Évaluation des méthodes supervisées pour la discrimination de protéines

Ricco Rakotomalala*, Faouzi Mhamdi**

*Laboratoire ERIC – Université Lyon 2
69500 BRON
ricco.rakotomalala@univ-lyon2.fr,
http://eric.univ-lyon2.fr/ ricco
**URPAH – Université d'El Manar
TUNISIE
faouzi.mhamdi@ensi.rnu.tn

Résumé. Nous évaluons différentes méthodes supervisées dans le cadre de la discrimination de protéines. Les descripteurs étant automatiquement générés, nous avons utilisé les n-grammes, la taille de l'espace de représentation est très élevée par rapport au nombre d'observations. Un grand nombre de descripteurs ne sont pas pertinents. Il apparaît que les méthodes linéaires telles que les SVM linéaires ou la régression PLS sont les plus performantes. Une étude détaillée montre que ce succès repose essentiellement sur la robustesse face à la dimensionalité qui devient, de fait, le critère le plus important dès lors que l'on traite des domaines où les descripteurs sont générés automatiquement en grand nombre. Dans ce contexte, une procédure de sélection de variables, fusse-t-elle très fruste, modifie significativement le comportement des algorithmes d'apprentissage.

1 Introduction

L'annotation et le classement de protéines est une activité importante du biologiste. L'augmentation du volume de données à traiter rend nécessaire l'automatisation de cette tâche. Ces dernières années, la fouille de données, plus généralement l'extraction de connaissances à partir de données (Fayyad et al., 1996), a permis de dégager un cadre qui rend reproductible la démarche complète de classement automatique des protéines à partir de leur structure primaire. En effet, une séquence de protéine est décrite par une suite de caractères pris dans un alphabet de 20 signes. Le rapprochement avec les nombreux travaux réalisés dans la catégorisation de textes est naturelle (Sebastiani, 2005). Par rapport à un traitement standard sur des données individus-variables, l'appréhension des données non-structurées introduit deux étapes supplémentaires : l'extraction de descripteurs à partir de la description primaire pour aboutir à un tableau de données et, éventuellement, la sélection des descripteurs les plus discriminants, afin que les algorithmes d'apprentissage puissent fonctionner de manière efficace. Compte tenu du nombre important de descripteurs que nous pouvons générer, la complexité informatique est un critère primordial dans cette démarche.

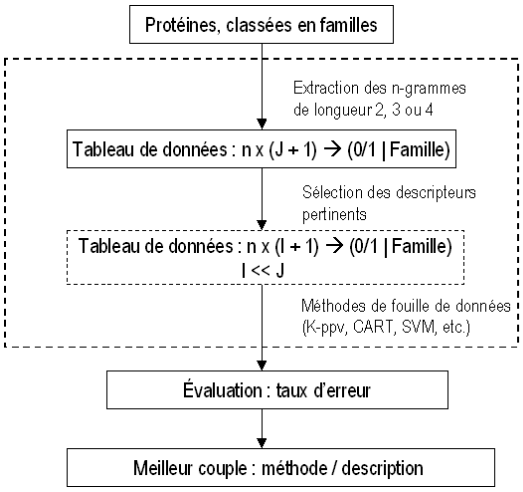

FIG. 1 – *Processus de discrimination de protéines.*

Dans cet article, nous traitons de la discrimination de protéines à partir de leur structure primaire. Une séquence de protéines est décrite par une séries d'acides aminés. Il existe 20 types d'acides aminés. L'analogie entre une chaîne de caractères et une séquence de protéine est raisonnable, à la différence qu'il n'existe pas dans ce cas de séparateurs naturels comme l'espace ou la ponctuation entre des groupes de caractères. Les protéines sont regroupées en familles selon leur fonction. Il est admis que les protéines appartenant à la même famille ont des structures identiques, encore faut-il pouvoir caractériser la similarité entre deux protéines. Notre objectif est de construire une fonction de classement qui permet d'associer automatiquement une protéine à sa famille d'appartenance. Pour ce faire, nous adoptons la démarche classique de la catégorisation de textes : nous constituons un fichier d'apprentissage de M observations à partir de protéines classées manuellement par des experts ; le traitement direct à partir de la forme initiale des données n'étant pas possible, nous extrayons J descripteurs, dans notre cas, des n-grammes ; éventuellement, nous procédons à une sélection des I descripteurs les plus pertinents pour la tâche de discrimination ; nous appliquons les méthodes d'apprentissage supervisé pour produire une fonction de classement ; enfin, nous évaluons la qualité de la démarche en utilisant des méthodes de ré-échantillonnage, la validation croisée en l'occurrence (Figure 1).

