Détection de faibles homologies de protéines par machines à vecteurs de support

Jérôme Mikolajczak*, Gérard Ramstein**
Yannick Jacques*

* Département de Cancérologie, Institut de Biologie
9 Quai Moncousu, F-44035 Nantes cedex
jmikolaj@nantes.inserm.fr, yjacques@nantes.inserm.fr
**LINA, équipe EGC, Ecole polytechnique de l'Université de Nantes
Rue Christian Pauc, BP 50609 44306 Nantes cedex 3
gerard.ramstein@polytech.univ-nantes.fr

Résumé. Cet article décrit une approche discriminative pour la recherche de nouveaux membres dans des familles de protéines à faibles homologies de séquences. L'originalité de la méthode repose sur une modélisation de ces familles par un ensemble M de motifs intégrant les propriétés physicochimiques des résidus. Nous proposons un algorithme de découverte de motifs suivant le paradigme de la classification hiérarchique ascendante. L'ensemble M définit un espace de représentation des séquences : chaque séquence est transformée en un vecteur indiquant la présence ou l'absence de chaque motif appartenant à M. Nous utilisons la technique d'apprentissage par machine à vecteurs de support (SVM) pour discriminer la famille d'intérêt vis-à-vis des séquences non apparentées. Cette méthode est testée sur la famille biologique des interleukines dont les membres possèdent des homologies de séquences faibles en dépit d'un repliement tridimensionnel en hélices alpha très conservé. Nous montrons que l'ensemble des motifs hiérarchiques modélise spécifiquement les interleukines par rapport aux autres familles structurales de la base de données SCOP (1.51). Notre classifieur est en effet plus performant sur notre famille de protéines que d'autres méthodes de classification dont le SVM basé sur les spectres de chaîne.

1 Introduction

La découverte de nouveaux membres d'une famille de protéines repose sur deux types de techniques. La plus courante est basée sur une mesure d'homologie de la protéine candidate avec un motif spécifique caractéristique de la famille d'intérêt. Cette méthode consiste à fouiller le génome à partir d'outils bioinformatiques tels que BLAST [Altschul *et al.*, 1990]. Certaines familles de protéine sont trop hétérogènes pour qu'on puisse retrouver des régions conservées au niveau de leur structure primaire. Pour lever cette difficulté, une démarche alternative a été suggérée par plusieurs auteurs [Jaakola *et al.*, 2000]. Elle est fondée sur des méthodes d'apprentissage dans lesquelles les séquences de protéines sont étiquetées selon leur appartenance ou non à la famille recherchée. Les exemples positifs (étiquette +1) regroupent les membres connus de la

famille. Les contre-exemples (étiquette -1) peuvent être extraits au sein de familles non apparentées. Une approche particulièrement prometteuse dans le domaine de la classification supervisée repose sur les machines à vecteurs de support [Vapnik, 1995] (ou *support vector machines*, nommées SVMs par la suite). Dans cette technique, le jeu d'apprentissage subit une transformation en un ensemble de vecteurs de taille fixe. Dans notre classe d'application, les séquences primaires des protéines seront donc projetées dans un espace vectoriel. Plusieurs espaces vectoriels ont été proposés avec des performances remarquables. Une méthode particulièrement efficace et rapide utilise des spectres de chaîne [Leslie *et al.*, 2002]. Ce type de représentation est abondamment utilisé en fouille de textes [Jalam et Teytaud, 2001]. Un spectre de chaîne regroupe toutes les combinaisons possibles de séquences de n caractères (ou n-gramme) à partir d'un alphabet Ω. Le spectre de chaîne d'une séquence est donc un vecteur représenté par les occurrences de ses k-sous-séquences. Il est à noter que l'espace de représentation est de haute dimension ($|\Omega|^n$ combinaisons possibles de n-grammes). La technique du spectre de chaîne est très simple à mettre en oeuvre et peu coûteuse en temps d'exécution. Les auteurs montrent que la performance de leur algorithme est comparable avec celle faisant intervenir des méthodes complexes, comme les HMMs [Karplus *et al.*, 1998]. Nos propres expérimentations sur la famille des cytokines démontrent l'efficacité de cette méthode en terme de classification. Il se trouve que notre famille d'intérêt possède des membres très éloignés entre eux en terme d'homologie de séquence. Nous proposons dans cet article d'utiliser un espace de représentation de faible dimension qui cible des propriétés spécifiques de notre famille d'intérêt. Nous allons dans un premier temps décrire le concept de motif hiérarchique, puis nous donnerons un algorithme d'extraction de ces motifs. Nous rappelerons ensuite les principes des SVMs avant de discuter des résultats obtenus sur la famille des cytokines.

