nodes in an organic and aesthetically pleasing way.

Within the scope of the complete proteome, it is clear that the negative interactions represent the bigger fraction of the possible pairing combinations, as positive interactions are rare and only occur under special circumstances. Obviously, when the number of positive interactions is the same as the negative interactions, the negative interactions should present a distribution at least as sparse as the distribution of the positive interactions in the proteome space. The observation of Figure 6.3 does not illustrate such scenario across all datasets. The unbiased dataset, does in fact portray a sparse distribution of the positive and the negative interactions, however in the benchmark datasets that is not verified, as the negative interactions are limited to small areas, due to the restrictions incorporated in the non co-localization technique. The unmissable biased distribution of the negative interactions is the main cause of the discrepancy between the results of the proposed models, as the more restrained the negative interactions are, the better the results. The models built upon the dataset with the more restricted negative interactions, Pan et al. dataset, are the ones with the best results, while the models built upon the dataset with the sparser negative interactions, the unbiased dataset, obtained the worst results. The proportionality between the restriction of the negative interactions and the improved performance, further justifies the previous statements.

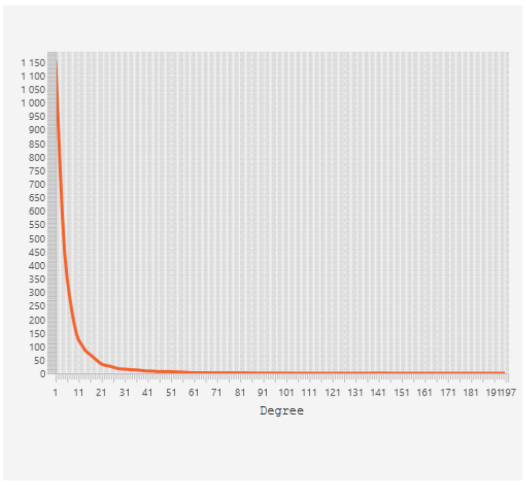
Beyond the technique used to generate the negative examples, another aspect that causes a restrained distribution of the negative interactions is the number of unique proteins from which the negative protein pairs were randomly selected. As expected, if a fixed number of interactions is desired then, the larger the sampling group, the sparser the distribution of such interactions will be, because there is a higher probability to include a larger number of unique proteins in the interactions. In Table 3.1, Table 3.2 and Table 3.3 this detail is easily verified. Again, the more restrict datasets are the ones that have a higher discrepancy between the number of unique proteins that generate the positive interactions and the number of unique proteins from which the negative interactions are sampled.

Aditionally, as if the mentioned discrepancy was not already enough, the Pan et al. dataset also does not assure that the proteins that compose the negative interactions are the same proteins of the positive interactions. In the Pan et al. dataset, from all the unique proteins that compose the negative interactions only 58,1% are also present in the unique proteins that produce the positive interactions. In this case there are a large number of proteins that only produce one interaction, consequently the models easily recognize which proteins are associated with negative interactions and then classifies every interaction with those a negative one, which clearly reduces the difficulty level of the classification task. However, the same approach can not be applied to the other two datasets, as almost all proteins that produce the negative interactions are also part of at least a positive pair and the models need to actually decipher from the raw sequence which structural and spatial patterns influence the PPIs. This comprises another detail that further justifies why the techniques implemented in the Pan et al. dataset achieve almost perfect performances.

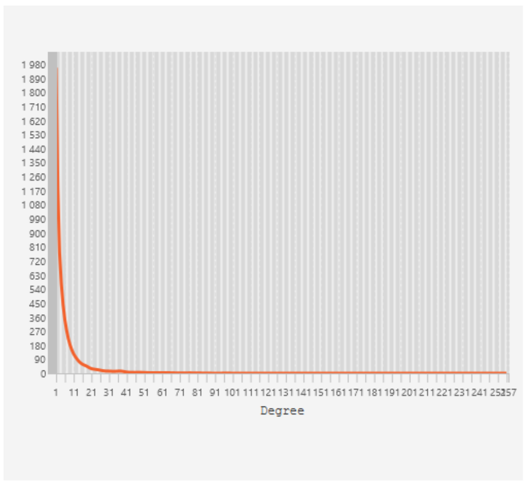
The visual identification of the distributions is a powerful tool to analyze the organization of the datasets, especially, when coupled with the formulation of plausible reasons that stimulate the embracement of such distributions. Nevertheless, a more rigourous examination is essential to justify some of the statements. The degree distribution is a simple metric commonly used when studying networks. A degree is the number of connections a particular node has with others. In a protein-protein network the degree can be interpreted as the number of possible interactions a specific protein has. With this metric it gets simpler to pinpoint some of the protein clusters that might be formed. The degree distribution of both the positive and negative interactions of each of the main datasets was computed and are illustrated in Figure 6.4. In the unbiased dataset both the positive and the negative interactions present a similar degree distribution, which legitimizes their equivalent distributions of the interactions, due to their non-apparent differences.