Dans la section suivante, nous présentons la démarche de discrimination de protéines, nous mettons en exergue les points que nous évaluerons par la suite. Dans la section 3, nous détaillerons les méthodes d'apprentissage supervisé que nous testerons, nous essayerons surtout de les positionner par rapport aux caractéristiques des fichiers de données que nous utilisons, peu d'observations et beaucoup de descripteurs. Les résultats de l'expérimentation seront étudiés dans la section 4. Nous discuterons de leur portée et des extensions que l'on pourrait apporter en introduisant la sélection de variables dans le section 5. Enfin, nous conclurons dans la 6ème et dernière section.

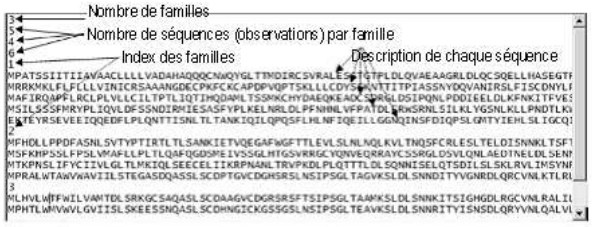

FIG. 2 – *Description native des séquences de protéines.*

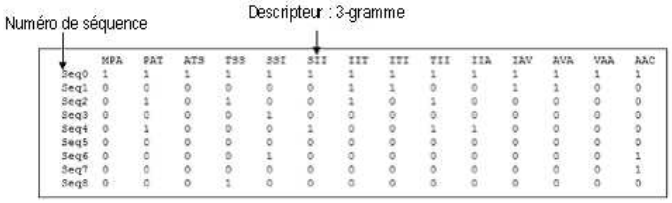

FIG. 3 – *Tableau de données après extraction des* 3*-grammes.*

2 La discrimination de protéines

2.1 Préparation du tableau de données

Les fichiers utilisés dans cet article proviennent de la banque de données SCOP (Murzin et al., 1995). Pour chaque famille de protéines, nous disposons d'une cinquantaine d'observations. Chaque observation est décrite par une chaîne de caractères de longueur variable (Figure 2). Traiter directement cette description native avec les algorithmes usuels de fouille de données n'est pas possible. Il nous faut les transformer de manière à disposer d'un tableau attribut-valeur. En nous inspirant des résultats mis en avant dans la catégorisation de textes (Sebastiani, 2005), nous nous sommes tournés vers la technique des n-grammes.

Un n-gramme correspond à une suite de caractères de longueur n. La transformation consiste à repérer tous les n-grammes possibles dans les fichiers. Chaque n-gramme correspond à un nouveau descripteur, nous signalons dans la colonne la présence ou l'absence du n-gramme pour chaque observation constituant la base d'apprentissage. Nous aboutissons à un tableau de données booléen $0/1$ (Figure 3).

Le premier enjeu est de définir la longueur adéquate du n-gramme, la valeur de n. Si elle est trop faible, par exemple si nous fixons $n = 1$, l'information capturée est trop pauvre. Chaque caractère étant de surcroît présent dans quasiment toutes les séquences de protéines, la colonne correspondante sera remplie de 1 (présence du 1-gramme). Si la valeur de n est trop élevée, l'information capturée devient trop spécifique, nous nous heurtons à deux problèmes. D'une part, le nombre de colonnes du tableau d'apprentissage sera colossal, le nombre théorique de colonnes étant égal à 20^n, le calcul ne sera pas possible dans la plupart des cas. D'autre part, chaque colonne sera de toute manière presque toujours remplie de 0 (absence du n-gramme). La longueur n ne peut donc être que la résultante d'un compromis. Rien ne

nous indique au départ sa valeur adéquate s'agissant de la discrimination de protéines. Notre expérience de la catégorisation de textes nous a tout simplement montré que dans certains cas $n = 3$ donne des résultats probants, dans le classement automatique de nouvelles par exemple (Jalam et al., 2004). Un de nos objectifs justement dans le travail que nous présentons ici, est d'essayer de déterminer la bonne taille de n. Non pas dans l'absolu, mais en relation avec les algorithmes d'apprentissage utilisés pour construire la fonction de classement. Nos descripteurs étant générés automatiquement, un grand nombre d'entre eux ne seront pas pertinents pour la discrimination. La capacité des méthodes d'apprentissage à faire face aux problèmes de dimensionalité sera nécessairement un critère important dans notre évaluation.