2 Motifs hiérarchiques

La structure primaire d'une protéine est représentée par une séquence $s = \langle s_1 s_2 \dots s_n \rangle$ où chaque s_i appartient à Ω, l'ensemble des acides aminés :

$$\Omega = \{A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y\}$$

Soit $P(\Omega)$ l'ensemble des parties de Ω. Certaines de ces parties possèdent des résidus partageant des propriétés physico-chimiques particulières. Plusieurs variantes de systèmes de classes ont été proposées ; nous avons opté pour celui de Taylor [Taylor, 1986] présenté en table 1. La pertinence de cette classification se vérifie par l'étude des régions conservées : on observe que les mutations s'opèrent généralement au sein d'une même classe (par exemple, les acides aminés I, L et V appartenant à la classe aliphatique sont très fréquemment interchangés). Les classes physico-chimiques définissent un sous-ensemble de $P(\Omega)$, auquel nous ajoutons l'ensemble des singletons de Ω ainsi que l'ensemble Ω lui-même. Nous noterons $C(\Omega)$ l'alphabet suivant :

$$C(\Omega) = \{\{A\},\{C\},\dots,\{Y\}\} \cup \{\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\eta,\theta\} \cup \Omega$$

On considérera l'ensemble ordonné $(C(\Omega), \subseteq)$ qui forme un sup-demi-treillis : toute paire (x,y) de $C(\Omega) \times C(\Omega)$ possède une borne supérieure, que l'on notera $sup(x,y)$.

Symbole	Classe	Membres
α	aliphatique	ILV
β	aromatique	$FHWY$
γ	non polaire	$ACFGHIKLMVWY$
δ	chargé	$DEHKR$
ε	polaire	$CDEHKNQRSTWY$
ζ	charge positive	HKR
η	chaîne latérale courte	$ACDGNPSTV$
θ	chaîne latérale très courte	$ACGST$

TAB. 1 – *Classes d'acides aminés basées sur des propriétés physico-chimiques*

Un motif $m = \langle m_1 m_2 \dots m_k \rangle$ est une k-séquence formée d'ensembles $m_i \in C(\Omega)$. Pour la simplicité de la notation, on notera le singleton $\{R\}$ par R directement ; le motif $K\alpha$ désignera ainsi le motif composé de la classe $\{K\}$ suivie de la classe $\{I,L,V\}$.

Bien que rien n'interdise dans notre méthode d'utiliser des motifs de taille k variable, nous supposerons pour la simplicité de l'exposé que k est fixe. Nous appelerons *occurrence* d'un motif m une sous-séquence $\langle s_{i+1} s_{i+2} \dots s_{i+k} \rangle$ de s telle que $s_{i+j} \in m_j \ \forall j, 1 \leq j \leq k$. On dira que la séquence s *vérifie* le motif m. Le *support* d'un motif m dans un jeu de séquences $\mathcal{S}$ est le nombre de séquences de $\mathcal{S}$ qui vérifie m. La séquence MH vérifie ainsi 21 motifs de taille 2, dont les motifs MH, $M\beta$, $\gamma\delta$, et $\Omega\Omega$. Le motif MH ne peut être vérifié que par une seule sous-séquence, tandis que le motif $\Omega\Omega$ est vérifié pour n'importe quelle séquence de taille supérieure ou égale à 2. Il importe donc de qualifier la spécificité d'un motif en prenant en compte la probabilité de le voir apparaître dans une séquence. Comme l'estimation précise de cette probabilité est complexe et inutile pour notre classifieur, nous avons opté pour la fonction de coût suivante : $c(m) = \prod_{i=1}^{k} f(m_i)$ où $f(m_i)$ est la fréquence de la classe m_i dans une base d'apprentissage comprenant de nombreuses familles de protéines différentes. La spécificité d'un motif m sera définie par $\phi(m) = -\log(c(m))$.

La table 2 montre qu'on observe une bonne corrélation entre l'estimation $\phi(m)$ et le support effectif de m dans la base de contre-exemples de séquences issues de SCOP [Murzin *et al.*, 1995].

Nous avons défini deux propriétés d'un motif hiérarchique : son support et sa spécificité. Il est important de remarquer que ces deux caractéristiques sont généralement opposées. Plus un motif possède un support élevé, moins il est spécifique, comme le montre l'exemple extrême d'un motif composé uniquement de la classe Ω. A l'inverse, un motif uniquement composé de singletons est fortement spécifique mais aura peu de chances d'être découvert.

