

A hierarchical n-grams extraction approach for classification problem

Faouzi Mhamdi¹, Ricco Rakotomalala² and Mourad Elloumi¹

¹ URPAH, Unité de Recherche en Programmation, Algorithmique et Heuristiques, Faculté des Sciences de Tunis, Université d'El Manar, Tunisie

² Laboratoire ERIC, Université Lyon 2, France

faouzi.mhamdi@ensi.rnu.tn, ricco.rakotomalala@univ-lyon2.fr,
mourad.elloumi@fsegt.rnu.tn

Abstract. We are interested in protein classification based on their primary structures. The goal is to automatically classify proteins sequences according to their families. This task goes through the extraction of a set of descriptors that we present to the supervised learning algorithms. There are many types of descriptors used in the literature. The most popular one is the n-gram. It corresponds to a series of characters of n-length. The standard approach of the n-grams consists in setting first the parameter n, extracting the corresponding n-grams descriptors, and in working with this value during the whole data mining process. In this paper, we propose a hierarchical approach to the n-grams construction. The goal is to obtain descriptors of varying length for a better characterization of the protein families. This approach tries to answer to the domain knowledge of the biologists. The patterns, which characterize the proteins' family, have most of the time a various length. Our idea is to transpose the frequent itemsets extraction principle, mainly used for the association rule mining, in the ngrams extraction for protein classification context. The experimentation shows that the new approach is consistent with the biological reality and has the same accuracy of the standard approach.

Keywords: Data mining, Protein Classification, SVM, Association Rules, Frequent itemsets, n-grams.

1 Introduction

Biologists play a central role in the classification of individuals such as animals, plants, genes, proteins, etc. However, the great amount of biological data such as proteins, DNA, RNAm etc. involves a strong need for the intervention of other research tools and techniques in order to help these biologists, mainly because the manual classification has become almost impossible.

Nowadays, Computer Science is the most requested tool. From the cooperation between computer scientists and biologists resulted a great number of disciplines specialised for the manipulation and analysis of biological data such as Bioinformatics and Biominer. The majority of these disciplines are inserted into the process of the knowledge discovery from databases [1]. There are many techniques and methods for the comparison and the classification of biological sequences such as BLAST, FASTA. They are based on the computation of the similarities between the sequences. For instance, Smith and Waterman developed a technique based on the

dynamic programming. These approaches reach to matrixes of scores such as PAM and BLOSUM. Some techniques are based on alignments between the sequences [2], whereas others are based on the hidden Markov Model: HMM [3], SAM[14] or HMMER[15]. Recently, several data mining techniques are used for the supervised classification. They are applied in the biological sequence analysis and especially in the protein classification. Various supervised learning algorithms are available (e.g. neural networks, nearest neighbours, decisions trees, etc.), the most popular one in the protein classification domain is the support vector machine (SVM) [16].

In this paper, we outline a new approach which is better consistent with the biological domain knowledge. The biological reality says that the protein families are characterized with patterns of different length. Biologists identify each protein family with a specific field. A field is a set of motifs that are dispersed throughout sequences of a given family.

Until now, we set the length of the motifs before the whole classification process. It is the principle of the n-grams descriptor extraction. In this case, the length n is necessarily a compromise between several constraints such as computational capability, the kind of relevant information captured, the number of obtained descriptors, etc. This constraint does not correspond to the biological domain knowledge. The idea is thus to go past this constraint by the construction of descriptors with varying length without significantly increasing the computational complexity. In this perspective, we develop a hierarchical approach where the length of the achieved descriptors is not constrained. In order to evaluate our method, on the one hand, we compare our results to a standard n-gram descriptor extraction where we set first the value of n to 3 (3-grams); and on the other hand, we compared our results to a trivial approach where we extract and set together all the n-grams with various values of n (n = 1, 2, etc.).

