Prédiction structurée multitâche itérative de propriétés structurelles de protéines

Francis Maes, Julien Becker, Louis Wehenkel

Université de Liège - Département d'Electricité, Electronique et Informatique et GIGA-R
Institut Montefiore, B28, B-4000, Liège - Belgique
{francis.maes, j.becker, l.wehenkel}@ulg.ac.be

Résumé : Le développement d'outils informatiques pour prédire de l'information structurelle de protéines à partir de la séquence en acides aminés constitue un des défis majeurs de la bioinformatique. Les problèmes tels que la prédiction de la structure secondaire, de l'accessibilité au solvant, ou encore la prédiction des régions désordonnées, peuvent être exprimés comme des problèmes de prédiction avec sorties structurées et sont traditionnellement résolus individuellement par des méthodes d'apprentissage automatique existantes. Etant donné que ces problèmes sont fortement liés les uns aux autres, nous proposons de les traiter ensemble par une approche d'apprentissage multitâche. A cette fin, nous introduisons un nouveau cadre d'apprentissage générique pour *la prédiction structurée multitâche*. Nous appliquons cette stratégie pour résoudre un ensemble de cinq tâches de prédiction de propriétés structurelles des protéines. Nos résultats expérimentaux sur deux jeux de données montrent que la stratégie proposée est significativement meilleure que les approches traitant individuellement les tâches.
Mots-clés : Bioinformatique, Multitâche, Apprentissage structuré

1. Introduction

La prédiction *ab initio* de la structure tertiaire des protéines (*i.e.* calculer les positions 3D de tous les atomes à partir de la séquence d'acides aminés) est un problème très important, extrêmement difficile, et toujours non résolu dans la recherche en biologie. Face à la difficulté de ce problème, différents sous-problèmes plus simples ont été introduits et étudiés en bioinformatique. On peut formaliser ces problèmes dans le cadre de l'apprentissage avec sorties structurées. Les problèmes les plus courants consistent à prédire des séquences d'étiquettes. Parmi ces problèmes, citons : (i) *la prédiction de la structure secondaire*, où les étiquettes correspondent à des structures 3D locales telles que les hélices alpha, les feuillets beta et le reste ; (ii) *la prédiction*

de l'accessibilité au solvant, où les étiquettes sont des niveaux d'exposition des résidus de la protéine au solvant ; et (iii) *la prédiction de régions désordonnées*, qui vise à identifier les acides aminés qui ne possèdent pas de structure tertiaire fixe dans la protéine. Au delà de ces problèmes d'étiquetage de séquences, il existe également des problèmes faisant intervenir des sorties possédant une structure plus complexe, tels que la prédiction des matrices de contacts où le but est de prédire un ensemble de liens dans un graphe.

Dans la littérature bioinformatique, les problèmes de prédiction de propriétés structurelles de protéines sont traditionnellement traités de manière indépendante : par exemple, on modélise et utilise un prédicteur pour l'inférence de la structure secondaire et, séparément, on modélise et utilise un autre prédicteur pour l'inférence de l'accessibilité au solvant. Depuis quelques années maintenant, le domaine de l'apprentissage automatique a mis au point des approches dites *multitâches*, qui visent à traiter conjointement un ensemble de tâches liées (à la fois lors de l'apprentissage et lors de l'inférence) avec l'espoir d'avoir une amélioration sur chacune des tâches traitées par rapport aux prédicteurs conçus dans une vision "tâche-unique". Etant donné les liens forts qui existent entre les différentes tâches de prédiction liées aux protéines, nous proposons de modéliser cet ensemble de tâches dans un cadre multitâche.

Dans la littérature consacrée à l'apprentissage multitâche, la plupart des travaux concernent la *classification* multitâche : chaque tâche d'apprentissage est un problème de classification. Ce cadre n'est pas satisfaisant pour la résolution des tâches de prédiction structurées relatives aux propriétés des protéines. Afin de pouvoir traiter des problèmes avec sorties structurées dans un cadre multitâche, nous proposons un nouveau cadre général reposant sur l'assemblage d'un ensemble de modèles de prédiction structurée simple-tâche. L'idée centrale de notre approche est d'exploiter les corrélations entres les sorties des différentes tâches, *via* un processus de ré-estimation itérative des prédictions.

La Section 2 présente les travaux similaires à celui présenté ici. Notre nouveau cadre d'apprentissage multitâche structuré est proposé en Section 3. Nous décrivons ensuite notre protocole expérimental et les résultats obtenus sur un ensemble de cinq tâches en Section 4. Nous concluons et donnons les perspectives de recherche future en Section 5.

2. Travaux similaires

Nous présentons à présent les travaux similaires au nôtre, tout d'abord dans le domaine de l'apprentissage multitâche puis dans le domaine de la bioinformatique.

