compte rendu – projet bioinformatique – Analyse statistique d'une famille de Protéines

Ben Kirane Malik

28 mars 2018

Résumé

Il nous est proposé d'analyser une famille de protéines préalablement séquencée. Nous rendrons compte de comment exploiter des outils statistiques et probabilistes pour analyser ces séquences protéiques et en déduire des propriétés de conservation au sein de la famille, d'appartenance à la famille et finalement pour esquisser une corrélation entre l'alignement des protéines et des mesures spatiales.

1 Préliminaire

Dans l'intégralité des analyses nous ferons référence à une unique famille de protéines, D_{train}. Nous avons accès à $M = 5643$ protéines toutes de cette famille. Chaque protéine est décrite par une séquence et ces séquences, forment l'*alignement* que nous étudions par la suite. Une protéine de l'alignement se présente comme une suite d'acide aminés (il y en a 20) avec éventuellement des trous dans la séquence.

Il s'agit dans un premier temps de lire l'alignement proposé (fichier `data/Dtrain.txt`) au format FASTA. Chaque ligne non précédée du caractère '>' décrit une protéine de la famille D_{train} sur l'alphabet

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

pour les $L = 48$ positions possibles de l'alignement que nous souhaitons étudier.

Les implémentations des analyses du projet sont réalisés en `python` (version `3.6`). Ici, pour la lecture de l'alignement le code se trouve dans le module `data` (`src/data.py`) où on peux utiliser la méthode `read_fasta(filename)` pour lire les lignes qui nous intéressent dans `data/Dtrain.txt` et avoir la liste des séquences de l'alignement.

À présent que nous avons accès à l'alignement, nous souhaiterions savoir ce qui caractérise les séquences qui le constituent et notamment comment mettre en évidence les positions caractéristiques fortes de l'alignement.

2 Conservation des positions et caractère de l'alignement

L'alignement se représente avec une matrice M lignes et L colonnes. Les lignes de cette matrice correspondent aux séquences et les colonnes aux positions de l'alignement.

2.1 Évaluation du caractère de l'alignement

On s'intéresse au nombre d'occurences d'un acide aminé $a \in \mathcal{A}$ à la position i, $0 \leq i \leq L-1$. On note $n_i(a)$ cette quantité. En supposant que les positions sont indépendantes on souhaite déterminer ω tel que

$$P(a_0 \dots a_{L-1}|\omega) = \prod_{i=0}^{L-1} \underbrace{P(a_i|\omega)}_{\omega_i(a_i)}$$

il s'agit de la matrice communément appelée "position-specific weight matrix" ou PSWM.

Il est donc question d'estimer la PSWM ω avec les $n_i(a)$ et en utilisant un estimateur de fréquence corrigé avec un pseudo-compteur de valeur 1, donc

$$\omega_i(a) = \frac{n_i(a) + 1}{M + |\mathcal{A}|}.$$

Le calcul d'estimation de la PSWM est implémenté dans le module `pswm` (`src/pswm.py`) par `estimate_pswm(data)` avec `data` la liste des séquences lues depuis `data/Dtrain.txt`.

Pour caractériser notre famille, on va chercher les position les plus conservées. Pour cela, nous mesurons la quantité d'information relative à une position

$$S_i = \log_2(q) + \sum_{a \in A} \omega_i(a) \cdot \log_2(\omega_i(a))$$

c'est l'entropie de Shannon légèrement corrigé. L'implémentation de ce calcul est réalisé par `shannon_pswm(i, pswm)` dans `pswm` et `real2()` du même module réalise toute la procédure jusqu'au tracé de l'entropie relative en fonction de la position (FIGURE 1).

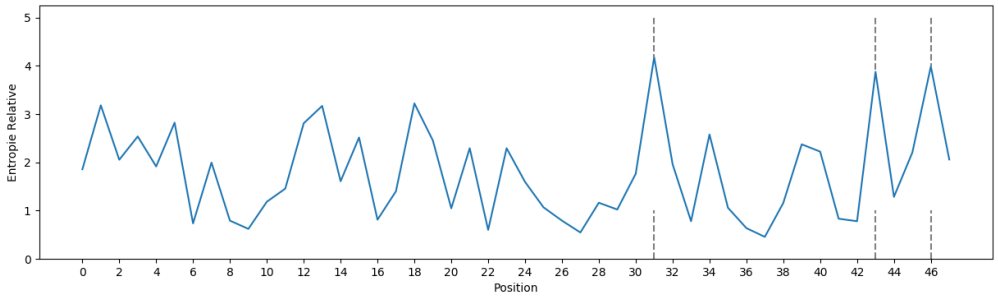

FIGURE 1 – Quantité d'information aux positions de l'alignement

L'entropie maximale est presque significativement atteinte aux positions 31, 46 et 43. Cela signifie que ces positions ont un rôle important pour la famille de protéines. Ces positions sont très fortement conservées et si il advenait qu'elles soient modifées, cela pertuberait fortement la fonction biologique première de la protéine.

