A Compromise between N-gram Length and Classifier Characteristics for Protein Classification

*Faouzi Mhamdi *,Ricco Rakotomalala** and Mourad Elloumi **

faouzi.mhamdi@ensi.rnu.tn, rakotoma@univ-lyon2.fr and mourad.elloumi@fsegt.rnu.tn
* URPAH, Faculté des Sciences de Tunis, Université d'El Manar, Tunisie
** Laboratoire ERIC Université de Lumière Lyon 2, France

Summary

Many scientific works deal with the protein classification problem and various learning methods and descriptors are used in them. In this paper, we want to systematize the analysis of the behavior of learning algorithms according to the features extracted from the primary description of proteins. We have used n-grams descriptors by testing the interaction between various length *n* of n-grams and the characteristics of the supervised learning methods. The main conclusion is that moderate length of n-grams (n = 2 or n = 3, ...) and linear support vector classifier (SVM) give the best compromise. But, a thorough analyze of the results puts into perspective this conclusion: the main characteristic which influences the accuracy of the classifier seems to be the dimensionality of the representation space.

Key words:
Data mining, Protein Classification, n-grams, KNN, SVM, CART

Introduction

The functional and structural annotation of proteins is an important problem in Proteomics which involves a strong need for efficient techniques for the analysis of protein sequence data. Fortunately, in the last years, the area of machine learning techniques was extended to unstructured dataset such as text processing and combinatorial chemistry.

In fact, the knowledge discovery framework [4], particularly text mining [11], represents a clear guideline for protein sequence analysis. In comparison to traditional approaches, text mining is characterized by two supplementary steps. The first step is the extraction of features from the original description in order to build an attribute-value table which is useful for ulterior data mining techniques. The second one consists of the selection of the "best" subset of features among the set produced by the first step. Indeed, he elevated number of potential features, computational efficiency is of a primary importance.

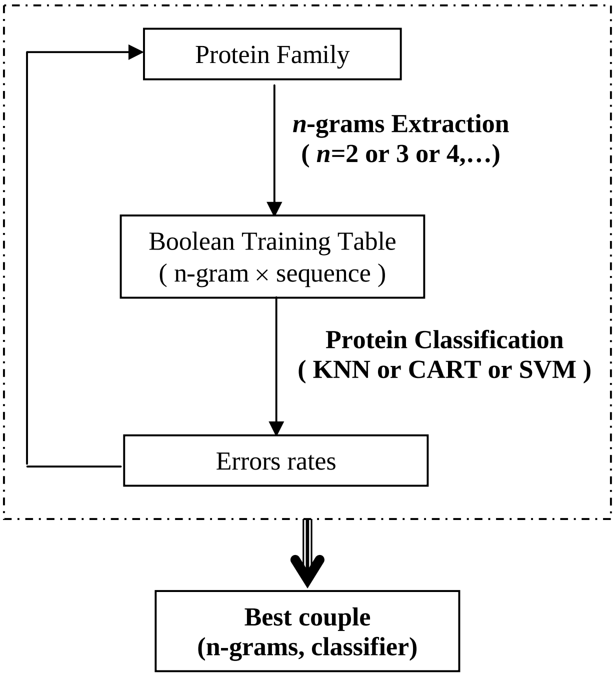


Fig. 1 Evaluation Process

In this paper, we use the text mining framework for a protein classification by giving their primary structures (Figure 2.). A protein sequence is a series of amino acids that have a specific order. There exist 20 amino acids that allow the description of a protein by a sequence of characters. Besides, proteins are grouped into several families according to the functions they perform, and all proteins contained in the same family have similar structures. Thus, by giving a set of proteins with known properties, we have to look for inducing classification rules that associate motifs to protein families (classes). This

Manuscript received April 5, 2006.
Manuscript revised April 30 , 2006.

problem can be considered as a classic data mining problem. The attributes used for data mining are the various existing motifs in each protein sequence. The machine learning algorithms permits to efficiently automate the discovery of a priori unknown predictive relationships from large datasets into computational biology [9]. Indeed, these research tools allow us to accelerate the manipulation and the treatments of the biological data. The classification process is showed in figure 1.