2.1. Les approches multitâches

La majorité des approches d'apprentissage multitâche existantes se basent sur l'utilisation d'une représentation interne partagée entre toutes les tâches considérées. Une telle représentation partagée permet de mieux extraire l'information essentielle des données d'entrée en exploitant les caractéristiques communes entres les différentes tâches (Maurer, 2006). Bien que ces méthodes sont élégantes sur le plan conceptuel, elles sont souvent limitées à des problèmes d'apprentissage simples tels que la classification et la régression. Les travaux exploitant une représentation partagée entre plusieurs tâches de prédiction structurée sont extrêmement rares (voir Collobert & Weston (2008) pour un exemple dans le domaine du traitement du langage naturel).

Dans cet article, nous reposons sur une approche radicalement différente pour aborder les problèmes d'apprentissage multitâche : l'approche "black-box" qui consiste à combiner un ensemble de modèles prédictifs simple-tâche pour former un modèle multitâche. L'intérêt principal d'une telle approche est de permettre de transformer n'importe quel ensemble de modèles élémentaires par tâche en un modèle multitâche. Ainsi, les modèles élémentaires peuvent être de toute sorte, des classificateurs linéaires aux approches modernes de prédiction structurée, tel que celles proposées par Lafferty *et al.* (2001) ou Tsochantaridis *et al.* (2004).

La méthode développée par Heitz *et al.* (2008) est, à notre connaissance, celle qui soit la plus proche de nos travaux. Les auteurs proposent une méthode de classification multitâche basée sur un ensemble de couches, où chaque couche est composée d'un classificateur par tâche. Les classificateurs d'une même couche sont appris indépendamment en utilisant comme entrée, l'entrée globale et les prédictions des classificateurs de la couche précédente. Cette méthode a été formulée et appliquée au problème de la compréhension de scènes naturelles, comprenant les tâches de catégorisation des scènes, de détection d'objets, de segmentation d'images et de reconstruction 3D. Ici, nous généralisons ces idées en les formulant dans le cadre général de la

prédiction structurée multitâche et proposons une application originale qu'est la prédiction de propriétés structurelles de protéines.

2.2. La prédiction de propriétés structurelles de protéines

Jusqu'à récemment dans la littérature bioinformatique, les problèmes de prédiction de propriétés structurelles des protéines étaient traités de manière indépendante : à une tâche, un prédicteur (Cheng *et al.*, 2005). Afin d'exploiter les liens pouvant exister entre les différentes tâches de prédiction relatives aux propriétés des protéines, plusieurs systèmes de *pipeline* ont été proposés (Adamczak *et al.*, 2005a; Cheng *et al.*, 2005). Ces systèmes reposent sur le chaînage de plusieurs modules de prédiction élémentaires.

L'idée de l'apprentissage multitâche à proprement parler n'est apparu que récemment dans la littérature en bioinformatique sans pour autant utiliser les formalismes développés en apprentissage automatique. Ainsi, une petite poignée de travaux proposent de faire la prédiction simultanée de la structure secondaire et de l'accessibilité au solvant (Dor & Zhou, 2007; Pollastri *et al.*, 2007).

A notre connaissance, les problèmes de prédiction ayant attrait aux protéines n'ont à ce jour jamais été formulés explicitement dans un cadre multitâche. De plus, aucun système combinant plus de deux tâches différentes n'a été proposé dans ce domaine. Notre formalisation et le fait d'envisager un grand nombre de tâches différentes de manière simultanée sont donc les contributions principales de notre travail.

3. Prédiction structurée multitâche itérative

Nous décrivons à présent notre approche pour l'apprentissage multitâche avec sorties structurées, appelée *prédiction structurée multitâche itérative* (PSMI).

3.1. Notations

On considère que le problème de prédiction structurée multitâche consiste à apprendre une fonction des entrées $x \in \mathcal{X}$ vers des sorties $y_1, \ldots, y_T \in \mathcal{Y}_1, \ldots, \mathcal{Y}_T$ pour chaque tâche $t \in [1, T]$. Pour apprendre cette fonction, nous avons accès à un ensemble d'apprentissage composé d'entrées associées à une ou plusieurs sorties structurées. L'ensemble d'apprentissage

est noté $D = \{(x^{(i)}, y_1^{(i)}, \ldots, y_T^{(i)})\}_{i\in[1,N]}$, où N est le nombre d'exemples d'apprentissage. L'espace d'états $\mathcal{S}$ du problème multitâche est défini par $\mathcal{S} = (\mathcal{Y}_1 \cup \{\epsilon_1\}) \times \cdots \times (\mathcal{Y}_T \cup \{\epsilon_T\})$, où ϵ_t est une valeur de sortie spéciale utilisée pour représenter une sortie y_t qui est absente.

3.2. Principe

Les approches de type *pipeline* consiste à chainer un ensemble de prédicteurs de manière séquentielle, chaque prédicteur reposant sur les sorties des prédicteurs précédents. Bien que cette approche s'avère souvent effective, elle soufre d'un défaut majeur : si un prédicteur fait une erreur, cette erreur est définitive : elle ne peut pas être réparée dans la suite du processus. De plus, du fait de la nature séquentielle du processus, seuls les prédictions de début profitent aux prédictions de fin ; l'inverse n'étant pas vrai.

