

# Apprentissage d'automates par fusions de paires de fragments significativement similaires et premières expérimentations sur les protéines MIP

François COSTE, Goulven KERBELLEC, Boris IDMONT, Daniel FREDOUILLE

IRISA, Symbiose, Campus de Beaulieu, 35042 Rennes Cedex, France

Christian DELAMARCHE

Université de Rennes 1, UMR CNRS 6026, CRM, Campus de Beaulieu, 35042 Rennes Cedex, France.

Courriel : francois.coste@irisa.fr, christian.delamarche@univ-rennes1.fr

## Résumé

*Nous proposons une nouvelle approche permettant d'apprendre des automates non déterministes pour la caractérisation et la modélisation de séquences protéiques. Cette approche est basée sur la fusion de fragments significativement similaires pour la caractérisation de la famille. Trois heuristiques d'ordonnement des paires de fragments sont introduites, dont une utilisant la présence de contre-exemples, et un processus de généralisation, basé sur l'identification des propriétés physico-chimiques des acides aminés, est proposé. Les premières expérimentations menées sur la caractérisation de protéines de la famille MIP montrent la pertinence des automates appris, attestée par un bon pouvoir de prédiction.*

**Mots-clés :** Inférence d'automates, découverte de motifs, séquences protéiques, propriétés physico-chimiques des acides aminés.

## Introduction

Prédire la (ou les) fonction(s) d'une protéine dont on connaît la séquence en acides aminés est un des grands défis de la bioinformatique. La fonction d'une protéine dépend d'une manière étroite de sa structure tridimensionnelle (3D) qui dépend elle-même de sa séquence en acides aminés. Cependant, la prédiction de structure 3D à partir de séquences est également un problème difficile et ne peut être utilisée comme étape intermédiaire pour la prédiction de la fonction qui doit alors être directement déduite de la structure primaire (et/ou secondaire).

Une première méthode procède par homologie et consiste à chercher dans les bases de données une séquence proche de fonction connue, transférée par analogie à la protéine d'intérêt. Cette inférence n'est possible qu'entre séquences suffisamment proches. Une autre méthode plus générale consiste à rechercher si la séquence contient des signatures connues, préalablement identifiées comme étant caractéristiques de certaines fonctions. Les signatures actuellement utilisées pour la caractérisation sont essentiellement des motifs, au pouvoir d'expression inférieur aux expressions régulières, éventuellement pondérés ou probabilisés (voir par exemple la banque de motifs Prosite, <http://us.expasy.org/prosite/>, ou la banque fédérative InterPro, <http://www.ebi.ac.uk/interpro/>).

Les programmes d'apprentissage de motifs simples sur l'ADN sont relativement nombreux (notamment pour la recherche de promoteurs) mais peu d'algorithmes d'apprentissage de motifs sur les protéines existent. Les méthodes les plus utilisées sont des méthodes d'échantillonnage statistique pour la recherche de sous séquences conservées (dont le plus connu est peut-être Gibbs sampler [1]) ou des méthodes d'entraînement de « HMM profils » (avec des outils comme SAM ou HMMER [2]). La structure sous jacente de ces motifs est pauvre (essentiellement séquentielle et généralement très localisée) et ne permet donc pas de prendre en compte les interactions à longue distance qui interviennent naturellement dès que l'on considère le déploiement des séquences dans l'espace. A notre connaissance Pratt [3] est le programme permettant d'apprendre les motifs les plus expressifs. Mais les motifs appris restent du niveau des motifs Prosite et ne permettent pas non plus d'exprimer des corrélations entre sous motifs. Des formalismes ad-hoc, superposés aux motifs peuvent aussi être utilisés pour prendre en compte la corrélation [4].

Si l'on se place dans le cadre de la Théorie des Langages, les langages réguliers permettent la prise en compte de dépendances à longue portée, bien qu'ils soient les moins expressifs de la hiérarchie de Chomsky. L'apprentissage de ces langages réguliers (notamment sous la forme de l'apprentissage d'automates) à partir de séquences a été beaucoup étudié en Inférence Grammaticale. Sous sa forme classique, le problème consiste à identifier un automate représentant un langage, dit cible, à partir d'exemples de séquences appartenant au langage (exemples positifs) et éventuellement de séquences n'appartenant pas au langage (exemples négatifs ou contre-exemples). L'apprentissage d'automates peut éventuellement être considéré comme l'apprentissage de la structure de modèles de Markov cachés (HMM) dont les probabilités resteraient à être estimées à l'aide des algorithmes classiques du domaine. Comme les HMM, les automates permettent d'analyser les séquences en leur totalité. Ils peuvent être utilisés pour prédire l'appartenance d'une séquence à une famille mais aussi, par exemple, pour localiser les « sites actifs » dont ils peuvent modéliser l'enchaînement. Cependant par rapport aux HMM (ou d'autres machines du type « boîtes noires » comme les réseaux de neurones), la nature non numérique des automates leur permet d'être plus facilement exploitable par un expert à qui ils présentent une modélisation explicite et exacte de la famille de séquences. Cela se paie en revanche par une plus grande difficulté lors de l'apprentissage, l'approximation n'étant guère possible.

Des résultats d'apprenabilité au niveau théorique et des schémas algorithmiques performants ont cependant été obtenus pour l'apprentissage d'automates. En particulier, l'approche dite par « fusions d'états » a donné de très bons résultats sur des jeux de données artificielles. Les premières expérimentations menées sur des séquences protéiques ont néanmoins montré les limites de ces approches génériques et motivent l'introduction de nouveaux biais d'inférence ainsi que l'intégration d'informations supplémentaires pour adapter ces algorithmes à la nature des séquences traitées.

Nous intéressés à la caractérisation de familles de protéines, nous proposons ici une nouvelle approche heuristique pour les algorithmes par fusions d'états, basée sur les paires de fragments similairement significatifs. Sans nécessiter de réaliser d'alignement multiple préalable, cette heuristique couplée à la représentation par automate permet d'obtenir à la fois une localisation des zones conservées et une modélisation de l'enchaînement de ces zones. La localisation des zones conservées permet de proposer une généralisation supplémentaire basée sur les groupes de Taylor [5] faisant à la fois ressortir les propriétés physico-chimiques des acides aminés concernés et permettant d'augmenter la capacité de généralisation des automates obtenus. La méthode présentée est souple et permet notamment de considérer plusieurs types d'ordonnement des paires de fragment à fusionner, notamment en fonction de la disponibilité, ou non, d'exemples de séquences n'appartenant pas à la famille à caractériser (contre-exemples).

L'approche permet d'obtenir avec le même algorithme toute une gamme de généralisations : de celles très spécifiques aux séquences d'apprentissage (de plus grande taille mais permettant par exemple de discerner les éventuelles sous familles) à des caractérisations beaucoup plus générales ne gardant que l'essentiel nécessaire à la caractérisation de la famille entière (plus compactes et plus dans l'esprit des motifs Prosite). L'introduction d'une mesure de type *Minimum Description Length* (MDL) permettrait de disposer d'un critère de sélection des modèles les plus pertinents. Dans cette première étude, pour illustrer et mesurer le potentiel de cette approche que nous souhaitons développer, nous avons évalué les modèles plus directement suivant leur qualité prédictive sur des séquences n'ayant pas servi pour l'apprentissage (l'apprentissage effectué assurant toujours une bonne prédiction de toutes les séquences de l'échantillon d'apprentissage). Une bonne prédiction de la fonction sur des séquences inconnues permet de vérifier que l'automate ne contient bien que l'information pertinente pour la caractérisation de la famille. Cette étude a été réalisée sur une de nos familles d'intérêt : la famille des protéines MIP (Major Intrinsic Protein). Cette famille désigne des canaux impliqués dans le transport de l'eau et/ou de petits solutés non chargés (principalement le glycérol) à travers les membranes biologiques. La motivation de cette étude est la recherche de motifs discriminants entre les deux sous-groupes majeurs, les aquaporines et les facilitateurs du glycérol, afin de comprendre le rôle de la formation des homotétramères dans la fonction des MIP. Dans ce cadre, la discrimination seule des deux sous-familles est insuffisante. Il faut aussi obtenir la caractérisation/modélisation la plus précise possible de chacune d'elles, si possible de façon différentielle, pour permettre une meilleure compréhension du mode fonctionnel de ces protéines.