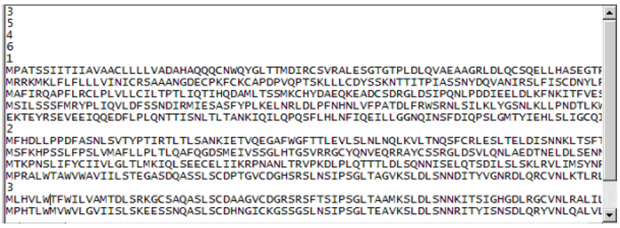
The rest of this paper is organized as follows. In section 2, we present the proteins representation modality that we used. In section 3, we present the different classifiers tested. Realized experiments and discussion of their results are presented in section 4. Finally, conclusion and some further works are given in section 5.

2. Protein representation

2.1 n-grams extraction

The used protein families in this work are well-selected from the data bank SCOP [10] and are summarized in one file (Figure 2.). For each family, every line defines a protein by a sequence of characters. Besides, given the elevated length of every protein, it is necessary to choose the optimal features from the original data description in order to classify the proteins. In fact, text mining approaches [11] show that character sequence (n-grams) produces relevant results. And our goal is to define the optimal length (n) of the extracted sequences.

In a previous work [8], we tested various values of *n* and we concluded that if the value of n is low, captured information will be of too bad quality. However, if it is high, features will be too specific and disturbed, but, the calculation time is therefore impracticable. Indeed, relatively to the n-grams length, the theoretical number of features is 20^n. And in order to define the best size, we test many sizes(n = 2, n = 3, n = 4,...).



```
3
5
4
6
1
MPATSSIITIIAVAACLLLLVADAHAQQQCNWQYGLTTMDIRCSVRALESGTGTPLDLQVAEAAGRLDLQCSQELLHASEGTF
MRRKMKLFLFLLLVINICRSAAANGDECPKFCKCAPDPVQPTSKLLLCDYSSKNTTITPIASSNYDQVANIRSLFISCDNYLF
MAFIRQAPFLRCLPLVLLCILTPTLIQTIHQDAMLTSSMKCHYDAEQKEADCSDRGLDSIPQNLPDDIEELDLKFNKITFVES
MSILSSSFMRYPLIQVLDFSSNDIRMIESASFYPLKELNRLDLPFNHNLVFPATDLFRWSRNLSILKLYGSNLKLLPNDTLKW
EKTEYRSEVEEIQQEDFLPLQNTTISNLTLTANKIQILQPQSFLHLNFIQEILLGGNQINSFDIQPSLGMTYIEHLSLIGCQI
2
MFHDLLPPDFASNLSVTYPTIRTLTLSANKIETVQEGAFWGFTTLEVLSLNLNQLKVLTNQSFCRLESLTELDISNNKLTSFT
MSFKHPSSLFPSLVMAFLLPLTLQAFQGDSMEIVSSGLHTGSVRRGCYQNVEQRRAYCSSRGLDSVLQNLAEDTNELDLSENN
MTKPNSLIFYCIIVLGLTLMKIQLSEECELIIKRPNANLTRVPKDLPLQTTTLDLSQNNISELQTSDILSLSKLRVLIMSYNF
MPRALWTAWVWAVIILSTEGASDQASSLSCDPTGVCDGHSRSLNSIPSGLTAGVKSLDLSNNDITYVGNRDLQRCVNLKTLRL
3
MLHVLWTFWILVAMTDLSRKGCSAQASLSCDAAGVCDGRSRSFTSIPSGLTAAMKSLDLSNNKITSIGHGDLRGCVNLRALIL
MPHTLWMVWVLGVIISLSKEESSNQASLSCDHNGICKGSSGSLNSIPSGLTEAVKSLDLSNNRITYISNSDLQRYVNLQALVL
```

Fig. 2 Protein family file

2.2 Building attribute-value dataset

The next step is the construction of the attribute-value table from the original unstructured dataset. For a given example, several kinds of values can be attributed to a feature. It is possible to use the occurrence of features, their frequencies, or simply their presence/absence. In order to make the best choice from different representations, we have tested 4 kinds of data representation :

- **Boolean:** indicates whether one n-gram is present within a sequence or not.

$$\overline{w}_i^j = 1 \text{ if } x_j^i > 0 \text{ and } 0 \text{ else} \qquad (1)$$

- **Occurrence:** number of occurrences of one n-gram in a sequence.

