Study and comparison of data reduction technics for Feature Subset Selection in bioinformatics

Fabien Richard

Summer Internship
at Graphic Era University, Dehradun INDIA
Guided by Dr Bhasker Pant

Abstract. This report aims to present the research and work I conducted at Graphic Era University during my summer internship. The subject of my internship was to study and compare existing algorithms for Feature Subset Selection and to apply them to a problem in bioinformatics. Unfortunately, the organization of my internship did not allow me to conduct the application properly. This is why, the results presented at the end of this report, should be considered as an example of application, but not as a proof of efficiency of the Feature Subset Selection technics detailed in this report.

Acknowledgements. I would like to thank Dr Bhasker Pant for guiding me during this internship. I would also like to thank Dr. Durgaprasad Gangodkar, the Dean of International Affairs at Graphic Era University and all the faculty members of GEU and Polytech Nantes for making this internship possible.

Introduction

In the first section of this report, I will present the context of this study. Then, I will present in section 2, my studies on how Principal Component Analysis (PCA), and algorithms based on Rough Set theory, during preprocessing, can improve the performances of classification algorithms. Then, in section 3, I will present the results of the application of PCA and algorithms based on Rough Set theory, on Protein Interaction Prediction.

Table of content

I. Context

When a dataset has too many attributes (over a thousand), the quality of the results of classification algorithms on this dataset can be affected. A way to prevent this, is to perform a Feature Subset Selection (FSS). By keeping only the attributes that contribute the most to the class to predict, one can improve the results of the classification.

A field of study of Dr Bhasker Pant, the professor who guided me during this internship, is bioinformatics and is called Protein Interaction Prediction. It consists of predicting whether two given proteins have a probability of interacting with each other. Some universities, like UCLA or the University of Singapore, have given access to their databases to searchers and data scientists. However, in these databases, each protein is described by over 1000 attributes like its mass, length, spinning properties, etc. Using a classification algorithm with as many attributes would not give good results and would require a lot of processing time and memory. The objective of my internship was to study different technics of Feature Subset Selection to build a model for Protein Interaction Prediction. Unfortunately, I did not have the time to conduct the application properly, so this report will mainly focus on the comparison of two technics.

II. Methodology

To improve the performances of classification algorithms on data described by many attributes (such as proteins), I have studied what are the benefits of using PCA and Rough Set theory to select the most relevant attributes, before performing the classification.

To compare the results of PCA and Rough Set theory, the same dataset will be used. It was extracted from the RSES software DATA folder. It is composed of 13 attributes and 1 binary decision class, and contains 270 instances.

1. PCA

General procedure. Principal Component Analysis can be used to reduce the number of variables in a dataset. To generate the components, it uses an orthogonal transformation. The first component has the largest possible variance. We then use the computation of

eigenvalues to determine how much information is explained by each component and rule out unwanted attributes.

To illustrate the importance of PCA during preprocessing, I first performed a classification on the raw dataset. I then proceeded to a PCA to rule out the attributes which may pollute the classification. Finally, I performed a second classification using the exact same process (same choice of training/testing split, same classification algorithm). The comparison of time, accuracy, F-Measure and ROC-area allowed me to conclude on whether a PCA during preprocessing can improve the performances of a classification algorithm.

Three softwares were used to perform these measures : RSES to discretize the dataset, Tanagra to perform the PCA and Weka to perform the classification.

Experiment and results.

The dataset used for the experiment is named 'Heart'. It has 13 attributes and 1 binary decision class, and contains 270 instances.

