Master in Informatics Engineering
Dissertation

PPRINT Prediction of Protein-Protein Interactions

Igor Nelson Garrido da Cruz
igorcruz@student.dei.uc.pt

Advisor:
Joel Perdiz Arrais
jpa@dei.uc.pt

Coimbra, 25th of June, 2014

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Acknowledgements

On a personal level, my first words of gratefulness must go to my supervisor, Professor Joel Perdiz Arrais, for all the advices and encouragement while accompanying me throughout this journey. Without his wealth of knowledge and friendship either as the most helpful suggestions to overcome difficulties, I would not be able to advance in my research.

On a professional level, this work was partially supported by iCIS CENTRO-07-0224-FEDER-002003. Also the Faculty of Dental Medicine from the Catholic University of Portugal had a big impact on this work for reasons of biological consulting, allocation of computing time and validation of the practical results on real datasets and for that we would like to acknowledge their support.

Abstract

Understanding life at a molecular level, while complex, encloses a myriad of opportunities for humanity's future. As important as being able to identify the molecular components of the cell it is of major relevance to understand their relationships and interactions. This way, the study of Protein-Protein Interactions (PPIs) has been used as a cornerstone to determine how most of the biological processes take place. Due to the large scale of the problem it is critical to use the appropriate computational tools and methods. Despite the existence of previous works in the field, the available methods are divided in two groups of approaches: experimental and computational. Experimental techniques have good prediction accuracy but are slow and expensive, therefore urges the need of developing computational approaches. These have low prediction accuracy but only require computational power and consequently are inexpensive since no laboratory machinery is required. A great amount of these algorithms are based on protein annotations, such as protein homology or protein domains. That makes such algorithms inapplicable to sparse multi-organism datasets usually composed only by the proteins sequences. In this work we start by analysing the existent state of the art methods for computational prediction of PPIs. It is our goal to explore their limitations and make improvements that can lead to more accurate results. After that we propose a new approach using the discrete cosine transform as a method of construction of features from the protein chain and a new method that calculates the three dimensional structure of the protein from its sequence. These new improved approaches will bequeath more accurate protein interactomes that can be used by Genomic Engineers in order to understand the intracellular structures relationship and biological processes. From these biological processes it is possible to extract semantic knowledge that can lead to new drug discoveries. Finally as a method of validation, our work is currently being experimentally validated by the Faculty of Dental Medicine from the Catholic University of Portugal from

the biological perspective using real sets of proteins extracted from humans saliva and from microorganisms presents in the oral cavity. It is also publicly available online for everyone to complement or use in other researches.

Key-Words: Bioinformatics, Protein Interaction Prediction, Protein Features, Machine Learning

Contents

List of Figures

Chapter 1

Introduction

The term bioinfomatics was coined by Paulien Hogeweg in 1970 to refer the study of the information processes in living organisms [1]. In the present days, bioinformatics is the name given to the research field that focus on the development and improvement of methods for storing and analysing biological data [2, 3]. One of the major characteristic of bioinformatics is the necessity to integrate diverse fields of knowledge. Informatics, mathematics, biology and statistics are used in order to analyse the biological data extracted from the diverse organisms or from their intracellular structures.

The main areas of research related with bioinformatics are sequence analysis, gene and protein expression, protein structure, and molecular networks. It is important to study the sequencing of genomes and their mutations, to understand protein structures and their interactions and to extract semantic knowledge from raw biological data in order to understand how the life works at its lower levels.

Proteins are biological molecules, composed by of one or more long chains of a simple organic compounds containing both a carboxyl and an amino group. These groups are called amino acid residues. Proteins perform a multitude of functions within the living organisms, among them the most important are catalysing metabolic reactions, replicating DNA, responding to stimuli, and transporting molecules from one location to another. Each protein differs from one another primarily in their chain

of amino acids, which is dictated by the nucleotide sequence of the genes, and which usually results in folding of the protein into a specific three-dimensional structure that determines its function.

Protein-Protein Interaction (PPI) is the process where pairs of proteins physically bind as a result of biochemical events and/or electrostatic forces in order to accomplish a biological function. These interactions are of critical importance because they modulate the cellular macromolecular structures and functions. Indeed they are the main mediators for several biological processes including the intracellular signalling pathways [4] commonly known as the the transmission of messages within different structures of the cells.

Having the knowledge of how proteins interact with each other can provide a great opportunity to understand pathogenic mechanisms, and subsequently support the development of new drugs, focusing on very specific intracellular structures or optimize already commercialized drugs re-targeting them to new gene products [5].

PPIs operate at almost every level of cell functioning [6]. From the definition of cell structures to the regulation of gene expression, all depends on the interaction of different proteins. This is why they became popular and object of so many different studies. Finding and labelling new interactions opens the door to new discoveries that vary from having detailed insights of how diseases are originated and how they can be prevented, to finding new medicines which can target some proteins and strengthen or weaken certain protein interactions in order to make the patients healthier, increasing their lifespan or even to completely cure them from untreatable diseases by the actual medicine.

With the present work our focus is upon the prediction of protein interactions using features extracted from their genetic sequences and their respective structures. We are going to study the limitations of the existent state of the art methods and try to develop new methods to overcome such limitations. As a final result, we have

developed a classification algorithm that is able to predict if pairs of proteins are susceptible to physically bind or not. This task is often denominated computational prediction of PPIs.

1.1 Motivation

The most standard way for detecting PPIs is throughout the usage of biological techniques. These methods test *in vitro* if a pair of proteins is susceptible to interaction. There are many experimental methods of predicting PPIs such as yeast Two-Hybrid Systems, Mass Spectrometry, Protein Microarrays and Fluorescence Resonance Energy Transfer [7], each one having his own advantages and disadvantages. Some of them, despite being very accurate, are monetary expensive and time consuming, others are more time efficient and consequently predict a higher amount of interactions per unit of time, however their results are sometimes inaccurate [8].

While predicting PPIs the adversities come not only from the fact that the predictions are hard to make, since specialists do not know what factors are directly related with protein bonding, but also because there is a large number of interactions to be tested. According to the last release of Uniprot [9] the human being has around 140,000 proteins (release Jul 9, 2014). If all pairs are to be tested there is an astonishing number of possibilities $\binom{140,000}{2} \simeq 9 \times 10^9$. In addition to the problem of testing all the human protein interactions the problem gains an even bigger dimension when the interactions that happen between the human and other organisms are considered. For instance, in the oral cavity there are evidences of around 2300 micro-organisms [10]. When considering the interactions of the human proteins with these micro-organisms proteins, the number of interactions to be computationally tested expands largely to untreatable numbers for the exiting computational power of recent computers.

The limitations described above have lead to a different approach to the prob-

lem. This approach consists in the development of computational techniques. These techniques are currently being used to allow researchers to more accurately find which proteins have a high probability of interaction for further testing with the experimental methods referred above.

There are several computational methods currently available. The most recent studies on this area already show interesting results (85.9% accuracy for Naive Bayes (NB) with Substitution Matrices [11] to 87.36% accuracy for Support Vector Machines (SVM) with Auto-Covariance (AC) [12].

While promising they present several limitations. Their results are sometimes not as valid as desired in these kind of researches, since the datasets used were not selected from official databases or sometimes not considered as a whole, but just small portions. The feature extraction strategies could be optimized by considering more factors that are related with protein interactions or even integrate diverse features in order to achieve better results or to overcome each others individual limitations. Also different methods can be used to test new concepts and hypothesis.

In addition the existent studies give focus to the prediction of interactions from intraspecies datasets, not considering PPIs from interspecies datasets, on which the precision is expected to decay since intracellular structures vary from organism to organism. In fact most of the published work restricts the datasets to a single organism, ignoring the fact that multi-organism interactions is a subject that needs to be explored since all of our body, but mainly oral cavity, nose, eyes, skin and gastrointestinal surfaces are environments in which a lot of micro-organisms are in constant interaction.

In sum, although the problem of identifying PPI have already been addressed both by the use of experimental and computational techniques, all the existent methods present major limitations such as high monetary costs, low throughput or accuracy of the results and therefore it still an open research problem to be studied.

1.2 Objectives

The methods described in the state of the art are applied and tested in intraspecies, and some times manually restricted datasets. With this work our objective is to be able to predict interactions between the proteins of different species with the best possible accuracy, since it will be used by the Faculty of Dental Medicine from the Catholic University of Portugal in order to discover protein interactions between bacteria and human immune system that occur on saliva. Their idea is to be able to predict *Periodontitis* and other dental deterioration diseases even before the patients show symptoms. So we need to predict PPIs happening between diverse organisms and the human saliva with the best possible accuracy. On most of the proteins that need to be experimentally tested only the amino acid chain is available, therefore our method should work based on this premiss.

Our first step should be to analyse the state of the art methods used to predict PPIs and apply them to our datasets in order to have a metric of comparison with the results of our methods on the same datasets. Then we should be able to find limitations present on the existent methods and propose new approaches that are able transcend such limitations.

With the current project we aim to build the PPRINT classifier that should be able to predict PPIs using machine learning techniques. This system input will receive pairs of protein amino-acids sequences as input and should output a value within the range [0,1], 0 representing lowest score of interaction between the pair and 1 representing the highest score of interaction between the pair of proteins.

Since there is no known simple mathematical rule that dictates if proteins are inclined to physically bind or not, different methods for extracting features from proteins primary sequence are going to be implemented and evaluated in this work. We should test and analyse the behaviour of different classifiers given the features extracted and choose the best classifiers and parameters to optimize the output results.

1.3 Document Structure

Here we present a brief description of the different chapters of this thesis and what is done in each one of them.

Chapter 1: Introduction This section. We show the importance of the work presenting a brief contextualization of the motivations behind the project and our objectives.

Chapter 2: State of the Art State of the art, biological background, work contextualization including background on machine learning and similar studies on the area.

Chapter 3: Methodology Description of the methodologies used to develop our approaches to the problem, allowing others to understand and implement our work.

Chapter 4: Results and Discussion Presentation and analysis of the results achieved with our work and further comparison with the existing methods.

Chapter 5: Final Considerations In the final chapter we conclude our work and make a reflection about the achievements of our research. We also present notes about some ideas that can be studied in future works that we think that are of great importance.

Chapter 2

State of the Art

With the present chapter we analyse the concepts necessary to the problem of predicting PPIs. The brief biological background presented in the next section aims to introduce the concepts to understand the work that took place in this thesis.

After presenting some biological concepts we formally define our problem and present diverse classification methods used previously in similar tasks.

On a later stage we present and analyse different feature extraction techniques used to extract features from proteins and fully describe some methods that have shown great results on previous works.

This analysis of the state of the art will allow us to have a better insight of the strategies used in similar tasks and what were the conclusions and limitations observed by the researchers, allowing us to achieve knowledge to posteriorly build our classifier more easily and efficiently.

2.1 Biological Background

Although this work is focused on the engineering and the mathematical concepts behind the prediction of PPIs, in this section we present an overview of the biological background required to understand the rest of the document.

2.1.1 Understanding the Role of Proteins

Proteins are complex molecules that play diverse critical roles in the bodies of living organisms. Most of proteins work is done within cells where they compose the intracellular structures and are responsible for tasks such as transport and communication. However, some proteins are not limited to control intracellular structures and are required for the structure, function, and regulation of the body tissues and organs. An example of this is the Growth Hormone which is a messenger protein that transmits signals between different cells to coordinate biological processes such as growth stimulation, cell reproduction and regeneration in humans and other animals.

Proteins are made of hundreds to thousands of smaller units called amino acids that are the basic organic compounds composed of amine and carboxylic groups. These units are attached to one another in long chains, defining different sizes of the proteins. Despite the existence of only 20 different types of amino acids when combined in these chains they are responsible for the generation of all the existent proteins since the order in what they appear and the number of amino acids in the each chain is variable. In terms of size proteins are classified as nanoparticles, having between 1 and 100 nanometres.

To produce a protein the cell first makes a copy of the DNA instructions called RNA. The RNA is then moved to the "ribosome", the intracellular structure responsible for assembling proteins. Every set of three RNA bases in a row controls which amino acid is to be added to a growing protein molecule. The copy of the DNA chain passes through this protein making machinery "like a tape through a tape player", assembling individual amino acids into proteins ready to be used by the cell or by the host organism.

Proteins structure can be be seen as primary, secondary, tertiary or quaternary. The proteins primary structure refers to amino acid linear sequence of the polypeptide chain as described above. The secondary structure refers to highly regular local

sub-structures and represents the hydrogen bonds that link the amino and carboxyl groups. Protein's tertiary structure corresponds to its three-dimensional structure and it represents the atomic position of each atom in three dimensional space. Finally the quaternary structure represents the multiple possibilities of folding the protein and the diverse shapes and structures that it can have. Many proteins are actually assemblies of more than one polypeptide chain, the quaternary structure represents the arrangement into which these subunits assemble.

The use of higher levels of representation adds more information about proteins and their amino acids, however they also requires a lot more computation time to be calculated.

Fig 2.1 presents a simple diagram that represents these 4 levels of structural analysis is shown.

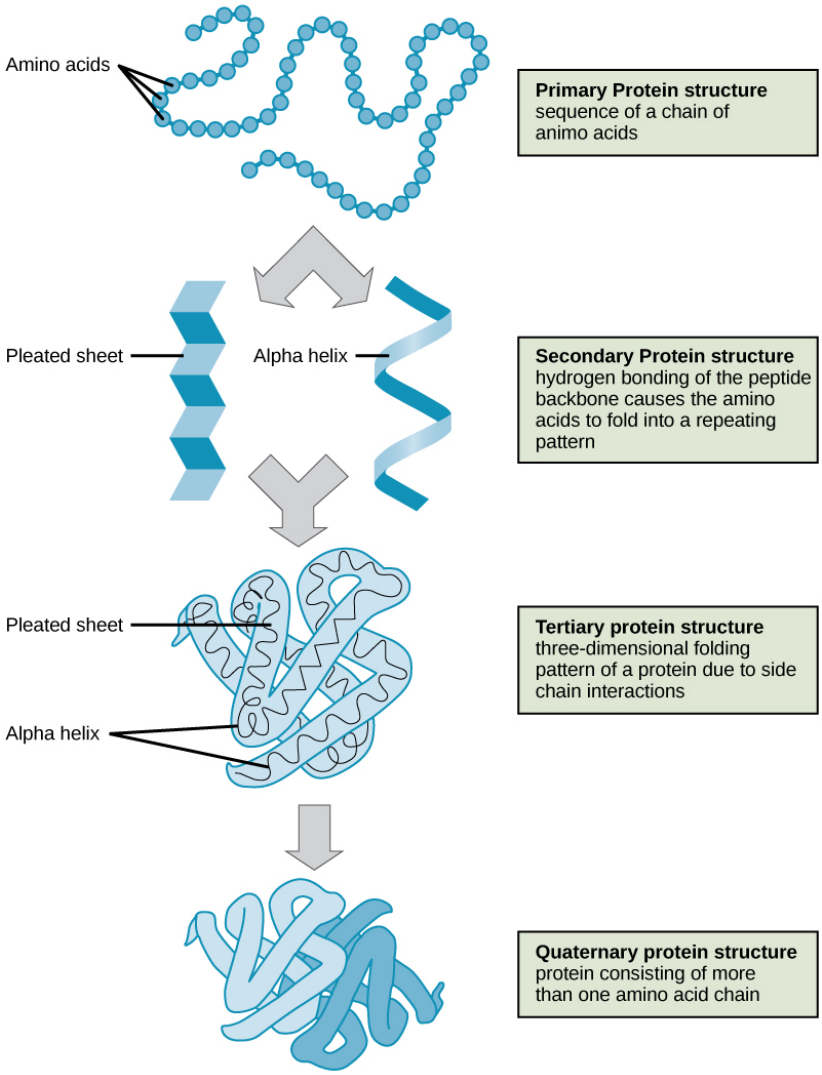


Figure 2.1: The four levels of protein structure, (source: OpenStax College, http://cnx.org/content/m44402/latest/?collection=col11448/latest)

2.1.2 The Role of Protein Interactions

Being proteins the main entities responsible for life at lower levels, it is important to understand how they work and how they are related with each other. Researchers found out that some of the cells responsible for our immune system have proteins that work as sensors in order to detect intruder bacteria and proceed with the appropriate immunological responses in order to activate the appropriate mechanisms for

the defence of the host organism.

Latest discoveries in bioinformatics allow the development of new drugs that inhibit or fix some proteins in order to tackle pathologies. For instance the origin of some diseases such as Alzheimer, Parkinson or Huntington's diseases is related with errors in the structure of proteins. Oregon Health & Science University researchers developed a new drug approach that could lead to cures for wide range of diseases [13] on which these ones are included. The new drug would fix the broken proteins restoring their normal structure and consequently healing the patients.

Being vital macromolecules for the organisms correct functioning, proteins rarely act alone. Multiple essential molecular processes that happen in an organism are carried out by molecular machines that are built from a large number of protein components organized and interconnected by their PPIs. Indeed, these interactions are at the core of the entire balance of any living cell. Consequently errors happening at the PPI level are on the basis of multiple diseases, such as Creutzfeldt-Jacob and cancer.

Due to its importance to living organisms arises the need to understand how these proteins interact with each other and what cellular structures are regulated by them. The final consequence of understanding PPIs is to attain interaction networks in the form of a graphs or interactomes [14]. Such interactomes will allow molecular biologists and specialists in genetics to more easily target individuals (proteins, pharmaceutics, DNA, mRNA, etc) for their studies and to understand what are the relationships between them or what is the influence that they inflict on each others.

This is why it is so important to understand how proteins interact, because such interactions can give us insights of how the cells are structured and how the different biological processes take place inside our bodies. The main goals of these interactomes are the development of a functional maps of cell's processes, drug target identification and to predict the functions of uncharacterised genes or proteins. Having an accu-

rate method or process of predicting interactomes of a bio-organism can boost the probability of discovering new drugs to tackle known problems or even understand the intracellular biological processes that take place in case of pathology.

After the release of the first draft sequence of the human genome in 2001 by the project named Human Genome Project (Lander *et al.* [15]), researchers are now more focused on protein interactions, rather than sequencing the human genome. In 2003, when the project was declared finished the number of human protein-coding genes was estimated to be 24,500. Although the actual number is a lot bigger than the initial estimation (now being around 140,000) due to alternative splicing, a process that makes a single gene responsible for coding multiple proteins. Therefore comes the need to develop machine learning algorithms able to predict interactions with good sensitivity, specificity and high throughput in order to speed up the process of learning relationships between proteins and reduce the costs of *in vitro* experimental tests, focusing on protein pairs with high probabilities of interaction.

An important fact about the actual state of the art is that most of the studies are focused on intra-species datasets and that is not the target of this work. As said above, the human genome is responsible for coding around 140,000 proteins, however our bodies are not isolated beings. By this we intend to emphasize that throughout all of our bodies we are in constant interactions with other organisms, being the mouth and the intestines the organs with higher amounts of them. These organisms are also composed by proteins and PPIs are also present between their proteins and ours affecting the normal functioning of each other organisms.