Nous commençons la présentation par un bref rappel sur l'apprentissage d'automates par fusions d'états en section 1 avant d'introduire notre approche par fusions de fragments significativement similaires en section 2. Différentes heuristiques par utilisation des fragments similaires sont présentées en section 3. La fusion des fragments constitue le cœur de la caractérisation de la famille. Cette étape doit cependant être suivie d'une phase de généralisation de l'automate obtenu décrite en section 4. Après avoir présenté la famille des protéines MIP en section 5, nous illustrons en section 6 le comportement de la méthode sur ces protéines avant de conclure sur les perspectives de développement de l'approche.

# 1 Apprentissage d'automates par fusions d'états

Les automates sont des machines à états finis permettant de représenter un ensemble potentiellement infini de séquences. Des exemples d'automates sont donnés en figure 1. Pour un *alphabet* de symboles  $\Sigma$  donné (par exemple, l'ensemble des 20 acides aminés), un automate est défini par un *ensemble d'états*  $Q$  (dans lequel sera distingué le sous ensemble des *états initiaux*, noté  $Q_i$ , et le sous ensemble des *états finaux*, noté  $Q_f$ ) et une *fonction de transition*  $\delta$ , définie sur  $Q \times \Sigma \rightarrow \wp(Q)$ , tel que  $\delta(q,a)$  est l'ensemble des états  $q'$  atteignable depuis l'état  $q$  par le symbole  $a$ .

Si pour toute paire  $(q,a)$ , il n'existe qu'au plus un seul état atteignable  $q'$  alors l'automate est dit déterministe. Une séquence sera *acceptée* par un automate si l'on peut atteindre un état final à partir d'un état initial en lisant les symboles de la séquence. Sinon la séquence est dite *rejetée* par l'automate. L'ensemble des séquences acceptées par un automate  $A$  définit le *langage reconnu* par cet automate et est noté  $L(A)$ .

Les automates ont le même pouvoir expressif que les expressions régulières et permettent de représenter n'importe quel langage régulier. Par rapport aux motifs couramment utilisés en bioinformatique, l'intégralité de la séquence est lue par un automate pour être acceptée. L'acceptation d'une séquence spécifiant un (ou plusieurs) chemin(s), les automates peuvent aussi être utilisés pour étiqueter les symboles des séquences selon les états visités, leur assignant ainsi la sémantique associée à ceux-ci (par exemple, l'appartenance à un site actif).

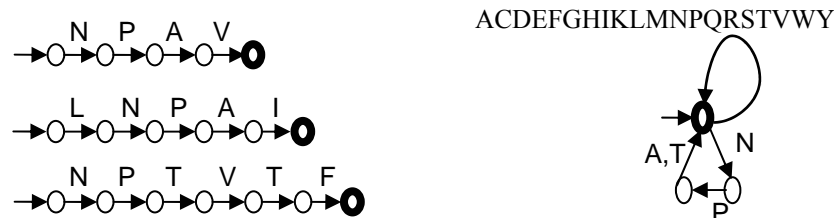


Figure 1. MCA(S) pour  $S = \{NPAVSF, HLNPAI, NPTVTF\}$  (gauche) et exemple d'automate obtenu par fusions d'états à partir du MCA(S) (droite). Les états initiaux sont représentés avec une flèche entrante. Les états finaux sont entourés en gras.

Le problème classique de l'apprentissage d'automates à partir de séquences consiste à retrouver un automate à partir d'un échantillon d'apprentissage (que nous noterons  $S$ ) composé d'exemples de séquences appartenant au langage de l'automate à inférer (ensemble des exemples positifs, noté  $S^+$ ) et éventuellement de séquences n'appartenant pas au langage (ensemble des exemples négatifs ou contre-exemples, noté  $S^-$ ).

Une des approches les plus utilisées pour l'apprentissage d'automates est celle dite par fusions d'états. Elle consiste à construire un automate canonique, le MCA (*Maximal Canonical Automata*) représentant exactement les exemples de l'échantillon d'apprentissage (voir figure 1), et à généraliser cet automate par une succession de fusions d'états [6].

Le pseudo code du schéma général de l'apprentissage d'automates par fusions d'état est donné par la procédure **sma(S)** prenant en entrée un échantillon d'apprentissage  $S$  et retournant l'automate  $A$  comme résultat de l'apprentissage (algorithme 1.). L'opération principale sur l'automate courant est la fusion d'état réalisée par la fonction  $\text{fusion}(A,e1,e2)$  qui permet de généraliser l'automate  $A$  en unifiant les états  $e1$  et  $e2$ . Cette opération n'est effectuée que si la fusion est « acceptable » suivant le test effectué sur l'automate courant par la fonction  $\text{fusion\_acceptable}(A,e1,e2)$  qui permet de limiter la généralisation et dépend du type d'apprentissage effectué (par exemple, si l'on dispose d'un ensemble de contre-exemple le test peut consister à vérifier qu'aucune séquence de  $S^-$  ne sera acceptée après la fusion des deux états). Les états considérés pour la fusion et l'ordre dans lequel ils sont considérés étant particulièrement important pour guider l'algorithme vers le meilleur automate, plusieurs types d'heuristiques peuvent être implémentés à l'aide de la fonction  $\text{choix\_paires\_états}(A)$ . L'algorithme termine lorsque plus aucune paire d'états n'est proposée à la fusion.

1. **Procédure sma(S)**
2.  $A \leftarrow \text{MCA}(S)$
3. **tant que**  $\langle e1,e2 \rangle \leftarrow \text{choix\_paires\_états}(A)$  **faire**
4.     **si**  $\text{fusion\_acceptable}(A,e1,e2)$
5.         **alors**  $A \leftarrow \text{fusion}(A,e1,e2)$
6.     **fsi**
7. **fait**
8. retourne  $A$

Une heuristique particulièrement performante pour l'apprentissage d'automates déterministes à partir d'exemples et de contre exemples s'inscrivant dans ce schéma est EDSM [7]. L'idée de cette heuristique consiste essentiellement à ordonner les fusions à effectuer suivant leur évidence, attestée par la mise en correspondance d'états finaux par la fusion et mesurant ainsi la similarité entre les parties de l'automate fusionnées.

Dans une première étude, nous avons tenté d'intégrer les mesures de similarité « biologique » dans EDSM pour l'apprentissage de protéines. Cependant comme l'ont montrées nos premières expérimentations, l'heuristique EDSM n'est pas adaptée à l'apprentissage sur les protéines. Un des principaux problèmes rencontré par EDSM sur les séquences protéiques est leur longueur et la faible pertinence de la notion d'état final lorsque l'on prend les séquences entières dans l'échantillon d'apprentissage. La représentation par automate déterministe n'est également pas forcément la plus adaptée et n'a certainement pas aidé à obtenir de bons résultats.

Suite à ces observations, nous avons développé une nouvelle approche heuristique, dans l'esprit de EDSM mais appliquée à l'apprentissage d'automates non déterministes et utilisant la nature des protéines, que nous présentons dans la section suivante.

## 2 Fusion de fragments significativement similaires

Comme dans EDSM, nous proposons ici de nous appuyer sur les similarités entre séquences pour ordonner les états à fusionner. Le problème de la similarité entre protéines a été beaucoup étudié. Une mesure simple de similarité entre deux séquences de protéines consiste à calculer la somme sur l'ensemble des positions en vis-à-vis des coûts donnés par une matrice de substitution. Pour une longueur donnée, il est ainsi possible de considérer l'ensemble des sous-séquences (nommées *fragments*) et d'ordonner les paires de fragments par similarité décroissante. Ceci impose cependant de choisir la longueur de fragment pertinente a priori, la comparaison n'étant pas possible entre paires de fragments de longueurs différentes.