The original dataset (as extracted from RSES data folder) can be found at the following url : https://drive.google.com/file/d/0B4Y4vCvFO0Wic19wZy00YVU4YTg

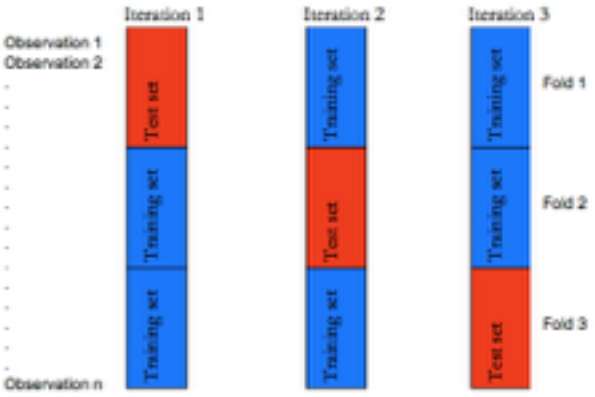
Table 1. Extract of dataset used for experiment

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	d
O:1	700	10	40	130	322	0	20	109	0	24	20	30	30	2
O:2	670	0	30	115	564	0	20	160	0	16	20	0	70	1
O:3	570	10	20	124	261	0	0	141	0	3	10	0	70	2
O:4	640	10	40	128	263	0	0	105	10	2	20	10	70	1
O:5	740	0	20	120	269	0	20	121	10	2	10	10	30	1
O:6	650	10	40	120	177	0	0	140	0	4	10	0	70	1

I used this dataset and applied a classification algorithm on it with Weka without making any change to the dataset. I then performed a PCA to rule out attributes which could affect the results of the classification. Finally, I applied the same classification algorithm but without the attributes ruled out with the PCA.

The classification algorithm I chose is named 'Logistic'. It is useful for building and using a multinomial logistic regression model. The ridge estimator used with this classifier is the default one : 1.0E-8. To train and test, I chose to use a cross-validation with 10 folds because the dataset had only 270 instances.

Evaluation. A classifier is best evaluated by applying it to a set of unseen observations (i.e. a test set). k-fold : If few observations are available, which is commonly the case, cross validation may get the most out of the data in terms of performance estimation. k-fold cross validation refers to dividing the examples into k equally sized subsets and using one subset for testing and the rest for training.

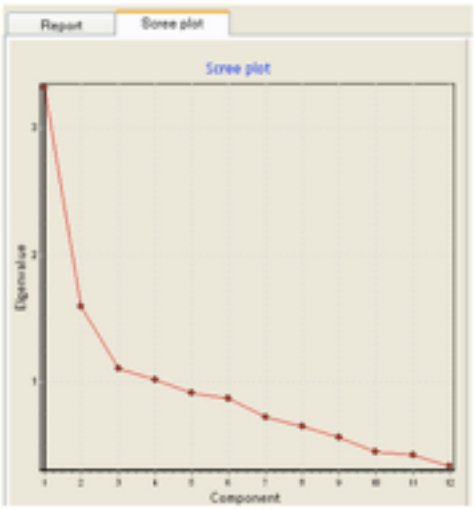


source : http://www.trhvidsten.com/docs/ROSETTATutorial.pdf

Another way of evaluating a model is to divide the set into two : one subset for training and one subset for testing. It is useful when a large amount of data is available.

---

1. Scree Plot to determine which axis explain the information



Here, we can notice that the eigenvalues drop really quickly until component 2. After this component, their is much less information explained by each axis (3-12). Thus, we decide to keep only the attributes explained by component 1 and 2 as they seem to explain the information the most.

2. Factor Loadings table adds precision to which attributes explain the most information

Report | Scree plot

Factor Loadings [Communality Estimates]

