# Mini projet 2 : Analyse statistique d'une famille de protéines

Fabien Tang
Valentin Colliard

2018

## 1.1 Introduction

Lors de ce projet, nous avons cherché à analyser statistiquement une famille de protéines donnée par un alignement de séquence en détectant :
- les positions conservées,
- les séquences appartenant à la même famille,
- les corrélations entre colonnes différentes de l'alignement et leur relation avec les distances entres acides aminés dans la structure 3D d'une protéine représentative de la famille.

## 1.2 Données

Afin d'étudier les données fournies, nous avons créé une fonction de lecture de nos fichiers :
- Dtrain.txt contenant M=5643 séquences de protéines d'une même famille. Chaque séquence a pour longueur L= 48 positions et chaque acide aminé appartient à A = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y,-} (q=21=longueur de A).
- testseq.txt contenant une séquence de longueur N = 114.
- distances.txt contenant les distances entre paires d'acides aminés sous forme position 1, position 2, aa.

## 1.3 Modélisation par PSWM

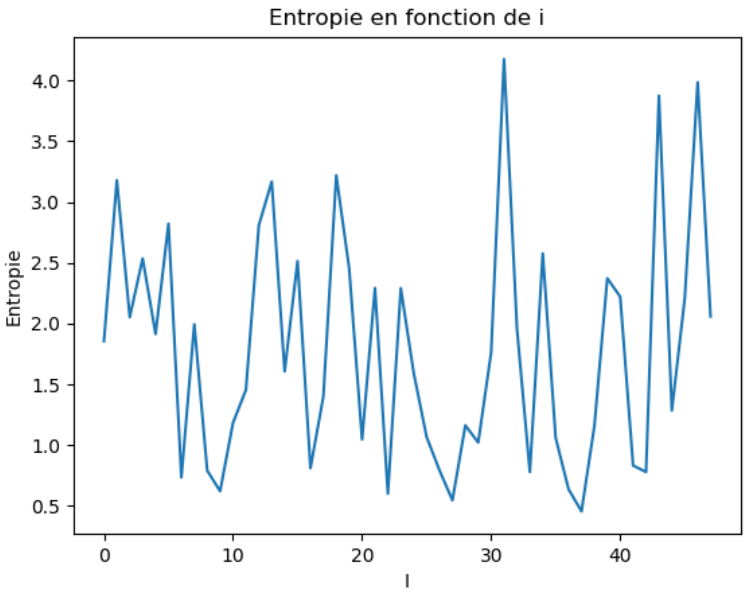
### 1.3.1 Matrice de poids spécifiques des positions

La PSWM signifie Position-Specific weight matrix soit la matrice de poids spécifiques des positions.
Afin d'obtenir cette matrice, nous avons défini les fonctions n(i,a,liste) et w(i,a,liste) permettant respectivement de compter le nombre d'occurences et de calculer le poids d'un acide aminé a à une position i donnée.
Les fonctions n_global(liste) et w_global(liste) utiliserons ainsi les fonctions précédentes pour la création de la matrice pour chaque position i = 0, ..., L-1 et chaque acide aminé a appartenant à A.

### 1.3.2 Conservation

Une fois la matrice obtenue, nous avons codé la fonction s(i,liste) calculant l'entropie relative en fonction d'une position i.
(entropie= 0, aucune position n'est conservée,
entropie=4.38, l'information est complètement sauvegardé)
Cette fonction est par la suite appelée par la fonction s_global_trie(liste) afin de déterminer les différentes positions qui ont un poids élevé.
Enfin, nous avons défini la fonction ai(liste) qui se charge de déterminer les 3 acides aminés les plus conservés.

Résultats :




| Position | Entropie | Acide aminé |
|---|---|---|
| 31 | 4.176827384837058 | W |
| 46 | 3.9862728301572674 | P |
| 43 | 3.875513964134912 | G |

### 1.3.3 Evaluer une nouvelle séquence

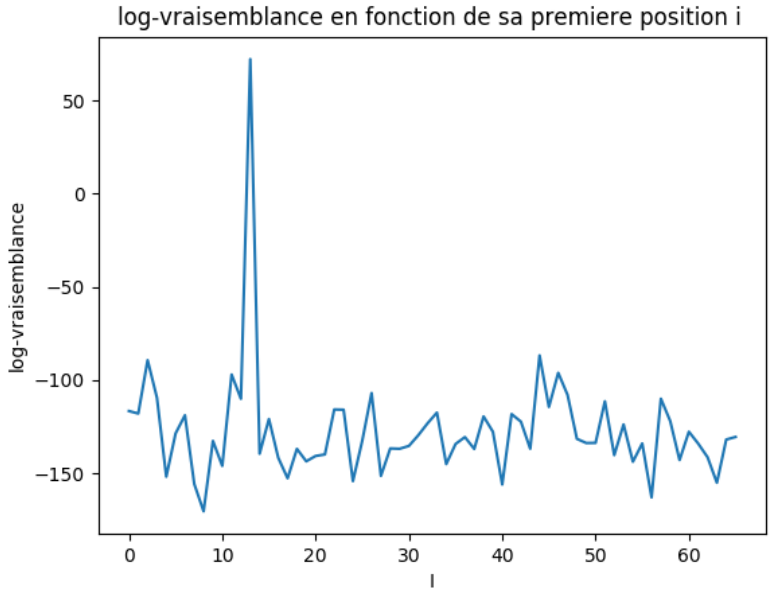
Dans un second temps, nous avons calculé la probabilité qu'une séquence b appartienne à une même famille de protéine (Dtrain) grâce à la fonction eq6 qui réalise le produit des poids pour chaque acide aminé présent dans b par rapport à la famille en question.
Pour décider si cette probabilité est "assez grande" pour dire que b appartient à la famille donnée, nous avons comparé nos résultats avec un modèle nul qui n'est pas spécifique dans les positions servant de base de comparaison.

