Learning Automata on Protein Sequences

François Coste and Goulven Kerbellec *

Symbiose project, IRISA/INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France
{francois.coste,goulven.kerbellec}@irisa.fr

Abstract: *Pattern discovery is limited to position-specific characterizations like Prosite's patterns or profile-HMMs which are unable to handle, for instance, dependencies between amino acids distant in the sequence of a protein, but close in its three-dimensional structure. To overcome these limitations, we propose to learn automata on proteins. Inspired by grammatical inference and multiple alignment techniques, we introduce a sequence-driven approach based on the idea of merging ordered partial local multiple alignments (PLMA) under preservation or consistency constraints and on an identification of informative positions with respect to physico-chemical properties . The quality of the characterization is asserted experimentally on two difficult sets of proteins by a comparison with (semi)-manually designed patterns of Prosite and with state-of-the-art pattern discovery algorithms. Further leave-one-out experimentations show that learning more precise automata allows to gain in accuracy by increasing the classification margins.*

Keywords: Pattern Discovery, Grammatical Inference, Proteins.

1 Introduction

Pattern Discovery. Proteins are essential to the structure and functions of all living cells and viruses. They are amino acid chains that fold into three-dimensional structures but most of the time only amino acid chains – a sequence over 20 letters each representing one amino acid – are available. Given the rapidly growing amount of available sequences in the databases, produced in particular by DNA sequencing projects, prediction of the structure or the function of proteins from their sequences is one of the major challenges in molecular biology.

One successful approach to assist the biologist in this task, is to define signatures of known *families* of biologically related proteins (typically at the functional or structural level). Signatures usually identify conserved regions among the family of proteins, revealing the importance for the function of their structural or physico-chemical properties. A representative example of this approach is the well-known Prosite database [15], gathering protein sequence patterns and profiles for a large number of families.

Prosite's patterns are restricted regular expressions while profiles (or weight matrices, the two terms being used synonymously) are tables of position-specific amino acid weights and gap costs. Among the other types of signatures which may be found in the integrated database of proteins signatures InterPro [36], profile-HMM [11] are probably the most widely used. Profile-HMM, like Prosite's profiles, are position-specific scoring systems but are enhanced with the addition of insertion and deletion states. Prosite's patterns and their statistical counter-part, the profile-HMMs, may be considered as

* supported by a PhD research grant from Région Bretagne.

the more expressive signatures [6] currently used for characterization of proteins, respectively in the class of exact patterns (returning directly whether the sequences are accepted or not) and probabilistic patterns (returning acceptation probabilities of the sequences).
Even if in Prosite these signatures are still defined essentially by experts on the basis of multiple sequence alignments, automatic discovery of such signatures (pattern discovery) is a dynamic research area [29].Among the state-of-the-art algorithms learning expressive patterns, Pratt [17], EMotif Maker [25], Teiresias [28] or Splash [7] have been shown to generate successfully Prosite-like patterns. Concerning stochastic models, commonly used tools such as HMMER [30] and SAM [18] are able to estimate the parameters of the simple architecture of the profile-HMM. Thus, to obtain a high degree of accuracy for the classification of proteins, stochastic models need to use elaborated adequate weighting schemes [11].
The major limitation of all these patterns is that they are restricted to *position-specific* characterizations: as a matter of fact, neither relations between positions (for instance, distant correlated amino acids in contact in the three-dimensional structure) nor alternative paths (disjunction over more than one position) can be expressed in these models, whereas it could be done in true regular expressions, automata or formal grammars.

Towards grammatical inference. Searls [32] has advocated the benefit of viewing the biological sequences as sentences derived from a formal grammar. This allows not only to overcome the position-specific characterization of the sequences, but also to benefit from the explicit modelling provided by grammars. Trying to parse a sequence using a grammar gives a prediction, e.g., whether a sequence belongs to a particular family, but a successful parse provides also information about the sequence: the semantics associated with the derivation, e.g. the reason why a sequence belongs to a particular family. A survey on learning grammars (Grammatical Inference) for biological sequence analysis has been published recently by Sakakibara [31]. The results presented are mainly on the estimation of probability parameters of stochastic grammars while the problem of learning the structure of grammars remains a difficult task with a few positive results on biological sequences. Concerning DNA sequences, one can mention the application of Sequitur [24] to infer a hierarchical structure on human genome (but without generalization) and the application of ADIOS to the genome of *Caenorhabditis elegans* [33], both algorithms showing good compression ratios. Although the algorithm can be applied to proteins, the authors have not published yet results of the application of ADIOS to proteins sequences (the authors present, instead, an experiment of classification by SVM using sets of extracted motifs). Concerning the application of such methods for the characterization of proteins, we are only aware of the early work of Yokomori [35] on learning locally k-testable languages for the identification of protein α-chain regions by using a subclass of automata, which may be linked to *n-grams* and to persistent splicing systems. Locally k-testable languages are languages for which it is sufficient to parse subsequences of length k to decide whether a sequence is accepted or not. This method is thus restricted to local characterization of length k, which has furthermore to be fixed usually to a small value to avoid over-specialization by the inference algorithm.

