Pondération d'automates obtenus par alignements partiels de séquences protéiques

Thibaut Antoine, François Coste

Pondération d'automates obtenus par alignements partiels de séquences protéiques

Rapport de stage de L3 sous la direction de François Coste à l'équipe Dyliss, IRISA, Rennes, France.

THIBAUT ANTOINE

L'identification de la fonction d'une protéine à partir de sa séquence d'acides aminés est un enjeu majeur en bioinformatique. Pour répondre à ce défi, l'équipe Dyliss a proposé un modèle statistique pour caractériser des familles fonctionnelles de protéines nommé protomate, un automate pondéré construit à partir d'un multiple alignement partiel et local d'un échantillon d'apprentissage de séquences : chaque colonne de chaque alignement partiel et local est associée à un état dans le protomate. Actuellement, le poids d'un état est calculé en prenant en compte seulement la colonne dont il est issu, mais du fait de la partialité de l'alignement, cette colonne ne contient pas toutes les séquences de l'échantillon d'apprentissage. Nous proposons ici une modification du calcul de ces poids qui prend en compte les séquences protéiques de l'échantillon qui ne passent pas dans la colonne. De plus, un protomate calcule pour une séquence un score d'appartenance à la famille qu'il modélise qui repose sur la comparaison de la probabilité que la séquence appartienne à la famille selon le protomate avec la probabilité d'appartenance selon un modèle aléatoire. Cependant la distribution de ce modèle n'était pas calculée de manière homogène au calcul de l'appartenance selon le protomate, nous avons donc proposé une modification de ce modèle de sorte à rendre les deux calculs plus cohérents.

Additional Key Words and Phrases: Protomates, Poids, Séquences, Régularisation, Modèle nul

REMERCIEMENTS

Je remercie d'abord chaleureusement François Coste pour avoir accepté de m'accueillir en tant que stagiaire, pour m'avoir accompagné dans le vaste monde de la recherche, pour sa patience et sa bonne humeur. Je souhaite également remercier l'équipe Dyliss, ainsi que les membres des équipes Genscale et Genouest pour leur accueil et l'ambiance chaleureuse qu'ils maintiennent dans les bureaux. Enfin, merci aux personnes avec qui j'ai partagé la salle des stagiaires, qui ont su donner une touche fort agréable de convivialité à ce stage.

1 INTRODUCTION

Les protéines sont des macro-molécules. Elles sont constituées de plusieurs composés chimiques, les *acides aminés*, qui se succèdent pour former une séquence. En associant une lettre à chacun, on peut alors représenter une protéine par un mot sur l'alphabet des acides aminés, qu'on appelle séquence protéique. Les protéines ont la propriété de se replier sur elles-mêmes en trois dimensions, et ce repliement détermine leur fonction, c'est à dire leur rôle dans un organisme vivant. Or ces dernières années, l'apparition de technologies plus performantes a accéléré la croissance du nombre de protéines séquencées et les bases de données se trouvent remplies de séquences dont on ne connaît pas la fonction. Un des défis de la recherche est donc aujourd'hui de trouver des moyens efficaces pour obtenir la fonction d'une protéine à partir de sa représentation en séquence.

Pour remplir cet objectif, l'approche principale aujourd'hui consiste à regrouper les protéines qui ont une fonction similaire en familles, puis entraîner des modèles statistiques à reconnaître ces familles, pour enfin utiliser ces modèles sur de nouvelles séquences, pour prédire leur fonction.

Pour caractériser une famille, les modèles à l'état de l'art utilisent un échantillon de séquences appartenant à la famille, dont ils déterminent les segments conservés pour ensuite calculer le score d'une séquence en fonction de la présence ou non de ces segments dans cette séquence. Ces segments conservés sont déterminés en superposant les séquences de sorte à maximiser un score de similarité sur les zones, c'est ce qu'on appelle un *alignement*. Ce fonctionnement repose sur

l'hypothèse qu'un ensemble de séquences appartenant toutes à une même famille fonctionnelle partage des segments conservés, mais ce n'est pas nécessairement le cas, on peut imaginer une situation où une famille fonctionnelle contient deux sous ensembles ou plus de séquences qui partagent des segments conservés différents. Chaque segment zone conservée correspond alors à un bloc d'acides aminés dans l'alignement des séquences.

Pour modéliser ces divergences au sein d'une même famille fonctionnelle, l'équipe Dyliss, à l'IRISA Rennes, a développé un nouveau modèle nommé *protomate*. Un protomate est un automate pondéré de type machine de Moore, qui représente les enchaînements de zones conservées dans une famille de séquences protéiques. Une zone conservée de la famille est modélisé dans le protomate par une succession d'états appelés états de *match*. Les poids de ces états permettent de calculer le score d'appartenance d'une séquence à la famille que modélise le protomate.

Il existe aujourd'hui plusieurs manières de déterminer ces poids, qui prennent en compte les acides aminés de la zone conservée représentée par les états. Cependant ces méthodes ne prennent pas en compte le reste des aminés, ceux qui n'appartiennent pas à la zone conservée mais présents dans les autres séquences de l'échantillon d'apprentissage. Notre but est donc de trouver une manière d'améliorer le calcul des poids des états, de manière à ce qu'il prenne en compte les acides aminés des autres séquences de l'échantillon d'apprentissage, pas seulement ceux dans les séquences qui passent dans le bloc conservé.

La section 2 détaille la manière dont les familles de protéines sont modélisées grâce aux protomates. La section 3 présente l'état actuel de la recherche en ce qui concerne la pondération des protomates. Dans la section 4, nous présentons nos contributions à l'objectif défini ci-dessus.

2 ALIGNEMENTS, PROTOMATES ET SCORE D'UNE SÉQUENCE

Les protomates sont un modèle développé par [Ker08] pour modéliser une famille de protéines. Nous expliquons ici comment ils sont construits et comment ils fonctionnent ; dans ce but nous introduisons les concepts d'*alignement de séquence*, de *matrice poids-position*, et de *score d'une séquence*.

2.1 Alignements de séquences

L'alignement de séquences est une méthode utilisée en bioinformatique pour identifier les zones, c'est à dire les sous ensembles de positions d'une séquence, conservées entre les séquences d'un échantillon.

2.1.1 Alignement de deux séquences.

Définition. On note $\mathcal{A}$ l'alphabet des acides aminés, et soit g un caractère qui n'est pas dans $\mathcal{A}$. Soit deux séquences u, v sur l'alphabet $\mathcal{A}$. Un alignement de u et v est déterminé par 4 tuples, qui représentent intuitivement un découpage de u et de v, et pour chaque découpage un nombre de trous à insérer à l'intérieur des coupures :

$$(u_1, \dots, u_p), (i_0, \dots, i_p), (v_1, \dots, v_q), (j_0, \dots, j_q), \text{ pour } p, q \in \mathbb{N}^*.$$

Ces tuples doivent vérifier $u = u_1 \cdots u_p, v = v_1 \dots v_q$, et que les mots $u' = g^{i_0} u_1 g^{i_1} \cdots u_p g^{i_p}$ et $v' = g^{j_0} v_0 g^{j_1} \cdots v_q g^{j_q}$ ont la même longueur (la notation g^i désigne le mot contenant l'unique caractère g répété i fois). On note (u', v') l'alignement de u et v. On dit que deux lettres sont alignées dans l'alignement (u', v') lorsqu'elles sont à la même position dans u' et v'. Si aucune de ces lettres n'est g, on dit alors que c'est un *match*. Formellement, si on note $u' = a_1 \cdots a_\ell, v' = b_1 \cdots, b_\ell$, deux lettres alignées sont un couple (a_k, b_k), et un *match* est un couple (a_k, b_k) tel que $a_k, b_k \neq g$.

Intuitivement, cela revient à écrire les séquences u et v l'une au dessus de l'autre en leur ajoutant des trous ou en les décalant d'un certain nombre de positions à droite ou à gauche. Les trous sont modélisés par le caractère g, qu'on appelle *gap* et qui se note - en pratique.

Score d'un alignement. Comme on cherche en alignant deux séquences à identifier des segments conservés entre elles, tous les alignements de ces deux séquences ne nous intéressent pas. On cherche un alignement qui maximise la ressemblance entre les acides aminés superposés dans cet alignement.

Une fonction $s : \mathcal{A}^2 \to \mathbb{R}$ qui associe à chaque couple d'acides aminés un score de ressemblance (en pratique, s est déterminée par une matrice de substitution ; voir l'annexe A) est appelée *fonction de score*. Il est important de noter que s ne valorise pas simplement les paires d'acides aminés identiques, car des protéines ayant des fonctions similaires ont parfois un ancêtre évolutif commun, ce qui implique que certains acides aminés d'une séquence ont pu subir des mutations et il est nécessaire de les prendre en compte.