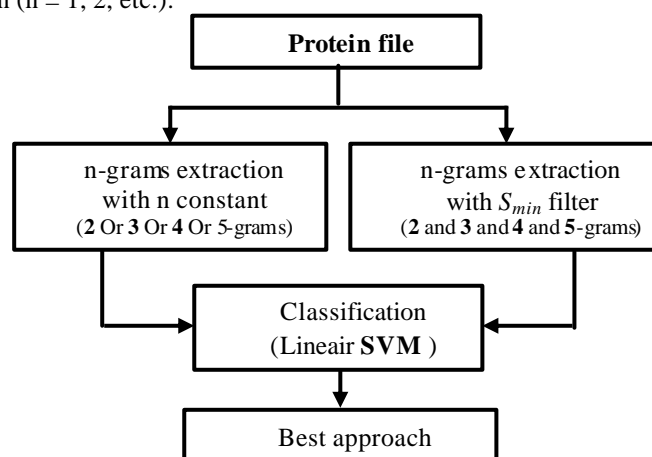


Fig. 1. Evaluation process of the tow n-grams construction approaches

This paper is organised as follows: In the second section, we present the protein classification problem. We outline the n-grams approach, especially the descriptor extraction and the construction of the learning set. In the third section, we explain the

hierarchical approach. Experiments and results are reported in the fourth section. We discuss these results. We conclude in the fifth and last section of this paper

2 The protein classification problem

The protein classification is one of the most important tasks of biologists. We want to propose a framework where the classification process relies mainly on the primary description of the proteins. A protein consists of amino acids. There are 20 kinds of amino acids. A protein sequence is thus a set of amino acids that have a given length.

Our process refers to the knowledge discovery process. The steps are well identified by Fayyad and al. [1]. We transpose the knowledge discovery in databases process into a knowledge discovery from biological data. This process involves three steps: the data preparation, including descriptor extraction and data cleaning; the data mining phase where we use the various learning algorithms, a supervised learning algorithm in our context; the validation and the deployment of the classifier e.g. classifying a protein into their family.

2.1 Data preparation

This step consists in selecting protein families and extracting the sequences from a data bank (SCOP). This step requires the intervention of a biologist, which is the domain expert. Then, we gather these sequences in files by grouping them according to their family membership, see figure 2.

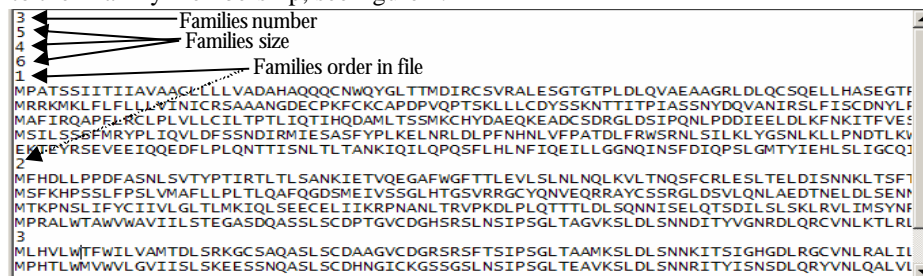


Fig. 2. Protein dataset

2.2 N-grams extraction

A protein sequence consists of a set of ordered amino acids. From a certain point of view, an amino acid can be considered as a character and a protein as a text. We then use text mining approach (text categorization approach) to extract descriptors [5]. These descriptors help us build a data table “proteins × descriptors”. The supervised learning algorithms are executed on this dataset.

Among the descriptor extraction techniques, we are especially interested in the n-grams extraction. A n-gram is a sub-sequence of n characters from a given sequence of characters. For any sequence, the n-grams set is obtained by moving a window of n -characters on the entire sequence. This moving is performed character by character.

In each moving, the sub-sequence of n amino-acids is extracted. The set of these sub-sequences build the n-grams that can be deduced from a sole sequence. This process will be repeated for all sequences. The algorithmic complexity of these n-grams extraction process is in $O(m \cdot n \cdot p)$ with n the size of n-gram, m the size of a sequence and p the number of sequences.

2.3 Data table construction

The obtained dataset T can have different forms, especially about the weighting values. The weighting consists in defining the way of filling out the table, which means the affected value $T(i,j)$ where i represents the i^{th} sequence and j represents the j^{th} descriptor or n-gram. There are many weighting types in literature. We can for instance cite:

- **Boolean weighting:** indicates whether a n-gram is present within a sequence or no.
- **Occurrence weighting :** indicates the number of occurrences of a n-gram within a sequence.
- **Frequency weighting :** indicates the relative frequency of a n-gram according to the number of 3-grams composing a sequence.
- **TF*IDF weighting:** corrects the 3 -grams frequency according to its frequency in a file.

In our context, we adopt a boolean weighting [7]. Our dataset will thus be a boolean table where 1 means the n-gram presence within a sequence, 0 its absence (Figure 4).