Attribute	Axis_1		Axis_2		Axis_3		Axis_4		Axis_5	
-	Corr.	% (Tot. %)	Corr.	% (Tot. %)	Corr.	% (Tot. %)	Corr.	% (Tot. %)	Corr.	% (Tot. %)
d	0.81831	67 % (67 %)	0.13895	2 % (69 %)	0.04332	0 % (69 %)	-0.11181	1 % (70 %)	-0.06844	0 % (71 %
7	-0.65088	42 % (42 %)	0.04646	0 % (43 %)	0.31884	10 % (53 %)	-0.32585	11 % (63 %)	-0.01246	0 % (63 %
12	0.62936	41 % (41 %)	0.35347	12 % (53 %)	0.26796	7 % (61 %)	-0.02174	0 % (61 %)	0.08237	1 % (61 %
8	0.60415	37 % (37 %)	0.18615	3 % (40 %)	-0.20690	4 % (44 %)	-0.18278	3 % (48 %)	0.39488	16 % (63 %
9	0.60114	36 % (36 %)	-0.02690	0 % (36 %)	0.22627	5 % (41 %)	0.18523	3 % (45 %)	0.21091	4 % (49 %
11	0.57853	33 % (33 %)	-0.17119	3 % (36 %)	-0.02619	0 % (36 %)	0.15648	2 % (39 %)	-0.52991	28 % (67 %
2	0.53553	29 % (29 %)	0.11885	1 % (30 %)	-0.54919	30 % (60 %)	-0.22154	5 % (65 %)	0.12272	2 % (67 %
1	0.29852	9 % (9 %)	0.64094	41 % (50 %)	0.29983	16 % (66 %)	-0.10051	1 % (67 %)	-0.24913	6 % (74 %
4	0.19021	4 % (4 %)	-0.58902	35 % (38 %)	-0.07542	1 % (39 %)	-0.44226	20 % (58 %)	0.11648	1 % (60 %
0	0.45594	21 % (21 %)	-0.56719	32 % (53 %)	-0.01129	0 % (53 %)	0.37937	14 % (67 %)	-0.22227	5 % (72 %
3	0.26563	7 % (7 %)	-0.41755	21 % (28 %)	0.58461	34 % (62 %)	0.06871	0 % (63 %)	0.40734	17 % (79 %
6	0.24660	6 % (6 %)	-0.28555	8 % (14 %)	0.17658	3 % (17 %)	-0.63498	40 % (58 %)	-0.34033	12 % (69 %
Var. Expl.	3.31445	28 % (28 %)	1.59942	13 % (41 %)	1.10912	9 % (50 %)	1.01796	8 % (59 %)	0.91469	8 % (66 %

To add precision to the decision made with the scree plot, we can use the factor loadings report. It helps us see how much information is explained by each axis. Here, we can see that the two first axis explain most of the information. We can also note that attributes 5 and 6 have very little influence on the targeted attribute as they do not explain any information on axis 1 to 5.

I ran the same classification algorithm after removing attributes 5 and 6, as they seemed to explain less information and they did not influence components 1 and 2. The table

below presents the comparison of the result of the classification algorithm used before and after the PCA.

Table 2. Comparison of classification results before and after PCA

	Before PCA	After PCA
Time taken to build model	0.05 seconds	0.03 seconds
Correctly Classified Instances	78.5185%	79.6296%
Incorrectly Classified Instances	21.4815%	20.3704%
F-Measure (Weighted Avg.)	0.785	0.797
ROC Area (Weighted Avg.)	0.842	0.848

The time taken to build the model is slightly lower after the PCA because two columns have been remove : 5 and 6. We can notice an increase in the proportion of correctly classified instances, as well as the F-Measure and the ROC Area.

Performances. A number of statistics exist for measuring the performance of a classifier on a test set.

Accuracy is simply the fraction of test observations classified to the correct class (error rate = 1-accuracy). However, accuracy may provide insufficient information when the classes contain different numbers of examples or when making one type of error is more severe than making another.

To distinguish the different error types, one can use a confusion matrix :

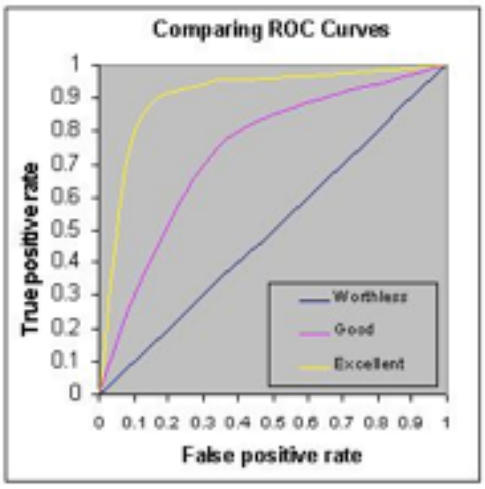
		Predicted	
		Negatives	**Positives**
Actual	**Negatives**	TN	FP
	Positives	FN	TP

source : http://www.trhvidsten.com/docs/ROSETTATutorial.pdf

The TP, TN, FP and FN allow us to calculate the F-Measure, using precision and recall :

precision = TP / TP + FP recall = TP / TP + FN
F-Measure = 2 x (precision x recall) / (precision + recall)