On peut alors définir le score d'un alignement avec s : pour deux séquences u, v, le score d'un alignement $A = (u', v')$ est donné par

$$\mathrm{Score}_s(A) = \sum_{\substack{k=1 \\ (a_k,b_k) \text{ match}}}^{\ell} s(a_k, b_k),$$

en reprenant la notation précédente. On ajoute généralement à ce score une pénalité pour les positions alignées de type *gap*-lettre ou *gap*-*gap* (voir [Dur+98]).

Un alignement optimal pour s est alors un alignement de u et v qui maximise Score_s. On peut déterminer un alignement optimal avec des algorithmes utilisant la programmation dynamique, comme l'algorithme de Needleman-Wunsch ([NW70]).

2.1.2 Alignements globaux, locaux, multiples, partiels. Le type d'alignement décrit précédemment est un alignement global : on essaye d'aligner les séquences dans leur entièreté. On définit par opposition des alignements locaux : on cherche seulement à aligner un segment dans chaque séquence. Les alignements locaux sont calculés aussi par programmation dynamique, grâce à l'algorithme de Smith-Waterman ([SW81]).

De plus, la section précédente ne décrit les alignements que pour deux séquences, mais on peut généraliser la définition pour trois ou plus. On parle alors d'alignement multiple. On peut alors définir les alignements multiples locaux de séquences, qui consistent à aligner un ensemble de segments tel que chacun est tiré d'une séquence distincte. Les positions alignées ne sont alors plus des paires mais des colonnes.

Enfin, [Ker08] introduit un dernier type d'alignement, l'alignement local partiel, qui consiste à aligner localement seulement un sous ensemble d'un ensemble de séquences, au lieu de toutes les aligner comme dans un alignement multiple. L'intérêt de cet alignement vient lorsqu'on en fait plusieurs : chaque alignement partiel peut concerner un sous ensemble de séquences différent. Ce dernier type d'alignement permet de définir la notion de *multiple alignement local et partiel* (PLMA), un ensemble d'alignements locaux et partiels d'un ensemble de séquences. Chaque alignement local et partiel d'un PLMA est appelé bloc de PLMA.

2.2 Matrices poids-position (PSSM)

Une matrice poids-position (en anglais *position specific scoring matrix*, PSSM) est un modèle de familles de protéines décrit dans [Dur+98], qui caractérise une famille par un ensemble de positions consécutives.

Définition. Une PSSM est une matrice dont les colonnes sont indexées par $1, \dots, \ell$ et les lignes par $\mathcal{A}$, et dont le coefficient en (a, i) donne la probabilité de trouver l'acide aminé a dans la colonne i.

Ainsi, une PSSM $M = (m_{a,i})$ attribue à un segment $x_1 \cdots x_\ell$ de taille ℓ une probabilité

$$P(x_1 \cdots x_l | M) = \prod_{i=1}^{\ell} m_{x_i,i},$$

c'est à dire le produit de la probabilité de chaque acide aminé du segment dans la colonne correspondante de la PSSM.

Construction. On peut construire une PSSM à partir d'un alignement multiple local de séquences. Si les colonnes de l'alignement sont $c_1, \dots, c_\ell$, les colonnes de la PSSM sont indexées par $1, \dots, \ell$, et le coefficient en (a, i) de la matrice donne la probabilité d'avoir l'acide aminé a dans la colonne c_i, que l'on peut calculer grâce à des méthodes présentées en section 3.

Score d'une séquence. Soit M une PSSM, et $u = x_1 \cdots x_\ell$ une séquence de taille ℓ que l'on veut comparer à M pour déterminer son score d'appartenance à la famille que décrit M. [Dur+98] explique que pour déterminer ce score, on compare M à un modèle aléatoire $\mathcal{R}$ qui associe à chaque $a \in \mathcal{A}$ une probabilité $P(a|\mathcal{R}) = q_0(a)$ (on dit que $\mathcal{R}$ est distribué selon q_0). Le score de u est alors calculé en *log-odds*, autrement dit

$$\text{Score}(u) = \log \frac{P(u|M)}{P(u|\mathcal{R})} = \sum_{i=1}^{\ell} \log \frac{m_{x_i,i}}{q_0(x_i)}.$$

Le score d'une séquence u de taille $k \geq \ell$ est le score maximal d'un segment de u de taille ℓ, autrement dit

$$\text{Score}(u) = \max_{1 \leq j \leq k-\ell+1} \text{Score}(u[j, \dots, j+\ell-1]),$$

où $u[j, \dots, j+\ell-1]$ est le segment de u compris entre les positions j et $j+\ell-1$ (incluses).

Comme ces matrices décrivent les positions consécutives d'une famille, elles ne peuvent pas modéliser des insertions ou des délétions dans les séquences d'une famille et donc ne donneront pas des bons scores à des protéines ayant subi de tels changements. Pour reconnaître de telles protéines, d'autres modèles ont été développés comme les profils HMM [Edd98] qui incluent des états de *gap*, permettant de consommer des caractères sans influencer le score.

2.3 Protomates

Les protomates sont des modèles introduits par [Ker08] pour modéliser des familles de protéines pouvant présenter des segments conservés chez seulement une partie des membres de la famille en exhibant ces divergences à l'aide d'un automate pondéré.

Définition. Soit un échantillon d'apprentissage contenant plusieurs séquences appartenant à une même famille de protéines. Pour construire un protomate qui reconnaît cette famille de protéines, on commence par calculer un PLMA (défini en 2.1.2) des séquences de l'échantillon d'apprentissage. Chaque bloc du PLMA est associé à une succession d'états dans le protomate, en associant chaque colonne c du bloc à un état du protomate. Ces états sont appelés états de *match*. Un protomate est un automate de type machine de Moore, c'est à dire que les états émettent des lettres et non les transitions ; ici, l'état du protomate associé à la colonne c est pondéré pour émettre une lettre $a \in \mathcal{A}$ avec une certaine probabilité notée $p_c(a)$. On trouve aussi dans un protomate des états de *gap* qui bouclent sur eux-mêmes pour consommer des caractères, et qui émettent chaque acide aminé a

avec sa probabilité de fond (c'est à dire la probabilité de le trouver dans une séquence quelconque) notée $p_0(a)$, ainsi qu'un état de début et un état de fin.

On ajoute ensuite une transition entre deux blocs de PLMA A et B lorsqu'il y a au moins une séquence de l'échantillon d'apprentissage utilisée dans A et B sans qu'un segment de cette séquence entre A et B soit utilisé dans un bloc C.

On remarque alors qu'une fois le protomate construit, les blocs qu'on obtient ne sont pas tous construits à partir du même nombre de séquences d'apprentissage (car l'alignement dont ils sont issus était partiel), et ce nombre est souvent inférieur au nombre total de séquences dans l'échantillon d'apprentissage. Or, on le montre par la suite, cela n'est pas pris en compte par l'état de l'art lors de la pondération des PSSM ; pourtant il semble naturel de ne pas pondérer de la même manière deux PSSM qui n'ont pas à disposition la même quantité d'information.

Score d'une séquence. Soit $\mathcal{M}$ un protomate qui modélise une famille de protéines et u une séquence dont on veut calculer le score d'appartenance à la famille.

Pour calculer ce score, on a besoin d'abord de définir le score d'un chemin d'états $e = (e_1, \ldots, e_n)$ dans $\mathcal{M}$ acceptant pour u. Comme pour le score d'une séquence dans une PSSM, on veut comparer en *log-odds* la probabilité de u selon le protomate à la probabilité de u selon un modèle nul. [Ruf17] montre alors que si on utilise le modèle nul des probabilités de fond, les états de *gap* ne contribuent pas au score de u sur le chemin e. Autrement dit si les états de *match* du chemin sont notés $f_1, \ldots, f_\ell$, chacun associé à une colonne c_i du PLMA et émettant un acide aminé a_i, le score de u sur e est

$$\text{Score}(u) = \sum_{i=1}^{\ell} \log \frac{p_{c_i}(a_i)}{p_0(a_i)}.$$

Finalement, de manière analogue au calcul du score pour les PSSM, on définit le score d'une séquence dans un protomate comme le maximum des scores de la séquence sur les chemins acceptants dans le protomate. Ce score est calculé avec l'algorithme de Viterbi ([Vit67]), qui utilise la programmation dynamique.

3 ÉTAT DE L'ART DE LA PONDÉRATION DES PROTOMATES

On trouve dans la littérature plusieurs méthodes pour calculer la probabilité qu'un certain acide aminé se trouve dans une certaine colonne. Dans la suite, on les présente sous l'angle de notre problématique, à savoir comment elles prennent en compte la taille de la colonne, si elles le font.

Notations. On rappelle ici certaines des notations déjà utilisées auparavant qui seront utilisées dans toute la suite.

