# Conception d'un programme de threading par double programmation dynamique

## Introduction

Les protéines sont des macromolécules biologiques présentes dans toutes les cellules vivantes. Elles sont constituées d'une chaîne polypeptidique ; les acides aminés consécutifs qui les composent ont tendance à se replier en modules plus ou moins compacts, appelés domaines. En biologie structurale, un domaine est défini comme une unité structurale, fonctionnelle et évolutive des protéines qui se replie spontanément [1]. La structure tertiaire est l'arrangement dans l'espace et la succession de structures locales appelées structures secondaires. Cette structure tertiaire peut être constituée d'un ou plusieurs domaines. Le repliement d'une protéine est un processus complexe dont la bonne conformation assure la fonction biologique de la protéine.

L'étude des protéines laisse envisager que le nombre total de séquences d'acides aminés possibles est pratiquement infini. De même, le nombre de conformation 3D possible pour une protéine donnée est à priori extrêmement grand et le temps nécessaire pour trouver un repliement optimal serait de l'ordre de 10e82s [2]. Cependant, il faut relativiser car seulement une fraction de ces possibilités est observable. Un nombre assez important de séquences, environ 2e8 séquences actuellement dans genbank [3], et de structures, plus de 160 000 en ce moment dans la Protein Data Bank [4], sont néanmoins connues. Ces protéines présentent également une grande diversité dans les fonctions réalisées. Les structures protéiques ont été classées selon leur similitudes structurales et le nombre de familles de protéines ayant des repliements différents définis est étonnamment bas. Il y a environ 2000 superfamilles de repliements protéiques déterminées pour plus de 85000 entrées PDB dans la dernière version de la classification structurale SCOPe [5] (2020). C'est un argument fort en la faveur de l'hypothèse que les repliements sont beaucoup plus souvent conservés au cours du temps que les séquences.

Au vu des observations précédentes, il serait donc possible de déterminer à partir d'une séquence d'acide aminé une structure correspondante. Une approche de modélisation par homologie est réalisé depuis plusieurs années. Cela fait intervenir la notion d'homologie qui caractérise deux séquences qui ont un taux d'identité élevé et qui ont une fonction proche. Cependant, cette méthode nécessite l'existence de séquences homologues ainsi qu'une structure déjà résolu du modèle. Dans le cas contraire, une approche de modélisation par threading doit être utilisée. Cette méthode consiste à prédire la structure 3D d'une protéine cible à partir de sa séquence en acide aminé (1D) en utilisant comme modèles des structures

3D déjà résolu dans la Protein Data Bank [4]. Cette approche fut inventé par David Jones, en collaboration avec Janet Thornton et Willie Taylor [2].

## Matériels et Méthodes

Le programme développé pour mener à bien ce projet utilise miniconda3 et a été codé en Python 3.8.5. Les modules utilisés sont les suivants :

- Os
- Sys
- NumPy
- Pandas

Un environnement conda a été créé pour pouvoir faciliter la reproductibilité de notre travail. Les modules et leurs versions sont listés dans le fichier *environment.yml*. Pour recréer un environnement similaire afin de tester nos résultats, il est nécessaire d'avoir conda et d'utiliser la commande

```
conda env create --file environment.yml
```

afin de créer un environnement de travail identique à celui utilisé pour le projet. Pour plus de détails sur l'utilisation du programme veuillez vous référer au **README.md**.

Le travail a été réalisé en binôme et mis en commun sur Github. Un outil de gestion de projet a été utilisé en parallèle. Voici les liens :

Github : https://github.com/Dylkln/Projet-Prog-Threading
Trello : https://trello.com/b/74ChZb9O/threading-programmation

L'algorithme utilisé pour mener à bien le projet est un algorithme de programmation dynamique double [2, 6]. Ce dernier consiste à fixer, à une position donnée P de la structure modèle, un acide aminé A de la séquence cible. Ensuite, il est déterminé les positions des autres acides aminés par rapport à A ; de tel sorte que la position des autres acides aminés minimise le score de A à la position P. Pour réaliser cette étape, l'algorithme de Needleman & Wunsch est utilisé. Cet algorithme de programmation dynamique permet, dans le cas classique, l'alignement global de deux séquences d'acide aminés. Il a été adapté afin de pouvoir aligner la séquence d'acides aminés de la protéine cible avec la structure de la protéine modèle. De plus, cet algorithme permet de trouver l'alignement de score minimal entre la structure et la séquence. Dans notre cas, un score minimal est déterminé car on utilise des potentiels statistiques DOPE négatif. De plus, il est déterminé pour tout couple (a,p) un score minimale ; score minimal renseigné dans une matrice de haut niveau (avec a

n'importe qu'elle acide aminé de la séquence cible et p n'importe qu'elle position de la structure modèle). Ensuite, l'alignement optimal entre la séquence et la structure est déterminé en trouvant le meilleur chemin dans la matrice de haut niveau.