L'idée centrale de PSMI est d'effectuer plusieurs passes de prédiction pour donner l'opportunité au modèle de réparer d'éventuelles erreurs de prédiction, ainsi que pour bénéficier des prédictions des tâches de fin pour améliorer les tâches de début. Le fait que chaque prédicteur ait comme entrée les prédictions de toutes les tâches, et puisse exploiter de manière avancée les corrélations qui existent entre les sorties de ces tâches, motive l'appellation multitâche de la méthode.

Le processus commence par un état ne comprenant que des prédictions manquantes, *i.e.* $s = (\epsilon_1, \ldots, \epsilon_T)$. A la première étape, la sortie y_1 de la première tâche est prédite avec le premier modèle élémentaire n'utilisant effectivement en entrée que l'entrée globale x. Un second modèle est ensuite utilisé pour prédire y_2 sachant x et la prédiction y_1. Un troisième modèle prédit y_3 sachant x et les prédictions de y_1 et y_2, et ainsi de suite. Une fois que toute les sorties ont été estimées une fois, on dit avoir effectué une *passe*. Un modèle à une passe est un *pipeline* traditionnel. L'apport de notre approche est de pouvoir utiliser un nombre de passes $P > 1$, P étant un hyper-paramètre de la méthode proposée.

Le modèle complet se compose de PT modèles élémentaires notés $(M_{1,1}, \ldots, M_{1,T}, M_{2,1}, \ldots, M_{P,T})$, où $M_{p,t}$ est le modèle de la p-ième passe et t-ième tâche. Par exemple, la Figure 1 illustre un modèle PSMI pour quatre tâches et cinq passes.

Des modèles distincts sont appris à chaque passe ; ceci est motivé par le fait que – comme les sorties sont re-estimées à chaque passe – la distribution d'entrée-sortie des problèmes sous-jacents évolue d'une passe à l'autre.

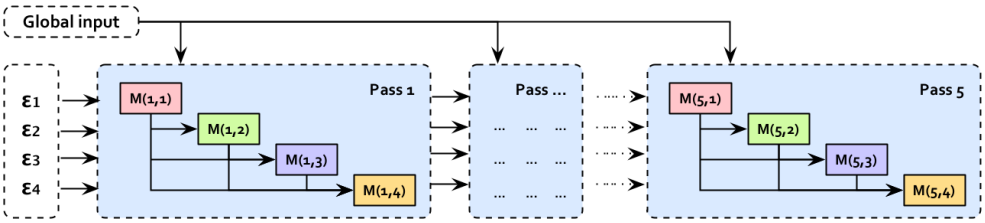

FIGURE 1: Illustration d'une chaîne de modèles à quatre tâches et cinq passes. Chaque étape est un modèle à sortie structurée distinct où les entrées sont l'entrée globale et l'état courant de toute les tâches.

Algorithm 1 Algorithme d'inférence PSMI

Given an input $x \in \mathcal{X}$ and a model chain $(M_{1,1}, \ldots, M_{1,T}, M_{2,1}, \ldots, M_{P,T})$

1: $s \leftarrow (\epsilon_1, \ldots, \epsilon_T)$ ▷ initial state
2: **for** $p = 1$ to P **do** ▷ for each pass
3: **for** $t = 1$ to T **do** ▷ for each task
4: $\hat{y}_t \leftarrow M_{p,t}(x, s)$ ▷ estimate target t
5: $s \leftarrow (s_1, \ldots, s_{t-1}, \hat{y}_t, s_{t+1}, \ldots, s_T)$ ▷ update targets state
6: **end for**
7: **end for**
8: **return** s ▷ return current state of all targets

Par exemple, la première passe d'estimation de la sortie de la première tâche (sachant l'entrée globale uniquement) n'est pas le même problème que sa seconde estimation (sachant l'entrée globale et les T prédictions issues de la première passe).

3.3. Apprentissage et inférence

Les algorithmes 1 et 2 décrivent respectivement l'inférence et l'apprentissage de PSMI. Sachant la chaîne de modèles, l'inférence consiste simplement à chaîner itérativement les inférences élémentaires en maintenant $s \in \mathcal{S}$, l'état courant de toutes les sorties, *i.e.* $s = (\hat{y}_1, \ldots, \hat{y}_T)$. Les sorties sont, dans un premier temps, initialisées avec les valeurs spéciales $(\epsilon_1, \ldots, \epsilon_T)$ (ligne 1). Chaque étape consiste à prédire la sortie structurée $\hat{y}_t$ puis à mettre à jour l'état courant (lignes 4–5). Les prédictions finales sont données par s à la fin de l'inférence (ligne 8).

