Faculdade de Ciências e Tecnologia
da Universidade de Coimbra

COMPUTATIONAL METHODOLOGIES FOR PREDICTING PROTEIN-PROTEIN INTERACTIONS

João Miguel Pereira Rebordão Castanheira

Dissertação de Mestrado na área científica de Bioquímica orientada pelos Senhores Professores Doutores Joel Perdiz Arrais e Nuno António Marques Lourenço apresentada ao Departamento de Ciências da Vida da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Setembro de 2018

UNIVERSIDADE Ð
COIMBRA

"Quem quis sempre pôde"

Luís Vaz de Camões

To my father Luís Alberto and to my mother Ana Maria I can´t say anything better than thank you, thank you and thank you

ACKNOWLEDGEMENTS

For the success of my academic career, I thank the support, values and culture imparted by my parents and the rest of the family, the companionship of my friends, the training that my country has given me in terms of infrastructures and professionals and to God for His constant presence. In the particular case of this thesis, I thank the support of my advisors Prof. Dr. Joel Arrais and Prof. Dr. Nuno Lourenço and the help in the art of programming of André Santiago, António José Preto and Enoque Moisés.

TABLE INDEX

FIGURE INDEX

LIST OF ABBREVIATIONS

AAC: Amino acid composition descriptors

ACC: Accuracy

AL: Active Learning

AUC: Area Under the ROC Curve

CT: Conjoint triad features

CTD: Composition, Transition, Distribution descriptors

DNA: Deoxyribonucleic acid

F1: F-score

ID: Protein identifier

KNN: k – Nearest Neighbour

LR: Logistic Regression

LRR: Leucine-rich repeat

MCC: Matthews Correlation Coefficient

Moranauto: Moran autocorrelation descriptors

NNM: Neuro Network Model

NPV: Negative Predictive Value

PPI: Protein-protein interaction

PPV: Precision or Positive Predictive Value

PTM: Post-Translational Modifications

PyDPI: Drug-Protein Interaction with Python

QSO: Quasi-sequence order descriptors

RNAi: Ribonucleic acid interference

ROC: Receiver Operating Characteristic

SGD: Stochastic Gradient Descent

SH2: Src (gene) Homology 2

SH3: Src (gene) Homology 3

SOCN: Sequence order coupling numbers

SSL: Semi-Supervised Learning

SVM: Support Vector Machine

TNR: Specificity or True Negative Rate

TPR: Sensitivity, Recall or True Positive Rate

Tree: Decision Tree

RESUMO

Devido à relevância das interações proteicas nas diferentes funções celulares, é importante conseguir detetar a existência das mesmas. Como os métodos computacionais conseguem lidar com um grande número de dados de forma rápida, têm sido muito usados na previsão das interações proteína-proteína. Dessa forma esta tese pretende, com o recurso a novas features, desenvolver um método que melhore a performance da previsão de interações proteicas num dataset aleatório. O resultado destas pesquisas foram três novas abordagens de extração de features sendo elas o recurso a bases de dados de inibidores, o recurso a redes de co-expressão génica e o recurso a módulos de reconhecimento peptídico. Destas três, devido à maior simplicidade e praticabilidade, desenvolveu-se estudos usando a última abordagem enunciada, recorrendo nomeadamente aos domínios SH3, SH2, PDZ, WW e LRR. Para saber se estes domínios são uma boa fonte de features e que podem ser usados utilizando qualquer dataset, analisaram-se os mesmos na deteção de novas interações entre proteínas que não os possuem. Assim no decorrer do trabalho desta tese foram criadas três estratégias, a primeira baseava-se na extração de features pelo software PyDPI (fazendo uso dos descriptors AAC, CTD, Moranauto, QSO, SOCN e CT) e na avaliação da performance dos datasets por descriptor; a segunda estratégia recorreu às mesmas features mas avaliou a performance em datasets com todos os descriptors; a terceira estratégia avaliou a performance como a segunda estratégia mas usando features criadas para o artigo *"A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions"*. As duas primeiras estratégias foram postas de parte devido a metodologia incorreta e a valores pouco significativos. A terceira estratégia apesar de ter sido efetuada de uma forma bastante controlada também resultou em valores pouco significativos, pelos que se aconselha ao prosseguimento do estudo desta abordagem com novas estratégias e features.

Palavras-chave: Interação proteína-proteína, melhorar performance, extração de features, módulos de reconhecimento peptídico, PyDPI

ABSTRACT

Due to the relevance of protein interactions in different cellular functions, it is important to be able to detect their existence. Because computational methods can handle large numbers of data quickly, they have been widely used in predicting protein-protein interactions. Thus, this thesis intends, with the use of new features, to develop a method that improves the predictive performance of protein interactions in a random dataset. The results of these researches were three new approaches of extraction of features, being: the use of databases of inhibitors, the use of gene co-expression networks and the use of peptide recognition modules. Of these three, due to the greater simplicity and practicality, studies were developed using the last approach enunciated, resorting in particular to the SH3, SH2, PDZ, WW and LRR domains. In order to know if these domains are a good source of features and that can be used using any dataset, they were analyzed in the detection of new interactions between proteins that do not possess them. Thus, in the course of the work of this thesis three strategies were created, the first one was based on the extraction of features by the software PyDPI (making use of descriptors AAC, CTD, Moranauto, QSO, SOCN and CT) and in the performance evaluation of datasets by descriptor ; the second strategy resorted to the same features but evaluated the performance in datasets with all descriptors; the third strategy evaluated performance as the second strategy but using features created for the article "*The Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions*." The first two strategies were set aside due to incorrect methodology and poor values. The third strategy, despite being carried out in a very controlled manner, also resulted in insignificant values, for which it is advisable to continue the study of this approach with new strategies and features.

Keywords: Protein-protein interaction, performance inprovement, features extraction, peptide recognition modules, PyDPI

INDEX

1. INTRODUCTION

1.1 Motivation

Currently, due to the great increase of information stored in databases regarding new protein genomes, the experimental methods that allow to unravel the networks of protein interaction (are the only ones able to validate protein interactions) are no longer relevant. Not only because they cover a small fraction of the protein network but also because they are costly in terms of time and money and have little precision and accuracy. In order to address these shortcomings, computational methods were developed to predict regions of protein interaction more correctly and economically than experimental methods (Melo, *et al.*, 2016)(You, *et al*., 2015).

However, because of the need to study a large amount of information (there are many sequenced proteins whose interactions have not yet been studied), existing computational methods have been very time consuming and not very rigorous in detecting interactions (weak classifiers), especially when interactions cause conformational changes (Reimand, *et al.*, 2012). For these reasons, it is imperative to improve existing methods, not only for a faster computational protein analysis but also for a more correct analysis.

To solve the problem of improving computational methods, several strategies were developed. Those strategies make use of the collection of new features from biochemical knowledge, especially from inherent characteristics to proteins: prevalence of amino acids, peptides (recognition or structural), functionality, presence in protein interfaces (interprotein contact), existence of water molecules or co-relational networks.

Thus, to contribute to the improvement of computational methods, this thesis will make use of some of the strategies listed above.

1.2 Goal

In the case of this work, in order to improve the prediction method of protein interactions in a random dataset, was used as solution, protein features whose interactions occur from the peptide recognition modules SH3, SH2, WW, PDZ and LRR [see Figure 1].

This objective follows the recent strategies of improvement of the methods of prediction of protein interaction, which from biochemical knowledge are added new features to an already developed computational method of prediction.

One advantage of the use of features derived from proteins, whose interactions are mediated by peptide recognition modules, is that they follow the feature extraction method, which, computationally, is less demanding than an interface analysis and much more simple. This is because in computational methods, there are two ways to predict the probability of interaction between proteins: one is from the analysis of the interfaces where the protein intersections occur (hot spots), the second is by feature extraction. As for the feature extraction method, it works from a database that holds several sequences that are responsible for protein intersections, then compare the characteristics of that sequences (features) with the ones of the proteins that are intended to study, if they have these features it is probable that in that region there is an intersection (You, *et al*., 2015)(Coelho, *et al*., 2013)(Coelho, *et al*., 2014)(Maruyama, O., 2013).

Finally, it is worth mentioning that the use of only datasets to prove the relevance of the use of these new features (two different types of datasets are opposite, one with features of protein pairs that only have generic protein interactions and another dataset equal to the previous one but with features of protein pairs with interactions mediated by the domains cited above), is an extremely easy and practical method whose relevance, when computationally demonstrated, requires an experimental test to prove its veracity.

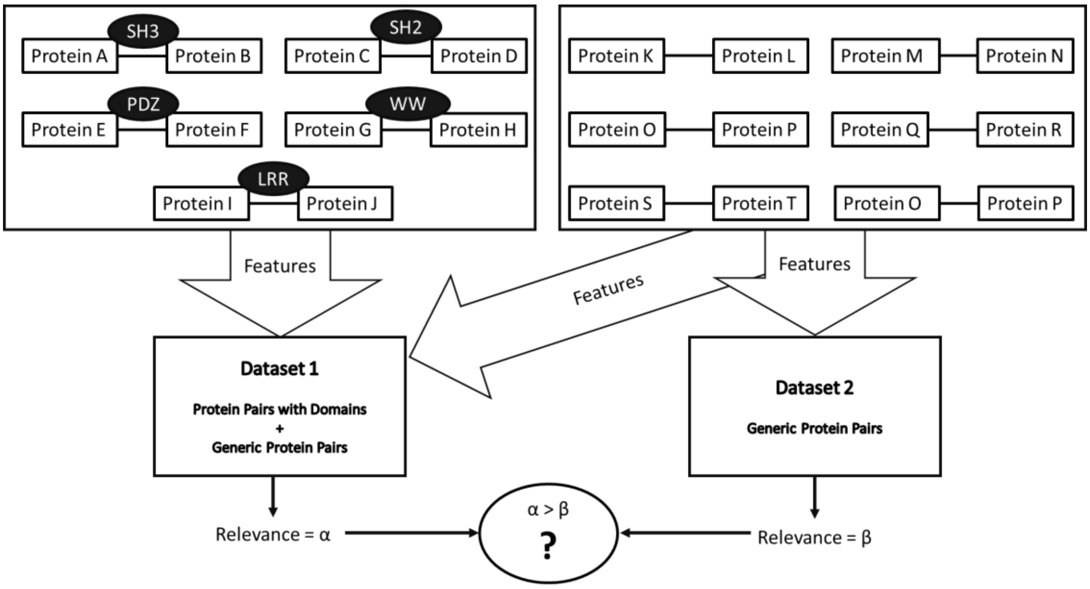


Figure 1: Model developed in this thesis

2. BACKGROUND

2.1 Machine Learning

Machine learning consists of the ability of a computer program to improve the performance of a task class from experience. Thus, to have a model of machine learning is therefore necessary to define a task, a source of empirical values and performance measure to improve. In the case of this thesis, the task is recognize protein-protein interactions, the source of empirical values are the characteristics (protein-protein interactions, protein localization, biological processes of the same proteins ...) present in the databases in which are the homologous proteins intended to study, the performance measure are the protein-protein interactions correctly detected (values achieved by the use of classifiers). In a general way, a machine learning model, adapted to this thesis, will use the mentioned characteristics and divided them into two groups, those of the characteristics present in homologous proteins that have interaction in the Y site and that of the characteristics present in homologous proteins that have no interaction in the Y site; the results will came in the form of a distribution of conditional probabilities that predict if two proteins have interactions and where (You, *et al.*, 2015)(Maruyama, O., 2013)[see Figure 2].

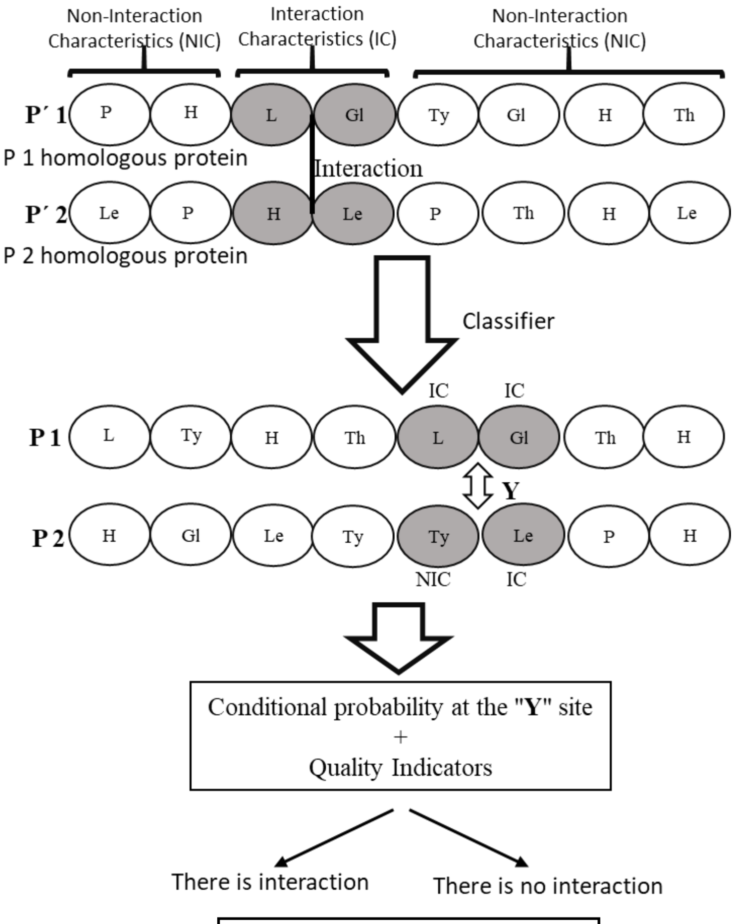


Figure 2: Functioning of Classifiers

2.1.1 Performance Metrics

In order to analyse the quality of the different classifiers as well as the relevance of the outputs there are several parameters that evaluate them, in particular:

Confusion matrix (You, *et al.*, 2015)(Maruyama, O., 2013)(Zhang, *et al.*, 2016)(Zahiri, *et al.*, 2013)		
Quality Indicator	Meaning	Calculation
True Positives (TP)	Predicted interactions that are confirmed experimentally	Analytical sum of cases
True Negatives (TN)	Predicted interactions that are experimentally refuted	Analytical sum of cases
False Positivies (FP)	Inexistence of predicted interactions that are confirmed experimentally	Analytical sum of cases
False Negatives (FN)	Inexistence of predicted interactions that are experimentally refuted	Analytical sum of cases
Sensitivity, Recall or True Positive Rate (TPR)	Percentage of interactions between correctly identified proteins	$TPR = \frac{TP}{TP + FN}$
Specificity or True Negative Rate (TNR)	Percentage of non-interactions between correctly identified proteins	$TNR = \frac{TN}{TN + FP}$
Precision or Positive Predictive Value (PPV)	Proportion of interactions between detected proteins to the detriment of existing interactions	$PPV = \frac{TP}{TP + FP}$

Quality Indicator	Meaning	Calculation
Negative Predictive Value (NPV)	Proportion of interactions between non-existent proteins detected to the detriment of all non-existent proteins	$NPV = \frac{TN}{TN + FN}$
Accuracy (ACC)	Proportion of correctly detected data (if you have a very high value of TN is no longer a good evaluator)	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$
Matthews Correlation Coefficient (MCC)	More accurate accuracy measurement	$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}}$
F-score (F1)	Evaluates overall method performance (balances sensitivity and precision when unbalanced results are obtained – many negatives)	$F1 = \frac{2 \times TPR \times PPV}{TPR + PPV}$
Receiver Operating Characteristic (ROC)	The closer the ROC chart line is to its left and top side margins, the more rigorous the method	Obtained by the graphical representation of the precision [PPV] versus negative predictive valiue [NPV]
Area Under the ROC Curve (AUC)	It is the probability of a classifier to classify a random data as positive instead of a random data as negative	$P = TP + FP \qquad N = TN + FN$ $AUC = \frac{S - \frac{P(P+1)}{2}}{P \times N}$ * S is the sum of the classifications of all positive samples in the list of all samples, ranked in ascending order by estimated probabilities belonging to positive samples.

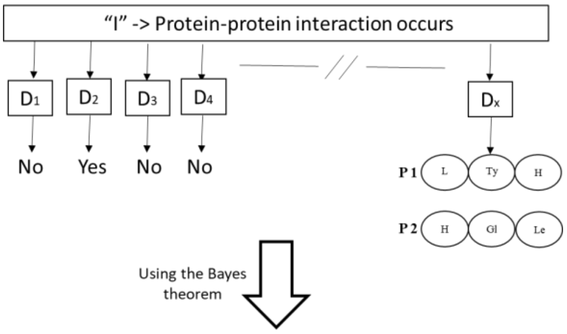
Table 1: Confusion matrix: Indicator, Meaning and Calculation

2.1.2 Methods

Currently there are several classifiers for his frequent use seven stand out, naïve Bayes, k-nearest neighbors, support vector machine, logistic regression, stochastic gradient descent, decision tree and neural network models.

The **naïve Bayes** classifier is a probabilistic model based on the Bayes theorem that calculates the probability of a given instance belonging to a class. In this case, "I" is a variable that represents the class "protein-protein interaction" and "D" the variable representing pairs of protein sequences to be studied (Maruyama, O., 2013)(Zhang, *et al*., 2016)(Zahiri, *et al.*, 2013). "I" is a random variable that has a value of 1 (there is interaction) or 0 (no interaction) and "D1, D2, ... Dx" are random variables (independent of each other), each representing a pair of protein sequences (Maruyama, O., 2013)(Zhang, *et al.*, 2016)(Zahiri, *et al.*, 2013)[see Figure 3].

The **k-nearest neighbor** classifier is one of the simplest methods to the extent that he attributes to the result "W" of a sample, the category of the nearest k results. In that terms, for different values of k there can be different classes assigned to W, depending on the features covered by k (Zahiri, *et al.*, 2013)(Hue, *et al.*, 2010)[see Figure 4].



$$P(I \mid D1, D2, D3, D4, \ldots, Dx) = \frac{P(D1, D2, D3, D4, \ldots, Dx \mid I)P(I)}{P(D1, D2, D3, D4, \ldots, Dx)}$$

Figure 3: Operation of the Naive Bayes method

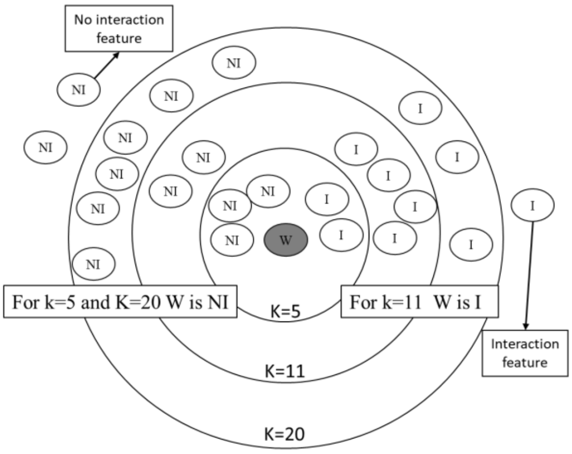


Figure 4: Operation of the nearest k-neighbour

The **support vector machine** classifier is a machine learning technique that seeks to find the optimal "hyperplane" that separates the samples from different classes. Achieving the optimum hyperplane consists in reaching the maximum distance between the samples that indicate there is protein-protein interaction from those indicating otherwise (You, *et al.*, 2015)(Zahiri, *et al.*, 2013). To handle the samples, we use kernel functions capable of categorizing the different samples in a high-dimensional space. These functions can be sigmoid, polynomial, linear or radial-based (You, *et al.*, 2015)(Zahiri, *et al.*, 2013)[see Figure 5].

The **logistic regression** consists in a regression model in which is usually used a binary dependent variable that is categorical, acquiring only the values 0 and 1 (Walker, *et al.*, 1967). From the logistic regression curve made by the data (0 and 1 correspond to the values in the data), it´s estimated the probability of a new data value (Walker, *et al.*, 1967)[see Figure 6].

The **stochastic gradient descent** classifier is an iterative method that minimizes a loss function with a linear function (Forcier, *et al.*, 2015). Taking one sample at a time, the algorithm approximates a true gradient and updates the model simultaneously based on the loss function gradient (Forcier, *et al.*, 2015). For regression, it returns predictors as sum minimizers (Forcier, *et al.*, 2015).

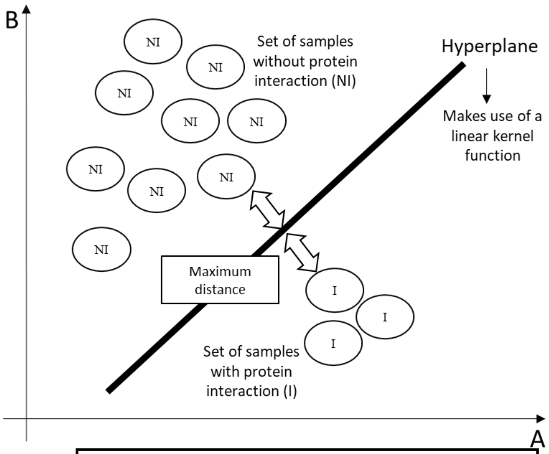


Figure 5: Support Vector Machine operation

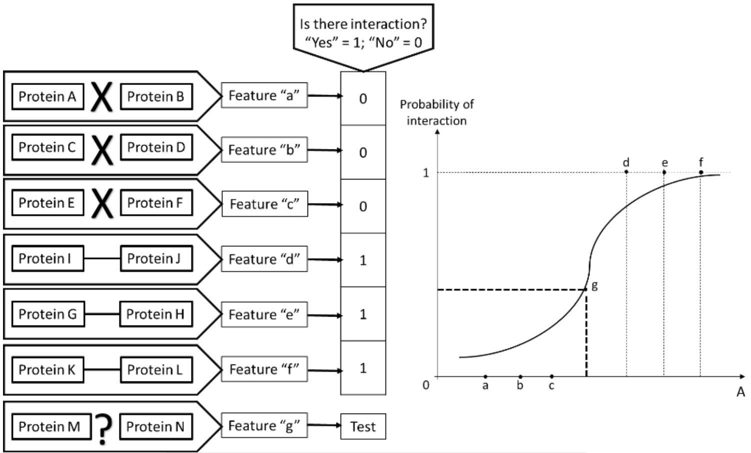


Figure 6: Logistic Regression method

The **decision tree** classifier is a machine learning technique characterized by its low computational cost (Zahiri, *et al.*, 2013). In the training phase, this technique constructs trees whose branches are the positive or negative response to a certain value, so that the feature dataset is subdivided recursively until we obtain subsets whose division does not change the classification (Zahiri, *et al.*, 2013)[see Figure 7].

Within the **neuro network models** the most used is the multilayer perceptron. This algorithm contains three types of layers the input layer, the hidden layer (may be more than one), and the output layer (Zahiri, *et al.*, 2013). Each layer has several nodes, each of these nodes being connected to all nodes of the next layer; in this algorithm the input layer nodes correspond to the features and the hidden/output layers to the processing units that act as neurons with an activation function in which the output layers classify the input ones by placing them in a class (Zahiri, *et al.*, 2013). In the case of the edges their weights are optimized from a supervised learning approach in order to improve the performance of the model (Zahiri, *et al.*, 2013)[see Figure 8].