On the other hand, in the benchmark datasets, there is a clear difference between the degree distribution of the negative interactions and the positive interactions. The latter presents an approximate power law curve, in which most of the nodes have low degrees, not disregarding some exceptions that have a high degree. This indicates that, predominantly, a specific protein interacts with a small number of proteins, which motivates the dispersion of the proteins in the PPI domain. Nonetheless, the negative interactions severely deviate from the power law curve, as there is a high emergence of nodes with a high degree, represented by the multiple local maximums of the distributions. Which means that there are several hub proteins that are paired with other non-interacting proteins. Due to the higher number of hub proteins, the negative interactions are more constrained in the PPI domain, covering a smaller area than the positive interactions, thence the biased distribution of the negative interactions in the benchmark datasets. The degree distributions of the negative interactions have to be regulated to ensure similar distributions and to propitiate a good generalization capability [98].

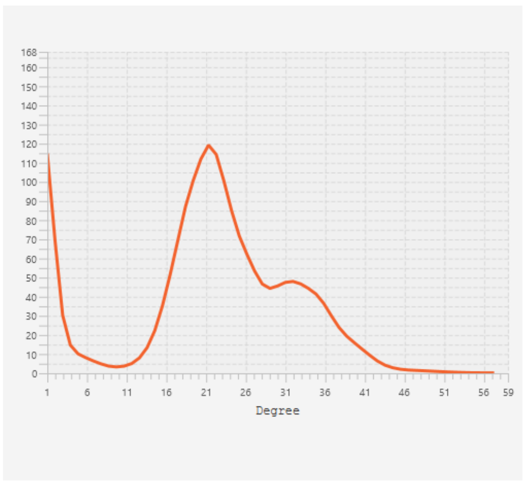
An ideal model perfectly identifies positive as well as negative interactions, which enables its insertion in experimental techniques as a validation technique or even as an independent effective one to discover new interactions. If a model aspires to



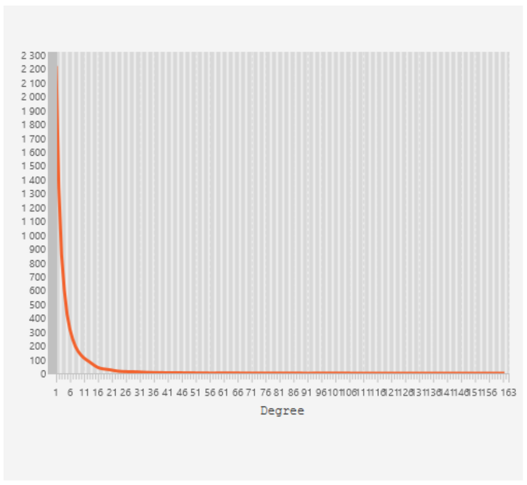
(a)



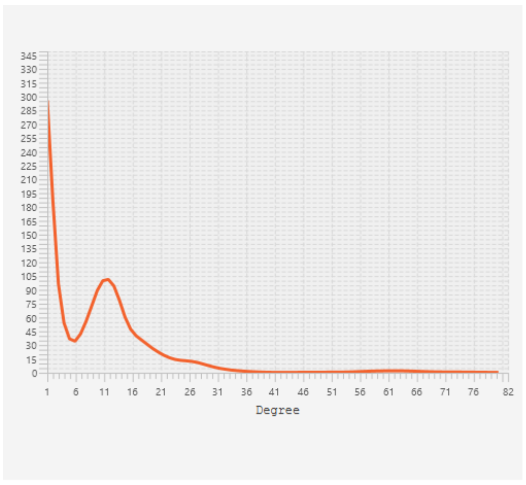
(b)



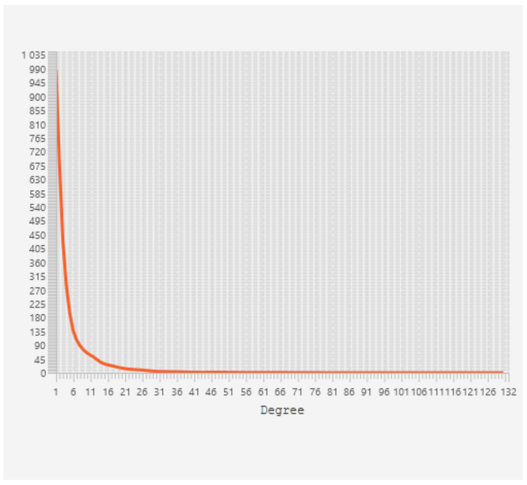
(c)