$$\overline{w}_i^j = x_j^i \qquad (2)$$

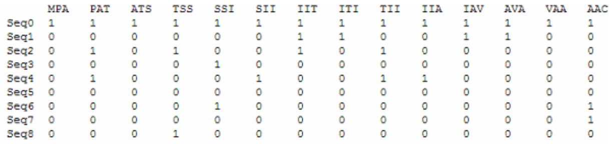
- **Frequency:** relative frequency of one n-gram with regard to the number of 3-grams composing a sequence.

$$\overline{w}_i^j = \frac{x_i^j}{x_i^*} \text{ where } x_i^* = \sum_{j=1}^{p} x_i^j \qquad (3)$$

- **TF*IDF:** corrects the frequency of n-gram according to its frequency within the file.

$$\overline{w}_i^j = x_i^j \log\left(\frac{n}{x_*^j}\right) \text{ where } x_*^j = \sum_{j=1}^{n} x_i^j \qquad (4)$$

The first data representation (Boolean) can give the impression that it is rather rough but several studies in the text mining domain showed their effectiveness. Besides, as we will use the genetic algorithms for feature selection in ulterior stages, it appears very reasonable to use a binary representation. Thus, our data table is Boolean where each line represents a protein and each column represents an n-grams. The binary value of a case indicates whether the relative n-grams belong (1) or not (0) to the considered protein (Figure 3.).



	MPA	PAT	ATS	TSS	SSI	SII	IIT	ITI	TII	IIA	IAV	AVA	VAA	AAC
Seq0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Seq1	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Seq2	0	1	0	1	0	0	1	0	1	0	0	0	0	0
Seq3	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Seq4	0	1	0	0	0	1	0	0	1	1	0	0	0	0
Seq5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Seq6	0	0	0	0	1	0	0	0	0	0	0	0	0	1
Seq7	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Seq8	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Fig. 3 Learning boolean file.

However, the number of the n-grams extracted for the given discrimination is variable with the value of n. And in this work, we are interested to identify the best classifier for every size of *n*. The experiments done show that every classifier has a kind of behavior with the size of n-grams.

In the next section we present the functioning of the used classifiers in detail.

3. Classifiers

It is always attracting in the experiments to show the results obtained using a large number of methods, then to mechanically select the best one with regard to a given criterion, which is often the generalization error rate. In our case, the situation is a little bit different, as we want to determine the method which appears most adequate according to the kind of descriptors (the length *n* of the *n*-grams) which we produce from the primary structure of proteins.

In this article, we choose to show the behavior of 3 learning methods which are very different from the point of view of their representation bias as well as from the point of view of their learning bias[5]. And what is most interesting in our context is that it was possible for us to understand their behavior from the characteristics of the dataset which we generated.

The first algorithm that we implemented is the **1-nearest neighbor** [5]. This algorithm produces a nonlinear model which makes it possible to find complex forms. Its principal weakness lies in its incapacity to estimate in a reliable way the probabilities when one has very scattered data. This is the case when one has a great number of irrelevant descriptors or when the relationship between the number of descriptors and the number of observations is reversed. We can thus expect that the performances of this method degrade as we increase the size n of the n-grams.

On the contrary, the **support vector machine** (SVM) has a very strong resistance to the noise [1]. From its very restrictive learning bias, expressed by the maximization of the margin, it is not very sensitive to very disturbed spaces of representation. This characteristic is accentuated by the fact that we chose a linear SVM. There were two reasons for this choice. The first reason is that the definition of the kernel is always difficult and is often based on the experimentation but we do not want to make this kind of the improvement, which brings us towards the overfitting, on our dataset. The second reason is that a linear model remains "readable", at least with regard to the sign of the computed parameters. The experiments show that the use of other kernels does not give better results on our dataset.

Lastly, we chose an approach which proposes an easily interpretable classifier and which is the **CART decision trees method** [6]. This nonlinear model of prediction can be transformed into a rule based system. It will be very easy to interpret the results with biologists thereafter. Another positive point is that the algorithm integrates a process of selection during the training because it should not be affected by the profusion of descriptors. On the other hand, we know that this method, carrying out a strong fragmentation of the data, can quickly be limited when the learning set comprises few observations which is actually the case in this experimentation.

In the following section, we will study the behavior of these learning methods in our protein classification context. We will try, above all, to connect these results with the size *n* of the *n*-grams used to generate the descriptors. Let us note that we have restricted ourselves with Boolean weighting in this paper, indicating the presence or absence of one n-gram in a protein sequence. The other kinds of weighting such as the frequency or the TF/IDF, frequently used in the text categorization, were the subject of another study [7]. The results showed that no kind of weighting was really outperformed by the other. Thus, the Boolean choice was essential because it made it possible to use in an undifferentiated way the methods that work on discrete or continuous descriptors.