Learning automata. We introduce in this paper a sequence-driven approach initiated in [10,9] to learn automata on proteins. Inspired by grammatical inference and multiple alignment techniques, the approach relies on the idea of *merging ordered partial local multiple alignments*.
Local multiple alignments are commonly used. They may be defined as the alignment of a significantly conserved region of a set of sequences. In pattern discovery, many tools, such as Gibbs Sampling [20], MEME [4] or Model [14], have been proposed to identify ungapped local multiple alignments. The purpose of these algorithms is to find the best alignment of the sequences on a given length l without indels. Since the classical setting is biased toward aligning all the sequences, we introduce the term of *partial* local multiple alignments (PLMA) to designate those which are not required to involve all the sequences of the set. The interest of PLMAs is that they are more likely to represent

strongly conserved regions. *Merging* a PLMA is defined precisely hereafter. Roughly speaking, this operation allows to obtain a local consensus on a subset of the sequences. By merging successively PLMAs, we will be able to build automata representing complex succession of local consensus.
Our approach can be summarized as follows in a two-phase algorithm: first, a *characterization* stage detects and orders representative PLMAs, then a *generalization* stage merges successively the PLMA candidates under preservation or consistency constraints to identify globally conserved areas in which only informative positions are conserved. The approach is sketched in figure 1 and its implementation for learning automata on proteins is detailed in the next sections. A first validation of the approach by leave-one-out experiments on difficult families of proteins is presented in the last section.

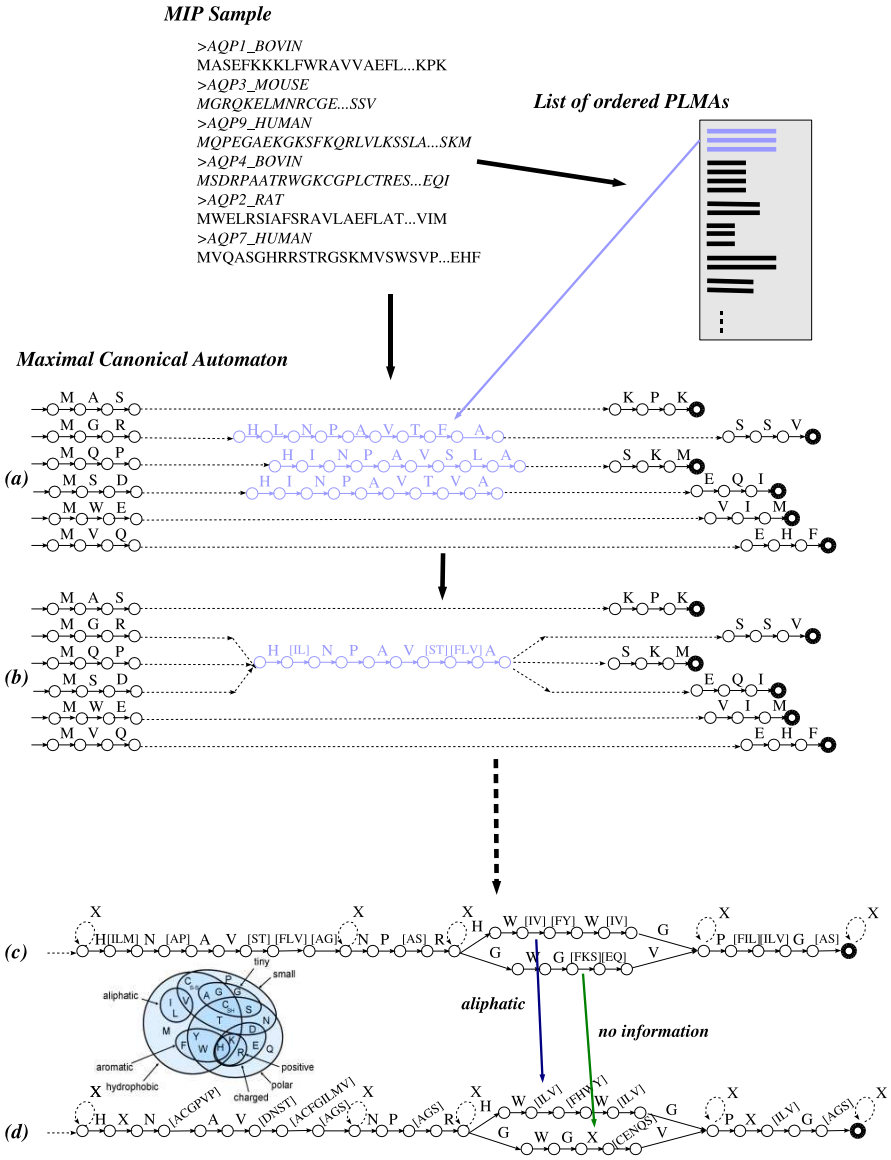

Figure 1. Main scheme of merging ordered PLMAs approach. The list of ordered PLMA is built during the characterization phase. During the generalization phase several automata are built: (a) Maximal Canonical Automaton, (b) result of merging the blue PLMA, (c) result of merging all the PLMAs and merging non representative positions, (d) Automata returned by the method after the identifi cation of physico-chemical positions according to Taylor's Venn diagram [34].

2 Characterization

In this section, we focus on the detection of the representative PLMAs of a set of protein sequences and on scoring them. The first part of the problem may be stated as that of finding PLMAs with high similarity but on enough sentences and on a sufficient length in order to be significant. The second part of the problem consists in defining a score that allows to compare two PLMAs, ideally even if they are of different length and involve a different number of sequences. Since there is no standard way to tackle these problems, many approaches can be proposed. We have chosen to privilege the reduction of the number of parameters by developing an approach based on significantly similar fragments pairs (SFP) as in the multiple alignment tool DIALIGN2 [23].

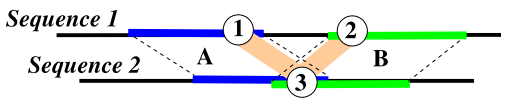

Figure 2. Incompatible SFPs: by merging SFP A with SFP B, position 3 would be merged with two position 1 and 2 of the first sequence.