The ROC (Receiver Operating Characteristic) curve provides a vehicle for controlling the number of false positives and false negatives. The accuracy of the test depends on how well the test separates the group being tested into those with and without the disease in question. Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of .5 represents a worthless test.



source : http://gim.unmc.edu/dxtests/roc3.htm

---

2. Rough Set theory

Another approach to Feature Subset Selection is by using Rough Set theory. I will first explain how it works and then present the results of its use on the same dataset as I used for the PCA approach.

The theory. Pawlak's rough set theory and Boolean reasoning constitute a mathematical framework for inducing rules from examples. (3) (4)

Approximating datasets. Let us consider the decision table below. Say we want to draw a subset which gathers P1, P2 and P5. If we only consider attributes Gene1, Gene2,

Gene3 and Smoking, it is impossible to distinguish P5 from P11, P12 and P17. Do try to approximate the desired subset, we define Upper Set = {P1,P2,P5,P11,P12,P17} and Lower Set = {P1, P2}.

Table 2 An example decision table.

		Conditional attributes				Decision attribute
	Patients	Gene1	Gene2	Gene3	Smoking	Site of origin
Objects (i.e. observations)	P1	↓	↓	0	Yes	Lung
	P2	0	0	0	Yes	Lung
	P3	0	↓	↑	No	Colon
	P4	0	0	0	Yes	Lung
	P5	0	↓	0	Yes	Lung
	P6	↓	↓	0	Yes	Lung
	P7	↓	↑	0	No	Colon
	P8	↓	↑	0	No	Colon
	P9	0	↑	0	Yes	Colon
	P10	↓	↓	↑	No	Lung
	P11	0	↓	0	Yes	Lung
	P12	0	↓	0	Yes	Lung
	P13	0	↓	↑	No	Colon
	P14	0	↑	↑	No	Colon
	P15	↓	↑	0	No	Colon
	P16	↓	↓	↑	No	Colon
	P17	0	↓	0	Yes	Lung
	P18	0	↓	↑	No	Lung

source : http://www.trhvidsten.com/docs/ROSETTATutorial.pdf (1)

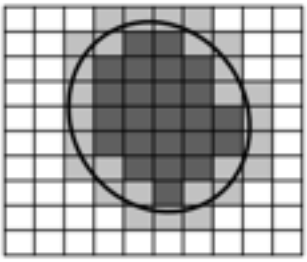


Figure 3. The rough set (the ellipse) cannot be uniquely defined by the equivalence classes (the squares), and is defined by the lower approximation (dark grey) and the upper approximation (dark plus light grey).

source : http://www.trhvidsten.com/docs/ROSETTATutorial.pdf

We can summarize the dataset with the following table :

Equivalence classes	Gene1	Gene2	Gene3	Sm.	Site of origin (Generalized decision)
E1 = {P1,P6}	↓	↓	0	Yes	{L}
E2 = {P2, P4}	0	0	0	Yes	{L}
E3 = {P3,P13,P18}	0	↓	↑	No	{C, L}
E4 = {P5, P11, P12, P17}	0	↓	0	Yes	{L}
E5 = {P7, P8, P15}	↓	↑	0	No	{C}
E6 = {P9}	0	↑	0	Yes	{C}
E7 = {P10, P16}	↓	↓	↑	No	{C, L}
E8 = {P14}	0	↑	↑	No	{C}

source : http://www.trhvidsten.com/docs/ROSETTATutorial.pdf

Reducts. A Boolean function (i.e. a function that evaluates to true or false), called the discernibility function, is constructed for each object. This function is true for all attribute combinations that discern this object from objects with a different decision and will constitute a set called reducts.