Le second enjeu est le choix de la pondération. Dans notre premier descriptif ci-dessus, il est entendu que nous détectons la présence ou absence des n-grammes pour aboutir à un tableau booléen (Figure 3). D'autres indicateurs peuvent être utilisés. Nous pouvons par exemple nous pencher sur l'occurrence, le nombre d'apparition du n-gramme dans les séquences de protéines ; nous pouvons également utiliser leur fréquence d'apparition, rapportée sur le nombre de n-gramme dans la séquence ; le TF/IDF qui rapporte la fréquence d'apparition sur le nombre d'occurrence du n-gramme dans l'ensemble des séquences etc. (Dumais et al., 1998). Selon le cas, nous ne captons pas le même type d'information. Nous avons adopté une approche pragmatique en expérimentant chaque type de pondération. Nos résultats montrent que dans le cadre de la discrimination de protéines, la représentation booléenne présente les mêmes performances en classement que les autres. Nous choisissons donc ce type de pondération qui autorise la mise en oeuvre indifférenciée d'algorithmes d'apprentissage gérant les descripteurs exclusivement discrets (Bayesien naïf), continus (SVM, PLS, PPV) ou les deux (CART).

2.2 Discrimination ou classement ?

Notre idée de départ était d'effectuer un classement. Nous voulions pouvoir assigner à une protéine sa famille d'appartenance parmi toutes les familles existantes. Nous nous sommes rapidement rendus compte que cette tâche était loin d'être triviale. En effet, nous avons procédé de manière habituelle c.-à-d. nous avons constitué un fichier de données contenant des observations de la famille cible, et des représentants d'un ensemble de quelques familles. Si l'apprentissage et son évaluation (validation croisée) ont donné des résultats très encourageants, il en a été tout autrement lorsque nous avons voulu déployer le modèle. En l'appliquant sur n'importe quelle séquence de protéine, dont certaines appartenaient à des familles qui n'ont pas été représentées dans le fichier d'apprentissage, le taux de rappel (la sensibilité – la capacité à reconnaître la famille cible) était très bon. La précision (la capacité à désigner sans erreur la famille cible) était en revanche très dégradée. Notre modèle de prédiction intégrait trop de faux positifs.

Il est impossible de constituer un fichier d'apprentissage comportant un nombre suffisant de représentants de toutes les familles existantes de protéines. Il apparaît très clairement que les méthodes classiques d'apprentissage supervisé ne sont pas adaptées à ce contexte. Il semble qu'il faille se pencher sur les méthodes dites "semi-supervisées" (Chapelle et al., 2003) pour répondre de manière adéquate à ce problème, ce qui définit un cadre complètement différent. Comme il est difficile de traiter tous les problèmes en même temps (choix et sélection des descripteurs, choix de la pondération, évaluation des méthodes, etc.), nous nous sommes recentrés sur la discrimination. Notre objectif, dans cet article, est de mettre en avant les déterminants

Fichier	2-grammes	3-grammes	4-grammes
F_{12}	400	6600	23408
F_{13}	400	6288	22515
F_{14}	400	6183	23662
F_{15}	397	6004	22790
F_{23}	400	7143	31973
F_{24}	400	7098	33185
F_{25}	400	7011	32126
F_{34}	400	6860	31809
F_{35}	400	6740	30954
F_{45}	400	6659	31904

TAB. 1 – *Nombre de descripteurs extraits selon la longueur n des n-grammes.*

qui permettent de discerner au mieux deux familles de protéines également représentés dans le fichier de d'apprentissage.

3 Expérimentation

3.1 Données et évaluation

Pour évaluer notre approche, nous avons sélectionné 5 familles de protéines au hasard parmi toutes les familles disponibles dans la banque de données SCOP (Murzin et al., 1995). Nous disposons approximativement de 50 observations par famille. Nous cherchons à discriminer les familles de protéines deux à deux, 10 fichiers de données comportant chacun une centaine d'observations ont été constitués.

A partir de la description native (Figure 2), nous avons construit des tableaux booléens de données en utilisant le principe des n-grammes (Figure 3). Nous avons fait varier n de 2 à 4. Si le nombre théorique de descripteurs que l'on peut obtenir est égal à 2^n, dans la pratique, nous en observons largement moins à mesure que n augmente (Tableau 1). Dans le cas de $n = 4$ par exemple, nous sommes bien en-deçà de $20^4 = 160000$. Il reste néanmoins que le traitement de fichiers de données comprenant une centaine d'observations et 30000 descripteurs reste une gageure.