Les motifs peuvent être hiérarchisés selon une relation de généralisation. Soit deux k-motifs m^1 et m^2. Nous noterons $\preceq$ la relation d'ordre suivante :

$m^1 \preceq m^2$ ssi pour tout $i \in [1,k]$ on a $m_i^1 \subseteq m_i^2$. L'estimateur $\phi(m)$ est construit de sorte que $\phi(m^1) \geq \phi(m^2)$ pour toute paire de motifs vérifiant $m^1 \preceq m^2$. En spécialisant un motif, on augmente sa spécificité. Nous appellerons borne supérieure des motifs m^1 et m^2 (notée $sup(m^1,m^2)$) le motif $m^{1,2}$ vérifiant :

$m_i^{1,2} = sup(m_i^1, m_i^2)$ pour tout $i \in [1,k]$.

motif	support	spécificité	fréquence
$\beta\alpha\varepsilon$	1.00	4.4	0.7984
$\alpha\delta\varepsilon\alpha$	0.96	5.1	0.6095
$\delta\Omega\gamma\varepsilon\Omega\alpha\zeta\varepsilon$	0.65	6.8	0.2427
$\Omega\alpha\zeta\varepsilon\alpha\varepsilon\varepsilon\gamma$	0.54	7.6	0.1130
LEE	0.28	7.9	0.0893
$\beta\gamma\Omega\Omega\gamma\zeta\varepsilon L$	0.28	8.4	0.0433
$\varepsilon\gamma\alpha\zeta\delta L\Omega\varepsilon$	0.37	9.2	0.0359
$L\varepsilon\varepsilon\gamma\alpha\varepsilon\delta L$	0.22	10.2	0.0107
$\Omega\eta L\alpha L\alpha\Omega L$	0.15	11.0	0.0019
$F\varepsilon R\gamma K\varepsilon\Omega\gamma$	0.15	11.4	0.0018

TAB. 2 – *Exemples de motifs avec leur support dans la famille des cytokines, leur spécificité estimée et leur fréquence effective dans la base SCOP.*

Le motif $m^{1,2}$ représente le motif le plus spécifique qui généralise m^1 et m^2 : toute sous-séquence vérifiant m^1 ou m^2 vérifiera $m^{1,2}$. La table 3 donne des exemples de bornes supérieures pour des motifs de taille 4. La section suivante présente comment extraire les motifs de spécificité minimale.

3 Découverte de motifs hiérarchiques

La recherche de motifs hiérarchiques procède en deux étapes :

1. L'extraction de motifs germes,
2. La génération des motifs hiérarchiques.

La première étape consiste à extraire des motifs germes à partir de la famille $\mathcal{S}$. Un motif germe est un motif qui ne possède pas de minorants. Plus pratiquement, les motifs germes sont les motifs formés uniquement de classes singletons (résidus). Une façon triviale d'obtenir la liste des motifs germes est de relever l'ensemble des k-sous-séquences présentes dans le jeu d'apprentissage. Nous avons procédé à un filtrage en ne considérant que les k-sous-séquences potentiellement intéressantes. Cette première étape consiste à croiser chaque k-sous-séquence avec une autre : chaque couple de k-sous-séquences fournit un motif. Si ce dernier possède un support suffisant, les k-sous-séquences vérifiant ce motif sont retenues. Avec un support minimal de 3, nous avons pu réduire ainsi le nombre de motifs germes d'un facteur 8.

La seconde étape opère un appariement des motifs pour déterminer leurs bornes supérieures. L'algorithme de découverte de motifs consiste donc à généraliser les motifs afin de rechercher des caractéristiques de support suffisamment représentatives. La définition précédente de borne supérieure permet de définir le motif commun le plus spécifique à partir de deux motifs. Il est donc possible par itérations successives d'agréger des motifs afin d'obtenir des motifs de support supérieur. La deuxième étape de notre algorithme s'inspire de la technique de la classification hiérarchique ascendante pour former des clusters de motifs généraux à partir de motifs germes composés

uniquement de singletons. La deuxième étape de l'algorithme de découverte de motifs est présentée ci-dessous :

algorithme découverteMotifs
entrées
M, l'ensemble de motifs germes obtenus lors de l'étape 1
$supMin$, le seuil de support minimal recherché
$speMin$, le seuil de spécificité minimale recherchée
sortie
E, l'ensemble de motifs hiérarchiques