A reduct is a minimal set of attributes discerning one object from all objects with a different decision. Finding all the reducts is a NP-complete problem. However, there is a number of approximation algorithms, including greedy algorithms (Johnson 1974) and genetic algorithms (Vinterbo and Øhrn 2000), that may be used to search for reducts. (2)

For our example, the reducts are the following :

	E1	E2	E3	E4	E5	E6	E7	E8
E1	∅							
E2	∅	∅						
E3	G1, G3, S	G2, G3, S	∅					
E4	∅	∅	G3, S	∅				
E5	G2, S	G1, G2, S	G1, G2, G3	G1, G2, S	∅			
E6	G1, G2	G2	G2, G3, S	G2	∅	∅		
E7	G3, S	G1, G2, G3, S	∅	G1, G3, S	G2, G3	G1, G2, G3, S	∅	
E8	G1, G2, G3, S	G2, G3, S	G2	G2, G3, S	∅	∅	G1, G2	∅

source : http://www.trhvidsten.com/docs/ROSETTATutorial.pdf

Indeed, Gene1, Gene3 and Smoking are the 3 attributes that are different between E1 and E3 and imply a different decision. The reducts between E1 and E2 is empty because the decision is Lung for the 2 sets.

Experiment and results.

The dataset used for the experiment is the same as for the PCA approach and is named 'Heart'. It has 13 attributes and 1 binary decision class, and contains 270 instances.

The original dataset (as extracted from RSES data folder) can be found at the following url : https://drive.google.com/file/d/0B4Y4vCvFO0Wic19wZy00YVU4YTg

The first step was to find the reducts, which I did by using RSES software. Indeed, RSES may be used to select the most relevant attributes in a dataset and to remove the unwanted attributes which pollute the classification algorithms. In this section, we will present the results and consequences of Feature Subset Selection (FSS) using RSES.

The most relevant reduct I found for this dataset was { attr0, attr2, attr3, attr4, attr6, attr7, attr9, attr11, attr12 }. Which means that attributes { attr1, attr5, attr8, attr10 } were ruled out from the dataset.

Table 3. Comparison of classification results before and after Rough Set

	Before Rough Set	**After Rough Set**
Time taken to build model	0.05 seconds	0.03 seconds
Correctly Classified Instances	78.5185%	78.8889%
Incorrectly Classified Instances	21.4815%	21.1111%
F-Measure (Weighted Avg.)	0.785	0.789
ROC Area (Weighted Avg.)	0.842	0.842

Same as for the PCA, we can note that the time taken to build the model is slightly lower after finding the reducts with Rough Set because four columns have been remove : 1, 5 and 8 and 10. We can notice a small increase of the proportion of correctly classified instances, a slight increase of the F-Measure and no increase of the ROC Area.

3. Comparison of PCA and Rough Set theory

Now that I have detailed the process of my experiments and briefly described how PCA and Rough Set theory work, I will present a table of comparison of the two technics.

Table 3. Comparison of classification results on raw data, after PCA and after Rough Set

	On raw data	After PCA	After Rough Set
Time taken to build model	0.05 sec	0.03 sec	0.03 sec
Correctly Classified Instances	78.5185%	79.6296%	78.8889%
Incorrectly Classified Instances	21.4815%	20.3704%	21.1111%
F-Measure (Weighted Avg.)	0.785	0.797	0.789
ROC Area (Weighted Avg.)	0.842	0.848	0.842

Both models were faster to build after PCA or Rough Set because the number of attributes is inferior to the original dataset. However, the PCA did better at selecting the most relevant attributes.