Par la suite nous allons examiner si la PSWM nous permet d'identifier si d'autres séquences sont aussi dans la famille.

2.2 Critére d'appartenance à la famille de protéines

Étant donné une séquence quelconque $b_0 \dots b_{N-1}$, $N \geq L$, nous souhaitons savoir si un fragment de cette séquence appartient à la famille de protéines. Supposons d'abord $N = L$.

On peut estimer un modèle qui ne soit pas spécifiique aux positions, c'est à dire tel que

$$P(b_0 \dots b_{L-1}) = \prod_{i=0}^{L-1} f^{(0)}(b_i)$$

et

$$\forall b \in \mathcal{A}, f^{(0)}(b) = \frac{1}{L} \sum_{i=0}^{L-1} \omega_i(b).$$

Il s'agit d'un modèle nul. Nous comparons alors ce modèle nul avec le modèle PSWM évalué sur la famille de protéines D_{train}. Pour cela, nous évaluons la quantité

$$l(b_0 \dots b_{L-1}) = \log_2 \frac{P(b_0 \dots b_{L-1}|\omega)}{P(b_0 \dots b_{L-1})}$$

ou encore

$$l(b_0 \dots b_{L-1}) = \sum_{i=0}^{L-1} \left(\log_2 \omega_i(b_i) - \log_2 f^{(0)}(b_i) \right).$$

Si $l(b_0 \dots b_{L-1}) > 0$, on saura que la séquence b est plus probable d'apparaître dans le modèle PSWM que le modèle nul et réciproquement si $l(b_0 \dots b_{L-1}) < 0$, b est peu probable d'appartenir à la famille. La mesure l est appelée *log-vraissemblance*.

Reprenons maintenant à $N \geq L$. Il suffit de faire glisser une fenêtre de taille L sur b et de noter pour chaque décalage la valeur de l sur la fenêtre.

Nous disposons d'une séquence dans `data/testseq.txt` pour laquelle nous aimerions savoir si une partie appartient à la famille de protéines D_{train}.

Dans le module `pswm` (`src/pswm.py`),
— `null_model(pswm)` est la méthode qui calcul les valeurs $f^{(0)}$ décrivant le modèle nul,
— `log_odds(sequence, pswm, f0)` calcul la log-vraissemblance d'une séquence,
— `sliding_odds(sequences, pswm)` calcul sur l'ensemble des fenêtres possibles les log-vraissemblances.
Finalement la fonction `real4()` réalise l'ensemble des manipulations décrites jusqu'ici et affiche le tracé de la log-vraissemblance en fonction du décalage de la fenêtre glissante (FIGURE 2).

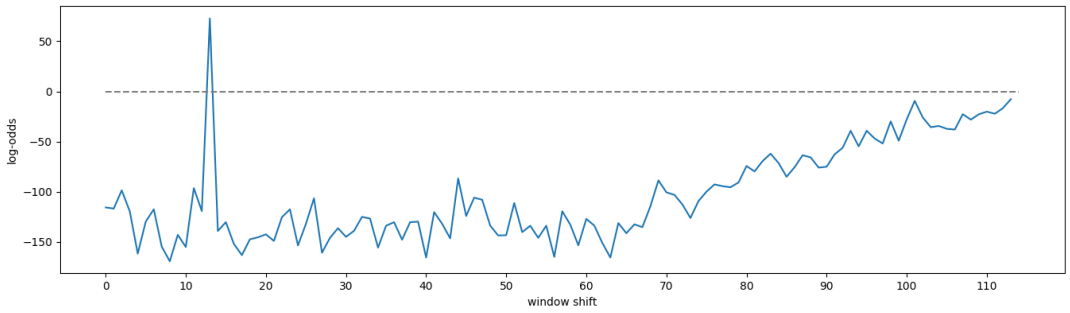

FIGURE 2 – Log-vraissemblances sur la séquence test

Pour la sous-séuence $(b_{13} \dots b_{13+L-1})$ on observe que la log-vraissemblance est positive, c'est d'ailleurs la seule fenêtre où c'est le cas, et par conséquent cette sous-séquence appartient très probablement à la famille de protéine que l'on étudie. Il est aussi très probable que cette sous-séquence ai les mêmes fonctions biologiques que les protéines de la famille étudié.

3 Co-évolution et corrélation avec la structure spatiale

Nous allons chercher à présent à mettre en relation l'information des séquences de l'alignement avec la structure spatiale d'une protéine représentative de la famille. Cela dit, nous devons enrichir le modèle PSWM décrit avant pour tenir compte de la structure spatiale.

La piste proposée est d'estimer la co-évolution de deux positions. Pour deux acides aminés a et b on souhaite savoir s'ils sont présents dans la séquence tel que a soit à une première position donnée et que b soit à une seconde position donnée : il s'agit là d'une co-occurence.