- $\mathcal{A}$ est l'alphabet des acides aminés, de cardinal 20.
- Si $\vec{x} = (x_j)_{j \in I}$ est un vecteur, alors on notera $\vec{x}(j) = x_j$ et $|\vec{x}| = \sum_{j \in I} x_j$.
- $p_c(a)$ est la probabilité de l'acide aminé a dans la colonne c d'un bloc de PLMA.
- $\vec{o}_c$ est le vecteur d'observations d'acides aminés dans la colonne c d'un bloc de PLMA, i.e. pour $a \in \mathcal{A}$, $\vec{o}_c(a)$ est le nombre de fois que a apparaît dans c.
- $p_0(a)$ est la probabilité de fond de l'acide aminé a, c'est à dire la probabilité de trouver l'acide aminé a dans une séquence quelconque.

On peut noter que si c est une colonne d'un bloc de PLMA, $|\vec{o}_c|$ est le nombre de séquences qui passent dans le bloc. Plus précisément, l'échantillon d'apprentissage étant biaisé, on ajoute des poids aux séquences de telle sorte qu'un ensemble de plusieurs séquences très similaires ne comptent que pour une seule (voir [Ruf17]). Ainsi, $|\vec{o}_c|$ désigne en réalité la somme des poids des séquences passant dans c.

3.1 Première approche

Une première approche du calcul des $p_c(a)$ consiste à modéliser la distribution des acides aminés dans une colonne à l'aide d'une loi multinomiale.

Loi multinomiale. Les lois multinomiales sont une généralisation des lois binomiales. En effet, si une loi binomiale modélise les lancers successifs d'une pièce pas nécessairement équilibrée, une loi multinomiale modélise des lancers successifs d'un dé pouvant avoir plus de deux faces, pas nécessairement équilibré non plus.

Formellement, si $N = (N_1, \ldots, N_k)$ est un vecteur aléatoire qui suit une loi multinomiale de paramètres $n, \pi = (\pi_1, \ldots, \pi_k)$, alors on a

$$P(N_1 = n_1, \ldots, N_k = n_k) = \begin{cases} \frac{n}{n_1! \cdots n_k!} \pi_1^{n_1} \cdots \pi_k^{n_k} & \text{si } \sum_{i=1}^{k} n_i = n \\ 0 & \text{sinon} \end{cases}.$$

Il est clair que pour une colonne donnée c, le vecteur $\vec{o}_c$ suit une loi multinomiale de paramètres $|\vec{o}_c|, \vec{\pi}_c$, avec $\vec{\pi}_c$ inconnu. [Kar95] permet de voir le calcul des $p_c(a)$ comme en fait une estimation des paramètres $\pi_c(a)$ d'une loi multinomiale.

Estimation des paramètres. Soit $N = (N_1, \ldots, N_k)$ un vecteur aléatoire qui suit une loi multinomiale dont on veut estimer les paramètres $\pi_1, \ldots, \pi_k$. Soit $n_1, \ldots, n_k$ les valeurs observées pour les variables aléatoires $N_1, \ldots, N_k$, et $n = \sum_i n_i$. Il a été montré ([AF97]) que sans *a priori* sur la distribution, le meilleur estimateur de π_i au sens du maximum de vraisemblance était la fréquence observée de i, n_i/n.

Dans notre contexte, cela revient à poser

$$p_c(a) := \frac{\vec{o}_c(a)}{|\vec{o}_c|}.$$

Cette approche a déjà été étudiée par la littérature ([CBK99]) qui a montré qu'elle ne donne de bons résultats que pour un nombre d'observations suffisamment large, ce qui n'est pas le cas en pratique.

De plus, on peut voir que cette approche ne prend pas en compte le nombre d'observations car même en multipliant la colonne considérée par un facteur positif quelconque, les probabilités estimées ne changent pas.

3.2 Approche générale

Pour obtenir des estimations des $\pi_c(a)$ plus intéressantes que la simple fréquence d'observations de l'acide aminé a dans la colonne, ([Kar95]) propose de se servir de connaissances *a priori* sur les distributions des acides aminés, c'est à dire régulariser les observations.

En notant $\vec{o}'_c(a)$ le compte d'acides aminés a dans la colonne régularisée, [Kar95] pose alors

$$p_c(a) := \frac{\vec{o}'_c(a)}{|\vec{o}'_c|},$$

c'est à dire le meilleur estimateur pour une loi multinomiale dont les valeurs observées auraient été données par $\vec{o}'_c$.

Nous donnons par la suite des exemples de régularisations présents dans la littérature.

3.3 Scores de substitution

[Kar95] explique que l'on peut utiliser les matrices de substitution (voir l'annexe A) pour régulariser les comptes d'une colonne, en posant

$$\vec{o}'_c(a) := \sum_{b \in \mathcal{A}} s(a, b) \vec{o}_c(b),$$

où $S = (s(a,b))_{a,b\in\mathcal{A}}$ est une matrice de substitution telle que pour tous $a, b \in \mathcal{A}$, $s(a,b)$ donne la probabilité que b mute en a.

Nous montrons ici que si $s(a,b) = p_0(a,b)/p_0(b) = P(a|b)$, la probabilité d'avoir l'acide aminé a sachant un vecteur d'observation ne contenant qu'un acide aminé b (avec $p_0(a,b)$ la probabilité de trouver les acides aminés a et b dans deux séquences alignées), alors on a

$$p_c(a) = \frac{\sum_{b\in\mathcal{A}} P(a|b)\vec{o}_c(b)}{\sum_{b'\in\mathcal{A}}\sum_{b\in\mathcal{A}} P(b'|b)\vec{o}_c(b)} = \frac{\sum_{b\in\mathcal{A}} P(a|b)\vec{o}_c(b)}{\sum_{b\in\mathcal{A}} \vec{o}_c(b) \underbrace{\sum_{b'\in\mathcal{A}} P(b'|b)}_{=1}} = \sum_{b\in\mathcal{A}} P(a|b)\frac{\vec{o}_c(b)}{|\vec{o}_c|}.$$

Cette expression ne dépend donc pas du nombre d'observations, car on normalise les $\vec{o}_c(b)$ par $|\vec{o}_c|$, et en multipliant la colonne par un nombre positif on obtient toujours la même probabilité $p_c(a)$.

3.4 Pseudo-comptes

Le problème principal de la méthode des scores de substitution décrite ci-dessus est que les $p_c(a)$ calculés avec cette méthode ne convergent pas vers la fréquence d'observation quand la taille de c tend vers l'infini, il y a toujours un biais qui donne aux acides aminés qu'on n'a jamais observés une probabilité non négligeable d'apparaître dans la colonne.

Les pseudo-comptes sont une manière de pallier à ce problème. L'idée est de calculer la fréquence d'acides aminés dans la colonne c à laquelle on a rajouté une certaine quantité $\vec{\psi}_c$ d'acides aminés qu'on prétend avoir vus, qui correspondent à une distribution *a priori* des acides aminés. Le nombre d'acides aminés ajoutés dépend de la méthode utilisée. Ainsi, on aura pour ces méthodes une formule de la forme :

$$p_c(a) \coloneqq \frac{\vec{o}_c(a) + \vec{\psi}_c(a)}{|\vec{o}_c| + |\vec{\psi}_c|},$$

autrement dit on régularise les observations avec $\vec{o}'_c(a) \coloneqq \vec{o}_c(a) + \vec{\psi}_c(a)$.

Cette formule prend en compte le nombre d'observations dans le calcul de la probabilité, en proposant un compromis entre les données *a priori* et les données observées. En effet, si on a peu de données, autrement dit $|\vec{o}_c| \ll |\vec{\psi}_c|$, alors on peut faire l'approximation $p_c(a) \approx \vec{\psi}_c(a)/|\vec{\psi}_c|$, c'est-à-dire qu'on estime la probabilité de a uniquement à l'aide des données *a priori*. Réciproquement, si $|\vec{o}_c| \gg |\vec{\psi}_c|$ (on a observé beaucoup de séquences), alors $p_c(a) \approx \vec{o}_c(a)/|\vec{o}_c|$, autrement dit on se rapproche du meilleur estimateur, contrairement à la méthode des scores de substitution.

Cependant, il reste la difficulté de trouver les $\vec{\psi}_c(a), a \in \mathcal{A}$ optimaux. On donne par la suite différentes méthodes présentées dans la littérature.

3.4.1 Pseudo-comptes uniformes.

Sans a priori. Une première approche consiste à utiliser l'estimation de Bayes-Laplace en posant $\forall a \in \mathcal{A} : \vec{\psi}_c(a) \coloneqq 1$, autrement dit tous les acides aminés sont équiprobables. Les auteurs de [CBK99] expliquent que cette méthode ne donne de bons résultats que si $|\vec{o}_c|$ est élevé.