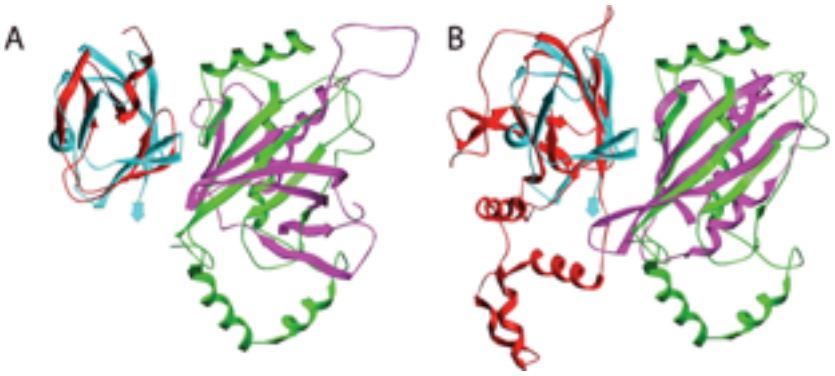
Discussion. It is important to note that this 'Heart' dataset contains only 13 attributes and 270 instances, which is very different from the protein databases that I used for the application (see section 3). To really demonstrate the efficiency of these two different approaches and to be able to compare them, it would have been necessary to perform these measures on more than just one dataset, with more attributes, different attribute types etc.

III. Application on Protein Interaction Prediction

Introduction. « Protein–protein interaction prediction is a field combining bioinformatics and structural biology in an attempt to identify and catalog physical interactions between pairs or groups of proteins. »

– Wikipedia

Problem to solve : Can protein A interact with protein B?



source : http://www.hhmi.org/research/computational-studies-structure-and-function-biological-macromolecules

It is possible to experimentally determine whether two proteins can interact or not in a laboratory. However, the cost of such experiments is slowing down the process and the number of combination is very important. This is why it is interesting to develop models to try to predict the protein-protein interaction programmatically. (5) (6)

What makes our application different from the existing ones in this field, is that we used two different datasets : one with proteins known from laboratory experiments, to be capable of interacting with each other (targeted class value = 'INTERACT'). And another dataset of proteins that we know cannot interact with each other (targeted class value = 'NOINTERACT'). The class that we want to predict is "A interacts with B".

Servers and datasets

I had to use 4 different sources of data to build the dataset to train the model to predict the interaction of the proteins :

1. Database of Interacting Proteins (DIP) http://dip.doe-mbi.ucla.edu

The first dataset we will use is collected and kept up to date by University UCLA and is called Database of Interacting Proteins (DIP). It gathers several millions records of pairs of proteins which can interact together (targeted class value = 'INTERACT').

2. The Negatome database http://mips.helmholtz-muenchen.de/proj/ppi/negatome/

The Negatome database provides the opposite information from DIP. It is database of thousands of pairs of protein-protein which are unlikely to interact with each other (targeted class value = 'NOINTERACT'). Using DIP and Negatome, we will be able to build a training dataset with two outputs : one class for proteins which can interact with each other and one for proteins which cannot.

3. Universal Protein Resource (Uniprot) http://www.uniprot.org/

From the identification numbers given by DIP and Negatome, it is possible to retrieve the FASTA of the protein with Uniprot database. FASTA is a string of letters representing the different amino acid composing the protein. (7) (8)

4. Protein Feature Server (Profeat) http://jing.cz3.nus.edu.sg/cgi-bin/prof/prof.cgi

Profeat is a web application developed by the National University of Singapore. From a FASTA sequence, it allows to compute physicochemical properties of proteins and peptides. They will be the attributes of our dataset. (9)

Building the Model

We need to build the training dataset to construct the model, which will then be used to predict whether two proteins can interact or not. I had to use DIP, Negatome, Uniprot and Profeat databases to build this dataset :