4. Experiments and results

4.1 Dataset and evaluation

We randomly extracted 5 families of proteins in SCOP data bank [10] and we roughly have 50 observations in each family.

Our goal is to recognize a family compared to another by using the supervised learning algorithm. We thus chose to oppose them two to two. This means that we have 10 datasets made up on an average of 100 observations i.e. 100 sequences of two families of proteins.

The number of descriptors varies according to the length *n* of the n-grams. We chose to work on 3 levels of representation: *n* = 2, 3 and 4. For these values, the number of distinct n-grams extracted in each dataset increases very quickly (Table 1). We know that when *n* increases, the ratio between the number of descriptors and the number of observations also increases. So, we will quickly be confronted with the curse of dimensionality problem, and by-there an overfitting problem.

Table 1 : Number of descriptors according to the length *n* of *n*-grams

Proteins pairs	***2-grams***	***3-grams***	***4-grams***
F12	400	6600	23408
F13	400	6288	22515
F14	400	6183	23662
F15	397	6004	22790
F23	400	7143	31973
F24	400	7098	33185
F25	400	7011	32126
F34	400	6860	31809
F35	400	6740	30954
F45	400	6659	31904

We compute the error rate using a resembling method. We chose a 5 X 2 cross validation which we repeat a great number of times. This approach is privileged because it makes it possible to obtain slightly biased and relatively stable results [2]. In our context, of strong risk of overfitting, we seek to produce general results and to avoid being too dependent on our dataset.

4.2 Results

Linear SVM is the most successful one. Whatever the length of n-grams used, it exceeds the two other methods. We show in table 2 the average of the error rates computed on our dataset. Note that because we have randomly extracted the proteins families from the whole database, we can expect that the average of the computed error rate on these families is representative in the performance of each learning method according to the length n of the n-grams.

Table 2 : Average error rate for each learning method on each data representation

Method	**2-grams**	**3-grams**	**4-grams**
1-NN	0.043	0.214	0.269
SVM	0.032	0.038	0.081
CART	0.210	0.155	0.141

We also note that always concerning the SVM, if *n* = 2 and *n* = 3 give equivalent results, the situation strongly degrades when we set *n* to 4 (Table 3) Therefore, we deduce that the choice of *n* = 4 gives irrelevant descriptors to discriminate the proteins. This conclusion must, however, be moderated by the fact that by increasing *n*, the dimension of the input space is high. The research of the boundary between the families in a space, where examples are strongly disseminated in the space of representation, works badly. Even if the SVM are to handle with high dimensionality, there is a limit in their capacity to produce a stable border taking into account the characteristics of our data.

Table 3: Detailed results for SVM algorithm on each pair of families

Proteins pairs	**2-grams**	**3-grams**	**4-grams**
F12	0.020	0.020	0.115
F13	0.051	0.070	0.117
F14	0.023	0.025	0.053
F15	0.055	0.052	0.094
F23	0.032	0.027	0.056
F24	0.009	0.018	0.047
F25	0.014	0.028	0.102
F34	0.046	0.039	0.062
F35	0.039	0.065	0.111
F45	0.032	0.033	0.050

The results obtained with the nearest neighbor method come to corroborate these conclusions. For *n* = 2, the method produces good results, close to those of the SVM. On the other hand, when we increase *n* (*n* >= 3), with thousands of descriptors compared with a hundred observations, it is impossible to carry out a reliable estimate of the probabilities, locally, in the neighborhood of the point to be classified. Thus, if *n* = 2 seems the good choice for nearest close, it is before all the consequence of the reduced number of the descriptor used for the training compared to the other choices (*n* >= 3). To evaluate this assertion, we have performed a very simple selection of variables for *n* = 3 which is suggested in [3]. We sorted the descriptors according to the chi-2 (or some information measure). After some attempts, we decided to select the 25 first features. The evaluation, using the cross validation, shows that the average error rate is 0.05, close to that obtained with the 2-grams (Figure 4.). We see also that when we have too many descriptors, the performance of the classifier decreases.