Nous mesurons le taux d'erreur à l'aide de la 5×2 validation croisée pour évaluer la qualité de l'apprentissage (Dietterich, 1999). Malgré la faible taille de nos échantillons, nous avons préféré cette procédure à la validation croisée classique ou le "leave-one-out" qui, dans nos expérimentations, produisaient des évaluations très optimistes, et surtout dépendantes de la méthode. L'évaluation "leave-one-out" par exemple nous fournissait un taux d'erreur nul sur plusieurs fichiers lorsque nous utilisons la méthode des plus proches voisins.

Enfin, nous utilisons la moyenne des taux d'erreurs mesurés sur les dix fichiers pour évaluer les combinaisons "méthode d'apprentissage" et "n-grammes". Généralement, il n'est pas très conseillé de noyer dans un indicateur synthétique, comme la moyenne, les résultats obtenus sur plusieurs fichiers "benchmarks". Dans notre cas, cela paraît plus justifié dans la mesure où les familles de protéines ont été tirés au hasard parmi toutes les familles possibles composant

Methode	**Biais**	**Variance**
SVM Linéaire	Linéaire	Faible
PLS (2 axes)	Linéaire	Faible
Bayesien Naïf	Linéaire	Modérément faible
CART	Non-linéaire	Élevée
1-PPV	Non-linéaire	Élevée
SVM RBF	Non-linéaire	Faible

TAB. 2 – *Caractéristiques des méthodes d'apprentissage utilisées*

la banque de données. Chaque fichier opposant deux familles de protéines constitue ainsi une observation représentative d'un problème de discrimination entre deux familles de protéines.

3.2 Méthodes d'apprentissage

Concernant les méthodes d'apprentissage, il existe une multitude de points de vue. Il est bien souvent difficile d'en discerner clairement les caractéristiques. Une bonne manière de procéder est de les positionner selon leur mode de représentation d'une part, selon leur préférence d'apprentissage d'autre part. Il est possible de trouver une vue synthétique très intéressante de ces algorithmes dans l'ouvrage de Hastie et al. (2001).

Le premier point de vue indique la capacité de la méthode à retraduire la "forme" d'un concept. Nous distinguons généralement les modèles linéaires des modèles non-linéaires. A priori, nous avons toujours intérêt à choisir un modèle non-linéaire : "qui peut le plus, peut le moins". En réalité, la situation est un peu plus compliquée. Sur les données synthétiques, le "concept" à apprendre existe vraiment puisque artificiellement créé par le chercheur. Sur les données réelles, la relation entre les descripteurs et la variable à prédire est une vue de l'esprit. Nous essayons de retraduire une hypothétique causalité avec une fonction mathématique. Cette (in)capacité à appréhender les concepts se traduit généralement par le terme de "biais de représentation" dans la littérature, nous pouvons aussi la décrire sous l'angle de la complexité du modèle.

Le second point de vue, la préférence d'apprentissage, décrit le mode d'exploration des solutions. Elle permet de choisir entre deux solutions concurrentes de la même représentation, elle permet aussi de restreindre la recherche. Bien souvent, mais ce n'est pas toujours le cas, les caractéristiques du mode d'exploration est retraduite par le critère à optimiser lors de l'apprentissage (maximum de vraisemblance, moindres carrés, etc.). A priori, nous avons tout intérêt à choisir une méthode qui teste toutes les hypothèses possibles de manière à choisir la meilleure. En réalité, ce n'est pas toujours vrai. Le principal danger est de retrouver des particularités propres au fichier d'apprentissage au détriment de la "vraie" relation que nous voulons mettre en évidence. Notre situation est d'autant plus difficile que nous traitons des données où le nombre de descripteurs est très élevé par rapport aux observations, les fonctions de distributions conditionnelles sont mal estimées. Cette (in)stabilité par rapport au fichier d'apprentissage se traduit généralement par le terme de "variance" dans la littérature.

Les termes "biais" et "variance" sont également utilisés en référence aux composantes de l'erreur quadratique, utilisée pour évaluer les performances des méthodes d'apprentissage.

Nous avons positionné 6 algorithmes d'apprentissage à partir de ces critères (Tableau 2). La première méthode est un **SVM Linéaire** (*Support Vector Machine*). Elle produit un séparateur linéaire qui maximise sa distance avec les exemples et les contre-exemples situés à proximité de la frontière (les vecteurs supports). Sa principale qualité est sa stabilité même dans des espaces de grande dimension. Elle est peu perturbée par des attributs non-pertinents. La seconde qualité des SVM est de pouvoir se projeter dans des espaces différents sans avoir à construire explicitement les descripteurs associés : un séparateur linéaire dans ce nouvel espace devient un séparateur non-linéaire dans l'espace originel. Nous avons donc introduit un **SVM RBF** (un noyau à fonction de base radiale) dans notre expérimentation : c'est bien un modèle non-linéaire mais la préférence d'apprentissage n'est pas modifiée.