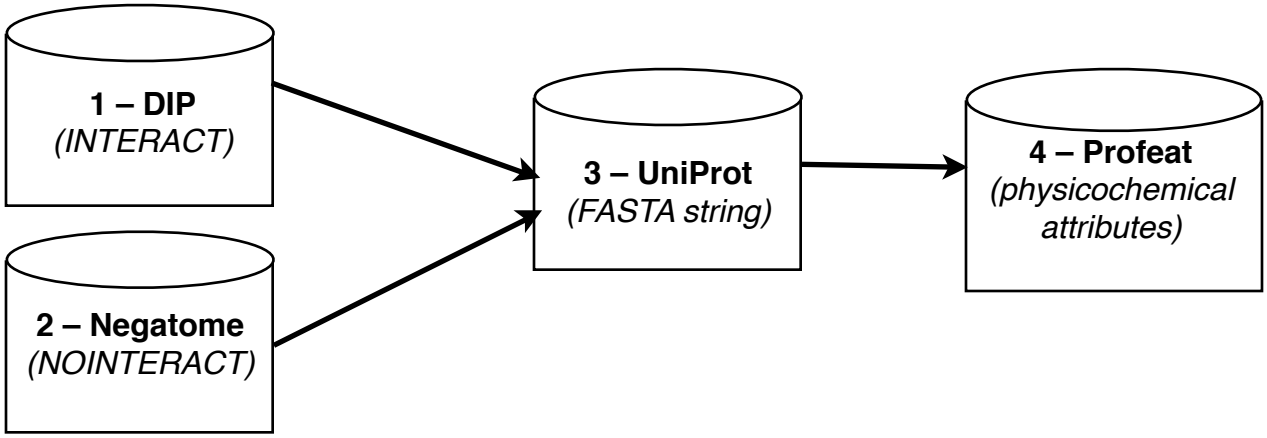


Figure 1. Process to collect information to build training set

From this process, I was able to put together a dataset containing 172 pairs of proteins which can or cannot interact with each other, and their physiochemical properties.

Table 4. Extract of the dataset created used to build the model

protA	protB	[F1.1.1.1]	[F1.1.1.2]	[F1.1.1.3]	[...]	[F1.2.1.396]	[F1.2.1.397]	[F1.2.1.398]	[F1.2.1.399]	[F1.2.1.400]	D
sp|P10415|BCL2_HUMAN	sp|P04049|RAF1_HUMAN	8.2719925	1.498657	4.748115	[...]	0.2318395	0	0	0.07728	0.07728	INTERACT
sp|P11540|BAR5_BACAM	sp|P00648|RNBR_BACAM	7.791932	1.4295825	5.088464	[...]	0.320513	0.320513	0	0	0.561798	INTERACT
sp|Q09472|EP300_HUMA	sp|P04637|P53_HUMAN	6.2017365	2.3286025	4.139393	[...]	0.148272	0.0621635	0.041442	0	0.041442	INTERACT
sp|P16220|CREB1_HUMA	sp|Q92793|CBP_HUMAN	9.0288235	1.6102615	3.3565825	[...]	0.0204835	0.0204835	0.2289925	0	0.040967	INTERACT
sp|Q92793|CBP_HUMAN	sp|P10243|MYBA_HUMAN	5.930731	1.755134	4.256626	[...]	0.0870615	0.1536395	0.1485115	0	0.040967	INTERACT
sp|A0A5B9|TRBC2_HUMA	sp|P01887|B2MG_MOUSE	6.7844915	1.9639315	4.3480315	[...]	0.423729	0	0.423729	0.423729	0	NOINTERACT
tr|A0N6Y3|A0N6Y3_9VIU	sp|P00127|QCR6_YEAST	5.2705195	1.6148515	10.286732	[...]	0.943396	0.471698	0	0.342466	0.943396	NOINTERACT
tr|A0N8N6|A0N8N6_MOU	sp|P61769|B2MG_HUMAN	4.713991	2.1561255	4.6955625	[...]	0.866207	0.423729	0.866207	0	0.866207	NOINTERACT
tr|A1BBA0|A1BBA0_PAR	sp|P22619|DHML_PARDE	9.042553	6.382979	7.978723	[...]	0	0	0	0.534759	0	NOINTERACT
tr|A1EKW0|A1EKW0_VIB	tr|Q9KN57|Q9KN57_VIB	8.739837	0.203252	4.674797	[...]	0.203666	0.203666	0.407332	0	0	NOINTERACT

At this stage, I have 172 instances. Each object is a pair of protein (protA and protB) described by 420 attributes and its binary decision class : 'INTERACT' or 'NOINTERACT'.

On this dataset, I applied a classification algorithm with Weka called DecisionTable. I applied this algorithm :
-on the raw dataset;
-on the dataset after running a PCA; and
-on the dataset after selecting reducts with the Rough Set theory.

In the table below are the results of the 3 classifications.