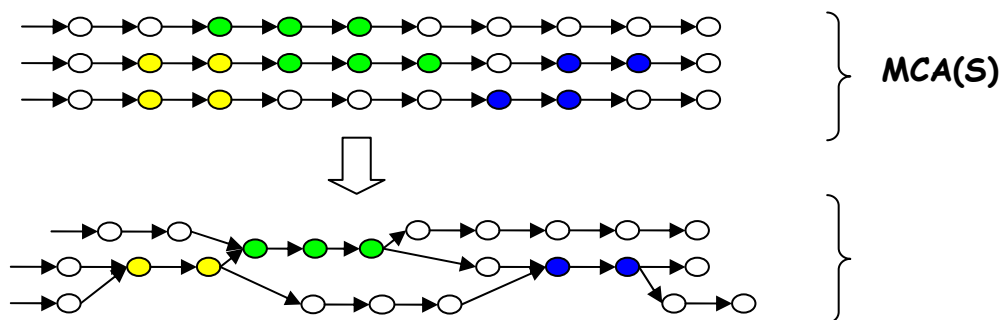


Figure 2. Fusions des fragments

La procédure  $\mathbf{fma}(S, \omega)$  présente le schéma algorithmique de cette approche par fusions de fragments significatifs. L'ensemble  $F$  contient l'ensemble des paires de fragments significativement similaires, ordonnés ici suivant leur probabilité d'occurrence mais d'autres tris sont possibles (voir section 3). La fonction  $\mathit{fusion\_fragment}(A, f1, f2)$  réalise la série de fusions d'états correspondants aux fragments. La fonction  $\mathit{fusion\_fragment\_acceptable}(A, f1, f2)$  vérifie que cette série de fusions est acceptable au niveau des paires d'états, mais aussi au niveau des fragments en vérifiant que les fusions effectuées préservent bien les fragments déjà fusionnés (i.e. ne fusionnent pas une partie d'un fragment déjà fusionné sur lui même), sinon la fusion de la paire de fragments courante est refusée. Ce test illustre l'importance de l'ordonnancement des paires de fragments, les premières paires de fragments seront préservées et les paires de fragments les plus loin dans la liste ont moins de chances d'être fusionnées que leur précédentes.

1. **Procédure**  $\mathbf{fma}(S, \omega)$
2.  $A \leftarrow \mathit{MCA}(S)$
3.  $F \leftarrow \mathit{fragments\_significatifs}(S, \omega)$
4.  $F \leftarrow \mathit{trier}(F)$
5. pour tout  $p = \langle f1, f2 \rangle$  dans  $F$  faire
6.     si  $\mathit{fusion\_fragment\_acceptable}(A, f1, f2)$
7.         alors  $A \leftarrow \mathit{fusion\_fragment}(A, f1, f2)$
8.     fsi
9. fpour
10. retourne  $A$

### 3 Réordonner les fragments

Dans la section précédente, nous avons introduit le schéma algorithmique d'apprentissage par fusion des paires de fragments significativement similaires. L'ensemble des paires de fragments significativement similaires peut n'être considéré que comme une première sélection de fusions potentiellement intéressantes, à réordonner en fonction de leur degré d'évidence par rapport à la famille de protéines à caractériser (procédure **fma**, ligne 4). Nous considérons ici deux cas selon que des exemples négatifs sont disponibles ou pas.

Si l'on n'a que des exemples positifs, l'évidence qu'une paire de fragments est pertinente étant donné que l'ensemble des paires de fragments peut être évaluée en mesurant son « support », c'est-à-dire le nombre de séquences qui comportent un fragment similaire aux fragments de la paire. Plusieurs critères peuvent être choisis pour décider qu'un fragment est similaire à deux autres. Nous avons choisi d'utiliser l'inégalité triangulaire, qui est un critère peu restrictif et qui n'ajoute pas de nouveau paramètre à la méthode. Ainsi, en notant  $w(f_i, f_j)$  le score donné par Dialign2 reflétant la probabilité d'avoir une similitude aussi élevée entre deux fragments, une paire de fragment  $(f_1, f_2)$  sera dite supportée par un fragment  $f$  si :

$$w(f, f_1) + w(f, f_2) \geq w(f_1, f_2) .$$

Une paire de fragment sera dite supportée par une séquence si celle-ci contient un fragment supportant la paire de fragment. Le nombre de séquences supportant chaque paire de fragment permet de réordonner les paires de fragments par rapport à leur représentativité dans la famille de séquence.

Lorsque l'on dispose de contre exemples, des mesures plus fines peuvent être élaborées pour mesurer l'évidence de la fusion de paires de fragment à partir de la connaissance de la nature (exemples ou contre-exemples) des séquences supportant ceux-ci. Nous présentons ici un indice d'implication  $\iota$  qui nous a été proposée par Lerman selon les idées présentées dans [9]. Celle-ci permet d'utiliser les contre-exemples pour favoriser la caractérisation de la famille de protéine en commençant par ce qui la distingue des contre-exemples disponibles plutôt que par ce qui est commun au sein de la famille. Ce que cherche à évaluer cet indice, c'est comment le support d'une paire de fragments dans des exemples implique sa proportion d'être supporté dans des exemples et des contre-exemples. Soit, en notant  $P$  la proportion d'un ensemble et en désignant par **support** et **support<sub>CE</sub>** les séquences supportant la paire de fragments respectivement dans l'ensemble des contre-exemples et dans l'ensemble de l'échantillon d'apprentissage, l'indice est :

$$\iota = (-P(\text{support}_{CE}) + P(\text{support}) \times P(\text{contre-exemples})) / \text{sqrt}(P(\text{support}) \times P(\text{contre-exemples}))$$

Cet indice permet d'utiliser les contre-exemples disponibles à condition qu'ils soient bien choisis, ce qui n'est pas toujours facile à faire. Un autre cadre d'application pour lequel ce type d'indice apparaît particulièrement adapté est la caractérisation d'une sous-famille par rapport à une autre, les séquences de chaque sous-famille servant de contre-exemples pour l'autre sous-famille. Cette utilisation est illustrée dans la partie expérimentation pour caractériser les MIP laissant passer l'eau par rapport à celles laissant passer le glycérol.

### 4 Post-généralisation

La fusion des fragments significativement similaires permet d'obtenir une première caractérisation de la famille de protéines. Cependant, la généralisation effectuée reste faible et il est peu probable de pouvoir accepter de nouvelles séquences avec l'automate obtenu. Pour augmenter le pouvoir prédictif, les fragments non impliqués dans une paire de fragments significativement similaires ne doivent plus être utilisés pour caractériser la famille. Une façon d'éliminer leur influence est de les fusionner sur eux-mêmes, ce qui permet d'obtenir un état avec une boucle sur lui-même pouvant être interprété comme un gap. Cette opération peut être effectuée sur chaque fragment jugé non utile à la caractérisation de la famille.

Pour déterminer les fragments inutiles dans l'automate, nous avons introduit un paramètre  $p$  représentant le pourcentage de séquences devant passer dans un état pour qu'il soit considéré comme utile à la caractérisation. La procédure de fusion des fragments inutiles consiste alors à fusionner les états adjacents empruntés par un nombre de séquences inférieur à  $p$ . Cette phase de généralisation élimine les états inutiles à la caractérisation.

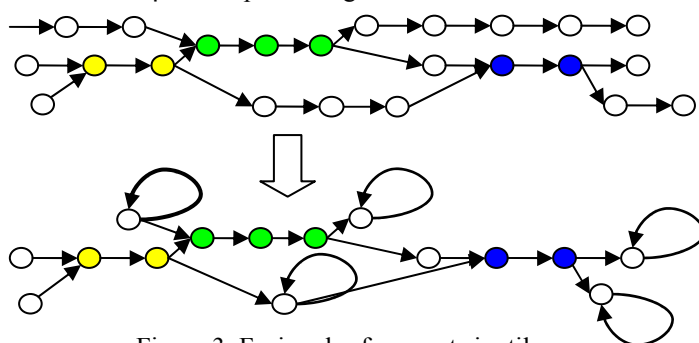


Figure 3. Fusion des fragments inutiles

Lors d'une deuxième phase, on s'intéresse à la généralisation des transitions de l'automate.