L'apprentissage consiste à créer la chaîne de modèles sachant l'ensemble

Algorithm 2 Algorithme d'apprentissage PSMI

Given a training set $D = \{(x^{(i)}, y_1^{(i)}, \ldots, y_T^{(i)})\}_{i \in [1,N]}$,
Given a structured learning algorithm $\mathcal{A}$,
Given a number of passes P,

1: $S \leftarrow \{s^{(i)} = (\epsilon_1, \ldots, \epsilon_T)\}_{i \in [1,N]}$ ▷ initial state
2: **for** $p = 1$ to P **do** ▷ for each pass
3: **for** $t = 1$ to T **do** ▷ for each task
4: $D_t \leftarrow \{((x^{(i)}, s^{(i)}), y_t^{(i)})\}_{i \in [1,N]}$ ▷ create training set
5: $M_{p,t} \leftarrow \mathcal{A}(D_t)$ ▷ train a model for task t
6: $S \leftarrow$ update S given D_t and $M_{p,t}$ ▷ update current state
7: **end for**
8: **end for**
9: **return** $(M_{1,1}, \ldots, M_{1,T}, M_{2,1}, \ldots, M_{P,T})$ ▷ return model chain

d'apprentissage. Tout comme l'inférence, l'apprentissage est effectué de manière itérative et repose sur un ensemble d'états courants $\{s^{(1)}, \ldots, s^{(N)}\}$. Ces états courants sont d'abords initialisés avec les étiquettes ϵ désignant des prédictions manquantes (ligne 1). Chaque étape de l'apprentissage vise à ajouter un élément à la chaîne de modèles. Ceci implique la création d'un ensemble d'apprentissage (ligne 4), l'apprentissage d'un modèle élémentaire de prédiction structurée (ligne 5) et la mise à jour de l'état courant de chaque exemple (ligne 6). Les entrées d'apprentissage des modèles élémentaires contiennent à la fois l'entrée globale x et l'état courant $s = (\hat{y}_1, \ldots, \hat{y}_T)$.

3.4. Eviter le sur-apprentissage

Etant donné que la chaîne de modèles peut potentiellement être longue (jusqu'à 40 modèles dans nos expériences), des précautions particulières doivent être prises pour éviter le surapprentissage. En effet, les exemples d'apprentissage peuvent rapidement être parfaitement appris par les premiers modèles de la chaîne, biaisant alors dangereusement les données d'apprentissage pour tous les modèles restants dans la chaîne. Dans nos expériences, nous utilisons de très grands ensembles d'apprentissage et des classificateurs linéaires simples et nous n'avons pas rencontré ce problème. Cependant, de tels problèmes de surapprentissage peuvent être évités par au moins deux techniques : soit en générant des prédictions intermédiaires par l'utilisation d'une validation croisée comme présenté dans l'approche d'apprentissage

en pile (Cohen & Carvalho, 2005), soit par l'introduction de bruit dans les prédictions intermédiaires comme proposé par Maes *et al.* (2009).

4. Expériences : Annotations structurées des protéines

Nous décrivons à présent nos expériences concernant un ensemble de cinq tâches de prédiction de propriétés structurelles de protéines.

4.1. Jeux de données

Nous avons utilisé deux jeux de données construits à partir de la base de données Protein Data Bank (PDB), laquelle contient des structures de protéines obtenues expérimentalement. Dans le première ensemble, PDB30, nous avons sélectionné aléatoirement 500 protéines à partir d'une version filtrée de PDB où l'alignement de chaque paire de séquences n'est identique qu'à hauteur de maximum 30%. Ce filtre permet d'assurer une différence significative entre les protéines des ensembles d'apprentissage et de test. Nous avons également utilisé le jeu de données standard PSIPRED, constitué par Jones (1999). Ces données se composent de 1329 protéines d'apprentissage et de 187 protéines de test. Chacune d'entre-elles a un repliement différent, *i.e.* leur forme est significativement différente des autres. Ce jeu de données nous permet de nous comparer avec la méthode PSIPRED, qui est considérée comme l'état de l'art dans le domaine de la prédiction *ab initio* de la structure secondaire Zhang *et al.* (2011).

Sur chacun des jeux de données, nous appliquons deux prétraitements : l'un pour déterminer les sorties correctes pour les différentes tâches (décrit au point suivant), l'autre pour enrichir les entrées. Cette dernière opération est courante lors de la manipulation de séquences d'acides aminés et consiste à calculer un "profil génétique" de la séquence d'acides aminés. Ce profil est créé en effectuant un alignement multiple entre la séquence d'entrée et une grande base de données d'autres séquences. Dans notre cas, nous utilisons des matrices de scores par position (*Position-Specific Scoring Matrices*, PSSMs) (Jones, 1999) où un score positif (resp. négatif) pour un acide aminé donné à une position donnée indique une fréquence de substitution plus élevée (resp. faible). L'alignement multiple est réalisé à l'aide de trois itérations de l'outil PSI-BLAST (Altschul *et al.*, 1997) en utilisant la base de données non-redondante du NCBI (Pruitt *et al.*, 2006). Une fois la PSSM générée, les valeurs prises par celle-ci (typiquement dans l'intervalle $[-7; 7]$) sont norma-

lisées à l'échelle $[0; 1]$. Plusieurs méthodes de normalisation sont disponibles dans la littérature, allant d'une simple réduction linéaire à l'utilisation d'une fonction logistique. Dans notre cas, nous avons opté pour la fonction (Kim & Park, 2003) :

$$f(x) = \begin{cases} 0.0 & si\ x \leq -5 \\ 0.5 + 0.1x & si\ -5 < x < 5 \\ 1.0 & si\ x \geq 5 \end{cases}$$