Avec l'a priori des probabilités de fond. On sait grâce aux probabilités de fond que tous les acides aminés ne sont pas équiprobables, c'est pourquoi [HH96] proposent de les utiliser pour le calcul des $\vec{\psi}_c(a)$. En supposant fixé un nombre total A_c de pseudo-comptes à ajouter en tout, ils posent $\vec{\psi}_c(a) \coloneqq A_c \cdot p_0(a)$.

Pour la valeur de A_c, les auteurs proposent plusieurs options : $A_c := \sqrt{|\vec{o}_c|}$, ou $A_c := \text{cst}, A_c \propto D_c$, le nombre d'acides aminés différents dans la colonne.

3.4.2 Pseudo-comptes d'Henikoff et Henikoff. Henikoff et Henikoff expliquent que les probabilités de fond ne sont pas suffisantes car elles ne prennent pas en compte les différentes relations entre les acides aminés. En effet, au cours du temps les acides aminés d'une protéine peuvent muter, mais sous la pression évolutive toutes les paires d'acides aminés n'ont pas une chance égale de se substituer. Les auteurs donnent l'exemple du tryptophane et de la phénylalaline qui se substituent souvent ; on s'attend donc à trouver une phénylalaline plus fréquemment que sa probabilité de fond si on observe un tryptophane.

En réponse à ce problème, les auteurs proposent d'utiliser les probabilités de substitution, déterminées avec des matrices de substitution (voir annexe A). En reprenant les notations précédentes, les auteurs posent :

$$\vec{\psi}_c(a) := A_c \sum_{b \in \mathcal{A}} p_0(a, b)$$

puisqu'on doit considérer tous les acides aminés qui ont pu se substituer avec a.

Cependant cette formule ne prend pas en compte les observations de la colonne. Pour les introduire, Henikoff et Henikoff proposent d'utiliser la formule de Bayes :

$$\vec{\psi}_c(a) := A_c \sum_{b \in \mathcal{A}} P(b|\vec{o}_c) P(a|b).$$

Finalement, avec $P(b|\vec{o}_c) = \vec{o}_c(b)/|\vec{o}_c|$ et $P(a|b) = p_0(a, b)/Q_b$ où $Q_b = \sum_{b' \in \mathcal{A}} p_0(b', b)$ (par la formule de Bayes) :

$$\vec{\psi}_c(a) = A_c \sum_{b \in A} \frac{\vec{o}_c(b)}{|\vec{o}_c|} \frac{p_0(a, b)}{Q_b}.$$

On peut noter que l'expression de $\vec{\psi}_c(a)/A_c$ est la même que celle donnée en 3.3 pour $p_c(a)$, à la valeur de $P(a|b)$ près. Celle donnée ici par les auteurs est plus générale, car [Kar95] suppose que $Q_b = p_0(b)$ ce qui n'est pas toujours le cas en pratique : les probabilités de fond ne sont pas déterminées de manière absolue mais par rapport à un contexte [Edd04].

3.5 Mixtures de Dirichlet

Si les méthodes de pseudo-comptes précédentes peuvent fournir de bons résultats, [Sjö+96] ont remarqué qu'elles pouvaient être réunies et généralisées grâce aux *distributions* et *mixtures de Dirichlet*, qui permettent de considérer comme *a priori* des distributions typiques de colonnes, à la place matrices utilisées les méthodes précédentes. Nous les présentons en dessous. Tous les calculs sont tirés de leur article, et certains sont fournis en annexe B.

3.5.1 Fonctions B *et* Γ*.* Les calculs réalisés par [Sjö+96] reposent sur les fonctions Γ et B d'Euler. On peut les exprimer comme ceci :

$$\forall x > 0 : \Gamma(x) = \int_0^{+\infty} t^{x-1} e^t \mathrm{d}t,$$

et

$$\forall x_1, \ldots, x_n > 0 : \mathrm{B}(x_1, \ldots, x_n) = \int_{\mathcal{P}_n} \prod_{i=1}^{n} p_i^{x_i - 1} \mathrm{d}\vec{p} = \frac{\Gamma(x_1) \cdots \Gamma(x_n)}{\Gamma(x_1 + \cdots + x_n)},$$

où $\mathcal{P}_n$ est le simplexe en dimension $n : \mathcal{P}_n = \left\{\vec{p} = (p_1, \ldots, p_n) \in [0, 1]^n : \sum_i p_i = 1\right\}$. La fonction Γ est le prolongement continu de la factorielle sur les réels, on a pour un entier k, $\Gamma(k + 1) = k!$.

3.5.2 Distribution de Dirichlet. Une distribution de Dirichlet ρ paramétrée par $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$ est une distribution de probabilités sur $\mathcal{P}_n$, définie par :

$$\rho(\vec{p}) = \frac{1}{Z} \prod_{i=1}^{n} p_i^{\alpha_i - 1}$$

où $Z = \mathrm{B}(\vec{\alpha})$ est telle que $\int_{\mathcal{P}_n} \rho(\vec{p}) \mathrm{d}\vec{p} = 1$.

Dans notre contexte, un certain $\vec{p} \in \mathcal{P}_n$ correspond à une distribution de probabilités sur les acides aminés, i.e. $n = 20$ et si $a \in \mathcal{A}$: $\vec{p}(a) = P(a|\vec{p})$ est la probabilité de l'acide aminé a. En faisant l'hypothèse que pour la probabilité d'une distribution sur les acides aminés $\vec{p}$ sachant une colonne *a priori* $\vec{\alpha}$, $P(\vec{p}|\vec{\alpha})$ est $\rho(\vec{p})$, on peut montrer que [1] :

$$P(a|\vec{o}_c, \vec{\alpha}) = \frac{\vec{o}_c(a) + \vec{\alpha}(a)}{|\vec{o}_c| + |\vec{\alpha}|}.$$

On retrouve exactement la formule précédente des pseudo-comptes, avec $\vec{\alpha}$ à la place de $\vec{\psi}_c$. Le nombre d'observations est donc pris en compte dans la formule, de la même manière que pour les pseudo-comptes.

Le formalisme mathématique permet aux auteurs de généraliser les pseudo-comptes en n'utilisant pas qu'un seul vecteur, mais plusieurs, desquels on prendra une combinaison linéaire en fonction des observations.

3.5.3 Mixtures de Dirichlet. Si $\rho_1, \ldots, \rho_M$ sont des distributions de Dirichlet chacune paramétrée par $\vec{\alpha}_j$, et $q_1, \ldots, q_M$ des réels positifs tels que $\sum_j q_j = 1$, alors

$$\rho = \sum_{j=1}^{M} q_j \rho_j$$

est une mixture de Dirichlet paramétrée par $\Theta = (\vec{\alpha}_1, \ldots, \vec{\alpha}_M, q_1, \ldots, q_M)$. Les ρ_j sont appelées *composantes de la mixture.*

On peut alors montrer que

$$p_c(a) \coloneqq P(a|\vec{o}_c, \Theta) = \sum_{j=1}^{M} \frac{\vec{o}_c(a) + \vec{\alpha}_j(a)}{|\vec{o}_c| + |\vec{\alpha}_j|} P(\vec{\alpha}_j|\vec{o}_c, \Theta),$$

ce qui peut s'interpréter comme expliqué en conclusion de la section précédente. On peut alors montrer que

$$P(\vec{\alpha}_j|\vec{o}_c, \Theta) = \frac{q_j P(\vec{o}_c|\vec{\alpha}_j)}{\sum_{k=1}^{M} q_k P(\vec{o}_c|\vec{\alpha}_k)} = \frac{\mathrm{B}(\vec{o}_c + \vec{\alpha}_j)/\mathrm{B}(\vec{\alpha}_j)}{\sum_{k=1}^{M} \mathrm{B}(\vec{o}_c + \vec{\alpha}_k)/\mathrm{B}(\vec{\alpha}_k)},$$

car

$$P(\vec{o}_c|\vec{\alpha}_j) = \frac{|\vec{o}_c|!}{\prod_{a \in \mathcal{A}} \vec{o}_c(a)!} \cdot \frac{\mathrm{B}(\vec{o}_c + \vec{\alpha}_j)}{\mathrm{B}(\vec{\alpha}_j)}.$$

On peut noter que cela revient à régulariser en posant

$$\vec{o}'_c(a) \coloneqq \sum_k q_k \frac{\mathrm{B}(\vec{o}_c + \vec{\alpha}_k)}{\mathrm{B}(\vec{\alpha}_k)} \frac{\vec{o}_c(a) + \vec{\alpha}_k(a)}{|\vec{o}_c| + |\vec{\alpha}_k|}.$$

Les auteurs expliquent dans l'article comment estimer les paramètres Θ optimaux, et donnent leurs résultats pour l'apprentissage de mixtures à 9 composantes.