The term protein *fragment* designates here a contiguous subsequence of a protein and we consider protein fragment pairs such that both fragments have the same length. The similarity of such a pair is the sum of the individual similarity values (given by a substitution matrix) of the facing amino acids. Difficulty in comparing the similarity of two different length fragment pairs is a well known problem. To overcome this problem, DIALIGN2 [23] uses sets of *significantly similar fragment pairs* (SFPs): DIALIGN's first step consists in finding all fragment pairs such that their similarity is significantly larger than expected on random sequences (as measured by a weight function $w(s, l)$ related to the probability of finding any fragment pair of length l with a score at least as large as s taking into account the lengths of the protein sequences). In DIALIGN2, these SFPs are then combined to make a multiple alignment optimizing the global sum of weights under consistency constraints.

Similar Fragments Pairs Characterization.

Algorithm 1 SFP characterization.

Require: a set P of SFP.
Result: ordered list L of PLMAs s.t. each one is a SFP.
  for each $p \in P$ **do**
    COMPUTE_SCORE(p) ▷ support, implication index ...
  while $P \neq \emptyset$ **do**
    $p \leftarrow$ BEST_SCORE_SFP(P)
    L.APPEND(p)
    $P \leftarrow P \setminus (\{p\} \cup \{q \in P \mid q$ incompatible with $p\})$
  return L

We propose here to first consider SFPs as being PLMAs of the simplest type (i.e. involving only two sequences) and to rank them according to their representativity. Ordering the SFPs is necessary for choosing between two *incompatible SFPs*. Two SFPs are incompatible if a position of one sequence is aligned by each SFP to two different positions in another sequence (see figure 2). We name this constraint *preservation constraint* since merging both SFPs in this case would merge on itself a characteristic fragment and thus loose the detected characterization. A pseudo-code for the SFP characterization is given in algorithm 1.

Different scores can be used for ordering the SFPs. To order the SFPs with respect to their representativeness of the whole family, we propose to estimate their *support* in other sequences of the family, *i.e.* we count for each SFP the number of sequences containing a fragment sufficiently similar to it. Several criteria can be chosen to decide whether a fragment is similar to two other ones. Let us note that transitivity does not hold for similarity. We use the triangular inequality since it is simple, robust and parameter-free. To simplify the expressions, we use $w(f_1, f_2)$ instead of $w(s, l)$ to designate the DIALIGN2 weight of a fragment pair $p = (f_1, f_2)$ having similarity score s and length l. A SFP (f_1, f_2) is said to be *supported* by a fragment f if: $w(f, f_1) + w(f, f_2) \geq w(f_1, f_2)$. A SFP is said to be supported by a sequence if it is supported by at least one fragment of the sequence. Let p be a SFP. We define the *support* of p in a set of sequences S as the number of sequences supporting

p in S, denoted by $\sigma_S(p)$. Hereafter, we denote $Support_S(p)$ as the set of sequences included in S supporting p.
More elaborate indexes based on the support may also be constructed. In particular, if a set of proteins known not to belong to the family is available (we will denote this set by N), we propose to rank the SFPs according to how discriminative they are, *i.e.* how their support in the family implies their proportion to be supported in the family and in the other set of sequences. To achieve this goal, we compute an *implication index* for each SFP p based on [22], denoted by $\iota(p)$: $\iota(p) = \frac{-P(Support_N(p))+P(Support_S(p))\times P(N)}{\sqrt{P(Support_S(p))\times|N|)}}$ where $|X|$ denotes the cardinality of a set X and $P(X)$ denotes its proportion with respect to S and N: $P(X) = \frac{|X|}{|S|+|N|}$. This formula is a normalized evaluation of how the support of the SFP in the family implies its proportion to be supported in the family and in the other set of sequences.
In the following, ordering the SFPs according to their support (resp. their implication index), and according to weights of the SFPs in case of tie score, will be referred to as the *support heuristic* (resp. *implication heuristic*).

Clique Characterization. From the SFP characterization presented in the previous section, PLMAs involving more than two sequences will be progressively built by merging the SFPs. One can be faced then with the situation of merging to SFP (f_1, f_2) and (f_2, f_3) such that the fragment pair (f_1, f_3) is not significantly similar. This reasoning can be extended to chains of SFPs. The PLMA corresponding to such single-linkage like merging of SFPs are likely to present a weak conservation. In order to have more homogeneous PLMAs, we propose here an alternative characterization based on cliques of SFPs, *i.e.* considering only PLMAs such that each pair of fragments involved in the PLMA is a SFP and by merging only non intersecting PLMAs.

The clique approach allows us to restrict the number of candidate PLMAs; we are then able to propose a practical algorithm performing an exhaustive search of cliques for decreasing target sizes. The pseudo-code is given in algorithm 2 where I_1, I_2, I_3 designate respectively incompatible, included and interfering fragment pairs with C. Let us note that while the maximum clique search is NP-Complete, in practice, the number of large cliques is small and finding a large clique may drastically reduce the number of remaining fragment pairs (especially with the inconsistency constraints introduced in section 3). Moreover, beginning by the most divergent sequences is a good heuristic to reduce the visited search space.

Algorithm 2 Clique characterization.