4.2. Tâches

Nous considérons l'ensemble suivant de 5 tâches : la prédiction de la structure secondaire (à 3 et à 8 étiquettes), la prédiction de l'accessibilité au solvant (2 étiquettes), la prédiction de régions désordonnées (2 étiquettes) et la prédiction d'étiquettes issues d'un alphabet structurel (27 étiquettes). La supervision de la structure secondaire, ainsi que de l'accessibilité au solvant, est déterminée par le programme DSSP (Kabsch & Sander, 1983) alors que la supervision de régions désordonnées et de l'alphabet structurel est générée à partir de la structure 3D par nos soins. La Figure 2 schématise le protocole de génération des entrées-sorties à partir du fichier PDB d'une protéine.

L'utilisation de deux versions de la structure secondaire peut sembler redondante mais permet d'obtenir deux niveaux de granularité, l'un est plus simple, en terme de prédiction (75-80% de précision), mais moins précis, en terme de structure locale, alors que l'autre est plus difficile (55-60%) mais plus précis. Nous avons pu ainsi constater, lors de nos expériences, une amélioration des prédictions quand les deux tâches sont présentes. L'alphabet structurel, proposé par Camproux *et al.* (2004), est une discrétisation de la conformation du squelette de la protéine en une série de fragments de quatre résidus, en chevauchement. Ce problème de prédiction n'est pas commun dans la littérature. Nous l'utilisons, comme un troisième niveau de granularité de la structure locale tridimensionnelle et son utilisation semble également améliorer la prédiction des autres tâches.

Etant donné que la définition d'une région désordonnée n'est pas définie de façon unique dans la communauté, nous utiliserons la définition établie par CASP (Noivirt-Brik *et al.*, 2009) lors de ses compétitions, *i.e.* les segments de longueur d'au moins trois résidus ne possédant pas de coordonnées atomique dans la structure cristalline de la protéine sont étiquetés comme

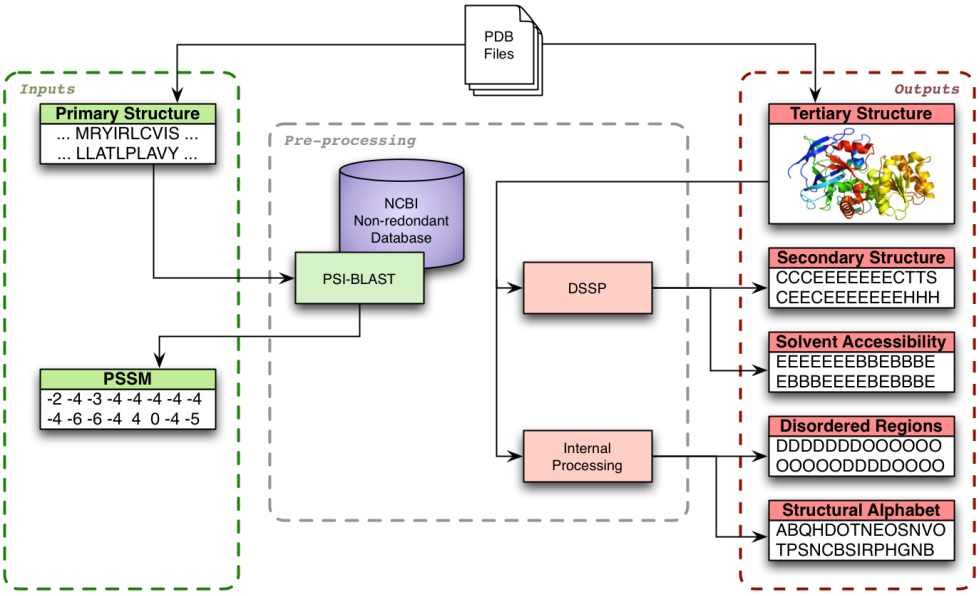

FIGURE 2: Protocole utilisé pour générer les entrées (la structure primaire et la PSSM) et les supervisions (la structure secondaire, l'accessibilité au solvant, les régions désordonnées et l'alphabet structurel) à partir de fichiers PDB.

"désordonnés", alors que tous les autres résidus sont étiquetés comme "ordonnés".

Nous avons utilisé le seuil courant de 20% pour définir les deux états "caché" et "exposé" de l'accessibilité au solvant. Notez que d'autres seuils sont également utilisés dans la littérature tel que 5%, 16%, 25% ou encore 30%.