Deux autres méthodes linéaires ont été évaluées. Le **bayesien naïf** ou modèle d'indépendance conditionnel appliqué sur des données $0/1$ produit un séparateur linéaire. C'est une méthode relativement stable, elle est néanmoins sensible à la présence de descripteurs non-pertinents. La méthode **PLS** (*Partial Least Square*) est plus utilisée dans le domaine de la régression. A l'instar de la régression linéaire multiple, nous pouvons la mettre en oeuvre dans le cadre de la discrimination. Très brièvement, nous dirons qu'il s'agit d'une analyse en composantes principales supervisée où la construction des axes factoriels dépend directement de la variable à prédire. Nous l'avons choisie car elle présente une qualité très intéressante, nous avons la possibilité de moduler la préférence d'apprentissage en modifiant le nombre d'axes factoriels sélectionnés. Plus nous augmentons le nombre d'axes, plus nous "collons" aux données. A l'extrême, en prenant tous les axes, nous obtenons les mêmes résultats que la régression multiple. En réduisant le nombre d'axes, nous améliorons la stabilité du classifieur, nous réduisons sa variance. Le risque bien sûr est de ne pas appréhender l'information utile contenue dans les données. Dans cette expérimentation, après quelques tâtonnements, il semble que les deux premiers axes suffisent pour obtenir des résultats satisfaisants.

Enfin, nous avons introduit deux méthodes non-linéaires dans notre expérimentation. Les **arbres de décision** (méthode CART) sont bien connus. Ils peuvent représenter toute forme non-linéaire. Mais ils ne sont pas assurés de les trouver car, d'une part, ils procèdent pas-à-pas, au risque de s'enferrer dans un optimum local, et d'autre part, en fractionnant rapidement le fichier de données, ils sont très vite bloquées faute d'observations sur ses feuilles. Elles sont particulièrement handicapées ici compte tenu de la taille de nos fichiers. Les arbres de décision en revanche choisissent automatiquement les variables pertinentes pour la tâche de discrimination. Par rapport aux autres méthodes, ils ne seront pas handicapées par la profusion des descripteurs bruités. A l'inverse, l'autre méthode non-linéaire que nous utilisons, l'algorithme du **1-plus proche voisin** (1-PPV), est particulièrement sensible à la dimensionalité. Lorsque la taille du fichier d'apprentissage est faible, et que le nombre de descripteurs est très élevé, même pertinents, les estimations locales des probabilités sont de très mauvaise qualité. La situation ne fait qu'empirer lorsque nous avons un grand nombre de descripteurs non-pertinents.

Toutes ces méthodes sont implémentées et sont disponibles dans le logiciel TANAGRA[1] (Rakotomalala, 2005). Dans le cas des SVM, nous appelons directement des bibliothèques externes[2].

[1] http ://eric.univ-lyon2.fr/ ricco

[2] LIBSVM – http ://www.csie.ntu.edu.tw/ cjlin/libsvm/

Methode	**2-grammes**	**3-grammes**	**4-grammes**
SVM Linéaire	0.032	0.038	0.081
PLS (2 axes)	0.041	0.048	0.102
Bayesien Naïf	0.048	0.071	0.248
CART	0.210	0.155	0.141
1-PPV	0.043	0.214	0.269
SVM RBF	0.063	0.130	0.479

TAB. 3 – *Moyenne du taux d'erreur selon les méthodes et le type de représentation.*

4 Résultats des expérimentations

Les résultats de nos expérimentations sont résumés dans le tableau 3. Les valeurs, rappelons-le, sont les moyennes des validations croisées réalisées sur les dix fichiers de données.

Deux résultats semblent s'imposer : les méthodes à fort biais (linéaires) sont les meilleures ; les n-grammes de longueur $n = 2$ suffit largement dans la discrimination des protéines. En effet, nous avons classés les méthodes selon leur performances globales. Nous constatons que les trois premières places sont occupées par des méthodes linéaires. En effectuant une lecture par colonne, nous remarquons que globalement les méthodes sont le plus efficace lorsque $n = 2$. La situation se dégrade à mesure que l'on augmente la taille des n-grammes, de manière plus sensible lorsque la méthode est réputée sensible à la dimensionalité (Bayesien naïf et 1-PPV).