Require: a set P of SFP, a set S of sequences.
Result: ordered list L of PLMAs s.t. each one is a clique of SFPs.
$z \leftarrow |S|$ ▷ target size of the clique
while $z > 1$ **do**
  $\mathcal{C} \leftarrow \{C = \{f_1, ..., f_z\} \mid \forall (f_i, f_j) \in C, (f_i, f_j) \in P\}$
  for each $C \in \mathcal{C}$ **do**
    COMPUTE_SCORE(C) ▷ classically: $\Sigma_{p\in C} w(p)$
  while $\mathcal{C} \neq \emptyset$ **do**
    $C \leftarrow$ BEST_SCORE_CLIQUES($\mathcal{C}$)
    L.APPEND(C)
    $I_1 \leftarrow \{p \in P \mid \exists q \in C, p$ incompatible with $q\}$
    $I_2 \leftarrow \{(f_a, f_b) \in P \mid \exists f_i, f_j \in C, f_a \subset f_i, f_b \subset f_j\}$
    $I_3 \leftarrow \{(f_a, f_b) \in P \mid \exists f_i \in C, f_a \cap f_i \neq \emptyset, \forall f_j \in C, f_b \cap f_j = \emptyset\}$
    $\mathcal{C} \leftarrow \mathcal{C} \setminus (\{C\} \cup \{C' \in \mathcal{C} \mid C' \cap (I_1 \cup I_2 \cup I_3) \neq \emptyset\}$
  $z \leftarrow z - 1$
return L

3 Generalization

Characterization allows to find representative PLMAs of the sample set: it allows to identify the important positions in the sequences with respect to the sample. We will see in this section how to use this characterization to build an automata modeling the family by generalizing the sample. The goal is to build a model allowing to recognize new members of the family (with a good accuracy) but also to get an explicit model of the family allowing to gain new insights on the sequences.

PLMA Merging. This first generalization step applies the classical state-merging scheme popularized by RPNI [26] and EDSM [19] to PLMA. We consider the more general case allowing to learn non-deterministic automata. Following the definitions of [8], to which we refer the reader for details, the general sketch of this kind of algorithm is to first construct an automaton, named *maximal canonical automaton* (MCA) representing exactly the training set of sequences and then to generalize the recognized language by *merging* (unifying) some of its states.

To define PLMA merging, we first extend state-merging to *local alignment merging*. Let us first remark that since MCA represents exactly the training set, one can define a one-to-one function from the amino acids positions to the corresponding transitions in MCA. The aligned positions of a local alignment determine thus a set of pairs of transitions. We define the local alignment merging procedure as merging, for each corresponding pair of transition, the two target states together and the two source states together. *PLMA merging* can then be defined as merging its pairs of local alignments (see example in figure 1). The pseudo-code of the PLMA merging generalization procedure is given in algorithm 3. It proceeds by merging successively all the PLMA in the ordered list unless they are not compatible with a previous merge. Of course, compatibility tests can be omitted if they have already been performed in the characterization phase. One can choose between two kinds of compatibility constraints: preservation constraints and inconsistency constraints. Preservation constraints (see figure 2 and section 2) aim at preserving the structure of previous merges: this can be done simply by memorizing after each PLMA merge that the resulting states should not be merged together. Consistency constraints are the classical constraints used in sequence alignment [23] and can be handled efficiently using the GABIOS library [1]. Preservation constraints are weaker than consistency ones (*i.e.* consistency implies preservation) but preservation constraints allow PLMA crossing whereas consistency constraints do not.

Algorithm 3 PLMAs Merging generalization.

Require: sample sequence set S, ordered list of PLMA L.
Result: Automata A.
  $A \leftarrow MCA(\mathrm{S})$ ▷ Maximal Canonical Automata
  for each $plma \in L$ **do**
    if COMPATIBLE($plma$) **then** MERGE(A, $plma$)
  return A

Merging non representative positions.

Algorithm 4 Merging non representative positions.

Require: Automata A, for each state s of A the number n_s of states of MCA merged on s, quorum q.
  DETECT_EXCEPTION_PATHS(A)
  Mergeable $\leftarrow \{s \in A \mid n_s < q$ and $s \notin$ exception path $\}$
  for each $s \in$ Mergeable **do**
    for each state t adjacent to s in A **do**
      if $t \in$ Mergeable **then**
        MERGE(A, s, t)
  return A

After merging PLMAs, some positions may be involved in no merges. These localizations are clearly not representative of the family. We propose to treat them as "gaps". The idea can be extended to regions not involved in a sufficient number of sequences to be considered as being representative: if we introduce classically a quorum parameter, the generalization procedure could be stated as follows: if a state is used by fewer sequences than specified by the quorum, it is merged with its neighbours. The purpose of this generalization is to keep only the identified characteristic regions. Several variations around this merging scheme could be implemented. Statistical information like the length or the amino acid composition of the gap could also be considered and added to the model. We name *characteristic path* a set of adjacent states used by at least as many sequences as specified by the quorum. The corresponding positions will be named *characteristic PLMA*.

The problem with such a procedure is that characteristic paths may be bypassed by merging an outlier protein. More generally, an *exception path* is defined to be a path allowing to bypass a characteristic path (see figure 3). This kind of path should not be treated as a gap but rather as an exception and should be kept as it is, or withdrawn from the automata. The pseudo-code for handling gaps is given in algorithm 4.

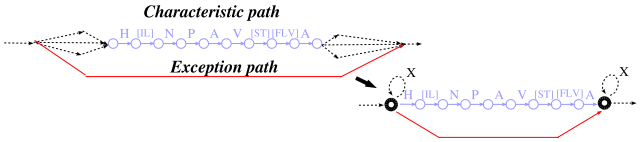

Figure 3. Exception example and result of algorithm 4.

Identification of physico-chemical properties.