$E = \{m \in M \mid support(m) \geq supMin$ et $\phi(m) \geq speMin\}$;
Répéter
Soit m^1 et m^2 la paire de motifs de M telle que :
1. $m^{1,2} = sup(m^1,m^2)$
2. $\phi(m^{1,2}) \geq \phi(m^{i,j})$ pour tout m^i et m^j dans M
$M \leftarrow M - \{m^1,m^2\}$;
$M \leftarrow M \cup \{m^{1,2}\}$;
si $support(m^{1,2}) \geq supMin$ et $\phi(m^{1,2}) \geq speMin$
alors $E \leftarrow E \cup \{m^{1,2}\}$;
jusqu'à $cardinal(M) = 1$ ou $\phi(m^{1,2}) < speMin$;

La table 3 présente les motifs hiérarchiques de taille $k = 4$ relatifs aux séquences HIWY, HIDY, KLTY, HVSG et DARG. Les motifs m^1 à m^5 sont les motifs germes extraits des sous-séquences de taille 4 dans le jeu de séquences précédent. La figure 1 montre la construction hiérarchique des motifs par l'algorithme découverteMotifs. A supposer que l'on se soit fixé un support minimal de 3 et une spécificité de 3.0, seul le motif m^6 serait retenu.

motif	chaîne	support	spécificité
m^1	HIWY	1	14.25
m^2	HIDY	1	12.83
m^3	KLTY	1	11.44
m^4	HVSG	1	11.79
m^5	DARG	1	10.96
$m^6 = m^{1,2}$	KIεY	2	10.64
$m^7 = m^{3,6}$	$\zeta\alpha\varepsilon$Y	3	7.55
$m^8 = m^{4,5}$	$\delta\eta\varepsilon$G	2	5.26
$m^9 = m^{7,8}$	$\delta\Omega\varepsilon\gamma$	5	2.56

TAB. 3 – *Motifs extraits par l'algorithme découverteMotifs.*

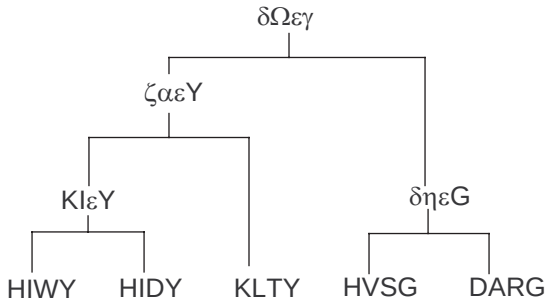

FIG. 1 – *Hiérarchie de motifs à partir de 5 motifs germes*

Notre algorithme diffère dans sa finalité des techniques usuelles d'extraction de motifs. Alors que ces dernières visent à découvrir un nombre restreint de motifs ayant le support le plus large possible (ce qui s'avère quasi impossible pour des familles fortement hétérogènes), nous cherchons au contraire un ensemble élevé M de motifs au support très variable. Le caractère hiérarchique de l'algorithme nous permet de trouver un nombre conséquent de motifs tout en s'affranchissant des problèmes de combinatoire.

Pour utiliser les machines à vecteurs de support, nous allons transformer une séquence quelconque en un vecteur booléen de dimension n (n désignant le cardinal de M). L'élément de rang i est à vrai ssi le motif de rang i dans M est présent dans la séquence (une alternative consiste à compter les occurrences des motifs, mais dans la pratique, il est fort rare de rencontrer deux fois le même motif dans une même séquence pour des valeurs de k importantes). Il est à noter que cette vectorisation s'effectue en $O(N)$, où N désigne la taille de la séquence. Si l'apprentissage est relativement complexe, la classification est peu coûteuse en temps d'exécution. La table 4 montre les valeurs de vecteurs associés aux séquences de l'exemple précédent en considérant l'ensemble des motifs donnés en table 3 (support et spécificité minimaux fixés à 0).

séquence	représentation vectorielle
HIWY	100001101
HIDY	010001101
KLTY	001000101
HVSG	000100011
DARG	000010011

TAB. 4 – *Représentation vectorielle des séquences à partir des motifs extraits. Le vecteur représente neuf valeurs booléennes (1 pour vrai, 0 pour faux) correspondant à la présence du motif m^i, $i = 1$ à 9, dans la séquence considérée.*