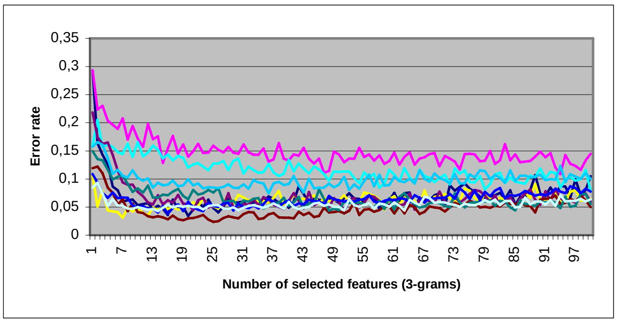


Fig. 4 Error rate of 1-nearest neighbor according to the number of selected features

The decision trees algorithm can bring answers to these questions. Indeed, they make it possible to automatically select the right descriptors, they should not suffer from the profusion of irrelevant descriptors. We, however, note that this method gives bad results whatever the length *n* of the n-grams is. The main explanation is probably the small size of our samples, more especially, as we subdivide our dataset during the cross-validation. The decision trees have as a characteristic to proceed by successive segmentations and to isolate the groups from observations coming from the same family. Mechanically, with the data fragmentation, the leaves of the tree hold few observations, the classifier is very unstable. CART accentuates that by needing a second part of the sample for post-pruning, we nevertheless noted same inefficiency with C4.5 [6].

When we study the detailed results (Table 1 and Table 4), we notice a new phenomenon. The method, by selecting the most relevant descriptors automatically shows that the best space of representation is not uniform whatever the problem treated. In our ten evaluations, we note that the 4-grams produce the best results on 6 datasets. That means that this kind of representation (*n* = 4) is not to set aside definitively. Its bad performance for the other methods seem to be related to the high dimensionality that it generates for the methods which do not integrate an automatic feature selection process.

Table 4. Detailed results for decision trees algorithm on each pair of families

Proteins pairs	2-grams	3-grams	4-grams
F12	0.167	0.213	0.159
F13	0.265	0.290	0.300
F14	0.211	0.048	0.120
F15	0.246	0.165	0.159
F23	0.173	0.230	0.189
F24	0.180	0.141	0.073
F25	0.143	0.139	0.141
F34	0.247	0.125	0.088
F35	0.259	0.134	0.121
F45	0.208	0.068	0.063

5. Discussion

These results show that to set the value of *n* can only be a compromise. The learning method most adapted in the protein classification relies on the characteristics of the representation space resulting from the choice of *n*.

Indeed, we must at the same time juggle with the low number of examples, the profusion of irrelevant descriptors, and its correlation, a representation space which makes the evaluation of the probabilities very difficult. In this context, it is not astonishing that the SVM, even linear, outperforms the other methods. From its very restrictive learning bias, it can handle the high dimensionality. But, we note that when the dimension is very high ($n = 4$), with a lot of irrelevant features, its performances are significantly degraded. When we perform a feature selection using the feature ranking method [3] for SVM with $n = 3$, we see that we obtain a very slightly reduction of the error rate, and we decrease significantly the number of selected features. In each pair of the families, we obtain results which are similar to 1-NN (Figure 4). One hundred of descriptors are useful for the classification process.

A further highly significant deduction is that we cannot set definitively the length n of the n-grams for the protein classification process. Put aside the situation, very particular, of the nearest neighbors, the right length depends on the problem and the used learning algorithm. A trivial solution would be to generate all the 2-grams, all the 3-grams, all the 4-grams, then to use a feature selection to choose the most powerful subset for the classification. But, this is an unsatisfactory approach because of several reasons. On the one hand, if we restrict to $n = 4$, perhaps it would be useful to try $n >= 5$ but the number of generated descriptors becomes very high. It is not tractable on a computer and a great number of descriptors will be redundant. This first approach seems to give an encouraging results according to the generalization error rate. On the other hand, it is unsatisfactory because in setting arbitrarily the length of n-grams, we restrict the scope of the solution. So, we have to manage simultaneously two problems. Firstly, we have to generate descriptors without ad hoc restriction on the length *n* of the n-grams. Secondly, we have to perform a feature selection in order to remove irrelevant and/or redundant descriptors. Those founded on the partial correlation seem to give promising results. They also make it possible to quickly treat data containing thousands of descriptors in a reasonable time [12].