2.4 The drawbacks of this approach

In the first approach, the standard approach, we define at first the length “ n ” of the n-gram descriptors used in the subsequent learning algorithm. There are some drawbacks here:

- The best value “ n ” is not obvious. The chosen value is in reality a compromise, if “ n ” is too small, the retrieved information for the classification is often too poor; if “ n ” is too large, we obtain very specific descriptors, which are not useful for the majority of families.
- The number of potential descriptors grows exponentially with the length “ n ” of the n-grams. If we have 20 amino acids, the possible 2-grams number are $20^2 = 400$ and the possible 3-grams number are $20^3 = 8000$. Usually, the n-grams number is equal to 20^n . Two problems arise from this potential large number of descriptors: the computational time and the memory usage become a constraint; the resulting classifiers are affected by the “curse of dimensionality”, they overfit the learning set when the ratio between the number of descriptors and the number of examples is too large, they generalize poorly when we want to classify a new protein.

- The a priori value "n" may fit to discriminate some families but totally inadequate for other families. In this point of view, setting n as a parameter is not compatible with the biological reality.

As a trivial solution and to overcome these drawbacks, we can extract all n-grams with (n = 2, 3, 4, 5) lengths and use them to classify the proteins. We restrict "n" to 5 because, beyond this value, the computation is not possible on our Personal Computer. Indeed, if we count the descriptors extracted according to this trivial approach, we obtain in average about 72000 descriptors. Even if the computation is possible, learning a classifier in this very large representation space on a hundred or so examples is not realistic.

Between these two solutions, we propose an heuristic approach where the length of the descriptors is an outcome of the features' extraction algorithm. Both the computational time and the number of extracted features must be reasonable. In the following section, we present the hierarchical approach and we compare it to the others.

3 The hierarchical approach of descriptors extraction

In this section, we describes the hierarchical approach of the descriptors extraction, we compare it with the trivial approach.

3.1 Hierarchical principle

The hierarchical approach consists in building n-grams with different lengths. This construction is carried out in a hierarchical way, which means we extract (n+1)-grams descriptors from n-grams. It is ascending because the initial step is the extraction of the "valid" 2-grams, then we detect the "valid" 3-grams from these 2-grams, etc.

The key point of the process is the definition of the "valid" term. If all of the extracted 2-grams are valid without restriction, and all deducted 3-grams are valid, and so on, this corresponds to the trivial method where we get together all the n-grams with different values of "n". In our context, we use the frequent itemsets principle suggested by the association rule extraction algorithm. We define a minimum support S_{min} to filter the n-grams at each step. A n-gram is "valid" if the relative number of its apparition (i.e. the ratio between the number of proteins where it appears and the total number of proteins in the dataset) is larger than the minimum support. This restriction enables to master the amount of computation, we are confident with this assertion. But we hope also, this is less certain, only the experiments allow checking this one, which enables to remove the irrelevant (because they are too infrequent) descriptors. Figure 3 describes the n-grams hierarchical construction principle.

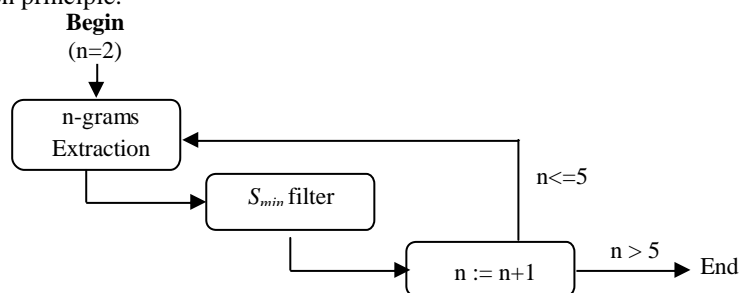


Fig. 3. n-grams hierarchical construction process

3.2 Frequent itemsets

Mining association rules is a privileged topic of the knowledge extraction from the data. The A-PRIORI algorithm based on the itemset support and the rules confidence is an efficient solution for the rule extraction problems [9]. The approach support/confidence consists in searching for association rules whose support and confidence go beyond fixed threshold by the user in preconditions, namely S_{min} and $Conf_{min}$. Here is an example taken from the marketing domain. Given R an association rule, $R : X \rightarrow Y (A\%, B\%)$ where $A\%$ points out that X and Y are present together in $A\%$ of the transactions (the support for the rule); and $B\%$ the customers who bought X have also bought Y (the confidence for the rule).