4 Machines à vecteurs de support

Cette section introduit la méthode des machines à vecteurs de support [Vapnik, 1995], [Vapnik, 1998]. Cette technique de classification supervisée est basée sur l'apprentissage d'une frontière de décision linéaire qui discrimine au mieux deux classes formées par des exemples et des contre-exemples. Le SVM optimise la frontière de décision par un hyperplan qui maximise la distance entre les points voisins (ou vecteurs de support) durant la phase d'apprentissage. Dans la phase de classification, un point est classé en exemple ou contre-exemple selon sa position par rapport à la frontière de décision. Dans la plupart des problèmes concrets, il y a peu de chances qu'on puisse trouver une séparation linéaire parfaite dans l'espace χ des données. L'efficacité remarquable des SVMs repose sur une transformation non linéaire de l'espace d'entrée χ en un nouvel espace $\Phi(\chi)$ de redescripteurs de plus grande dimension dans lequel les points sont séparables par un hyperplan.

Des travaux théoriques ont permis de montrer que le problème d'optimisation des SVMs revient à résoudre un problème de forme duale qui dépend des produits scalaires entre les vecteurs des exemples $\Phi(x)$ de l'ensemble d'apprentissage dans l'espace de redescription. De ce fait, toute la difficulté repose sur le calcul des produits scalaires $< \Phi(x),\Phi(y) >$ entre chaque couple (x,y) de vecteurs d'exemples dans un espace de redescription $\Phi(\chi)$ de dimension très élevée, voire infinie. Cette difficulté est levée grâce à l'introduction de fonctions bilinéaires symétriques positives $K(x,y)$ appelées *fonctions noyaux*. Les fonctions noyaux permettent d'effectuer tous les calculs nécessaires dans l'espace des données sans jamais devoir passer par l'espace des descripteurs : $K(x,y) =< \Phi(x),\Phi(y) >$. Il existe trois types de fonctions noyaux K simples : les fonctions polynomiales, les fonctions à bases radiales (RBF) et les fonctions sigmoïdes. Les fonctions polynomiales sont définies par $K(x,y) = (x.y + 1)^p$. Le degré du polynome est choisi par l'utilisateur. Les fonctions à bases radiales RBF sont définies par $K(x,y) = e^{\|x-y\|^2/2\sigma^2}$. La valeur de l'écart type σ est estimée empiriquement. Les fonctions sigmoïdes sont de la forme $K(x,y) = tanh(a(x.y - b)$. En pratique, il s'agit de tester les différentes fonctions noyaux pour déterminer celle avec laquelle on obtient l'hyperplan optimal et la marge minimale. Dans le cadre de ce travail, nous avons testé la fonction noyau linéaire (polynome de degré 1) et les fonctions à bases radiales.

La première application des SVMs pour la classification d'homologues éloignées dans les familles protéiques est la méthode des Fisher SVMs [Jaakola *et al.*, 2000]. Cette méthode nécessite tout d'abord l'apprentissage d'un modèle de Markov caché (HMM) sur la famille de protéines d'intéret [Karplus *et al.*, 1998]. Ensuite un jeu d'apprentissage constitué d'exemples et de contre-exemples est représenté dans un espace vectoriel dont chaque dimension est liée à un état ou une transition présent dans le modèle. La valeur de chaque dimension du vecteur est appelée score de Fisher et représente le taux d'utilisation de chaque paramètre du modèle pour modéliser chaque séquence exemple. Les vecteurs obtenus sont utilisés conjointement avec une fonction noyau particulière, appelée fonction de Fisher, pour l'apprentissage du modèle SVM. Les auteurs ont montré que les classifications par la méthode des Fisher SVMs surpassent les méthodes génératives telles que les HMMs ou BLAST. Cependant cette méthode trouve ses limites dans la nécessité d'avoir une famille de protéines possédant