Soit $n_{i,j}(a,b)$ le nombre de séquences tel que a est à la position i et b est à la position j. Pour estimer les poids $\omega_{i,j}(a,b)$ associés aux $n_{i,j}(a,b)$, il faut conserver la relation $\sum_b \omega_{i,j}(a,b) = \omega_i(a)$.

Donc les

$$\omega_{i,j}(a,b) = \frac{n_{i,j}(a,b) + 1/q}{M + q}$$

sont les composantes de notre modèle enrichit.

Il est possible mainteant de mesurer la quantité d'information mutuelle. Il s'agit de déterminer à quel point deux positions sont corrélées.

La quantité d'information mutuelle pour un couple de positions (i, j) est la quantité

$$M_{i,j} = \sum_{a,b \in \mathcal{A}} \omega_{i,j}(a,b)[\log_2 \omega_{i,j}(a,b) - \log_2 \omega_i(a)\omega_j(b)]$$

Nous disposons d'une table des distances pour des couples de positions (fichier `data/distances.txt`). On considère qu'une paire est en contact si la distance entre ces positions est inférieur à 8Å (0.8nm). En triant les paires (i, j) par valeurs décroissantes de $M_{i,j}$ autrement dit des paires les plus corrélées au moins corrélées et en déterminant la fraction de paires en contact pour chaque tranche de paires selon ce tri, on souhaite déterminer une relation entre la structure spatiale et l'information statistique dont on dispose pour la famille. La FIGURE 3 est le tracé de cette fraction en fonction de la taille des tranches (il y a au plus L^2 tranches).

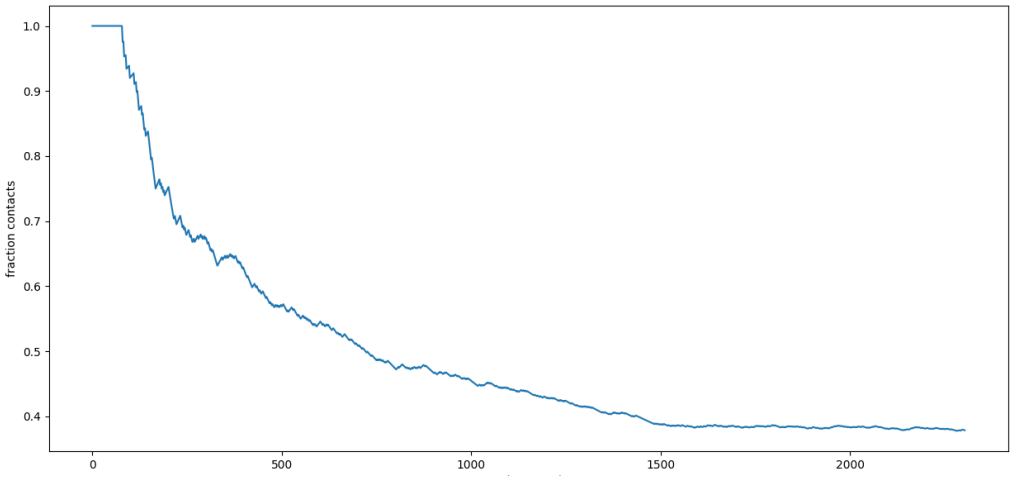

FIGURE 3 – Résidus en contacts (taille de la tranche en abscisse)

Tout le procédé jusqu'à la réalisation du tracé est implémenté par `real4()` dans le moduel `coev` (`src/coev.py`). Cette réalisation s'appui sur les fonctions

— `estimate_cooc(data)` pour le calcul des $\omega_{i,j}(a, b)$;
— `mutual_information(cooc, pswm, M)` pour le calcul des $M_{i,j}$;
— `induced_contacts(mim, distances)` qui détermine les fractions des paires en contacts ;

toutes dans même module `coev`.

On observe que plus on considère une tranche de paires fortement corrélées, plus la fraction de paires en contacts est importante. D'une part il s'agit d'un résultat obtenu par comptage sur les séquences et d'autre part il s'agit d'une mesure physique et d'une propriété spatiale. Pour cette famille de protéines, on peut induire la propriété spatiale de contact à partir du procédé statistique décrit ici, i.e., les paires les plus corrélées ont une probabilité élevée d'être en contact.

Conclusion : Nous avons mis en évidence que les outils statistiques sont des outils puissant et utiles en biologie. Les procédés décrits pour l'analyse de la famille de protéines étudiée sont reproductibles si on dispose d'un alignement d'une autre famille et ils nous permettraient d'identifier

— les positions caractérisitiques de la famille qui déterminent l'expression de ses fonctions biologiques,
— d'autres séquences appartenant probablement à la famille et
— des propriétés spatiales, qui, enrichies par d'autres procédures, nous permettrait de visualiser en 3D la structure spatiale d'une protéine représentative de la famille.