The general substitution matrices used so far for the definition of SFP are estimated from large sets of close proteins and thus reflect only average similarity (over various contexts involving different physico-chemical properties of the amino acids). We propose here to use these localizations as contexts to recover the important physico-chemical properties of the amino acids with respect to the function or the structure of the family.

Algorithm 5 Identification of physico-chemical properties.

Require: Automata A, set of a.a. groups $\mathcal{G}$, thresholds $\lambda_G, \lambda_\Sigma$.
  for each $(q_1, q_2) \in A$ **do**
    $P \leftarrow \{\, a \in \Sigma \mid (q_1, a, q_2) \in A\}$
    if $P \neq \emptyset$ **then**
      $G \leftarrow$ SMALLEST$(\{G \in \mathcal{G} \mid P \subseteq G\})$
      if $LR_{G/P} \geq \lambda_G$ **then**
        for each $a \in G \setminus P$ **do** ADD$(A, (q_1, a, q_2))$
      else if $LR_{\Sigma/P} \geq \lambda_\Sigma$ **then**
        for each $a \in \Sigma \setminus P$ **do** ADD$(A, (q_1, a, q_2))$
  return A

The approach takes as input a set $\mathcal{G}$ of eventually overlapping substitution groups representing important physico-chemical properties (typically the groups proposed by Taylor see figure 1). The sketch of a naive identification would be to test for each set of amino acids P aligned by the approach if it is equal to one of the given groups. This approach may be applied only to small groups [25], or else it will require a large amount of training sequences to identify all the important groups (consider for instance the probability of aligning all the 13 hydrophobic amino acids in a small set of homologous proteins). We propose to use a statistical test to decide if the multi-set P has been generated according to a physico-chemical group G or not (see algorithm 5). Given two states q_1, q_2 of the automata, let P be the set of all the amino acids allowing to reach q_2 from q_1 and let n be the total number of sequences using these transitions. We decide to replace the current set of amino acids P by the smallest physico-chemical group G including P based on the result of a likelihood ratio test. To compute this ratio, we use the background probability p_a of each amino acid a and we estimate the probability $p_{a|G}$ of this amino acid given that it belongs to G by $p_{a|G} = c_G p_a$ where c_G is a proportional redistribution factor of the missing amino acids: $c_G = \frac{1}{\sum_{a \in G} p_a}$. In that setting, we can compare the likelihood L_G of G when n amino acids are drawn from G to its likelihood when the amino acids are drawn from P by the ratio: $LR_{G/P} = \frac{L_G}{L_P} = \left(\frac{\sum_{a \in P} p_a}{\sum_{a \in G} p_a}\right)^n$. Given a threshold λ_G, we test the expansion of P to G and reject it when $LR_{G/P} < \lambda_G$. If the expansion to G is rejected, there is no evidence of a physico-chemical property in the set of amino acids P. In such cases, one may wonder whether the amino acids in P have been generated randomly and then replace the set by the whole alphabet Σ, or whether the composition of the set is important and should be kept as it is. By replacing G by Σ and introducing the threshold λ_Σ, we test in a similar way the expansion of P to Σ by rejecting it when $LR_{\Sigma/P} = \frac{L_G}{L_P} = (\sum_{a \in S} p_a)^n < \lambda_\Sigma$.

4 Experiments

In this section we present a first validation of the approach on proteins. Our approach has been implemented in a program named Protomata-Learner. In the following, Protomata-PL will refer to the version using the similar fragment pair characterization while Protomata-CL will refer to the clique characterization. We used DIALIGN2 with the options `-nta -thr 5 -afc` for Protomata-PL and `-nta -thr 0 -afc` for Protomata-CL (the restriction to cliques allows to consider all the significant SFPs). Identification of physico-chemical properties were performed with the sets of physico-chemical properties proposed in figure 5 of [34], except the "unions" group[1], and $\lambda_G = 10^{-7}$, $\lambda_\Sigma = 10^{-19}$. Even with our unoptimized code, the execution never exceeded 10 minutes on a 3GHz desktop station.

4.1 MIP Family.

The Major Intrinsic Protein (MIP) family [12] is a family with a high level of similarity defined according to functional and structural properties. MIPs are transmembrane channels, well-known to be important for water, alcohol and small molecules transport across cell membranes thanks to P. Agre (Nobel Prize in Chemistry "for the discovery of water channels", 2003). The protein sequence database UNIPROT, contains 911 proteins annotated as being members of the MIP family. Of these 911, 159 protein sequences (denoted hereafter by the set T) are in SWISSPROT which is the annotated public reference database used by Prosite. Out of this set, a biology expert has identified only 79 sequences with a real biological experiment-based annotation (a lot of proteins being annotated "by similarity"). By filtering out the sequences with more than 90% of identity [2], this set was then reduced to 44 sequences (set M). Out these, the expert has identified 24 water-specific sequences (set W+) and 16 glycerol or small molecule facilitator sequences (set W-). Let us notice the difficulty of the discrimination task between these MIPs, some sequences of W+ being closer to some sequences of W- than to the other sequences of W+. We have established also a control set composed of sequences close to MIP sequences (first Blast hits) and identified by the expert as being outside the family (set C).

First Common Fragment Characterization. For this first set of experiments, in order to be able to compare our approach with Pratt [17] and Teiresias [28] methods and Prosite hand-made pattern, we restricted Protomata-PL to return only the first common fragment shared by all sequences, using support index. Pratt and Teiresias were used with their default parameters, except the parameter W (maximum length) of Teiresias that was set to 50 to allow longer patterns to be discovered. The patterns were learned from the set M and parsed on the sequences of the set T.
A scan of the Prosite's pattern on SWISSPROT database returns false positive as well as false negative sequences with respect to T (while T was used to define Prosite's pattern).