un nombre important de membres pour pouvoir élaborer le modèle HMM. Elle se caractérise aussi par une complexité temporelle importante autant en phase d'apprentissage que lors de la prédiction de la classe d'une protéine. La méthode des pairwise SVMs [Li et Noble, 2003] utilise l'algorithme d'alignement local de Smith et Waterman. Les exemples sont vectorisés dans un espace dont la dimension correspond au cardinal du jeu d'apprentissage. Chaque dimension du vecteur correspond au score de similarité obtenu par l'alignement local avec une séquence du jeu d'apprentissage. Les auteurs de cette méthode ont obtenu des performances de classification supérieure à la méthode des Fisher SVMs. Cependant elle ne résoud pas le problème de la complexité temporelle et rend la dimension des vecteurs dépendante de la taille du jeu d'apprentissage. Une alternative au problème de complexité temporelle est le spectre de chaîne ou *String Spectrum* [Leslie *et al.*, 2002], dont nous avons présenté le principe en introduction. Dans cette méthode, chaque vecteur représente les fréquences des n-grammes trouvés sur la séquence de l'exemple. Les vecteurs servent ensuite à l'apprentissage du modèle SVM. Ce simple procédé s'affranchit des méthodes génératives et permet un gain de temps d'exécution remarquable. Une adaptation de cette méthode prend en compte tous les motifs de taille n fixée et autorise au plus m variations entre deux motifs [Leslie *et al.*, 2004]. Cette adaptation prend en compte le concept biologique de mutation des résidus. Les auteurs de cette technique montrent que la performance de leur classification est comparable à la méthode des Fisher SVMs tout en optimisant les temps d'exécution. Nos propres expérimentations n'ont pas révélé une amélioration des performances sur notre famille d'intérêt par rapport à la méthode de spectre de chaîne. C'est donc cette dernière méthode qui nous servira de point de comparaison avec notre propre algorithme de classification. Les méthodes décrites jusqu'à présent ne prennent en compte que l'information située au niveau de la structure primaire des protéines. La méthode des SVM-I-sites [Hou *et al.*, 2003] tente d'intégrer une information structurale lors de l'étape de vectorisation des exemples. Une étape préalable consiste à rechercher un profil de séquences à partir de la séquence primaire de l'exemple. Le profil est obtenu au moyen du logiciel PSI-BLAST [Altschul *et al.*, 1997] et de la base de données Swiss-Prot [Boeckmann *et al.*, 2003]. La création des vecteurs se base ensuite sur la recherche des occurences de 263 domaines structuraux au sein du profil. Les auteurs indiquent que cette méthode apporte des résultats équivalents à la méthode des pairwise SVMs. Elle est aussi plus efficace lors de l'étape de vectorisation.

Les SVMs ont démontré leur efficacité dans la détection d'homologies éloignées. Les différents types de classification présentés mettent en évidence l'importance de l'étape de vectorisation des exemples dans la performance de ce type de classifieur.

5 Classification des protéines utilisant les SVMs

La classification des protéines passe par une première étape de vectorisation suivie de la prédiction proprement dite par SVM. Dans notre application, la transformation des séquences s'opère par la détection des motifs retenus par l'algorithme découverteMotifs. Les séquences étant représentées par des vecteurs booléens de taille fixe, il est possible de définir le produit scalaire entre deux séquences s et s' par le nombre d'éléments à vrai dans le vecteur $x \wedge x'$ où x et x' désignent les vecteurs

booléens associés respectivement à s et s'. Le choix de la fonction noyau a été dictée par certaines propriétés spécifiques à notre espace de redescription. En effet, soit O le vecteur nul. La fonction noyau doit vérifier les points suivants :

- si $x \simeq O$ et $x' \simeq O$ alors $K(x,x')$ doit avoir une valeur proche de la valeur maximale de la fonction noyau. En effet, dans ce cas, les deux séquences appartiennent toutes deux à la classe des contre-exemples.
- si $x \simeq O$ et x' appartient à la classe des exemples, alors la valeur de $K(x,x')$ doit être d'autant plus faible que l'exemple contient un nombre de motifs important.

Nous avons retenu les fonctions à bases radiales qui vérifient bien ces propriétés (parce qu'elles reposent sur le calcul de $\| x - x' \|$, contrairement aux fonctions polynomiales basées sur le calcul de $x.x'$). Il est à noter que les fonctions à bases radiales donnent généralement de meilleurs résultats que les fonctions polynomiales et sigmoïdes.

Si l'utilisation des SVMs ne pose aucun problème d'adaptation, il est cependant important d'observer qu'à l'origine les SVMs définissent une frontière optimale entre deux classes d'individus d'intérêt égal. Dans notre type d'application, l'importance égale de ces deux classes soulève une difficulté. S'il est en effet facile de définir ce qu'est la famille d'intérêt, la définition des contre-exemples est moins évidente. Une solution consiste à utiliser des représentants dans chaque superfamille défini par la base SCOP. Il faut alors veiller à la robustesse de ces choix et à la sensibilité des SVMs face au déséquilibre entre le nombre des exemples et celui des contre-exemples. Nous avons effectué des tests portant sur des échantillons aléatoires de contre-exemples de même taille que le jeu d'exemples positifs et répartis uniformément parmi les classes structurales de la base SCOP. Ces tests montrent que les performances de la classification sont peu sensibles au tirage effectué, mais se dégradent lorsque le nombre de contre-exemples s'accroît par rapport au nombre d'exemples positifs. Une autre solution consiste à utiliser la version de SVMs basée sur une classe unique, proposée par B. Schölkopf [Schölkopf *et al.*, 2001]. Les résultats obtenus avec cette méthode se sont montrés inférieurs à ceux des SVMs à deux classes. Nous interprétons ce phénomène par le nombre restreint d'exemples présenté en apprentissage ainsi qu'à leur grande dispersion (faible homologie intra-classe).