Une lecture plus attentive des résultats relativise ces premières impressions. Nous pouvons nous attendre en effet qu'un SVM avec un noyau RBF (radial basis function) s'en sorte mieux par rapport à un SVM Linéaire dans la mesure où il est capable de retraduire des concepts plus complexes. Or c'est la méthode globalement la moins performante au point d'être au même niveau que le classement aléatoire lorsque $n = 4$.

Un autre point important attire notre attention. Dans la plupart des cas, la qualité de l'apprentissage se dégrade à mesure que l'on augmente la dimensionalité ($n = 3$ et $n = 4$), sauf pour les arbres de décision (CART) qui réalisent automatiquement une sélection des attributs pertinents. Il semble que $n = 4$ propose un espace de description intéressant sur certains fichiers. Certes, CART est globalement en retrait par rapport aux méthodes linéaires. Mais il faut surtout y voir un problème lié à la faiblesse des effectifs, les arbres souffrent très vite de la fragmentation des données.

Nous pouvons nous demander dès lors si les faibles valeurs du taux d'erreur pour $n = 2$ ne reposent pas avant tout sur un apprentissage plus performant dans un espace de faible dimension, et non pas sur un choix judicieux de l'espace de description. De la même manière, nous pouvons nous demander si dans ce contexte, le fait que les méthodes linéaires se placent en meilleure position ne soit pas tout simplement la conséquence de leur capacité à résister à un espace sur-dimensionné et très bruité. En fixant une contrainte forte de représentation, elles sont moins soumises aux perturbations occasionnées par les descripteurs non-pertinents. Nous comprenons mieux pourquoi les méthodes SVM Linéaires et régression PLS se comportent bien. Elles allient à une faible variance naturelle, un système de représentation (linéaire) qui exacerbe cette qualité.

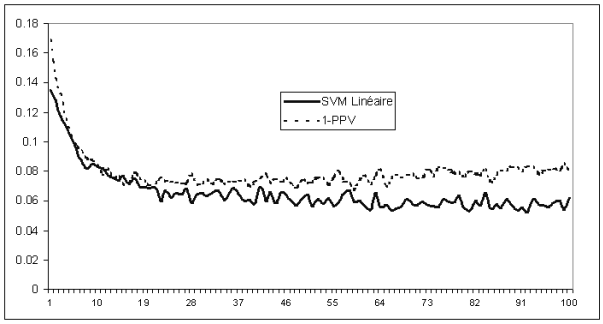

FIG. 4 – *Evolution du taux d'erreur selon le nombre de descripteurs sélectionnés.*

5 Discussion – Quelques expérimentations complémentaires

5.1 Filtrer les descripteurs

Visiblement, la gestion de la dimensionalité est le principal écueil de la discrimination des protéines. Les caractéristiques des méthodes les plus performantes, le comportement des arbres de décision, nous conforte dans cette idée. Il semble naturel de se tourner vers les méthodes de sélection automatique de variables (Guyon et Elisseeff, 2003).

Mis à part les arbres de décision, les autres méthodes n'intègrent pas un processus naturel de sélection des descripteurs. Les approches de type "wrapper", où l'on explore les différentes combinaisons de descripteurs de manière à minimiser l'erreur, ne sont pas envisageables compte tenu du nombre potentiel de descripteurs que peuvent nous proposer les n-grammes. Le temps de calcul est prohibitif, la méthode d'apprentissage étant mise à contribution à chaque évaluation.

Nous devons donc nous tourner vers les méthodes de filtrage. Notamment vers celles qui permettent de classer automatiquement les descripteurs par ordre de pertinence en se basant sur des critères apparentés à des mesures de corrélation. Nous avons pu montrer par ailleurs qu'en combinant cette approche avec l'évaluation de type "wrapper", nous obtenons de bons résultats (Mhamdi et al., 2005). L'approche procède en deux temps. Tout d'abord, nous trions les descripteurs selon leur corrélation avec la variable à prédire. Puis, nous nous appuyons sur cet ordre pour tester des sous-ensembles de taille croissante de descripteurs. La complexité de calcul mettant en jeu l'apprentissage du modèle de prédiction devient linéaire. Nous recherchons bien le sous-ensemble de descripteurs optimisant les performances de l'algorithme d'apprentissage. En contrepartie, nous ne pouvons tester que des solutions imbriquées. Le sous ensemble contenant J descripteurs contient les $J-1$ attributs testés à l'étape précédente. Nous avions évalué cette approche dans un cadre assez restreint au départ (1-PPV, 3-grammes). Nous avons l'occasion ici d'étudier son comportement en opposant des méthodes aux caractéristiques très différentes.