Method	Precision	Recall	F-meas.
Prosite (reference)	**0.95**	**0.91**	**0.93**
Pratt	0.90	0.78	0.83
Teiresias	0.23	0.89	0.37
Protomata-PL	**1**	**0.87**	**0.93**

Table 1. Precision, recall and F-measure on T for 4 MIP patterns.

Table 1 summarizes the results of such scans for the three patterns. The recall is the ratio of the number of relevant records retrieved to the total number of relevant records. The precision is the ratio of the number of relevant records to all the documents retrieved. The traditional F-measure is $\frac{2 \times Precision \times Recall}{(Precision + recall)}$. The recall of our approach is close to Prosite's pattern recall while our precision remains at 1. Let us notice that in our false negatives, one was not a full sequence and 16 were annotated as MIP only by similarity. When comparing our approach to Pratt and Teiresias pattern discovery tools, the comparison is clearly in favor of Protomata-PL with respect to both the precision and the recall. Let us note that the three patterns focus on the

[1] Identifying two alternative properties were not likely to be interesting here.

same site, the so-called NPA box. Our pattern is much longer than the other patterns. It contains also some positions of alpha helices turned to the channel which are likely to be important for the structure and function of this family. In the next paragraph we focused on a more precise characterization (combining several characteristic regions) of a subclass of the MIP family with the help of counter-examples.

Sub-Families Discrimination.

In this second set of experimentations, we focused on the characterization of the water-specific MIP subfamily set W+, using the set W- as counter-example. This discrimination task is motivated by a better understanding of the transport of these molecules. We used it to study the quality of the characterization on closely related sets of sequences at increasing specificity levels. Due to the small number of available sequences, a leave-one-out cross-validation scheme was used to evaluate our approach. For each pair of positive and negative sequences $(w+, w-)$, the training was achieved using the remaining sequences of W+ and W-. For each leave-one-out datasets, several automata – ranging from short automata (like in the previous paragraph) to larger automata characterizing almost all the length of the MIP topology – were obtained by using an increasing number of PLMA. Each automaton was then evaluated according to the distance for acceptation of the positive sequence left out $w+$, the negative sequence left out $w-$, and also of the closest sequence c in the control set C. The *distance for acceptation* (or error correcting cost) is defined as the minimal cost of amino acid substitutions needed in the sequence for its acceptation by the automaton (the cost of each amino acid substitution being given by the classical substitution matrix Blosum62 [13]). Figure 4 presents the results of all these experiments when using the implication index and a quorum of 100%. On the size axis, we highlighted 4 attraction points which are related to the progressive emergence of common sub-patterns, the first one corresponding to the first common fragment. The separation of the different sets of sequences is manifest[2] and grows along the automata size axis until an inflexion point near 100 states. Behind this inflexion point, the merged SFPs do not contribute anymore to the discrimination but only to a more precise characterization of the MIP family without showing over-generalization evidence.

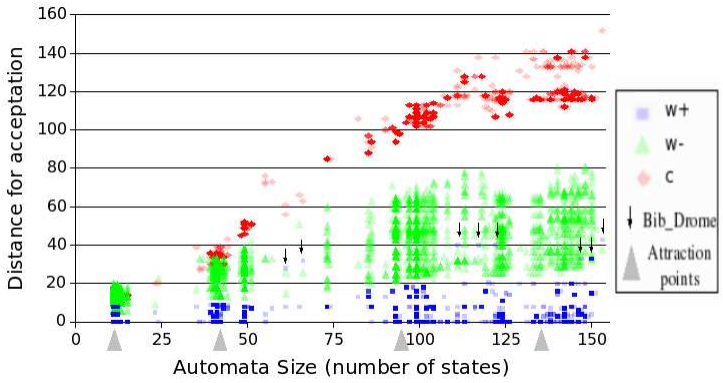

Figure 4. Distance of each test sequence for acceptation by automata when using the implication index. The set to characterize was the Water-Specific W+ set with W- as counter-example set. Set C was a non-MIP control set, only the smallest distance was reported on the plot for C.

Table 2 sums up the results of the automata at the attraction points for the classification task between W+ and W-, with strict parsing acceptation and with a distance threshold acceptation. In the latter case, the distance to the automata of the closest counter-example in the training set, was taken as the threshold distance for acceptation. The approach was then able to raise 100% of precision and 100% of recall for automata sizes ranging from 40 to 100 states.

Automata Size	Strict Parsing			Threshold Parsing		
	Prec.	Recall	F-meas.	Prec.	Recall	F-meas.
10	1	0.92	0.96	1	0.96	0.98
40	1	0.71	0.83	**1**	**1**	**1**
100	1	0.54	0.70	**1**	**1**	**1**
130	1	0.42	0.59	1	0.96	0.98

Table 2. Performance on classifi cation task (W+ vs W-).

[2] Only one sequence from set W+, which is called Bib_Drome is sometimes plotted at the level of the usual distance of W- sequences. Bib_Drome is known to be divergent from the other MIPs and it is not surprising if more substitutions were needed to parse this sequence when no other representative of this family were available in the training set. Nevertheless, this distance needed to parse this sequence was always smaller than the one needed to parse sequences outside the family (from W- or C).