De plus, ils expliquent que le nombre d'observations est pris en compte dans le calcul de la probabilité : si on a beaucoup d'observations à disposition alors on fait confiance à ces données

1. Le calcul est fourni en annexe B.

(autrement dit on s'approche de l'estimation par le maximum de vraisemblance), mais le cas contraire est important aussi. Car contrairement aux méthodes de pseudo-comptes simples qui n'ont qu'une seule colonne typique, ici on utilise les quelques observations que l'on a pour inférer quelle composante de la mixture (i.e. quel vecteur de pseudo comptes) est la plus probable. Pour cette raison, les mixtures de Dirichlet font plus usage des observations dans la colonne que les simples méthodes de pseudo-comptes. Néanmoins, on remarque que ce calcul ne prend pas en compte les séquences de l'échantillon d'apprentissage qui ne passent pas par la colonne c dans le PLMA à l'origine du protomate qu'on considère, il considère uniquement les séquences qui passent dans c.

4 AMÉLIORATION DE LA PONDÉRATION ET DU MODÈLE NUL

Cette section présente les avancées que nous proposons, c'est à dire la mise en place d'un nouveau calcul des $p_c(a)$, plus adapté aux protomates. On fixe dans toute la section un protomate, une séquence s et une colonne c telle que le meilleur chemin du protomate acceptant s passe par c, de vecteur de comptes $\vec{o}_c$; soit $m = |\vec{o}_c|$ et n le nombre total de séquences dans l'échantillon d'apprentissage du protomate. On utilisera aussi la notation suivante : pour un entier $n \geq 1$, $\vec{u}_n$ est une *colonne avec a priori p_0 de norme n*, i.e. pour $a \in \mathcal{A}$, $\vec{u}_n(a) = np_0(a)$. On a bien $|\vec{u}_n| = n$.

4.1 Prise en compte des séquences manquantes

Notre contexte de départ est le suivant : on cherche à savoir si une séquence protéique s appartient à une famille, modélisée par un protomate. On détermine alors le chemin optimal du protomate acceptant pour s, et on calcule les probabilités des différents acides aminés de la séquence selon les poids des transitions du chemin optimal. Or, les poids des transitions sont calculés uniquement par rapport à la colonne du PLMA à laquelle chacune est associée, autrement dit on fait la supposition que s a un lien évolutif avec l'une des séquences de la colonne, mais ce n'est pas nécessairement le cas. La séquence peut être issue d'une toute autre séquence parmi les séquences d'entraînement. Nous voulons donc prendre cette deuxième possibilité en compte dans le calcul de $p_c(a)$.

4.1.1 Prise en compte des autres séquences. Soit a un acide aminé présent dans c, $1 \leq i \leq n$ l'indice d'une séquence d'entraînement. On note b_i l'acide aminé de la séquence i aligné avec a dans l'alignement multiple initial, et E_i l'événement suivant : « La séquence i est un ancêtre évolutif de la séquence s ».

Notre hypothèse précédente implique que les E_i forment un système complet d'événements et on peut alors utiliser la loi des probabilités totales. Cela donne :

$$p_c(a) = \sum_{i=1}^{n} P(a|E_i)P(E_i).$$

Dans l'alignement optimal, on considère que $P(a|E_i) = P(a|b_i)$ par définition de E_i et b_i. De plus, comme il n'y a *a priori* pas de raison particulière pour qu'une séquence spécifique soit un ancêtre évolutif de s, on supposera $P(E_i) = 1/n$ pour tout i [2]. Ainsi :

$$p_c(a) = \frac{1}{n} \sum_{i=1}^{n} P(a|b_i).$$

On peut alors regrouper la somme en paquets de b_i identiques, ce qui nécessite de connaître pour chaque $b \in \mathcal{A}$ le nombre d'acides aminés b alignés avec a. On sait déjà qu'il y a tous les acides aminés de la colonne c puisque a est supposé aligné avec, il y en a $\vec{o}_c(b)$, mais il y a aussi tous les

2. En réalité, du fait de la pondération des séquences (voir le début de la section 3), cette probabilité sera supposée égale au poids de la séquence i sur la somme de tous les poids des séquences. Voir l'annexe C pour plus de détails.

acides aminés dans les $n - m$ autres séquences d'entraînement à considérer. Comme on n'a aucune information sur ces séquences, on fait l'hypothèse la moins forte : les acides aminés sont distribués selon p_0 dans le reste des séquences. Avec cette hypothèse, on peut donc rajouter $(n - m)p_0(b)$ observations pour chaque acide aminé b. On obtient alors :

$$p_c(a) = \frac{1}{n} \sum_{b \in \mathcal{A}} (\vec{o}_c(b) + (n - m)p_0(b))P(a|b) = \frac{1}{n} \sum_{b \in \mathcal{A}} (\vec{o}_c(b) + \vec{u}_{n-m}(b))P(a|b),$$

d'où :

$$p_c(a) = \frac{m}{n} \sum_{b \in \mathcal{A}} \frac{\vec{o}_c(b)}{m} P(a|b) + \frac{n - m}{n} \sum_{b \in \mathcal{A}} \frac{\vec{u}_{n-m}(b)}{n - m} P(a|b),$$

soit

$$p_c(a) = \frac{m}{n} P(a|\vec{o}_c, \mathcal{S}) + \left(1 - \frac{m}{n}\right) P(a|\vec{u}_{n-m}, \mathcal{S}),$$

où $P(a|\vec{x}, \mathcal{S})$ est la probabilité de l'acide aminé a sachant un vecteur d'observations $\vec{x}$, régularisé par une matrice de substitution $\mathcal{S}$ (voir 3.3 et 3.4.2). On peut noter que la taille de la colonne uniforme n'a aucune influence sur la valeur de $p_c(a)$.

En mettant l'expression de $p_c(a)$ sous cette forme, on se rend compte qu'elle peut se généraliser pour d'autres régularisations que $\mathcal{S}$, en utilisant une autre information *a priori*, $\mathcal{K}$, en remplaçant $P(a|\vec{o}_c, \mathcal{S})$ par $P(a|\vec{o}_c, \mathcal{K})$ dont on donne l'expression avant (voir la section 3). Néanmoins, nous n'avons pas de justification théorique que cette généralisation est légitime.

4.2 Choix du modèle nul

Comme on l'a expliqué en 2.3, le calcul du score d'une séquence utilise un modèle nul, de comparaison. La question du changement de ce modèle est déjà évoquée dans [Ruf17] qui montre que l'utilisation du score de la séquence miroir comme modèle nul donne de bons résultats. Néanmoins le modèle nul actuellement utilisé dans l'outil d'apprentissage des protomates est toujours le modèle classique basé sur les probabilités de fond, i. e. pour une séquence u dont le chemin acceptant optimal dans le protomate passe par les colonnes $c_1, \ldots, c_\ell$, et en notant a_j l'acide aminé émis par l'état associé à la colonne c_j :

$$\text{Score}(u) = \sum_{j=1}^{\ell} \log \frac{p_{c_j}(a_j)}{p_0(a_j)}.$$

Or il y a là une hétérogénéité, car on ne calcule pas $p_c(a)$ et $p_0(a)$ de la même manière. Notre seconde contribution est de rendre le calcul de $p_0(a)$ homogène avec celui de $p_c(a)$.

Modèle nul homogène. On a montré dans la section précédente que l'on pouvait calculer les probabilités des acides aminés sachant une colonne en complétant les séquences manquantes par des acides aminés apparaissant avec un *a priori* p_0. On applique ce même raisonnement pour calculer les probabilités du modèle nul : la probabilité $q_0(a)$ d'un acide aminé a dans le modèle nul est la probabilité de a sans qu'on ait aucun *a priori*, c'est à dire qu'on complète toutes les séquences avec des acides aminés d'*a priori* p_0, pas seulement une partie. Cela revient donc à poser $q_0(a) = P(a|\vec{u}_n, \mathcal{S})$.

De la même manière que dans la section précédente, on voudrait pouvoir généraliser le calcul des $q_0(a)$ pour des données *a priori* $\mathcal{K}$ différentes de $\mathcal{S}$. Cette généralisation semble valable si $P(a|\vec{u}_n, \mathcal{K})$ est indépendante de n, car on aurait alors effectivement une homogénéité entre le modèle nul et le modèle informé sans avoir à donner d'argument concernant la norme de la colonne uniforme utilisée, mais il faut donner un argument permettant de choisir entre $\vec{u}_n$ ou $\vec{u}_k$ pour un autre entier k si $P(a|\vec{u}_n, \mathcal{K})$ dépend de n (ce qui est le cas pour les méthodes de pseudo-comptes ou les mixtures

de Dirichlet). Une fois la généralisation faite, on pourrait alors calculer les $q_0(a)$ en utilisant les formules données dans la section 3.