6. Conclusion

The realized work in this paper has aimed to present an optimal protein classification process. The process of getting the maximum benefit from it consists, on the one hand, in identifying the best couple of n-grams and classifier. On the other hand, the classification processes must be realized in a reasonable time. Therefore, we have variated the size of n-grams between {2,3,4}, and then, evaluated every kind of n-grams with various classifier, such as KNN, Decision tree and SVM. The results show that the optimal couple is not the same usually. But the SVM and the decision tree classifiers give the best results respectively with 2-grams and 3-grams. We must note that when we use the feature ranking process, we ameliorate the error rate of classification with a spectacular reduction of the number of n-grams.

In a future work, we would like to ameliorate these results by creating an heterogeneous n-grams set i.e. 2-grams, 3-grams, 4-grams etc. After this, we would like to try to classify protein with the best subset n-grams from the initial set.

References

[1] Nello Cristianini and John Shawe-Taylor, An Introduction to Support Vector Machines and other kernel-base learning methods, Cambridge University Press, 2000.

[2] T. Dietterich, Approximate statistical tests for comparing supervised classification learning, Neural Computation journal, v 10, n 7, pp. 1895--1924, 1999.

[3] W. Duch and T. Wieczorek and J. Biesiada and M. Blachnik, Comparison of feature ranking methods based on information entropy, Proceedings of International Joint Conference on Neural Networks (IJCNN), IEEE Press, pp. 1415--1420, 2004.

[4] U. Fayyad and G. Shapiro and P. Smyth, From data mining to knowledge discovery : A overview, Advances in Knowledge Discovery and Data Mining, MIT Press, pp. 1--34, 1996.

[5] T. Hastie and R. Tibshirani and J. Friedman, The elements of statistical learning, Springer Series in Statistics, Springer-Verlag, ISBN 0-387-95284-5, New York, p. 533, 2001.

[6] R. Kohavi and J.R. Quinlan, Decision-tree Discovery, Handbook of Data Mining and Knowledge Discovery, Oxford University Press, pp. 267--276, 2002.

[7] F. Mhamdi and M. Elloumi and R. Rakotomalala, Desciptors Extraction for Proteins Classification, In Proceeding of NCEI'2004, New Zealand, 2004.

[8] F. Mhamdi and M. Elloumi and R. Rakotomalala, textmining, features selection and datamining for proteins classification,IEEE/ICTTA' 04,2004.

[9] L.C. Molina and L. Belanche and A. Nebot, Feature Selection Algorithms: A Survey and Experimental Evaluation, In Proceedings of ICDM'02, Maebashi City, Japan, 2002.

[10] G.A. Murzin and E.S. Brenner and T. Hubbard and C. Chothia, SCOP: a structural classification of proteins database for the investigation of sequences and structures, J. Mol. Bio., v.247, pp. 536--540, 1995.

[11] F. Sebastiani, Machine learning in automated text categorisation, ACM Survey, V. 34, nuber 1, pp. 1--47, 2002.

[12] Lei Yu and Huan Liu, Efficiently handling feature redundancy in high-dimensional data,KDD '03: Proceedings of the ninth ACM SIGKDD, pp. 685—690, 2003.

Faouzi Mhamdi, received the Licence’s Degree in Computer Sscience in 1999 from the Faculty of Sciences of Tunis, Tunisia. Then, he received a Master’s Degree in Computer Science from the National School of Computer Science, Tunis, Tunisia. Now, he is preparing a PhD Degree in Computer Science at the Faculty of Sciences of Tunis. His main research area is Knowledge Discovery in Databases and its application in Bioinformatics.

Dr. Ricco Rakotomalala is associate professor in computer science at the University Lumière (Lyon 2) since 1998, he is member of the ERIC Laboratory. His main research area is knowledge discovery in databases, especially supervised machine learning methods. He designed the software TANAGRA which implements numerous data mining algorithms, freely available on the web, successor of the widely distributed SIPINA software.

Dr. Mourad Elloumi received an Undergraduate Degree in Mathematics and Physics in 1984, and a Master's Degree in Computer Engineering in 1988, from the Faculty of Sciences of Tunis, Tunisia. He also received a Master's Degree in Computer Science in 1989, and a PhD Degree in Computer Science in 1994, from the University of Aix-Marseilles III, France. Then, he received a *Habilitation* for conducting research in Computer Science in 2003, from the National School of Computer Science, Tunis, Tunisia. He is currently an Master of Conference in the Computer Science Department in the Faculty of Economic Sciences and Management of Tunis, Tunisia. His research interests are Computational Molecular Biology, Algorithmics, and Knowledge Discovery and Data Mining.