La présence de fragments similaires au sein d'une famille fonctionnelle est un indice de la sélection et de la préservation de propriétés physico-chimiques particulières caractéristiques de la famille. La deuxième phase de généralisation vise à identifier ces propriétés parmi les acides aminés positionnés en vis-à-vis par la fusion des fragments. Etant donné un ensemble d'acides aminés, la procédure utilise un ensemble de groupements d'acides aminés par propriétés pour généraliser chaque ensemble d'acides aminés positionnés en vis-à-vis au plus petit groupe le contenant. Plus précisément, en notant  $\Sigma$  l'ensemble de tous les acides aminés, la procédure adoptée pour étendre un ensemble d'acides aminés  $E$  suivant un ensemble de groupes  $T$  (typiquement l'ensemble des groupes de Taylor [5], mais il est aussi possible de spécifier ses propres groupes en fonction de ses centres d'intérêt ou de la nature des séquences traitées) est la suivante :

1. **Procédure groupe(E, T)**
2. si  $|E| \leq 2$  alors retourner E
3. sinon
4.      $\Gamma \leftarrow \{G \in T / E \subseteq G\}$
5.     si  $\Gamma$  non vide
6.     alors retourner minimal( $\Gamma$ )
7.     sinon retourner  $\Sigma$
8.     fsi
9. fsi

Dans cette procédure, la généralisation n'est pas effectuée si l'ensemble d'acides aminés ne comporte pas plus de 2 membres, considérant que en dessous de ce nombre les acides aminés eux-mêmes sont plus importants que les groupes auxquels ils appartiennent. A l'opposé, si les acides aminés ne sont inclus dans aucun groupe, aucune évidence physico-chimique n'est détectée pour cette position et la généralisation s'effectue à tous les acides aminés (l'acide aminé présent à cette position n'est pas considéré comme informatif).

Cette généralisation est appliquée aux acides aminés présents dans les transitions en vis-à-vis (transitions partant du même état  $q_1$  et arrivant au même état  $q_2$ ). Son rôle est complémentaire à la fusion de fragments dont elle utilise les propriétés de localisation pour affiner au niveau des acides aminés les informations de similarités obtenues sur les fragments. Son application permet d'abaisser considérablement le nombre d'exemples pour l'apprentissage (cette généralisation serait inutile si l'on disposait de suffisamment d'exemples pour que tous les cas de substitutions d'acides aminés au sein des groupes des propriétés pertinentes dans l'échantillon d'apprentissage, mais le nombre de séquence nécessaire serait alors prohibitif). Cette généralisation permet aussi de gagner en pouvoir explicatif en désignant à l'expert les positions importantes et les propriétés qui y sont associées.

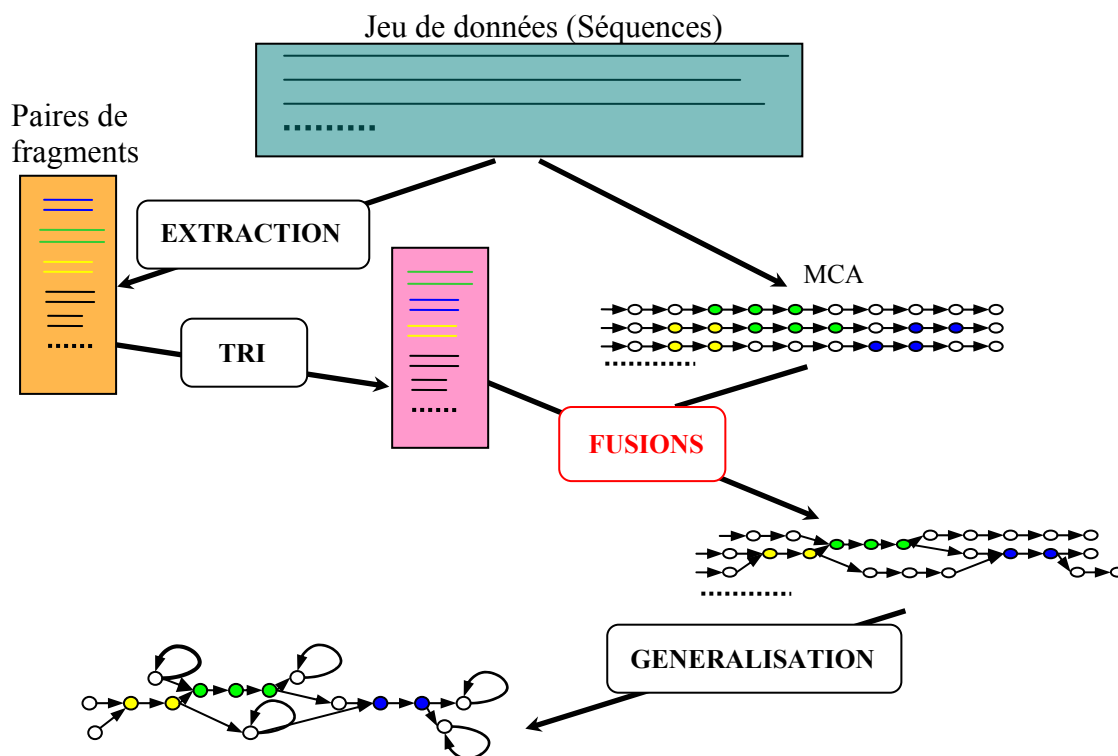


Figure 4. Schéma de l'algorithme et flux de données.

## 5 La famille des protéines MIP

Les protéines de la famille MIP (Major Intrinsic Proteins) sont impliquées dans les mécanismes de maintien de l'homéostasie cellulaire [10]. La famille MIP désigne des canaux qui sont impliqués dans le transport de l'eau et/ou de petits solutés non chargés (principalement le glycérol) à travers les membranes biologiques. Pour faciliter la description des MIP, il est habituel de distinguer deux types fonctionnels : Les aquaporines et les facilitateurs du glycérol. La découverte des aquaporines par Peter AGRE (prix Nobel de chimie 2003) a révolutionné les connaissances biophysiques sur les mécanismes de perméabilité des membranes biologiques. L'analyse de plusieurs pathologies humaines confirme que les aquaporines sont fondamentales dans l'organisme : dysfonctionnement rénal, œdème cérébral et cardiaque, vision, intoxication ...

**Les aquaporines** ont été identifiées dans les trois règnes du vivant. Elles permettent un transport passif rapide et sélectif de l'eau au travers de la membrane plasmique des cellules tout en excluant le passage des ions. Le dysfonctionnement d'aquaporines est associé à un nombre de plus en plus grand de pathologies humaines : Diabète insipide, cataracte, œdèmes... La protéine modèle des aquaporines est AQP1 chez les mammifères et AqpZ chez *E. coli*.

**Les facilitateurs du glycérol** permettent le transport facilité du glycérol et/ou de petits solutés non chargés au travers de la membrane plasmique tout en excluant l'eau et les ions. Le modèle historique de ce groupe fonctionnel est la protéine GlpF de *E. coli*, qui forme dans la membrane un canal de diffusion facilitée au glycérol. Après phosphorylation dans le cytoplasme, le glycérol ne peut plus franchir le canal en sens inverse.

A ce jour, on connaît 13 gènes de protéines MIP chez l'homme et 38 gènes chez la plante *Arabidopsis thaliana*. Les membres de la famille MIP possèdent environ 300 acides aminés et présentent une organisation structurale commune comportant 6  $\alpha$ -hélices entourant un pore central. Cette organisation correspond au modèle topologique proposé par P. Agre et ses collaborateurs en 1994 et connu sous le nom de "modèle en sablier". Le modèle en sablier a été validé au plan structural sur l'aquaporine AQP1 et sur le facilitateur du glycérol GlpF. Toutes les protéines MIP présentent de nombreux résidus hautement conservés localisés le long de la séquence. On remarque en particulier, une bonne conservation du motif asparagine-proline-alanine (NPA), présent dans les 2 petites hélices HB et HE qui plongent symétriquement à l'intérieur du pore et qui participent à la zone de constriction du canal.

La structure des 2 protéines modèles de la famille MIP, AQP1 et GlpF, a été déterminée par cristallographie aux rayons X avec une résolution de 2,2 Å [11,12]. Ce sont les 2 seules structures dont on dispose actuellement pour cette famille. Ces études montrent que AQP1 et GlpF cristallisent sous forme d'homotétramères dans lesquels chaque monomère forme un « canal actif ». Ces structures ont permis de proposer un mécanisme moléculaire de fonctionnement des canaux MIP, dans lequel la sélectivité à l'eau ou au glycérol s'expliquent par la distribution d'acides aminés spécifiques tout le long du pore et au niveau de la zone de restriction [13,14]. Des expériences de mutagenèse dirigée confirment le rôle clé de certains acides aminés localisés à l'intérieur du pore. Cependant, il faut noter que ce modèle n'explique pas totalement les mécanismes moléculaires de sélectivité des substrats *in vivo*, par exemple le fonctionnement des canaux mixtes (eau / glycérol), la régulation bidirectionnelle du flux d'eau ou de glycérol, l'imperméabilité aux ions...