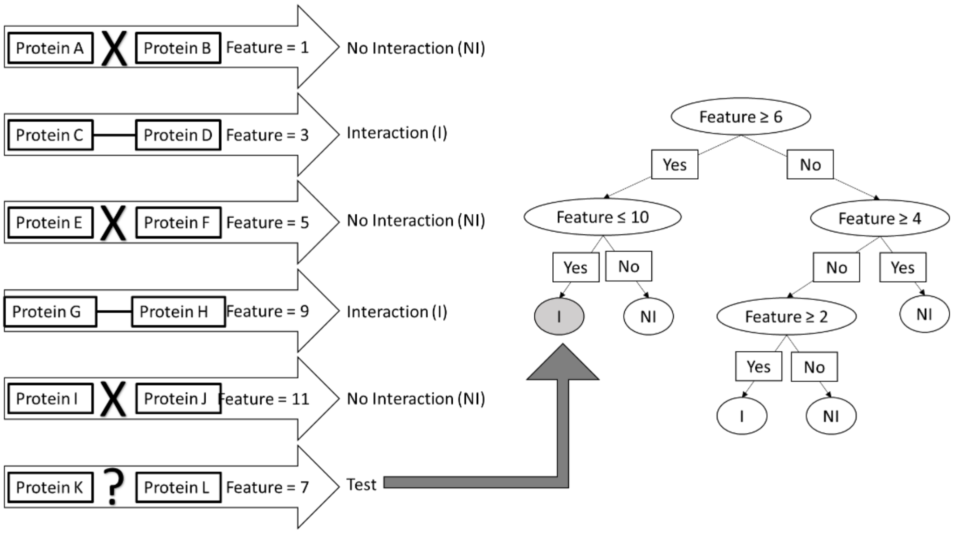


Figure 7: Operation of Decision Tree method

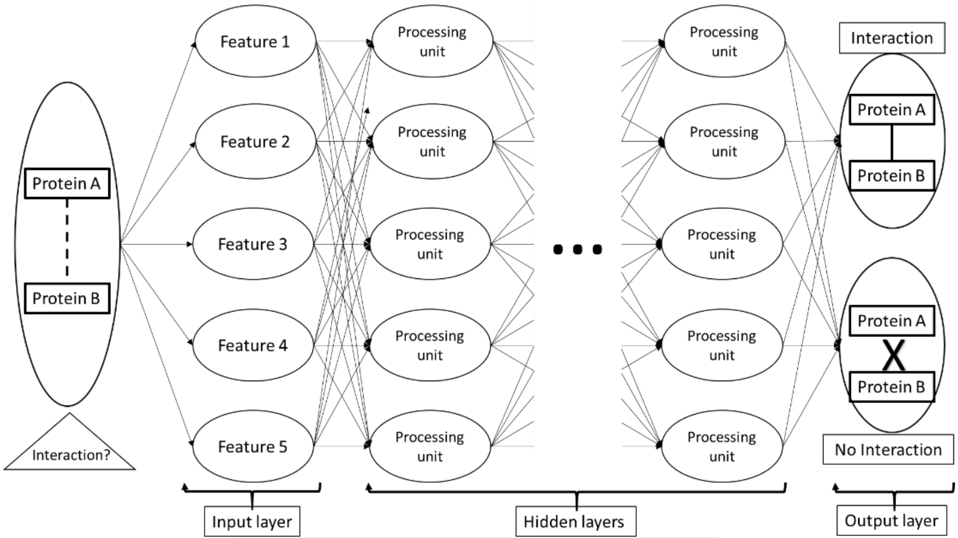


Figure 8: Operation of Neuro network model

2.2 Protein-Protein Interactions

Proteins are vital in cellular activity, mediating signal transduction, intracellular transport, secretion, cell cycle, DNA replication, translation, transcription and splicing. To perform these tasks proteins need to bind to other biomolecules with interactions, so that to better understand cell activity it is important to identify and characterize interactions between biomolecules (proteins) and their global network (Melo, *et al.*, 2016)(Coelho, *et al.*, 2013)(Coelho, *et al.*, 2014).

With respect to protein interactions, these may be characterized in several ways: localization - may interact with one or more polypeptide chains; force - can be permanent or transient; specificity - can be specific or non-specific; chemical interaction - may be covalent or non-covalent; similarity of the interacting subunits - may be homo or hetero-oligomers (Coelho, *et al.*, 2013).

A fundamental rule in the interaction between proteins is that in order to occur, the two proteins have to be complementary, which is dependent on the interface size, the polar and nonpolar residue alignments, and the number of water molecules (Moreira, *et al.*, 2006). Moreover, in the regions where they interact, there are conserved sequences that allow them to have physical-chemical and structural affinity (Coelho, *et al.*, 2013)(Coelho, *et al.*, 2014).

Within these sequences, most of the binding energy is quantified from the interaction between compacted and centralized regions in the form of small residues (mainly tryptophan, arginine and tyrosine) called hot spots (Melo, *et al.*, 2016)(Moreira, *et al.*, 2006). Hot-spots are usually organized in clusters and located near the centre of the binding region having a hydrophobic ring conformation, charged residues that form salt bridges and hydrophobic residues that bind directly to another protein (Moreira, *et al.*, 2006). They present great adaptability because different proteins bind to the same hot-spot (Moreira, *et al.*, 2006).

Finally, the stabilization of these interactions is ensured by hydrophobicity (main force), by a gain in the free energy variation, by the energy of desolvation and by the interactions of van der waals (Gurung, *et al.*, 2017). Hydrogen bonds are also of great importance because, in addition to composing a fifth of the interface region, they contribute positively to the bond free energy and are the ones that most promote the electrostatic interactions which stipulate the temporal longevity of the interprotein complex (Gurung, *et al.*, 2017)(Moreira, *et al.*, 2007).

2.2.1 Review of the state-of-the-art

In this section a survey of the previous work is carried out, evaluated by the mentioned indicators.

Objective: "Improve methods for predicting protein-protein interactions" (Huang, *et al*., 2015)

To predict PPIs from amino acid sequences was used at first, substitution matrix representation based on BLOSUM62 to portrait proteins as SMR matrixes. After that discrete cosine transform was used to construct a 400-dimensional vector from the SMR matrixes, this lead to a 800-dimensional feature vector for each protein pair. WSRC was then used as a classifier in the PPIs datasets of Yeast, Human and H. pylori and its performance was compared with the support vector machine classifier performance descript in the literature. In other part of this work, cross-species experiments were made in five PPI datasets, where using the concept of the homologous proteins, identified interactions in one organism was used to predict the interactions in the other five. The results confirm the relevance of the developed model in detecting interacting protein pairs and present a better performance than the previous methods.

Objective: "Discover the best method to detect hotspots" (Melo, *et al*., 2016)

To discover the best method to detect hotspots were increased protein-protein complexes and 3D complex structure-based features like interface size, the type of interaction at the interface and the number of types of residues at the connection spot. In terms of features, a relevant point was the use of Position-Specific Scoring Matrix in the evolutionary sequence-based features. In the performing method, 27 algorithms were tested with many classifier models using different cost functions. The results based on the performance of conditional inference random forest show that the best predictor was c-forest algorithm, making these values higher than the previous techniques.

Objective: "Construct a new method to better detect the sequence pairs of protein-protein interactions" (You, *et al*., 2015)

To detect pairs of sequences of protein interactions, each protein sequence was first transformed into feature arrays, from which a new matrix-based protein descriptor is extracted and numbered each protein sequence. Then, pairs of proteins with several characteristic vectors were analysed, encoding the vectors of two proteins in that pair of proteins. To evaluate the performance, a SVM model was used that has as input the vectors of the protein pair, in PPI datasets of Saccharomyces cerevisiae and Helicobacter pylori. The results demonstrate the relevance of this method in the detection and protein interactions.

Objective: "Combine active learning and semi-evaluated learning by support vector machine to improve the detection of protein interactions" (Song, *et al*., 2011)

To improve the detection of protein interactions, a new PPI extraction technique was developed called PPISpotter, which uses SSL based on deterministic annealing and AL to extract PPI. A large number of MEDLINE records were also imported by Natural Language Processing (NLP) techniques, in which the syntactic, semantic and lexical properties of the text were very relevant in feature selection, which increased performance and resulted in better values of performance of SVM. As for the results, it has been proven by the use of three different PPI datasets, that the technique developed is superior to the other techniques studied such as Random Sampling, Clustering, and Transductive SVMs, proving to be an innovative and pertinent PPI detection technique.

Objective: "Develop methods that detect whether post-translational modifications (PTM) are within or outside the protein interaction region" (Saethang, *et al*., 2016)

As the first strategy to predict whether posttranslational modifications occur within or outside the regions of protein interaction, PDB and PhosphoSitePlus data describing the empirical location of the modifications were first used. Then from the use of features derived from the index of hydropathy, secondary structure, Position-Specific Scoring Matrix, sequence conservation and web applications, the prediction of the location of the modifications was done resulting in a low performance. The second strategy was to use machine learning, so the information from PhosphoSitePlus was coded using AAindex (database of numerical indices representing amino acid properties) and the data obtained were modelled using algorithms. As results of this second strategy were obtained several models with quite high performance values.

Objective: "Develop a learning method for predicting heterodimeric protein complexes" (Maruyama, O., 2013)

In order to predict heterodimeric complexes, heterodimeric complexes from CYC2008 and PPIs from WI-PHI (yeast database) were imported. From these data, features were obtained which, based on the use of machine learning, served as a basis for a method that was evaluated by the Naïve Bayes classifier. Thus, the log-likelihood ratio, obtained from the performance values equated by the Naive Bayes classifier, with the values of the parameters obtained by the maximum likelihood estimation, gave a score that predicted if a protein pair was a heterodimeric complex. Finally, a cross-validation with five replications was done and results were obtained with a good performance, being even superior to other models described in the literature.

As a summary of the above mentioned works, below was created this table.

Method	Objective	Results								
		ACC (%)	TNR (%)	TPR (%)	PPV (%)	NPV (%)	MCC (%)	AUC (%)	F1 (%)	Conclusion
Weighted Sparse Representation Model Combined with Discrete Cosine Transformation	Improve methods for predicting protein-protein interactions (Huang, *et al.*, 2015)	96.30 ± 0.10 Humans	-	92.63 ± 0.44 Humans	99.59 ± 0.29 Humans	-	92.82 ± 0.19 Humans	96.47 ± 0.54 Humans	-	Better prediction results were obtained
c-forest algorithm	Discover the best method to detect hotspots (Melo, *et al.*, 2016)	80 Test Group	82 Test Group	76 Test Group	70 Test Group	86 Test Group	-	78 Test Group	73 Test Group	The best algorithm to use in the proposed methods is c-forest
SBHD2 algorithm		71 Test Group	71 Test Group	70 Test Group	56 Test Group	82 Test Group	-	69 Test Group	62 Test Group	
Robetta algorithm		66 Test Group	88 Test Group	29 Test Group	60 Test Group	67 Test Group	-	62 Test Group	39 Test Group	
KFC2-A algorithm		71 Test Group	81 Test Group	53 Test Group	59 Test Group	77 Test Group	-	66 Test Group	56 Test Group	
KFC2-B algorithm		73 Test Group	96 Test Group	28 Test Group	80 Test Group	72 Test Group	-	67 Test Group	42 Test Group	
CPORT algorithm		49 Test Group	47 Test Group	54 Test Group	35 Test Group	66 Test Group	-	54 Test Group	42 Test Group	
Representation based on matrix protein sequence combined with the support vector machine	Construct a new method to better detect the sequence pairs of protein-protein interactions (You, *et al.*, 2015)	90.06 ± 0.64 Gaussian kernel function	94.37 ± 0.95 Gaussian kernel function	85.74 ± 0.94 Gaussian kernel function	93.84 ± 0.98 Gaussian kernel function	86.89 ± 0.48 Gaussian kernel function	82.03 ± 1.03 Gaussian kernel function	95.28 ± 0.64 Gaussian kernel function	89.61 ± 0.76 Gaussian kernel function	The method chosen, compared to existing ones, better distinguished the protein pairs in interaction from no interactions

Method	Objective	Results							
		ACC (%)	TNR (%)	TPR (%)	PPV (%)	MCC (%)	AUC (%)	F1 (%)	Conclusion
SVM	Combine active learning and semi-evaluated learning by support vector machine to improve the detection of protein interactions (Song, *et al.*, 2011)	-	-	51.21 BioCreative 2 PPI data set	70.23 BioCreative 2 PPI data set	-	84.3 BioInfer data set	58.33 BioCreative 2 PPI data set	This combination of methods results in a better detection of protein-protein interactions, the best method being BTDA-SVM
RS-SVM		-	-	56.54 BioCreative 2 PPI data set	71.70 BioCreative 2 PPI data set	-	84.7 BioInfer data set	62.50 BioCreative 2 PPI data set	
C-SVM		-	-	88.68 BioCreative 2 PPI data set	78.23 BioCreative 2 PPI data set	-	86.0 BioInfer data set	83.65 BioCreative 2 PPI data set	
BT-SVM		-	-	93.50 BioCreative 2 PPI data set	81.75 BioCreative 2 PPI data set	-	91.8 BioInfer data set	85.96 BioCreative 2 PPI data set	
BTDA-SVM		-	-	95.32 BioCreative 2 PPI data set	85.92 BioCreative 2 PPI data set	-	93.0 BioInfer data set	86.85 BioCreative 2 PPI data set	
k-NN	Develop methods that detect whether post-translational modifications (PTM) are within or outside the protein interaction region (Saethang, *et al.*, 2016)	89 Phosphorylation Table 6	95 Phosphorylation Table 6	85 Phosphorylation Table 6	95 Phosphorylation Table 6	79 Phosphorylation Table 6	92 Phosphorylation Table 6	-	It was possible to create the first models of detection of PTM in or out of the protein-protein interactions having obtained the obtained data with good quality values
RF		90 Phosphorylation using Information Gain	-	-	-	80 Phosphorylation using Information Gain	93 Phosphorylation using Information Gain	-	
C4.5		91 Phosphorylation Table 6	97 Phosphorylation Table 6	87 Phosphorylation Table 6	96 Phosphorylation Table 6	84 Phosphorylation Table 6	92 Phosphorylation Table 6	-	
Kstar		89 Phosphorylation using Information Gain	-	-	-	79 Phosphorylation using Information Gain	93 Phosphorylation using Information Gain	-	
MLP		91 Phosphorylation using Information Gain	-	-	-	83 Phosphorylation using Information Gain	93 Phosphorylation using Information Gain	-	
Naïve Bayes	Develop a learning method for predicting heterodimeric protein complexes (Maruyama, O., 2013)	95.5 Calculated from Table 1	98.6 Calculated from Table 1	64.4 Calculated from Table 1	81.7 Calculated from Table 1	70.2 Calculated from Table 1	97.4	72.0 Calculated from Table 1	The developed method presented better performance in the prediction of the protein complexes than the existing methods

Table 2: Previous work: Method, Objective and Results

3. APPROACH

As already mentioned in "GOAL", the approach used consisted in the discovery of new features that allowed to improve the performance of a random dataset. In this way a research was elaborated by the scientific literature that resulted in the formulation of three strategies.

3.1 Use of co-expressed gene database

A co-expression network is characterized by holding nodes that correspond to genes and edges that correspond to co-expression relationships, to the interconnected genes is assigned the same co-regulation by the action of small RNAi, metabolites, proteins and epigenetic mechanisms (Vella, *et al.*, 2017).

In order to obtain a gene co-expression network, the expression levels of the genes are evaluated from statistical tools (the result of certain alterations translates into different transcription values) in this way one can identify genes functionally regulated by the same metabolic pathway. It is usually intended to obtain the highest possible co-expression score (which is achieved if the genes are topologically and functionally homologous), except when one intends to study the effect of specific changes in the network (Vella, *et al.*, 2017).

When an interaction occurs between species, namely viruses / bacteria with a host, this results in the overexpression of several genes and in protein interactions which, because they are different species, are difficult to detect (Zhang, *et al.*, 2017)(Tugaeva, *et al.*, 2017). Since the expression of a gene normally results in a protein, and in the host / invader relationship the overexpression of genes occurs, it is intended to study in the literature which proteins are overexpressed in two different organisms during an infection and to analyse the performance in the detection of protein interactions [see Figure 9].

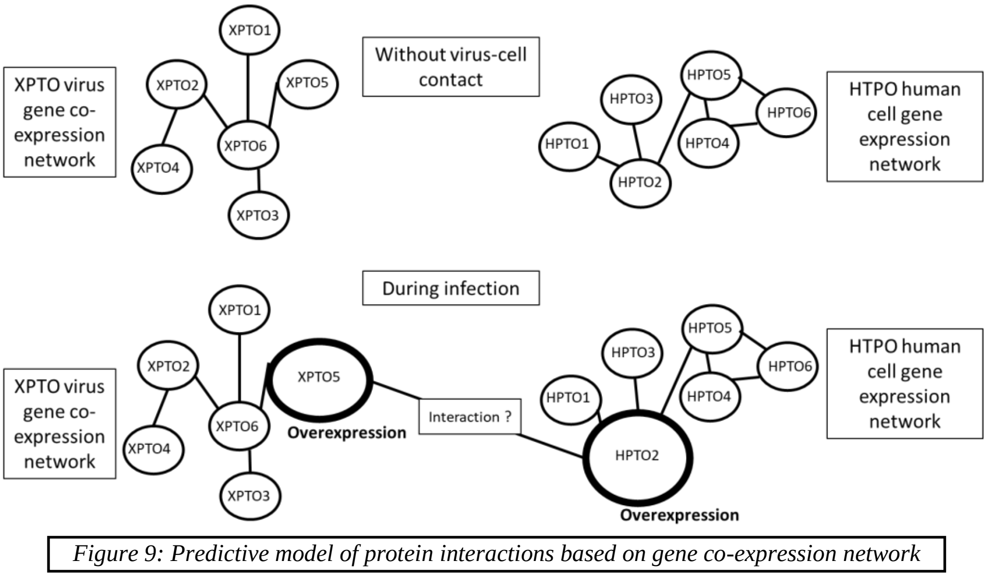

XPTO virus gene co-expression network
XPTO1
XPTO2
XPTO5
XPTO6
XPTO4
XPTO3
Without virus-cell contact
HPTO5
HPTO3
HPTO6
HPTO1
HPTO4
HPTO2
HTPO human cell gene expression network
During infection
XPTO virus gene co-expression network
XPTO1
XPTO2
XPTO5
XPTO6
XPTO4
XPTO3
Overexpression
Interaction ?
HPTO5
HPTO3
HPTO6
HPTO1
HPTO4
HPTO2
Overexpression
HTPO human cell gene expression network

Figure 9: Predictive model of protein interactions based on gene co-expression network

3.2 Use of inhibitor databases

Due to the large increase of databases containing interactions between proteins, the study of compounds that inhibit this interaction was started (Sugaya, *et al.*, 2009). These compounds can inhibit by various mechanisms: by orthostatic inhibition the inhibitor interacts with the ligands and obstructs the responsible sites (it does not bind them), by allosteric regulation the inhibitor binds to the molecule outside the interface and triggers a conformational change that prevents the interaction with ligands, finally, by the binding to the interface the inhibitor binds to the interaction sites, preventing its contact with ligands (Gurung, *et al.*, 2017).

With the progressive study of potential inhibitors of protein interactions, there are today several, described as susceptible to being inhibited by "small molecules" from inhibition by interfacial bonding, which are characterized by being joined with great affinity to the protein hot spots responsible for interactions (Sugaya, *et al.*, 2009).

However, this affinity is not uniform to all protein interactions, in the case of domain-domain interactions (between two globular proteins) small molecule inhibition occurs if there is a main segment responsible for the interconnection, in the case of peptide-domain interactions (mediated by a peptide), all of them have the possibility of being inhibited by small molecules because the interaction falls on a single continuous binding epitope (London, *et al.*, 2013). It should be noted that of the four classes of narrow, wide, tight or loose interactions, those that are more likely to be inhibited are narrow and / or tight to the detriment of others (London, *et al.*, 2013).

Thus a potential way to detect new interactions between proteins is to access databases of protein inhibitors and find inhibitors that are common to two different proteins. Subsequently it is enough to test the possibility of the proteins with the inhibitor in common to have interaction between them [see Figure 10].

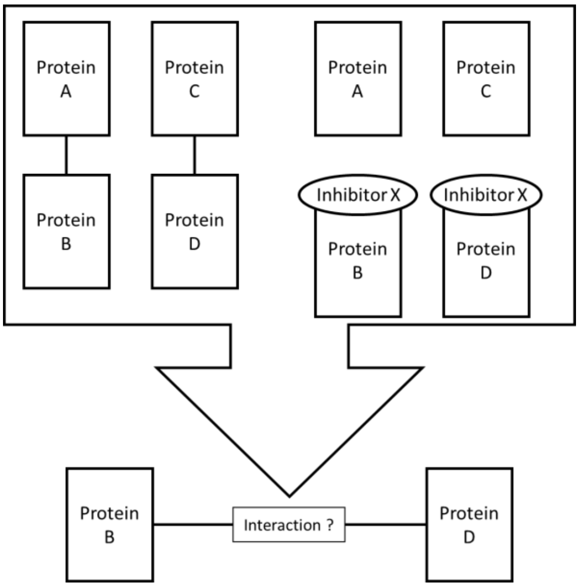


Figure 10: Predictive model of protein interactions based on inhibitors of interactions

3.3 Use of peptide recognition modules

Peptide recognition modules (SH3, SH2, PDZ, etc ...) exist mainly in complexes and their bonds are usually responsible for restructuring the protein complexes and can be reused for various cellular functions (Reimand, *et al.*, 2012). They are widely used to detect protein interactions because they are common in the eukaryotic genome and there are several well-studied modules (they are easy to detect constituting a large database) (Reimand, *et al.*, 2012)(Jain, *et al.*, 2016). Protein interactions are generally considered very likely if a binding site (activation) is accessible and the proteins are expressed at the same time and site in the cell (Reimand, *et al.*, 2012).

The **SH3 domains** are composed of about 60 amino acids distributed over 5 beta chains linked in two perpendicular beta sheets interrupted by a 3-10 helix, mainly binding to proline rich regions, also having affinity for regions rich in arginine and lysine , are involved in cell signalling, regulation of the actin cytoskeleton and endocytosis (Jain, *et al.*, 2016).

The **SH2 domains** are composed of about 100 amino acids with a central sheet having on each face a short helix, bind primarily to phosphotyrosine peptides, but also to N-terminal arginine and C-terminal histidine, are associated to the intracellular signal transduction, to the enzymatic activity and to the activation of T-cell (Tong, *et al.*, 1996).

The **PDZ domains** are composed of about 85 amino acids, which are in a compact structure with 5 to 6 beta chains and two alpha helices, bind primarily to the C-terminal of the proteins and are involved in channel regulation, signal detection, stabilization of cellular polarity complexes and in neural development (Reimand, *et al.*, 2012).

The **WW domains** are composed of about 35 amino acids with a core anti-parallel beta-sheet triple chain and two tryptophan residues conserved in signature separated by 20 residues, bind to regions that are rich in proline and are involved in growth control and ubiquitin-mediated proteolysis (Reimand, *et al.*, 2012).

The **LRR domains** are composed of approximately 25 amino acids in the form of a horseshoe with the concave having face parallel chains surrounded by loops and the convex face helical structures, due to having a large amount of leptin (hydrophobic) interact with hydrophobic residues, are associated with apoptosis, autophagy, processes related to ubiquitin, nuclear mRNA transport and neuronal development (Ng, *et al.*, 2010).

In this way, was intended to analyse the possibility of the structure of pairs of proteins holding interactions mediated by peptide recognition modules predict, based on structural similarity, interactions between pairs of proteins that do not have peptide recognition modules. If this possibility is validated, the features of the peptide recognition modules can then be used to predict interactions in any dataset [see Figure 11].