The high amount of protein combinations to be tested aside with the lack of information available about the genes of some micro-organisms opens the door to optimize the computational methods that use polypeptide chain as input.

2.1.3 The Importance of Primary Structure

There are a several computational based methods of predicting PPIs. In the present work our focus in upon the proteins primary sequence of amino acids and their structure. These methods should able to predict PPIs only from the amino acid sequence of proteins with a good performance, either in terms of reduced computational time and with with a relatively high accuracy.

The information extracted from the primary structure of proteins with simple metrics such as physico-chemical properties or amino acid counts or even amino acid distributions should be enough to predict some interactions between pairs of proteins. This method has some advantages: It can be applied to all of the protein sequences; It does not require the three dimensional structure or other high complexity processing neither other biologic based information such as the location of the proteins in the intracellular space. The disadvantages of these methods are the lack of specificity of the results. On one hand, these methods can be applied to any protein from any given organism, however there is no detailed information about each protein, such as intra cellular location, that could make the data more specific and facilitate the classification task. Joel R. Bock and David A. Gough [16] studied PPIs using the primary structure and concluded that on average 80% could be correctly predicted, only using these techniques.

On other hand the three-dimensional position of atoms is more discriminatory of proteins and consequently it could bring better prediction accuracy to the classification methods, however it has to be calculated using algorithms with high computation complexity that become time expensive for bigger datasets.

The Fig. 2.2 presents an example of a protein's three-dimensional structure. This structure represents a *dehydrogenase* enzyme from the bacteria *Colwellia psychrerythraea*. The enzyme is capable of generating harmful reactive oxygen species and has been implicated in neurodegeneration, ischemia-reperfusion, cancer and several other

disorders.

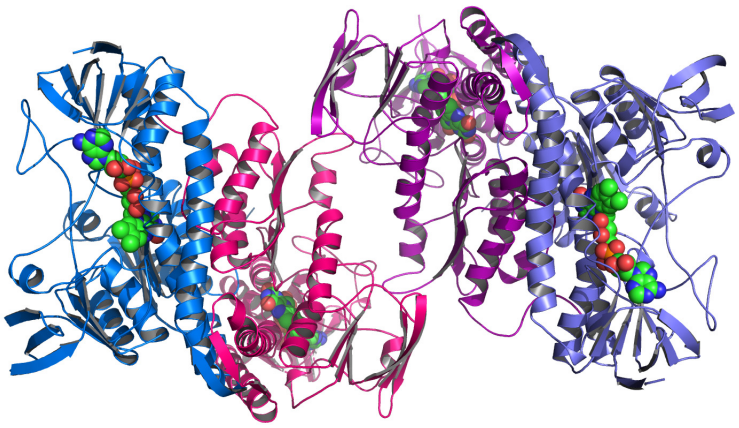


Figure 2.2: Protein's 3-D structure of *dehydrogenase* (source: Flicker, http://www.flickr.com/photos/argonne/3762337272/)

2.2 Classification Problem

In this section we will leave aside the experimental methods and the biological background behind this task and will focus our attention on the state of the art computational methods and algorithms used to predict PPIs. This methods have followed a multitude of perspectives and have varied much since the start of this field of investigation.

In machine learning and statistics, classification is the problem of identifying to which of a set of categories or classes a new observation belongs, on the basis of a training set of data containing observations whose category membership is known. These observations consist in features that are quantifiable properties extracted from each observation.

Considering that we want to computationally verify if pairs of proteins are susceptible to interaction or not, we have a binary classification problem. In Machine

Learning a binary classification problem is defined by:

$$f : \mathbb{R}^{\mathbb{N}} \rightarrow y, \text{ where } yi \in [0,1] \tag{1}$$

Applied to our case, $\mathbb{R}^{\mathbb{N}}$ is the whole dataset of features extracted from the protein pairs, $y = 0$ defines the not interacting class and $y = 1$ defines the interacting class. f is the machine learning method that transforms the input features in output data y. The learning method uses input-output training data:

$$D = \{(xi, yi) \in S \subseteq \mathbb{R}^{\mathbb{N}}\} \tag{2}$$

such that f correctly classifies unobserved data (xi, yi) that was not introduced during the training.

xi represents the features of a pair of proteins to be tested, yi represents the desired output for the classification problem. S is used as notation to describe a single pair (xi, yi) from the entire dataset $\mathbb{R}^{\mathbb{N}}$.

In machine learning there are diverse strategies used to build statistical classifiers. It is important to make a brief description and analysis of some classifiers that are most used in the bioinformatics field, in order to demonstrate their power and what they are being used for. During this work some these algorithms are going to be implemented to build our final classifier.

2.2.1 k-Nearest Neighbour

The k-Nearest Neighbour (k-NN) is a very simple machine learning algorithm.

In a given classification problem, the output classification for the input features is calculated by a measure of proximity between the input and the closest k patterns

provided by the training set. A pattern is classified by a majority vote of these neighbours classes. Despite being so simple this method is used for a multitude of problems and sometimes present interesting results.

k-NN has been used in the bioinfomatics field before to predict secondary structures, to predict protein functioning and even to predict PPIs. Mario R. Guarracino and Adriano Nebbia [17] were able to correctly classify protein interactions with an accuracy of 98.11%. However the dataset used by them was independently selected and consisted only in 3,291 interactions which in our opinion is not a representable amount of interactions.

2.2.2 Neural Networks

Artificial Neural Networks (NNs) are data structures that approximate the operation of the human brain. The models consist on interconnected neurons with weights in the links between them. In fact these connections simulate synapses (transmission of information between the different neurons of the brain). The organization and weights of the connections determine the output.

NNs are currently used prominently in voice recognition systems, image recognition systems, industrial robotics, medical imaging, data mining and aerospace applications.

The Fig. 2.3 presents a diagram that represents a NN. On the left, represented with green background there are the input features and on the right with blue background there is the output layer. The data flows from the input layer to the output layer passing in the hidden layers and being multiplied by different weights that are adapted in the learning process.

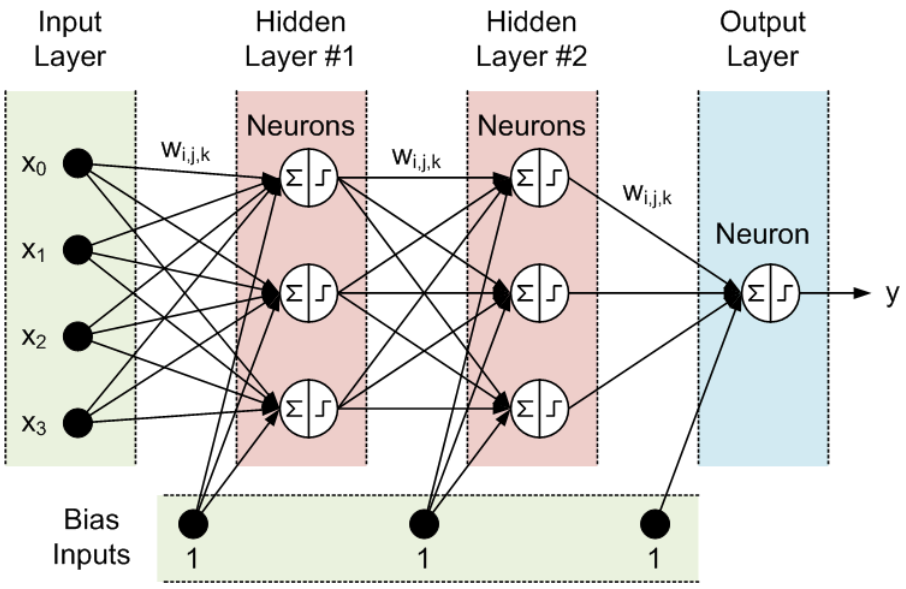


Figure 2.3: Neural Network Structure (source: Next price predictor using Neural Network, http://codebase.mql4.com/5738)

To predict protein interactions Neural Networks are also very effective, Jae-Hong E. and Byoung-tak [18] built a NN that out-performs other state of the art algorithms in a independent *Saccharomyces cerevisiae* dataset, achieving 91.4% accuracy. After that Jae-Hong Eom and Byoung-Tak Zhang improved the feature extracting mechanism and could achieve 96.1% accuracy using the same dataset [19].

Despite the good results achieved, this method uses additional information that is not provided only by looking at a proteins primary structure.

2.2.3 Naive Bayes

The naive Bayes classifiers are simple probabilistic classifiers. In fact they are named naive because they assume that all the feature variables present in a given pattern are independent.

In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other

feature [20]. For example, a protein pair may be considered to be interacting if the sum of proteins amino acids is higher than a given value and the polarity of each one is contrary to each other. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this pair is interacting. However that does not modulate the real world problems, since a lot of variables chosen as description for the datasets are dependent.

Despite of the simplistic approach and assumptions, naive Bayes is one of the most efficient and effective inductive learning algorithms for machine learning and data mining [21].

In Protein Interaction context, naive Bayes has been used in some works in the past. The University of Dundee has a database of predicted human protein-protein interactions, in which the predictions have been made using a naive Bayesian classifier to calculate a score of interaction [22, 23].

In the bioinformatics field Chishe Wang *et al.* [8] also used a Bayesian classifier in order to identify PPI sites (three-dimensional locations where each protein binds with another), the results were round 60-65% accuracy.

2.2.4 SVM

Traditional machine learning algorithms such as Neural Networks use the empirical risk as a minimization objective, this objective function leads to accurate results in the well known patterns, but also reduces the classifier ability to generalize since the problem's structure is not taken in account.

SVMs were initially proposed by Vladimir N. Vapnik in 1964 and were posteriorly work on by Corinna Cortes and Vladimir N. Vapnik in 1995 to allow the existence of non-linear separable patterns [24].

A SVM is a representation of the patterns as points in space, mapped so that

the patters of the separate categories are divided by a clear gap that is as wide as possible (the support vectors). New patterns are then mapped into that same space and predicted to belong to a category based on which side of the margin they fall on. The SVMs differ from the other methods because the principal concept behind them is to calculate the optimal structure in order to perform a structural risk minimization maintaining a good generalization performance.

The Fig. 2.4 is a graphical representation of the concept behind SVMs. On the left, the error generated by underfitting (the classifier was not able to learn the correct data model) and, on the right side, the error generated by overfitting (the classifier is well trained for the training data, but is not able to generalize outputs for new data). The idea here is to reach the optimal structure which is somewhere in the middle of the both these concepts.

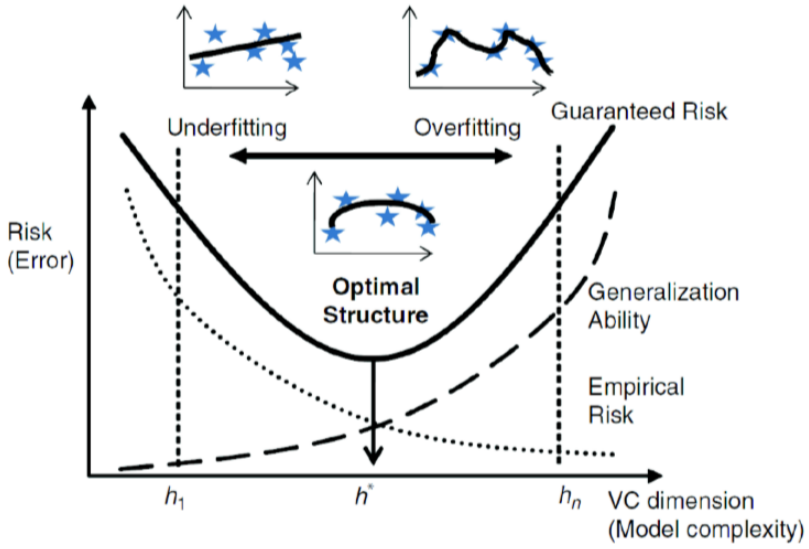


Figure 2.4: Generalization ability in dependence of VC-dimension h (source: Bernardete Ribeiro (2013), Pattern Recognition Techniques Slides (2013))

When applied to the PPI prediction task, SVMs have shown the best results. Citing Shinsuke D. and Asako K, Toshihisa T.: "SVMs are very useful in that domain information and several protein features including amino acid composition, sequential amino acid usage, and localization, whether they are continuous or discrete values, can be easily combined into a feature vector, and take them all into account" [25].

SVMs were used on some PPI prediction methods that are explained in more detail in the upcoming sections. More emphasis was given to that classifiers since they will be implemented for further testing and optimization.

2.2.5 Other Classifiers

There are other classification methods available, Linear Discriminant Analysis (LDA) classifies data making a classification decision based on the value of a linear combination of the characteristics provided. Quadratic Discriminant Analysis (QDA) separates the samples by a quadratic surface. In bioinformatics is rare to find studies that implement LDA or QDA as classifiers and for that reason they are not studied in more detail.

Also decision trees are used, but the results are usually not great in comparison with other methods. Most of these classifiers are sometimes too much simplistic and do not achieve the desired performance in PPI classification tasks.

2.3 Feature Extraction

A feature is an individual measurable heuristic property. When dealing with a classification problem it is important to use the most descriptive features in order to build the best possible classifier. The more discriminating and independent a given feature is, the best. This happens because it allows the classifier to discern between classes more easily.

One perspective of feature extraction was used by Rodriguez-Soca *et al.* [28, 29] in 2009 when he used the publicly available tertiary structures of *Trypanosome* (parasites well known for causing sleeping sickness and Chagas disease) proteins to build an artificial neural network using the linear activation function, achieved results of 90.9% accuracy. They also performed similar strategies on *Plasmodium*(responsible for malaria) and scored 96.8% accuracy on the validation data.

Patrick Aloy and Robert B. Russell [30] also built a classifier to predict protein interactions based on their tertiary structure. Given a pair of proteins, they search for homologue's in a database of interacting domains of known three dimensional structures and inference if there is interaction from that information.

However the tertiary structure of proteins is only available for a small amount of proteins, since complex and consequently slow algorithms are used to generate such models, so theses methods are not the best ones, considering that a classifier able to classify a lot of different proteins from interspecies dataset is needed.

In alternative to these high complex structures, physico-chemical properties of amino acids can also be used to study protein sequence profiles, folding and function [31].

In 2012 Xiao *et al.* [32] published protr, a state of the art library for protein sequence extraction methods for the R language. It focus in the implementation of methods of extraction of protein sequence information for other researches usage. These features are the Amino Acid composition (Amino Acid, Dipeptide and Tripeptide), Auto-correlation, Composition, Transition, Distribution and Pseudo Amino Acid composition because they are usually implemented in bioinformatics tools for protein studies and PPI prediction.

In order to correctly describe each protein for the classifier to be able to learn if PPI are going to happen or not, features are going to be extracted from the FASTA format correspondent to each protein.

2.3.1 The FASTA Format

Despite some methods making usage of three-dimensional structures or other features that require high amounts of computation, in this study mainly the primary sequence is going to be used, as ways to define and describe proteins for the sake of using less amount of computation.

Being our final objective the prediction of PPIs between different organisms we have to stick to these representations, since more advanced features are not usually studied for inter-species datasets, and consequently there is not enough consistent information about diverse organism's proteins.

The genetic code is the set of rules by which information encoded within genetic material (DNA or mRNA sequences) is translated into proteins by living cells. Groups of three nucleotides are grouped in order to discern what amino acids to use. This identifiers of amino acids are usually known as codon.

In order to represent the sequence that composes a protein the most standard representation is the FASTA format. FASTA format is a text-based format that uses single-letter codes to refer to each individual amino acid derived from the groups of three nucleotides acids. The format also allows for sequence names, comments, species name and additional information to precede the sequences. The format originates from the FASTA software package [33], but has now become a standard in the field of bioinformatics.

Each amino acid as a corresponding single-letter code in FASTA code as shown in the table 2.1:

Table 2.1: FASTA Supported Codes

Amino acid Code	Meaning
*	translation stop
-	gap of indeterminate length
A	Alanine
B	Aspartic acid or Asparagine
C	Cysteine
D	Aspartic acid
E	Glutamic acid
F	Phenylalanine
G	Glycine
H	Histidine
I	Isoleucine
J	Leucine or Isoleucine
K	Lysine
L	Leucine
M	Methionine
N	Asparagine
O	Pyrrolysine
P	Proline
Q	Glutamine
R	Arginine
S	Serine
T	Threonine
U	Selenocysteine
V	Valine
W	Tryptophan
X	any
Y	Tyrosine
Z	Glutamic acid or Glutamine

2.3.2 Amino Acids and N-Grams

Amino acid composition is a simple feature extraction technique. It consists in counting the occurrences of amino acids on the protein chain. This method gives us features looking at the protein as a whole, sometimes leading to insufficient information to determine if two proteins interact or not.

N-Grams consist in sub-sequences of the amino acids that compose the protein with predefined length N. Such sequences are built using a sliding window along all the proteins chain.

N-Grams is a technique that can be used to improve the performance of the amino acids counting. In fact the method described above is a particular case of N-Grams with N =1. With N = 2 (Dipeptide composition) and N = 3 (Tripeptide composition) there is more information available for the algorithm to deal with and usually better results.

Despite being simplistic N-Grams is commonly used as text and speech feature extraction techniques, being one of the most used strategies for these tasks.

When applied to the proteins, the results of these methods are not the best in terms of performance since they lose locational information, neither consider long range bonds between amino acids which are important to predict interactions. Sometimes the noise added by number of features extracted causes the performance to deteriorate while increasing the length of the N-Gram.

2.3.3 Domain Composition

With the recent advances in the field the known methods have become faster and a large amount interacting proteins is now known. This allows the researches to focus more on the inner parts of each protein, rather than looking at proteins as a whole. By focusing more on proteins, domains where found. Domains are conserved parts of proteins amino acid sequences, structures that can evolve, function and exist independently of the rest of the amino acid chain.

Protein domains were first proposed in 1973 when scientists studied the structure and function of immunoglobins [34]. Up to the date, these domains are distinct functional and/or structural units in a protein, they have a compact three-dimensional structure and some of them can be independently stable and/or folded. Most of the times these domains are responsible for particular functions or interactions, contributing to the overall role of proteins. Proteins with similar domains can have the most different functions.

These domains work like sub-divisions of the protein structure. Domains are important since a high quantity of protein interactions occurs between domains that compose each of the involved proteins [35, 36]. When building a machine learning classifier that predicts PPIs using domains there is the necessity to divide proteins in their domains and furthermore extract information for classification from each one of the involved domains.

The Fig 2.5 represents two FASTA code sequences that could represent two proteins and a possible interaction between domain A and domain B. The idea behind this approach is that proteins are composed by structures that can define interactions just by themselves. Similarly to PPI, in domain-domain interactions there is no known single mathematical rule or formula that dictates if interactions exist or not.