The rule extraction algorithm, which is based on the support-confidence checking, go through the trellis of itemsets to look for the frequent itemsets, of which support goes beyond S_{min} [8]. *A-PRIORI* is the most popular algorithm [9], it mainly consists of two main steps: We search the frequent itemsets i.e. the itemsets (a set of items, a set of product in the marketing domain, a n-grams in our protein classification context) of which support goes beyond S_{min} by sweeping up the trellis of itemsets within its width and by computing the frequencies with a counter within a base. This method requires a scan of the whole database for each trellis level.

For each frequent itemset X , we keep only the rules that have $X / Y \rightarrow Y$ as a type, with $Y \subset X$ of which confidence goes beyond the threshold $Conf_{min}$.

3.3 Frequent n-grams

In our context, we are interested in the first phase of the previous algorithm, that is, look for the frequent itemsets. We are going to adapt this approach in order to extract n-grams of varying length and which are frequent. The only parameter of the algorithm is the minimum support S_{min} : in the n-grams extraction phase, we remove the n-grams that have a frequency lower than S_{min} . The number of the descriptors and the length of each descriptor (n-gram) are outcomes of the algorithms.

We can thus present our problem in the following way. To discriminate two families F1 and F2, a relevant descriptor must be frequent in the first family and infrequent in the second one. When the descriptor is frequent for both families, it is not relevant for the classification task. For this reason, the minimum support S_{min} is defined on the families and not on the whole dataset. The question is: which is the family of reference used when we define the S_{min} parameter? A simple example enables to understand the alternative situations.

Take the example of the figure 3, let's take F1 and F2 two protein families, where N1 is the number of F1 sequences and N2 is the number of F2 sequences. We have N1=9, N2=4 and N1+N2=13. We set $S_{min}=50\%$. This value can be used in three different ways:

- 50% according to the file. In this case, we just save the n-grams that exist at least in 50% of the sequences, which means those that have a frequency ≥ 4.5 . As a result, the set of saved n-grams is {ng1, ng2, ng4}.
- 50% according to the most frequent family. In this case, we just save the n-grams that exist at least in 50% of F1. This refer to those that have a

frequency ≥ 6.5 . As a result, the set of saved n-grams is {ng1, ng2, ng4, ng5}.

- 50% according to the less frequent family. In this case, we just save the n-grams that exist at least in 50% of F2, i.e. those of a frequency ≥ 2 . As a result, the set of saved n-grams is {ng1, ng2, ng4, ng5, ng6}.

	ng1	ng2	ng3	ng4	ng5	ng6	ng7	class
seq1	1	1	1	0	0	0	1	1
seq2	1	1	1	1	1	0	0	1
seq3	1	1	0	1	1	0	0	1
seq4	1	0	0	0	0	0	0	1
seq5	1	1	1	0	0	0	0	1
seq6	1	1	1	1	1	0	0	1
seq7	1	1	0	1	1	0	0	1
seq8	1	0	0	1	0	0	0	1
seq9	1	0	0	0	0	0	0	1

seq10	0	1	0	1	0	1	0	2
seq11	0	0	0	1	1	1	0	2
seq12	0	1	0	1	0	1	0	2
seq13	0	0	0	1	1	1	0	2

Fig. 4. boolean data table

The main conclusion that we can draw from this example is that ng6 is eliminated by the first two hypotheses. However, it exists in 100% of F2's sequences and in 0% of F1's sequences. It is a very relevant descriptor because it perfectly discriminates the two families. From this observation, we adopt the third approach. At last but not least, we must define the right S_{min} 's value. In fact, this problem is resolved with experiments. The true value depends on the type of the data and the opinion of the domain expert.

3.3 Algorithms and theoretical comparison between the two approaches

<u>Classical algorithm</u>	<u>Hierarchical algorithm</u>
E : n-grams set with a single size	E : n-grams set with a single size
Fich : protein file	Fich : protein file
M : n-grams set with different sizes	M : n-grams set with different sizes
n : n-grams size	S_{min} : n-grams support
	n : n-grams size
Begin	Begin
M=NULL	M :=NULL
E=NULL	E :=NULL
n:=2	S_{min} := constante
	n:=2
do	do
E := Extract_n_grams (n, NULL, Fich)	E := Extract_n_grams (n,E,Fich)
M := M + E	E := Filter(E, S_{min})
n:=n+1	M := M + E
E=NULL	n :=n+1
While (n<=5)	While (n<=5)
return (M)	return (M)
End	End

Tab 1. CPU time(second) for n-grams extraction by the tow approaches(average of 10 files)

	Trivial Approach	Smin=30	Smin=50	Smin=70
Average	318	584	250	155

According to the table 1, we observe the influence of the parameter S_{min} . The computation time decreases when the parameter increases. When we set $S_{min} \geq 50$, the computation time is significantly small in regard to the trivial approach, which consists in extracting all the n-grams.