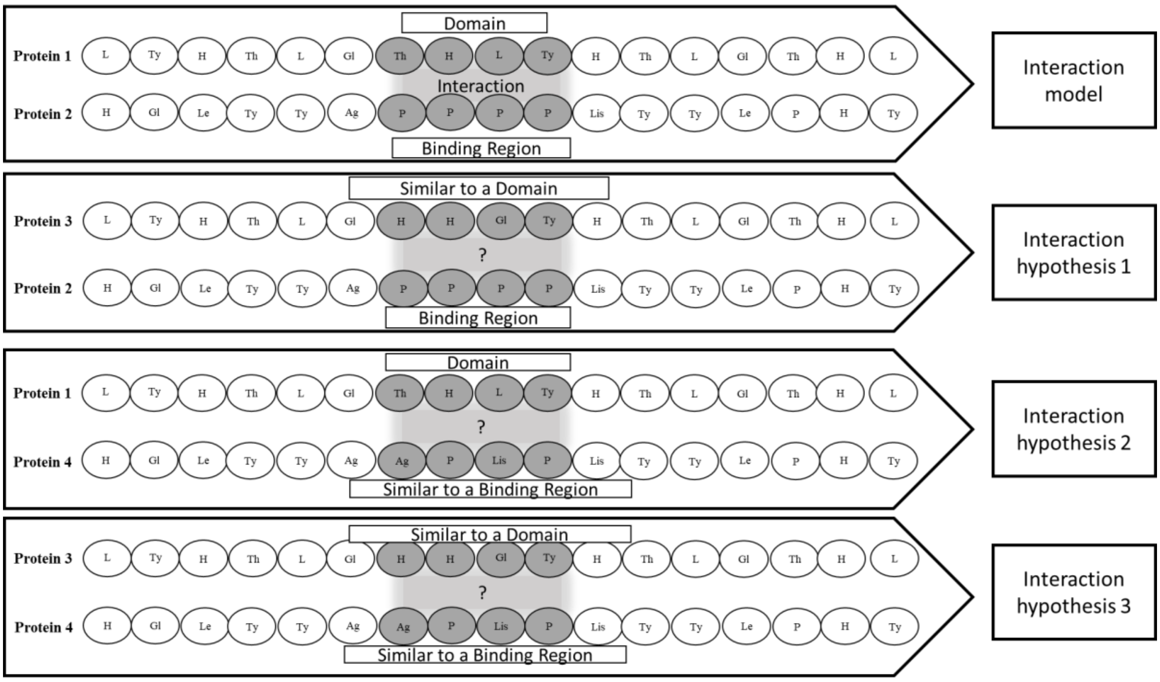

Domain
Protein 1
L Ty H Th L Gl Th H L Ty H Th L Gl Th H L
Interaction
Protein 2
H Gl Le Ty Ty Ag P P P P Lis Ty Ty Le P H Ty
Binding Region
Interaction model
Similar to a Domain
Protein 3
L Ty H Th L Gl H H Gl Ty H Th L Gl Th H L
?
Protein 2
H Gl Le Ty Ty Ag P P P P Lis Ty Ty Le P H Ty
Binding Region
Interaction hypothesis 1
Domain
Protein 1
L Ty H Th L Gl Th H L Ty H Th L Gl Th H L
?
Protein 4
H Gl Le Ty Ty Ag Ag P Lis P Lis Ty Ty Le P H Ty
Similar to a Binding Region
Interaction hypothesis 2
Similar to a Domain
Protein 3
L Ty H Th L Gl H H Gl Ty H Th L Gl Th H L
?
Protein 4
H Gl Le Ty Ty Ag Ag P Lis P Lis Ty Ty Le P H Ty
Similar to a Binding Region
Interaction hypothesis 3

Figure 11: Predictive model of protein interactions based on peptide recognition modules

4. EXPERIMENTAL ANALYSIS

4.1 Preliminary work

In order to accomplish the proposed work, a learning process of Python and Machine Learning was necessary.

In the consolidation of this learning, the task was to detect catalytic proteins in a dataset of human proteins. For the success of this same task, 100 human catalytic proteins were imported from Uniprot, the data were processed in order to obtain only the primary structure and to obtain sequence features, an amino acid count was made. Subsequently, 100 human proteins were imported from Uniprot and the same treatment was carried out. Then a vector was created in which 1 corresponded to the existence of catalytic proteins and 0 to the non-existence of catalytic proteins. Finally, the "Support Vector Machine" and "Naive Bayes" classifiers were used and their performances evaluated [see Figures 12 and 13].

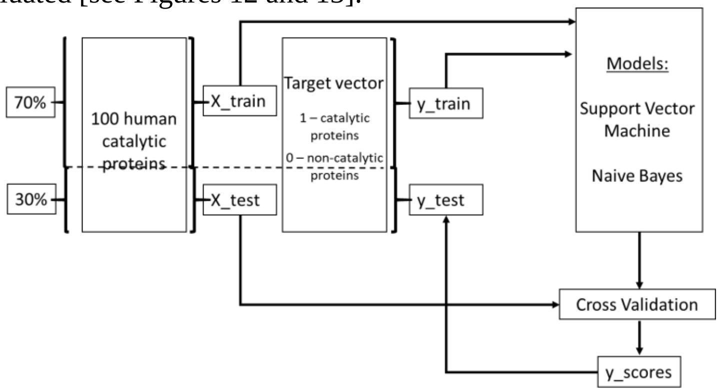
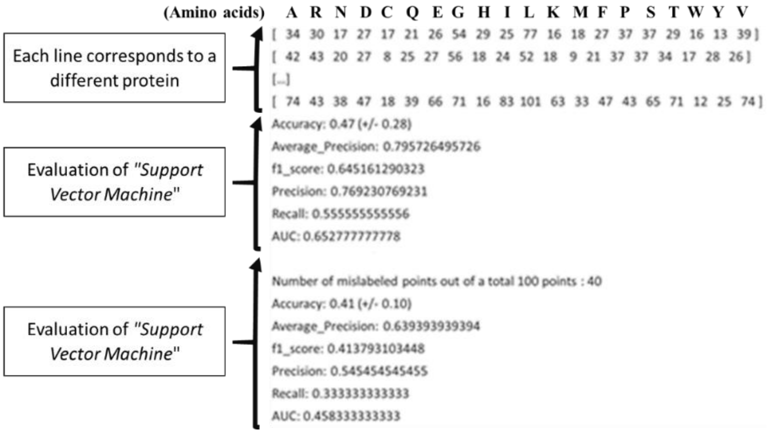


Figure 12: Operation of the developed model


Each line corresponds to a different protein
Evaluation of *"Support Vector Machine"*
Evaluation of *"Support Vector Machine"*

```
(Amino acids)  A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V
[ 34  30  17  27  17  21  26  54  29  25  77  16  18  27  37  37  29  16  13  39]
[ 42  43  20  27   8  25  27  56  18  24  52  18   9  21  37  37  34  17  28  26]
[...]
[ 74  43  38  47  18  39  66  71  16  83 101  63  33  47  43  65  71  12  25  74]
Accuracy: 0.47 (+/- 0.28)
Average_Precision: 0.795726495726
f1_score: 0.645161290323
Precision: 0.769230769231
Recall: 0.555555555556
AUC: 0.652777777778

Number of mislabeled points out of a total 100 points : 40
Accuracy: 0.41 (+/- 0.10)
Average_Precision: 0.639393939394
f1_score: 0.413793103448
Precision: 0.545454545455
Recall: 0.333333333333
AUC: 0.458333333333
```

Figure 13: Results of the developed model

Later, proceeded to the analysis and understanding of the commands written in Python of the model of Machine Learning present in the article *“A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions”* where the process of collection of the used features is detailed.

From the previous approaches, we opted for the use of peptide recognition modules due to the computational simplicity and because in literature there are more information addressing this theme. Using this approach, three strategies were elaborated in the course of this thesis, two of which were set aside because of the incorrect methodology used or the inconclusive values. In this way, this section is divided in three parts each one with an executed strategy that are listed in chronological order, being the last one the strategy to which this thesis refers.

4.2 Relevance of peptide recognition modules by descriptors PyDPI

4.2.1 Procedure

1 - In order to obtain datasets with protein IDs that interact with the peptide recognition modules on the Uniprot site searched for the words "homo sapiens". In the definition "View" had clicked in "Gene Ontology" → "molecular function" → "binding" → "protein binding" → "protein domain specific binding" → results of " SH3 / SH2 / PDZ / WW / LRR domain binding" (these five domains were chosen because they were the ones that had the most results SH3=15763 results, PDZ = 567 results, WW = 39 results, SH2 = 38 results and LRR 19 results) → finally in the side definition "Popular organisms" clicked on "Human".

2 – These 5 datasets (one for each domain) were downloaded in the "uncompressed" option for Excel sheets.

2.1 – All information has been removed until there is only one protein ID in each cell of column A and in front of it in the cell of column B the IDs of the proteins with which there is interaction.

2.2 – In column B, the lines with blank cells and the word "Itself" were removed, cells that had more than one ID were separated into different columns on the same line.

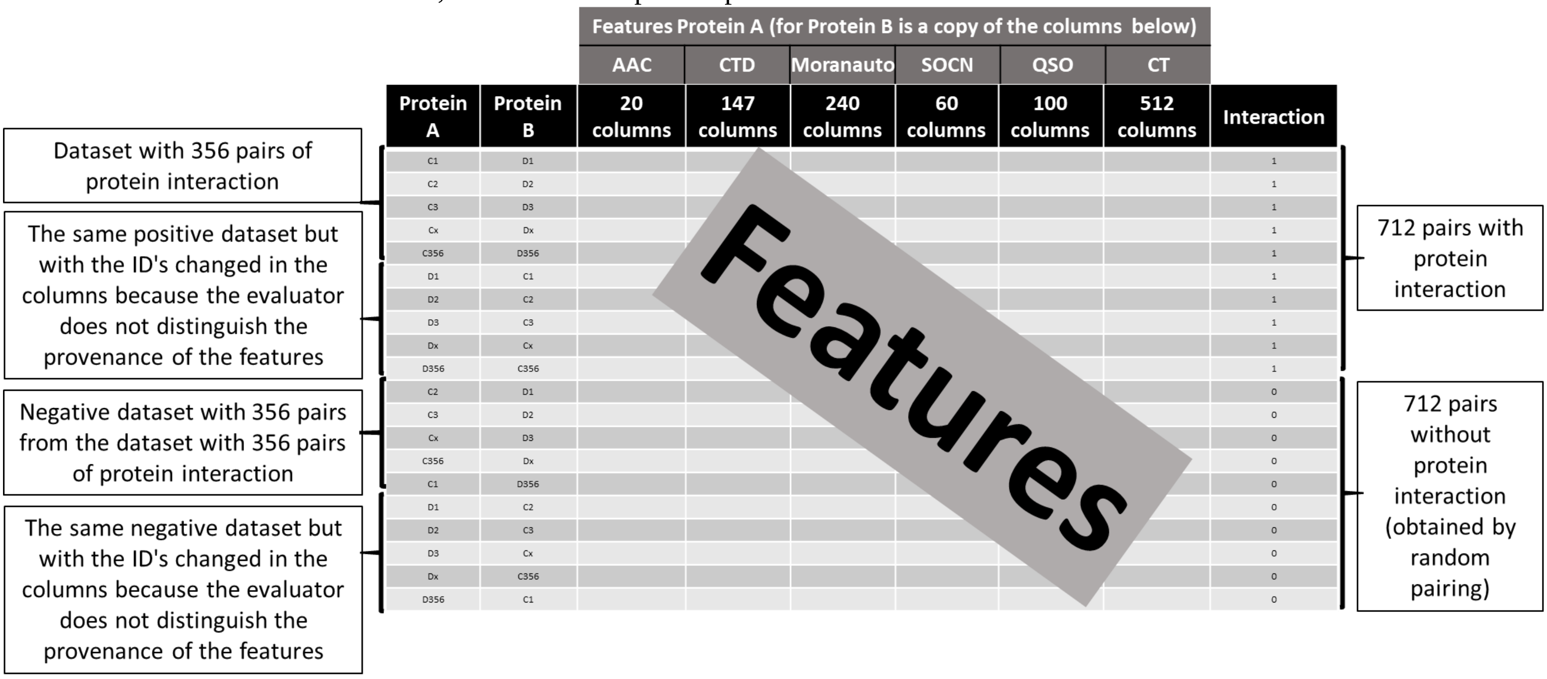
2.3 – Finally the ID's of column A and the ID's of column B were organized and it is sometimes necessary to repeat the same ID several times. In order to avoid overfitting problems (excessive repetition of features that damage classification), the smallest possible number of IDs in column A was repeated, obtaining 144 pairs of SH3, SH2 and PDZ, 110 of WW and 80 of LRR.

3 – In terms of interaction pairs with generic proteins (it is considered that due to the randomness of their choice their interaction is not dependent on peptide recognition modules), protein pairs were imported from the Uniprot by searching for the name "Homo sapiens", having thus been created two types of datasets, one with generic pairs and other with generic pairs and pairs of

peptide recognition module. The protein pairs that were used as generic were 356 pairs for the datasets with SH3, SH2 and PDZ, 390 pairs for the dataset with WW and 420 for the dataset with LRR (in order to account for each dataset a total of 500 protein pairs). In this way, 5 datasets were created with the pairs of each domain and with generic pairs (each having 500 protein pairs) and 3 different datasets with only generic pairs (having 356, 390 and 420 protein pairs).

4 – Subsequently the datasets were augmented and subdivided into four parts.

4.1 - In the case of datasets with interactions only between generic protein pairs, the first part corresponded to the pairs taken from the Uniprot, in the second part a copy of the previous one was made but changing the order of the pairs as the ID's that were in the column A from the first part of the dataset, are passed to column B and vice versa (solution to the fact that the performance evaluator does not distinguish which features will correspond to the IDs of column A or B) , in the third part was created the negative dataset (protein pairs where no interactions occur) for this columns from the first part were copied, however the first ID coming from column A was placed at the end of the column, being at the beginning of the column A of the third part, the second ID of the column A of the first part (it was considered that by changing the order of the protein pairs in this way, the probability of them interacting with each other was low), in the fourth part a copy of the previous one was made but changing the order of the pairs (ID's of column A went to column B and vice versa). In the end the three datasets with only generic pairs were left with 1424, 1560 and 1680 protein pairs.



Dataset with 356 pairs of protein interaction

The same positive dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

Negative dataset with 356 pairs from the dataset with 356 pairs of protein interaction

The same negative dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

		Features Protein A (for Protein B is a copy of the columns below)						
		AAC	CTD	Moranauto	SOCN	QSO	CT	
Protein A	Protein B	20 columns	147 columns	240 columns	60 columns	100 columns	512 columns	Interaction
C1	D1							1
C2	D2							1
C3	D3							1
Cx	Dx							1
C356	D356							1
D1	C1							1
D2	C2							1
D3	C3							1
Dx	Cx							1
D356	C356							1
C2	D1							0
C3	D2							0
Cx	D3							0
C356	Dx							0
C1	D356							0
D1	C2							0
D2	C3							0
D3	Cx							0
Dx	C356							0
D356	C1							0

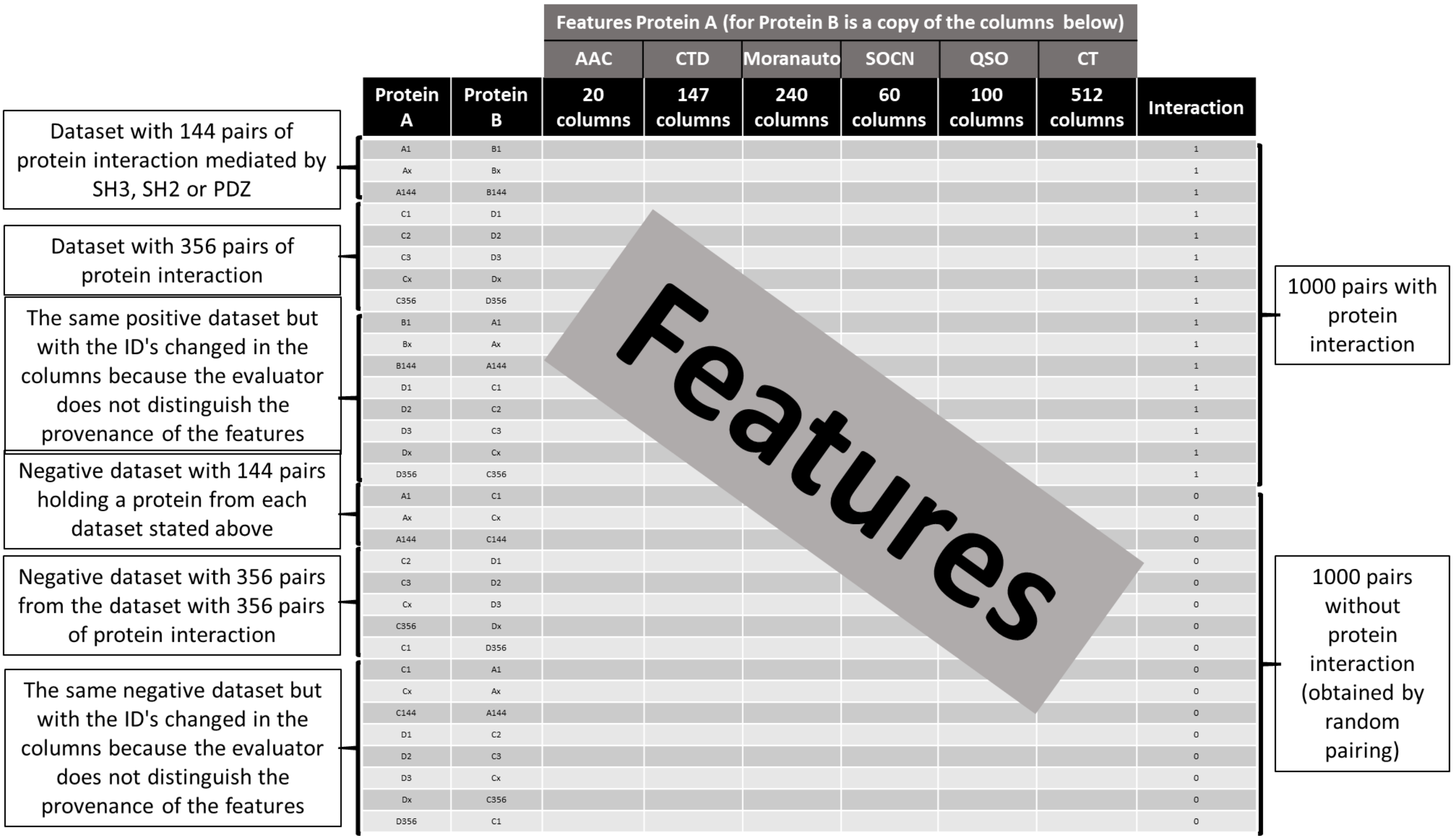
712 pairs with protein interaction

712 pairs without protein interaction (obtained by random pairing)

Table 3: Generic dataset organization with 1424 protein pairs

The only difference for datasets with 1560 and 1680 pairs of proteins is that in the case of the dataset with 1560 pairs, instead of 356 pairs imported are 390 pairs and instead of 712 pairs of interaction or non-pairs, they are 780 pairs. In the case of the dataset with 1680 pairs, instead of 356 pairs imported are 420 pairs and instead of 712 pairs of interaction or non-interaction, they are 840 pairs.

4.2 – The datasets with pairs of proteins containing the enunciated domains plus the generic pairs had the same procedure, the only change being the addition, to the previous datasets, of the pairs with the corresponding domains (pairs with SH3, SH2 and PDZ were added to the dataset with 1424 pairs, pairs with WW were added to the dataset with 1560 and pairs with LRR were added to the dataset with 1680), the only difference was that in the third part in column A the order of the proteins ID's with domains of the first part maintained and in column B generic ID's were imported from column A of the first part. In the end the five datasets with pairs of proteins having domains and generic pairs had 2000 protein pairs.


Features
Dataset with 144 pairs of protein interaction mediated by SH3, SH2 or PDZ
Dataset with 356 pairs of protein interaction
The same positive dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features
Negative dataset with 144 pairs holding a protein from each dataset stated above
Negative dataset with 356 pairs from the dataset with 356 pairs of protein interaction
The same negative dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features
1000 pairs with protein interaction
1000 pairs without protein interaction (obtained by random pairing)

		Features Protein A (for Protein B is a copy of the columns below)						
		AAC	CTD	Moranauto	SOCN	QSO	CT	
Protein A	Protein B	20 columns	147 columns	240 columns	60 columns	100 columns	512 columns	Interaction
A1	B1							1
Ax	Bx							1
A144	B144							1
C1	D1							1
C2	D2							1
C3	D3							1
Cx	Dx							1
C356	D356							1
B1	A1							1
Bx	Ax							1
B144	A144							1
D1	C1							1
D2	C2							1
D3	C3							1
Dx	Cx							1
D356	C356							1
A1	C1							0
Ax	Cx							0
A144	C144							0
C2	D1							0
C3	D2							0
Cx	D3							0
C356	Dx							0
C1	D356							0
C1	A1							0
Cx	Ax							0
C144	A144							0
D1	C2							0
D2	C3							0
D3	Cx							0
Dx	C356							0
D356	C1							0

Table 4: Organization of datasets with SH3, SH2 or PDZ and 1424 generic pairs

The only difference for the datasets with WW and LRR is that in the case of the dataset with WW pairs, instead of 144 pairs imported (mediated by WW) are 110 pairs and instead of 356 pairs imported, they are 390 pairs. In the case of the dataset with LRR pairs, instead of 144 imported pairs (mediated by LRR) are 80 pairs and instead of 356 pairs imported, there are 420 pairs.

5 – Then using the PyDPI 1.0 software and the Python language [Annex I – Page 36], six different descriptors (structures that contain descriptive information of the different data) were used to collect features, thus creating six equal groups with the eight previous datasets (in all worked with 48 different datasets). Each of the descriptors (AAC, CTD, Moranauto, SOCN, QSO and CT) has a characteristic functioning and gives rise to different numbers of features.

5.1 - The AAC descriptors also called amino acid composition obtain features using the fraction between the number of the amino acid type and the length of the sequence having 20 different features (Cao, *et al.*, 2012).

5.2 – The CTD descriptors also called composition, transition and distribution starts with each amino acid acquiring an index (1,2 or 3) in result of their hydrophobicity, normalized van der Waals volume, polarity, polarizability, charge, secondary structure or solvent accessibility (Cao, *et al.*, 2012). The composition is calculate using the fraction between the percentage of each class in the sequence and the length of the sequence (Cao, *et al.*, 2012). The transition is obtain by the fraction between the frequency in each an index is follow by other one and the length of sequence (Cao, *et al.*, 2012). The distribution show the spreading of each index in the sequence, is determined by the fraction between the position of an index (the first time an index appear or when the same index reaches 25%, 50%, 75% or 100% of its distribution) and the length of the sequence, all multiplied by 25%, 50%, 75% or 100% (taking care its distribution) (Cao, *et al.*, 2012). These descriptors have 147 different features.

5.3 – The Moranauto descriptors also called Moran autocorrelation use the distribution of amino acid properties along the sequence in the form of indices, which are first centralized and standardized, and then calculated by the Moran´s I formula (Cao, *et al.*, 2012). These descriptors have 240 different features.

5.4 – The SOCN descriptors also called sequence-order-coupling numbers use the squared sum of the distance between 2 of the 20 amino acids at different positions, having 60 different features (Cao, *et al.*, 2012).

5.5 – The QSO descriptors also called quasi-sequence-order QSO is obtain by the fraction between the normalized occurrence of an amino acid and the sum of the normalized occurrence of the same amino acid with the multiplication of the weighting factor with the different distances between this amino acid and other (Cao, *et al.*, 2012). These descriptors have 100 different features.

5.6 – The CT descriptors also called conjoint triad to be calculated, at first the amino acids are cluster in seven classes (the dipoles and the volume of its side chains are use to cluster them) (Cao, *et al.*, 2012). Then it´s used a binary space (V,F) where the vector V represent the features (three neighbour amino acids in the form of classes) and the vector F the frequency of each feature (Cao, *et al.*, 2012). Finally, to normalized the value of frequencies it´s done a fraction between the subtraction of the dimension of F with the minimum frequency of a feature and the maximum frequency of the same feature. These descriptors have 512 different features (Cao, *et al.*, 2012).