Nous avons classé les descripteurs selon leur pertinence, puis nous avons calculé les performances des sous-ensembles composés du premier attribut, des 2 premiers attributs, etc., jusqu'au 100 premiers attributs. Nous avons opposé la méthode des 1-PPV, très sensible à la dimensionalité, et les SVM linéaires, apparemment peu perturbés par la profusion des des-

Methode	**Tous 3-grammes**	**400 3-grammes**
SVM Linéaire	0.038	0.041
PLS (2 axes)	0.048	0.054
Bayesien Naïf	0.071	0.067
CART	0.155	0.134
1-PPV	0.214	0.144
SVM RBF	0.130	0.049

TAB. 4 – *Moyenne du taux d'erreur en sélectionnant les 400 premiers* 3*-grammes.*

cripteurs. L'évolution du taux d'erreur pour les 100 premières solutions est retracée dans la figure 4. Il s'agit toujours de la moyenne du taux d'erreur sur nos 10 fichiers, nous utilisons les 3-grammes ici, le nombre initial de descripteurs est de l'ordre de 6000.

Pour nos deux méthodes, le taux d'erreur diminue rapidement à mesure que nous ajoutons les premiers descripteurs. Nous obtenons de bons résultats à partir d'une vingtaine de descripteurs. La situation diffère par la suite. Pour les 1-PPV, l'adjonction de nouveaux descripteurs dégrade la qualité de l'apprentissage. Sans que l'on puisse réellement déterminer s'il s'agit là d'une conséquence de l'introduction de descripteurs non-pertinents, ou tout simplement parce que le ratio nombre de descripteurs - taille du fichier d'apprentissage atteint un niveau critique pénalisant la méthode des 1-PPV. Pour les SVM linéaires, le taux d'erreur continue à diminuer faiblement. Pour avoir un point de comparaison par rapport aux résultats de la section précédente (Tableau 3), les 400 premiers 3-grammes présentent un taux d'erreur de 0.041, il est de 0.038 lorsque nous utilisons tous les descripteurs, la différence est infime. Autre comparaison, le taux d'erreur avec tous les 2-grammes (400 descripteurs) est de 0.032. Nous avons voulu systématiser cette étude en sélectionnant, toujours avec les 3-grammes, les 400 premiers descripteurs pour chaque méthode (Tableau 4).

Manifestement, 1-PPV et SVM RBF, qui souffraient de la dimensionalité, améliorent significativement leurs résultats au point que le second se situe parmi les meilleures approches. En ce qui concerne les autres méthodes, notamment celles qui sont très stables (SVM linéaires et Régression PLS), la qualité de l'apprentissage est maintenue. Ce qui est positif puisque nous n'utilisons plus que 400 descripteurs parmi les 6000 3-grammes de départ.

5.2 Gérer la redondance des descripteurs

Une réduction de la dimension assez simpliste modifie dans le bon sens les résultats. Et encore, nous pouvons légitimement penser que la solution du filtrage est très imparfaite, un grand nombre de descripteurs sont très vraisemblablement redondants parmi les solutions proposées. Il faudrait dans la constitution des solutions successives, les sous-ensembles imbriqués d'attributs, tenir compte de la redondance entre les descripteurs en calculant par exemple les corrélations croisées.

Nous avons voulu évaluer cette idée en introduisant une sélection de variables plus agressive avant l'apprentissage. Nous avons choisi la méthode FBCF qui allie à la rapidité une réduction drastique de la dimensionnalité (Yu et Liu, 2003). Pour simplifier, nous dirons que cette méthode intègre les descripteurs dont la corrélation avec la variable à prédire est supérieure à la corrélation avec toutes les autres variables sélectionnées. L'inconvénient ici est que nous ne

Methode	Tous 3-grammes	FBCF 3-grammes
SVM Linéaire	0.038	0.059
PLS (2 axes)	0.048	0.069
Bayesien Naïf	0.071	0.060
CART	0.155	0.140
1-PPV	0.214	0.073
SVM RBF	0.130	0.057

TAB. 5 – *Moyenne du taux d'erreur en sélectionnant les meilleurs* 3*-grammes avec FCBF.*

contrôlons plus le nombre de descripteurs obtenus après filtrage. Sur les 6000 3-grammes de départ, FCBF ne conserve en moyenne qu'une cinquantaine de descripteurs, nous appliquons alors l'algorithme d'apprentissage (Tableau 5).