4 Experiments and results

4.1 Data and evaluation methods

Data bank. In order to evaluate the efficiency of this new approach, we used real biological dataset. We randomly draw 5 protein families from the SCOP database [10]. SCOP gathers different types of proteins family structures. The used classification is organised in several hierarchical levels: super families, families and folds. There are approximately 50 observations in each family. Our goal is to discriminate a family in regard to another family by using a supervised learning algorithm. We carry out a pairwise analysis i.e. we want to differentiate each pair of families. Thus, we build 10 datasets based on our 5 families. Each dataset contains about 100 observations.

Error rate evaluation and classifier. In order to estimate the prediction error rate, we use a 10 x 2 cross-validation that we repeat many times. This technique gives a rather good estimation [11]. The second problem is the classifier choice. We select a linear SVM (Support Vector Machine with a linear kernel) as a classifier. In a previous work, we carried out a comparative study of linear and non-linear classifiers such as RBF-SVM (Radial Basis Function kernel), CART Decision Trees, Naive Bayes classifier, nearest neighbor, etc [12]. The main result is that the SVM with a linear kernel is one of the most efficient classifier in our context. It seems that because we combine a restrictive representation bias (linear classifier) and learning bias (maximum margin principle), this classifier is particularly stable, with a very strong resistance to the overfitting. It was the main issue in our context where we combine a very high dimensionality and a small number of available examples.

4.2 Results

In this section, we will present the results in the following way:

1. We describe the results of the standard n-grams approach. We try various value of "n" and detect the best one (Table 2 and 3). It will be our reference in the examination of results, especially in order to evaluate the efficiency of the hierarchical approach.
2. We describe the results of the hierarchical method. We examine the impact of the S_{min} parameter on the number of descriptors obtained (Table 4).
3. Finally, we examine the results according to three ways:
 - We compare the standard 3-grams to the trivial approach where we set together all the 2+3+4+5-grams.
 - We compare these approaches to our hierarchical approach.
 - On the hierarchical approach, we inspect the impact of the S_{min} parameter on the accuracy of the resulting classifier. The underlying question is: is there an "optimal" value that we can infer on all protein families discrimination?

Standard approach: First, we want to mention that we limit the length "n" of n-grams to 5. Beyond this value, our computer is not operational because there are too many descriptors, it is crashed because a lack of memory.

Tab 2. average number of extract n-grams for 10 file of protein families couple

	2-grams	3-grams	4-grams	5-grams	Total (2+3+4+5)
Avg(F_XY)	399	6658	28432	37412	72902

The table 2 describes the number of n-grams (descriptors) for each value of n. These n-grams are built with the classical approach. In the last column, we gathered all n-grams between n=2 and n=5. The first conclusion from the table 2 is that the number of n-grams is large (except 2-grams), it rapidly increases with the n-value. The result is coherent; even if we do not reach the theoretical number of n-grams when n increases, the number of obtained descriptors is still large.

The right value of "n" remains generally an open question. Only the experiments can supply an answer, which is limited to the dataset and the classifier used. In our context, n=3 seems a good approximation. We may be confident to this indication because: we randomly draw the protein families of our study, in this point of view, the result may be extended to the other families; in a previous work where we used a very different classifier (a nearest neighbours classifier), n=2 gives also a valuable results [6].