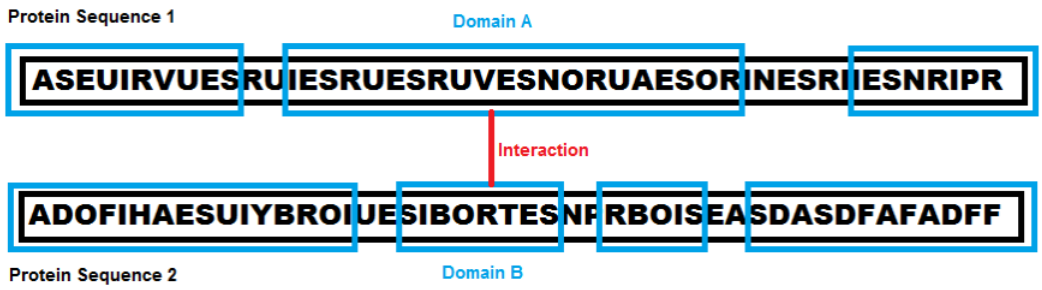


Figure 2.5: Domain Interactions Diagram

Shawn *et al.* [37] implemented a similar approach, employing a relatively simple model that learns dynamically from a large collection of data. They projected an "attraction-repulsion" model in which the interaction between a pair of proteins is represented as the sum of attractive and repulsive forces associated with small, domain or motif-sized features along the length of each protein. They used Hidden Markov Models to extract protein domains, and to extract the E-values of each domain [38] choosing the best match. Then they represented amino acid sub-sequences reduced from 20 to 6 categories of biochemical similarity as studied by Taylor and

Jones, 1993 [39]. After various testing environments the best performance achieved was 0.818 Area Under Curve (AUC) with a standard variation of 0.011, not making any reference to accuracy.

Pagel *et al.* [40] defend that the conserved domains carry many of the functional features found in the proteins of an organism. In their opinion the domains define not only the protein functions and but also molecular adapters. Such adapters are responsible for interactions between proteins and/or proteins and other molecules. They also defend that domain interaction is defined not by physical contact, but by common functioning.

Regarding the classification problem of PPIs, the use of domains needs to be considered since they are compact three-dimensional structures and often can be independently stable and folded and seen as parts of proteins.

2.3.4 Physico-chemical Properties

When the topic is physico-chemical properties it is possible to consider the properties of a protein as a whole or the properties of each one of its amino acids individually and build vectors with such properties. These vectors are like signals of the properties along the protein chain. The first method does not provide enough information to make correct assumptions about PPIs of the proteins studied since it falls back on a small amount of physico-chemical properties to describe each protein, not providing detailed information about its composition or structure. On other hand, the second method provides information enough to predict some PPIs since the amount of information that can be extracted from the chain of amino acids is greater.

Different proteins have different lengths. So, in order to classify pairs of proteins a method of extracting features from these signals is going to be used for further acquiring the same amount of indicators for both proteins of the interacting pair. In

order to do this, metrics like sum, average or weighted average can be used, since they equalize the number of features of each protein.

The following approaches (Shen and Guo) are state of the art methods that use physico-chemical properties information in order to predict PPIs. In the present work are going to implement and explain them in more detail.

On one hand they are the state of the art methods to predict PPIs using physico-chemical properties (providing more information than the classic N-Grams approach) and on other hand their functioning is based upon the proteins primary sequence (a requirement for our final classifier). These methods have shown good results while tested with the developers independent datasets as described below.

2.3.4.1 Shen Classification Strategy

The usage of physico-chemical properties was introduced by Shen *et al.* [41] in 2007 when he introduced the idea that isolated information of each amino acid is not relevant and consequently there is the necessity of relying on methods that somehow evaluate how the amino acids interact with each other. To his study he used as features the physico-chemical properties of amino acids such as the hidrophobicity (the capacity of repelling water) or hidrophilicity (the capacity of having affinity for water) building a substitution table with 7 categories and afterwards combining these categories with the conjoin triad method, a method which considers three continuous amino acids as a unit obtaining promising results. In his study Shen scored 83.90% accuracy on a human restricted dataset.

The results achieved by the process were good, however focusing on the two more proximate amino acid, makes the classifier susceptible to close range interactions between amino acids leaving out the long range interactions that are also important to predict PPIs in the complex biological world.

The proteins primary sequence is substituted by the physico-chemical categories using the substitution table (Attribution letters to colors in Fig 2.6). After that the groups of 3 categories in each sequence are counted and the amount of occurrences each triplet is normalized using the standard normalization building a proteins feature space. In order to represent pairs of proteins the feature spaces of two proteins are concatenated into a vector of features. Finally a SVM classifier is used in order to classify the whole dataset.

The Fig. 2.6 presents a diagram that elucidates how features are extracted from the proteins primary sequence.

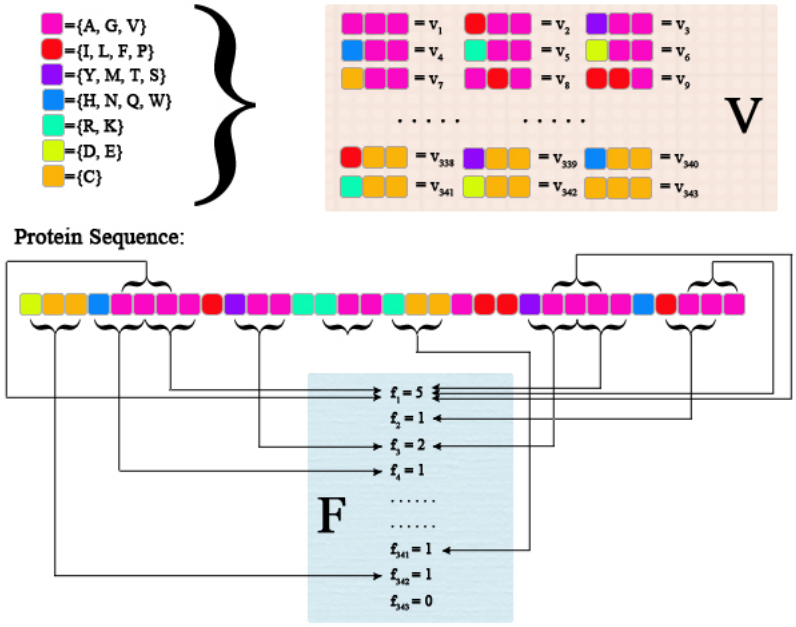


Figure 2.6: Constructing the feature space of a proteins sequence. Extracted from [41] Source PNAS: http://www.pnas.org/content/104/11/4337/suppl/DC1, available in July 2014

2.3.4.2 Guo Classifier

Similarly to Shen, the Guo classifier is going to be explained in detail since they are going to be implemented and we are going to use their results as a baseline to be

transcended by our final methods.

Guo implements a different approach that considers the amino acid properties of the protein chain as a signal. The intention was to overcome Shen's limitation that consists in leaving out the long range interactions.

In the year of 2008, using support vector machines combined with auto covariance to predict protein-protein interactions from protein sequences [12] in an independent dataset of 11,474 interactions from the yeast *Saccharomyces cerevisiae* (used to wine-making, baking, and brewing since ancient times) the authors achieved 87.36% accuracy, when applied to predicting the PPIs. As physico-chemical properties of the amino acids they used hydrophobicity, hydrophilicity, volume of side chains, polarity, polarizability, solvent accessible surface area, net charge and index of side chains. The way that the researchers implemented the auto-covariance as features is extremely meaningful since that for a given window n, they store the auto-covariances of the amino acid physico-chemical properties vectors from 1 to n. These features allow the classifier to learn the influence of short range amino acid relationships, but also the long range interactions, overcoming the strategy previously studied by Shen*et al.*

The method works as follows: when in presence of a protein sequence 7 signals are built substituting each amino acid by the information present in the table S1 present in the Annexes. Then for each one of the 7 signals the auto-covariance of the signal is calculated in a window from 0 to 30. 0 giving us a measure of the amino acid influence on himself, 30 giving us the influence of each amino acid 30 positions ahead.

Each one of the 30 values are then concatenated with the other for each physico-chemical property. A protein is described by a concatenation of all the values for all the physico-chemical properties. Finally, in order to represent a pair of proteins, information from two proteins is concatenated into a single feature vector for training and/or classification purposes.

Auto-covariance (AC) is the covariance of the variable against a time-shifted version of itself. AC is given by:

$$AC_{lag,j} = \frac{1}{n-lag} \sum_{i=1}^{n-lag} \left(x_{i,j} - \frac{1}{n} \sum_{i=1}^{n} x_{i,j}\right).\left(x_{(i+lag),j} - \frac{1}{n} \sum_{i=1}^{n} x_{i,j}\right) \tag{3}$$

This method is the one described in the state of the art that predicts PPIs from only the protein's primary structure with most accuracy.

Chapter 3

Predicting Protein-Protein Interactions

3.1 Task Overview

The present work has as objective the development of a computational method able to predict PPIs between proteins of different species. Our method will be available to be used by the community.

We will implement some existing methods, try to improver their performance, verify existing limitations and finally explore new methods that can perform better than the existing in terms of classification accuracy.

During the development of the work multiple challenges come up. The size of the datasets, the computational complexity of the feature extraction methods or the efficiency of the features used are some examples of these challenges.

When building the classifier of PPIs the main problem is that we do not know what path to follow in order to achieve the desired results, since there is a multitude of factors that influence proteins interactions.

3.2 Summary

A Gold Dataset was built in order to allow comparison between all the methods available in the state of the art and the methods developed. This dataset is described in more detail in the following chapters.

In order to build our final classifier several approaches were followed.

We didn't exactly know how the existing classifiers were going to perform and neither had the guarantee to be able to achieve a classifier that could outperform the existing methods.

So we proceeded as following:

1 - Implementation of the best state of the art methods that could be applied to the existing data;

2 - Trying to improve the state of the art methods implemented;

3 - Combining classifiers in order to optimize results;

4 - Verifying limitations in the implemented methods;

5 - Developing our final classifiers.

In order to implement machine learning algorithms we used the scikit-learn for Machine Learning in Python which provides simple and efficient tools for data mining and data analysis.

During our attempts to optimize and combine the state of the art methods we verified that looking at proteins as a whole could be the most beneficial way to approach the problem. So, our final computational methods are classifiers that use new feature extraction techniques that represent proteins as a whole, allowing the classifiers to learn interactions base on proteins structure. The final tools are two classifiers, one that uses the discrete cosine transform in order to represent proteins chain and another that builds the three-dimensional model of proteins and extracts information from there. However also the combination of multiple classification methods revealed itself advantageous in comparison with existing methods.

3.3 PPRINT Classifier

3.3.1 Generating Datasets

The datasets used are explained in detail in the results section. However, in this section a description of the methods used in order to generate the datasets is made.

The datasets used for training and testing the state of the art methods correspond to independent datasets not publicly available, since they are usually not provided. In addition to this, most of the researches about protein interactions are limited to intraspecies interactions and classifiers are not trained with examples from interspecies datasets. Because of this, we need to use sub-sets of existing large datasets, analyse the behaviour of the existing methods and then develop our own methods.

In order to build the datasets the required biological data was collected from UniProt and BioGRID.

UniProt is the central hub for the collection of functional information regarding proteins. Each entry contains the amino acid sequence, protein name, taxonomic data as well as supplementary annotations such as ontologies, classifications, cross-references, and clear indications of the quality of annotation in the form of evidence attribution of experimental and computational data. BioGRID is an online interaction repository with data compiled through comprehensive curation efforts. The current version compiles 42,004 publications for 720,840 raw protein and genetic interactions from major model organism species. All interaction data are freely provided through search indexes and available via download in a wide variety of standardized formats. Contrasting with other interaction databases, BioGRID provides protein interactions for multiple organisms.

BioGrid is our source of protein interactions and Uniprot is our source of proteins amino acid sequences for the proteins used.

Due to the large amount of protein combinations possible, its only possible to

represent a limited amount of the whole representation space. So in order to build datasets for training and testing of the computational methods, we used randomly selected subsets of the known protein interactions available on the described tools and joined them in different datasets with the same amount of positive and negative interactions. However for the classifiers to be able to predict if pairs of proteins do not interact we have to use datasets with negative examples as described in the following subsection.

3.3.2 Generating Negative Interactions

A recurrent problem in this kind of work is the absence of negative interactions since only the positive interactions are published. In other words, when a pair of interacting proteins is found, researchers publish the discovery on multiple online platforms available for that matter, but examples of not interacting pairs are hardly find.

Negatome [42] is the exception to the rule. It is a collection of protein and domain pairs which are unlikely engaged in direct physical interactions. The database currently contains experimentally supported non-interacting protein pairs derived from two distinct sources: by manual curation of literature and by analysing protein complexes. However the protein pool covered by this dataset is small.

In order to overcome this problem there are diverse methods of generating not interacting pairs described in the literature [43].

In the present work we are going to use Negatome and randomly selected pairs of proteins from the pool of proteins and establish them as not interacting proteins. For instance in the human body the possible combinations of proteins are of $\binom{140,000}{2}$ $\simeq 9 \times 10^9$ pairs, although only 237,498 interactions are actually known (Uniprot 3.2.115 - August 2014). So, the probability of randomly choosing a pair of proteins and it interacts is of 0.000026 of the whole set. So because of the high number of

proteins in the pool, the probability of a pair of randomly selected proteins interact is very low.

Using both Negatome and randomly selected pairs, we can build multiple datasets with different sizes and make sure our method works independently of the dataset used and is not under overfitting conditions.

3.3.3 Implementing State of the Art Methods

We started this work implementing existing methods and applying them to our datasets to observe how they would perform.

The simplest method available for this task is to use N-Grams of the chain as features (as described in the state of the art chapter), so we started by implementing some N-Grams approaches (with length = 1,2,3,4) to serve as baseline of classification performance. When we use N-Grams with length 3,4 the amount of features is too big, for the classification task, so we chose the 1000 features with higher term frequency in the whole dataset in order to represent proteins.

The state of the art methods implemented were the one proposed by Guo *et al*, called "Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences" and the one proposed by Shen *et al*" Predicting protein-protein interactions based only on sequences information".

Despite the existence of other methods available to predict PPIs, these are the ones that use reliable datasets for validation and the ones that predict interactions based only on the proteins primary sequence which is a requirement for our project.

3.3.3.1 Our First Methods

After implementing the Guo's method and watching its results. We focused our attention in searching and developing new ways of extracting features from the physico-chemical properties of amino acids, so our first method emerged. For each

protein we subdivided the amino-acid chain in various equal parts and for each one of the parts we used the physico-chemical properties of amino acids (present in the table in the annexes). For each one of these parts the features are the average of each physico-chemical property hydrophobicity, hydrophilicity, volume of side chains, polarity, polarizability, solvent accessible surface area and net charge index of side chains. Features are then concatenated.

In parallel with this approach another one was tested. For each amino acid belonging to the protein, we extracted the index on which it occurred, and then normalized it in relation to the length of the protein in the interval [0,1]. 0 representing the first position of the protein and 1 the last.

For the strategies presented above pairs of proteins are represented by the concatenation of features of two proteins and the classification task a SVM classifier was used.

3.3.3.2 Improving The Shen Classifier

The conjoint triad method proposed by Shen, considers psychico-chemical properties of one amino acid and from its neighbours regarding three continuous amino acids as a unit. It uses 7 amino acid categories making a total of (7*7*7) = 349 features.

We tried to improve this method reducing the number of considered amino acids to a window of 2 amino acids and adding a supplementary counting of the occurrences of 2 amino acids with intervals of 1. Later we also appended to these features the counting of the occurrences of 2 amino acids with intervals of 2.

The Fig 3.1 demonstrates how to proceed. On top we have the standard method proposed by Shen *et al* on mid our first idea to improve the method considering amino acid categories that are 1 unit distant from each other and on bottom we append the count of categories that are 2 units distant. Despite the image only show the first

occurrence for terms of simplicity, these methods consider all the sub-windows from the start to finish of proteins chain. We also tried to use the standard Shen method in together with our idea, however the high number of features causes too much noise and the results deteriorate.

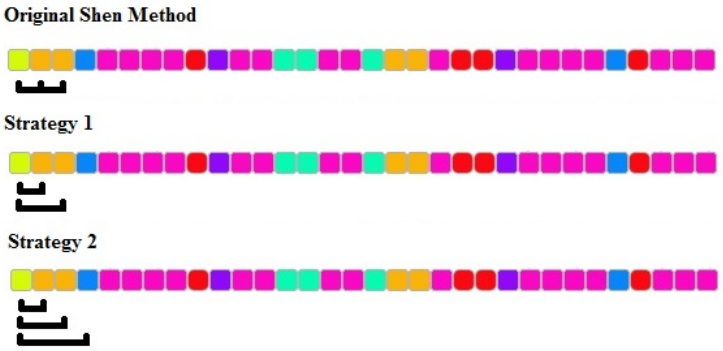


Figure 3.1: Improving Shen

A different approach was followed when we tried to improve Shen performance using conjoint methods larger than 3 units and manually limiting the number of features, however we didn't attain the desired results, since this increment adds to much noise to the features.

Afterwards, we tested complementing Shen classifier with our positional based method, since it would give the classifier more information about the position of amino acids on the protein chain. This optimization was made using both the feature extraction methods and then concatenating them in a vector representing a protein. Also an interaction was represented by the concatenation of two proteins.

3.3.4 Combining Classifiers

In order to combine different classifiers we should have simple methods that measure different features from individuals. We had a multitude of methods available, however any of them provided the desired results, so we chose the simplest of them and the ones that did not need so much computational power to build our classification network. The idea to be tested was that combining multiple different classifiers could boost the overall performance of the classification task.

Using Shen classifier, our positional approach and our physicho-chemical approach we combined their results using a majority voting system.

In terms of design this classification network is very simple. Given a pair of proteins it calculates the features for each one of these methods as explained above and calculates the correspondent output. Each one of these classifiers outputs a value in the interval [0,1] (0 for a possible not interacting pair and 1 for possible interacting pair). Afterwards, the values of output of the three classifiers are combined using different methods:

Method 1 - Choosing the average class: with this method we calculated the average values of probability of a pair belonging to the class 0 or 1 and chose the class with higher average value;

Method 2 - Choosing the most certain classifier: with this method we looked at the most discerning classifier and choose its result;

Method 3 - Using a SVM Classifier: this was the most complex method used. We got the output values from the three classifiers and used them as input for a second layer SVM-RBF classifier. Like any SVM classifier the best parameters have to be chosen and the classifier needs to be trained, so we needed to be extra careful in order to choose the correct data to train this classification network. We made cross-validation with 5 folds, and then used 80% of the test data to choose the SVM parameters and to train this classifier, and the last 20% to evaluate its performance.

The parameters used for the RBF Kernel were C= 1,000,000 and gamma = 0.0001 since they were the ones that achieved better performance in the tested window.

On the Fig. 3.2 a simple diagram helps to elucidate how the classification network works.