4.2 TNF Family.

The Tumour-Necrosis Factor (TNF) family [3] is included in the cytokine super-family. Playing the role of ligands in the signal network of apoptosis, these proteins are studied in particular for their implication in cancer diseases. Each TNF protein is a combination of beta-sheets. Contrary to MIPs, the sequence divergence in the family is very high. The positive set is made of the 18 human sequences which are the most representative sequences of this partially known family. The average percentage of identity [2] in the positive set is 33,6% with a minimum of 0% and a maximum of 71%. The negative test set contains the 4 false positive hits of the Prosite pattern plus 16 cytokines members known to be outside of the TNF family. The average percentage of identity between positive and negative sequences is 28,56% with a minimum of 0% and a maximum of 81%.

Method	Precision	Recall	F-measure
Strict Parsing			
MCA	0	0	0
Prosite	0.75	0.67	0.71
Teiresias	0	1	0
Pratt	0.85	0.94	0.89
Protomata-PL Q=17	0.88	0.89	0.88
Threshold Parsing			
Pratt	0.86	1	0.92
Protomata-CL Q=7	0.96	0.94	0.95
Protomata-CL Q=6	**1**	**1**	**1**
Protomata-CL Q=5	1	0.94	0.97

Table 3. Comparison of Protomata-CL to other methods on the TNF family using a leave-one-out test. Q is the minimum value of the quorum.

Table 3 reports the results of each method for leave-one-out experiments on these sets. As expected, the Maximal Canonical Automaton (corresponding to rote learning) showed bad results. We ran Teiresias, but it always got a poor overgeneralized motif. Pratt and Protomata-PL (with 100% quorum) obtained both a good F-measure by being able to characterize the same localization than Prosite's hand-made motif. But this localization did not allow to reject the false positive sequences of Prosite. The best results were obtained by Protomata-CL with a quorum of 6 and by using threshold acceptation. For each step of the leave-one-out, we computed the error correcting distance of the positive test sequence and the negative test sequences to the automaton. We fixed the threshold acceptation to be 3/4 of the distance to a reference counter-example sequence (SWISSPROT ID Q92838) [3].

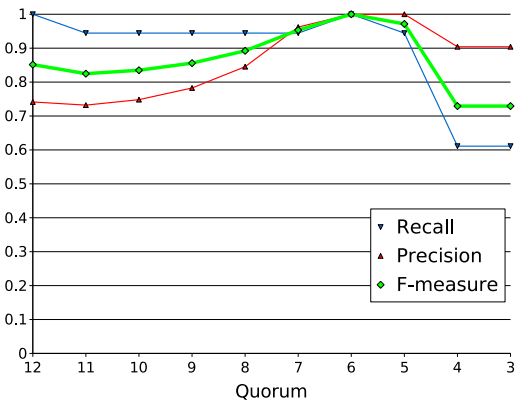

Figure 5. Impact of quorum on F-measure for Protomata-CL on TNFs.

Figure 5 shows a more detailed view of the precision, recall and F-measure according to the quorum for Protomata-CL. The largest cliques found by Protomata-CL were of size 12. The automata built with these first cliques were not perfect; in fact the recall was good but the precision was not so good. But, while the quorum decreased from 12 to 6, the precision increased. It can be explained by the fact that more informative positions that characterize subparts of the family were discovered and were taken into account in the automata. The recall decreased rapidly for a quorum under 5, this indicates that the fragments used to build the automata were no more representative of the family. Before the quorum of 7, automata were overgeneralized, and after the quorum of 5 they were over-specified. We can see a perfect point at the quorum of 6 with 100% of recall and 100% of precision.

[3] This sequence was chosen by picking up the the most representative TNF member (SWISSPROT ID P57369) and by selecting its closest sequence (using blast on NCBI server) with clear annotation allowing to conclude that the sequence was not a TNF.

5 Conclusion

This study shows that good automata can be learned successfully on proteins. Our approach is inspired by grammatical inference and multiple alignment techniques. It relies on fragment similarity to identify locally conserved regions and then refine eventually the characterization by identifying informative positions. This fragment-based sequence-driven approach allows to identify either conserved fragments without strong amino acids conservation (for instance structural elements of the protein like helix or regions with particular physico-chemical bias) or important positions (for instance one position involved in an active site) under the assumption that their neighborhood shows also some conservation (for instance to ensure the correct 3D localization of the important position). By introducing preservation or consistency constraints, we are able to build explicit models (automata in this study) of the family linking and generalizing the identified representative conserved regions, with good prediction accuracy asserted by leave-one-out cross-validation.
Many improvements can still be introduced in the approach. Many alternative PLMAs characterization schemes may be developed (see for instance Gemoda [16], a generic interesting tool but with many parameters) and their advantages should be compared. The procedure for merging non-representative positions that we have presented is the simplest and should be elaborated. We have also ideas to raise prediction accuracy by developing distances taking into account the weights of the amino acids at each position with respect to the training sequences. An alternative way to handle unpredictable family variation would be to use the learned automata as the underlying structure of probabilistic automata, or hidden Markov models, and estimate their stochastic parameters by the classical well-studied training methods; but the advantage of the distance approach is that these variations are treated outside the model by measuring the distance to it, allowing the models to focus only on an explicit characterization of the important properties of the training sequences.
An interesting feature of learning syntactical model is that this kind of models allow to parse the sequences. A less expected application of our approach is that specific automata that we are able to learn can be used to align proteins as recently introduced in [21,5,27]. Differences and synergies between all these approaches converging from pattern discovery, multiple alignment and grammatical inference to learn explicit models on proteins are still to be studied but they constitute an emerging exciting area of research.

Acknowledgments. The authors would like to thank Israël-César Lerman, Basavanneppa Tallur and Anne Siegel for helpful discussions and ideas about this work. The authors are grateful to Christian Delamarche and Thierry Guillaudeux for their help for the constitution of the MIP and TNF data sets.