La mesure de "score" utilisée par défaut est la précision par étiquette, *i.e.* le pourcentage d'acides aminés dont les étiquettes sont correctement prédites sur l'ensemble de test. Vu que l'étiquetage des régions désordonnées est un problème fortement déséquilibré, cette mesure n'est pas très appropriée pour cette tâche. À la place, nous utilisons une mesure d'évaluation classique pour ce problème de prédiction : le coefficient de corrélation de Matthews (MCC) (Adamczak *et al.*, 2005b).

4.3. Modèle et attributs

Nous avons utilisé une approche de classification simple où chaque étiquette est prédite indépendamment des autres, sur base d'attributs décrivant des propriétés locales et globales de la protéine. Plus précisément, le classificateur de base est une machine à vecteurs de support linéaire entrainée par une descende de gradient stochastique. Cette méthode simple à l'avantage d'avoir un temps d'apprentissage court et de bonnes propriétés de passage à l'échelle. Les meta-paramètres d'apprentissages ont été réglés sur l'ensemble d'apprentissage en ajustant, dans un premier temps, les meta-paramètres relatifs à la première passe et, dans un second temps, en fixant ces-derniers et ajustant ceux de la second passe. Etant donné que les nombres d'exemples et d'entrées ne varie plus au-delà de la seconde passe et que les valeurs optimales des méta-paramètres sont fortement liés à ceux-ci, nous avons utilisé les valeurs des méta-paramètres de la seconde passe pour toutes les passes ultérieures.

L'ordre dans lequel les tâches sont apprises est celui-ci : i) la structure secondaire à 3 états, ii) la structure secondaire à 8 états, iii) l'accessibilité au solvant, iv) les régions désordonnées et v) l'alphabet structurel. Néanmoins, cet ordonnancement n'a pas de grosse influence. Tout au plus, les scores subissent un déplacement d'une passe. Bien que non développé dans cet article, nous avons également mis en place une version parallèle de notre système d'apprentissage, dans laquelle tous les modèles d'une même passe sont appris en parallèle. Cette approche parallèle permet de s'affranchir de la dépendance par rapport à l'ordre des tâches.

Les attributs utilisés sont similaires à ceux proposés par Jones (1999) et Zhang *et al.* (2008). Un premier sous-ensemble d'attributs décrit de façon globale la distribution en acides aminés au sein de la protéine et la longueur de sa séquence. Un second ensemble d'attributs décrit l'information locale relative aux positions voisines dans la séquence d'acides aminés ; ces attributs locaux sont constitués à partir des éléments d'entrée (acides-aminés, PSSM) et des étiquettes prédites contenus dans une fenêtre glissante de taille 15, ce qui est illustré par la figure 3. Le choix de cette taille de fenêtre est un choix classique dans la littérature.

Pour construire le vecteur de caractéristiques utilisé par le SVM, nous appliquons une étape de post-traitement sur les attributs. Cette étape consiste à, dans le cas de valeur continues, discrétiser ces valeurs en cinq intervalles et, dans le cas de valeurs catégoriques, représenter l'ensemble des classes pos-

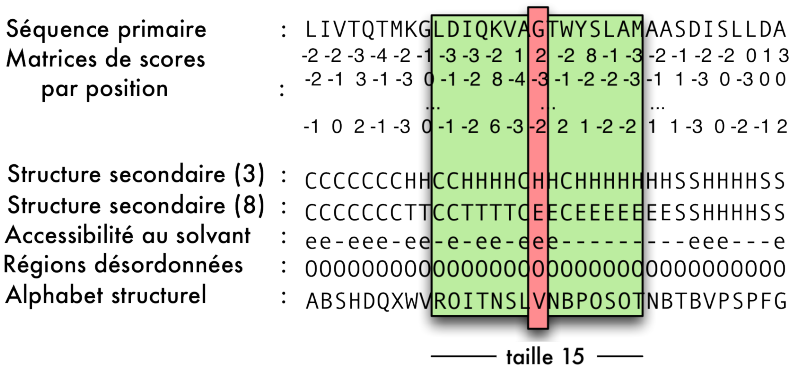

FIGURE 3: Illustration des attributs locaux utilisés à un position donnée dans la séquence. En rouge, la position que l'on souhaite décrire. En vert, la fenêtre de taille 15 centrée sur la position.

sibles où la caractéristique qui représente la valeur de l'attribut vaut 1 et les autres 0. Ce qui porte à $\sim$3800 le nombre d'attributs utilisé par le SVM à partir de la seconde passe.

4.4. Résultats

Nous avons entrainé un modèle de prédiction structurée multitâche itérative jusqu'à $P = 8$ passes, ce qui donne lieu à une chaîne de modèles de longueur $PT = 40$. Pour observer l'effet de la ré-estimation itérative des sorties, nous avons évalué chaque tâche en "coupant" la chaîne de modèles après un certain nombre de passes $P_{max} \in [1, 8]$. La Figure 4 donne le score de test, sur les données PSIPRED, pour chaque tâche en fonction du nombre de passes. Il est clair que toutes les tâches bénéficient de la ré-estimation des sorties, et plus particulièrement pendant les premières passes. Vers les dernières ré-estimations, certains scores se dégradent un peu. Cependant, nous n'observons pas de phénomène de surapprentissage très marqué (voir la discussion à la Section 3) dans ces expériences. Il est important de noter que dans tous les cas, la ré-estimation des sorties, après plusieurs passes, est significativement plus précise que l'estimation initiale (une seule passe).