Nous avons utilisé le logiciel open-source libsvm [C.Chang et Lin, 2001] pour implémenter notre classifieur SVM.

6 Application à la superfamille des cytokines

Les cytokines ont pour fonction d'assurer la médiation des signaux de prolifération, de différenciation, et d'activation entre les différentes cibles cellulaires. Dans cet article, nous nous intéressons plus particulièrement aux interleukines à hélices courtes (IL-2), hélices longues (IL-6) et aux interleukines de type IL-10 dont les précurseurs possèdent six hélices. Nous avons retenu 45 séquences primaires relatives à la famille des cytokines chez l'homme.

La base d'apprentissage qui nous a servi à l'estimation de la spécificité est issue de la base de données SCOP (Structural Classification Of Proteins, [Murzin *et al.*, 1995]). Les séquences de SCOP forment donc un échantillon qui recouvre un large spectre

de protéines. Après suppression des interleukines, notre base de test comporte 6615 séquences.

La famille qui nous intéresse est caractérisée par une structure secondaire riche en hélice alpha. Le pas de cycle d'une hélice étant de 3.6, nous avons considéré des motifs de taille 8 (des motifs de taille 4 seraient moins discriminants).

La table 5 présente les résultats obtenus avec différents classifieurs. Les valeurs présentées sont des moyennes de performances obtenues par la technique de *leave-one-out* (pour laquelle une séquence sert de test et les autres pour l'apprentissage). La ligne KNN présente les résultats obtenus avec la méthode des k plus proches voisins ($k = 3$). La notion de voisinage se réfère à la proximité entre spectres de chaîne : la mesure de similarité utilisée est le produit cartésien des vecteurs normalisés. Les SVMs à base de spectre de chaînes donnent de meilleurs résultats (ligne SCSVM) que les KNNs, ce qui confirme l'intérêt des machines à vecteurs de support. Les résultats sont identiques pour les fonctions noyaux linéaires et à bases radiales ; la seule différence consiste en une légère amélioration des performances sur la base SCOP pour la fonction à base radiale (table 6). Notre méthode MotifsSVM surpasse largement la technique à base de spectre de chaîne (100% de bonne classification) à condition de bien optimiser le type des motifs. Un seuil de spécificité de 14 apparaît comme le meilleur compromis ; au delà de cette valeur, on ne découvre pas assez de motifs sur certaines cytokines, en deçà, les motifs ne sont assez sélectifs. Un seuil de support minimal de 2 en *leave-one-out* (de 3 en apprentissage normal) est nécessaire pour obtenir de bons résultats dans notre jeu particulier d'application. Il s'avère que les exemples présentés possèdent peu de séquences proches. Nous avons pu vérifier ce fait en calculant les valeurs de fonction noyau à partir de spectres de chaîne : la valeur de fonction noyau tend rapidement vers 0 à partir de la deuxième plus proche voisine.

classifieur	taux d'erreurs	VP	FN	VN	FP
KNN	18.9	88.9	11.1	73.3	26.7
SCSVM linéaire	13.3	84.4	15.6	88.9	11.1
SCSVM RBF	13.3	84.4	15.6	88.9	11.1
MotifsSVM 13	2.2	95.6	4.4	100	0
MotifsSVM 14	0	100	0	100	0
MotifsSVM 15	5.5	88.9	11.1	100	0

TAB. 5 – *Résultats de classification. VP, FN, VN, FP sont les pourcentages respectivement des vrais positifs, faux négatifs, vrais négatifs, faux positifs.*

La table 6 indique le pouvoir discriminant de notre classifieur sur les séquences négatives extraites de la base SCOP. Sur les 6615 séquences de la base SCOP, le classifieur MotifsSVM en a mal classé 8. Le faible pourcentage de faux-positifs autorise l'emploi de MotifsSVM pour rechercher de nouveaux membres dans le génome.