Table 5. Comparison of classifications of protein interaction on raw data, after PCA and after Rough Set

	On raw data	After PCA	After Rough Set
Time taken to build model	126.9 sec	0.08 sec	0.02 sec
Correctly Classified Instances	71.345%	74.5098%	84.3137%
Incorrectly Classified Instances	28.655%	25.4902%	15.6863%
F-Measure (Weighted Avg.)	0.713	0.743	0.841
ROC Area (Weighted Avg.)	0.769	0.749	0.856

The Feature Subset Selection using PCA or Rough Set theory allowed to drastically divide the time necessary to build the model by more than 1000. It also improved the classification results : the proportion of correctly classified instances went from 71.3% to 74.5% using the PCA and from 71.3% to 84.3% using Rough Set theory. Unlike in section 2, Rough Set produced a better subset than the PCA. This can be due to the number of attributes which is higher than in the 'Heart' dataset, and also to the data type.

Discussion. These results are not sufficient to prove the efficiency of PCA or Rough Set theory during preprocessing. More measures, with more instances should be done, using different classifiers.
Also, one more important question should be raised : can the two sources of protein pairs be used together to build the model? Indeed, the pairs of non-interacting proteins (labeled 'NOINTERACT' in the dataset) and the pairs of interacting proteins (labeled

'INTERACT'), do not come from the same databases. The first set of pairs comes from the Negatome database and the second from the DIP database. Let us imagine that the first set was build with proteins coming from the brain and the second with proteins coming from the lungs. If we take the physiochemical attribute 'F.1.1.1' (which we use for the classification), it may always be higher for the first set of proteins than the second one, just because of the function of these proteins. So the classifier will use this attribute to determine whether the two given proteins can interact or not, but what it will actually predict is whether the proteins are brain or lung proteins. So more work should be done with biologists to determine whether these two different databases (Negatome and DIP) are similar enough and were build using the same experimentation methodology, to use them to build the model.

Conclusion

This internship made me realize how interesting bioinformatics is. Using machine learning technics in biology is something that can help medicine a lot. I am even considering working in this field later on. It is unfortunate that the organization of my internship with Dr Bhasker Pant did not allow me to produce results of a better quality. Overall, PCA and Rough Set theory are two different approaches to Feature Subset Section, which can be very useful in bioinformatics or any domain dealing with datasets which contain many attributes. However, the study presented in this report is not enough to say if the two approaches can really apply in protein interaction prediction or to say which of the two technics is more efficient.

References

(1) Torgeir R. Hvidsten *A tutorial-based guide to the ROSETTA system* October 2013

(2) Adriano Donizete Pila *Rough Sets Reducts as a Filter Approach for Feature Subset Selection: An Empirical Comparison with Wrapper and Other Filters* University of Sao Paulo March 2001

(3) Zdzisław Pawlak *Rough Set and Data Mining*

(4) Richard Jensen Qiang Shen *Computational Intelligence and Feature Selection Rough and Fuzzy Approaches* Aberystwyth University 2008

(5) Zhu-Hong You1, Ying-Ke Lei, Lin Zhu, Junfeng Xia, Bing Wang. «*Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis*» From *The 2012 International Conference on Intelligent Computing (ICIC 2012)* Huangshan, China. 25-29 July 2012

(6) Torgeir R. Hvidsten and Jan Komorowski. «*Rough Sets in Bioinformatics*» From *The Linnaeus Centre for Bioinformatics,* Uppsala University, Uppsala, Sweden

(7) Arunkumar Chinnasamya, Ankush Mittalb, Wing-Kin Sungc. «*Probabilistic prediction of protein–protein interactions from the protein sequences*» From *Computers in Biology and Medicine 36* (2006) 1143–1154

(8) Changhui Yan, Vasant Honavar, and Drena Dobbs. *Predicting Protein-Protein Interaction Sites From Amino Acid Sequence* Department of Computer Science Iowa State University October 2002

(9) Guilherme T. Valente, Marcio L. Acencio, Cesar Martins, Ney Lemke. *The Development of a Universal In Silico Predictor of Protein-Protein Interactions* Semmelweis University, Hungary Published May 31, 2013