(d)



(e)



(f)

Figure 6.4: Degree distributions, (a) and (b) are, respectively, the negative and the positive interactions of the unbiased dataset, (c) and (d) also the negative and positive interactions of Pan et al. dataset and ,finally, (e) and (f) are the negative interactions and positive interactions of Du et al. dataset.

be used as a reliable technique it needs to master the distinction of the different interactions. The accuracy is the evaluation metric that encapsulates this distinguishing capability the best. However, in the current spectrum of things the correct identification of positive interactions is a critical aspect to scrutinize and must receive the necessary attention, since these comprise the experimentally determined interactions, whereas the negative interactions are computationally generated and do not exhibit the same level of credibility. The sensitivity is the metric to resort to analyze the capacity of the model to classify positive interactions. Strangely enough, while interpreting the results of Table 6.1, only the models trained on the unbiased dataset presented a sensitivity value higher than its specificity, which expresses the capacity of the model to classify negative interactions. Undeniably, these findings imply that these models have a hard time identifying the negative interactions, when compared to their ability to identify the positive interactions. Which makes perfect sense in the context of the problem, as they more easily recognize the reasoning behind the conception of positive interactions than comprehend the inherent randomness of the negative interactions. The models built upon benchmark datasets demonstrate the opposite scenario, probably due to the visible spatial restriction of the negative interactions. This finding, as small as it seems, adds credibility to the unbiased dataset.

Several published papers solely focus on developing models and novel negative sampling techniques that lead to unprecedented performances, while completely disregarding the evaluation of how representative of the proteome their datasets actually are. On the other hand, in this work the research was revolved around the attempt to create an unbiased and biologically representative dataset, even if that produced worse results of the innovative models trained on that dataset. Notwithstanding further validation, the unbiased dataset as well as the proposed architectures could be the starting point for new DL approaches to come.

Performance on the external datasets

The results from Table 6.2 clearly indicate that the FCNN models built upon the benchmark datasets perform well when classifying positive protein interactions, but when the goal is to classify negative interactions they falter and obtain poor results. The high results of these models when predicting positive interactions may even be a consequence of the bias developed during the training phase on the benchmark datasets, as they predominantly assign to any new input sample the positive label. Considering that the benchmark datasets have a very restricted set of negative interactions this is a plausible reason. Due to their limited distribution, these models learn that there is a higher likelihood for a new protein pair to be considered a interacting one, thence the contrasting results on the positive and negative external datasets.

On the other hand the CNN model, which was trained on the unbiased dataset, achieves similar ordinary performances when distinguishing the positive and the negative interactions of the external datasets. The similar performance on both examples was an expectable outcome, given the equally distributed positive and negative interactions of the unbiased dataset. However, the level of performance leaves room for doubting to what extent the PPI prediction problem, based solely on the sequence, is a resolvable one.

The model based on the unbiased dataset developed a better mechanism to distinguish negative interactions, while the models based on the benchmark datasets are predisposed to classify the positive interactions. An ensemble model that is capable of integrating the three proposed models was established. The results obtained with this model are still somewhat questionable due to its higher preponderance to correctly classify positive interactions, while slightly improving the performance on the negative interactions of the *Mammalian* dataset.

Shortcomings

In spite of all the thorough research done, there are some aspects that in the future could be refined, or even different approaches could be adopted. Starting from the grid search technique integrated, a larger diversity of hyperparameters could be explored and even the range of the values of each specific hyperparameter could be increased. Resulting in a more fastidious search for the best hyperparameters combinations. However, as six different models were created, derived from the two architectures explored in the three main datasets, the current approach seemed like a reasonable compromise between the computational time consumed and the results attained with the various combinations explored.

One of the main purposes of this thesis work comprises the assessment of whether the sequential information of the proteins, incorporated in innovative DL models was a valid alternative to the existing approaches. From the results obtained it is possible to claim that all the approaches are in fact valid alternatives to the existing ones, however, as usual in DL implementations, the number of samples used to create the models can always be increased, allowing the models themselves to learn from more samples and to better comprehend the distribution of those in the proteome.

The other main goal consisted in evaluating the reliability of several PPIs datasets. To compare the existing approaches, an unbiased dataset was created. It was concluded that the benchmark datasets evaluated are in fact skewed, especially the distribution of the negative interactions. Nonetheless the various papers developed based on this benchmark datasets, a more in-depth analyses could be included, consisting on the investigation of more established datasets and the works developed in those, to assess to what extent are all the existing datasets biased representations of the proteome or not. Not only that, but also a more detailed examination, could reinforce the inclusion of the unbiased dataset in future works.