Pour mesurer l'amélioration apportée par l'approche multitâche, nous avons aussi effectué une expérience de "référence" par tâche en utilisant un étiquetage de séquences itératif dans un contexte où seule la tâche concernée est présente. Dès lors, ces références se basent également sur une ré-

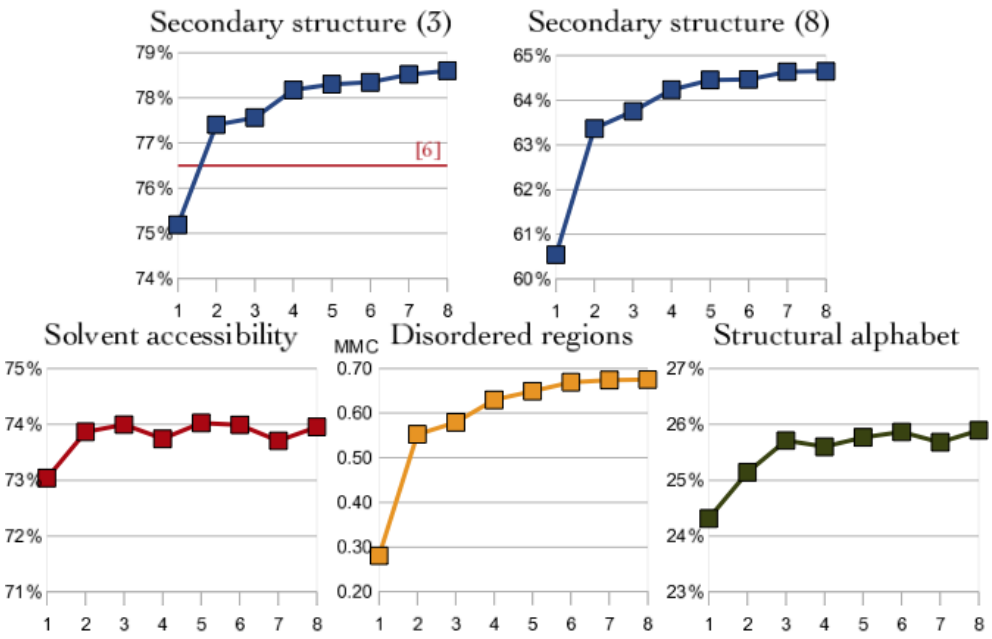

FIGURE 4: Evolution du score de test après un nombre croissant de passes sur PSIPRED. Le résultat de Jones (1999), faisant office d'état de l'art, est disponible pour la tâche "Secondary Structure (3)".

		PDB30		PSIPRED	
Tâche	Etiquettes	Unique	Multitâche	Unique	Multitâche
Secondary structure	3	75.45 %	76.35 %	76.29 %	78.60 %
Secondary structure	8	60.38 %	62.69 %	62.25 %	64.64 %
Solvent accessibility	2	71.56 %	73.52 %	73.51 %	73.95 %
Disordered regions	2	0.4212	0.4983	0.5611	0.6749
Structural alphabet	27	16.81 %	18.14 %	24.88 %	25.89 %

TABLE 1: Comparatif des résultats obtenus, sur PDB30 et PSIPRED (8 passes), entre l'approche à tâches unique et l'approche multitâche.

estimation itérative des sorties mais n'utilisent par contre plus les prédictions provenant des autres tâches. La comparaison entre notre modèle multitâche et ces références à "tâche unique" est donnée à la Table 1. Nous pouvons observer que sur les deux jeux de données, les résultats de notre approche multitâche sont systématiquement meilleurs que l'approche à tâche unique, *e.g.* :

$+2.31\%$ pour la prédiction de la structure secondaire et $+0.114$ MCC pour la prédiction de régions désordonnées sur l'ensemble de test de PSIPRED. Nous pouvons aussi observer que seul l'approche multitâche dépasse les résultats de l'état de l'art sur PSIPRED avec une amélioration de $+2.1\%$.

5. Conclusion

Nous avons présenté un cadre d'apprentissage, générique et conceptuellement simple, pour la *prédiction structurée multitâche*. Il s'agit d'une nouvelle approche itérative pour l'apprentissage automatique en mode multitâche mis en oeuvre pour résoudre plusieurs problèmes liés d'étiquetage de séquences. Cette approche a l'avantage de pouvoir utiliser n'importe quel algorithme de prédiction structurée. Nous avons réalisé des expériences sur un ensemble de cinq tâches d'étiquetages de séquences de protéines en utilisant comme modèle de base une machine à vecteur de support entrainée par une descende de gradient stochastique. Avec cette configuration, nous avons montré que notre approche est systématiquement meilleure que les approches n'utilisant qu'une seule tâche et ce pour toutes les tâches et les deux jeux de données (PSIPRED et PDB30). Nous avons également montré que notre approche surpasse de manière significative (+2.1% d'amélioration) les résultats de l'état de l'art de la prédiction de la structure secondaire.