6 – Subsequently, the performances of the 48 datasets were evaluated using a Machine Learning tool called "Orange". For this, we used the classifiers naïve Bayes, k-nearest neighbour (k=5), support vector machine, tree decision, logistic regression and stochastic gradient descent, whose

performance was evaluated by the following indicators: area under the ROC curve, accuracy, F-score, precision and recall.

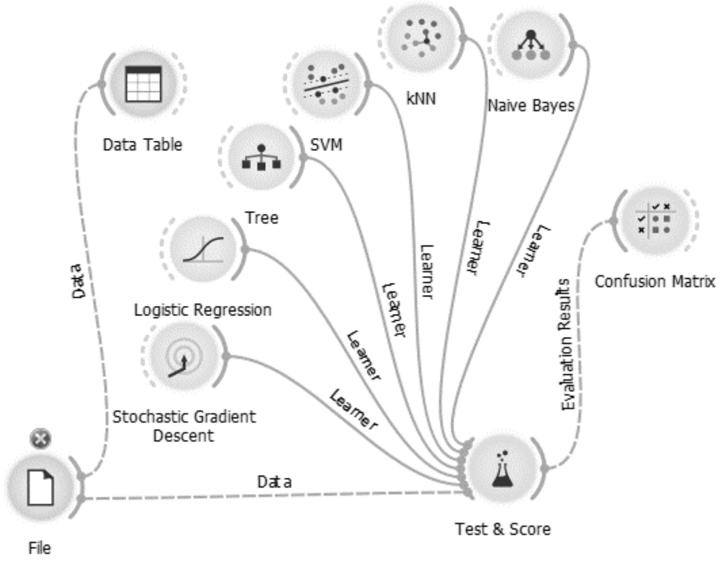


Figure 14: Model structure using the Orange tool

6.1 – All of these values were subsequently treated and subtractions were made for each quality indicator value, within the same descriptor but between datasets with generic / domains and datasets with only corresponding generic ones. Subsequently we added the value of these differences by classifier (analysis of the relevance of the classifiers by domain), the result of these sums was also added to each other by domain (analysis of domain pertinence by descriptor) and finally the results of these sums were also summed between them (analysis of the pertinence of the descriptor) [Annex II – Page 43].

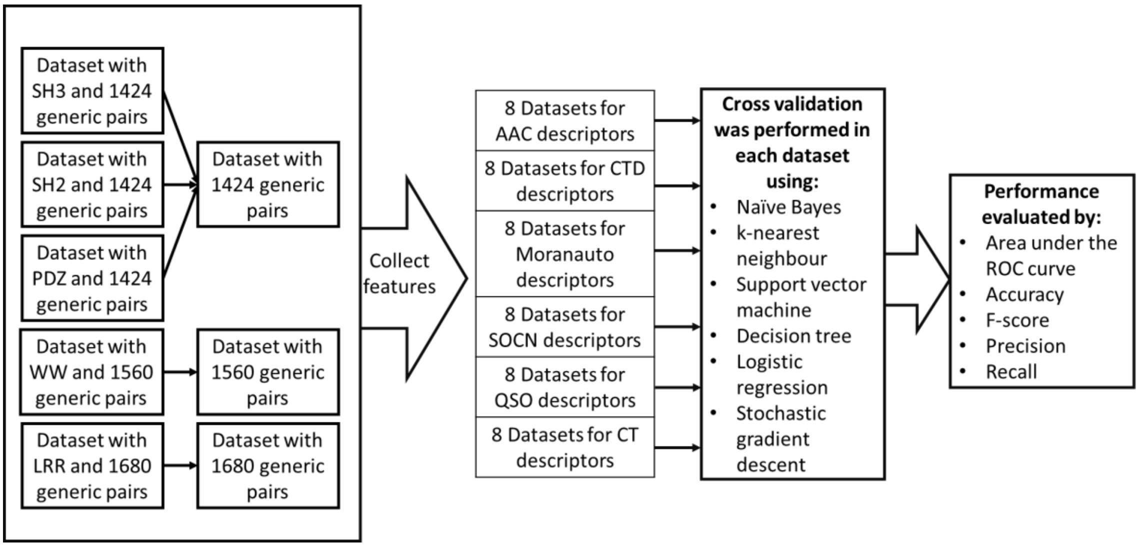

Dataset with SH3 and 1424 generic pairs
Dataset with SH2 and 1424 generic pairs
Dataset with PDZ and 1424 generic pairs
Dataset with 1424 generic pairs
Dataset with WW and 1560 generic pairs
Dataset with 1560 generic pairs
Dataset with LRR and 1680 generic pairs
Dataset with 1680 generic pairs
Collect features
8 Datasets for AAC descriptors
8 Datasets for CTD descriptors
8 Datasets for Moranauto descriptors
8 Datasets for SOCN descriptors
8 Datasets for QSO descriptors
8 Datasets for CT descriptors
Cross validation was performed in each dataset using:
• Naïve Bayes
• k-nearest neighbour
• Support vector machine
• Decision tree
• Logistic regression
• Stochastic gradient descent
Performance evaluated by:
• Area under the ROC curve
• Accuracy
• F-score
• Precision
• Recall

Figure 15: Summary of the 1st strategy model

4.2.2 Values

As can be seen in "Annex II" in the page 43, the values of the quality indicators are generally in the range of values below 0.6. Overall performance improved from datasets that have domains to the non-domain (shaded black are the values whose subtractions indicate otherwise). The k-nearest neighbour classifier had the worst results because detected more times a better performance in the domain-free datasets than in those who have a domain. In contrast, the naïve Bayes classifier had the best difference values. As for the quality indicators, the recall was the one that detected fewer improvements of the datasets with domains for those that do not have them whereas the precision was the one that had results that are more positive. In terms of the SH2 domains and to a lesser extent the SH3 had the best values added of difference and the datasets with LRR had the worst values added. Finally, the QSO was the descriptor that had the highest value of sum of differences in the performance evaluation and the SOCN the one that had the lowest value.

4.2.3 Discussion

The use of the "orange canvas" software although simple and easy to handle implies some disadvantages, namely the fact that it can not run very large datasets and have a high cost in terms of time. As for the PyDPI software is a seemingly viable source of features that, depending on the chosen descriptors, makes a quick collection of features (CT and Moranauto take more time). In addition, it uses as features several biochemical components present in proteins, which is why we follow the methodology of the most recent strategies, which use biochemical knowledge to improve the performance of computational methods. However, the relevance of the subtractions made in the treatment part of the results is questionable, since what is important is to analyse the performance values obtained. The mathematical operations performed with the obtained results must be observed in the perspective of which descriptors that appear to have more discrepant results (between the datasets with domain and the ones without domain) and can not be observed in the perspective of which descriptor that worked best, because this is only detected if the performance values are high. By analysing the values coming from the quality indicators it is verified that the values are inconclusive since all are less than 0.6 (would be viable if they were higher than 0.7). These poor values may be the result of the datasets being small (just 500 different features) and being divided by the descriptors. Moreover, the part of the negative interactions of the datasets with respect to the proteins with domains (it is at the beginning of the third and fourth part of the datasets) did not have in column B ID's of proteins with domains, but of generic proteins. For all these reasons this strategy was considered obsolete.

4.3 Relevance of the domains interaction in a large dataset

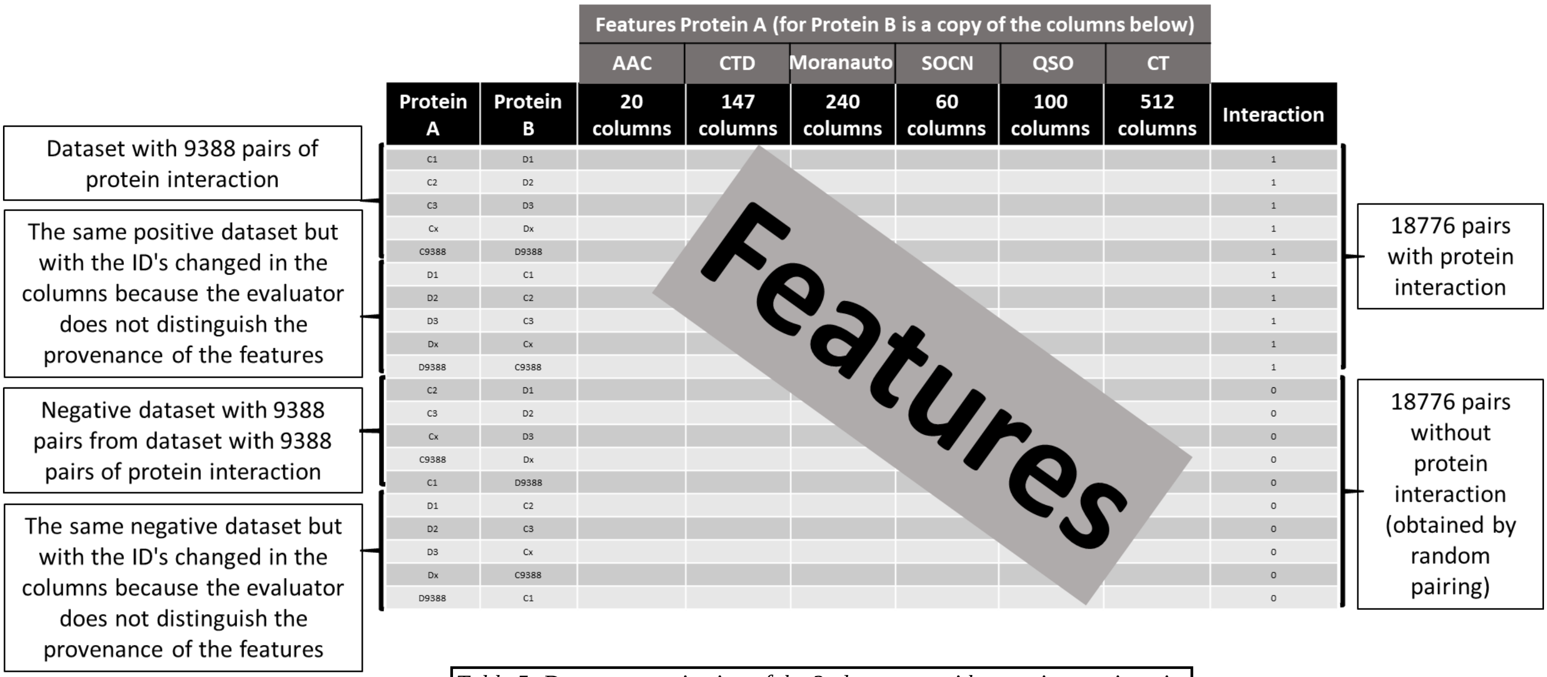
In order to solve the shortcomings of the previous strategy, only two datasets were created, one with pairs of generic proteins (more protein pairs were imported) and another with all protein pairs whose interaction is mediated by domains plus generic pairs.

4.3.1 Procedure

1 - In order to obtain a dataset with protein IDs that interact with the peptide recognition modules, were used those of the previous strategy, which is the 144 pairs of SH3, SH2 and PDZ, 110 of WW and 80 of LRR with the difference that all them had gone to the same dataset.

2 – In terms of interaction pairs with generic proteins, protein pairs were imported from the Uniprot by searching for the name "Homo sapiens", having thus been created two datasets, one with generic pairs and other with generic pairs and pairs of all peptide recognition module. In this way the generic dataset remained with 9388 protein pairs and the dataset with generic and domain-mediated pairs with 10010 (there are 622 domain mediated interaction pairs).

3 – Subsequently, the datasets were augmented and subdivided into four parts like in the first strategy. However, because they are very large datasets and so that the different parts are detected, in the second part, the characters are blue, the third is yellow and the fourth is red. Also in the third part was altered the way to generate negative data in the region that contains only non-protein interactions mediated by domains, it was therefore maintained the order of the protein ID's with domains coming from column A of the first part regarding the ID's of proteins with domains from column B of the first part, the first ID passed to the end of the 622 ID's and the second ID



		Features Protein A (for Protein B is a copy of the columns below)						
		AAC	CTD	Moranauto	SOCN	QSO	CT	
Protein A	Protein B	20 columns	147 columns	240 columns	60 columns	100 columns	512 columns	Interaction
C1	D1							1
C2	D2							1
C3	D3							1
Cx	Dx							1
C9388	D9388							1
D1	C1							1
D2	C2							1
D3	C3							1
Dx	Cx							1
D9388	C9388							1
C2	D1							0
C3	D2							0
Cx	D3							0
C9388	Dx							0
C1	D9388							0
D1	C2							0
D2	C3							0
D3	Cx							0
Dx	C9388							0
D9388	C1							0

Features

Dataset with 9388 pairs of protein interaction

The same positive dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

Negative dataset with 9388 pairs from dataset with 9388 pairs of protein interaction

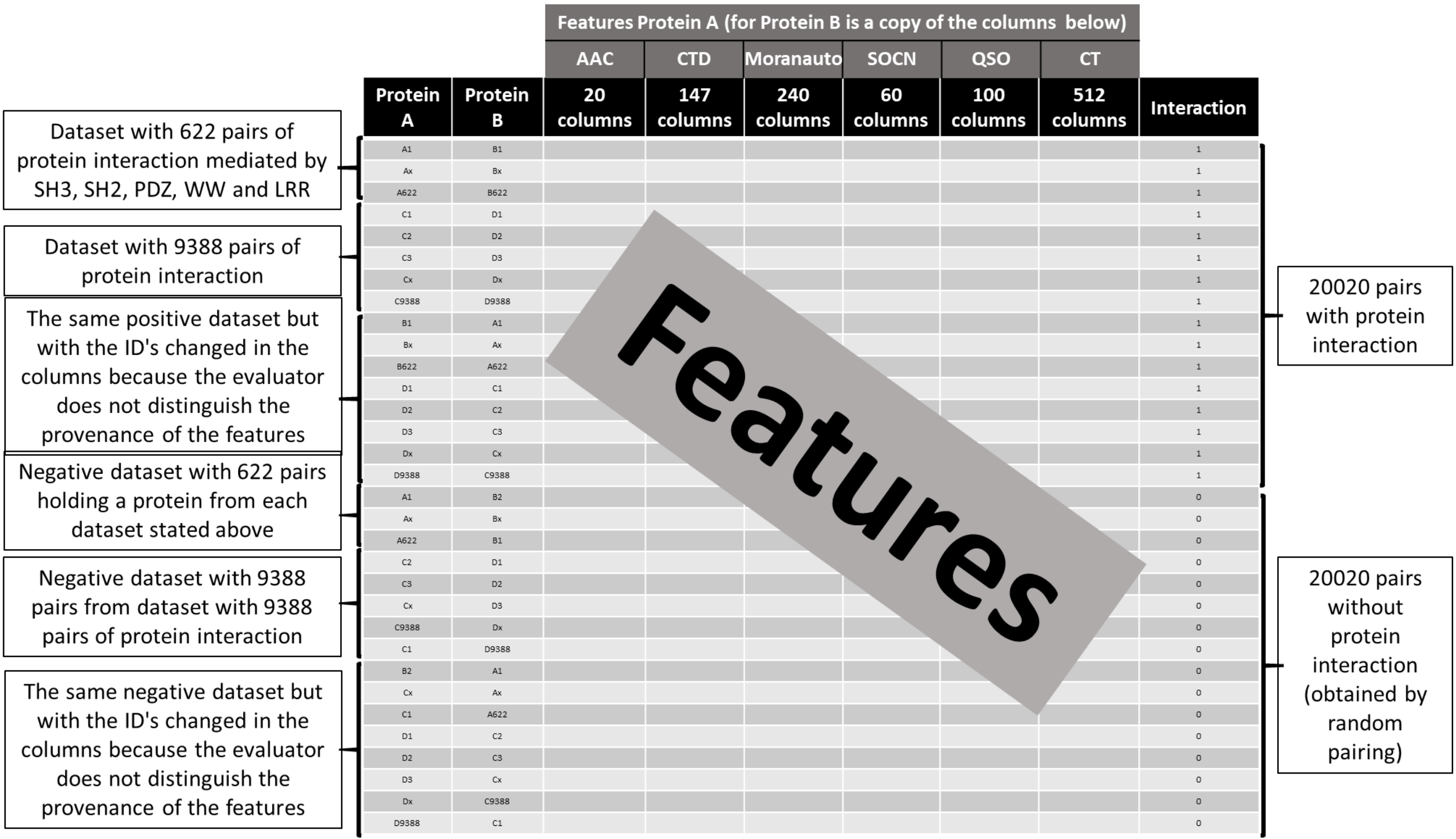
The same negative dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

18776 pairs with protein interaction

18776 pairs without protein interaction (obtained by random pairing)

Table 5: Dataset organization of the 2nd strategy with generic protein pairs

of the column B of the first part became the first ID of the column B of the third part. At the end the dataset with pairs of generic proteins and pairs of proteins with domains remained with 40040 protein pairs and the dataset with only pairs of generic proteins remained with 37552 protein pairs.



Dataset with 622 pairs of protein interaction mediated by SH3, SH2, PDZ, WW and LRR

Dataset with 9388 pairs of protein interaction

The same positive dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

Negative dataset with 622 pairs holding a protein from each dataset stated above

Negative dataset with 9388 pairs from dataset with 9388 pairs of protein interaction

The same negative dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

20020 pairs with protein interaction

20020 pairs without protein interaction (obtained by random pairing)

		Features Protein A (for Protein B is a copy of the columns below)						
		AAC	CTD	Moranauto	SOCN	QSO	CT	
Protein A	Protein B	20 columns	147 columns	240 columns	60 columns	100 columns	512 columns	Interaction
A1	B1							1
Ax	Bx							1
A622	B622							1
C1	D1							1
C2	D2							1
C3	D3							1
Cx	Dx							1
C9388	D9388							1
B1	A1							1
Bx	Ax							1
B622	A622							1
D1	C1							1
D2	C2							1
D3	C3							1
Dx	Cx							1
D9388	C9388							1
A1	B2							0
Ax	Bx							0
A622	B1							0
C2	D1							0
C3	D2							0
Cx	D3							0
C9388	Dx							0
C1	D9388							0
B2	A1							0
Cx	Ax							0
C1	A622							0
D1	C2							0
D2	C3							0
D3	Cx							0
Dx	C9388							0
D9388	C1							0

Table 6: Organization of 2nd strategy datasets with domains and generic pairs

4 – Then using the PyDPI 1.0 software and Python language [Annex I – Page 36] from the same six descriptors were obtained the features of the generic proteins (the features of the pairs with domains were copied from the first strategy).

5 – Subsequently, the performances of the two datasets were evaluated using Python programming (the dataset was too large to be evaluated in a useful time by the "Orange" tool) [Annex III –Page 49]. For this, the classifiers used were support vector machine, naïve Bayes, decisions tree, neuro network models, k-nearest neighbours (k=6) and stochastic gradient descent, whose performance was evaluated by the following indicators:, accuracy, F-score, precision, recall and area under the ROC curve.

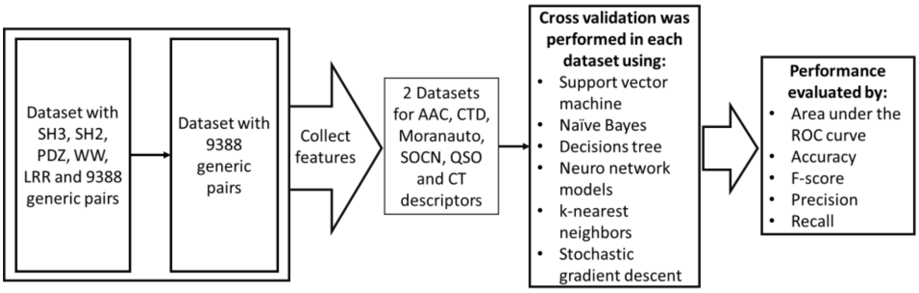

Dataset with SH3, SH2, PDZ, WW, LRR and 9388 generic pairs
Dataset with 9388 generic pairs
Collect features
2 Datasets for AAC, CTD, Moranauto, SOCN, QSO and CT descriptors
Cross validation was performed in each dataset using:
• Support vector machine
• Naïve Bayes
• Decisions tree
• Neuro network models
• k-nearest neighbors
• Stochastic gradient descent
Performance evaluated by:
• Area under the ROC curve
• Accuracy
• F-score
• Precision
• Recall

Figure 16: Summary of the 2nd strategy model

4.3.2 Values

Classifiers	Datasets	Accuracy	F-score	Precision	Recall	Area under the ROC curve
SVM	Domains + Generic	0.5495984583343922	0.24377737421215684	0.8464391691394659	0.14239361038312742	0.5582641549354098
	Generic	0.5413887035491264	0.23823663773412518	0.8123052959501558	0.13958779443254818	0.5538315180932114
Naïve Bayes	Domains + Generic	0.49718674654982953	0.634234396859064	0.49674819737028136	0.8769499563209784	0.4937042671771079
	Generic	0.4957858698985277	0.631297672836807	0.4955380977804897	0.8695128479657388	0.4966851562652909
Decisions Tree	Domains + Generic	0.505238061211202	0.5161727558375948	0.509734728644439	0.5227754898290279	0.5096696392041553
	Generic	0.5021053136981447	0.5159511618747539	0.5063128059778408	0.5259635974304069	0.5091733472646802
NNM	Domains + Generic	0.49943130484859277	0.6223141661573182	0.4992456246228123	0.8259078996630476	0.4982344696365969
	Generic	0.4917022055438893	0.43658769163060224	0.48303846778120435	0.39828693790149894	0.48818837556089645
KNN (k=6)	Domains + Generic	0.5008217768506313	0.42796822443283483	0.5016778523489933	0.3731436415824286	0.5010164040724838
	Generic	0.500799115276487	0.42959537572254336	0.5064510267127021	0.37299250535331907	0.5066048763354223
SGD	Domains + Generic	0.5012157226154965	0.5097388632872504	0.5020576131687243	0.5176588044427805	0.5018007879517413
	Generic	0.5003998254775998	0.6387043592105901	0.5007984183712265	0.8814239828693791	0.5058861866923395

Table 7: Results of the 2nd strategy

In the table above, in black are the quality values whose dataset performance with generic proteins was higher than the values of the dataset with generic proteins and proteins with domains. Overall, all values are around 0.5. The classifier that had the best performance comparing the dataset with generic/domains protein pairs with the dataset with only generic protein pairs was the neuro network models and the worst was stochastic gradient descent. As for the quality indicators, what gave the best results in terms of demonstrating that the generic dataset / domains had a better performance was the accuracy and what had worse results was the area under the ROC curve.

4.3.3 Discussion

The two datasets constructed, despite having a considerable size, may not have contained enough features to give relevant values. Although each dataset has 10000 different features, there is the possibility of having a repetition of feature values which implies values with a small significance. In this case, since the values have been of little relevance, it is necessary to rethink the use of the PyDPI software and its potential replacement by another source of features. The results obtained were not high enough to be significant in that almost none was higher than 0.7. In addition, the differences between the two datasets studied, although there is a tendency for the quality values to be higher in the dataset with generic/domains pairs, it is not enough to conclude that the use of features coming from protein pairs interacting from peptide recognition modules, improve the performance of the detection of interactions in a random dataset. Another situation to point out was the fact that several times the performance value was higher for the datasets without domain, which in the case of kNN could be due to the use of an incorrect number of k. One of the reasons these values are so insignificant may be that the construction of the datasets has been excessively manual because it was not used computer tools in the construction of datasets and the correct location of the features (all this work was done manually as well as the import of features). All of this may have jeopardized the correct placement of the features and by the insignificant results.

4.4 Pertinence of the protein domains using new features

As the reason for the values, being insignificant may have been due to the excessive manual manipulation of the datasets, the third strategy was to use features from the article *"A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions"*. In order to increase the significance of the values, the number of protein pairs whose interaction was dependent on the enunciated domains as well as the generic proteins was also increased.