Tab 3. Comparison between extracted n-grams with classical approach

	2-grams	3-grams	4-grams	5-grams
F12	0.0186	0.0198	0.0919	0.2314
F13	0.0511	0.0851	0.1096	0.1500
F14	0.0275	0.0242	0.0417	0.0600
F15	0.0583	0.0500	0.0667	0.1037
F23	0.0260	0.0240	0.0470	0.0970
F24	0.0086	0.0117	0.0367	0.0977
F25	0.0202	0.0193	0.0579	0.1281
F34	0.0590	0.0507	0.0769	0.0940
F35	0.0262	0.0393	0.0779	0.1008
F45	0.0291	0.0324	0.0405	0.0561

Hierarchical approach: As mentioned earlier, this approach uses a minimum support to filter the n-grams in the feature extraction process. The parameter enables to master the resulting number of descriptors. If the parameter is too restrictive, we obtain a very small number of descriptors; numerous relevant descriptors may be filtered out. If we set a permissive value, the resulting number of descriptors is very large. Numerous irrelevant descriptors will disturb the classification task. The S_{min} parameter setting plays an important role on the computation time and the accuracy of the subsequent classifier.

Let us also note that we limit the maximum length of n to 5 in our experiments. It is not an intrinsic limitation of the approach or of the used computer. We set this limitation in order to obtain comparable results with the other methods of this paper (standard and trivial approaches). In effect, we can also let pursue algorithm without limitation on the n-grams length ($n > 5$) until that one has no more valid n-grams (i.e. frequent). We test various number of S_{min} values (30,50,70).

In table 4 we present the average number of resulting descriptors in our 10 datasets according to the 3 values of tested S_{min} .

Tab 4. Evolution of n-grams average number with filtering (S_{min} =30%-50%-70%)

	2-grams		3-grams		4-grams		5-grams		n-grams ($S_{min,filter}$)
	Init	S_{min}	Init	S_{min}	Init	S_{min}	Init	S_{min}	
Avg(30%)	399	385	6583	1175	10450	199	580	119	1879
Avg(50%)	399	364	6425	385	3853	53	165	30	833
Avg(70%)	399	352	6272	138	1471	19	63	10	520

We observe several interesting results:

- The resulting number of descriptors of the hierarchical process is reasonable (Table 4), in comparison to the trivial approach.
- In a logical way, the larger is the S_{min} parameter, the smaller is the number of descriptors obtained.
- The larger is the length of the n-gram, the stronger is the filtering effect. We observe this phenomenon in the difference between the columns *Init* and S_{min} in table 4.

4.3 Comparison between the various approaches

Tab 5. Evaluation of n-grams set for the tow approaches

		Nb	Error rate
Standard approach	E_2	399	0,0324
Trivial approach	E_m	71761	0,0423
Hierarchical Approach	E_{m30}	1902	0,0309
	E_{m50}	848	0,0325
	E_{m70}	531	0,0322

Note that, E_2 is the set of 2-grams (standard); E_m is the set of a 2+3+4+5-grams without filtering (trivial); E_{m30} , E_{m50} and E_{m70} are the sets of descriptors obtained from the hierarchical approach with respectively $S_{min}=30$, $S_{min}=50$, $S_{min}=70$. Each set has two properties Nb and err_rate (Table 5), they represent the number of resulting n-grams and the error rate obtained with the classifier C-SVC (linear SVM).

E2 against Em: In this part, we compare the error rates of the obtained classifications with the E_2 set, and this which are obtained with the E_m set. E_2 shows a significantly better accuracy as $E_2(err_rate) \ll E_m(err_rate)$; moreover, the number of descriptors is really smaller $E_2(Nb) < E_m(Nb)/10$. Thus, with a smaller number of n-grams, we get better error rates. The standard approach with $n=2$ totally outperforms the trivial approach, in both the computation and the accuracy considerations. However, we are not satisfied with the results because we described in previous sections the drawbacks of being limited to one length of n-grams. The biologists expect descriptors with varying length.

Em against {Em30, Em50, Em70}: The proposed solution is a heuristic solution where we extract the n-grams and filter them in an hierarchical way by respecting the support S_{min} value. We varied the value of S_{min} between 30, 50 and 70. The table 5 shows that with the three values we always find excellent accuracy than the E_m set. Even the results of E_{m30} , E_{m50} and E_{m70} are also better than the E_2 set.

This result is particularly interesting in regard to the composition of the selected set of descriptors obtained. According to table 4, we observe that we have a mix of different descriptors. Using a feature ranking process we could associate the most relevant descriptors to the families in a detailed examination of the results with the biologists.