Concernant les méthodes linéaires (SVM linéaire et PLS), FCBF est manifestement trop restrictif. Il entraîne une dégradation des performances. Il en est autrement en ce qui concerne les méthodes particulièrement sensibles à la dimensionalité. Le 1-PPV et SVM RBF notamment améliorent spectaculairement leurs performances, au point de rattraper les autres algorithmes. Il faut y voir bien entendu l'effet de la sélection des variables pertinentes. Mais il faut y voir également un effet mécanique dû à l'amélioration de la densité des points dans le nouvel espace de représentation. Ce qui laisse à penser qu'une réduction de la dimensionalité, même trop agressive, les résultats de SVM linéaire et PLS en attestent, peut néanmoins être bénéfique aux méthodes sensibles au ratio nombre de descripteurs - nombre d'observations.

6 Conclusion

Dans cet article, nous avons pris le parti de placer la discrimination de protéines dans un cadre inspiré de la catégorisation de textes. Nous avons mis de côté les autres approches existantes sur le classement de protéines comme celles qui recherchent les homologies entre les séquences en se basant sur la similarité (Jaakkola et al., 2000). Il apparaît que l'utilisation des n-grammes permet d'extraire des descripteurs performants. Les méthodes linéaires s'imposent par la suite pour fournir des taux d'erreur de classement faibles. Une étude détaillée des expérimentations semble indiquer que ce résultat repose avant tout sur la capacité de ces méthodes à résister aux espaces de représentation sur-dimensionnés, avec de nombreux descripteurs non-pertinents.

Ces résultats nous suggèrent la voie à suivre pour l'amélioration de notre processus de discrimination des protéines : développer des méthodes efficaces de sélection de variables pour ne pas perturber l'apprentissage ; lever la restriction sur la taille n des n-grammes et donc proposer une procédure pour extraire des descripteurs de longueur variable, en nous appuyant sur la recherche de co-occurrences significatives des n-grammes par exemple.

Références

Chapelle, O., J. Weston, et B. Schölkopf (2003). Cluster kernels for semi-supervised learning. Volume 15 of *Neural Information Processing Systems*.

Dietterich, T. (1999). Approximate statistical tests for comparing supervised classification learning. *Neural Computation 10*(7), 1895–1924.

Dumais, S., J. Platt, D. Heckerman, et M. Sahami (1998). Inductive learning algorithms and representations for text categorization. In *CIKM '98 : Seventh international conference on Information and knowledge management*, pp. 148–155. ACM Press.

Fayyad, U., G. Piatetsky-Shapiro, et P. Smyth (1996). From data mining to knowledge discovery in databases. *Ai Magazine 17*, 37–54.

Guyon, I. et A. Elisseeff (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res. 3*, 1157–1182.

Hastie, T., R. Tibshirani, et J. Friedman (2001). *The elements of statistical learning*. Springer Series in Statistics. New York : Springer-Verlag.

Jaakkola, T., M. Diekhans, et D. Haussler (2000). A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology* (7), 95–114.

Jalam, R., J. Clech, et R. Rakotomalala (2004). Cadre pour la catégorisation de textes multilingues. In C. Fairon, G. Prunelle, et A. Dister (Eds.), *7èmes Journées internationales d'Analyse statistique des Données Textuelles*, Louvain-la-Neuve, Belgique, pp. 650–660.

Mhamdi, F., R. Rakotomalala, et M. Elloumi (2005). Feature ranking for protein classification. In *4th International Conference on Computer Recognition Systems*, pp. 611–617. Springer.

Murzin, G., E. Brenner, T. Hubbard, et C. Chothia (1995). Scop : a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Bio. 247*, 536–540.

Rakotomalala, R. (2005). Tanagra : une plate-forme d'expérimentation pour la fouille de données. *Revue MODULAD* (32), 70–85.

Sebastiani, F. (2005). Text categorization. In A. Zanasi (Ed.), *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pp. 109–129. WIT Press.

Yu, L. et H. Liu (2003). Efficiently handling feature redundancy in high-dimensional data. In *KDD '03 : Proceedings of the ninth ACM SIGKDD*, pp. 685–690. ACM Press.

Summary

We evaluate various supervised methods in a protein discrimination framework. The descriptors being automatically generated, we used n-grams, a great number of descriptors are irrelevant. It appears clearly that linear methods such as the Linear SVM or PLS Regression are most powerful. A detailed study shows that this success rests primarily on the robustness vis-a-vis the dimensionnality which becomes, in fact, the most significant criterion when we treat fields where numerous descriptors are automatically generated. In this context, a feature selection, even very elementary, significantly modifies the behavior of the training algorithms.