The ensemble model introduced is a simple alternative to combine the three models developed on the main datasets. A wider variety of datasets could be considered, providing for the creation of more individual models built upon distinct datasets, which could potentiate a more efficient classification, as more knowledge is integrated in a single ensemble classifier. Not only a bigger variety of models could be incorporated, but also, during the association phase of the outputs, a more rigorous association technique could be integrated. Rather than merely using the average of the outputs, a weighted average could be explored, in which the best models are more influential towards the classification task of the ensemble classifier.

7

Conclusion

In this master thesis a novel unbiased independent dataset and two innovative DL architectures were established. The performance of those two innovative architectures was evaluated on three datasets and four external datasets and during the analyses of such results a reasonable explanation was given that was capable of interpreting the discrepancies. The sole comparison of the proposed models with the state of the art algorithms trained on the benchmark datasets validated our decisions. The obtained results support the ability of the models to extract valuable features from complete raw protein sequences, which enhanced the predictive potential. This validation was crucial, as it corroborates that the disparate results with the distinct datasets were a consequence of the contrasting negative sampling techniques and not a result of an unsatisfactory model. At the same time, the reliability of the innovative FCNN architecture was authenticated and it might develop into a standard architecture, not disregarding the need for further investigation.

On the other hand, we also concluded that the dataset is a valuable element of a DL experiment, as the data strongly influences the performance. A model is only as good as the insights it can extract from the dataset fed to it, we can implement different algorithms, tweak the parameters however we want, but the final word will come from the quality of the dataset. Not every dataset is representative of the problem it was designed to, some datasets are built to provide an easy way to groundbreaking performances, by implementing strategies that simplify the classification task, while some datasets are built with the intent to actually create a realistic representation of the domain of the problem. Obviously, the latter involve much more complex tasks in order to develop a satisfactory predictive algorithm, which evidently may lead to worse results. The exclusive analyze of the results of several published papers can incorrectly lead us to believe that the large-scale PPIs prediction problem is well addressed and close to be considered a solved problem. However, with the work developed in this master thesis, the viability of several state of the art PPI prediction algorithms built upon biased datasets is questioned. The abilities that some of these guideline algorithms develop to distinguish between interacting and non-interacting protein pairs are not suitable to all the unbiased data of the PPI domain.

In spite of the middle-of-the-road results obtained with the unbiased dataset, it clearly represents a more biological realistic domain of the PPIs, due to its sparser conformation, which closer resembles the proteome of an organism. A sequence-based model, that is on par with the state of the art algorithms trained on biased benchmark datasets, but is not capable to achieve similar performances on an unbiased dataset, can only indicate that the PPI prediction problem might be much harder than it looks. There are several environmental factors that can influence the success of a PPI *in vivo*, because they are time and condition dependent, and those factors are barely taken into consideration during *in silico* and even *in vitro* techniques. Transient interactions, that are temporary and occur dynamically can be hard to detect *in vivo*. Aditionally, the concentration of the proteins highly influences PPIs, as well as the presence of competing proteins, that can prevent the establishment of specific PPIs. The simplifications integrated coupled with the obtained results in the unbiased dataset makes us wonder how far we really are to be able to computationally predict, from the amino acids sequence, large-scale PPIs and even if it is an attainable goal in the near future.

The work developed in this master thesis open the door towards the establishment of two innovative DL architectures, that integrate the complete protein sequences as an alternative to the standard protein descriptors. Not only that, but also a different PPI dataset was built, characterized by its fair distribution of the positive as well as the negative interactions, contrary to some benchmark datasets.

There are some interesting ideas, that are worth noting as future work propositions to improve the obtained results. As always, larger and less biased datasets can be incorporated. Also, with the progressive unveiling of more types of protein data, as their coverage of the whole proteome gradually increases, much more protein data can be combined in order to create more polished and informative datasets. It would give more information to the models to work with and they would not have to solely rely on the amino acids sequence. In this case the models decide, from all the diverse information available, which are the more relevant and decisive for the PPI prediction problem.

Most of the current available ML state of the art algorithms, consider the PPIs prediction a binary problem, where two proteins either interact or not. Redesigning the problem into a regression one, in which every possible interaction is assigned a probability value that portrays the binding affinity, could be an interesting strategy. Proteins interact with other proteins to different extents, since some pairs produce stronger interactions, possessing a higher binding affinity. In this scenario, the model identifies the interactions that have a higher likelihood to occur. Only determining whether two proteins interact or not is just the first step to understand, to what degree, that interaction takes place *in vivo*. Aditionally, the binding sites of experimentally determined interactions could also be integrated during the model conception. Forcing the model to play close attention to the regions, that, notably, are more prone to influence interactions.