4.4.1 Procedure

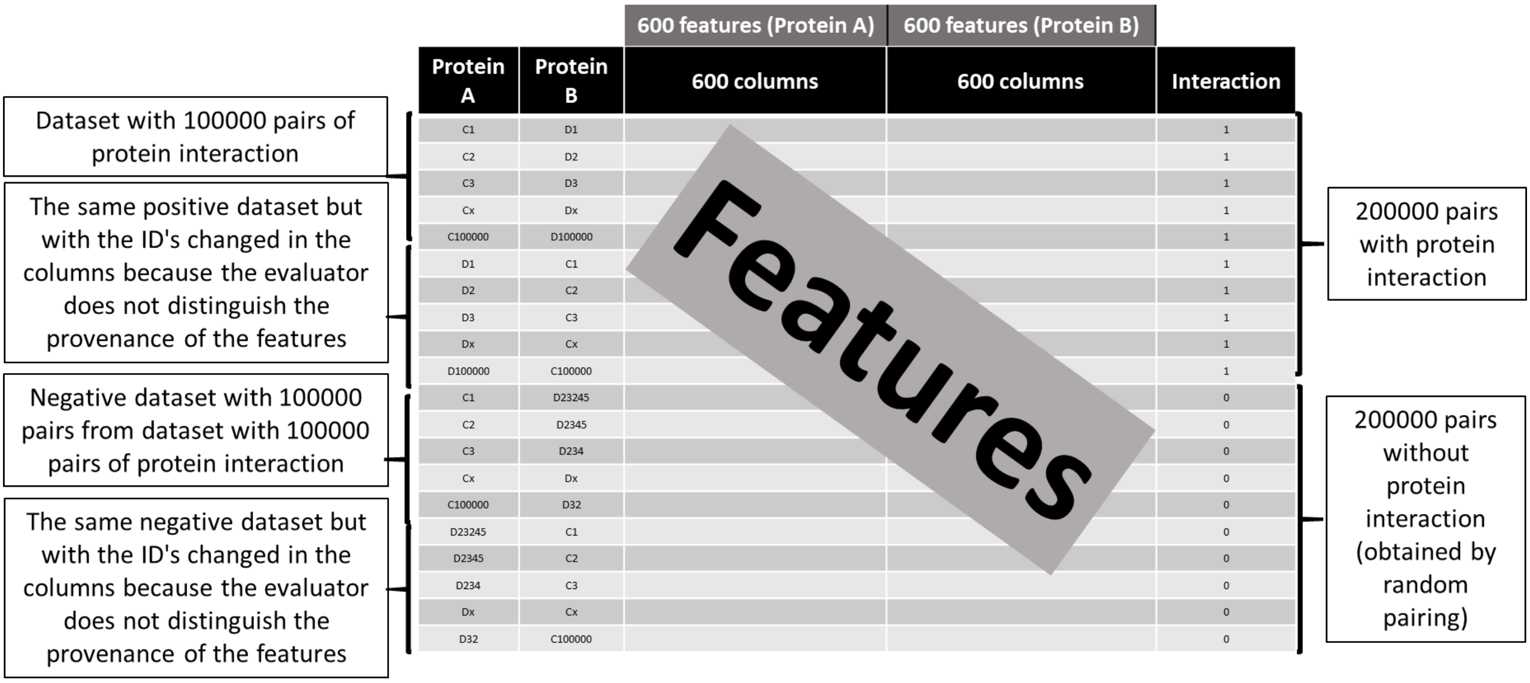
1 - In order to obtain datasets with protein IDs that interact with the peptide recognition modules, were used the same strategy of the last approach, with the difference that all the pairs of protein with interaction performed by the five domains were saved. In this way were obtained 2333 pairs of domain proteins with 1038 pairs of SH3, 439 pairs of SH2, 494 pairs of PDZ, 197 pairs of WW and 165 pairs of LRR.

2 – In order to reduce the excess of manual manipulation of the datasets, we used the features already calculated from the dataset of the article "A Sequence-Based Mesh Classifier for the Prediction of Protein-Interactions" which had 963471 pairs of proteins. In this way, it was sufficient to know the proteins of the dataset with 2333 pairs that existed in the dataset of the article and, if so, to make use of the corresponding features (to detect proteins that existed simultaneously in both datasets, we used the "COUNTIF" function of Excel). At the end, 2031 pairs of proteins were found to exist in the two datasets simultaneously, with 921 being SH3 pairs, 385 being SH2 pairs, 418 being PDZ pairs, 168 being WW pairs and 139 being pairs of LRRs.

2.1 - In terms of interaction pairs with generic proteins, pairs of proteins were imported from the data of the article cited above, taking into account that, in order to supress overfitting, the "RAND" function of Excel was needed to randomize the pairs that were in alphabetic order. In this way was created the generic dataset with 100000 protein pairs and the dataset with generic and domains with 102031 protein pairs.

3 – Subsequently, the datasets were augmented and subdivided into four parts like in the first strategy. However, because they are very large datasets and so that the different parts are detected, the protein pairs belonging to SH2 had their characters painted of red, the PDZ painted of blue, the WW painted of green and the LRR painted of yellow. Also the four parts were distinguished, in the second part the bottom were painted of green, in the third part of blue and in the fourth part of yellow (in different shades of the protein domain IDs character colours). The third part was also altered in that, column A was copied in full but, column B was randomized by the "RAND" function which was used separately for the set of ID's that had only pairs of proteins mediated by domains and the set of ID's of the generic proteins At the end the dataset with pairs of generic

proteins and pairs of proteins with domains remained with 408124 protein pairs and the dataset with only pairs of generic proteins remained with 400000 protein pairs.



		600 features (Protein A)	600 features (Protein B)	
Protein A	Protein B	600 columns	600 columns	Interaction
C1	D1			1
C2	D2			1
C3	D3			1
Cx	Dx			1
C100000	D100000			1
D1	C1			1
D2	C2			1
D3	C3			1
Dx	Cx			1
D100000	C100000			1
C1	D23245			0
C2	D2345			0
C3	D234			0
Cx	Dx			0
C100000	D32			0
D23245	C1			0
D2345	C2			0
D234	C3			0
Dx	Cx			0
D32	C100000			0

Dataset with 100000 pairs of protein interaction

The same positive dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

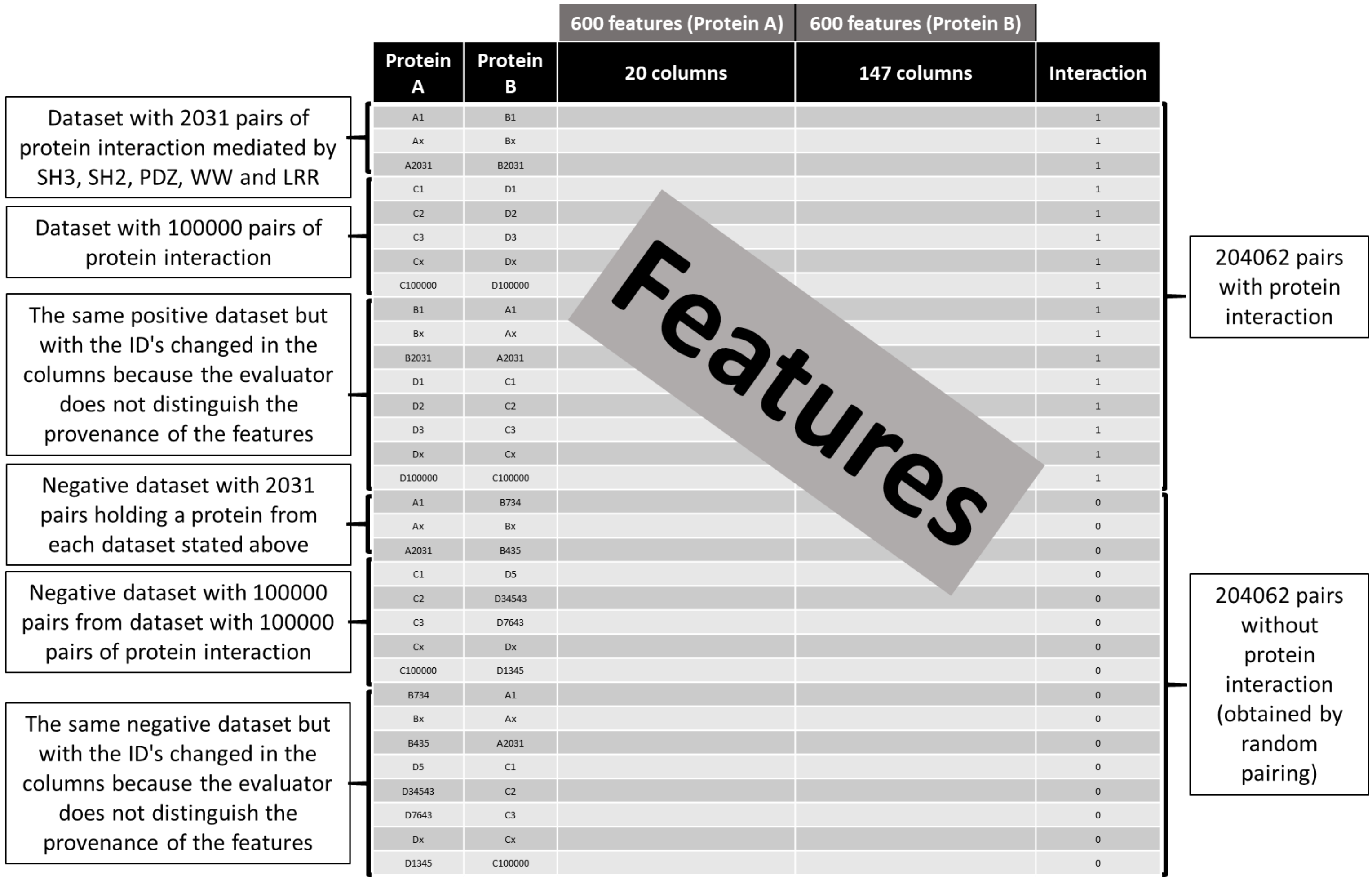
Negative dataset with 100000 pairs from dataset with 100000 pairs of protein interaction

The same negative dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

200000 pairs with protein interaction

200000 pairs without protein interaction (obtained by random pairing)

Table 8: Dataset organization of the 3rd strategy with generic protein pairs



		600 features (Protein A)	600 features (Protein B)	
Protein A	Protein B	20 columns	147 columns	Interaction
A1	B1			1
Ax	Bx			1
A2031	B2031			1
C1	D1			1
C2	D2			1
C3	D3			1
Cx	Dx			1
C100000	D100000			1
B1	A1			1
Bx	Ax			1
B2031	A2031			1
D1	C1			1
D2	C2			1
D3	C3			1
Dx	Cx			1
D100000	C100000			1
A1	B734			0
Ax	Bx			0
A2031	B435			0
C1	D5			0
C2	D34543			0
C3	D7643			0
Cx	Dx			0
C100000	D1345			0
B734	A1			0
Bx	Ax			0
B435	A2031			0
D5	C1			0
D34543	C2			0
D7643	C3			0
Dx	Cx			0
D1345	C100000			0

Dataset with 2031 pairs of protein interaction mediated by SH3, SH2, PDZ, WW and LRR

Dataset with 100000 pairs of protein interaction

The same positive dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

Negative dataset with 2031 pairs holding a protein from each dataset stated above

Negative dataset with 100000 pairs from dataset with 100000 pairs of protein interaction

The same negative dataset but with the ID's changed in the columns because the evaluator does not distinguish the provenance of the features

204062 pairs with protein interaction

204062 pairs without protein interaction (obtained by random pairing)

Table 9: Organization of 3rd strategy datasets with domains and generic pairs

4 – The method present in the article "*A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions"*, from which the features were extracted consists of few steps, first the amino acids are categorized by their physical and chemical characteristics and replaced by seven different characters (1 – Ala, Gly,Val; 2 – Ile, Leu, Phe, Pro; 3 – Tyr, Met, Thr, Ser; 4 – His, Asn, Gln, Trp; 5 – Arg, Lys; 6 – Asp, Glu; 7 – Cys), then the Discrete Cosine Transform is used to remove features from the primary protein structures (Coelho, *et al.*, 2017). In this way, 600 features were created for each 91293 different proteins in a file called "data.xls".

4.1 - To join these 600 features, with the corresponding pairs of ID's that proved to be present in the dataset of the aforementioned article, had been used the programming in C# (C Sharp) [Annex IV – Page 51]. That way the data.xls file was converted to data.csv just like the "Generic" dataset only with generic pairs and the "Genericdomain" dataset with generic pairs and those with domains interactions. The developed code allowed to open the file data.csv read it line by line and write in memory a dictionary in which the key was the 1st column (ID's) and the value the other 600 columns (features). Then the code read the datasets "Generic" and "Genericdomain" if the previous dictionary contained the keys of column 1 and column 2 simultaneously, we obtain a final string that is the value corresponding to key 1 and the value corresponding to the key 2. At the end was recorded what was in memory, namely, two files with errors that were referring to the lines in which at least one of the ID's of the columns of the datasets "Generic" and "Genericdomain" was not present in the data.csv.

4.1.1 – Of the values with errors, 132 pairs that had no features were obtained, corresponding to 66 pairs that were withdrawn in each part of the positive (first and second part) and negative (third and fourth part) datasets.

5 – Subsequently, the performances of the two datasets were evaluated using Python programming [Annex V – Page 55]. For this, the classifiers used were support vector machine, naïve Bayes, decisions tree and neuro network models, whose performance was evaluated by the following indicators: F-score, precision, recall and area under the ROC curve.

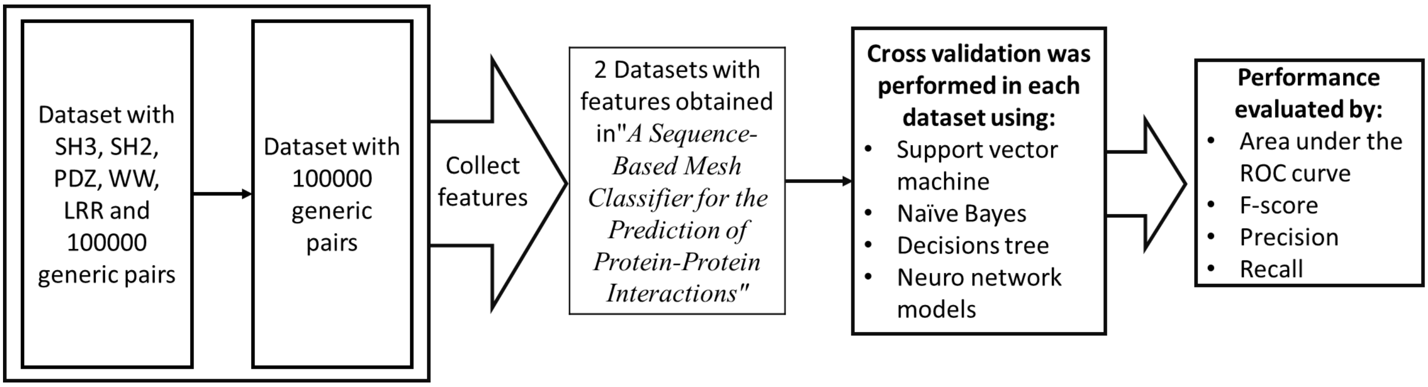

Dataset with SH3, SH2, PDZ, WW, LRR and 100000 generic pairs
Dataset with 100000 generic pairs
Collect features
2 Datasets with features obtained in"*A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions"*
Cross validation was performed in each dataset using:
• Support vector machine
• Naïve Bayes
• Decisions tree
• Neuro network models
Performance evaluated by:
• Area under the ROC curve
• F-score
• Precision
• Recall

Figure 17: Summary of the 3rd strategy model

4.4.2 Values

Classifiers	Datasets	F-score	Precision	Recall	Area under the ROC curve
SVM	Domains + Generic	0.5289767005798812	0.5026561520440822	0.5582060019786743	0.5010854415972757
	Generic	0.5637514094606282	0.5033821430666067	0.640573688924271	0.5009834606821567
Naïve Bayes	Domains + Generic	0.4678854514484599	0.47114479615594196	0.46467089272409706	0.4697807269680197
	Generic	0.4640195062788994	0.47139226859695055	0.45687381829037715	0.4695815116893084
Decisions Tree	Domains + Generic	0.5246099023387617	0.5223862521647512	0.5268525643374494	0.520951735909318
	Generic	0.5265636006376747	0.5231564472165606	0.5300154244203403	0.5209217853159069
NNM	Domains + Generic	0.5223614247213927	0.5192036199095023	0.5255578762229306	0.5177959072718608
	Generic	0.5213094350550527	0.5164882122407121	0.526221514578565	0.5142106114800412

Table 10: Results of the 3rd strategy

In the table above, in black are the quality values whose dataset performance with generic proteins was higher than the values of the dataset with generic proteins and proteins with domains. Overall, all values are around 0.5. The classifier that had the best performance comparing the dataset with generic/domains protein pairs with the dataset with only generic protein pairs was the neuro network models and the worst was support vector machine. As for the quality indicators, what gave the best results in terms of demonstrating that the generic dataset / domains had a better performance was the area under the ROC curve and what had worse results was the precision.

4.4.3 Discussion

In this case the datasets were quite large, the source of features was reliable (they gave rise to positive results in the article "*A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions*") and minimal manual handling (manipulation was used manual only in the initial organization of the datasets by the ID's). However the size of the datasets may have been disadvantageous because, because they were too large, it was impossible to open and analyze them (computationally the computer could not open such large documents), so potential inconsistencies in the datasets could not be detected and corrected. Since it does not hold great knowledge in the area of programming, perhaps the codes used may contain some error that may have translated into errors that have taken significance to the values obtained. The results obtained were not high enough to be significant in that almost none was higher than 0.7. In addition, the differences between the two datasets studied, although there is a tendency for the quality values to be higher in the dataset with generic/domains pairs, it is not enough to conclude that the use of features coming from protein pairs interacting from peptide recognition modules, improve the performance of the detection of interactions in a random dataset. However, it is quite relevant that AUC, considered the quality indicator par excellence, has demonstrated in this strategy to always be higher in the values corresponding to the Domains + Generic dataset.

5. CONCLUSION

In this thesis, it was intended to improve the prediction of protein interactions from computational methods using feature extraction. In this way, several methodologies were analysed, and three pertinent approaches were formulated: the use of inhibitor databases, the use of gene co-expression networks and the use of peptide recognition modules. From these three, due to the greater pertinence, studies using the last mentioned approach were developed, using the SH3, SH2, PDZ, WW and LRR domains that were evaluated in the detection of new interactions between proteins that do not possess them (in order to evaluate the relevance of their features to be used in a random dataset). Thus during the work of this thesis were created three strategies, the first one was based on the extraction of features by the software PyDPI and in the evaluation of the performance of the datasets by descriptor; the second strategy resorted to the same features but evaluated the performance in datasets with all descriptors; the third strategy evaluated the performance as the second strategy but using features created for the article "*A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions*".

Despite the different strategies employed in the context of this thesis, the relevance of peptide recognition modules in the detection of protein interactions in random datasets could not be conclusively verified. Although the performance values are generally in the order of 0.5, a tendency of the same ones to be superior in the datasets that have interactions that are known to be mediated by them is detected.

However, this work has contributed to the formulation of a new field of strategies in the detection of protein-protein interactions using computational methodologies since, although no relevant values have been obtained, it can serve as a basis for other working groups that, with better training computing, can detect relevance in the approach used here, and even in the other two that have not yet been explored (databases of inhibitors and co-expression networks).

As improvements a smaller dataset could have been used that was easier to handle (detect errors), or have tried to get other features with other strategies. However, the lack of knowledge in the programming area and the time available for the construction of new strategies was limiting.

For the future, should will be a focus on the use of new features and other strategies to prove the appropriateness of the approach developed or the other two approaches, since the little significance of the current values can be due to the lack of knowledge in the programming area and to faults in the handling of the features.

6. BIBLIOGRAPHY

Cao, Dongsheng. *Molecular Descriptors Guide*: Description of the Molecular Descriptors Appearing in the PyDPI Software Package. China. 2012. pp. 39-47

Coelho, E., Arrais, J. and Oliveira, J. (2013). From Protein-Protein Interactions to Rational Drug Design: Are Computational Methods Up to the Challenge?. Current Topics in Medicinal Chemistry, 13(5).

Coelho, E., Arrais, J., Matos, S., Pereira, C., Rosa, N., Correia, M., Barros, M. and Oliveira, J. (2014). Computational prediction of the human-microbial oral interactome. BMC Systems Biology, 8(1), p.24.

Coelho, E., Cruz, I., Santiago, A., Oliveira, J., Dourado, A., and Arrais, J. (2017). A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions.

Forcier, Jeff. Stochastic Gradient Descent. Orange Data Mining. 2015. Available in: https://docs.orange.biolab.si/3/visual-programming/widgets/model/stochasticgradient.html. Access in: 10/7/2018

Gurung, A., Bhattacharjee, A., Ajmal Ali, M., Al-Hemaid, F. and Lee, J. (2017). Binding of small molecules at interface of protein–protein complex – A newer approach to rational drug design. Saudi Journal of Biological Sciences, 24(2), pp.379-388.

Huang, Y., You, Z., Gao, X., Wong, L. and Wang, L. (2015). Using Weighted Sparse Representation Model Combined with Discrete Cosine Transformation to Predict Protein-Protein Interactions from Protein Sequence. BioMed Research International, 2015, pp.1-10.

Hue, M., Riffle, M., Vert, J. and Noble, W. (2010). Large-scale prediction of protein-protein interactions from structures. BMC Bioinformatics, 11(1), p.144.

Jain, S. and Bader, G. (2016). Predicting physiologically relevant SH3 domain mediated protein–protein interactions in yeast. Bioinformatics, 32(12), pp.1865-1872.

London, N., Raveh, B. and Schueler-Furman, O. (2013). Druggable protein–protein interactions – from hot spots to hot segments. Current Opinion in Chemical Biology, 17(6), pp.952-959.

Maruyama, O. (2013). Heterodimeric protein complex identification by naïve Bayes classifiers. BMC Bioinformatics, 14(1), p.347.

Melo, R., Fieldhouse, R., Melo, A., Correia, J., Cordeiro, M., Gümüş, Z., Costa, J., Bonvin, A. and Moreira, I. (2016). A Machine Learning Approach for Hot-Spot Detection at Protein-Protein Interfaces. International Journal of Molecular Sciences, 17(8), p.1215.

Moreira, I., Fernandes, P. and Ramos, M. (2006). Unraveling the Importance of Protein−Protein Interaction: Application of a Computational Alanine-Scanning Mutagenesis to the Study of the IgG1 Streptococcal Protein G (C2 Fragment) Complex. The Journal of Physical Chemistry B, 110(22), pp.10962-10969.

Moreira, I., Fernandes, P. and Ramos, M. (2007). Backbone Importance for Protein−Protein Binding†. Journal of Chemical Theory and Computation, 3(3), pp.885-893.

Ng, A., Eisenberg, J., Heath, R., Huett, A., Robinson, C., Nau, G. and Xavier, R. (2010). Human leucine-rich repeat proteins: a genome-wide bioinformatic categorization and functional analysis in innate immunity. Proceedings of the National Academy of Sciences, 108(Supplement_1), pp.4631-4638.

Reimand, J., Hui, S., Jain, S., Law, B. and Bader, G. (2012). Domain-mediated protein interaction prediction: From genome to network. FEBS Letters, 586(17), pp.2751-2763.

Saethang, T., Payne, D., Avihingsanon, Y. and Pisitkun, T. (2016). A machine learning strategy for predicting localization of post-translational modification sites in protein-protein interacting regions. BMC Bioinformatics, 17(1).

Song, M., Yu, H. and Han, W. (2011). Combining active learning and semi-supervised learning techniques to extract protein interaction sentences. BMC Bioinformatics, 12(Suppl 12), p.S4.