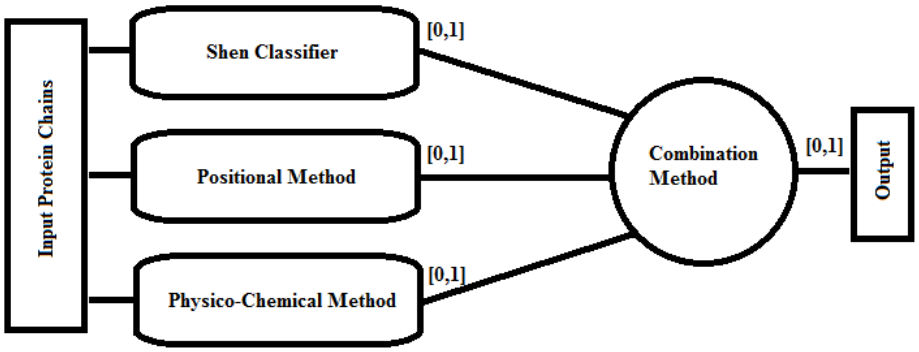


Figure 3.2: Combining Classifiers

3.3.5 Limitations Verified

During the implementation of the methods explained above we verified multiple limitations.

The N-Grams method is too simplistic. The features extracted with this method consider the ratio of the protein that is composed by a given N-Gram. It does not provide any positional information of the location of these elements and the fact of using 20 amino acids can limit the algorithm since some of the amino acids can physically be replaced by others with similar characteristics during its assembling time, phenomena named synonymous mutation.

Shen's method tries to improve these limitations by reducing the dimensions of the vector space and suiting synonymous mutation using a amino acid physico-chemical substitution table, however there is no positional neither structural information being used.

Guo's method tries to overcome these problems implementing a feature extraction technique that calculates the auto-covariance between the elements of the chain. Such metric can be seen as a structural feature, since it measures the way of how the physico-chemical properties change in a given window. However it does not present any metric of amino acid composition.

The other methods implemented are somehow simplistic and do not provide enough information in order to correctly predict PPIs, although we verified that having too much features causes the classifier to lose performance due to the noise generated by the quantity of features that are not relevant to the interaction. As a result of our first ideas to build classifiers, we noticed that the location of amino acids and the structure of the protein is important.

Given these verifications we proceeded with the idea of using a new method that could overcome such limitations.

3.3.6 Discrete Cosine Transform

3.3.6.1 Why This Approach?

At the primary level proteins are linear chains of amino acids. In this approach, each protein sequence is represented by a signal that modulates the variations of amino acids along the protein sequence.

The Discrete Cosine Transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. The DCT is well known for its practical applications in codecs such as MP3 or JPEG, allowing compression by discarding the higher frequencies.

In our opinion a method of representing proteins as a whole can be beneficial to predict PPIs that occur based the structure instead of other metrics shown in the sate of the art like amino acid counting that learn some strict parameters that can

be limited to approach all the details of the task.

3.3.6.2 Using DCT as a Feature Extraction Tool

In his previous work Shen *et al.* proposed that to reduce the dimensions of the vector space and suit synonymous mutation the 20 amino amino acids could be transformed in 7 different categories calculated accordingly to their physico-chemical properties. In the table 3.1 there is the substitution table initially used by Shen *et al.* based on the dipole scale and in the volume scale. This table was used in the present work considering that similar amino acids in the protein sequence can be susceptible to mutation.

Table 3.1: Amino acid substitution table

Category	Amino Acids
1	Ala, Gly, Val
2	Ile, Leu, Phe, Pro
3	Tyr, Met, Thr, Ser
4	His, Asn, Gln, Tpr
5	Arg, Lys
6	Asp, Glu
7	Cys

The procedure used to extract features from a protein consists of getting its amino acid sequence convert it to a vector of physico-chemical categories and then apply the DCT to the resulting vector. The signal is then reconstructed dependently of the number of features and concatenated with another signal in order to represent a protein interaction.

The DCT of a signal is given by following formula:

$$y(k) = w(k).\sum_{n=1}^{N} x(n).cos(\tfrac{\pi}{2N}(2n-1)(k-1)), k = 1, 2, ..., N,$$

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases} \tag{4}$$

And its inverse, for terms of signal reconstruction, is given by:

$$x(n) = \sum_{k=1}^{N} w(k).y(k).cos(\frac{\pi(2n-1)(k-1)}{2N}), k = 1, 2, ..., N,$$
$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases} \tag{5}$$

An arbitrary number of frequencies (F) can be used to represent a protein. If the protein is bigger than F, the first F frequencies are selected. If smaller zeros are padded until the number of desired features is archived.

After having the frequencies that describe the signal the inverse formula is used to reconstruct the original signal and to apply a standard normalization. This new signal is less noisy, since the high frequencies are ignored. It also has the same length for all the proteins and can be used to solve the classification problem. By doing this, it is possible to have representations of the proteins as a whole.

In the Fig. 3.3 we present the protein A0AQH0 sequence after replacing its amino acids with the physico-chemical categories as explained above.

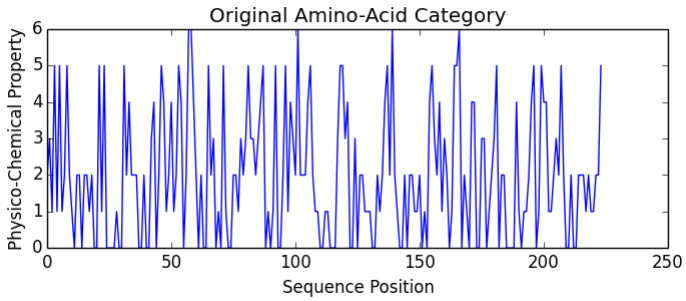


Figure 3.3: A1EKW0 Protein sequence after substitution with categories

The Fig. 3.4 presents the same protein after reconstruction using the DCT method explained above with 600 features and performing a standard normalization.

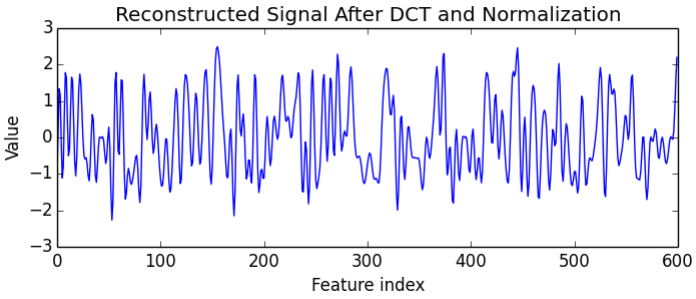


Figure 3.4: A1EKW0 Reconstructed using DCT

3.3.7 Three Dimensional Structure

Another possible approach to our work is to inference proteins three dimensional structure from the amino acid sequence. Using PyMOL, a open-source software that allows molecular visualization with the support of a script that predicts molecular structures from the amino acids chains called *build_seq.py* [44] we can build a pipeline able to create the proteins three-dimensional structures. The standard file format used to save these structures is the he Protein Data Bank (PDB).

The pipeline works as following:

FASTA file: Contains each proteins amino acid sequence.

PyMOL: Provides the libraries necessary to use the *build_seq.py* script and allows visualization of the three dimensional strucutres. It also allows to save such structures in PDB files.

build_seq.py: Script that is responsible for building a three-dimensional structure for a specified chain.

The Fig. 3.8 represents the protein A1EKW0 three-dimensional structure in PyMOL when predicted using this pipeline.

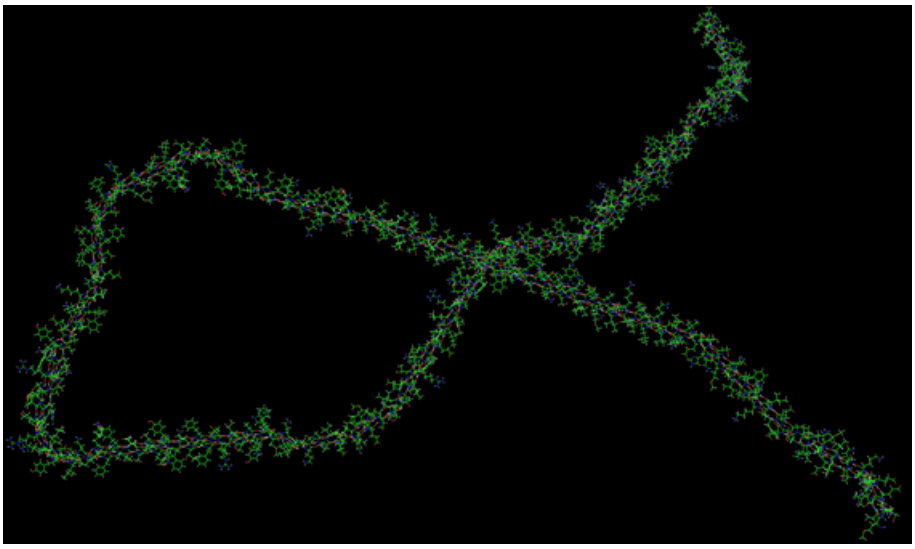


Figure 3.5: 3-D structure of A1EKW0 Protein built using *build_seq.py* and represented in PyMOL

This method is very complex from computational point of view. Given the proteins chain, we have to calculate the proteins three dimensional structure and posteriorly calculate the centers of clusters necessary for the classification algorithm. While other algorithms execute in terms of hours this takes multiple days to run in our gold dataset.

Having a prediction of the positions of the atoms for all the proteins is then

necessary to find a way to represent them in order to build a computational method. In order to do that we used the K-Means Clustering technique. We applied this clustering technique to obtain the center of the clusters and the applied standard normalization to their coordinates in the three different axis (X,Y,Z). Finally this information was used as input to machine learning methods.

The Fig. 3.6 shows the original positions of atoms extracted with Pymol and the Fig. 3.7 shows the centers of 150 clusters calculated with K-Means algorithm and ready for the classification task.

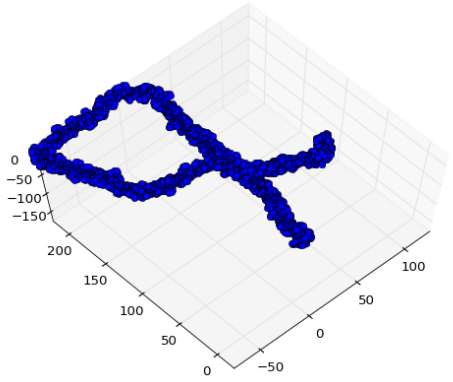


Figure 3.6: 3-D structure of A1EKW0

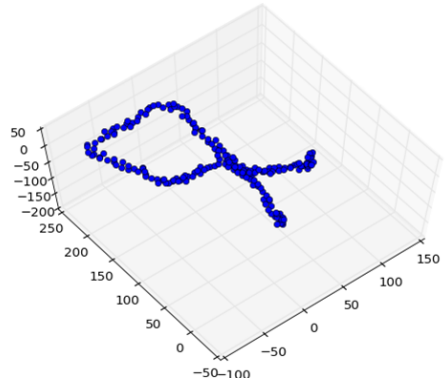


Figure 3.7: 150 Centroids of 3-D structure of A1EKW0

3.3.8 Discrete Cosine Transform - An Improvement

The usage of the Discrete Cosine Transform as in the method presented above looks at the dataset of proteins to train and classify as having all the same importance and do not work with additional biological information. In other words it uses only proteins sequence, despite the existence of other informations that could improve its behaviour.

In order to increase the performance we implemented one last optimization to this method. The Uniprot database has a Gene Ontology (GO) section that provides a set of controlled vocabulary with terms split into 3 categories: Biological Process, Cellular Component and Molecular Function. Our idea is that choosing the most predominant vocabulary terms existent in the Uniprot GO for the proteins in the dataset and posteriorly creating independent classifiers for the combinations of these GO terms can lead to more accurate results. Each classifier will be more specific, since it will only be responsible for the proteins pairs with a given GO identification.

Uniref [45] provides clustered sets of sequences from the UniProt Knowledgebase. It combines identical sequences and subfragments from any source organism into a single UniRef entry.

In order to reduce the number of protein interactions that have no GOs available in Uniprot, we used Uniref. If the protein that we want to classify as no annotation but another protein in the same Uniref cluster as GO annotation we choose the GO annotation from the other protein in order to classify our protein.

The method works as follows:

1 - Given a Protein-Protein interaction we check Uniprot for GO vocabulary terms of the proteins.

2 - If there is no GO information for the searched proteins we search other proteins within the same Uniref cluster for GO information and use that GO information.

3 - If any protein within the same cluster as GO information we use our General

Classifier.

4 - If the pair of proteins as one or more GO they are sent the respective GO-GO classifiers and their probability of interaction is calculated. Given the fact that only the top GO-GO classifiers for a given dataset are used, some GO-GO classifiers are non existent and probability of interaction is automatically defined as 0. This is not a problem since these protein pairs represent only a small portion of the dataset.

In the Fig. 3.5 a diagram shows how this classification network works. The Black color marks a GO-GO classifier as being available for training and classification. The red color is attributed to non existent GO-GO classifier. The existence or not of a GO-GO classifier is determined by the amount of proteins interactions that are attributed to that GO-GO classifier during training time. The user chooses only to train the top 5 GO-GO classifiers for a given dataset :

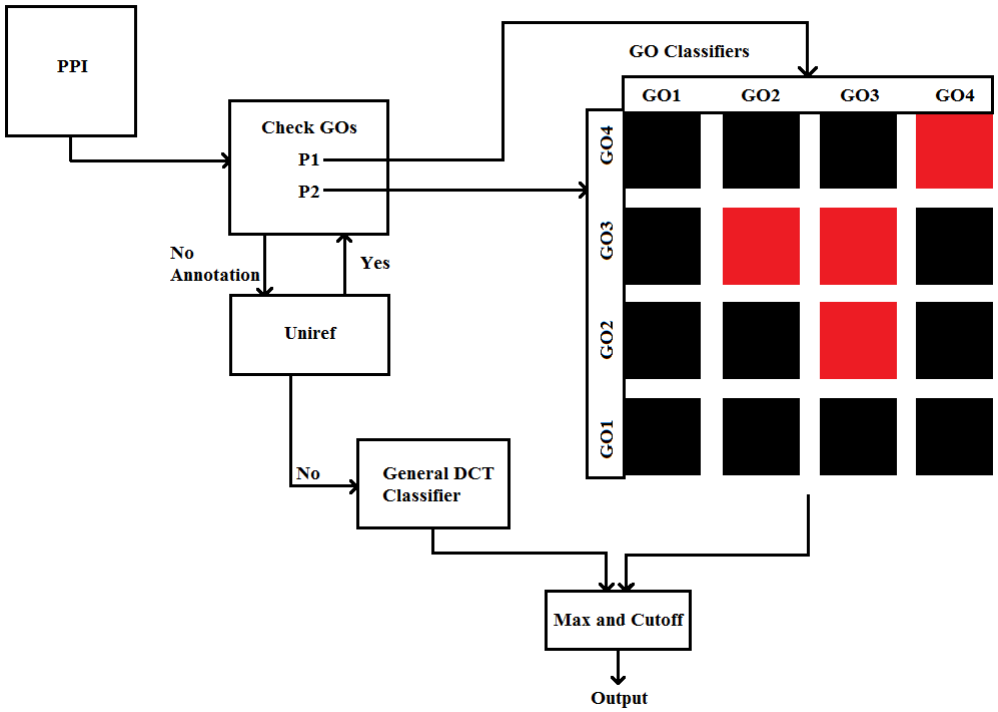


Figure 3.8: DCT Method using GO informations

3.3.9 Validating The Results

When dealing with machine learning methods, there is a condition called overfitting that occurs when the classification methods describe random noise instead of the appropriate relationships that should define the underlying relationship. It usually happens when the training data is not enough to extrapolate the correct assumptions or the number of features is to big relatively to the number of observations used for training.

In order to minimize overfitting we will use a step of validation called k-fold cross-validation. This method allows us to estimate how the results of the training are generalized when dealing with new data not used during training time.

The k-fold cross-validation method consists in dividing data into k subsets. In k steps, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average performance calculated across all k tests is computed. The order of the dataset is not much relevant since every observation gets to be in a test set exactly once, and gets to be in a training set k-1 times. When using a high value for k the variance of the result estimation is reduced. However, as a disadvantage, the training-classification process has to run k times, which means it takes k times more execution time than a simple classic training-test evaluation, making it computationally expensive to use high values of k with large datasets and large numbers of features.

Usually, researchers often use k ranges from 3 to 20, being k = 10, the most frequently value used. However, in the present work we will use large amounts of features and large datasets (as explained in the following section)and we have limited time to develop this work, so we will have to stick with k = 5 folds for the validation process.

An additional step of proficient evaluation will be conduced by experts from the Faculty of Dental Medicine from the Catholic University of Portugal. They will use

our classifier and apply it to a set of well known proteins, predict interactions and analyse the performance from the biological point of view. The final goal is to test if despite the performance of the classifier on the used datasets the predicted PPIs make sense considering the biologically known interactomes.

Chapter 4

Results and Discussion

4.1 Dataset

Due to the vast amount of interactions available on the online platforms, only small sub-sets can be used to train classifiers, due to the limitation of additional computational power.

As described in the section 3.3.1 BioGrid is our source of protein interactions and Uniprot is our source of proteins amino acid sequences for the proteins used. Negatome was used in some datasets in order to build the negative part of the dataset. We used randomly selected subsets of the known protein interactions available on the described tools and joined them in different datasets with the same amount of positive and negative interactions.

Three datasets were created in our work. However only gold dataset (dataset 1) was applied to all the methods. Datasets 2 and 3 were only applied to the Discrete Cosine Transform method since it is the main method built during our work and we wanted to test it in mode detail.

Dataset 1 (Gold dataset): This dataset consists of 6,484 interactions of a pool of 3,351 proteins extracted from the Negatome collection and an equal number of pos-

itive interactions of the same proteins from BioGrid. The chosen proteins proteins were the ones that were also available on UniProt in order to extract the amino acid sequences. The positive half of the dataset was built searching BioGrid for known interactions of the same 3,351 proteins used in the negative pool.

Dataset 2: The dataset number two was built with 10,000 protein interactions randomly selected from the BioGRID dataset. The negative interactions were a combination of the 6,484 known negative interactions from Negatome used in Dataset 1 and 3,516 random combinations of the protein pool used in the positive interactions. The dataset was also balanced having a total of 20,000 protein interactions, 10,000 positives and 10,000 negatives. This dataset allowed to test the classifier when applied to more diverse data. In fact the protein pool of this dataset was higher than the previous, consisting of 9,686 proteins.

Dataset 3: This dataset contains 20,000 protein interactions from 14,470 proteins randomly selected from the BioGRID dataset. The negative interactions were obtained by randomly construct pairs of proteins from the positive. This strategy to obtain negative interactions is acceptable, since the probability of randomly selecting a positive interaction is very low. This dataset was used to test if our method could keep achieving good results independently of the usage of the Negatome.

The Fig. 4.1 represents the datasets used on our work.