It is obvious that not only the positive, but also the negative data play a crucial role during the training phase, and subsequently performance evaluation. Manually curate negative interactions from the literature, rather than computationally generate the negative examples, could contribute to a more solid and robust dataset. Negatome [99] is a public database that provides this type of information, however its implementation is limited due to the still insufficient number of interactions. A more extensive review of the literature is needed in order to be able to implement the derived datasets in large-scale experiments.

There is a slight possibility that with the current datasets and DL algorithms it is not possible to viably predict the whole spectrum of PPIs solely with the amino acids sequence. So, the prediction problem could be simplified. The mechanisms behind PPI are, in a sense, function-dependent, in which the molecular functions of the proteins of a pair are associated with their affinity to interact. Therefore, rather than trying to predict every type of PPI, dividing the dataset according to the functions annotations of various proteins could be an appealing alternative. Afterwards, a model specific to every function annotation pair could be created and only the proteins under those annotations are fed to the model.

Bibliography

[1] Harold Hartley. Origin of the word 'protein'. *Nature*, 168(4267):244–244, 1951.

[2] Hubert Bradford Vickery. The origin of the word protein. *The Yale journal of biology and medicine*, 22(5):387, 1950.

[3] Barnali N Chaudhuri and Todd O Yeates. A computational method to predict genetically encoded rare amino acids in proteins. *Genome biology*, 6(9):R79, 2005.

[4] Yanay Ofran and Burkhard Rost. Analysing six types of protein–protein interfaces. *Journal of molecular biology*, 325(2):377–387, 2003.

[5] Mitsuaki Yanagida. Functional proteomics; current achievements. *Journal of Chromatography B*, 771(1-2):89–106, 2002. doi: 10.1016/s1570-0232(02)00074-0.

[6] Javier De Las Rivas and Celia Fontanillo. Protein–protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLoS Computational Biology*, 6(6), 2010. doi: 10.1371/journal.pcbi.1000807.

[7] Irene MA Nooren and Janet M Thornton. Diversity of protein–protein interactions. *The EMBO journal*, 22(14):3486–3492, 2003.

[8] Pascal Braun and Anne-Claude Gingras. History of protein–protein interactions: From egg-white to complex networks. *Proteomics*, 12(10):1478–1498, 2012.

[9] Peter Csermely, Robin Palotai, and Ruth Nussinov. Induced fit, conformational selection and independent dynamic segments: an extended view of binding events. *Nature Precedings*, pages 1–1, 2010.

[10] Till Siebenmorgen and Martin Zacharias. Computational prediction of protein–protein binding affinities. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 10(3):e1448, 2020.

[11] Lars Kiemer and Gianni Cesareni. Comparative interactomics: comparing apples and pears? *Trends in biotechnology*, 25(10):448–454, 2007.

[12] Mileidy W Gonzalez and Maricel G Kann. Protein interactions and disease. *PLoS computational biology*, 8(12), 2012.

[13] Daniel P Ryan and Jacqueline M Matthews. Protein–protein interactions in human disease. *Current opinion in structural biology*, 15(4):441–446, 2005.

[14] Luca Laraia, Grahame Mckenzie, David R. Spring, Ashok R. Venkitaraman, and David J. Huggins. Overcoming chemical, biological, and computational challenges in the development of inhibitors targeting protein-protein interactions. *Chemistry Biology*, 22(6):689–703, 2015.

[15] Kevin Titeca, Irma Lemmens, Jan Tavernier, and Sven Eyckerman. Discovering cellular protein-protein interactions: Technological strategies and opportunities. *Mass Spectrometry Reviews*, 38(1):79–111, 2018.

[16] Neil P King, Jacob B Bale, William Sheffler, Dan E McNamara, Shane Gonen, Tamir Gonen, Todd O Yeates, and David Baker. Accurate design of co-assembling multi-component protein nanomaterials. *Nature*, 510(7503):103–108, 2014.

[17] Kamil Khafizov, Carlos Madrid-Aliste, Steven C Almo, and Andras Fiser. Trends in structural coverage of the protein universe and the impact of the protein structure initiative. *Proceedings of the National Academy of Sciences*, 111(10):3733–3738, 2014.

[18] Dong Xu. Protein databases on the internet. *Current protocols in protein science*, 70(1):2–6, 2012.

[19] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences*, 98 (8):4569–4574, 2001.

[20] Xiaoqing Peng, Jianxin Wang, Wei Peng, Fang-Xiang Wu, and Yi Pan. Protein–protein interactions: detection, reliability assessment and applications. *Briefings in bioinformatics*, 18(5):798–819, 2017.

[21] Nan Lin, Baolin Wu, Ronald Jansen, Mark Gerstein, and Hongyu Zhao. Information assessment on predicting protein-protein interactions. *BMC bioinformatics*, 5(1):154, 2004.