Sugaya, N. and Ikeda, K. (2009). Assessing the druggability of protein-protein interactions by a supervised machine-learning method. BMC Bioinformatics, 10(1), p.263.

Tong, L., Warren, T., King, J., Betageri, R., Rose, J. and Jakes, S. (1996). Crystal Structures of the Human p56lckSH2 Domain in Complex with Two Short Phosphotyrosyl Peptides at 1.0 Å and 1.8 Å Resolution. Journal of Molecular Biology, 256(3), pp.601-610.

Tugaeva, K., Tsvetkov, P. and Sluchanko, N. (2017). Bacterial co-expression of human Tau protein with protein kinase A and 14-3-3 for studies of 14-3-3/phospho-Tau interaction. PLOS ONE, 12(6), p.e0178933.

Vella, D., Zoppis, I., Mauri, G., Mauri, P. and Di Silvestre, D. (2017). From protein-protein interactions to protein co-expression networks: a new perspective to evaluate large-scale proteomic data. EURASIP Journal on Bioinformatics and Systems Biology, 2017(1).

Walker, S. and Duncan, D. (1967). Estimation of the Probability of an Event as a Function of Several Independent Variables. Biometrika, 54(1/2), p.167

You, Z., Li, J., Gao, X., He, Z., Zhu, L., Lei, Y. and Ji, Z. (2015). Detecting Protein-Protein Interactions with a Novel Matrix-Based Protein Sequence Representation and Support Vector Machines. BioMed Research International, 2015, pp.1-9.

Zahiri, J., Bozorgmehr, J. and Masoudi-Nejad, A. (2013). Computational Prediction of Protein–Protein Interaction Networks: Algorithms and Resources. Current Genomics, 14(6), pp.397-414.

Zhang, H., Jiang, T. and Shan, G. (2016). Identification of Hot Spots in Protein Structures Using Gaussian Network Model and Gaussian Naive Bayes. BioMed Research International, 2016, pp.1-9.

Zhang, T., Wang, X. and Yue, Z. (2017). Identification of candidate genes related to pancreatic cancer based on analysis of gene co-expression and protein-protein interaction network. Oncotarget.

7. ANNEX

Annex I – Features import code by PyDPI

```
from pydpi import pypro
from pydpi.pypro import AAComposition
import csv
id_list = [xxx,xxx,xxx,xxx,xxx,xxx,xxx,xxx,xxx,…]
sequences = []
AAComp_features = ["A", "R", "N", "D", "C", "Q", "E", "G", "H", "I", "L", "K", "M", "F", "P", "S", "T", "W", "Y", "V"]

def csv_writer(file_name, input_dict,input_name,input_feature_list):
    with open(file_name,"ab") as csv_file:
        writer = csv.writer(csv_file, delimiter = ";")
        final_comp_values = [input_name]
        for AAC_comp in input_feature_list:
            final_comp_values.append(input_dict[AAC_comp])
        writer.writerow(final_comp_values)

for prot_id in id_list:
    current_sequence = pypro.GetProteinSequence(prot_id)
    sequences.append(current_sequence)

csv_writer("test.csv",AAComposition.CalculateAAComposition(current_sequence),prot_id,AAComp_features)
```

<u>Subtitle:</u>

The "id_list" corresponds to the protein ID´s separated by commas;

The "test.csv" file was the csv file to which the features were exported;

In order to obtain the features of the other descriptors the following substitutions were made:

- In the case of CTD

Instead of "from pydpi.pypro import AAComposition" was used "from pydpi.pypro import CTD"

Instead of "AAComposition.CalculateAAComposition" was used "CTD.CalculateCTD"

AAComp_features = ['_NormalizedVDWVD1075', '_PolarityD1075', '_SecondaryStrD3025', '_PolarityD3100', '_ChargeD1100', '_SecondaryStrT23', '_PolarityD3025',

```
'_NormalizedVDWVC1', '_NormalizedVDWVC3', '_HydrophobicityT23',
'_SolventAccessibilityD1025', '_PolarityD1100', '_NormalizedVDWVD2100', '_ChargeD3050',
'_PolarityD2001', '_SolventAccessibilityD2025', '_SecondaryStrD2025', '_PolarityT12',
'_PolarityT13', '_ChargeT23', '_HydrophobicityD2100', '_SolventAccessibilityD3001',
'_ChargeD1075', '_SecondaryStrD1075', '_PolarizabilityD2050', '_SolventAccessibilityD3100',
'_PolarizabilityD1075', '_HydrophobicityD3025', '_PolarizabilityD3025', '_SecondaryStrD3050',
'_SolventAccessibilityD2050', '_HydrophobicityD2001', '_SecondaryStrD2075',
'_PolarityD3050', '_PolarityD1001', '_ChargeD2075', '_NormalizedVDWVD3025',
'_SolventAccessibilityD1100', '_SecondaryStrD1100', '_NormalizedVDWVD1001',
'_NormalizedVDWVD2050', '_NormalizedVDWVC2', '_PolarizabilityC2', '_PolarizabilityC3',
'_PolarizabilityC1', '_SecondaryStrC2', '_SecondaryStrC3', '_ChargeT12', '_ChargeT13',
'_ChargeD1001', '_NormalizedVDWVD3001', '_NormalizedVDWVD2025',
'_NormalizedVDWVD3050', '_SecondaryStrD1025', '_SolventAccessibilityD1050',
'_PolarizabilityD3001', '_PolarityD2075', '_SecondaryStrD3001', '_ChargeD2025',
'_HydrophobicityD2050', '_PolarizabilityD1100', '_HydrophobicityC1', '_HydrophobicityC2',
'_SolventAccessibilityD2100', '_PolarizabilityD2001', '_PolarizabilityD1025',
'_NormalizedVDWVT13', '_PolarizabilityD3050', '_SolventAccessibilityT13',
'_SolventAccessibilityT12', '_PolarityD1050', '_ChargeD3075', '_SolventAccessibilityD1001',
'_HydrophobicityC3', '_HydrophobicityD3100', '_SecondaryStrD1050', '_ChargeD3001',
'_SecondaryStrD2050', '_PolarityD2025', '_NormalizedVDWVD2001',
'_HydrophobicityD3075', '_ChargeD1025', '_SolventAccessibilityD2001',
'_HydrophobicityD1025', '_ChargeD1050', '_SolventAccessibilityD3025',
'_PolarizabilityD3075', '_PolarityD2100', '_SecondaryStrD3100', '_PolarizabilityT13',
'_PolarizabilityT12', '_SecondaryStrD2001', '_NormalizedVDWVD1050',
'_PolarizabilityD1050', '_SolventAccessibilityT23', '_ChargeC3', '_ChargeC2', '_ChargeC1',
'_NormalizedVDWVT23', '_NormalizedVDWVD3100', '_PolarizabilityD2025',
'_SolventAccessibilityD3075', '_HydrophobicityD2075', '_HydrophobicityD1075',
'_ChargeD2050', '_ChargeD3100', '_SecondaryStrC1', '_ChargeD3025', '_PolarityD3001',
'_PolarityD3075', '_PolarityD1025', '_SecondaryStrT13', '_SecondaryStrT12',
'_HydrophobicityD1001', '_SolventAccessibilityD1075', '_HydrophobicityD1050',
'_HydrophobicityT13', '_HydrophobicityT12', '_SecondaryStrD1001',
'_NormalizedVDWVD1100', '_NormalizedVDWVD3075', '_NormalizedVDWVD1025',
'_SolventAccessibilityD2075', '_SecondaryStrD2100', '_PolarizabilityD3100',
'_SecondaryStrD3075', '_NormalizedVDWVT12', '_PolarizabilityT23',
'_SolventAccessibilityD3050', '_PolarityT23', '_PolarityD2050', '_HydrophobicityD2025',
'_PolarizabilityD2075', '_HydrophobicityD3050', '_HydrophobicityD3001',
'_HydrophobicityD1100', '_ChargeD2001', '_SolventAccessibilityC1',
```

'_SolventAccessibilityC2', '_SolventAccessibilityC3', '_PolarizabilityD1001', '_NormalizedVDWVD2075', '_ChargeD2100', '_PolarizabilityD2100', '_PolarityC1', '_PolarityC3', '_PolarityC2']

- In the case of Moranauto

Instead of "from pydpi.pypro import AAComposition" was used "from pydpi.pypro import Autocorrelation"

Instead of "AAComposition.CalculateAAComposition" was used "Autocorrelation.CalculateMoranAutoTotal"

AAComp_features = ['MoranAuto_ResidueVol8', 'MoranAuto_ResidueVol9', 'MoranAuto_ResidueVol4', 'MoranAuto_ResidueVol5', 'MoranAuto_ResidueVol6', 'MoranAuto_ResidueVol7', 'MoranAuto_ResidueVol1', 'MoranAuto_ResidueVol2', 'MoranAuto_ResidueVol3', 'MoranAuto_Steric2', 'MoranAuto_FreeEnergy1', 'MoranAuto_FreeEnergy2', 'MoranAuto_FreeEnergy3', 'MoranAuto_FreeEnergy4', 'MoranAuto_FreeEnergy5', 'MoranAuto_FreeEnergy6', 'MoranAuto_FreeEnergy7', 'MoranAuto_FreeEnergy8', 'MoranAuto_FreeEnergy9', 'MoranAuto_Steric8', 'MoranAuto_Steric9', 'MoranAuto_Hydrophobicity15', 'MoranAuto_ResidueASA30', 'MoranAuto_Steric28', 'MoranAuto_Steric30', 'MoranAuto_ResidueASA26', 'MoranAuto_Hydrophobicity13', 'MoranAuto_Steric1', 'MoranAuto_Steric7', 'MoranAuto_ResidueASA23', 'MoranAuto_ResidueASA20', 'MoranAuto_AvFlexibility30', 'MoranAuto_Hydrophobicity10', 'MoranAuto_ResidueASA29', 'MoranAuto_ResidueASA28', 'MoranAuto_Hydrophobicity19', 'MoranAuto_Hydrophobicity18', 'MoranAuto_Hydrophobicity17', 'MoranAuto_Hydrophobicity16', 'MoranAuto_ResidueASA27', 'MoranAuto_Hydrophobicity14', 'MoranAuto_ResidueASA21', 'MoranAuto_Hydrophobicity12', 'MoranAuto_Hydrophobicity11', 'MoranAuto_ResidueASA22', 'MoranAuto_Steric21', 'MoranAuto_Steric20', 'MoranAuto_Steric23', 'MoranAuto_Steric22', 'MoranAuto_Steric25', 'MoranAuto_Steric24', 'MoranAuto_ResidueVol18', 'MoranAuto_ResidueVol19', 'MoranAuto_ResidueVol16', 'MoranAuto_ResidueVol17', 'MoranAuto_ResidueVol14', 'MoranAuto_ResidueVol15', 'MoranAuto_ResidueVol12', 'MoranAuto_ResidueVol13', 'MoranAuto_ResidueVol10', 'MoranAuto_ResidueVol11', 'MoranAuto_AvFlexibility26', 'MoranAuto_AvFlexibility27', 'MoranAuto_AvFlexibility24', 'MoranAuto_AvFlexibility25', 'MoranAuto_AvFlexibility22', 'MoranAuto_AvFlexibility23', 'MoranAuto_AvFlexibility20', 'MoranAuto_AvFlexibility21', 'MoranAuto_ResidueASA25', 'MoranAuto_Steric6', 'MoranAuto_Mutability30', 'MoranAuto_AvFlexibility28', 'MoranAuto_AvFlexibility29', 'MoranAuto_AvFlexibility19', 'MoranAuto_AvFlexibility18', 'MoranAuto_Steric4', 'MoranAuto_Steric5', 'MoranAuto_AvFlexibility13', 'MoranAuto_AvFlexibility12', 'MoranAuto_AvFlexibility11', 'MoranAuto_AvFlexibility10', 'MoranAuto_AvFlexibility17', 'MoranAuto_AvFlexibility16', 'MoranAuto_AvFlexibility15', 'MoranAuto_AvFlexibility14', 'MoranAuto_ResidueASA18', 'MoranAuto_ResidueASA19', 'MoranAuto_ResidueASA24', 'MoranAuto_ResidueASA10', 'MoranAuto_ResidueASA11', 'MoranAuto_ResidueASA12', 'MoranAuto_ResidueASA13', 'MoranAuto_ResidueASA14', 'MoranAuto_ResidueASA15', 'MoranAuto_ResidueASA16', 'MoranAuto_ResidueASA17', 'MoranAuto_Steric14', 'MoranAuto_Steric15', 'MoranAuto_Steric16', 'MoranAuto_Steric17', 'MoranAuto_Steric10', 'MoranAuto_Steric11', 'MoranAuto_Steric12', 'MoranAuto_Steric13', 'MoranAuto_Steric18', 'MoranAuto_Steric19', 'MoranAuto_Mutability29', 'MoranAuto_Mutability6', 'MoranAuto_Mutability7', 'MoranAuto_Mutability4', 'MoranAuto_Mutability5', 'MoranAuto_Mutability2', 'MoranAuto_Mutability3', 'MoranAuto_Mutability1', 'MoranAuto_Mutability8', 'MoranAuto_Mutability9', 'MoranAuto_Mutability10', 'MoranAuto_Mutability28', 'MoranAuto_Mutability12', 'MoranAuto_Mutability13', 'MoranAuto_Mutability14', 'MoranAuto_Mutability15', 'MoranAuto_Mutability16', 'MoranAuto_Mutability17', 'MoranAuto_ResidueVol30', 'MoranAuto_Mutability19', 'MoranAuto_Hydrophobicity9', 'MoranAuto_Hydrophobicity8', 'MoranAuto_Hydrophobicity7', 'MoranAuto_Hydrophobicity6', 'MoranAuto_Hydrophobicity5', 'MoranAuto_Hydrophobicity4', 'MoranAuto_Hydrophobicity3', 'MoranAuto_Hydrophobicity2', 'MoranAuto_Hydrophobicity1', 'MoranAuto_Polarizability19', 'MoranAuto_Polarizability18', 'MoranAuto_Steric3', 'MoranAuto_Polarizability11', 'MoranAuto_Polarizability10', 'MoranAuto_Polarizability13', 'MoranAuto_Polarizability12', 'MoranAuto_Polarizability15', 'MoranAuto_Polarizability14', 'MoranAuto_Polarizability17', 'MoranAuto_Polarizability16', 'MoranAuto_Hydrophobicity25', 'MoranAuto_Steric27', 'MoranAuto_FreeEnergy18', 'MoranAuto_FreeEnergy19', 'MoranAuto_FreeEnergy12', 'MoranAuto_FreeEnergy13', 'MoranAuto_FreeEnergy10', 'MoranAuto_FreeEnergy11', 'MoranAuto_FreeEnergy16', 'MoranAuto_FreeEnergy17', 'MoranAuto_FreeEnergy14', 'MoranAuto_FreeEnergy15', 'MoranAuto_ResidueVol27', 'MoranAuto_ResidueVol26', 'MoranAuto_ResidueVol25', 'MoranAuto_ResidueVol24', 'MoranAuto_ResidueVol23', 'MoranAuto_ResidueVol22', 'MoranAuto_ResidueVol21', 'MoranAuto_ResidueVol20', 'MoranAuto_Steric26', 'MoranAuto_ResidueVol29', 'MoranAuto_ResidueVol28', 'MoranAuto_Polarizability28', 'MoranAuto_Polarizability29', 'MoranAuto_Mutability11', 'MoranAuto_Polarizability24', 'MoranAuto_Polarizability25', 'MoranAuto_Polarizability26', 'MoranAuto_Polarizability27', 'MoranAuto_Polarizability20', 'MoranAuto_Polarizability21', 'MoranAuto_Polarizability22', 'MoranAuto_Polarizability23', 'MoranAuto_Steric29', 'MoranAuto_Mutability18', 'MoranAuto_ResidueASA8', 'MoranAuto_ResidueASA9', 'MoranAuto_ResidueASA6', 'MoranAuto_ResidueASA7', 'MoranAuto_ResidueASA4', 'MoranAuto_ResidueASA5', 'MoranAuto_ResidueASA2', 'MoranAuto_ResidueASA3', 'MoranAuto_ResidueASA1', 'MoranAuto_Hydrophobicity26', 'MoranAuto_Hydrophobicity27', 'MoranAuto_Hydrophobicity24', 'MoranAuto_Polarizability30', 'MoranAuto_Hydrophobicity22', 'MoranAuto_Hydrophobicity23', 'MoranAuto_Hydrophobicity20', 'MoranAuto_Hydrophobicity21', 'MoranAuto_Hydrophobicity28', 'MoranAuto_Hydrophobicity29', 'MoranAuto_AvFlexibility9', 'MoranAuto_AvFlexibility8', 'MoranAuto_AvFlexibility7', 'MoranAuto_AvFlexibility6', 'MoranAuto_AvFlexibility5', 'MoranAuto_AvFlexibility4', 'MoranAuto_AvFlexibility3', 'MoranAuto_AvFlexibility2', 'MoranAuto_AvFlexibility1', 'MoranAuto_Polarizability9', 'MoranAuto_Polarizability8', 'MoranAuto_FreeEnergy30', 'MoranAuto_Polarizability1', 'MoranAuto_Polarizability3', 'MoranAuto_Polarizability2', 'MoranAuto_Polarizability5', 'MoranAuto_Polarizability4', 'MoranAuto_Polarizability7', 'MoranAuto_Polarizability6', 'MoranAuto_Mutability25', 'MoranAuto_Hydrophobicity30', 'MoranAuto_Mutability24', 'MoranAuto_Mutability27', 'MoranAuto_Mutability26', 'MoranAuto_Mutability21', 'MoranAuto_Mutability20', 'MoranAuto_Mutability23', 'MoranAuto_Mutability22', 'MoranAuto_FreeEnergy23', 'MoranAuto_FreeEnergy22', 'MoranAuto_FreeEnergy21', 'MoranAuto_FreeEnergy20', 'MoranAuto_FreeEnergy27', 'MoranAuto_FreeEnergy26', 'MoranAuto_FreeEnergy25', 'MoranAuto_FreeEnergy24', 'MoranAuto_FreeEnergy29', 'MoranAuto_FreeEnergy28']

- In the case of SOCN

Instead of "from pydpi.pypro import AAComposition" was used "from pydpi.pypro import QuasiSequenceOrder"
Instead of "AAComposition.CalculateAAComposition" was used "QuasiSequenceOrder.GetSequenceOrderCouplingNumberTotal"
AAComp_features = ['taugrant23', 'taugrant24', 'tausw8', 'tausw9', 'tausw6', 'tausw7', 'tausw4', 'tausw5', 'tausw2', 'tausw3', 'tausw1', 'taugrant26', 'taugrant30', 'tausw29', 'tausw28', 'taugrant25', 'tausw21', 'tausw20', 'tausw23', 'tausw22', 'tausw25', 'tausw24', 'tausw27', 'tausw26', 'taugrant15', 'taugrant14', 'taugrant17', 'taugrant16', 'taugrant11', 'taugrant10', 'taugrant13', 'taugrant12', 'taugrant19', 'taugrant18', 'taugrant9', 'taugrant8', 'taugrant5', 'taugrant4', 'taugrant7', 'taugrant6', 'taugrant1', 'taugrant3', 'taugrant2', 'tausw30', 'taugrant27', 'taugrant28', 'taugrant29', 'tausw18', 'tausw19', 'tausw14', 'tausw15', 'tausw16', 'tausw17', 'tausw10', 'tausw11', 'tausw12', 'tausw13', 'taugrant20', 'taugrant21', 'taugrant22']

- In the case of QSO

Instead of "from pydpi.pypro import AAComposition" was used "from pydpi.pypro import QuasiSequenceOrder"
Instead of "AAComposition.CalculateAAComposition" was used "QuasiSequenceOrder.GetSequenceOrderCouplingNumberTotal"

AAComp_features = ['QSOSW39', 'QSOSW38', 'QSOSW33', 'QSOSW32', 'QSOSW31', 'QSOSW30', 'QSOSW37', 'QSOSW36', 'QSOSW35', 'QSOSW34', 'QSOSW1', 'QSOSW3', 'QSOSW2', 'QSOSW5', 'QSOSW4', 'QSOSW7', 'QSOSW6', 'QSOSW9', 'QSOSW8', 'QSOgrant49', 'QSOgrant48', 'QSOgrant41', 'QSOgrant40', 'QSOgrant43', 'QSOgrant42', 'QSOgrant45', 'QSOgrant44', 'QSOgrant47', 'QSOgrant46', 'QSOSW28', 'QSOSW29', 'QSOSW20', 'QSOSW21', 'QSOSW22', 'QSOSW23', 'QSOSW24', 'QSOSW25', 'QSOSW26', 'QSOSW27', 'QSOgrant38', 'QSOgrant39', 'QSOgrant30', 'QSOgrant31', 'QSOgrant32', 'QSOgrant33', 'QSOgrant34', 'QSOgrant35', 'QSOgrant36', 'QSOgrant37', 'QSOSW50', 'QSOgrant29', 'QSOgrant28', 'QSOgrant27', 'QSOgrant26', 'QSOSW18', 'QSOgrant24', 'QSOgrant23', 'QSOgrant25', 'QSOgrant21', 'QSOgrant20', 'QSOgrant22', 'QSOSW11', 'QSOSW10', 'QSOSW48', 'QSOSW49', 'QSOSW46', 'QSOSW47', 'QSOSW44', 'QSOSW45', 'QSOSW42', 'QSOSW43', 'QSOSW40', 'QSOSW41', 'QSOgrant16', 'QSOgrant17', 'QSOgrant14', 'QSOgrant15', 'QSOgrant12', 'QSOgrant13', 'QSOgrant10', 'QSOgrant11', 'QSOgrant50', 'QSOgrant18', 'QSOgrant19', 'QSOSW15', 'QSOSW14', 'QSOSW17', 'QSOSW16', 'QSOgrant8', 'QSOgrant9', 'QSOSW13', 'QSOSW12', 'QSOgrant4', 'QSOgrant5', 'QSOgrant6', 'QSOgrant7', 'QSOSW19', 'QSOgrant1', 'QSOgrant2', 'QSOgrant3']

- In the case of CT

Instead of "from pydpi.pypro import AAComposition" was used "from pydpi.pypro import ConjointTriad"

Instead of "AAComposition.CalculateAAComposition" was used "ConjointTriad.CalculateConjointTriad"