Pour se faire, nous avons implémenté les fonctions :

- eq8 qui calcule la fréquence de chaque acide aminé de b dans la famille de protéine sans regarder la position i en réalisant la somme des poids des acides aminés de b divisé par la longueur L.
- eq7 qui calcule le produit des fréquences à l'aide de l'eq8 pour chaque acide aminé de b.
- eq9 qui compare le modèle spécifique (PSWM) avec le modèle nul en calculant le log de la vraisemblance, soit la somme des log2 des poids de chaque acide aminé de b divisé par la fréquence de chaque b sans regarder la position i.

Enfin, nous avons déterminé si il y a des sous-séquences de la famille définie par Dtrain (dans testseq) grâce à l'eqq4 qui pour chaque pour chaque première position i = 0, ...,N-L calcule le log de la vraisemblance.

Résultats :



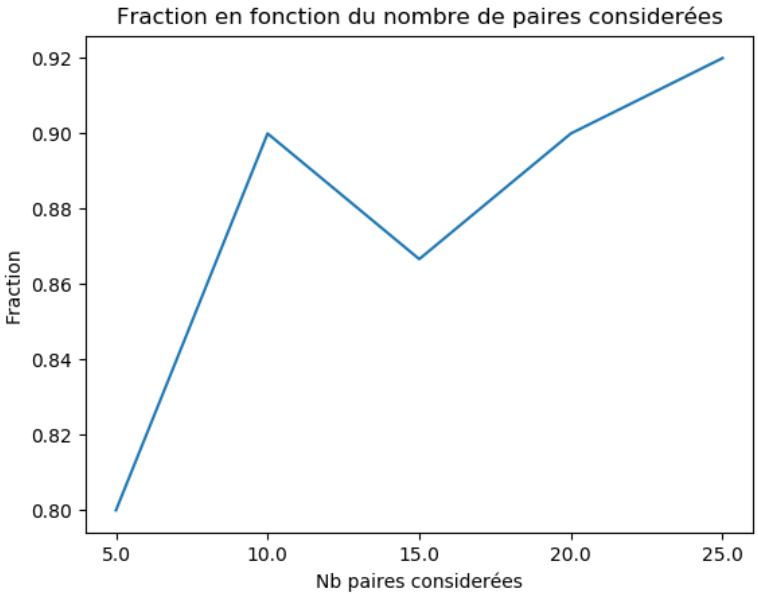

D'après nos resultats, nous observons un pic à i=13 pour un log de vraisemblance égal à 72.23472837323038. Cela nous permet d'en déduire que la séquence de position 13 à 13+L correspond à une sous séquence de la famille définie par Dtrain.

## 1.4 Coévolution de résidues en contact

Dans cette seconde partie, nous avons amélioré notre modèle en permettant la détection de corrélations entre les occurrences des acides aminés pour 2 positions données. Pour cela, nous avons défini :
- la fonction eq10 comptant le nombre de séquences ayant un acide aminé a en position i et un acide aminé b en position j,
- la fonction eq11 calculant le poids pour 2 acides aminés aux positions i et j,
- la fonction eq12 calculant l'information mutuelle (qui permet de quantifier les corrélations).
Les fonctions eq10g, eq11g et eq12g réaliseront les mêmes opérations que précédentes mais pour chaque paire de positions 0 <= i < j < L sur une matrice de Dtrain obtenu à l'aide de la fonction liste_to_mat(dtrain).
Remarque : la fonction eq10g (utilisé par les fonctions eq11g et eq12g) prend du temps à se réaliser (environ 5min).

Pour finir, la fonction func_4 trie les 50 paires de positions avec les valeurs M plus grandes puis cherche les distances associées dans le fichier distances.txt. Il calcule ensuite la fraction des paires sélectionnées qui ont une distance plus petite que 8 afin de générer un graphique de cette fraction en fonction du nombre de paires considerées.

Résultats :




Nous remarquerons ainsi d'après le graphique que plus le nombre de paires considerées est important, plus la fraction des paires qui sont des contacts est importante.

## 1.5 Conclusion

En conclusion, ce projet nous a permis de réaliser une analyse statistique d'une famille de protéines en détectant :
- les positions conservées grâce au calcul l'entropie relative en fonction de la position,
- les séquences appartenant à la même famille grâce à la comparaison du modèle spécifique (PSWM) avec le modèle nul à l'aide du log de la vraisemblance,
- les corrélations entre les différentes colonnes de l'alignement et leur relation avec les distances entre acides aminés par le calcul de l'information mutuelle.