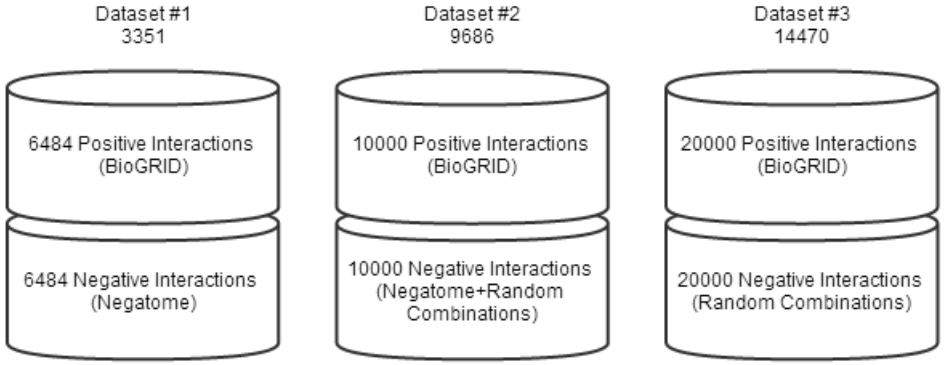


Figure 4.1: Datasets used in our work

4.2 Baseline

For the present results an SVM classifier with RBF kernel was used. Using our gold dataset the N-Grams methods achieved a good performance. With N = 3 and selecting the 1000 features with the highest term frequency, the accuracy almost hits the more complex methods described in the state of the art. We tried to increase N to 4, but the number of features becomes too big and too much noise appears in the selected features. The limitation of features based in term frequency of the top 1000 features was good enough to have a working method with window length 3, but was not was not satisfactory when applied to window length 4

The Table 4.1 presents the accuracy of the method when applied to our Gold Dataset. As we can see the performance is increased with the augmentation of the length of the windows used, however stabilizing around .80 accuracy.

Table 4.1: N-Grams Accuracy when applied to Gold Dataset

N-Gram Method	Accuracy	Std. Deviation
N = 1	0.725	0.008
N = 2	0.780	0.001
N = 3	0.800	0.001
N = 4	0.770	0.001

The truth is that we didn't expect results so accurate from a method so simple. However this algorithm is known for its results on a multitude of fields from speech recognition and computational linguistics to DNA sequencing. The performance of this algorithm increases with the increment of window length N, however it is difficult to optimize it after achieving its limitations. In our case it was impossible to experiment with bigger windows or with a number of features higher than 1000 since the execution time would increase to multiple days. On other hand the accuracy results seem to achieve its maximum around 0.80, so it would be too much

computation time for probably a really small increment of performance.

4.3 State of the Art Methods

The state of the art methods are more complex than the N-Grams, however more complexity was not equal to better performance. With results slightly higher than simpler methods, Shen and Guo methods showed a performance line to be transcended in this work.

The table 4.2 presents the state of art results when applied to our Gold Dataset.

Table 4.2: State of the art accuracy when applied to Gold Dataset

Method	Accuracy	Std. Deviation
Shen et al (2007)	0.803	0.011
Guo et al (2008)	0.804	0.001

We expected results slightly higher from the methods enunciated in the state of the art, as it happens in the referenced papers, but that didn't happen.

Despite presenting higher performances in their presentation papers, both methods fall to around 0.80 accuracy when applied to our gold dataset. Such fall in performance can be caused by a multitude of factors: Our dataset is composed by interactions of proteins from multiple organisms, increasing variability in data; Our dataset uses proteins from any part of cells, which can difficult the classification process and we did not implement any filter to the dataset that could boost the overall accuracy of methods.

4.4 Our First Methods

When implementing state of the art methods we noticed that both physico-chemical properties and the positioning of the amino acids had big importance on the classification task, so we developed feature extraction methods that consider these measures as explained in the subsection 3.3.3.1.

The table 4.3 and the Fig 4.2 present the accuracy and standard deviation of our simple physico-chemical based method on our Gold Dataset.

Table 4.3: Our Physico-Chemical approach applied to Gold Dataset

Number of Divisions	Accuracy	Std. Deviation
2	0.635	0.015
3	0.661	0.017
4	0.708	0.018
5	0.718	0.006
6	0.717	0.001
7	0.722	0.007
8	0.727	0.009
9	0.729	0.014
10	0.735	0.009
11	0.730	0.018
12	0.729	0.013
13	0.737	0.031
14	0.686	0.009
15	0.634	0.017
16	0.611	0.018
17	0.598	0.007

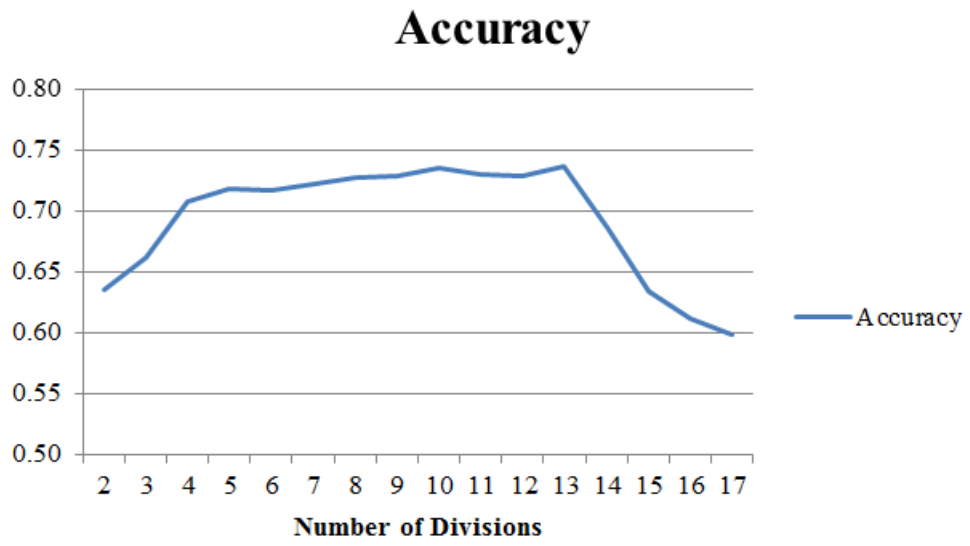


Figure 4.2: Dividing protein chains in parts

As we can see, the results were not great. The accuracy is dependent on the number of divisions to a given point, but then it drops and the method becomes unstable and a lot of wrong predictions are made.

It was just an idea that need to be explored, however this method cannot compete with other previously implemented methods since its performance is too low, even with the most suitable number of divisions it never passes 0.737 accuracy.

In parallel with the physico-chemical based method we implemented another simple method consisting on the location of the N-Grams on the protein chain, for terms of simplicity we call it the positional method, for more details consult subsection 3.3.3.1

On the table 4.4 the results of our positional method on our Gold Dataset are shown.

Table 4.4: Positional method accuracy on our Gold Dataset

N-Gram Positions	Accuracy	Std. Deviation
N = 1	0.758	0.020
N = 2	0.788	0.016
N = 3	0.599	0.012

Despite the simplicity of the method and the reduced number of features the results were better than the ones achieved with the physico-chemical method indicating that the position of amino acids in the protein chain is a good indicator of interaction between proteins. In fact using only the positional method we achieved a reasonable performance of 0.788 accuracy, that is proximate to the existing state of the art methods. We tried to increase the N-Gram window to 3, but the number of features becomes too big and the performance starts to decrease, this happens mainly due to the increase of noisy features that disturb the learning process.

4.5 Optimizing Shen

After having the above implementations working we tried to upgrade Shen's method. We chose this method mainly due to its simplicity and good results obtained and we tried a multitude of perspectives in order to upgrade it.

In the table 4.5 the results obtained in diverse experiments on our Gold Dataset are presented:

Table 4.5: Trying to Optimize Shen - Accuracy on Gold Dataset

Method	Accuracy	Std. Deviation
Shen window 2	0.787	0.007
Shen window 3	0.803	0.011
Shen window 2 + positional features N = 2	0.803	0.180
Shen window 3 + positional features N = 2	0.783	0.015
Shen window 4	0.763	0.020
Shen window 5	0.767	0.013

The standard Shen method is represented by the "Shen window 3", since it uses groups of three units. We reduced the length of the conjoints to 2 in order to reduce the number of features used and have a valid comparison with other similar methods.

When using Shen window 2 in addition with positional features the number of features is 98 (7*7+7*7) and when using standard Shen method the number of features is of 343 (7*7*7), so we can say that we achieved results similar to the ones published in the state of the art, using a number of features 3.5 times smaller, which boosts the execution time of the feature extraction technique and of the classification task. However this was not our objective, just an intermediate fruit of our work, so we did not stop here, since we wanted to outscore the existing methods.

The other results correspond to experiences on which we did not achieve the desired performances. We thought that using Shen window 3 in combination with our positional method we could achieve better performance, however the results dropped, in our opinion, due to the large amount of features being used that provokes noise in the learning process. For the Shen window 4 and Shen window 5, we limited the number of features to the 1000 most frequent. We expected that protein sub-structures would be recognized by this method and that would increase the classification per-

formance, however that didn't happen and the results decayed. Our limitation of features to the 1000 most frequent may have caused this decay, we could probably explore this issue with more detail but we had plenty of other ideas that could work better and decided to follow them.

4.6 Combining Classifiers

We tried to combine Shen classifier, our positional method and the physicochemical approach using different methods. We chose this classifiers due to its simplicity and for that fact that measure different characteristics of proteins. So we thought that combining such characteristics a classification network we could boost the overall performance of the classification task.

In the table 4.6 the results obtained on our Gold Dataset are shown, for more detailed description of the methods, please read subsection 3.3.4:

Table 4.6: Combining classifiers

Method	Accuracy	Std. Deviation
Method 1 - Average Output Class	0.810	0.017
Method 2 - Best certain classifier	0.809	0.012
Method 3 - SVM Classifier	0.810	0.009

Any of the three combination strategies outscores all the previous implemented methods. The results obtained are similar in all of them and consequently we cannot say that one is better than another. The usage of simple metrics on method 1 and 2 can compete with method 3 in terms of performance, so we advice choosing them for a future work. Method 3 uses an additional SVM classifier, and for that reason it has computationally costs that need to be considered either for the training

and for the classification tasks.

As expected, combining simple classifiers that consider multiple feature extraction techniques allows the development a classification algorithm that outperforms the results of more complex methods that only look at a simple feature. This happens, because the interaction of proteins is not subject of a simple property or conditioned by a simple rule. On opposite, PPIs are the result of a multitude of physico-chemical, structural, positional, and other factors. Increasing the amount of features used, and looking at proteins from different perspectives increases the performance of computational methods. At this point we had a valid method that could compete with the state of the art, but we kept testing other ideas.

4.7 Discrete Cosine Transform

In this section we present the results achieved with our DCT approach to the problem, the main computational method developed by us. We will use more datasets and more metrics in order to make a better analysis of this method.

4.7.1 Choosing the Optimal Number of Features

The number of frequencies used in the DCT can be manually selected. However there is the need to test different values to have some indication of how many should be used in order to attain the optimal results. One of our first tests was made using the KNN classifier in order to choose the number of features to be used.

Fig. 4.3 presents the evaluation of the classifier accuracy when compared with the increase in the number of features when applied to our Gold Dataset. The line in green contains the raw frequencies extracted from the protein features. Clearly this result is not as good as the other methods. In red the accuracy score using the recon-

struction of the protein amino acid chain while considering the original 20 amino acids is shown. Finally in blue there is the representation of a replacement of the 20 amino acids with the 7 categories where feature extraction and signal reconstruction were performed. The peak occurs in the blue line while using 600 frequencies achieving an accuracy of 0.825, so for the rest of the work the strategy of using a reconstructed signal after substitution of the amino acids with the 7 amino acid categories.

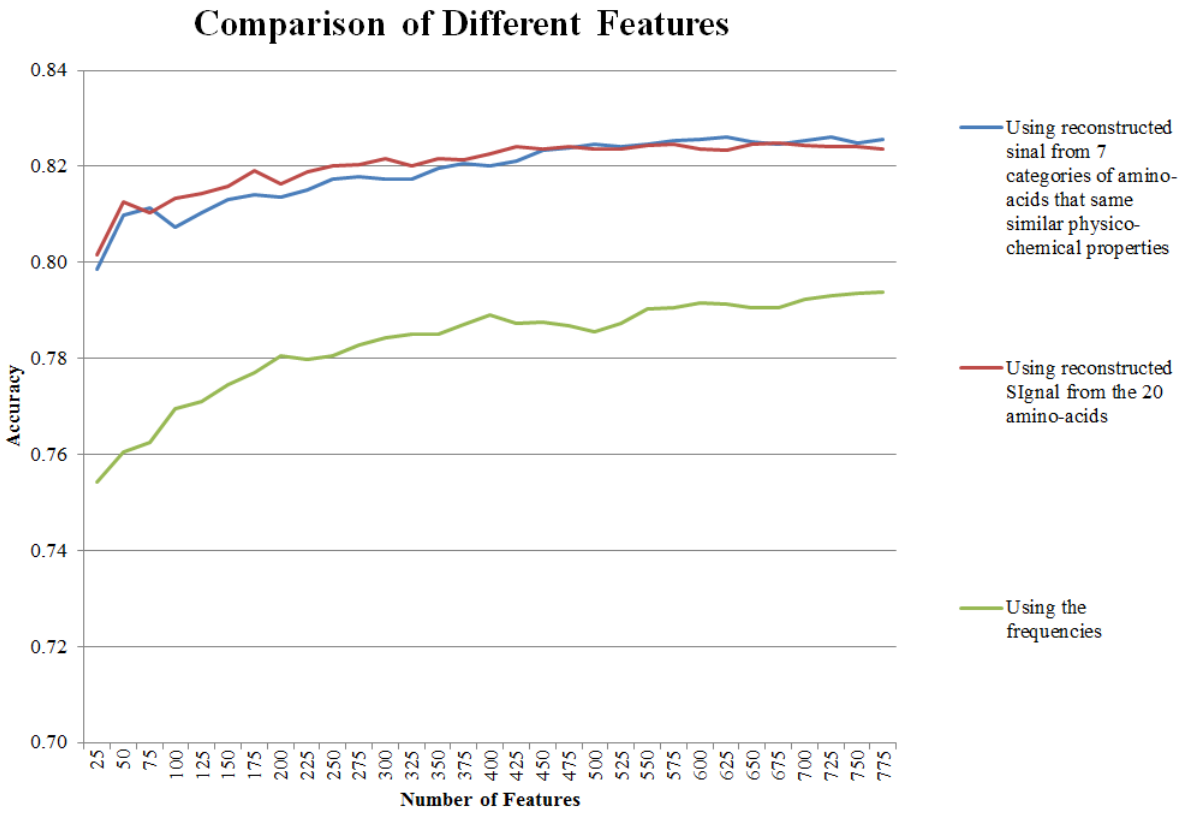


Figure 4.3: Choosing the number of features

4.7.2 KNN Classifier

In the present work two classification methods were tested, the K-Nearest Neighbour (KNN) classifier and the Support Vector Machine (SVM). On this section we are going to study the best parameters for the KNN Classifier.

The KNN classifier uses as parameters the number of neighbours to consider in order classifying a sample as being of one class or of another. In order to optimize our classifier an experimentally evaluation of this parameter was made in the range from 3 to 19.

As present in the Fig. 4.4 the optimal number of neighbours for classification was 9 neighbours, achieving the accuracy of 0.825, the test was made using the dataset number one with cross-validation with 5 folds.

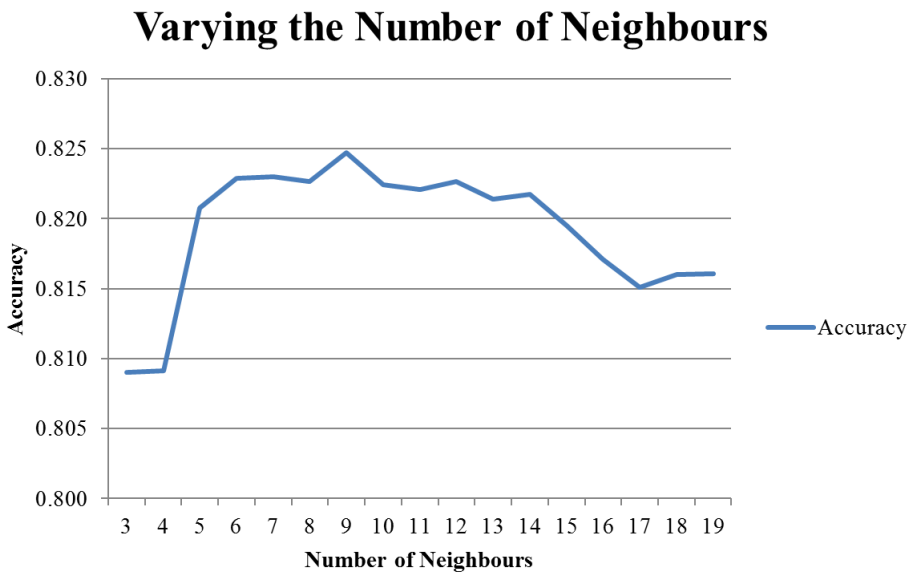


Figure 4.4: Choosing the number of neighbours

4.7.3 SVM Classifier

The SVM classifier with RBF kernel, as used in the present work uses two parameters, C and gamma. In order to test our method, these parameters were changed in the window shown in Fig 4.5, using the dataset number one and cross-validation with 5 folds applied to our Gold Dataset. As shown in the figure the optimal parameters were C = 100 and gamma = 0.001, achieving an accuracy of 0.827. Such parameters were afterwards used for testing with the other datasets.

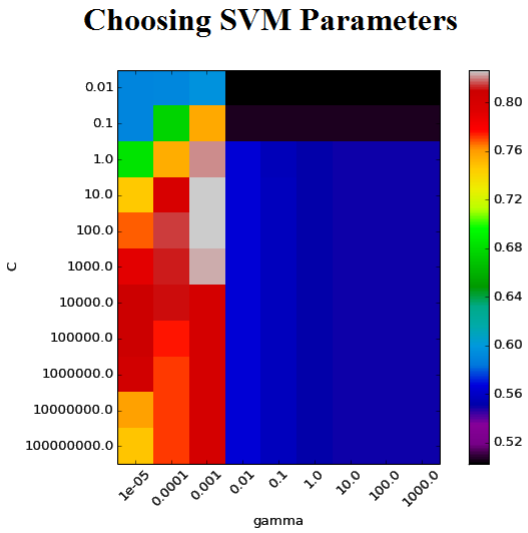


Figure 4.5: Choosing the best SVM parameters

4.7.4 Comparing Accuracy With the State of the Art

In the Table 4.7 and Fig. 4.6 the results achieved with the different methods in the different datasets are presented. Fig. 4.6 presents the results of our DCT method using the SVM classifier. It outperforms the existing state of the art methods while tested on every dataset. Although slower, the increase on the accuracy might be beneficial for the extra computing time. The KNN method for classification of our features behaves acceptably. On some datasets it even outperformed the results achieved with the existing state of the art.