classifieur	taux de FP dans SCOP
SCSVM linéaire	4.32
SCSVM RBF	4.08
MotifsSVM 14	0.12

TAB. 6 – *Taux de faux-positifs dans la base SCOP.*

7 Conclusion et perspectives

Les excellents résultats obtenus en classification dans la famille des cytokines démontrent la pertinence d'une description hiérarchique des motifs. Ces derniers assurent un rôle de signature, au sens où ils sont spécifiques à la famille étudiée. Nous avons proposé un paramétrage simple du degré de spécificité souhaité et avons observé qu'une valeur moyennement haute donne les meilleurs performances ($\phi(m) \sim 14$). La capacité des SVMs à gérer des espaces de grande dimension nous permet d'obtenir une classification sans erreurs sur les exemples positifs et un très faible taux d'erreurs sur les exemples négatifs issus de SCOP (0.12%). Certaines améliorations de l'algorithme d'extraction restent à mettre en oeuvre, afin d'optimiser le nombre de motifs retenus. Sur les quelques 600 motifs extraits, une proportion non négligeable d'entre eux peuvent certainement être éliminés sans nuire à la performance du classifieur. Nous envisageons dans un premier temps de filtrer plus finement les motifs extraits et de procéder ensuite à une étude de leurs co-occurrences dans les séquences. Cette analyse permettra de déterminer des patrons de motifs propres à une famille de protéines (un patron est défini comme une séquence de motifs situés à des intervalles variables sur une même séquence).

Références

[Altschul *et al.*, 1990] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, et D. J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[Altschul *et al.*, 1997] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, et D. J. Lipman. Gapped blast and psi-blast : a new generation of protein database search programs. *Nucleic Acid Research*, 25(3389-3402):17, 1997.

[Boeckmann *et al.*, 2003] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, et M. Schneider. The swiss-prot protein knowledge base and its supplement trembl in 2003. *Nucleic Acid Research*, 31:365–370, 2003.

[C.Chang et Lin, 2001] C. C.Chang et C. J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Hou *et al.*, 2003] Y. Hou, W. Hsu, M. L. lee, et C. Bistroff. Efficient remote homology detection using local structure. *Bioinformatics*, 19(17):2294–2301, 2003.

[Jaakola *et al.*, 2000] T. Jaakola, M. Diekhans, et D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computationnal Biology*, 7(1-2):95–114, 2000.

[Jalam et Teytaud, 2001] Radwan Jalam et Olivier Teytaud. Identification de la langue et catégorisation de textes basées sur les n-grammes. *Extraction de Connaissance et Apprentissage*, 1(1-2):227–238, Janvier 2001.

[Karplus *et al.*, 1998] K. Karplus, C. Barret, et R. Hugley. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14:846–856, 1998.

[Leslie *et al.*, 2002] C. Leslie, E. Eskin, et W. S. Noble. The spectrum kernel : a string kernel for svm protein classification. In *Proceedings of the Pacific Biocomputing Symposium*, pages 564–575, 2002.

[Leslie *et al.*, 2004] C. Leslie, E. Eskin, D. Zhou, et W. S. Noble. Mismatch string kernel for svm protein classification. *Bioinformatics*, 2004. à paraître.

[Li et Noble, 2003] L. Li et W. S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionnary and structural relationships. *Journal of Computationnal Biology*, 10(6):857–868, 2003.

[Murzin *et al.*, 1995] A.G. Murzin, S.E.Brenner, T. Hubbard, et C. Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

[Schölkopf *et al.*, 2001] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, et R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), 2001.

[Taylor, 1986] J. Taylor. Classification of amino acid conservation. *Theorical Biology*, 119:205–218, 1986.

[Vapnik, 1995] V. N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, 1995.

[Vapnik, 1998] V. N. Vapnik. *Statistical Learning Theory.* Springer, 1998.

Summary

This article presents a discriminative approach to the protein classification in the particular case of remote homology. The protein family is modelled by a set M of motifs related to the physicochemical properties of the residues. We propose an algorithm for discovering motifs based on the ascending hierarchical classification paradigm. The set M defines a feature space of the sequences : each sequence is transformed into a vector that indicates the possible presence of the motifs that belongs to M. We then use the SVM learning method to discriminate the target family. Our hierarchical motif set specifically modelises interleukins among all the structural families of the SCOP (1.51) database. Our method yields significantly better remote protein classification compared to spectrum kernel techniques.