[22] Christian Von Mering, Roland Krause, Berend Snel, Michael Cornell, Stephen G Oliver, Stanley Fields, and Peer Bork. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, 417(6887):399–403, 2002.

[23] Ozlem Keskin, Nurcan Tuncbag, and Attila Gursoy. Predicting protein–protein interactions from the molecular to the proteome level. *Chemical reviews*, 116 (8):4884–4909, 2016.

[24] Adam Hospital, Josep Ramon Goñi, Modesto Orozco, and Josep L Gelpí. Molecular dynamics simulations: advances and applications. *Advances and applications in bioinformatics and chemistry: AABC*, 8:37, 2015.

[25] Giacomo Bastianelli, Anthony Bouillon, Christophe Nguyen, Elodie Crublet, Stéphane Pêtres, Olivier Gorgette, Dung Le-Nguyen, Jean-Christophe Barale, and Michael Nilges. Computational reverse-engineering of a spider-venom derived peptide active against plasmodium falciparum sub1. *PLoS One*, 6(7), 2011.

[26] Christin Rakers, Marcel Bermudez, Bettina G Keller, Jérémie Mortier, and Gerhard Wolber. Computational close up on protein–protein interactions: how to unravel the invisible using molecular dynamics simulations? *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(5):345–359, 2015.

[27] Krishnakumar M Ravikumar, Wei Huang, and Sichun Yang. Coarse-grained simulations of protein-protein association: an energy landscape perspective. *Biophysical journal*, 103(4):837–845, 2012.

[28] Andras Szilagyi and Yang Zhang. Template-based structure modeling of protein–protein interactions. *Current opinion in structural biology*, 24:10–23, 2014.

[29] Veronica Salmaso and Stefano Moro. Bridging molecular docking to molecular dynamics in exploring ligand-protein recognition process: An overview. *Frontiers in pharmacology*, 9:923, 2018.

[30] Brian G Pierce, Kevin Wiehe, Howook Hwang, Bong-Hyun Kim, Thom Vreven, and Zhiping Weng. Zdock server: interactive docking prediction of protein–protein complexes and symmetric multimers. *Bioinformatics*, 30(12):1771–1773, 2014.

[31] Sergey Lyskov and Jeffrey J Gray. The rosettadock server for local protein–protein docking. *Nucleic acids research*, 36(suppl_2):W233–W238, 2008.

[32] Mark Nicholas Wass, Gloria Fuentes, Carles Pons, Florencio Pazos, and Alfonso Valencia. Towards the prediction of protein interaction partners using physical docking. *Molecular systems biology*, 7(1), 2011.

[33] Yuri Matsuzaki, Masahito Ohue, Nobuyuki Uchikoga, and Yutaka Akiyama. Protein-protein interaction network prediction by using rigid-body docking tools: application to bacterial chemotaxis. *Protein and peptide letters*, 21(8): 790–798, 2014.

[34] Shaowei Dong, Vincent Lau, Richard Song, Matthew Ierullo, Eddi Esteban, Yingzhou Wu, Teeratham Sivieng, Hardeep Nahal, Allison Gaudinier, Asher Pasha, et al. Proteome-wide, structure-based prediction of protein-protein interactions/new molecular interactions viewer. *Plant physiology*, 179(4):1893–1907, 2019.

[35] Nurcan Tuncbag, Attila Gursoy, and Ozlem Keskin. Prediction of protein–protein interactions: unifying evolution and structure at protein interfaces. *Physical biology*, 8(3):035006, 2011.

[36] Anna CF Lewis, Nick S Jones, Mason A Porter, and Charlotte M Deane. What evidence is there for the homology of protein-protein interactions? *PLoS Comput Biol*, 8(9):e1002645, 2012.

[37] Debasree Sarkar and Sudipto Saha. Machine-learning techniques for the prediction of protein–protein interactions. *Journal of biosciences*, 44(4):104, 2019.

[38] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[39] Xue-Wen Chen and Mei Liu. Prediction of protein–protein interactions using random decision forest framework. *Bioinformatics*, 21(24):4394–4400, 2005.

[40] Zhu-Hong You, Keith C. C. Chan, and Pengwei Hu. Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest. *Plos One*, 10(5), Jun 2015.

[41] K SRIVASTAVA Durgesh and B Lekha. Data classification using support vector machine. *Journal of theoretical and applied information technology*, 12(1):1–7, 2010.

[42] Nahla Barakat and Andrew P Bradley. Rule extraction from support vector machines: a review. *Neurocomputing*, 74(1-3):178–190, 2010.

[43] Martial Hue, Michael Riffle, Jean-Philippe Vert, and William S Noble. Large-

scale prediction of protein-protein interactions from structures. *BMC bioinformatics*, 11(1):1–9, 2010.