Score des états de gap. Avant l'amélioration que l'on propose ici, le modèle nul avait pour distribution de probabilité p_0, et les états de gap étaient aussi pondérés par p_0 ce qui résultait en un score de 0 pour l'alignement sur ces états lors du calcul du score de la séquence ([Ruf17]). On vérifie si c'est notre cas aussi.

On modélise un état de *gap* par une colonne vide, notée $\varnothing$. Alors on a, vue notre formule précédente :

$$p_\varnothing(a) = \frac{0}{n} P(a|\varnothing, \mathcal{S}) + \left(1 - \frac{0}{n}\right) P(a|\vec{u}_n, \mathcal{S}) = P(a|\vec{u}_n, \mathcal{S}).$$

Ainsi, on a pour le score d'un état de *gap* et vu le choix du modèle nul (4.2) :

$$\text{Score}(gap) = \log\left(\frac{p_\varnothing(a)}{P(a|\vec{u}_n, \mathcal{S})}\right) = \log\left(\frac{P(a|\vec{u}_n, \mathcal{S})}{P(a|\vec{u}_n, \mathcal{S})}\right) = 0.$$

Ainsi, on associe bien un score nul aux états de *gap*, qui ne contribuent donc pas au score de la séquence.

5 PUBLICATIONS EN LIEN AVEC LE SUJET

5.1 Les protomates

La thèse de Goulven Kerbellec ([Ker08]) introduit pour la première fois le concept de *protomate*. L'objectif était de construire un modèle mathématique et informatique plus expressif que ceux à l'état de l'art en modélisant les dépendances et les divergences dans les segments conservés à l'intérieur d'une famille de protéines. Le choix s'est porté sur les automates, plus expressifs que les modèles à l'état de l'art, plutôt qu'une machine à états encore plus expressive car les caractéristiques des familles de protéines se remarquent à échelle locale ce qui correspond aux langages réguliers, contrairement aux séquences d'ADN ou d'ARN qui présentent des similarités qu'on ne peut modéliser qu'avec des classes de langages plus expressives, mais également parce que des machines plus expressives auraient été beaucoup plus coûteuses en calcul de paramètres.

Pour cela, comme on l'explique en section 2, [Ker08] introduit également l'idée d'alignement local partiel, et donne des algorithmes pour les construire ainsi que pour construire les protomates.

5.2 Pondération des protomates et acceptation d'une séquence

Suite à cette thèse, deux stages : [Pic09 ; Ruf17] ont étudié la pondération des protomates.

[Pic09] propose plusieurs méthodes de pondération des automates reprises de celles qui existaient déjà pour d'autres modèles, comme les PSSMs et les profils HMM décrits en [Dur+98] : les pseudo-comptes et les mixtures de Dirichlet. Il introduit également l'idée du calcul d'une p-valeur pour déterminer l'appartenance ou non d'une séquence à une famille en fonction de son score dans le protomate.

[Ruf17] poursuit le travail de [Pic09] en introduisant l'idée de la pondération de séquences afin de limiter les biais dans l'échantillon d'apprentissage. Elle propose aussi une normalisation du score par le nombre d'états traversés par une séquence, en remarquant que plus une séquence traverse d'états, plus elle aura un score élevé sans pour autant avoir plus de chances d'appartenir à la famille modélisée. Enfin, elle améliore le calcul de la significativité des scores en donnant une preuve expérimentale que la loi de distribution des scores est une loi de Gumbel.

Notre apport consiste à améliorer la pondération des automates proposée par [Pic09] pour prendre en compte le fait que les colonnes d'un bloc de PLMA à l'origine d'un protomate ne contiennent pas toutes les séquences de l'échantillon d'entraînement.

5.3 Régularisation des observations

Enfin, un dernier axe important de notre travail repose sur l'idée de la régularisation des observations. La formalisation que l'on donne du problème repose sur [Kar95], qui montre que l'on peut voir toutes les méthodes de calcul des probabilités des acides aminés dans une colonne comme une estimation des paramètres d'une loi multinomiale. La meilleure estimation au sens du maximum de vraisemblance est la fréquence d'observation, mais en pratique celle-ci ne donne pas de bons résultats car on a trop peu de données à disposition. Ainsi, [Kar95] propose d'ajouter aux observations des connaissances *a priori* sur les acides aminés, c'est à dire les régulariser, pour améliorer la qualité de l'estimation. Cette approche générale a été implémentée par de nombreuses approches : les mixtures de Dirichlet, les pseudo-comptes et l'approche des scores de substitution.

6 CONCLUSION

L'équipe Dyliss de l'IRISA a proposé un modèle de familles de protéines appelé *protomate*, plus expressif que ce qu'on trouve dans l'état de l'art aujourd'hui, en permettant de décrire des alternatives dans les positions caractéristiques d'une famille de protéines, contrairement aux modèles actuels qui ne proposent qu'une description linéaire de la famille.

Si les méthodes de pondération originellement conçues pour les outils de l'état de l'art peuvent être utilisées sur les protomates, elles ne prennent pas en compte la spécificité des multiples alignements locaux et partiels qui est qu'une colonne d'un bloc de ces alignements ne contient pas toutes les séquences de l'échantillon d'apprentissage, autrement dit la colonne ne contient pas toutes les informations de l'échantillon.

Notre travail propose d'adapter ces méthodes de pondération en prenant en compte le manque d'informations mentionné au dessus. Nous avons également proposé une modification du calcul du score d'une séquence en modifiant les probabilités du modèle nul dont le calcul n'était pas homogène au calcul des probabilités des acides aminés dans les colonnes.

Si notre travail donne une piste d'amélioration des protomates, nous n'avons pas eu l'occasion de la tester sur des données expérimentales afin de la valider empiriquement, ce qui est l'étape suivante nécessaire pour valider l'approche. Par ailleurs, il pourrait être pertinent à l'avenir de chercher des justifications théoriques et empiriques à une éventuelle généralisation de l'expression de la probabilité d'un acide aminé, ainsi que l'expression des probabilités du modèle nul. En effet, on ne les a démontrées que pour pour une régularisation des observations avec une matrice de score, mais il semble intéressant de vérifier si elles se généralisent pour d'autres régularisations, comme les mixtures de Dirichlet.

RÉFÉRENCES

[AF97] Khursheed Alam et Zhuojen Feng. "Estimating probability of occurrence of the most likely multinomial event". In : *Journal of Statistical Planning and Inference* 59.2 (avr. 1997), p. 257-277. issn : 03783758. doi : 10.1016/S0378-3758(96)00112-7. url : https://linkinghub.elsevier.com/retrieve/pii/S0378375896001127.

[Alt91] Stephen F. Altschul. "Amino acid substitution matrices from an information theoretic perspective". In : *Journal of Molecular Biology* 219.3 (5 juin 1991), p. 555-565. issn : 0022-2836. doi : 10.1016/0022-2836(91)90193-A. url : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7130686/.

[CBK99] Melissa Cline, Christian Barrett et Kevin Karplus. *ISMB99 Tutorial Material Making the most of your hidden Markov models Handout Material*. 1999. url : https://compbio.soe.ucsc.edu/ismb99.tutorial.html.

[Dur+98] Richard Durbin et al. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge : Cambridge University Press, 1998. isbn : 9780521629713. doi : 10.1017/CBO9780511790492. url : https://www.cambridge.org/core/books/biological-sequence-analysis/921BB7B78B745198829EF96BC7E0F29D (visité le 12/07/2022).

[Edd98] S. R. Eddy. "Profile hidden Markov models". In : *Bioinformatics (Oxford, England)* 14.9 (1998), p. 755-763. issn : 1367-4803. doi : 10.1093/bioinformatics/14.9.755.

[Edd04] Sean R Eddy. "Where did the BLOSUM62 alignment score matrix come from ?" In : *Nature Biotechnology* 22.8 (août 2004), p. 1035-1036. issn : 1087-0156, 1546-1696. doi : 10.1038/nbt0804-1035. url : http://www.nature.com/articles/nbt0804-1035.

[HH96] Jorja G. Henikoff et Steven Henikoff. "Using substitution probabilities to improve position-specific scoring matrices". In : *Bioinformatics* 12.2 (1996), p. 135-143. issn : 1367-4803, 1460-2059. doi : 10.1093/bioinformatics/12.2.135. url : https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/12.2.135.

[HH92] S Henikoff et J G Henikoff. "Amino acid substitution matrices from protein blocks." In : *Proceedings of the National Academy of Sciences* 89.22 (15 nov. 1992), p. 10915-10919. issn : 0027-8424, 1091-6490. doi : 10.1073/pnas.89.22.10915. url : https://pnas.org/doi/full/10.1073/pnas.89.22.10915.

[Kar95] K. Karplus. "Evaluating regularizers for estimating distributions of amino acids". In : *Proceedings. International Conference on Intelligent Systems for Molecular Biology* 3 (1995), p. 188-196. issn : 1553-0833.