Un potentiel statistique sous la forme d'un fichier dope.par, contient les valeurs des interactions de Van der Walls entre deux atomes à une distance donnée par pas de 0.5 Angströms. Ce fichier a été "nettoyé" afin de ne garder que les valeurs entre carbones alphas (fichier dope-CA.par) car uniquement ces atomes sont utilisés pour déterminer l'alignement optimal.

La structure protéique utilisée est le domaine de liaison à l'ARN de la SLBP (code uniprot : 2KJM, code PDB : 2KJM). Le programme a également été testé sur les deux petites $\beta / \alpha$ protéines flavdoxin et la chemotaxis-Y (PDB codes: 4FXN et 3CHY respectivement). Dans ce cas, la séquence en acide aminé de 3CHY a été utilisé comme séquence cible. Ces deux protéines présentent un repliement équivalent mais pas de similarité au niveau de leurs séquences [6].

## Résultats

Exécuté le programme entraîne la génération de 961 matrices de bas niveau dans le cas de 2KJM (31 acides aminés a pour 31 positions p posible dans la structure). Chacune des matrices de bas niveau est remplis à l'aide de l'algorithme de Needleman & Wunsch et permettent d'obtenir le score minimal pour tout couple (a,p). Ainsi, 961 matrices de bas niveau sont générées.

La matrice de haut niveau stocke pour chaque couple (a,p) le score minimum obtenu par la matrice de bas niveau correspondante (alignement de la séquence à la structure pour a fixé à la position p). Ensuite, le parcours de la matrice de haut niveau permet de déterminer un alignement global optimal de la séquence cible sur la structure modèle.

L'alignement optimal obtenu est ensuite affiché dans le terminal sous la forme acide aminé par rapport à la position de l'atome dans la structure. Les gaps sont représentés par un "-".

## Conclusion

Le programme codé a permis la création des matrices de bas niveau grâce à la programmation dynamique simple. Ces matrices ont ensuite été utilisé afin de créer une matrice de haut niveau grâce à la programmation dynamique double. Etant donné la grande complexité d'une telle programmation ($O(n^2 \times m^2)$), nous avons réalisé les tests sur une petite protéine de 31 acides aminés.

Cependant un test sur deux plus grandes protéines faisant 129 acides aminés et 139 positions possible ce qui revient à 17 931 matrices de bas niveau à générer. Pour ce test, le code a dû tourner pendant 1 heure 20 minutes afin de nous donner le résultat.

Il peut être envisageable d'améliorer le code afin de réduire ce temps pour les futurs tests.

De plus, la programmation dynamique double est utilisé à l'origine pour aligner deux structures [6]. Les méthodes ont été adapté afin de pouvoir aligner une séquence à une structure. Cependant, les liens pour tester les programmes fournis par les articles [2 et 6] sont non fonctionnels et nous ont donc pas permis de comparer les résultats obtenus avec ceux des articles.

## Bibliographie

[1] Koonin EV, Wolf YI, Karev GP. The structure of the protein universe and genome evolution. Nature. 2002 Nov;420(6912):218–23.

[2] Jones, D.T., Taylor, W.R. & Thornton, J.M. (1992) A new approach to protein fold recognition. Nature. 358, 86-89. Jones, D.T., Miller, R.T. & Thornton, J.M. (1995) Successful protein fold recognition by optimal sequence threading validated by rigorous blind testing. Proteins. 23, 387-397. Jones, D.T. (1998) THREADER : Protein Sequence Threading by Double Dynamic Programming. (in) Computational Methods in Molecular Biology. Steven Salzberg, David Searls, and Simon Kasif, Eds. Elsevier Science. Chapter 13.

[3] Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, et al. GenBank. Nucleic Acids Res. 2013 Jan;41(Database issue):D36-42.

[4] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Res. 2000 Jan 1;28(1):235–42.

[5] Fox NK, Brenner SE, Chandonia J-M. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Res. 2014 Jan 1;42(D1):D304–9.

[6] William R. Taylor, Protein Structure Comparison Using SAP.