AAComp_features = ['010', '011', '012', '013', '014', '015', '016', '017', '344', '345', '346', '347', '340', '341', '342', '343', '717', '716', '715', '714', '713', '712', '711', '710', '270', '271', '272', '273', '274', '275', '276', '277', '524', '525', '526', '527', '520', '521', '522', '523', '443', '442', '441', '440', '447', '446', '445', '444', '102', '103', '100', '101', '106', '107', '104', '105', '601', '641', '640', '643', '642', '645', '644', '647', '646', '436', '437', '434', '435', '432', '433', '430', '431', '335', '334', '337', '336', '331', '330', '333', '332', '054', '055', '056', '057', '050', '051', '052', '053', '740', '741', '742', '743', '600', '745', '746', '747', '555', '554', '557', '556', '551', '550', '553', '552', '234', '235', '236', '237', '230', '231', '232', '233', '146', '147', '144', '145', '142', '143', '140', '141', '612', '613', '610', '611', '616', '617', '614', '615', '133', '132', '131', '130', '137', '136', '135', '134', '407', '406', '405', '404', '403', '402', '401', '400', '025', '024', '027', '026', '021', '020', '023', '022', '371', '370', '373', '372', '375', '374', '377', '376', '704', '705', '706', '707', '700', '701', '702', '703', '607', '245', '244', '247', '246', '241', '240', '243', '242', '511', '510', '513', '512', '515', '514', '517', '516', '623', '622', '621', '620', '627', '626', '625', '624', '450', '451', '452', '453', '454', '455', '456', '457', '177', '176', '175', '174', '173', '172', '171', '170', '656', '657', '654', '655', '652', '653', '650', '651', '061', '060', '063', '062', '065', '064', '067', '066', '322', '323', '320', '321', '326', '327', '324', '325', '201', '200',

```
'203', '202', '205', '204', '207', '206', '667', '666', '665', '664', '663', '662', '661', '660', '542', '543',
'540', '541', '546', '547', '544', '545', '120', '121', '122', '123', '124', '125', '126', '127', '414', '415',
'416', '417', '410', '411', '412', '413', '776', '313', '312', '311', '310', '317', '316', '315', '314', '032',
'033', '030', '031', '036', '037', '034', '035', '366', '367', '364', '365', '362', '363', '360', '361', '605',
'604', '573', '572', '571', '570', '577', '576', '575', '574', '606', '252', '253', '250', '251', '256', '257',
'254', '255', '603', '602', '731', '730', '733', '732', '735', '734', '737', '736', '506', '507', '504', '505',
'502', '503', '500', '501', '630', '631', '632', '633', '634', '635', '636', '637', '465', '464', '467', '466',
'461', '460', '463', '462', '164', '165', '166', '167', '160', '161', '162', '163', '076', '077', '074', '075',
'072', '073', '070', '071', '003', '002', '001', '000', '007', '006', '005', '004', '357', '356', '355', '354',
'353', '352', '351', '350', '216', '217', '214', '215', '212', '213', '210', '211', '762', '763', '760', '761',
'766', '767', '764', '765', '674', '675', '676', '677', '670', '671', '672', '673', '263', '262', '261', '260',
'267', '266', '265', '264', '537', '536', '535', '534', '533', '532', '531', '530', '775', '774', '777', '115',
'114', '117', '116', '111', '110', '113', '112', '771', '770', '773', '772', '421', '420', '423', '422', '425',
'424', '427', '426', '300', '301', '302', '303', '304', '305', '306', '307', '047', '046', '045', '044', '043',
'042', '041', '040', '744', '753', '752', '751', '750', '757', '756', '755', '754', '560', '561', '562', '563',
'564', '565', '566', '567', '227', '226', '225', '224', '223', '222', '221', '220', '726', '727', '724', '725',
'722', '723', '720', '721', '151', '150', '153', '152', '155', '154', '157', '156', '472', '473', '470', '471',
'476', '477', '474', '475']
```

Annex II – Results of the 1st strategy

Descriptor	Domain	Classifier	AUC	ACC	F1	PPV	TPR	AUC dif	ACC dif	F1 dif	PPV dif	TPR dif	∑ dif	∑ domain	∑ descriptor
AAC	SH3	Naive Bayes	0,554	0,551	0,481	0,569	0,417	0,335	0,258	0,190	0,277	0,126	1,186	3,657	16,221
		kNN	0,451	0,453	0,498	0,460	0,543	0,018	-0,001	0,025	0,003	0,053	0,098		
		SVM	0,577	0,535	0,612	0,525	0,733	0,069	0,044	0,053	0,032	0,088	0,286		
		Tree	0,544	0,539	0,543	0,538	0,549	0,085	0,077	0,091	0,077	0,105	0,435		
		LR	0,619	0,614	0,490	0,722	0,371	0,299	0,251	0,135	0,362	0,020	1,067		
		SGD	0,621	0,621	0,518	0,710	0,407	0,159	0,159	0,062	0,249	-0,044	0,585		
	SH2	Naive Bayes	0,562	0,555	0,495	0,572	0,436	0,343	0,262	0,204	0,280	0,145	1,234	3,744	
		kNN	0,461	0,466	0,484	0,469	0,501	0,028	0,012	0,011	0,012	0,011	0,074		
		SVM	0,565	0,532	0,595	0,524	0,688	0,057	0,041	0,036	0,031	0,043	0,208		
		Tree	0,572	0,565	0,561	0,566	0,556	0,113	0,103	0,109	0,105	0,112	0,542		
		LR	0,631	0,619	0,498	0,729	0,379	0,311	0,256	0,143	0,369	0,028	1,107		
		SGD	0,615	0,615	0,528	0,683	0,430	0,153	0,153	0,072	0,222	-0,021	0,579		
	PDZ	Naive Bayes	0,567	0,551	0,488	0,568	0,428	0,348	0,258	0,197	0,276	0,137	1,216	3,421	
		kNN	0,436	0,447	0,448	0,447	0,449	0,003	-0,007	-0,025	-0,010	-0,041	-0,080		
		SVM	0,568	0,540	0,600	0,530	0,690	0,060	0,049	0,041	0,037	0,045	0,232		
		Tree	0,560	0,557	0,551	0,559	0,543	0,101	0,095	0,099	0,098	0,099	0,492		
		LR	0,603	0,618	0,479	0,750	0,352	0,283	0,255	0,124	0,390	0,001	1,053		
		SGD	0,608	0,608	0,501	0,690	0,393	0,146	0,146	0,045	0,229	-0,058	0,508		
	No Domain	Naive Bayes	0,219	0,293	0,291	0,292	0,291	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,433	0,454	0,473	0,457	0,490	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,508	0,491	0,559	0,493	0,645	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,459	0,462	0,452	0,461	0,444	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,320	0,363	0,355	0,360	0,351	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,462	0,462	0,456	0,461	0,451	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	LRR	Naive Bayes	0,488	0,507	0,406	0,511	0,337	0,178	0,146	0,044	0,150	-0,025	0,493	2,223	
		kNN	0,493	0,494	0,497	0,494	0,499	0,118	0,095	0,067	0,085	0,046	0,411		
		SVM	0,509	0,504	0,577	0,503	0,676	0,009	0,011	0,014	0,009	0,022	0,065		
		Tree	0,550	0,545	0,538	0,546	0,530	0,133	0,125	0,129	0,129	0,129	0,645		
		LR	0,569	0,548	0,378	0,606	0,275	0,182	0,124	-0,046	0,182	-0,149	0,293		
		SGD	0,558	0,558	0,469	0,588	0,390	0,108	0,108	0,020	0,138	-0,058	0,316		
	No Domain	Naive Bayes	0,310	0,361	0,362	0,361	0,362	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,375	0,399	0,430	0,409	0,453	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,500	0,493	0,563	0,494	0,654	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,417	0,420	0,409	0,417	0,401	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,387	0,424	0,424	0,424	0,424	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,450	0,450	0,449	0,450	0,448	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	WW	Naive Bayes	0,511	0,540	0,408	0,572	0,317	0,194	0,172	0,033	0,201	-0,063	0,537	3,176	
		kNN	0,498	0,494	0,514	0,494	0,535	0,129	0,093	0,084	0,084	0,083	0,473		
		SVM	0,564	0,532	0,598	0,524	0,697	0,057	0,035	0,030	0,027	0,035	0,184		
		Tree	0,566	0,565	0,561	0,565	0,557	0,147	0,146	0,156	0,150	0,161	0,760		
		LR	0,636	0,598	0,494	0,665	0,393	0,249	0,181	0,081	0,250	-0,017	0,744		
		SGD	0,582	0,581	0,478	0,634	0,384	0,136	0,135	0,049	0,191	-0,033	0,478		
	No Domain	Naive Bayes	0,317	0,368	0,375	0,371	0,380	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,369	0,401	0,430	0,410	0,452	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,507	0,497	0,568	0,497	0,662	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,419	0,419	0,405	0,415	0,396	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,387	0,417	0,413	0,415	0,410	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,446	0,446	0,429	0,443	0,417	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		

Subtitle:

AUC dif – Difference, within the same descriptor/classifier, between the AUC value of a domain and corresponding non-domain

ACC dif – Difference, within the same descriptor/classifier, between the ACC value of a domain and corresponding non-domain

F1 dif – Difference, within the same descriptor/classifier, between the F1 value of a domain and corresponding non-domain

PPV dif – Difference, within the same descriptor/classifier, between the PPV value of a domain and corresponding non-domain

TPR dif – Difference, within the same descriptor/classifier, between the TPR value of a domain and corresponding non-domain

∑ dif – Sum, within the same descriptor/domain, of all values of differences for the same classifier

∑ domain – Sum, within the same descriptor, of all values of sum of differences for the same domain

∑ descriptor – Sum of all values of sum of sum of differences for the same descriptor

N.A. – Not covered by dataset values

In black are the negative values - values of differences in which the values of non-domain performance were higher than the performance values of the corresponding domains

Descriptor	Domain	Classifier	AUC	ACC	F1	PPV	TPR	AUC dif	ACC dif	F1 dif	PPV dif	TPR dif	∑ dif	∑ domain	∑ descriptor
CTD	SH3	Naive Bayes	0,523	0,500	0,496	0,500	0,493	0,395	0,272	0,268	0,272	0,265	1,472	4,199	18,666
		kNN	0,417	0,438	0,460	0,443	0,479	-0,019	-0,014	-0,006	-0,011	0,000	-0,050		
		SVM	0,556	0,526	0,589	0,520	0,680	0,041	0,035	0,037	0,027	0,054	0,194		
		Tree	0,550	0,545	0,548	0,544	0,552	0,141	0,133	0,152	0,137	0,166	0,729		
		LR	0,520	0,516	0,463	0,520	0,417	0,339	0,261	0,217	0,271	0,174	1,262		
		SGD	0,538	0,538	0,519	0,542	0,498	0,124	0,124	0,114	0,131	0,099	0,592		
	SH2	Naive Bayes	0,544	0,515	0,511	0,516	0,507	0,416	0,287	0,283	0,288	0,279	1,553	5,150	
		kNN	0,483	0,491	0,506	0,492	0,522	0,047	0,039	0,040	0,038	0,043	0,207		
		SVM	0,577	0,526	0,605	0,519	0,727	0,062	0,035	0,053	0,026	0,101	0,277		
		Tree	0,586	0,583	0,576	0,585	0,568	0,177	0,171	0,180	0,178	0,182	0,888		
		LR	0,578	0,566	0,470	0,603	0,385	0,397	0,311	0,224	0,354	0,142	1,428		
		SGD	0,583	0,583	0,557	0,593	0,524	0,169	0,169	0,152	0,182	0,125	0,797		
	PDZ	Naive Bayes	0,491	0,465	0,476	0,466	0,487	0,363	0,237	0,248	0,238	0,259	1,345	4,222	
		kNN	0,412	0,435	0,447	0,438	0,456	-0,024	-0,017	-0,019	-0,016	-0,023	-0,099		
		SVM	0,568	0,533	0,602	0,524	0,707	0,053	0,042	0,050	0,031	0,081	0,257		
		Tree	0,550	0,541	0,549	0,540	0,558	0,141	0,129	0,153	0,133	0,172	0,728		
		LR	0,545	0,526	0,475	0,532	0,429	0,364	0,271	0,229	0,283	0,186	1,333		
		SGD	0,546	0,546	0,536	0,548	0,525	0,132	0,132	0,131	0,137	0,126	0,658		
	No Domain	Naive Bayes	0,128	0,228	0,228	0,228	0,228	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,436	0,452	0,466	0,454	0,479	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,515	0,491	0,552	0,493	0,626	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,409	0,412	0,396	0,407	0,386	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,181	0,255	0,246	0,249	0,243	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,414	0,414	0,405	0,411	0,399	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	LRR	Naive Bayes	0,398	0,412	0,404	0,410	0,398	0,235	0,173	0,175	0,178	0,172	0,933	2,183	
		kNN	0,398	0,414	0,422	0,416	0,428	-0,020	-0,013	-0,024	-0,016	-0,034	-0,107		
		SVM	0,519	0,515	0,564	0,512	0,628	0,016	0,011	-0,024	0,009	-0,079	-0,067		
		Tree	0,497	0,495	0,489	0,495	0,484	0,060	0,062	0,050	0,061	0,040	0,273		
		LR	0,462	0,455	0,424	0,450	0,401	0,254	0,181	0,158	0,181	0,138	0,912		
		SGD	0,496	0,496	0,493	0,496	0,489	0,044	0,044	0,051	0,046	0,054	0,239		
	No Domain	Naive Bayes	0,163	0,239	0,229	0,232	0,226	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,418	0,427	0,446	0,432	0,462	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,503	0,504	0,588	0,503	0,707	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,437	0,433	0,439	0,434	0,444	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,208	0,274	0,266	0,269	0,263	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,452	0,452	0,442	0,450	0,435	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	WW	Naive Bayes	0,440	0,434	0,436	0,434	0,439	0,296	0,203	0,195	0,197	0,195	1,086	2,912	
		kNN	0,384	0,406	0,401	0,405	0,398	-0,030	-0,019	-0,043	-0,025	-0,061	-0,178		
		SVM	0,543	0,517	0,587	0,513	0,686	0,018	0,034	0,039	0,026	0,058	0,175		
		Tree	0,520	0,519	0,510	0,519	0,501	0,074	0,080	0,070	0,080	0,060	0,364		
		LR	0,523	0,494	0,454	0,492	0,421	0,327	0,218	0,190	0,223	0,161	1,119		
		SGD	0,527	0,527	0,516	0,529	0,504	0,075	0,075	0,065	0,077	0,054	0,346		
	No Domain	Naive Bayes	0,144	0,231	0,241	0,237	0,244	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,414	0,425	0,444	0,430	0,459	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,525	0,483	0,548	0,487	0,628	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,446	0,439	0,440	0,439	0,441	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,196	0,276	0,264	0,269	0,260	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,452	0,452	0,451	0,452	0,450	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		

Subtitle:

AUC dif – Difference, within the same descriptor/classifier, between the AUC value of a domain and corresponding non-domain

ACC dif – Difference, within the same descriptor/classifier, between the ACC value of a domain and corresponding non-domain

F1 dif – Difference, within the same descriptor/classifier, between the F1 value of a domain and corresponding non-domain

PPV dif – Difference, within the same descriptor/classifier, between the PPV value of a domain and corresponding non-domain

TPR dif – Difference, within the same descriptor/classifier, between the TPR value of a domain and corresponding non-domain

∑ dif – Sum, within the same descriptor/domain, of all values of differences for the same classifier

∑ domain – Sum, within the same descriptor, of all values of sum of differences for the same domain

∑ descriptor – Sum of all values of sum of sum of differences for the same descriptor

N.A. – Not covered by dataset values

In black are the negative values - values of differences in which the values of non-domain performance were higher than the performance values of the corresponding domains

Descriptor	Domain	Classifier	AUC	ACC	F1	PPV	TPR	AUC dif	ACC dif	F1 dif	PPV dif	TPR dif	∑ dif	∑ domain	∑ descriptor
Moranauto	SH3	Naive Bayes	0,513	0,492	0,472	0,492	0,454	0,376	0,274	0,262	0,279	0,246	1,437	5,291	22,364
		kNN	0,548	0,546	0,456	0,569	0,380	0,277	0,224	0,093	0,226	-0,006	0,814		
		SVM	0,481	0,521	0,597	0,516	0,709	-0,080	0,054	0,065	0,042	0,104	0,185		
		Tree	0,572	0,562	0,562	0,562	0,561	0,148	0,136	0,134	0,135	0,132	0,685		
		LR	0,518	0,570	0,429	0,637	0,324	0,332	0,302	0,160	0,368	0,054	1,216		
		SGD	0,539	0,539	0,513	0,544	0,485	0,208	0,208	0,178	0,211	0,149	0,954		
	SH2	Naive Bayes	0,422	0,417	0,428	0,420	0,437	0,285	0,199	0,218	0,207	0,229	1,138	4,727	
		kNN	0,553	0,530	0,474	0,538	0,423	0,282	0,208	0,111	0,195	0,037	0,833		
		SVM	0,548	0,518	0,586	0,514	0,681	-0,013	0,051	0,054	0,040	0,076	0,208		
		Tree	0,560	0,558	0,549	0,561	0,538	0,136	0,132	0,121	0,134	0,109	0,632		
		LR	0,524	0,531	0,437	0,547	0,364	0,338	0,263	0,168	0,278	0,094	1,141		
		SGD	0,497	0,497	0,482	0,497	0,468	0,166	0,166	0,147	0,164	0,132	0,775		
	PDZ	Naive Bayes	0,459	0,445	0,432	0,442	0,423	0,322	0,227	0,222	0,229	0,215	1,215	5,383	
		kNN	0,566	0,534	0,489	0,541	0,446	0,295	0,212	0,126	0,198	0,060	0,891		
		SVM	0,561	0,521	0,594	0,516	0,699	0,000	0,054	0,062	0,042	0,094	0,252		
		Tree	0,581	0,573	0,566	0,575	0,558	0,157	0,147	0,138	0,148	0,129	0,719		
		LR	0,548	0,574	0,449	0,634	0,347	0,362	0,306	0,180	0,365	0,077	1,290		
		SGD	0,547	0,547	0,529	0,551	0,508	0,216	0,216	0,194	0,218	0,172	1,016		
	No Domain	Naive Bayes	0,137	0,218	0,210	0,213	0,208	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,271	0,322	0,363	0,343	0,386	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,561	0,467	0,532	0,474	0,605	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,424	0,426	0,428	0,427	0,429	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,186	0,268	0,269	0,269	0,270	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,331	0,331	0,335	0,333	0,336	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	LRR	Naive Bayes	0,316	0,344	0,343	0,343	0,343	0,242	0,197	0,189	0,191	0,188	1,007	2,726	
		kNN	0,451	0,422	0,428	0,424	0,432	0,105	0,044	0,050	0,046	0,053	0,298		
		SVM	0,510	0,497	0,578	0,498	0,689	-0,013	0,007	0,019	0,006	0,043	0,062		
		Tree	0,486	0,484	0,481	0,484	0,478	0,054	0,052	0,075	0,059	0,090	0,330		
		LR	0,416	0,425	0,356	0,405	0,319	0,285	0,205	0,121	0,175	0,080	0,866		
		SGD	0,438	0,438	0,436	0,437	0,435	0,035	0,035	0,031	0,033	0,029	0,163		
	No Domain	Naive Bayes	0,074	0,147	0,154	0,152	0,155	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,346	0,378	0,378	0,378	0,379	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,523	0,490	0,559	0,492	0,646	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,432	0,432	0,406	0,425	0,388	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,131	0,220	0,235	0,230	0,239	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,403	0,403	0,405	0,404	0,406	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	WW	Naive Bayes	0,350	0,369	0,370	0,369	0,371	0,291	0,240	0,232	0,232	0,231	1,226	4,237	
		kNN	0,525	0,489	0,443	0,486	0,406	0,188	0,124	0,142	0,151	0,133	0,738		
		SVM	0,515	0,510	0,570	0,508	0,651	-0,019	0,036	0,030	0,028	0,033	0,108		
		Tree	0,522	0,516	0,513	0,516	0,509	0,101	0,106	0,116	0,110	0,121	0,554		
		LR	0,496	0,471	0,403	0,462	0,358	0,380	0,272	0,196	0,257	0,149	1,254		
		SGD	0,473	0,472	0,467	0,472	0,462	0,077	0,076	0,068	0,075	0,061	0,357		
	No Domain	Naive Bayes	0,059	0,129	0,138	0,137	0,140	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,337	0,365	0,301	0,335	0,273	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,534	0,474	0,540	0,480	0,618	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,421	0,410	0,397	0,406	0,388	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,116	0,199	0,207	0,205	0,209	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,396	0,396	0,399	0,397	0,401	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		

<u>Subtitle:</u>

AUC dif – Difference, within the same descriptor/classifier, between the AUC value of a domain and corresponding non-domain

ACC dif – Difference, within the same descriptor/classifier, between the ACC value of a domain and corresponding non-domain

F1 dif – Difference, within the same descriptor/classifier, between the F1 value of a domain and corresponding non-domain

PPV dif – Difference, within the same descriptor/classifier, between the PPV value of a domain and corresponding non-domain

TPR dif – Difference, within the same descriptor/classifier, between the TPR value of a domain and corresponding non-domain

∑ dif – Sum, within the same descriptor/domain, of all values of differences for the same classifier

∑ domain – Sum, within the same descriptor, of all values of sum of differences for the same domain

∑ descriptor – Sum of all values of sum of sum of differences for the same descriptor

N.A. – Not covered by dataset values

In black are the negative values - values of differences in which the values of non-domain performance were higher than the performance values of the corresponding domains

Descriptor	Domain	Classifier	AUC	ACC	F1	PPV	TPR	AUC dif	ACC dif	F1 dif	PPV dif	TPR dif	∑ dif	∑ domain	∑ descriptor
SOCN	SH3	Naive Bayes	0,568	0,542	0,553	0,540	0,566	0,129	0,091	0,090	0,087	0,093	0,490	2,786	11,068
		kNN	0,517	0,512	0,515	0,512	0,518	0,035	0,033	0,040	0,033	0,047	0,188		
		SVM	0,578	0,525	0,589	0,519	0,680	0,083	0,029	0,015	0,021	0,002	0,150		
		Tree	0,515	0,517	0,516	0,517	0,514	0,143	0,135	0,141	0,138	0,144	0,701		
		LR	0,588	0,586	0,469	0,653	0,366	0,243	0,199	0,080	0,265	-0,024	0,763		
		SGD	0,620	0,620	0,550	0,674	0,464	0,124	0,124	0,070	0,178	-0,002	0,494		
	SH2	Naive Bayes	0,582	0,540	0,543	0,540	0,546	0,143	0,089	0,080	0,087	0,073	0,472	3,106	
		kNN	0,516	0,511	0,528	0,510	0,546	0,034	0,032	0,053	0,031	0,075	0,225		
		SVM	0,561	0,514	0,524	0,513	0,536	0,066	0,018	-0,050	0,015	-0,142	-0,093		
		Tree	0,575	0,575	0,562	0,579	0,546	0,203	0,193	0,187	0,200	0,176	0,959		
		LR	0,635	0,627	0,489	0,776	0,357	0,290	0,240	0,100	0,388	-0,033	0,985		
		SGD	0,626	0,626	0,572	0,668	0,500	0,130	0,130	0,092	0,172	0,034	0,558		
	PDZ	Naive Bayes	0,551	0,531	0,553	0,528	0,581	0,112	0,080	0,090	0,075	0,108	0,465	2,651	
		kNN	0,520	0,511	0,515	0,511	0,519	0,038	0,032	0,040	0,032	0,048	0,190		
		SVM	0,546	0,515	0,550	0,513	0,593	0,051	0,019	-0,024	0,015	-0,085	-0,024		
		Tree	0,535	0,536	0,534	0,536	0,532	0,163	0,154	0,159	0,157	0,162	0,795		
		LR	0,584	0,592	0,470	0,671	0,362	0,239	0,205	0,081	0,283	-0,028	0,780		
		SGD	0,613	0,613	0,536	0,669	0,448	0,117	0,117	0,056	0,173	-0,018	0,445		
	No Domain	Naive Bayes	0,439	0,451	0,463	0,453	0,473	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,482	0,479	0,475	0,479	0,471	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,495	0,496	0,574	0,498	0,678	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,372	0,382	0,375	0,379	0,370	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,345	0,387	0,389	0,388	0,390	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,496	0,496	0,480	0,496	0,466	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	LRR	Naive Bayes	0,499	0,496	0,509	0,497	0,523	0,076	0,051	0,076	0,054	0,099	0,356	0,353	
		kNN	0,476	0,474	0,481	0,475	0,487	-0,035	-0,027	-0,033	-0,026	-0,042	-0,163		
		SVM	0,502	0,502	0,484	0,502	0,466	0,012	0,004	-0,145	0,003	-0,386	-0,512		
		Tree	0,481	0,482	0,477	0,482	0,472	0,040	0,038	0,041	0,040	0,042	0,201		
		LR	0,499	0,501	0,399	0,501	0,331	0,169	0,121	0,007	0,116	-0,068	0,345		
		SGD	0,563	0,563	0,476	0,594	0,397	0,062	0,062	-0,012	0,093	-0,079	0,126		
	No Domain	Naive Bayes	0,423	0,445	0,433	0,443	0,424	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,511	0,501	0,514	0,501	0,529	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,490	0,498	0,629	0,499	0,852	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,441	0,444	0,436	0,442	0,430	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,330	0,380	0,392	0,385	0,399	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,501	0,501	0,488	0,501	0,476	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	WW	Naive Bayes	0,559	0,531	0,547	0,529	0,566	0,136	0,093	0,111	0,091	0,133	0,564	2,172	
		kNN	0,528	0,519	0,531	0,519	0,544	0,022	0,025	0,028	0,025	0,032	0,132		
		SVM	0,543	0,501	0,576	0,501	0,679	0,043	0,004	-0,001	0,003	-0,006	0,043		
		Tree	0,545	0,537	0,527	0,539	0,516	0,104	0,093	0,094	0,097	0,092	0,480		
		LR	0,579	0,577	0,417	0,671	0,302	0,273	0,217	0,074	0,319	-0,033	0,850		
		SGD	0,578	0,578	0,512	0,606	0,444	0,081	0,081	-0,026	0,109	-0,142	0,103		
	No Domain	Naive Bayes	0,423	0,438	0,436	0,438	0,433	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,506	0,494	0,503	0,494	0,512	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,500	0,497	0,577	0,498	0,685	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,441	0,444	0,433	0,442	0,424	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,306	0,360	0,343	0,352	0,335	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,497	0,497	0,538	0,497	0,586	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		