Le but de cette étude est la recherche de motifs discriminants entre les deux sous-groupes majeurs, les aquaporines et les facilitateurs du glycérol, afin de comprendre le rôle de la formation des homotétramères dans la fonction des MIP. Nous présentons ici les résultats obtenus lors d'un premier travail de caractérisation de ces deux sous-familles. Le but est d'évaluer le potentiel de la méthode d'apprentissage avant de poursuivre son développement pour la discrimination de sous familles.

Figure 4. Modèle 3D inséré dans une membrane biologique et Organisation fonctionnelle des aquaporines en tétramères, ou chaque monomère fonctionne comme canal hydrique.

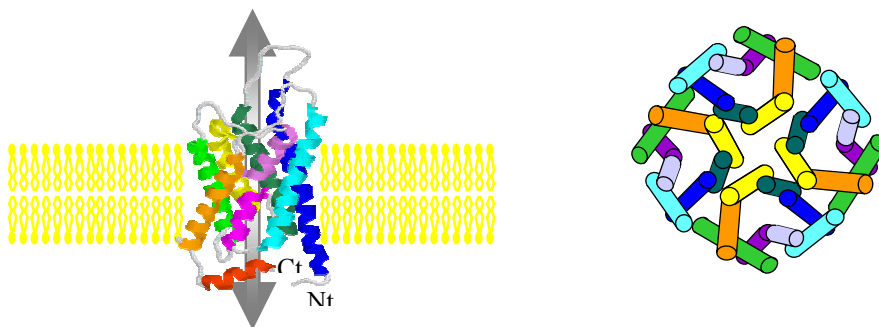


Figure 5. Modèle 3D inséré dans une membrane biologique et organisation fonctionnelle des aquaporines en tétramères, ou chaque monomère fonctionne comme canal hydrique.

## 6 Expérimentations

Les expérimentations suivantes ont été effectuées dans le but d'évaluer tout d'abord l'efficacité de notre approche par fusions de fragments, puis de comparer les 3 heuristiques proposées. Nous avons choisi d'analyser la famille fonctionnelle MIP au travers de ses 2 sous-familles, AQP et GlpF. Notre objectif a été de vérifier si nos méthodes ont effectivement bien appris ces sous-familles. C'est-à-dire que nous avons jugé si les automates appris à partir d'une sous-famille peuvent reconnaître de nouvelles séquences appartenant à celle-ci, tout en ne reconnaissant pas ou peu de séquences n'appartenant pas à cette sous-famille. Dans un premier temps, l'évaluation se fait sur une quantité de séquences reconnues directement par l'automate. Dans un second temps, une évaluation plus fine met en jeu une notion de distance des séquences par rapport aux automates. En définissant une erreur comme l'équivalent d'une substitution, insertion ou délétion d'un acide aminé dans l'automate, la distance équivaut au nombre d'erreur minimal que l'on autorise lors de la validation d'une séquence par un automate. En effet, on sera d'abord satisfait d'observer que des séquences négatives ne sont pas acceptées, mais on se demandera à quel distance sont-elles de l'acceptation, ou encore, combien d'erreurs seraient nécessaires à leur reconnaissance.

Un des premiers problèmes à résoudre pour pouvoir évaluer raisonnablement l'efficacité d'un apprentissage est la constitution d'un jeu de données qui soit fiable. Sur une première sélection de 250 protéines MIP, nous avons conservé 62 AQP et 22 GlpF comme représentatifs de leur famille. En vue d'un apprentissage et surtout d'une évaluation équilibrée, des séquences ont été retirées de ces groupes de manière à ce qu'il puisse y avoir au maximum 90% de similarité entre chaque paire de séquences, réduisant ainsi les jeux de données à 52 MIP, dont 36 AQP et 16 GlpF.

D'autre part, un jeu de séquences non MIP mais proche de celles-ci a été créé en effectuant un Blast à partir des séquences Aqp1 et Glpf et en conservant les séquences en haut de la liste dont l'annotation permettait de rejeter catégoriquement l'appartenance à la famille des MIP. Ce jeu comporte 41 séquences et sera appelé ci-après « blast non MIP ».

Nous avons décidé d'évaluer nos automates en fonction de leur qualité de prédiction. La taille des échantillons d'apprentissage étant réduite, nous avons mesuré cette qualité de prédiction par une méthode de *leave one out* : l'apprentissage s'effectue sur l'ensemble d'un échantillon moins une séquence, avec itération de cette opération pour chaque séquence de l'échantillon. Le taux de reconnaissance est alors la somme des séquences acceptées lors du *leave one out* divisé par la taille de l'échantillon. On dispose également des analyses des jeux de validation négatifs sur chaque automate généré par le *leave one out*. Pour chacun de ces jeux, une moyenne de qualité de prédiction est considérée sur les automates obtenus par le *leave one out*. Nos expérimentations permettent d'évaluer la qualité de l'apprentissage indépendamment sur les AQP et sur les GlpF.

De manière à pouvoir comparer les résultats, nous avons utilisé pour chaque heuristique le même ensemble de paires de fragments généré par le programme Dialign2. Les heuristiques ne changent que l'ordre dans lequel ces fragments sont considérés. L'option -afc de cet outil permet l'accès au calcul préliminaire à sa phase d'alignement et donc d'obtenir un ensemble de fragments significativement similaires 2 à 2. Le seuil minimal de Dialign2 a été fixé arbitrairement à 10. Ce qui correspond sur les MIP à des fragments suffisamment importants et significatifs. Nous disposons donc pour chaque sous-famille (AQP et GlpF) d'un ensemble de paires de fragments que nous trions selon nos 3 heuristiques (par similarité, par support et par l'indice d'implication), soit 6 ensembles. Un paramètre supplémentaire est ajouté : on ne considère que les  $n$  premières paires de fragments de chaque ensemble. La valeur de  $n$  correspond intuitivement au degré de généralisation désiré : un petit  $n$  renvoie des automates de petite taille ne contenant que les fragments les plus significatifs, alors qu'un très grand  $n$  renvoie des automates de plus grande taille, mais pour lesquels la caractérisation de la classe est plus précise. Sur chacun des graphiques, l'axe des abscisses indique la taille moyenne des automates. En effet, on se focalise ici sur la qualité d'apprentissage des automates, puis sur les heuristiques en faisant varier  $n$  (ici entre 500 et 300000 paires de fragments). Ainsi, une valeur de 40 états sur l'axe des abscisses correspond à la taille moyenne des automates obtenus par le *leave one out* correspondant à la fusion d'un nombre de fragment fixé par la valeur de  $n$ .

Pour ces expérimentations, afin de se focaliser sur l'influence des ordonnancements, le seuil fixant le pourcentage de séquences devant passer dans un état pour le préserver a été fixé à la valeur maximale de 100%. La généralisation des transitions a été effectuée en utilisant les groupes classiques de Taylor, tels qu'ils sont donnés dans [5]. Le *leave one out* génère, à partir de l'exécution de l'algorithme d'apprentissage, autant d'automates que de séquences contenues dans l'échantillon d'apprentissage. On observe enfin le pouvoir de prédiction pour des tailles proches.

On analysera, pour chacune des 3 heuristiques, 3 courbes correspondant à la validation des 3 échantillons.