[44] Piyali Chatterjee, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, and Dariusz Plewczynski. Ppi_svm: Prediction of protein-protein interactions using machine learning, domain-domain affinities and frequency tables. *Cellular & molecular biology letters*, 16(2):264–278, 2011.

[45] Sanghamitra Bandyopadhyay and Koushik Mallick. A new feature vector based on gene ontology terms for protein-protein interaction prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 14(4):762–770, 2016.

[46] Roger A Craig and Li Liao. Improving protein–protein interaction prediction based on phylogenetic information using a least-squares support vector machine. *Annals of the New York Academy of Sciences*, 1115(1):154–167, 2007.

[47] Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research*, 36(9):3025–3030, Apr 2008.

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

[49] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436–444, 2015.

[50] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[51] Daniele Ravì, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang. Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, 21(1):4–21, 2016.

[52] Jun Wang, Long Zhang, Lianyin Jia, Yazhou Ren, and Guoxian Yu. Protein-protein interactions prediction using a novel local conjoint triad descriptor of amino acid sequences. *International Journal of Molecular Sciences*, 18(11): 2373, Aug 2017.

[53] Long Zhang, Guoxian Yu, Dawen Xia, and Jun Wang. Protein–protein interactions prediction based on ensemble deep neural networks. *Neurocomputing*, 324:10–19, 2019.

[54] Tanlin Sun, Bo Zhou, Luhua Lai, and Jianfeng Pei. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinformatics*, 18(1), 2017.

[55] Lei Wang, Hai-Feng Wang, San-Rong Liu, Xin Yan, and Ke-Jian Song. Predicting protein-protein interactions from matrix-based protein sequence using convolution neural network and feature-selective rotation forest. *Scientific Reports*, 9(1), Aug 2019.

[56] Henning Hermjakob, Luisa Montecchi-Palazzi, Chris Lewington, Sugath Mudali, Samuel Kerrien, Sandra Orchard, Martin Vingron, Bernd Roechert, Peter Roepstorff, Alfonso Valencia, et al. Intact: an open source molecular interaction database. *Nucleic acids research*, 32(suppl_1):D452–D455, 2004.

[57] C. Stark. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34(90001), Jan 2006.

[58] Ioannis Xenarios, Danny W Rice, Lukasz Salwinski, Marisa K Baron, Edward M Marcotte, and David Eisenberg. Dip: the database of interacting proteins. *Nucleic acids research*, 28(1):289–291, 2000.

[59] Christian von Mering, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, and Berend Snel. String: a database of predicted functional associations between proteins. *Nucleic acids research*, 31(1):258–261, 2003.

[60] Yungki Park. Critical assessment of sequence-based protein-protein interaction prediction methods that do not require homologous protein sequences. *BMC Bioinformatics*, 10(1), 2009.

[61] J. Yu, M. Guo, C. J. Needham, Y. Huang, L. Cai, and D. R. Westhead. Simple sequence-based kernels do not predict protein-protein interactions. *Bioinformatics*, 26(20):2610–2614, 2010.

[62] Leonardo G Trabuco, Matthew J Betts, and Robert B Russell. Negative protein–protein interaction datasets derived from large-scale two-hybrid experiments. *Methods*, 58(4):343–348, 2012.

[63] Long Zhang, Guoxian Yu, Maozu Guo, and Jun Wang. Predicting protein-protein interactions using high-quality non-interacting pairs. *BMC Bioinformatics*, 19(S19), 2018.

[64] István A Kovács, Katja Luck, Kerstin Spirohn, Yang Wang, Carl Pollis, Sadie Schlabach, Wenting Bian, Dae-Kyum Kim, Nishka Kishore, Tong Hao, et al.

Network-based prediction of protein interactions. *Nature communications*, 10 (1):1–8, 2019.

[65] Yu Zhen Zhou, Yun Gao, and Ying Ying Zheng. Prediction of protein-protein interactions using local description of amino acid sequence. *Communications in Computer and Information Science Advances in Computer Science and Education Applications*, page 254–262, 2011.

[66] Asa Ben-Hur and William Noble. Choosing negative examples for the prediction of protein-protein interactions. *BMC Bioinformatics*, 7(Suppl 1), 2006.

[67] Xiao-Yong Pan, Ya-Nan Zhang, and Hong-Bin Shen. Large-scale prediction of human proteinprotein interactions from amino acid sequence based on latent topic features. *Journal of Proteome Research*, 9(10):4992–5001, 2010.

[68] Xiuquan Du, Shiwei Sun, Changlin Hu, Yu Yao, Yuanting Yan, and Yanping Zhang. Deepppi: Boosting prediction of protein–protein interactions with deep neural networks. *Journal of Chemical Information and Modeling*, 57(6): 1499–1510, 2017.