Table 4.7: Comparing DCT with other methods

	Guo	Shen	Our method using KNN	Our method using SVM
Dataset 1 Accuracy	0.804 +/- 0.010	0.803 +/- 0.011	0.825 +/- 0.004	0.827 +/- 0.004
Dataset 1 Precision	0.769 +/- 0.017	0.797 +/- 0.014	0.814 +/- 0.011	0.817 +/- 0.012
Dataset 1 Recall	0.861 +/- 0.007	0.812 +/- 0.028	0.845 +/- 0.008	0.843 +/- 0.008
Dataset 2 Accuracy	0.720 +/- 0.012	0.746 +/- 0.014	0.765 +/- 0.004	0.761 +/- 0.004
Dataset 2 Precision	0.669 +/- 0.010	0.727 +/- 0.019	0.781 +/- 0.008	0.736 +/- 0.005
Dataset 2 Recall	0.867 +/- 0.026	0.782 +/- 0.014	0.665 +/- 0.010	0.789 +/- 0.009
Dataset 3 Accuracy	0.646 +/- 0.014	0.705 +/- 0.004	0.664 +/- 0.005	0.707 +/- 0.005
Dataset 3 Precision	0.639 +/- 0.012	0.741 +/- 0.009	0.761 +/- 0.009	0.678 +/- 0.007
Dataset 3 Recall	0.642 +/- 0.010	0.630 +/- 0.012	0.639 +/- 0.003	0.723 +/- 0.004

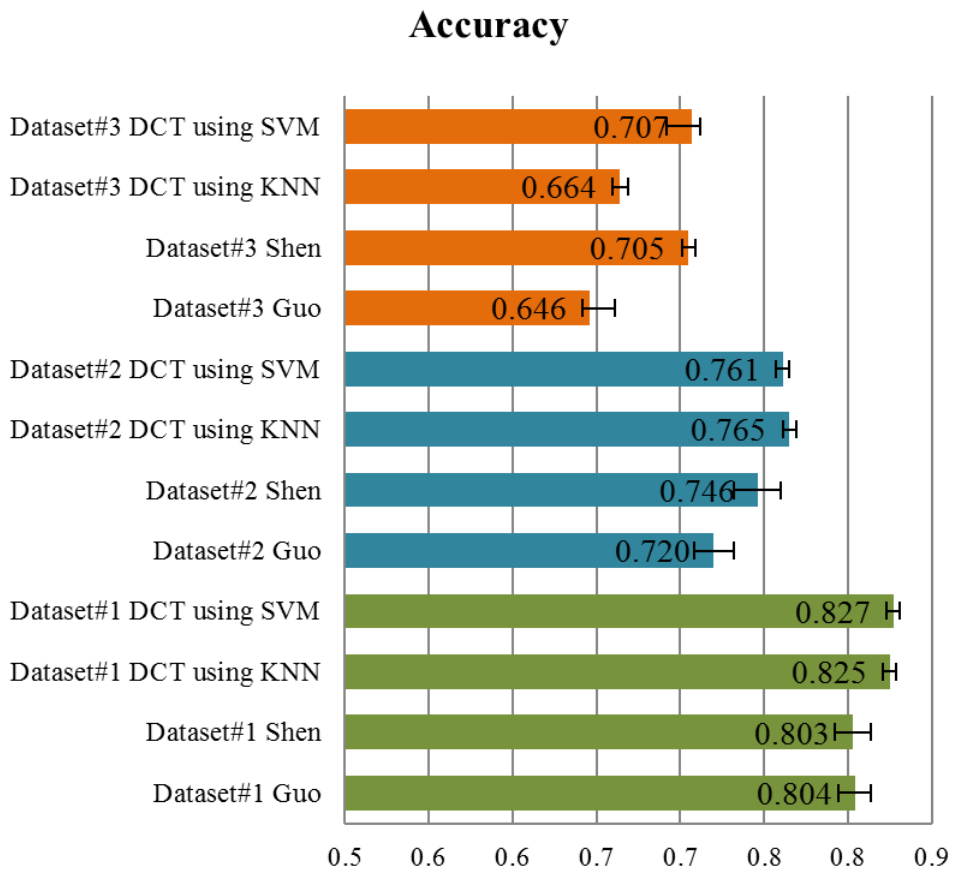


Figure 4.6: Comparing DCT with other methods

4.7.5 Execution time comparison

The differences in the execution time were noticed since the start of testing the existing methods. When testing in small datasets such differences are low and

unnoticeable, however when working with bigger datasets similar to the ones used by biologist, this factor needs to be considered.

The following tests were made with the dataset number one. The execution times are showed in seconds and correspond to executions of feature extraction and cross-validation (training and test) of the dataset using 5 folds. The tests were made using a common i7-4700mq CPU and 8 GB of RAM.

In table 4.8 the execution time results for our Gold Dataset are present:

Table 4.8: Comparing execution time in seconds

Method	Execution Time (Seconds)			Average	Std. Deviation
	Run 1	Run 2	Run 2		
Guo	5913	6212	5984	6036	128
Shen	7259	6496	6748	6834	317
Our method using k-NN	595	568	617	593	20
Our method using SVM	11739	12791	11865	12132	469

Using KNN classifier with our feature extraction method it was the fastest method tested. The classification method used by Shen et al. and Guo et al. is based on SVM, when using an SVM classifier our method was the slowest. Despite being slower, the increase on the accuracy, while using DCT with SVM, might be beneficial for the time spent on computing.

4.7.6 Using a Validation Dataset

A further step of validation was made in this study in order to infer our classifiers capabilities under real conditions.

The dataset 2 was used for training the classifiers on this sub-section since it has a good balance between two factors. It includes the Negatome, providing a good in-

dication of the negative interactions, but not limiting the information to the proteins contained on such dataset proteins pool.

In order to test the classifier performance on real data, 1000 random interactions were selected from the BioGRID to build the independent validation dataset. Performing training and validation with these datasets we obtained the results present in Fig. 4.7:

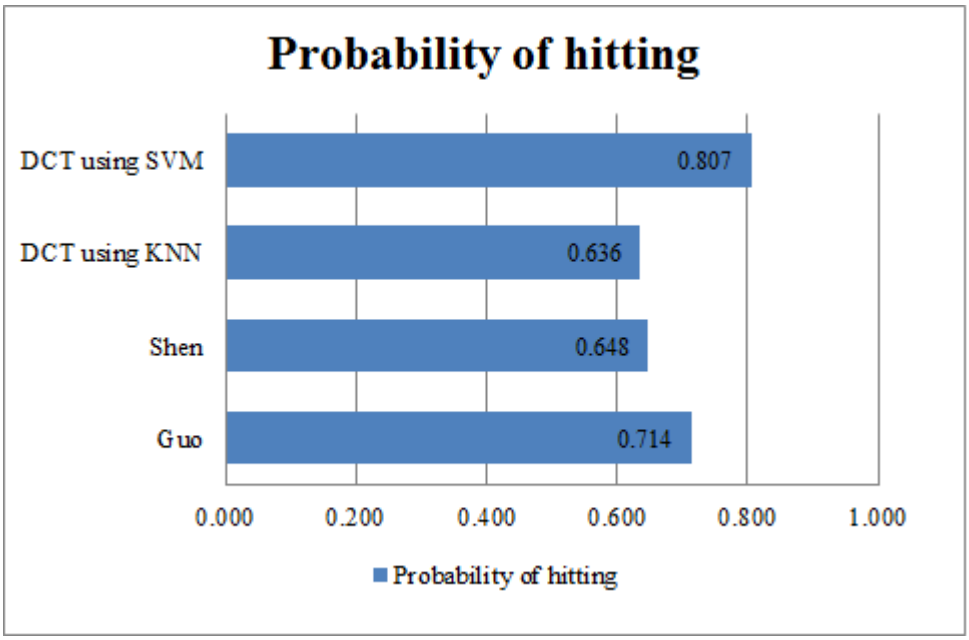


Figure 4.7: DCT performance on extra validation dataset

Our method was the one that achieved the best performance accurately predicting 80.7% of the known positive interactions and with a great margin of performance above the other tested classifiers.

4.8 Three Dimensional Structure

When extracting data from the three dimensional structure of proteins the results were very interesting. The Table 4.9 presents these results of this method when applied to our Gold Dataset:

Table 4.9: Three Dimensional Method Accuracy

Method	Accuracy per Number of Clusters			
Clusters	50	100	150	200
LDA	0.657	0.710	0.727	0.759
QDA	0.808	0.838	0.842	0.841
SVM - Kernel RBF	0.814	0.819	0.817	0.820

In terms of number of clusters considered for the machine learning algorithm the optimal number was of 150. When using a Linear Discriminant classifier the results were worse than other methods previously tested. Using a SVM with RBF kernel we outscore the state of the art methods, but we did not outscore our DCT method. Finally when using a Quadratic Discriminant classifier the results are the highest achieving 0.842 accuracy and and outscoring any other method. However this method is very complex from computational point of view. The calculation of proteins three dimensional structure and posteriorly calculate the centers of clusters makes the algorithm inapplicable to bigger datasets. Even for this smaller dataset took multiple days.

4.9 Discrete Cosine Transform - An Improvement

When boosting our DCT method with information from GOs, as explained in the section 3.3.7, we cannot use the same datasets, because it needs much more volume of data to be able to train the different classifiers. So, in order to test this method we used for the positive part of the dataset of the publicly available protein interactions from *Saccharomyces cerevisiae* in addition with randomly selected pairs of proteins of the same organism for the negative part. By using this dataset we can evaluate the performance of the method and also compare it with the state of the art method

proposed by Guo. He achieved 0.8736% accuracy on this dataset.

In the table 4.10 we present the results achieved with our classification network, 237,572 protein interactions were considered but only 125,256 were used to train the different classifiers, since we limited the max number of interactions to 15000 for each classifier.

Table 4.10: GO Classifier Results

Classifier	# Interactions	# Interactions Used	Accuracy	Std. Dev.
go3a5515 - go3a5515	88775	15000	0.969	0.002
go3a5488 - go3a5515	48269	15000	0.931	0.008
General Classifier	47841	15000	0.932	0.002
go3a3824 - go3a5515	16921	15000	0.871	0.008
go3a5488 -go3a5488	10638	15000	0.810	0.003
go3a3824 - go3a5488	6337	12674	0.777	0.002
go3a5198 - go3a5515	5897	11794	0.903	0.006
go3a5215 - go3a5515	5209	10418	0.826	0.007
go3a30234 - go3a5515	4175	8350	0.934	0.006
go3a5198 - go3a5488	3510	7020	0.851	0.007
Total	237572	125256	-	-
Weighted Average	-	-	0.927	0.004

On this dataset the GOs had the following meaning:

Table 4.11: Meaning of GO IDs

GO ID	Meaning
go3a5515	Protein Binding
go3a5488	Binding
go3a3824	Catalytic Activity
go3a5198	Structural Molecule Activity
go3a5215	Transporter Activity
go3a30234	Enzyme Regulator Activity

As the results show, having multiple classifiers that subdivide the dataset is multiple parts according to biological or molecular terms associated with the pairs of proteins is beneficial in terms of performance.

Considering that different amounts of data are assigned to each classifier and some classifiers have better performance than others, we were able to predict protein interactions with an weighted average accuracy of 0.927 outscoring the Guo's state of the art method for similar dataset. This method is well suited for large datasets since it creates different classifiers for the most frequent GOs.

4.10 Result Summary

As ways of comparison on the following figure we present the results achieved during the development of this work. In black the baseline and state of the art methods. In grey some of the experiments made and described in the document that didn't perform as good as expected. In green our combination methods. In blue our extraction of features using the DCT and finally in red our three dimensional method.

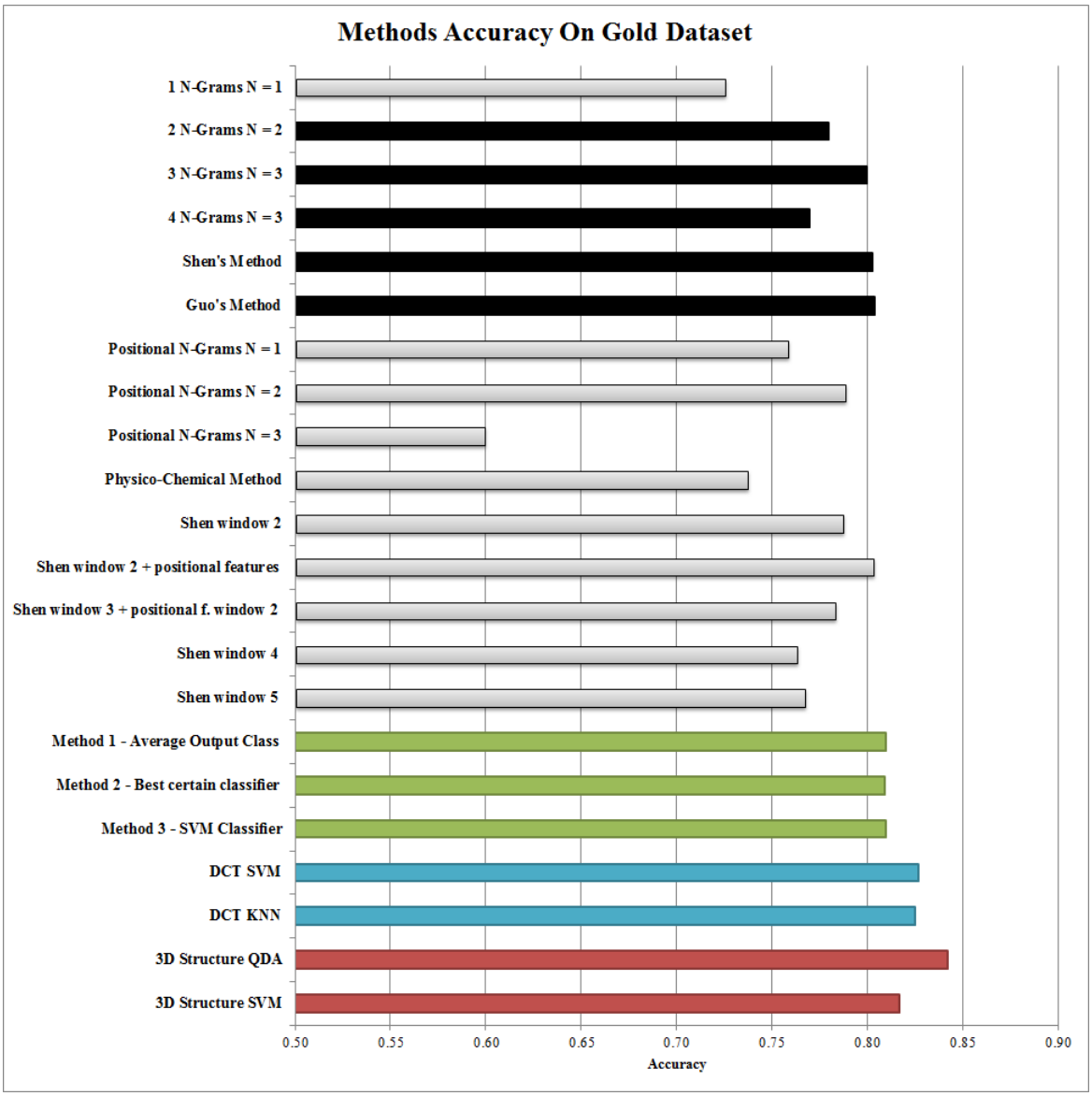


Figure 4.8: Comparing Accuracy

Our best method, DCT using GO annotations requires large amounts of data to function. We had two options, either run one of these large datasets for all the methods or run a smaller dataset for all the methods and a larger one for this method. Due to the limited computational power and time for the task we chose the second option. This is the reason why this method is not represented in the above plot.

Chapter 5

Final Considerations

During the whole project, multiple decisions and corrections were made, we didn't know if we were to succeed in the task of developing a new functional method. Time constraints or computational constraints, were factors that were always compelling us to make the most diverse decisions.

With the development of this work, either during the feature extraction time, construction of our classifiers or as finished tool, we performed tests which gave us important information that allowed us to make new considerations regarding the work that was developed during this dissertation. Therefore, in this chapter we present our main conclusions during the development of the project. Furthermore, we also detail the limitations of the proposed approaches, and propose additional challenges or issues that can be addressed in future works.

5.1 Tool Availability

The final tool for predicting PPIs is available online for the community to use and optimize. It can be found at https://code.google.com/p/pprint-protein-protein-interaction-prediction/ under the Apache License 2.0, it can be useful for future researches in the Bioinformatics field.

5.2 Conclusions

The objectives proposed in an earlier stage were achieved. During the development of this work we were able to produce a computational classifier that predicts PPIs with better accuracy than the previously existing methods described in the state of the art. In fact diverse approaches idealized and implemented outscored the existing methods when applied to our interspecies datasets.

We were able to implement and detect limitations on the existing state of art methods. As expected since the start of the work, the existing methods work better in the intraspecies datasets than in our interspecies datasets. Their performance was acceptable, but the results achieved by such methods were lower than the ones published in the existing papers. So the prediction of proteins between multi-organisms is really a task that needs to be considered by future investigations, since existing method's scores reduced when applied to this kind of datasets.

During the development of our project, it was relatively easy to build new methods based on new feature ideas, however these methods were difficult to optimize in order to outperform the state of the art. It was hard to build methods able to achieve an accuracy higher than 0.80 on our Gold Dataset and some experiences ended up being loose ends.

Earlier in the project we started by implementing existing baseline and state of the art methods and verifying their performance and limitations. Either N-Grams, Shen and Guo methods scored .80 on our Gold Dataset. So from that moment on we had that value as an goal to outscore.

On an intermediate phase we spent some time trying to improve the existing classifier proposed by Shen mainly due to its good performance and high simplicity but we could not achieve better results. However we could achieve the existing performance reducing the window length to 2 and complementing it with our positional method. This strategy has a quantity of features considerably smaller. Such reduc-

tion on the number of features can be beneficial for questions of time performance or less computational needs allowing a larger throughput by the classification algorithm. However we had other ideas to build classifiers, and preferred to use the existing time implementing methods that could outperform the existing results rather then make further testing about execution time comparison of this method.

When we combined 3 simple classifiers using different strategies we outscored the state of the art methods for the first time, achieving 0.810 accuracy. As we thought, combining simple classifiers that consider multiple feature extraction techniques allows the development a classification algorithm that outperforms the results of more complex methods. This happens, because the interaction of proteins is not subject of a simple property or conditioned by a simple rule. On opposite, PPIs are the result of a multitude of factors that may lead to interaction or not. In our conclusion, increasing the amount of features used, and looking at proteins from different perspectives, can increase the performance of computability methods. However we did not stop here and we were more ambitious, 0.810 accuracy was just a small step from the 0.804 that needed to be passed.

The N-Grams method is too simplistic. It does not provide any positional information of the location of each element and the fact of using 20 amino acids can limit the algorithm since some of the amino acids can physically be replaced by others by a process called synonymous mutation.

Shen's method tries to improve N-Grams limitations however there is no positional neither structural information being used in the method.

On other hand Guo's method implements a feature extraction technique that calculates the auto-covariance between the elements of the chain. Metric that can be seen as a structural feature, since it measures how the physico-chemical properties change in a given window. However it does not present any metric of amino acid composition.

The representation of the protein sequence as a whole, an idea proposed in this work, allows the classifiers to predict interactions while making a generalization of the shape of the sequence, instead of dividing it in smaller problems as is explored by other machine learning techniques.

We built a method that uses the Discrete Cosine Transform that performs better than existing alternatives. In addition to the performance improvement the SVM classifier allied to our method has proven to be more resilient to changes.