Pour lire clairement ces graphiques il est essentiel d'avoir à l'idée :

- d'une part que les courbes de validation positive (vertes pour l'apprentissage des AQP, bleue pour les GlpF) permettent de connaître le niveau de reconnaissance des séquences positives.
- d'autre part qu'il existe des « automates universels » pour les *leave one out* donnant une taille moyenne inférieure à 40. Cela signifie que l'information contenue dans l'ensemble des fragments est en trop faible quantité

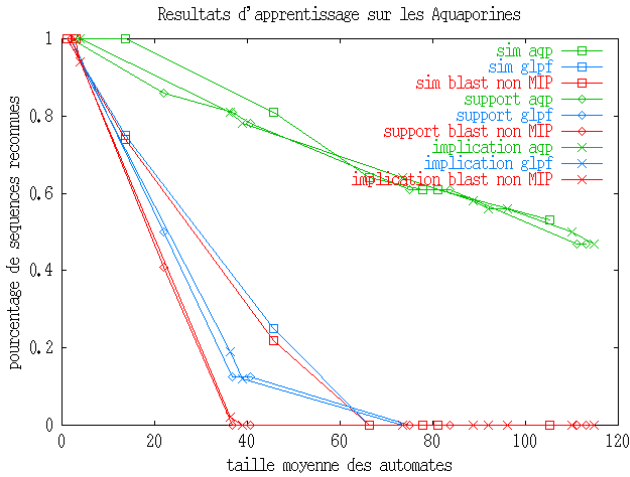


pour être pertinente. L'étape de Post-généralisation entraîne donc l'apparition d'automates universels qui reconnaissent alors aussi bien les séquences positives que négatives (toutes les valeurs sont à 100%).

- et enfin que la discrimination entre les échantillons, pour chaque heuristique, se juge à l'écart entre la courbe de validation positive et les deux courbes de validations négatives.

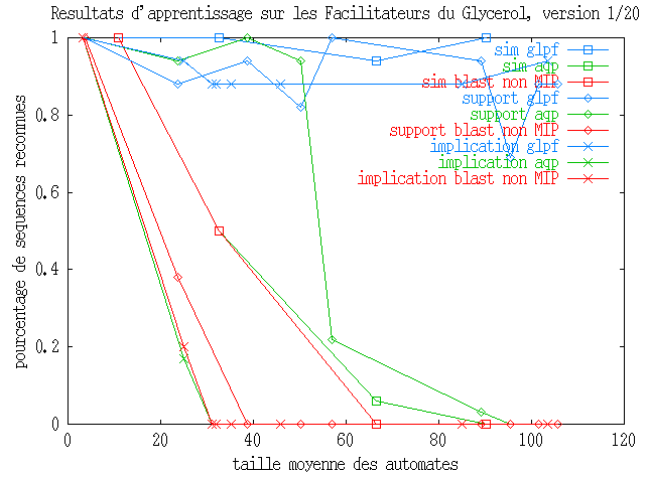
Pour les graphiques 1 et 2 l'axe des ordonnées indique quel est le pourcentage de séquences validées. Les couleurs correspondent aux trois échantillons de validation dont on dispose : échantillon positif (somme des séquences extraites lors du *leave one out* puis analysées), échantillon négatif "Blast non MIP" et échantillon négatif de la sous-famille GlpF si l'on apprend les AQP et vice versa.

### Apprentissage sur le jeu des AQP

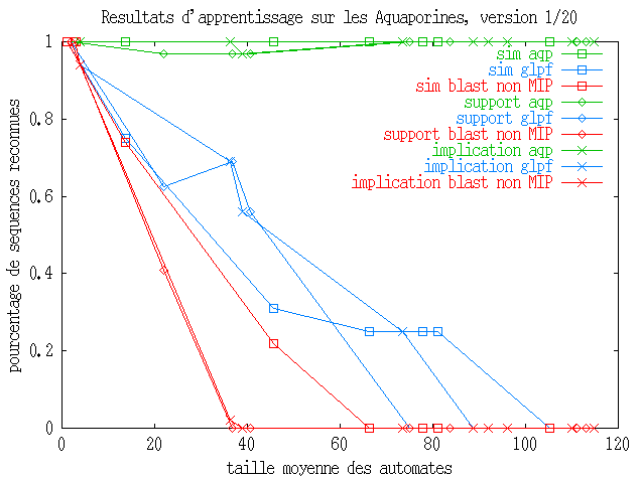


Graphique 1.

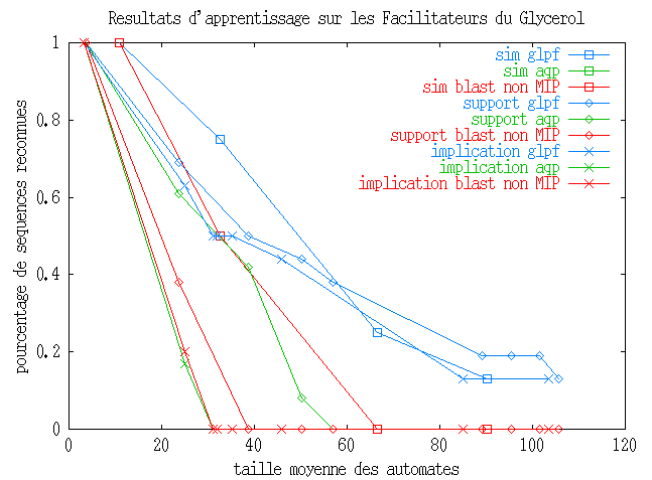
### Apprentissage sur le jeu des GlpF



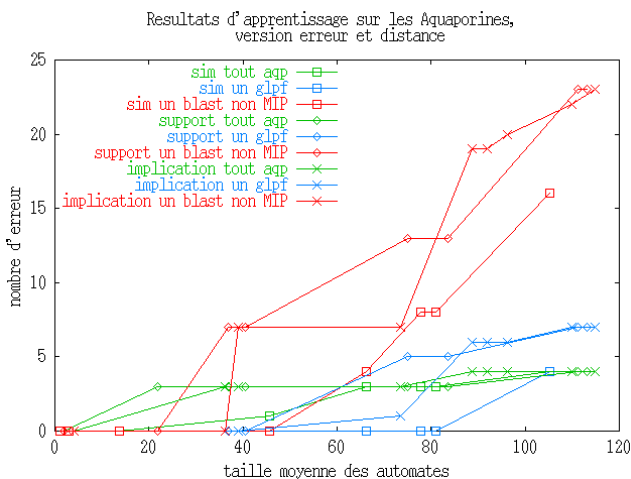
Graphique 2.



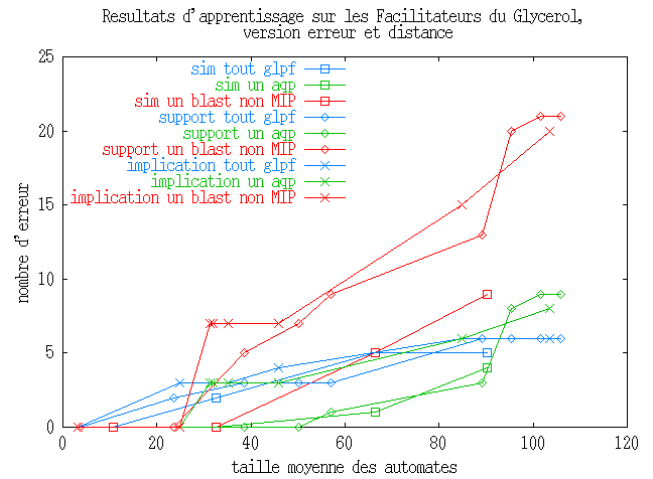
Graphique 3.



Graphique 4.



Graphique 5.



Graphique 6.

Sur le graphique 1, on constate que pour les automates d'une taille moyenne de 40, appris sur des AQP, on obtient 80% de reconnaissance de nouvelles AQP, moins de 40% de GIpF et aucune séquence négative de "Blast non MIP". Si l'on observe des automates plus spécifiques, par exemple autour de 80 états, aucune séquence négative n'est reconnue, mais seulement 60% des nouvelles AQP le sont.

Sur le graphique 2, les résultats semblent un peu moins bons au niveau de la reconnaissance, cependant nous sommes basés sur un jeu d'apprentissage relativement restreint, à savoir 15 GIpF pour chaque *leave one out*.

On peut établir une remarque qui sera générale : les 3 heuristiques ont des pouvoirs de prédiction assez proches et cependant, l'heuristique du support ainsi que celle de l'implication permettent une discrimination plus nette par rapport aux séquences négatives (graphique 1, pour une taille de 40, similarité valide environ 40% des GIpF alors que implication et discrimination en valide moins de 20%).