References

[1] S. Abdeddaïm and B. Morgenstern. Speeding up the dialign multiple alignment program by using the 'greedy alignment of biological sequences library' (gabios-lib). In *JOBIM*, pages 1–11, 2000.
[2] S.F. Altschul, W. Gish, E.W. Miller, and D.J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.
[3] A. Ashkenazi. Ttargeting death and decoy receptors of the tumour-necrosis factor superfamily. *Nature Reviews Cancer*, 2002.
[4] T.L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Second International Conference on Intelligent Systems for Molecular Biology*, 1994.
[5] M. Blanchette, W.J. Kent, C. Riemer, L. Elnitski, A. Smit, K.M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E.D. Green, D. Haussler, and W. Miller. Aligning Multiple Genomic Sequences With the Threaded Blockset Aligner. *Genome Res.*, 14(4):708–715, 2004.
[6] A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5(2):277–304, 1998.
[7] A. Califano. Splash: structural pattern localization analysis by sequential histograms. *Bioinformatics*, 16(4):341–357, 2000.
[8] F. Coste and D. Fredouille. What is the search space for the inference of nondeterministic, unambiguous and deterministic automata ? Technical report, IRISA - INRIA, RR-4907, 2003.
[9] F. Coste and G. Kerbellec. A similar fragments merging approach to learn automata on proteins. In *ECML*, pages 522–529, 2005.

[10] F. Coste, G. Kerbellec, B. Idmont, D. Fredouille, and C. Delamarche. Apprentissage d'automates par fusions de paires de fragments signifi cativement similaires et premières expérimentations sur les protéines mip. In *JOBIM*, Montréal, june 2004.
[11] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, July 1999.
[12] K. El Karkouri, H. Gueune, and C. Delamarche. Mipdb: a relational database dedicated to mip family proteins. *Biol Cell*, 97(7):535–543, July 2005.
[13] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89:10915–10919, 1992.
[14] D. Hernández, R. Gras, and R.D. Appel. Model: an effi cient strategy for ungapped local multiple alignment. *Computational Biology and Chemistry*, 28(2):119–128, 2004.
[15] N. Hulo, C.J. Sigrist, V. Le Saux, P.S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. De Castro, P. Bucher, and A. Bairoch. Recent improvements to the PROSITE database. *Nucl. Acids Res.*, 32(90001):D134–137, 2004.
[16] K.L.L. Jensen, M.P.P. Styczynski, I. Rigoutsos, and G.N.N. Stephanopoulos. A generic motif discovery algorithm for sequential data. *Bioinformatics*, October 2005.
[17] J.F. Jonassen, I.Collins and D. Higgins. Finding flexible patterns in unaligned protein sequences. *Protein Science*, 4(8):1587–1595, 1995.
[18] K. Karplus. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14:846–865, 1998.
[19] K.J. Lang, B.A. Pearlmutter, and R.A. Price. Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. *Lecture Notes in Computer Science*, 1433:1–12, 1998.
[20] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: A gibbs sampling strategy for multiple alignment.
[21] C. Lee, C. Grasso, and M. Sharlow. Multiple sequence alignment using partial order graphs . *Bioinformatics*, 18(3):452–464, 2002.
[22] I.C. Lerman and J. Azé. Indice probabiliste discriminant de vraisemblance du lien pour des données volumineuses. *RNTI-E-1, numéro spécial Mesures de Qualité pour la Fouille des Données, H. Briand, M. Sebag, R. Gras, F. Guillet, CEPADUES*, pages 69–94, 2004.
[23] B. Morgenstern. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15:211–218, 1999.
[24] C. Nevill-Manning, , and I. Witten. Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artifi cial Intelligence Research*, 7:67–82, 1997.
[25] C.G. Nevill-Manning, T.D. Wu, and L. Brutlag, Douglas. Highly specifi c protein sequence motifs for genome analysis. *PNAS*, 95(11):5865–5871, 1998.
[26] J. Oncina and P. Garcia. Inferring regular languages in polynomial update time. *Pattern Recognition and Image Analysis*, pages 49 – 61, 1992.
[27] P. A. Pevzner, H. Tang, and G. Tesler. De novo repeat classifi cation and fragment assembly. *Genome Res*, 14(9):1786–1796, September 2004.
[28] I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics*, 14(1):55–67, January 1998.
[29] I. Rigoutsos, A. Floratos, L. Parida, Y. Gao, and D. Platt. The emergence of pattern discovery techniques in computational biology. *Metabolic Engineering*, 2:159–177, 2000.
[30] Eddy S. Hmmer user's guide: biological sequence analysis using prole hidden markov models. http://hmmer.wustl.edu/, 1998.
[31] Y. Sakakibara. Grammatical inference in bioinformatics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1051–1062, 2005.
[32] David B. Searls. The language of genes. *Nature*, 420:211–217, November 2002.
[33] Z. Solan, D. Horn, E. Ruppin, and S. Edelman. Unsupervised learning of natural languages. *Proc Natl Acad Sci U S A*, 102(33):11629–11634, August 2005.
[34] W.R. Taylor. The classifi cation of amino acid conservation. *Journal of theoretical Biology*, 119:205–218, 1986.
[35] T. Yokomori. Learning non-deterministic fi nite automata from queries and counterexamples. *Machine Intelligence*, 13:169–189, 1994.
[36] E.M. Zdobnov and R. Apweiler. Interproscan - an integration platform for the signature-recognition methods in interpro. *Bioinformatics*, 17(9):847–848, 2001.