[Ker08] Goulven Kerbellec. "Apprentissage d'automates modélisant des familles de séquences protéiques". These de doctorat. Rennes 1, 1er jan. 2008. url : https://www.theses.fr/2008REN1S052.

[NW70] S. B. Needleman et C. D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". eng. In : *Journal of Molecular Biology* 48.3 (mars 1970), p. 443-453. doi : 10.1016/0022-2836(70)90057-4.

[Pic09] Vincent Picard. *Pondération d'automates modélisant des familles de protéines et significativité des scores*. report. 2009. url : https://hal.inria.fr/inria-00431111.

[Ruf17] Manon Ruffini. "Better scoring schemes for the recognition of functional proteins by protomata". other. Rennes 1, 28 juin 2017. url : https://hal.inria.fr/hal-01557941.

[Sjö+96] Kimmen Sjölander et al. "Dirichlet mixtures : a method for improved detection of weak but significant protein sequence homology". In : *Bioinformatics* 12.4 (1996), p. 327-345. issn : 1367-4803, 1460-2059. doi : 10.1093/bioinformatics/12.4.327. url : https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/12.4.327.

[SW81] T.F. Smith et M.S. Waterman. "Identification of common molecular subsequences". In : *Journal of Molecular Biology* 147.1 (mars 1981), p. 195-197. issn : 00222836. doi : 10.1016/0022-2836(81)90087-5. url : https://linkinghub.elsevier.com/retrieve/pii/0022283681900875.

[Vit67] A. Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In : *IEEE Transactions on Information Theory* 13.2 (avr. 1967), p. 260-269. issn : 1557-9654. doi : 10.1109/TIT.1967.1054010.

Annexe A MATRICES DE SUBSTITUTION

Une matrice de substitution est une matrice dont les lignes et les colonnes sont indexées par $\mathcal{A}$, et si $a, b \in \mathcal{A}$, le coefficient a, b de de la matrice est le score de substitution de la paire (a, b), c'est à dire une mesure d'à quel point a est susceptible de muter en b, et réciproquement. Il existe plusieurs manières de calculer ce score.

La matrice BLOSUM62. La construction de la matrice BLOSUM62 est décrite dans [HH92] et [Edd04]. L'idée principale derrière cette matrice est l'utilisation de score *log-odds* pour comparer deux hypothèses à propos de deux acides aminés $a, b \in \mathcal{A}$: la première notée $\mathcal{H}_0$, qui postule que a et b sont indépendants et leur alignement dans une colonne n'est due qu'au hasard, et la seconde, notée $\mathcal{H}_1$, qui énonce que a et b sont liés évolutivement, autrement dit lors d'un alignement de séquences la présence de l'un dans une colonne influe sur la présence de l'autre. Le score de substitution de a et b s'exprime alors :

$$s(a, b) = \frac{1}{\lambda} \log \frac{P(a \cap b|\mathcal{H}_1)}{P(a \cap b|\mathcal{H}_0)},$$

où l'événement $a \cap b$ est « Les acides aminés a et b apparaissent dans la même colonne lors de l'alignement de deux séquences » ou de manière plus concise : « a et b sont alignés ». Le facteur $1/\lambda$ permet de normaliser les probabilités.

Par définition, on alors $P(a \cap b|\mathcal{H}_1) = p_0(a, b)$ et $P(a \cap b|\mathcal{H}_0) = p_0(a)p_0(b)$, puis

$$s(a, b) = \frac{1}{\lambda} \log \frac{p_0(a, b)}{p_0(a)p_0(b)}.$$

On en déduit

$$p_0(a, b) = p_0(a)p_0(b)e^{\lambda s(a,b)},$$

et λ est tel que

$$\sum_{a,b \in \mathcal{A}} p_0(a)p_0(b)e^{\lambda s(a,b)} = 1.$$

La probabilité $p_0(a, b)$ est mesurée dans [HH92] à l'aide d'un grand nombre d'alignements de séquences, sélectionnés par les auteurs de manière à avoir une similitude d'au plus 62%. Cela implique que l'on peut construire d'autres matrices de substitution en changeant ce seuil, et [HH92] présentent en particulier les matrices BLOSUM45 et BLOSUM80, mais ils montrent qu'elles donnent de moins bons résultats que BLOSUM62.

D'autres matrices. Les matrices BLOSUM et leurs alternatives PAM ([Alt91]) sont aujourd'hui très utilisées, mais on trouve dans la littérature d'autres exemples. En particulier [Kar95] introduit la matrice $S = (p_0(a, b)/p_0(b))_{a,b \in \mathcal{A}}$ dont le coefficient a, b est $P(a|b)$, le *relatedness odds ratio* de a par rapport à b. En fait, l'égalité

$$P(a|b) = \frac{p_0(a, b)}{p_0(b)}$$

est vraie théoriquement, mais pas forcément en pratique car elle dépend du calcul de $p_0(a, b)$. Par exemple elle est fausse si comme précédemment on calcule les probabilités de substitution seulement sur des séquence ayant un certain degré maximal de ressemblance.

Annexe B MIXTURES DE DIRICHLET

Cette annexe présente les calculs fournis dans [Sjö+96] qui prouvent les expressions fournies en 3.5. On réutilise les notations établies dans cette section.

Lemme 1. *Si $\vec{p} \in \mathcal{P}_n$, alors :*

$$P(\vec{o}_c|\vec{p}, |\vec{o}_c|) = \frac{|\vec{o}_c|!}{\prod_{a\in\mathcal{A}} \vec{o}_c(a)!} \prod_{a\in\mathcal{A}} p_a^{\vec{o}_c(a)}.$$

Démonstration. Cela découle du fait que par hypothèse et interprétation des éléments de $\mathcal{P}_n$, $\vec{o}_c$ suit une loi multinomiale de paramètres $\vec{p}, |\vec{o}_c|$. □

Lemme 2. *Soit $\vec{p} \in \mathcal{P}_n$. Si $\vec{\alpha}$ est un vecteur de paramètres d'une distribution de Dirichlet sur $\mathcal{P}_n$, alors $P(\vec{p}|\vec{\alpha})$ (i.e. la probabilité d'un élément de $\mathcal{P}_n$ sous l'hypothèse que ses éléments suivent une distribution de Dirichlet paramétrée par $\vec{\alpha}$) vaut :*

$$P(\vec{p}|\vec{\alpha}) = \frac{1}{\mathrm{B}(\vec{\alpha})} \prod_{a\in\mathcal{A}} p_a^{\alpha_a - 1}.$$

Démonstration. Cela vient du fait que $P(\vec{p}|\vec{\alpha}) = \rho(\alpha)$ si ρ est une distribution de Dirichlet paramétrée par $\vec{\alpha}$. □

Lemme 3.

$$P(\vec{o}_c|\vec{\alpha}, |\vec{o}_c|) = \frac{|\vec{o}_c|!}{\prod_{a\in\mathcal{A}} \vec{o}_c(a)!} \frac{\mathrm{B}(\vec{o}_c + \vec{\alpha})}{\mathrm{B}(\vec{\alpha})}.$$

Démonstration. Par la loi des probabilités totales, on a :

$$P(\vec{o}_c|\vec{\alpha}, |\vec{o}_c|) = \int_{\mathcal{P}_n} P(\vec{o}_c|\vec{\alpha}, |\vec{o}_c|, \vec{p}) P(\vec{p}|\vec{\alpha}, |\vec{o}_c|) \mathrm{d}\vec{p} = \int_{\mathcal{P}_n} P(\vec{o}_c|\vec{p}, |\vec{o}_c|) P(\vec{p}|\vec{\alpha}) \mathrm{d}\vec{p},$$

car si $\vec{\alpha}$ est fixé, alors $\vec{p}$ ne dépend pas du nombre d'observations, et de même si $\vec{p}$ est fixé, le nombre d'observations ne dépend plus de $\vec{\alpha}$. Ainsi par les lemmes 1 et 2 :

$$\begin{aligned} P(\vec{o}_c|\vec{\alpha}, |\vec{o}_c|) &= \int_{\mathcal{P}_n} \frac{|\vec{o}_c|!}{\prod_{a\in\mathcal{A}} \vec{o}_c(a)!} \frac{1}{\mathrm{B}(\vec{\alpha})} \prod_{a\in\mathcal{A}} p_a^{\vec{o}_c(a)+\alpha_a-1} \\ &= \frac{|\vec{o}_c|!}{\prod_{a\in\mathcal{A}} \vec{o}_c(a)!} \frac{1}{\mathrm{B}(\vec{\alpha})} \underbrace{\int_{\mathcal{P}_n} \prod_{a\in\mathcal{A}} p_a^{\vec{o}_c(a)+\alpha_a-1} \mathrm{d}\vec{p}}_{=\mathrm{B}(\vec{o}_c+\vec{\alpha})}, \end{aligned}$$

et finalement

$$P(\vec{o}_c|\vec{\alpha}, |\vec{o}_c|) = \frac{|\vec{o}_c|!}{\prod_{a\in\mathcal{A}} \vec{o}_c(a)!} \frac{\mathrm{B}(\vec{o}_c + \vec{\alpha})}{\mathrm{B}(\vec{\alpha})},$$

ce qui conclut. □

Lemme 4.