Les graphiques 3 et 4 ont la particularité par rapport aux graphiques 1 et 2 d'exprimer les résultats obtenus lorsque l'on autorise 1 erreur pour 20 transitions de l'automate, ce qui rejoint la notion de distance d'édition. On obtient alors un apprentissage de positifs quasi parfait (proche ou égal à 100 %), tout en acceptant très rarement les séquences négatives. Etant toutes à 100% ou presque, les heuristiques sont difficilement comparables sur la validation des positifs. Cependant, sur le graphique 3, pour l'heuristique de similarité, le rejet de l'échantillon négatif des "Blast non MIP" est rapide (rejet total pour une taille d'automate de 65). De plus, une amélioration est possible par les deux autres heuristiques (rejet total par support et implication pour des tailles d'automates de 40). Cette amélioration est constatée sur l'échantillon négatif des GIpF, mais uniquement sur des automates de taille supérieure à 80. L'augmentation de la spécificité joue donc ici un rôle important.

Les graphiques 5 et 6 ont pour objectif d'observer les distances des échantillons par rapport aux automates appris. Nous avons choisi de représenter combien d'erreurs sont nécessaires pour qu'il y ait une acceptation totale des échantillons positifs (courbes vertes des AQP sur le graphique 5), alors que pour les échantillons négatifs on a cherché à partir de combien d'erreurs on commence à accepter au moins une séquence négative. Une configuration idéal est observée par exemple sur le graphique 5 pour l'heuristique d'implication (les croix) avec des automates de tailles environ 85. En effet, les croix vertes (4 erreurs) sont en dessous des croix bleues (6 erreurs), elles-mêmes très nettement en dessous des croix rouges (19 erreurs). Ce qui signifie que l'on peut en augmentant le seuil d'erreur (ici jusqu'à 4), reconnaître tout l'échantillon positif sans reconnaître une seule séquence des échantillons négatifs. Des automates de petite taille (40 à 60), qui ne sont donc pas très spécifiques, reconnaissent souvent au moins une séquence des échantillons négatifs sans reconnaître l'échantillon positif dans sa totalité. Pour diminuer cette intersection, il semble que l'augmentation de la spécificité ainsi que le choix des heuristiques support ou implication plutôt que similarité aient une importance.

Ci-dessous, les histogrammes 1 à 6, permettent d'évaluer encore plus finement la qualité de l'automate appris par rapport à l'ensemble des jeux de validation dont on dispose. Cette représentation se base sur des automates de taille moyenne 80. Nous avons choisi une taille importante pour montrer que la qualité de la caractérisation reste bonne même lorsque l'on cherche des modèles précis de la famille de séquence. L'axe des abscisses équivaut à la distance des séquences par rapport aux automates. Les histogrammes montent en ordonnées jusqu'au nombre de séquences reconnues pour chaque classe donnée.

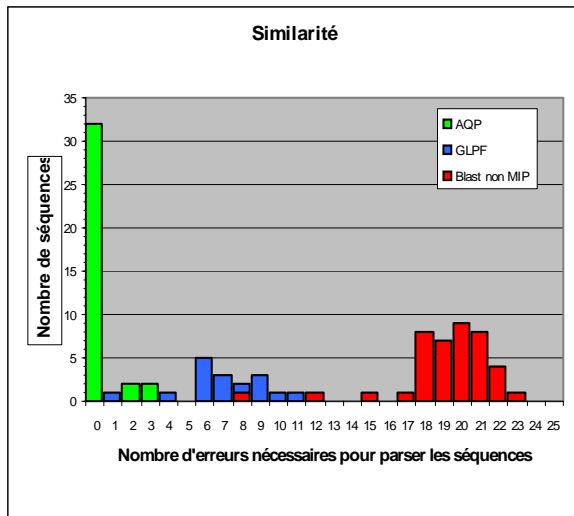
Ainsi, les trois classes se distinguent nettement : la classe apprise nécessite peu d'erreurs pour être entièrement reconnue, la classe des « blast non MIP » se trouve en général très distante des automates appris avec des erreurs nécessairement importantes et ne possédant jamais d'intersection avec la classe apprise, et au milieu s'intercalent les négatifs proches. La similarité semble déjà faire ses preuves de par la bonne répartition des données sur les histogrammes 1 et 2. On peut constater que la méthode du support (histogrammes 3 et 4) ou de l'indice d'implication (histogrammes 5 et 6) accentuent encore plus la séparation des classes. La méthode du support semble se comporter de manière à regrouper les classes et induit des séparations plus importantes. La méthode de l'indice d'implication quant à elle, engendre des validations plus étalées, mais augment la distance avec l'autre sous-famille ainsi qu'avec les séquences non MIP alors qu'elles n'étaient pas utilisées lors de l'apprentissage et du calcul de l'indice.

L'augmentation de la taille moyenne des automates va de pair avec l'augmentation de la précision des motifs engendrés. Cette précision entraîne une plus grande facilité à rejeter les échantillons négatifs, ce qui est recherché et montre la qualité du modèle, y compris sur des tailles d'automates dépassant les 100 transitions. Mais on peut aussi constater pratiquement sur les graphiques que l'apprentissage des GIpF est plus difficile lorsque l'on devient plus précis. On relativisera cependant, car ceci est dû en partie à un jeu d'apprentissage de taille très réduite. Et de plus, les histogrammes nous ont montré de quelle manière les séquences sont reconnues et à quelle distance elles se situent. Cette analyse par groupes successifs nous permet d'être confiant sur le pouvoir de prédiction de notre méthode.

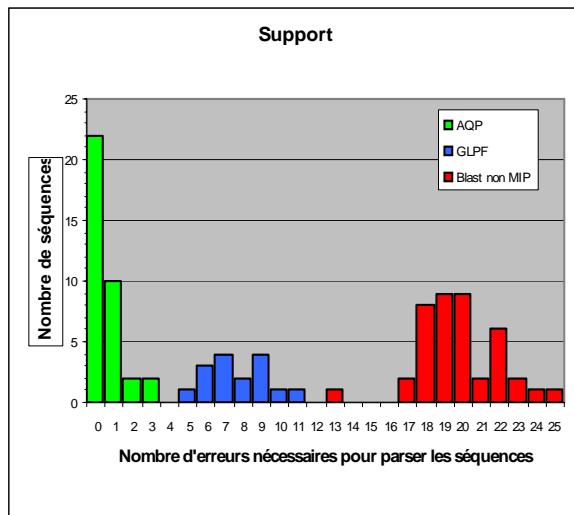
On assiste également à l'apparition de phénomènes de convergence. On sait par les observations biologiques que les MIP sont caractérisées par deux boîtes NPA. L'augmentation de la valeur de  $n$  se traduit par des convergences successives dues à l'apparition du motif d'une boîte NPA (environ 40 transitions dans l'automate), puis de la deuxième, etc.

La caractérisation des sous-familles AQP et GlpF est déjà avérée comme bonne par la méthode simple de similarité. Les heuristiques plus poussées, de support et d'implication, permettent néanmoins encore d'améliorer les résultats par une plus forte discrimination avec les « blast non MIP », mais aussi avec la sous-famille opposée.

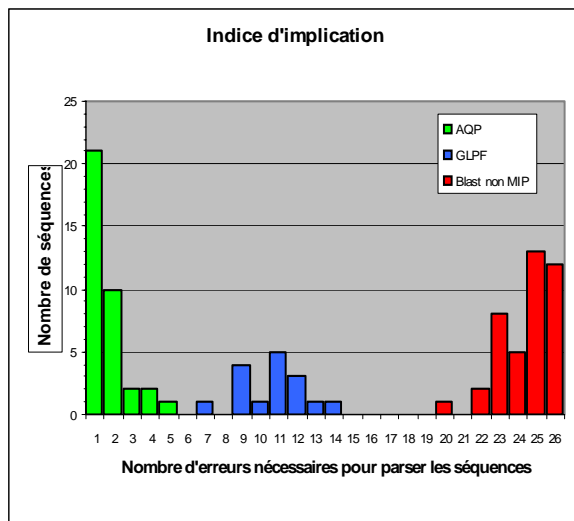
### Apprentissage sur le jeu des AQP



Histogramme 1.