Subtitle:

AUC dif – Difference, within the same descriptor/classifier, between the AUC value of a domain and corresponding non-domain

ACC dif – Difference, within the same descriptor/classifier, between the ACC value of a domain and corresponding non-domain

F1 dif – Difference, within the same descriptor/classifier, between the F1 value of a domain and corresponding non-domain

PPV dif – Difference, within the same descriptor/classifier, between the PPV value of a domain and corresponding non-domain

TPR dif – Difference, within the same descriptor/classifier, between the TPR value of a domain and corresponding non-domain

∑ dif – Sum, within the same descriptor/domain, of all values of differences for the same classifier

∑ domain – Sum, within the same descriptor, of all values of sum of differences for the same domain

∑ descriptor – Sum of all values of sum of sum of differences for the same descriptor

N.A. – Not covered by dataset values

In black are the negative values - values of differences in which the values of non-domain performance were higher than the performance values of the corresponding domains

Descriptor	Domain	Classifier	AUC	ACC	F1	PPV	TPR	AUC dif	ACC dif	F1 dif	PPV dif	TPR dif	∑ dif	∑ domain	∑ descriptor
QSO	SH3	Naive Bayes	0,554	0,522	0,528	0,521	0,535	0,252	0,161	0,175	0,164	0,187	0,939	5,940	23,253
		kNN	0,537	0,514	0,475	0,516	0,440	0,248	0,219	0,193	0,229	0,163	1,052		
		SVM	0,567	0,519	0,626	0,512	0,804	0,067	0,023	0,127	0,015	0,303	0,535		
		Tree	0,539	0,538	0,534	0,538	0,530	0,371	0,267	0,459	0,436	0,471	2,004		
		LR	0,604	0,650	0,462	1,000	0,300	0,221	0,227	0,062	0,584	-0,085	1,009		
		SGD	0,575	0,575	0,542	0,587	0,504	0,086	0,086	0,075	0,098	0,056	0,401		
	SH2	Naive Bayes	0,563	0,524	0,513	0,526	0,500	0,261	0,163	0,160	0,169	0,152	0,905	6,240	
		kNN	0,591	0,555	0,504	0,570	0,451	0,302	0,260	0,222	0,283	0,174	1,241		
		SVM	0,561	0,528	0,605	0,521	0,722	0,061	0,032	0,106	0,024	0,221	0,444		
		Tree	0,591	0,583	0,580	0,584	0,576	0,423	0,312	0,505	0,482	0,517	2,239		
		LR	0,638	0,644	0,447	1,000	0,288	0,255	0,221	0,047	0,584	-0,097	1,010		
		SGD	0,574	0,574	0,543	0,586	0,506	0,085	0,085	0,076	0,097	0,058	0,401		
	PDZ	Naive Bayes	0,524	0,484	0,497	0,485	0,509	0,222	0,123	0,144	0,128	0,161	0,778	5,857	
		kNN	0,557	0,521	0,492	0,524	0,464	0,268	0,226	0,210	0,237	0,187	1,128		
		SVM	0,594	0,532	0,612	0,523	0,737	0,094	0,036	0,113	0,026	0,236	0,505		
		Tree	0,545	0,543	0,545	0,542	0,548	0,377	0,272	0,470	0,440	0,489	2,048		
		LR	0,600	0,650	0,462	1,000	0,300	0,217	0,227	0,062	0,584	-0,085	1,005		
		SGD	0,571	0,571	0,543	0,580	0,510	0,082	0,082	0,076	0,091	0,062	0,393		
	No Domain	Naive Bayes	0,302	0,361	0,353	0,357	0,348	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,289	0,295	0,282	0,287	0,277	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,500	0,496	0,499	0,497	0,501	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,168	0,271	0,075	0,102	0,059	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,383	0,423	0,400	0,416	0,385	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,489	0,489	0,467	0,489	0,448	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	LRR	Naive Bayes	0,426	0,429	0,430	0,429	0,430	0,178	0,107	0,104	0,105	0,101	0,595	1,792	
		kNN	0,486	0,461	0,455	0,460	0,450	0,113	0,059	0,038	0,053	0,023	0,286		
		SVM	0,535	0,512	0,587	0,509	0,695	0,031	0,017	-0,017	0,012	-0,074	-0,031		
		Tree	0,503	0,504	0,497	0,504	0,491	0,056	0,058	0,049	0,057	0,042	0,262		
		LR	0,529	0,583	0,283	1,000	0,165	0,167	0,182	-0,122	0,598	-0,243	0,582		
		SGD	0,515	0,515	0,497	0,516	0,479	0,039	0,039	0,007	0,039	-0,026	0,098		
	No Domain	Naive Bayes	0,248	0,322	0,326	0,324	0,329	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,373	0,402	0,417	0,407	0,427	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,504	0,495	0,604	0,497	0,769	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,447	0,446	0,448	0,447	0,449	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,362	0,401	0,405	0,402	0,408	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,476	0,476	0,490	0,477	0,505	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	WW	Naive Bayes	0,487	0,465	0,483	0,467	0,501	0,266	0,160	0,164	0,154	0,175	0,919	3,424	
		kNN	0,539	0,498	0,471	0,498	0,446	0,169	0,099	0,051	0,092	0,011	0,422		
		SVM	0,545	0,516	0,613	0,511	0,767	0,051	0,003	0,047	0,001	0,131	0,233		
		Tree	0,534	0,529	0,522	0,530	0,514	0,097	0,096	0,101	0,100	0,101	0,495		
		LR	0,581	0,616	0,376	1,000	0,231	0,239	0,231	-0,011	0,615	-0,157	0,917		
		SGD	0,543	0,542	0,522	0,546	0,500	0,086	0,085	0,089	0,093	0,085	0,438		
	No Domain	Naive Bayes	0,221	0,305	0,319	0,313	0,326	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,370	0,399	0,420	0,406	0,435	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,494	0,513	0,566	0,510	0,636	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,437	0,433	0,421	0,430	0,413	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,342	0,385	0,387	0,385	0,388	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,457	0,457	0,433	0,453	0,415	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		

Subtitle:

AUC dif – Difference, within the same descriptor/classifier, between the AUC value of a domain and corresponding non-domain

ACC dif – Difference, within the same descriptor/classifier, between the ACC value of a domain and corresponding non-domain

F1 dif – Difference, within the same descriptor/classifier, between the F1 value of a domain and corresponding non-domain

PPV dif – Difference, within the same descriptor/classifier, between the PPV value of a domain and corresponding non-domain

TPR dif – Difference, within the same descriptor/classifier, between the TPR value of a domain and corresponding non-domain

∑ dif – Sum, within the same descriptor/domain, of all values of differences for the same classifier

∑ domain – Sum, within the same descriptor, of all values of sum of differences for the same domain

∑ descriptor – Sum of all values of sum of sum of differences for the same descriptor

N.A. – Not covered by dataset values

In black are the negative values - values of differences in which the values of non-domain performance were higher than the performance values of the corresponding domains

Descriptor	Domain	Classifier	AUC	ACC	F1	PPV	TPR	AUC dif	ACC dif	F1 dif	PPV dif	TPR dif	∑ dif	∑ domain	∑ descriptor
CT	SH3	Naive Bayes	0,484	0,490	0,487	0,490	0,484	0,363	0,281	0,304	0,301	0,307	1,556	4,049	16,996
		kNN	0,384	0,411	0,447	0,421	0,476	-0,011	-0,008	-0,010	-0,008	-0,014	-0,051		
		SVM	0,523	0,513	0,597	0,509	0,720	-0,001	0,017	0,039	0,012	0,084	0,151		
		Tree	0,540	0,537	0,538	0,537	0,538	0,087	0,095	0,108	0,097	0,118	0,505		
		LR	0,437	0,430	0,441	0,432	0,449	0,314	0,241	0,255	0,245	0,264	1,319		
		SGD	0,471	0,471	0,500	0,474	0,529	0,092	0,092	0,127	0,098	0,160	0,569		
	SH2	Naive Bayes	0,491	0,491	0,470	0,491	0,451	0,370	0,282	0,287	0,302	0,274	1,515	4,889	
		kNN	0,470	0,482	0,488	0,482	0,494	0,075	0,063	0,031	0,053	0,004	0,226		
		SVM	0,550	0,524	0,598	0,518	0,708	0,026	0,028	0,040	0,021	0,072	0,187		
		Tree	0,601	0,597	0,587	0,602	0,573	0,148	0,155	0,157	0,162	0,153	0,775		
		LR	0,519	0,470	0,445	0,467	0,424	0,396	0,281	0,259	0,280	0,239	1,455		
		SGD	0,538	0,538	0,509	0,544	0,478	0,159	0,159	0,136	0,168	0,109	0,731		
	PDZ	Naive Bayes	0,406	0,431	0,438	0,433	0,443	0,285	0,222	0,255	0,244	0,266	1,272	3,595	
		kNN	0,373	0,399	0,401	0,400	0,401	-0,022	-0,020	-0,056	-0,029	-0,089	-0,216		
		SVM	0,511	0,512	0,583	0,509	0,681	-0,013	0,016	0,025	0,012	0,045	0,085		
		Tree	0,542	0,539	0,538	0,539	0,538	0,089	0,097	0,108	0,099	0,118	0,511		
		LR	0,447	0,439	0,442	0,439	0,444	0,324	0,250	0,256	0,252	0,259	1,341		
		SGD	0,489	0,489	0,500	0,489	0,511	0,110	0,110	0,127	0,113	0,142	0,602		
	No Domain	Naive Bayes	0,121	0,209	0,183	0,189	0,177	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,395	0,419	0,457	0,429	0,490	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,524	0,496	0,558	0,497	0,636	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,453	0,442	0,430	0,440	0,420	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,123	0,189	0,186	0,187	0,185	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,379	0,379	0,373	0,376	0,369	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	LRR	Naive Bayes	0,347	0,377	0,394	0,384	0,405	0,174	0,113	0,111	0,108	0,115	0,621	1,676	
		kNN	0,367	0,391	0,394	0,392	0,395	-0,023	-0,021	-0,047	-0,028	-0,069	-0,188		
		SVM	0,502	0,494	0,597	0,496	0,748	-0,013	0,001	0,037	0,001	0,103	0,129		
		Tree	0,497	0,498	0,493	0,498	0,488	0,047	0,053	0,051	0,054	0,048	0,253		
		LR	0,342	0,367	0,368	0,367	0,369	0,217	0,166	0,169	0,167	0,170	0,889		
		SGD	0,417	0,417	0,420	0,418	0,421	0,006	0,006	-0,013	0,001	-0,028	-0,028		
	No Domain	Naive Bayes	0,173	0,264	0,283	0,276	0,290	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,390	0,412	0,441	0,420	0,464	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,515	0,493	0,560	0,495	0,645	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,450	0,445	0,442	0,444	0,440	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,125	0,201	0,199	0,200	0,199	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,411	0,411	0,433	0,417	0,449	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
	WW	Naive Bayes	0,379	0,400	0,427	0,409	0,447	0,242	0,179	0,203	0,186	0,221	1,031	2,787	
		kNN	0,387	0,406	0,417	0,409	0,425	-0,009	-0,017	-0,031	-0,020	-0,043	-0,120		
		SVM	0,521	0,512	0,581	0,509	0,678	0,001	0,017	-0,030	0,012	-0,116	-0,116		
		Tree	0,550	0,544	0,539	0,545	0,533	0,105	0,098	0,114	0,104	0,123	0,544		
		LR	0,410	0,416	0,416	0,416	0,416	0,284	0,220	0,228	0,226	0,230	1,188		
		SGD	0,462	0,462	0,450	0,461	0,441	0,060	0,060	0,046	0,058	0,036	0,260		
	No Domain	Naive Bayes	0,137	0,221	0,224	0,223	0,226	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	
		kNN	0,396	0,423	0,448	0,429	0,468	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SVM	0,520	0,495	0,611	0,497	0,794	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		Tree	0,445	0,446	0,425	0,441	0,410	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		LR	0,126	0,196	0,188	0,190	0,186	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		
		SGD	0,402	0,402	0,404	0,403	0,405	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.		

Subtitle:

AUC dif – Difference, within the same descriptor/classifier, between the AUC value of a domain and corresponding non-domain

ACC dif – Difference, within the same descriptor/classifier, between the ACC value of a domain and corresponding non-domain

F1 dif – Difference, within the same descriptor/classifier, between the F1 value of a domain and corresponding non-domain

PPV dif – Difference, within the same descriptor/classifier, between the PPV value of a domain and corresponding non-domain

TPR dif – Difference, within the same descriptor/classifier, between the TPR value of a domain and corresponding non-domain

∑ dif – Sum, within the same descriptor/domain, of all values of differences for the same classifier

∑ domain – Sum, within the same descriptor, of all values of sum of differences for the same domain

∑ descriptor – Sum of all values of sum of sum of differences for the same descriptor

N.A. – Not covered by dataset values

In black are the negative values - values of differences in which the values of non-domain performance were higher than the performance values of the corresponding domains

Annex III – 2nd strategy evaluation code

```
import numpy as np
import os
import csv
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.naive_bayes import GaussianNB
from sklearn import tree
from sklearn.neural_network import MLPClassifier
from sklearn import neighbors
from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import roc_auc_score

filename = "Generic.csv"
location = r"C:\Users\Joao Castanheira\Downloads\Tese"
os.chdir(location)
filename_target = [1,1,1,1,… ,0,0,0,0]
new_data_frame = pd.read_csv(filename, header = None, sep=";")
X_train, X_test, y_train, y_test = train_test_split(new_data_frame, filename_target,
test_size=0.4, random_state=0)
```

```
clf = svm.SVC(kernel='rbf', verbose=True, max_iter=5000, C=1).fit(X_train, y_train)  #1
scores = cross_val_score(clf, new_data_frame, filename_target, cv=5)                 #2
y_scores = clf.predict(X_test) # Cálculo do y_scores ou y_pred                        #3
```

A

```
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
print("f1_score:", f1_score(y_test, y_scores, average='binary'))
print("Precision:", precision_score(y_test, y_scores, average='binary'))
print("Recall:", recall_score(y_test, y_scores, average='binary'))
print("AUC:", roc_auc_score(y_test, y_scores))
```

```
gnb = GaussianNB().fit(X_train, y_train)                                    #1
scores = cross_val_score(gnb, new_data_frame, filename_target, cv=5)        #2
y_scores = gnb.predict(X_test)                                              #3
```

B

```
clf = tree.DecisionTreeClassifier().fit(X_train, y_train)                   #1
scores = cross_val_score(clf, new_data_frame, filename_target, cv=5)        #2
y_scores = clf.predict(X_test)                                              #3
```

C

```
clf = MLPClassifier().fit(X_train, y_train) #neural network                 #1
scores = cross_val_score(clf, new_data_frame, filename_target, cv=5)        #2
y_scores = clf.predict(X_test)                                              #3
```

D

```
nbrs = neighbors.KNeighborsClassifier(n_neighbors=6).fit(X_train, y_train)#1
scores = cross_val_score(nbrs, new_data_frame, filename_target, cv=5)       #2
y_scores = nbrs.predict(X_test)                                             #3
```

E

```
clf = SGDClassifier(loss="log").fit(X_train, y_train)                       #1
scores = cross_val_score(clf, new_data_frame, filename_target, cv=5)        #2
y_scores = clf.predict(X_test)                                              #3
```

F

Subtitle:

The "filename" corresponds to the datasets, also using the dataset GenericDomain.csv;

The "filename_target" holds repeats of characters 1 and 0, corresponding to the existence or not of interactions, with different numbers of replicates of the Generic.csv dataset for GenericDomain.csv;

Lines # 1, # 2 and # 3 of group A were replaced by the corresponding lines of groups B, C and D, E and F.

Annex IV – Add the features of the 3rd strategy with the ID's

```
using System;
using System.Collections.Generic;
using System.IO;
namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Dictionary<string, string> dict = new Dictionary<string, string>();
            Console.WriteLine("Opening data.csv");
            int i = 0;
            using (var fileStream = File.OpenText(@"C:\tmp\joao-castanheira\data.csv"))
            {
                do
                {
                    var fileLine = fileStream.ReadLine();
                    //Console.WriteLine(fileLine.Substring(0, 30));
                    var key = fileLine.Substring(0, fileLine.IndexOf(","));
                    var rest = fileLine.Substring(fileLine.IndexOf(",") + 1);
                    //Console.WriteLine(key);
                    //Console.WriteLine(rest);
                    if (!dict.ContainsKey(key))
                    {
                        dict.Add(key, rest);
                    }
                    if (i%10000 == 0)
                    {
                        Console.WriteLine(i);
                    }
                    i++;
                } while (!fileStream.EndOfStream);
            }
            Console.WriteLine(i);
            Console.WriteLine("Data.csv opened");
```

```
/*
int g = 0;
var list = new List<string>();
var genericErrors = new List<string>();
using (var fileStream = File.OpenText(@"C:\tmp\joao-castanheira\Generic.csv"))
{
  do
  {
    //if (g == 1)
    //{
    //    break;
    //}
    var fileLine = fileStream.ReadLine();
    //Console.WriteLine(fileLine);
    var key1 = fileLine.Substring(0, fileLine.IndexOf(","));
    var key2 = fileLine.Substring(fileLine.IndexOf(",") + 1);
    //Console.WriteLine(key1);
    //Console.WriteLine(key2);
    if (dict.ContainsKey(key1))
    {
      if (dict.ContainsKey(key2))
      {
        string final = string.Format("{0},{1}", dict[key1], dict[key2]);
        //Console.WriteLine(final);
        list.Add(final);
      }
      else
      {
        var error = string.Format("Key2 does not exist - {0}:{1}", g, key2);
        genericErrors.Add(error);
        Console.WriteLine(error);
      }
    }
    else
    {
      var error = string.Format("Key1 does not exist - {0}:{1}", g, key1);
      genericErrors.Add(error);
```

```
                        Console.WriteLine(error);
                    }
                    if (g % 10000 == 0)
                    {
                        Console.WriteLine(g);
                    }
                    g++;
                } while (!fileStream.EndOfStream);
            }
            Console.WriteLine(g);
            Console.WriteLine("Generic.csv processed");
            Console.WriteLine("Saving GenericErrors.txt");
            File.WriteAllLines("GenericErrors.txt", genericErrors);
            Console.WriteLine("Saving ProcessedGeneric.csv");
            File.WriteAllLines("ProcessedGeneric.csv", list);
            */
            int gd = 0;
            var listgd = new List<string>();
            var genericDomainsErrors = new List<string>();
            using (var fileStream = File.OpenText(@"C:\tmp\joao-
castanheira\GenericDomains.csv"))
            {
                do
                {
                    //if (gd == 1)
                    //{
                    //    break;
                    //}
                    var fileLine = fileStream.ReadLine();
                    //Console.WriteLine(fileLine);
                    var key1 = fileLine.Substring(0, fileLine.IndexOf(","));
                    var key2 = fileLine.Substring(fileLine.IndexOf(",") + 1);
                    //Console.WriteLine(key1);
                    //Console.WriteLine(key2);
                    if (dict.ContainsKey(key1))
                    {
                        if (dict.ContainsKey(key2))
                        {
                            string final = string.Format("{0},{1}", dict[key1], dict[key2]);
                            //Console.WriteLine(final);
                            listgd.Add(final);
                        }
                        else
                        {
                            var error = string.Format("Key2 does not exist - {0}:{1}", gd, key2);
                            genericDomainsErrors.Add(error);
                            Console.WriteLine(error);
                        }
                    }
                    else
                    {
                        var error = string.Format("Key1 does not exist - {0}:{1}", gd, key1);
                        genericDomainsErrors.Add(error);
                        Console.WriteLine(error);
                    }
                    if (gd % 10000 == 0)
                    {
                        Console.WriteLine(gd);
                    }
                    gd++;
                } while (!fileStream.EndOfStream);
            }
            Console.WriteLine(gd);
            Console.WriteLine("GenericDomains.csv processed");
            Console.WriteLine("Saving GenericDomainsErrors.txt");
            File.WriteAllLines("GenericDomainsErrors.txt", genericDomainsErrors);
            Console.WriteLine("Saving ProcessedGenericDomains.csv");
            File.WriteAllLines("ProcessedGenericDomains.csv", listgd);
            Console.WriteLine("Done!");
            Console.ReadKey();
        }
    }
}
```

Annex V – 3rd strategy evalution code

```
import numpy as np
import os
import csv
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.naive_bayes import GaussianNB
from sklearn import tree
from sklearn.neural_network import MLPClassifier
from sklearn import neighbors
from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import average_precision_score
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import roc_auc_score
from sklearn import metrics

filename = "Generic.csv"
class_filename = "class_GD.txt"
target_class = np.genfromtxt(class_filename, delimiter= ",")
target_class = target_class.transpose()
print(1)
location = r"/home/student/Desktop/Castanheira"
os.chdir(location)
new_data_frame = np.genfromtxt(filename, delimiter=",")
X_train, X_test, y_train, y_test = train_test_split(new_data_frame, target_class, test_size=0.4,
random_state=0)
print(3)
clf = MLPClassifier().fit(X_train, y_train) #neural network          #1
print(4)
scores = cross_val_score(clf, new_data_frame, target_class, cv=5)     #2
print(5)
y_scores = clf.predict(X_test)                                        #3
```

A

```
print("f1_score:", f1_score(y_test, y_scores, average='binary'))
print("Precision:", precision_score(y_test, y_scores, average='binary'))
print("Recall:", recall_score(y_test, y_scores, average='binary'))
print("AUC:", roc_auc_score(y_test, y_scores))
```

gnb = GaussianNB().fit(X_train, y_train)	#1	B
scores = cross_val_score(gnb, new_data_frame, filename_target, cv=5)	#2	
y_scores = gnb.predict(X_test)	#3	

clf = tree.DecisionTreeClassifier().fit(X_train, y_train)	#1	C
scores = cross_val_score(clf, new_data_frame, filename_target, cv=5)	#2	
y_scores = clf.predict(X_test)	#3	

clf = MLPClassifier().fit(X_train, y_train) #neural network	#1	D
scores = cross_val_score(clf, new_data_frame, filename_target, cv=5)	#2	
y_scores = clf.predict(X_test)	#3	

Subtitle:

The "filename" corresponds to the datasets, also using the dataset GenericDomain.csv;

The "class_filename" is the document with the 1 and 0 corresponding to the existence or not of interactions being used the document with the amount of 1 and 0 corresponding to the datasets Generic.csv or GenericDomain.csv;

Lines # 1, # 2 and # 3 of group A were replaced by the corresponding lines of groups B, C and D.