Etant donné que l'approche multitâche itérative n'est pas restreinte à la prédiction de séquences d'étiquettes, nous l'appliquerons dans le future à d'autres problèmes de prédiction de protéines, tels que la prédiction des fonctions de protéine, la prédiction de contact résidu-résidu, la prédictions de l'alignement des feuillets beta, la prédiction des interactions protéine-protéine, et la prédiction de la structure tertiaire. Nous pensons également que le cadre d'apprentissage proposé dans ce papier peut-être appliqué à bien d'autres domaines (*e.g.* : l'analyse d'images, le traitement de textes, la surveillance et le contrôle de réseaux, la robotique), où les données sont naturellement disponibles sous la forme d'un ensemble de représentations complémentaires et où les problèmes de prédiction concernent généralement plusieurs tâches liées de manière temporelle ou spatiale.

6. Remerciements

Ce papier présente les résultats de recherche du Belgian Network BIOMAGNET (Bioinformatics and Modelling : from Genomes to Networks), financé

par le programme Interuniversity Attraction Poles, initié par l'Etat belge, et du réseau d'excellence EU FP7 PASCAL2. Julien Becker est bénéficiaire d'une bourse de recherche F.R.I.A. du Fonds National de la Recherche Scientifique belge (FNRS).

Références

ADAMCZAK R., POROLLO A. & MELLER J. (2005a). Combining prediction of secondary structure and solvent accessibility in proteins. *Proteins : Structure, Function, and Bioinformatics*, **59**(3), 467–475.

ADAMCZAK R., POROLLO A. & MELLER J. (2005b). Combining prediction of secondary structure and solvent accessibility in proteins. *Proteins*.

ALTSCHUL S. F., MADDEN T. L., SCHAFFER A. A., ZHANG J., ZHANG Z., MILLER W. & LIPMAN D. J. (1997). Gapped BLAST and PSI-BLAST : a new generation of protein database search programs. *Nucleic Acids Research*, **25**(17), 3389–3402.

CAMPROUX A., GAUTIER R. & TUFFERY P. (2004). A hidden markov model derived structural alphabet for proteins. *Journal of molecular biology*.

CHENG J., RANDALL A. Z., SWEREDOSKI M. J. & BALDI P. (2005). SCRATCH : a protein structure and structural feature prediction server. *Nucleic Acids Research*, **33**(suppl 2), W72–W76.

COHEN W. & CARVALHO V. R. (2005). Stacked sequential learning. In *International Joint Conferences on Artificial Intelligence*.

COLLOBERT R. & WESTON J. (2008). A unified architecture for natural language processing : deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, p. 160–167, New York, NY, USA : ACM.

DOR O. & ZHOU Y. (2007). Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. *Proteins : Structure, Function, and Bioinformatics*, **66**(4), 838–845.

HEITZ G., GOULD S., SAXENA A. & KOLLER D. (2008). Cascaded classification models : Combining models for holistic scene understanding. In *Neural Information Processing Systems*.

JONES D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, **292**(2), 195 – 202.

KABSCH W. & SANDER C. (1983). Dictionary of protein secondary structure : pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**(12), 2577–2637.

KIM H. & PARK H. (2003). Protein secondary structure prediction based on an improved support vector machines approach. *Protein Engineering*, **16**(8), 553–560.

LAFFERTY J., MCCALLUM A. & PEREIRA F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

MAES F., PETERS S., DENOYER L. & GALLINARI P. (2009). Simulated iterative classification : A new learning procedure for graph labeling. In *European Conference on Machine Learning*.

MAURER A. (2006). Bounds for linear multi-task learning. *Journal of Machine Learning Research*, **7**, 117–139.

NOIVIRT-BRIK O., PRILUSKY J. & SUSSMAN J. L. (2009). Assessment of disorder predictions in casp8. *Proteins*.

POLLASTRI G., MARTIN A., MOONEY C. & VULLO A. (2007). Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. *BMC Bioinformatics*, **8**(1), 201.

PRUITT K. D., TATUSOVA T. & MAGLOTT D. R. (2006). NCBI reference sequences (RefSeq) : a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, **35**(suppl 1), D61–D65.

TSOCHANTARIDIS I., HOFMANN T., JOACHIMS T. & ALTUN Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning*.

ZHANG H., ZHANG T., CHEN K., KEDARISETTI K. D., MIZIANTY M. J., BAO Q., STACH W. & KURGAN L. (2011). Critical assessment of high-throughput standalone methods for secondary structure prediction. *Briefings in Bioinformatics*.

ZHANG H., ZHANG T., CHEN K., SHEN S., RUAN J. & KURGAN L. (2008). Sequence based residue depth prediction using evolutionary information and predicted secondary structure. *BMC bioinformatics*.