[69] Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, et al. A proteome-scale map of the human interactome network. *Cell*, 159(5): 1212–1226, 2014.

[70] Y. Park and E. M. Marcotte. Revisiting the negative example sampling problem for predicting protein-protein interactions. *Bioinformatics*, 27(21):3024–3028, Sep 2011.

[71] G. R. Mishra. Human protein reference database–2006 update. *Nucleic Acids Research*, 34(90001), Jan 2006.

[72] Uniprot: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47 (D1), May 2018.

[73] L. Salwinski. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32(90001), Jan 2004.

[74] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, Jan 2018.

[75] Ahmet Sureyya Rifaioglu, Tunca Doğan, Maria Jesus Martin, Rengul Cetin-Atalay, and Volkan Atalay. Deepred: Automated protein function prediction

with multi-task feed-forward deep neural networks. *Scientific Reports*, 9(1), 2019.

[76] Angela Lopez-Del Rio, Alfons Nonell-Canals, David Vidal, and Alexandre Perera-Lluna. Evaluation of cross-validation strategies in sequence-based binding prediction using deep learning. 2018.

[77] Yuan-Miao Gui, Ru-Jing Wang, Xue Wang, and Yuan-Yuan Wei. Using deep neural networks to improve the performance of protein–protein interactions prediction. *International Journal of Pattern Recognition and Artificial Intelligence*, page 2052012, 2020.

[78] Yuanmiao Gui, Rujing Wang, Yuanyuan Wei, and Xue Wang. Dnn-ppi: A large-scale prediction of protein–protein interactions based on deep neural networks. *Journal of Biological Systems*, 27(01):1–18, 2019.

[79] Yu Yao, Xiuquan Du, Yanyu Diao, and Huaixu Zhu. An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*, 7, 2019.

[80] Hang Li, Xiu-Jun Gong, Hua Yu, and Chang Zhou. Deep neural network based predictions of protein interactions using primary sequences. *Molecules*, 23(8): 1923, Jan 2018.

[81] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.

[82] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[83] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

[84] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[85] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[86] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[87] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 7, 2017.

[88] Zheng Hu, Yongping Li, and Zhiyong Yang. Improving convolutional neural network using pseudo derivative relu. *2018 5th International Conference on Systems and Informatics (ICSAI)*, 2018.

[89] François Chollet. keras. `https://github.com/fchollet/keras`, 2015.

[90] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

[91] Yuanmiao Gui, Rujing Wang, Yuanyuan Wei, and Xue Wang. Construction of protein-protein interactions model by deep neural networks. *Proceedings of the 2018 International Workshop on Bioinformatics, Biochemistry, Biomedical Sciences (BBBS 2018)*, 2018.

[92] Xue Wang, Yuejin Wu, Rujing Wang, Yuanyuan Wei, and Yuanmiao Gui. A novel matrix of sequence descriptors for predicting protein-protein interactions from amino acid sequences. *Plos One*, 14(6), Jul 2019.

[93] Yanbin Wang, Zhuhong You, Liping Li, Li Cheng, Xi Zhou, Libo Zhang, Xiao Li, and Tonghai Jiang. Predicting protein interactions using a deep learning method-stacked sparse autoencoder combined with a probabilistic classification vector machine. *Complexity*, 2018:1–12, Oct 2018.

[94] Hita Sony Garapati, Gurranna Male, and Krishnaveni Mishra. Predicting subcellular localization of proteins using protein-protein interaction data. *Genomics*, 112(3):2361–2368, 2020.

[95] Supatcha Lertampaiporn, Sirapop Nuannimnoi, Tayvich Vorapreeda, Nipa Chokesajjawatee, Wonnop Visessanguan, and Chinae Thammarongtham. Pso-

locbact: A consensus method for optimizing multiple classifier results for predicting the subcellular localization of bacterial proteins. *BioMed Research International*, 2019, 2019.

[96] Motonori Ota, Hideki Gonja, Ryotaro Koike, and Satoshi Fukuchi. Multiple-localization and hub proteins. *Plos One*, 11(6), Oct 2016.

[97] P. Shannon. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, Jan 2003.

[98] Sminu Izudheen and Sheena Mathew. Identifying negative interactions in protein-protein interaction network using weak edge-edge domination set. *Procedia Technology*, 24:1423–1430, 2016.

[99] Philipp Blohm, Goar Frishman, Pawel Smialowski, Florian Goebels, Benedikt Wachinger, Andreas Ruepp, and Dmitrij Frishman. Negatome 2.0: a database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. *Nucleic acids research*, 42(D1):D396–D400, 2014.