Based on the results achieved, our DCT method can compete with the state of the art methods or even provide better results in terms of performance. Considering the accuracy results, our method outscored the existing methods while using the SVM classifier on every dataset tested. Independently of number of interactions, number of proteins, and method of choosing negative samples for the dataset, the results achieved were always better, achieving the peak of 0.827 accuracy on our Gold Dataset.

In terms of computation time required, we verified that our DCT method becomes slower than the other methods, but due to the expensiveness of experimentally*in vitro* testing of protein pair interaction it might be useful to have a slower but more accurate computational method to serve as a filter of what proteins to experimentally test.

The KNN classifier allied to our DCT method showed good results while using the Negatome as a source for non-interacting protein pairs. However when tested with randomly selected protein pairs from the protein pool as negative examples the results were poor. Despite this, it is still a viable classification method to do some experiments since it execution time is ten times faster than the next classifier.

A classification network was built using our DCT method and Gene Ontology information publicly available in Uniprot. Each classifier was responsible for a more specific set of data attributed based on each pair of proteins molecular function, bi-

ological process or cellular component. By combining different classifiers responsible for more specific data we were able to achieve an accuracy of 0.927 when applied to *Saccharomyces cerevisiae* protein interactions, outscoring this way Guo's state of the art.

Finally our three dimensional methods presented highest results when combined with a QDA classifier 0.842 accuracy. However the execution time of this method (multiple days, just for the smallest dataset) limited further testing. For being so slow it cannot be applied to larger dataset on acceptable time.

For this reason we propose our DCT feature extraction method allied with an SVM classifier to classify datasets of proteins from multiple organisms.

5.3 Future Work

Along side with the development of this document and all the source code we also wrote a paper that was submitted to IEEE International Conference Bioinformatics and Biomedicine (BIBM).

In terms of improving the actual work there are some optimizations that can be studied. We suggest a deeper and more profound study of the machine learning algorithms to operate with our feature extraction method. For example the study of other classifiers or the development of a custom kernel could be beneficial for improvement of the method, since a pair of proteins can be represented by protein A concatenated with protein B or protein B concatenated with protein A and still the same interactions, however our method is blind about that matter.

On other hand combining classifiers revealed itself a good solution for our problem when applied to low performance methods, but what would be the results now that we have higher accuracy methods? Will the results get even better? That could be a good idea to explore.

Some of the classifiers studied in the state of the art and implemented often recur to manually selection of data in order to explore more locational problems. It would be interesting to implement multiple classifiers in grid with our feature extraction techniques using data with less variation then combine them together. We think this would increase the results because classifiers could learn different strategies for different data.

References

[1] M. S. Waterman, *Introduction to Computational Biology: Sequences, Maps and Genomes*. Spring Harbor Press, 1995.

[2] D. W. Mount, *Bioinformatics: Sequence and Genome Analysis*. 2002.

[3] C. Claverie, J.M.; Notredame, *Bioinformatics for Dummies*. Willey, 2003.

[4] T. Pawson and P. Nash, "Protein-protein interactions define specificity in signal transduction," *Genes Dev*, 2000.

[5] O. J. Coelho ED, Arrais JP, "From protein-protein interactions to rational drug design: are computational methods up to the challenge?," *Current topics in medicinal chemistry, Bentham Science Publishers*, 2013.

[6] C. Royer, "Protein-protein interactions," 2013.

[7] K.-T. E. Guan H, "Advanced technologies for studies on protein interactomes," *PubMed*, 2008.

[8] F. L. J. L. Chishe Wang, Jie Song, "Identifying protein-protein interaction sites using adapted bayesian classifier," *BMC Bioinformatics*, 2009.

[9] "Uniprot - protein knowledgebase." http://www.uniprot.org. [Online; accessed Dezember-2013].

[10] R. P. W. D. Loo JA, Yan W, "Comparative human salivary and plasma proteomes," *Journal of Dental Research 89*, 2010.

[11] M. M. Rabie Saidi and E. M. Nguifo, "Research article protein sequences classification by means of feature extraction with substitution matrices," *BMC Bioinformatics*, 2010.

[12] Z. W. Yanzhi Guo, Lezheng Yu and M. Li, "Using support vector machine combined with auto covariance to predict proteinprotein interactions from protein sequences," *Nucleic Acids Research*, 2008.

[13] M. C. P. Ulloa-Aguirre A, "Pharmacoperones: a new therapeutic approach for diseases caused by misfolded g protein-coupled receptors," *PubMed*, 2011.

[14] P. H. G. Giuseppe Agapito and M. Cannataro, "Visualization of protein interaction networks: problems and solutions," *BMC Bioinformatics*, 2013.

[15] L. E. et al., "International human genome sequencing consortium, initial sequencing and analysis of the human genome," *Nature 409*, 2001.

[16] J. R. Bock and D. A. Gough, "Predicting proteinprotein interactions from primary structure," *Bioinformatics*, 2001.

[17] A. N. Mario R. Guarracino, "Predicting protein-protein interactions with k-nearest neighbors classification algorithm," *Lecture Notes on Computer Science*, 2010.

[18] B.-T. Z. Jae-Hong Eom, "Prediction of protein interaction with neural network-based feature association rule mining," *Lecture Notes in Computer Science Volume 4234*, 2006.

[19] J.-H. Eom and B.-T. Zhang, *Prediction of Protein Interaction with Neural Network-Based Feature Association Rule Mining.*

[20] P. D. Daniel Lowd, "Naive bayes models for probability estimation,"

[21] H. Zhang, "The optimality of naive bayes,"

[22] S. M. McDowall, MD and Barton, "Gj pips: Human protein-protein interactions prediction database," *Nucleic Acids Research*.

[23] M. Scott and Barton, "Gj probabilistic prediction and ranking of human protein-protein interactions," *BMC Bioinformatics*.

[24] C. C. V. V., "Support-vector networks," *AT T Labs*, 1995.

[25] T. T. Shinsuke D., Asako K, "Support vector machines for predicting protein-protein interactions," *Genome Informatics*, 2003.

[26] Y.-S. H. H. P. Yoojin Chung, Gyeong-Min Kim, "Predicting protein-protein interactions from one feature using svm,"

[27] "Dip," 2013.

[28] D. J. P. A. P.-P. e. a. Rodriguez-Soca Y, Munteanu CR, "Trypano-ppi: A web server for prediction of unique targets in trypanosome proteome by using electrostatic parameters of proteinprotein interactions," *Journal of Proteome*, 2009.

[29] D. J. R. J. P. A. Rodriguez-Soca Y, Munteanu CR, "Plasmod-ppi: A web-server predicting complex biopolymer targets in plasmodium with entropy measures of proteinprotein interactions," 2010.

[30] R. B. R. Patrick Aloy, "Interprets: protein interaction prediction through tertiary structure," *Bioinformatics*, 2002.

[31] V. S. Mathura and D. Kolippakkam, "Apdbase: Amino acid- physicochemical properties database," *Bioinformation*, 2005.

[32] X. Q. L. Y. Xiao N, Cao D, "protr: Protein sequence feature extraction with r," 2012.

[33] "Fasta software package," 2014.

[34] P. RR., "Structural studies of immunoglobulins," *Science 180 (4087): 7136*, 1973.

[35] de Souza SJ, "Domain shuffling and the increasing complexity of biological networks," *Pub Med*, 2012.

[36] M. E. B. L. Terrapon N, Gascuel O, "Fitting hidden markov models of protein domains to a target species: application to plasmodium falciparum," *BMC Bioinformatics*, 2012.

[37] J. S. Richardson, "The anatomy and taxonomy of protein structure," *Adv Protein*, 1981.

[38] W. S. N. Shawn M. Gomez and A. Rzhetsky, "Learning to predict proteinprotein interactions from protein sequences," *Bioinformatics*, 2003.

[39] E. SR., "Prole hidden markov models," *Bioinformatics 14*, 1988.

[40] J. D. Taylor WR, "Deriving an amino acid distance matrix," *Pub Med*, 1993.

[41] L. X. Z. W. Y. K. C. K. L. Y. J. H. Shen J, Zhang J, "Predicting protein protein interactions based only on sequences information," *Pub Med*, 2007.

[42] P. P. W. P. B. B. D. I. F. G. F. G. M. C. R. T. F. D. R. A. Smialowski, P., "The negatome database: a reference set of non-interacting protein pairs,"

[43] W. S. N. Asa Ben-Hur, "Choosing negative examples for the prediction of protein-protein interactions," *BMC Bioinformatics 7*, 2006.

[44] "My pymol script repository," 2014.

[45] P. M. R. M. Baris E. Suzek, Hongzhan Huang and C. H. Wu, "Uniref: comprehensive and non-redundant uniprot reference clusters,"

Annexes

Supplementary Material

Table S1. The original values of the seven physicochemical properties for each amino acid.

code	H_1	H_2	V	P_1	P_2	SASA	NCI
A	0.62	-0.5	27.5	8.1	0.046	1.181	0.007187
C	0.29	-1	44.6	5.5	0.128	1.461	-0.03661
D	-0.9	3	40	13	0.105	1.587	-0.02382
E	-0.74	3	62	12.3	0.151	1.862	0.006802
F	1.19	-2.5	115.5	5.2	0.29	2.228	0.037552
G	0.48	0	0	9	0	0.881	0.179052
H	-0.4	-0.5	79	10.4	0.23	2.025	-0.01069
I	1.38	-1.8	93.5	5.2	0.186	1.81	0.021631
K	-1.5	3	100	11.3	0.219	2.258	0.017708
L	1.06	-1.8	93.5	4.9	0.186	1.931	0.051672
M	0.64	-1.3	94.1	5.7	0.221	2.034	0.002683
N	-0.78	2	58.7	11.6	0.134	1.655	0.005392
P	0.12	0	41.9	8	0.131	1.468	0.239531
Q	-0.85	0.2	80.7	10.5	0.18	1.932	0.049211
R	-2.53	3	105	10.5	0.291	2.56	0.043587
S	-0.18	0.3	29.3	9.2	0.062	1.298	0.004627
T	-0.05	-0.4	51.3	8.6	0.108	1.525	0.003352
V	1.08	-1.5	71.5	5.9	0.14	1.645	0.057004
W	0.81	-3.4	145.5	5.4	0.409	2.663	0.037977
Y	0.26	-2.3	117.3	6.2	0.298	2.368	0.023599

H_1, hydrophobicity; H_2, hydrophilicity; V, volume of side chains; P_1, polarity; P_2, polarizability; SASA, solvent accessible surface area; NCI, net charge index of side chains.

Computational prediction of Protein-Protein interaction with Discrete Cosine Transform

Igor Garrido da Cruz, Joel P. Arrais
Department of Informatics Engineering (DEI), Centre for Informatics and Systems of the University at Coimbra (CISUC)
University of Coimbra
Coimbra, Portugal
igorcruz@student.dei.uc.pt, jpa@uc.pt

José Luís Oliveira
Department of Electronics, Telecommunications and Informatics (DETI), Institute of Electronics and Telematics Engineering of Aveiro (IEETA)
University of Aveiro
Aveiro, Portugal
jpa@uc.pt

***Abstract*— Understanding life at a molecular level encloses a myriad of opportunities. As important as being able to identify the molecular components of the cell it is of foremost relevance to understand their relationships. The study of Protein-Protein Interactions (PPI) has been a cornerstone to understand how biological processes take place. Despite the advance of laboratorial techniques the problem requires the use of computational methods to determine the protein interaction networks at the organism level. In this paper we propose an improved sequence-based method for predicting protein interactions. The usage of the Discrete Cosine Transform (DCT) as a feature extraction strategy for proteins sequence alongside with a Support Vector Machine (SVM) classifier is introduced with this work. Several datasets were used for validation, and it outscores the state of the art of sequence-based methods. Using a gold dataset of 12968 PPIs with 3351 proteins of multi-organisms extracted from BioGrid and Negatome database a consistent result of 0.827 accuracy was attained.**

Keywords—discrete cosine transform; protein interaction; protein feature extraction; machine learning

I. INTRODUCTION

Protein-Protein Interaction (PPI) is the process where a pair of proteins physically binds in order to accomplish a biological function. These interactions are of critical importance because they influence the cellular macromolecular structures and functions. Indeed they are the main mediators for several biological processes including the intracellular signaling pathways [1] that correspond to the transmission of messages within cells. Having the knowledge of how proteins interact with each other can provide a great opportunity to understand pathogenesis mechanisms, and subsequently support the development of drugs focused on very specific disease pathways and re-targeting already commercialized drugs to new gene products [2].

At the cellular level, the operations that each cell need to execute for maintaining their function are sustained by PPIs. These interactions are represented as cascade of interactions that are mapped to a network. The study of protein interaction networks has practical applications in drug discovery, since it gives a better notion of how diseases work and allows researchers to more accurately target proteins interactions associated with a given pathology.

The most standard way for detecting PPIs is throughout the use of laboratorial techniques. These methods test *in vitro* if a pair of proteins is prone to interact or not. There are many techniques for predicting PPIs such as yeast Two-Hybrid Systems, Mass Spectrometry, Protein Microarrays and Fluorescence Resonance Energy Transfer [3], each having his own advantages and drawbacks. Some of them, despite being very accurate, are expensive and time consuming, others, despite being high throughput, result in low accuracy outputs [4].

According to the last release of Uniprot [5] the human being has around 135,000 proteins. If all pairs were to be tested there were an astonishing number of putative interactions C (135,000, 2), approximately 9 × (10^9). In addition to the problem of testing all the human protein interactions the problem gains an even bigger dimension when considering the interactions that happen between the human and other organisms. For instance, in the oral cavity there is evidence of around 2300 micro-organisms [6,7,8]. Considering the interactions of the human proteins with these micro-organisms the number of interactions to be tested expands largely to untreatable numbers, therefore emerging the need to accurately predicting protein interactions using computational methods.

Multiple computational methods have been developed in recent years. Among the multitude of methods some are based on the three dimensional structure [9,10], some based on functional domains [11] and others on the phylogenetic profiles [12]. Despite their major importance, these have limitations since prior biological knowledge about proteins is required and cannot be applied to datasets on which only protein amino-acid sequences are available.

Several authors have already addressed the problem of predicting PPIs from the amino-acid sequence. Shen *et al.*[13] developed a method that scores 83.9% accuracy on a human restricted dataset. Such method categorizes the amino-acids based on their physico-chemical properties and then counts the occurrences of triplets of such categories. Despite scoring good results, the method only focuses on each amino-acid and on its two most proximate ones, leaving out information that could be beneficial to predict long range interactions.

Guo *et al.*[14] verified the limitations on long range interactions on the features proposed by Shen and made an

improved method that looks at the proteins sequence information as a signal using the auto-covariance of that signal as features to classify PPIs. This method scored 87.36%, when applied to predicting the PPIs in an independent dataset *of S. cerevisiae.*

In this article we propose an improved sequence-based method for predicting protein interactions using the DCT (Discrete Cosine Transform) as a feature to describe proteins sequence, and the Support Vector Machine (SVM) classifier. DCT describes the sequence of the proteins taking in consideration the physical properties of each amino acid. It is responsible for transforming proteins with different lengths in the same number of features, ignoring high frequencies. The SVM classifier is thereafter used to give a good estimation if proteins with dimensional similar features interact or not.

II. Materials and Methods

A. Dataset

Three datasets were used to test the efficiency of the method under different conditions. The required biological data was collected from UniProt Knowledgebase (UniProtKB), BioGRID and Negatome.

UniProtKB is the central hub for the collection of functional information regarding proteins. Each entry contains the amino acid sequence, protein name, taxonomic data as well as supplementary annotations such as ontologies, classifications, cross-references, and clear indications of the quality of annotation in the form of evidence attribution of experimental and computational data. BioGRID [15] is an online interaction repository with data compiled through comprehensive curation efforts. The current version compiles 42,004 publications for 720,840 raw protein and genetic interactions from major model organism species. All interaction data are freely provided through our search index and available via download in a wide variety of standardized formats. Contrasting with other interaction databases, BioGRID provides protein interactions for multiple organisms. Negatome [16] is a collection of protein and domain pairs which are unlikely engaged in direct physical interactions. The database currently contains experimentally supported non-interacting protein pairs derived from two distinct sources: by manual curation of literature and by analyzing protein complexes from the PDB.

Dataset 1: This dataset consists of 6484 interactions of a pool of 3351 proteins extracted from the Negatome collection and an equal number of positive interactions from BioGrid. These proteins were the ones that were also available on UniProt in order to extract the amino-acid sequences. The positive half of the dataset was built searching BioGrid for known interactions of the same 3351 proteins used in the negative pool.

Dataset 2: The dataset number two was built with 10,000 protein interactions randomly selected from the BioGRID dataset. The negative interactions were a combination of the 6484 known negative interactions from Negatome used in Dataset 1 and 3516 random combinations of the protein pool used in the positive interactions. The dataset was also balanced having a total of 20000 protein interactions, 10000 positives and 10,000 negatives. This dataset allowed to test the classifier when applied to a more diverse data. In fact the protein pool of this dataset was higher than the previous, consisting of 9686 proteins .

Dataset 3: This dataset contains 20,000 protein interactions from 14470 proteins randomly selected from the BioGRID dataset. The negative interactions were obtained by randomly construct pairs of proteins from the positive. This strategy to obtain negative interactions is acceptable, since the probability of randomly selecting a positive interaction is very low. This dataset was used to test if our method could keep achieving good results independently of the usage of the Negatome.

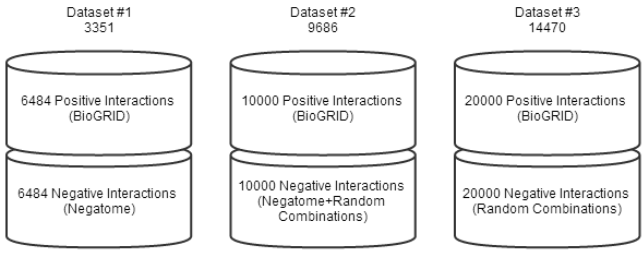


Fig. 1. Datasets used on the present work

B. Feature Extraction

At the primary level proteins are linear chains of amino-acids. In this approach, each protein sequence is represented by a signal that modulates the variations of amino-acids along the protein sequence.

The DCT expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. The DCT is well known for its practical applications in codecs such as MP3 or JPEG, allowing compression by discarding the higher frequencies.

In his previous work Shen *et al.* proposed that to reduce the dimensions of the vector space and suit synonymous mutation the 20 amino-amino acids could be transformed in 7 different categories calculated accordingly to their physico-chemical properties. In the table 1 there is the substitution table initially used by Shen *et al.* based on the dipole scale and in the volume scale. This table was used in the present work considering that similar amino-acids in the protein sequence can be susceptible to mutation.

TABLE I. Amino-Acids Substitution Table

a) Category	*b)* Amino-acids
1	Ala, Gly, Val
2	Ile, Leu, Phe, Pro
3	Tyr, Met, Thr, Ser
4	His, Asn, Gln, Tpr
5	Arg, Lys
6	Asp, Glu
7	Cys

[a.] According to their physico-chemical properties amino-acids are grouped in categories

The procedure used to extract features from a protein consists of getting its amino-acid sequence convert it to a vector of physico-chemical categories and then apply the DCT to the resulting vector. The signal is then reconstructed dependently of the number of features and concatenated with another signal in order to represent a protein interaction.