The results evolution according to E_{m30} , E_{m50} and E_{m70} : As a selected solution, we varied S_{min} between three values 30, 50 and 70. We noticed from the table 5 that $E_{m30}(err_rate)$, $E_{m50}(err_rate)$ and $E_{m70}(err_rate)$ are very similar, $E_{m30}(err_rate)$ is the best one in this experiment. Regarding the number of obtained descriptors, if we compare $E_{m30}(Nb)$, $E_{m50}(Nb)$, and $E_{m70}(Nb)$, we say that E_{m70} is the best one because the selected descriptors is smaller to E_{m30} . The main result is most of all that the process is not very sensitive to this parameter about the accuracy rate. This means we select S_{min} according to the treated problem, the demands of biologists, etc.

5 Conclusion

In this paper, we outline a new descriptor extraction approach in a protein classification context. This approach is named ‘‘hierarchical approach’’ and consists in building n-grams that have variable length to better respond to the biologists' needs. We compared this approach to the standard approach of the n-grams, where we define first the ‘‘n’’ value and later we extracted the n-grams with n-length.

Indeed, we used just one type of the n-grams. The results show that the accuracy of the new approach. The extracted n-grams have good classification rates. Moreover, the resulting descriptors number is reasonable. This enables to carry out the

construction of the supervised classifier in good conditions. A further important advantage is that the results are consistent with the biological domain knowledge. The extracted descriptors are compatible with the notion of patterns and the fields of the proteins' family.

We can improve these results. This can be realized by the reducing of the number of n-grams extracted by the hierarchical approach. Indeed, the hierarchical approach operates in a unsupervised way. There are certainly a large number of redundant descriptors regarding to their relevance in the prediction. Using efficient supervised feature selection process should be removing more descriptors without a deterioration of the accuracy [17].

References

1. Fayyad, U., Shapiro, G., Smyth, P.: From data mining to knowledge discovery : A overview, *Advances in Knowledge Discovery and Data Mining*, MIT Press (1996) 1--34
2. Gibas, C., Jambeck, P.: *Introduction à la bioinformatique*, Oreilly (2002).
3. Karplus, K., Barrett, C., Hughey, R.: Hidden Markov models for detecting remote protein homologies. *Bioinformatics* 14 (1998) 846-856
4. Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C.J.A., Hofmann, K., Bairoch, A.: The PROSITE database, its status in 2002. *Nucleic Acids Res.*, 30 (2002) 235—238
5. Sebastiani, F.: Machine learning in automated text categorisation, *ACM Survey*, V. 34, number 1, (2002) 1-- 47
6. Mhamdi, F., Elloumi, M., Rakotomalala, R.: Textmining, features selection and datamining for proteins classification, *IEEE/ICTTA' 04* (2004)
7. Mhamdi, F., Elloumi, M., Rakotomalala, R. : Descriptors Extraction for Proteins Classification, In *Proceeding of NCEI'2004*, New Zealand (2004)
8. Lallich, S., Teytaud, O. : Évaluation et validation de l'intérêt des règles d'association, n°spécial "Mesures de qualité pour la fouille des données", *Revue des Nouvelles Technologies de l'Information*, RNTI-E-1, (2004) 193-218
9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules, *Proceedings of the 20th VLDB Conference*, Santiago, Chile (1994)
10. Murzin, G.A., Brenner, E.S., Hubbard, T., Chothia, C.: SCOP, a structural classification of proteins database for the investigation of sequences and structures, *J. Mol. Bio.*, v.247 (1995) 536--540.
11. Dietterich, T.: Approximate statistical tests for comparing supervised classification learning, *Neural Computation journal*, v 10, n 7 (1999) 1895--1924
12. Rakotomalala, R., Mhamdi, F. : Évaluation des méthodes supervisées pour la discrimination de protéines, dans le proceeding de la conférence SFC'06, Metz (2006)
13. Cristianini, N., Shawe-Taylor, J. : *An Introduction to Support Vector Machines and other kernel-base learning methods*, Cambridge University Press (2000)
14. Eddy, S., Mitchison, G., Durbin, R.: Maximum discrimination hidden Markov models of sequences consensus. *Journal of Computational Biology* 2 (1995) 9-23
15. Krogh, A., Brown, M., Mian, I.S., Sjolander, K., Haussler, D.: Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology* 235(5) (1994) 1501-1531
16. Vapnik, V. : *The nature of statistical learning theory*, Springer-Verlag
17. Guyon, I., Gupta, H. : An introduction to variable and feature selection. *Journal of Machine Learning Research*, (2003) 157-1182