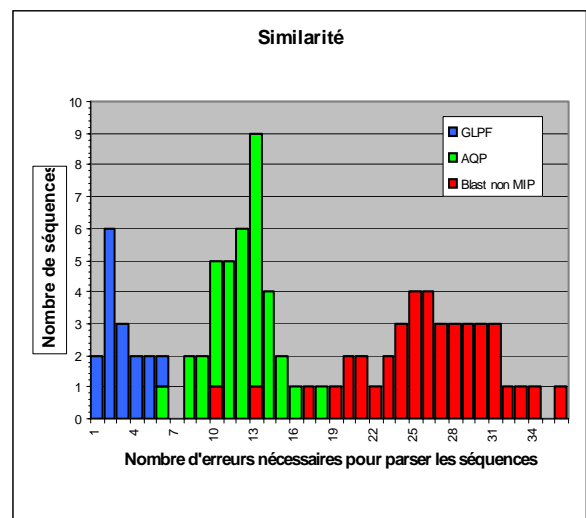


Histogramme 3.

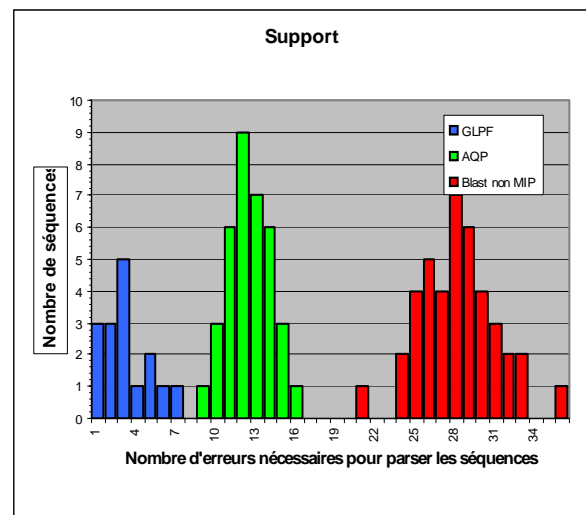


Histogramme 5.

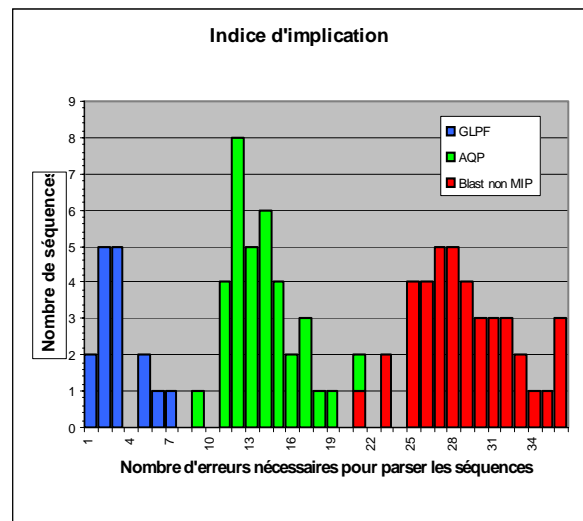
### Apprentissage sur le jeu des GlpF



Histogramme 2.



Histogramme 4.



Histogramme 6.

## 7 Perspectives

Nous avons proposé un programme permettant d'apprendre, pour la première fois à notre connaissance, des automates pertinents pour la caractérisation de séquences protéique. Les premières expérimentations effectuées montrent que les automates obtenus peuvent être utilisés pour la prédiction, ce qui permet une première validation de l'approche, du moins sur la famille des protéines MIP. L'approche par fusion de fragments significativement similaires est originale et permet l'introduction de différentes heuristiques en fonction du style de caractérisation recherché ainsi que des schémas de généralisation découlant naturellement de celle-ci.

Cependant, même si l'approche paraît intéressante, la méthode doit encore être développée pour permettre une modélisation encore plus fine des sous-familles des protéines MIP et pouvoir être confrontée à d'autres familles de protéines. Le but est de trouver des automates ne nécessitant aucune correction d'erreur pour prédire parfaitement la classe des séquences dans une évaluation par *leave one out*. Un premier travail consiste à étudier l'influence et perfectionner les phases de post-généralisation qui sont pour l'instant des implémentations très simples des idées présentées. Au niveau de l'ordonnement des paires de fragments, des progrès peuvent encore être réalisés à l'aide d'indices plus élaborés et le processus de sélection des paires de fragments significativement similaires peut lui aussi être remis en cause. Il est également nécessaire d'automatiser encore plus le processus d'apprentissage. L'introduction d'une mesure MDL (pour sélectionner les meilleurs automates) et d'un mécanisme de validation croisée devrait permettre d'aller dans ce sens.

Le problème de la caractérisation « différentielle » de sous-familles nous intéresse bien sûr vivement et reste à être proprement défini. La normalisation des indices est une piste qui permettrait de réaliser l'apprentissage sur les deux sous-familles simultanément et qui permettrait d'affiner la caractérisation pour la discrimination, des mesures d'information pourraient aussi être utilisées. A un niveau plus global, il nous semble important d'intégrer un mécanisme de détection de co-variations semblable à celui introduit pour Pratt dans [4]. En effet ce type de co-variations, au niveau d'un seul acide aminé, est actuellement indétectable par notre méthode et nous semble être primordiale pour capturer plus d'information structurelle sur la conformation spatiale de la séquence.

## Bibliographie

- [1] Lawrence CE, Altschul SF, Bogouski MS, Liu JS, Neuwald AF, Wooten JC, Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignments, *Science*, 1993, 262, 208-214.
- [2] Grate L, Hughey R, Karplus K, Sjölander K, Stochastic Modeling Techniques: Understanding and using hidden Markov models, University of California, Santa Cruz, CA, June 1996.
- [3] Jonassen I, Collins JF, Higgins D, Finding flexible patterns in unaligned protein sequences. *Protein Science* 1995;4(8):1587-1595.
- [4] Brazma A, Jonassen I, Eidhammer I, Ukkonen E, Relation patterns and their automatic discovery in biosequences, Dept. of Informatics, Univ. of Bergen, Reports in Informatics no 135, Juin 1997.
- [5] Taylor WR, The classification of amino acid conservation, *J Theor Biol.* 1986 Mar 21;119(2):205-18.
- [6] Dupont P, Miclet L, Inférence grammaticale régulière : fondements théoriques et principaux algorithmes. Technical report, INRIA 3449, 1998.
- [7] Lang KJ, Pearlmutter BA, and Price R. Results of the Abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. In *Grammatical Inference*, number 1433 in *Lecture Notes in Artificial Intelligence*, pages 1-12. Springer-Verlag, 1998.
- [8] Morgenstern B, DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics* 1999, 15, 211 – 218.
- [9] Lerman IC, Azé J, Une mesure probabiliste contextuelle discriminante de qualité des règles d'association, dans EGC2003, Extraction des Connaissances et Apprentissage, RSTI série RIA-ECA-Vol 17-n°1-2-3/2003, Hacid D.B MS., Kodratoff Y(Eds), p. 247-262, 2003.
- [10] Agre P, Kozono D., Aquaporin water channels: molecular mechanisms for human diseases, *FEBS Lett.* 2003, 555, 72-78
- [11] Sui H, Han BG, Lee JK, Walian P, Jap BK, Structural basis of water-specific transport through the AQP1 water channel. *Nature.* 2001 Dec 20-27;414(6866):872-8.
- [12] Fu D, Libson A, Miercke LJ, Weitzman C, Nollert P, Krucinski J, Stroud RM. Structure of a glycerol-conducting channel and the basis for its selectivity. *Science.* 2000 Oct 20;290(5491):481-6.
- [13] Fujiyoshi Y, Mitsuoka K, de Groot BL, Philippsen A, Grubmuller H, Agre P, Engel A. Structure and function of water channels. *Curr Opin Struct Biol.* 2002 Aug;12(4):509-15.
- [14] Stroud RM, Miercke LJ, O'Connell J, Khademi S, Lee JK, Remis J, Harries W, Robles Y, Akhavan D. Glycerol facilitator GlpF and the associated aquaporin family of channels. *Curr Opin Struct Biol.* 2003 Aug;13(4):424-31.