The DCT of a signal is given by following formula:

$$y(k) = w(k)\sum_{n=1}^{N} x(n)\cos\left(\frac{\pi}{2N}(2n-1)(k-1)\right), \quad k = 1,2,\dots,N,$$

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 1, \\ \sqrt{\frac{2}{N}}, & 2 \le k \le N, \end{cases} \tag{1}$$

And its inverse, for terms of signal reconstruction, is given by:

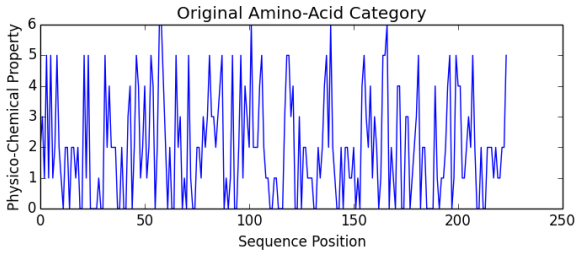
$$x(n) = \sum_{k=1}^{N} w(k)\,y(k)\cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), \quad n = 1,2,\dots,N,$$

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 1, \\ \sqrt{\frac{2}{N}}, & 2 \le k \le N, \end{cases} \tag{2}$$

An arbitrary number of frequencies (F) can be used to represent a protein. If the protein is bigger than F, the first F frequencies are selected. If smaller, zeros are padded until the number of desired features is archived.

After having the frequencies that describe the signal the inverse formula is used to reconstruct the original signal and to apply a standard normalization. This new signal is less noisy, since the high frequencies are ignored. It also has the same length for all the proteins and can be used to solve the classification problem. By doing this, it is possible to have representations of the proteins as a whole.

On the top of Fig. 2, the protein *A0AQH0* sequence after substitution with the physico-chemical categories is shown. On the bottom the same protein is shown after the reconstruction using the DCT with 600 features and performing a standard normalization ready for concatenation with other protein and further classification.



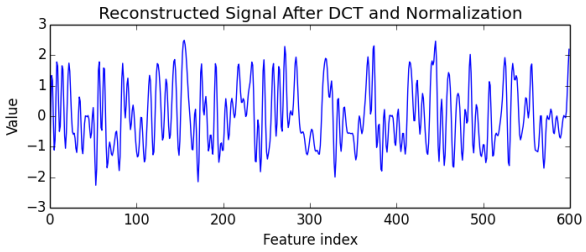


Fig. 2. Example of the DCT extraction and reconstruction applied to A0AQH0 Protein with 600 features

C. *Choosing the Optimal Number of Features*

The number of frequencies used in the DCT can be manually selected. However there is the need to test different values to have some indication of how many should be used in order to attain the optimal results.

One of our first tests was made using the KNN classifier in order to choose the number of features to be used.

Fig. 3 presents the evaluation of the classifier accuracy when compared with the increase in the number of features. The line in orange contains the raw frequencies extracted from the protein features. Cleary this result is not as good as the other methods. In red the accuracy score using the reconstruction of the protein amino-acid chain while considering the original 20 amino-acids is shown. Finally in blue there is the representation of a replacement of the 20 amino-acids with the 7 categories where feature extraction and signal reconstruction were performed. The peak occurs in the blue line while using 600 frequencies achieving an accuracy of 0.825, so for the rest of the work the strategy of using a reconstructed signal after substitution of the amino-acids with the 7 amino-acid categories.

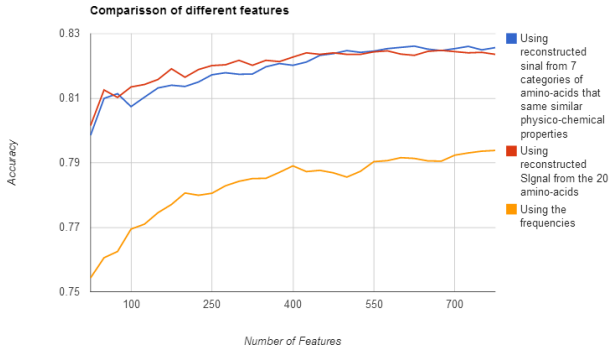


Fig. 3. Number of Features versus accuracy

D. *Classifiers*

In the present work two classification methods were tested, the K-Nearest Neighbor (KNN) classifier and the Support Vector Machine (SVM). On this section the results achieved with them on our datasets are studied.

1) Choosing KNN Parameters

The KNN classifier uses as parameters the number of neighbors to consider in order classifying a sample as being of one class or of another. In order to optimize our classifier an experimentally evaluation of this parameter was made in the range from 3 to 19.

As present in the Fig. 4 the optimal number of neighbors for classification was 9, achieving the accuracy of 0.825, the test was made using the dataset number one with cross-validation with 5 folds.

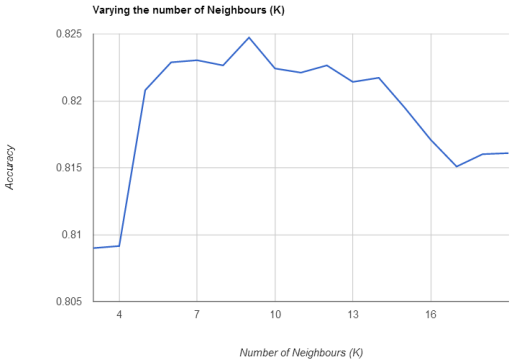


Fig. 4. Number of Neighbours versus accuracy

2) SVM Parameters

The SVM classifier with RBF kernel, as used in the present work uses two parameters, C and gamma. In order to test our method, these parameters were changed in the window shown in Fig 5, using the dataset number one and cross-validation with 5 folds.

As shown in Fig. 5 the optimal parameters were C = 100 and gamma = 0.001, achieving an accuracy of 0.827. Such parameters were afterwards used for testing with the other datasets.

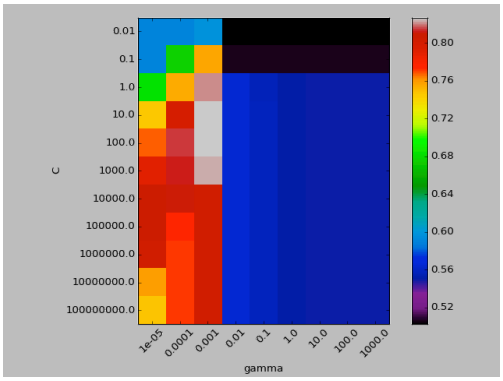


Fig. 5. Accuracy evalutation when comparing the Gamma and C parameters

E. Classifier Evaluation

After making the experimental estimation of the number of features to use with our method, and the best parameters for each classification method, the obtained results using dataset one were the following:

TABLE II. RESULTS ACHIEVED WITH OUR METHOD

Classifier	Accuracy	Precision	Recall
Nearest Neighbor classifier	0.825 +/- 0.015	0.814 +/- 0.011	0.845 +/- 0.008
SVM classifier	0.827 +/- 0.016	0.817 +/- 0.012	0.843 +/- 0.008

Both classifiers present clearly good results. Despite that the best result were obtained using the SVM classifier that achieved 0.827 accuracy. The KNN classifier also gave relatively good results 0.825 accuracy. Further testing of these methods efficiency alongside with their execution time is discussed on the following section.

III. DISCUSSION

On this section a discussion of the results achieved with our method alongside with a comparison of the existing state of the art is made.

A. Accuracy comparision with the state of the art

It is important to test our method under diverse circumstances and on multiple datasets.

In our case different datasets were built, some of them were built using then Negatome as the database for negative interactions others the negative part was selected from random combinations of proteins from the protein pool.

In the Table II and Fig. 6 the results achieved with the different methods in the different datasets are presented. Fig. 6 presents the results of our method using the SVM classifier. It outperforms the existing state of the art methods while tested on every dataset. Although slower, the increase on the accuracy might compensate for the extra computing. The KNN method for classification of our features behaves acceptably. On some datasets it even outperformed the results achieved with the existing state of the art. However on the dataset 3 the results were not as good as expected revealing that its behavior might deteriorate on big, discrepant datasets. However it might still being a viable option due to its low amount of computing time.

TABLE III. PERFORMANCE COMPARISON

	Guo	Shen	Our method using KNN	Our method using SVM
Dataset 1 Accuracy	0,804 +/- 0,010	0,803 +/- 0,011	0,825 +/- 0,004	0,827 +/- 0,004
Dataset 1 Precision	0,769 +/- 0,017	0,797 +/- 0,014	0,814 +/- 0,011	0,817 +/- 0,012
Dataset 1 Recall	0,861 +/- 0,007	0,812 +/- 0,028	0,845 +/- 0,008	0,843 +/- 0,008
Dataset 2 Accuracy	0,720 +/- 0,012	0,746 +/- 0,014	0,765 +/- 0,004	0,761 +/- 0,004
Dataset 2 Precision	0,669 +/- 0,010	0,727 +/- 0,019	0,781 +/- 0,008	0,736 +/- 0,005
Dataset 2 Recall	0,867 +/- 0,026	0,782 +/- 0,014	0,665 +/- 0,010	0,789 +/- 0,009
Dataset 3 Accuracy	0,646 +/- 0,014	0,705 +/- 0,004	0,664 +/- 0,005	0,707 +/- 0,005
Dataset 3 Precision	0,639 +/- 0,012	0,741 +/- 0,009	0,761 +/- 0,009	0,678 +/- 0,007
Dataset 3 Recall	0,642 +/- 0,010	0,630 +/- 0,012	0,639 +/- 0,003	0,723 +/- 0,004

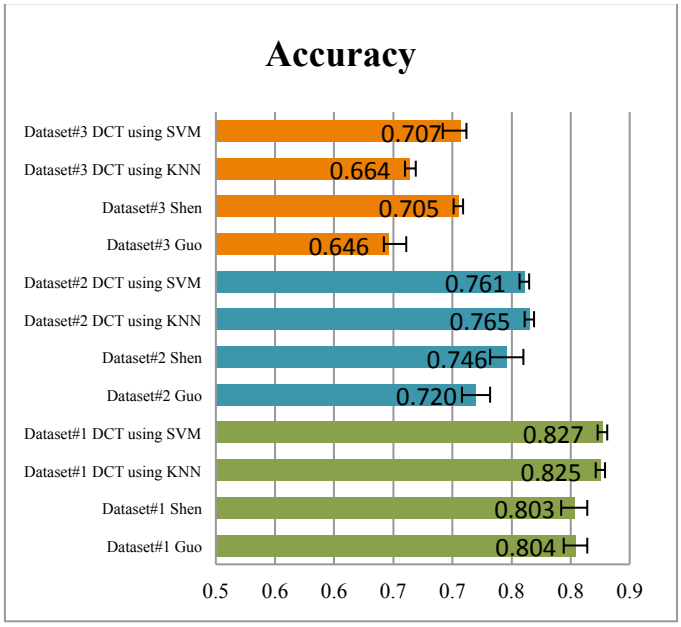


Fig. 6. Accuracy per method using different datasets

B. *Execution time comparision with the state of the art*

The differences in the execution time were noticed since the start of testing the existing methods. When testing in small datasets such differences are low and unnoticeable, however when working with bigger datasets similar to the ones that need to be tested in biological testing they increase significantly and need to be considered.

The following tests were made with the dataset number one since it was the smallest of them all there was limitations in terms of time. The execution times are showed in seconds and correspond to executions of feature extraction and cross-validation (training and test) of the dataset using 5 folds. The tests were made using i7-4700mq CPU and 8 GB of RAM.

Both in Table IV and Fig. 7 the execution time results are presented. Using KNN classifier with our feature extraction method is the fastest method tested by us. The classification method used by *Shen et al.* and *Guo et al.* is based on SVM, when using an SVM classifier our method was the slowest.

TABLE IV. PERFORMANCE COMPARISON

Method	**Execution Time (Seconds)** Execution 1	Execution 2	Execution 3	**Mean**	**Std Deviation**
Guo	5913	6212	5984	6036,33	127,55
Shen	7259	6496	6748	6834,33	317,42
Our method using KNN	595	568	617	593,33	20,04
Our method using SVM	11739	12791	11865	12131,67	469,05

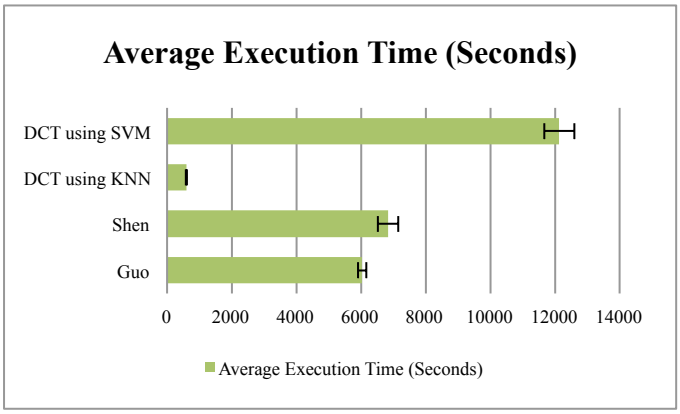


Fig. 7. Comparison of execution times

As we can see in the Fig. 6 our method using the SVM classifier outperforms the existing state of the art methods while tested on every dataset. It is slower, but the increase on the accuracy might compensate for the time spent on computing. The KNN method for classification of our features behaves acceptably. On some datasets it even outperformed the results achieved with the existing state of the art. However on the dataset 3 the results were not as good as expected revealing that its behavior might deteriorate on big, discrepant datasets. However it might still being a viable option due to its low amount of computing time.

C. *Using a Validation Dataset*

A further step of validation was made in this study in order to infer our classifiers capabilities under real conditions.

The dataset 2 was used for training the classifiers on this sub-section since it has a good balance between two factors. It includes the Negatome, providing a good indication of the negative interactions, but not limiting the information to the proteins contained on such dataset proteins pool.

In order to test the classifier perfomance on real data, 1000 random interactions were selected from the BioGRID to build the independent validation dataset.

Performing training and validation with these datasets we obtained the results present in Fig. 8.

Our method was the one that achieved the best performance hitting 80.7% of the known positive interactions and with a great margin of performance above the other tested classifiers.

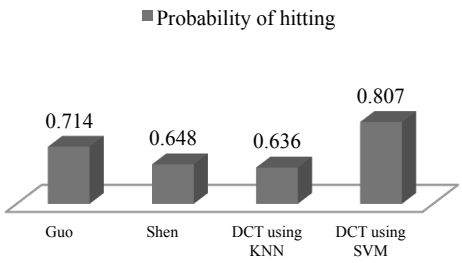


Fig. 8. Comparison of methods using an independent validation dataset

IV. Conclusions

The representation of the protein sequence as a whole allows the classifiers to predict interactions seeing proteins as a whole instead of dividing it in smaller problems as is explored by other machine learning techniques.

Most of the times authors do not provide the datasets used in their works or just select smaller sub-sets selected by them. Consequently it is not possible to replicate the metrics presented in the state of the art in order to make direct comparisons. To this end we tried to be as systematic as possible in the definition of training/validation datasets. In addition to the performance improvement the SVM classifier allied to our method proven to be more resilient to changes.

Based on the results achieved, our method can compete with the state of the art methods. Considering the accuracy results our method outscored the existing methods while using the SVM classifier on every dataset tested. Independently of number of interactions, number of proteins, and method of choosing negative samples for the dataset, the results achieved were always better, achieving the peak of 0.827 accuracy on our golden dataset.

The KNN classifier showed good results while using the Negatome as a source for non-interacting protein pairs. However when tested with randomly selected protein pairs from the protein pool as negative examples the results were poor. Despite this, it is still a viable classification method to do some experiments since it execution time is ten times faster than the next classifier.

Acknowledgment

The Faculty of Dental Medicine from the Universidade Católica Portuguesa had a big impact on this work for reasons of biological consulting, allocation of computing time and validation of the practical results on real datasets and for that we would like to acknowledge their support.

References

[1] T. Pawson and P.Nash, "Protein-protein interactions define specificity in signal transduction", *Genes Dev*, 2000

[2] Coelho ED, Arrais JP, Oliveira JL.,From protein-protein interactions to rational drug design: are computational methods up to the challenge?,Current topics in medicinal chemistry,13,5,602-618,2013,Bentham Science Publishers

[3] Guan H, Kiss-Toth E., "Advanced technologies for studies on protein interactomes", *PubMed*, 2008

[4] Chishe Wang, Jie Song, Fangping Li, Junsong Lv, "Identifying Protein-Protein Interaction Sites Using Adapted Bayesian Classifier", *BMC Bioinformatics*, 2009

[5] The UniProt Consortium (2012). "The protein space at the Universal Protein Resource (UniProt)", *Nucleic Acids Res. 40*: Database issue (in press).

[6] Arrais JP, Rosa N, Melo J, Coelho ED, Amaral D, Correia MJ, Barros M, Oliveira JL.,"OralCard: a bioinformatic toolfor the study of the oral proteome", , Archives of oral biology, 2013

[7] Coelho, Edgar D; Arrais, Joel P; Matos, Sérgio; Pereira, Carlos; Rosa, Nuno; Correia, Maria José; Barros, Marlene; Oliveira, José Luís; ,Computational prediction of the human-microbial oral interactome,BMC systems biology,8,1,24,2014,BioMed Central Ltd

[8] Loo JA, Yan W, Ramachandran P, Wong DT, "Comparative Human Salivary and Plasma Proteomes", *Journal of Dental Research 89*, 2010

[9] Rodriguez-Soca Y, Munteanu CR, Dorado Jn, Pazos A, Prado-Prado et al.,"Trypano-PPI: A Web Server for Prediction of Unique Targets in Trypanosome Proteome by using Electrostatic Parameters of Protein−protein Interactions", *Journal of Proteome,* 2009

[10] Rodriguez-Soca Y, Munteanu CR, Dorado J, Rabuñal J, Pazos A,"Plasmod-PPI: A web-server predicting complex biopolymer targets in plasmodium with entropy measures of protein–protein interactions", 2010

[11] Eddy. SR., "Prole hidden markov models", *Bioinformatics 14*, 1988

[12] Sun J., Xu J., Liu Z., Liu Q., Zhao A., Shi T., Li Y., "Refined phylogenetic profiles method for predicting protein-protein interactions", *PubMed*, 2005

[13] Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, Li Y, Jiang H., "Predicting protein– protein interactions based only on sequences information", *PubMed*, 2007

[14] 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*, 2008

[15] Stark C., Breitkreutz B.-J., Reguly T., Boucher L., Breitkreutz A. and Tyers M., "Biogrid", *Nucleic Acids Research*

[16] Smialowski P1, Pagel P, Wong P, Brauner B, Dunger I, Fobo G, Frishman G, Montrone C, Rattei T, Frishman D, Ruepp A., "The Negatome database: a reference set of non-interacting protein pairs.", *Pub Med*, 2010