$$P(\vec{p}|\vec{\alpha}, \vec{o}_c) = \frac{1}{\mathrm{B}(\vec{\alpha} + \vec{o}_c)} \prod_{a\in\mathcal{A}} p_a^{\vec{\alpha}(a)+\vec{o}_c(a)-1}.$$

Démonstration. Par la formule de Bayes :

$$P(\vec{p}|\vec{\alpha}, \vec{o}_c) = \frac{P(\vec{p}, \vec{\alpha}, \vec{o}_c||\vec{o}_c|)}{P(\vec{\alpha}, \vec{o}_c||\vec{o}_c|)} = \frac{P(\vec{o}_c|\vec{p}, \vec{\alpha}, |\vec{o}_c|) P(\vec{\alpha}, \vec{p})}{P(\vec{o}_c|\vec{\alpha}, |\vec{o}_c|) P(\vec{\alpha})}.$$

Or par le même argument qu'on utilise pour le lemme 3, $P(\vec{o}_c|\vec{p}, \vec{\alpha}, |\vec{o}_c|) = P(\vec{o}_c|\vec{p}, |\vec{o}_c|)$, et la formule de Bayes donne $P(\vec{p}, \vec{\alpha})/P(\vec{\alpha}) = P(\vec{p}|\vec{\alpha})$. Ainsi, par les 3 lemmes précédents :

$$P(\vec{p}|\vec{\alpha}, \vec{o}_c) = \frac{P(\vec{o}_c|\vec{p}, |\vec{o}_c|) P(\vec{p}|\vec{\alpha})}{P(\vec{o}_c|\vec{\alpha}, |\vec{o}_c|)} = \frac{\left(\frac{|\vec{o}_c|!}{\prod_{a\in\mathcal{A}} \vec{o}_c(a)!} \prod_{a\in\mathcal{A}} p_a^{\vec{o}_c(a)}\right) \left(\frac{1}{\mathrm{B}(\vec{\alpha})} \prod_{a\in\mathcal{A}} p_a^{\alpha_a-1}\right)}{\frac{|\vec{o}_c|!}{\prod_{a\in\mathcal{A}} \vec{o}_c(a)!} \frac{\mathrm{B}(\vec{o}_c+\vec{\alpha})}{\mathrm{B}(\vec{\alpha})}},$$

et après quelques simplifications :

$$P(\vec{p}|\vec{\alpha},\vec{o}_c) = \frac{1}{\mathrm{B}(\vec{\alpha}+\vec{o}_c)} \prod_{a\in\mathcal{A}} p_a^{\vec{\alpha}(a)+\vec{o}_c(a)-1},$$

ce qui conclut. □

Remarque. Cela prouve que les distributions d'acides aminés *a posteriori* (c'est à dire après avoir observé une colonne) suivent une loi de Dirichlet de paramètres $\vec{\alpha}+\vec{o}_c$.

Théorème. *Soit $a \in \mathcal{A}$ un acide aminé, $\vec{\alpha}$ les paramètres d'une distribution de Dirichlet et $\vec{o}_c$ le vecteur d'observations d'une colonne. Alors*

$$P(a|\vec{o}_c,\vec{\alpha}) = \frac{\vec{o}_c(a)+\vec{\alpha}(a)}{|\vec{o}_c|+|\vec{\alpha}|}.$$

Démonstration. Par la loi des probabilités totales :

$$P(a|\vec{o}_c,\vec{\alpha}) = \int_{\mathcal{P}_n} P(a|\vec{p},\vec{o}_c,\vec{\alpha})P(\vec{p}|\vec{o}_c,\vec{\alpha})\mathrm{d}\vec{p}.$$

Or par interprétation des $\vec{p}$, on sait que $P(a|\vec{p},\vec{o}_c,\vec{\alpha}) = p_a$, et par le lemme 4 on a :

$$P(a|\vec{o}_c,\vec{\alpha}) = \int_{\mathcal{P}_n} p_a \frac{1}{\mathrm{B}(\vec{\alpha}+\vec{o}_c)} \prod_{b\in\mathcal{A}} p_b^{\vec{\alpha}(b)+\vec{o}_c(b)-1}\mathrm{d}\vec{p} = \frac{\mathrm{B}(\vec{\alpha}'+\vec{o}_c)}{\mathrm{B}(\vec{\alpha}+\vec{o}_c)},$$

où $\vec{\alpha}'$ est tel que $\forall b \neq a : \vec{\alpha}'(b) = \vec{\alpha}(b)$ et $\vec{\alpha}'(a) = \vec{\alpha}(a)+1$. On peut alors simplifier l'expression :

$$\begin{aligned}\frac{\mathrm{B}(\vec{\alpha}'+\vec{o}_c)}{\mathrm{B}(\vec{\alpha}+\vec{o}_c)} &= \frac{\Gamma(\vec{\alpha}(a)+1+\vec{o}_c(a))\prod_{b\neq a}\Gamma(\vec{\alpha}(b)+\vec{o}_c(b))}{\Gamma(|\vec{o}_c|+|\vec{\alpha}|+1)} \frac{\Gamma(|\vec{o}_c|+|\vec{\alpha}|)}{\prod_{b\in\mathcal{A}}\Gamma(\vec{\alpha}(b)+\vec{o}_c(b))} \\ &= \frac{\Gamma(\vec{o}_c(a)+\vec{\alpha}(a)+1)}{\Gamma(\vec{o}_c(a)+\vec{\alpha}(a))} \frac{\Gamma(|\vec{o}_c|+|\vec{\alpha}|)}{\Gamma(|\vec{o}_c|+|\vec{\alpha}|+1)}.\end{aligned}$$

Or comme $\Gamma(x) = (x-1)!$ si x est entier, $\Gamma(x+1)/\Gamma(x) = x!/(x-1)! = x$. Ainsi :

$$P(a|\vec{o}_c,\vec{\alpha}) = \frac{\vec{o}_c(a)+\vec{\alpha}(a)}{|\vec{o}_c|+|\vec{\alpha}|},$$

ce qui conclut. □

Annexe C CALCUL DES PROBABILITÉS AVEC PRISE EN COMPTE DU POIDS DES SÉQUENCES

Comme on l'explique au début de la section 3, afin de limiter les biais dans l'échantillon d'apprentissage, on attribue à chaque séquence i de l'échantillon un poids w_i, de sorte à ce que plusieurs séquences très similaires ne comptent que pour une. Cela modifie légèrement le calcul des $p_c(a)$ qu'on fait en section 4, nous le détaillons ici.

Comme on considère ici que les séquences sont pondérées, on aura $n = \sum_i w_i$ et m la somme des poids des séquences qui passent dans c.

Comme précédemment, on a :

$$p_c(a) = \sum_{i=1}^{n} P(a|E_i)P(E_i),$$

avec $P(a|E_i) = P(a|b_i)$, et cette fois-ci $P(E_i) = w_i/n$. Ainsi

$$p_c(a) = \sum_{i=1}^{n} \frac{w_i}{n} P(a|b_i).$$

On peut alors regrouper la somme par paquets de b_i identiques :

$$p_c(a) = \sum_{b \in \mathcal{A}} \sum_{\substack{i \\ b_i = b}} P(a|b_i) \frac{w_i}{n} = \sum_{b \in \mathcal{A}} \frac{1}{n} P(a|b) \sum_{\substack{i \\ b_i = b}} w_i.$$

On découpe ensuite la somme $\sum_{\substack{i \\ b_i = b}} w_i$ entre les séquences qui passent dans c et celles qui n'y passent pas :

$$\sum_{\substack{i \\ b_i = b}} w_i = \sum_{\substack{i \\ b_i = b \\ i \text{ passe dans } c}} w_i + \sum_{\substack{i \\ b_i = b \\ i \text{ ne passe pas dans } c}} w_i.$$

Le premier terme vaut $\vec{o}_c(b)$ par définition, et le second correspond à la somme des poids des séquences ne passant pas dans c dont l'acide aminé aligné sur c est b. On s'attend en moyenne à trouver une proportion $p_0(b)$ de telles séquences, et comme la somme des poids des séquences ne passant pas dans c est $n - m$, on évalue le second terme à $(n - m)p_0(b)$.

Finalement,

$$p_c(a) = \frac{1}{n} \sum_{b \in \mathcal{A}} (\vec{o}_c(b) + (n - m)p_0(b))P(a|b),$$

ce qui correspond à l'expression obtenue sans considérer les poids des séquences.