Rapport Projet Court

Conception d’un programme de threading par double programmation dynamique

Auteurs :

Estelle Mariaux : estelle.mariaux@hotmail.fr

Théo Ferreira : theo.ferreira.med@gmail.com

Contact : Jean-Christophe Gelly
jean-christophe.gelly@univ-paris-diderot.fr

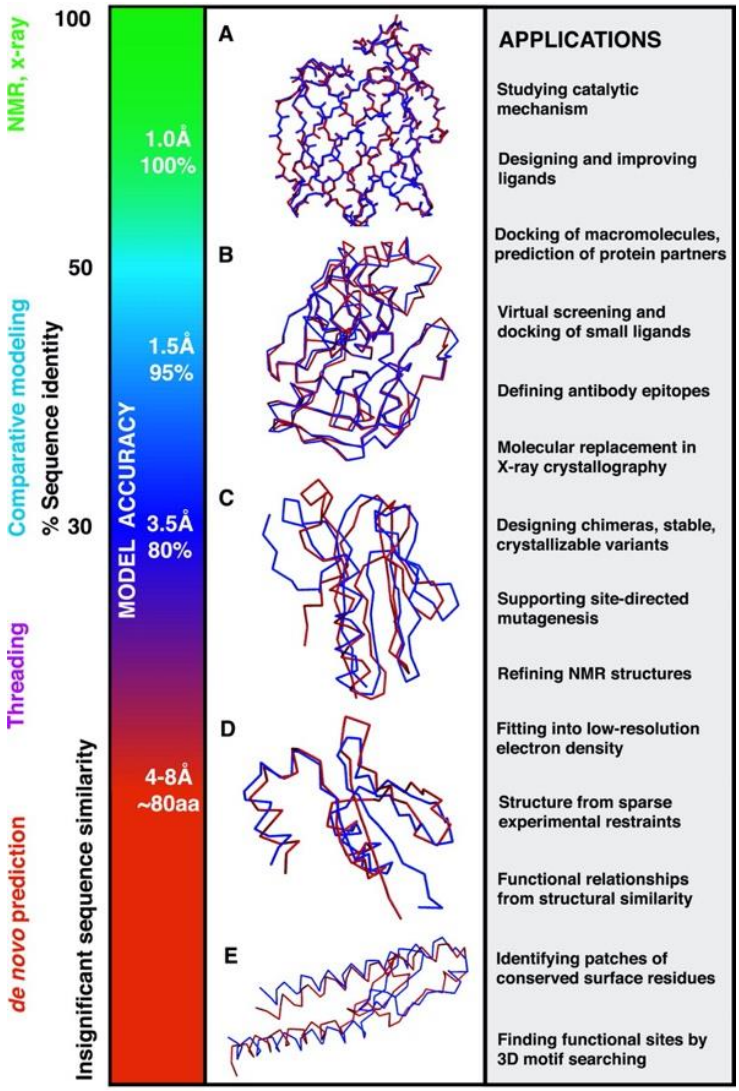
Année Universitaire : 2020 – 2021

Table des matières

Table of Contents

Introduction

Que ce soit par détermination expérimentale (*e.g.* technique de diffraction des rayons X, spectrométrie de résonance magnétique nucléaire) ou par des algorithmes de prédiction, la détermination de la structure tertiaire d'une protéine est un enjeu majeur à la compréhension de son fonctionnement.



La modélisation de cette structure tridimensionnelle peut être faite par [1] : (i) homologie ; méthode comparative qui recherche une séquence homologue dont la structure est déjà connue (*e.g.* SWISS-MODEL, CPHmodels, ESyPred3D). Cependant cette méthode est vite non performante lors d'un défaut d'alignement. C'est pourquoi (ii) la modélisation par enfilage de protéine « threading » est basée sur l'observation des structures ayant le même repliement avec peu ou pas d'identité de séquence et recherchant une compatibilité entre la structure tridimensionnelle et la séquence protéique linéaire. En résumé, c'est la recherche d'un homologue structural en essayant toutes les structures disponibles dans une base de données. Les points déterminants majeurs sont l'existence d'une bibliothèque exhaustive de repliement et l'outil pour aligner une séquence sur une structure tridimensionnelle (e.g. PHYRE2, RaptorX, iTASSER). (iii) La modélisation de structure peut aussi être faite *de novo*, c'est-à-dire uniquement par la séquence protéique (*e.g.* HMMSTR/Rosetta)

Objectif

Le projet était d'implémenter une méthode de prédiction de structure tertiaire d'une protéine basé sur le « threading » par double programmation dynamique d'après le travail de Jones D. (THREADER) [2].

Matériels & Méthodes

Environnement informatique

Le programme a été codé en python 3. Le programme est constitué de 4 fichiers (main.py, parsing.py, distance.py, alignment.py) et fonctionne sous un environnement Linux. Un environnement conda spécifique a été utilisé pour améliorer au maximum la reproductibilité et les modules à installer sont disponibles dans le fichier Threading_DPD.yml

Organisation travail en équipe (2 personnes)

Le travail a été mis en commun dans le GitHub permettant ainsi le travail en parallèle sur le même code. De plus nous avons utilisé un outil de gestion de projet en ligne (https://trello.com) permettant l'attribution des tâches et l'allocation du temps.

Gestion du code

Le code a été vérifié et corrigé selon la convention PEP 8 (via Pylint). La documentation a été générée avec Doxygen.

Structure du code

Choix d'une séquence à aligner :

Nous avons choisi une séquence protéique courte permettant de réduire les temps de calcul (ß-hairpin 1N09), le fichier est en extension .fasta. La séquence est découpée en acides aminés qui sont ensuite stockés dans un dictionnaire et le code de l'acide aminé à 1 lettre est transformé en code à 3 lettres.

Création de la banque de données

Ici, nous avons utilisé uniquement une structure unique (ß-hairpin 1N0A) pour réduire les temps de calcul mais une banque de donnée plus complète est disponible https://zhanglab.ccmb.med.umich.edu/library [3]. Les modèles de structure sont en extension .pdb. Les fichiers .pdb ont été traités pour extraire les coordonnées uniquement des atomes de carbones – alpha. Elles sont stockées dans un dictionnaire ayant comme clef les numéros des résidus liés à ces coordonnées.

Alignement séquence – structure

1 – Nous avons créé une matrice de distance qui est fixe pour une structure protéique donnée.

Celle-ci détermine la distance, selon les coordonnées disponibles dans le fichier .pdb, de tous les acides aminés entre eux (entre les carbones-alpha uniquement).

2 – Ensuite une matrice de bas niveau a été créée pour un acide aminé donné de la séquence dans une position donnée dans la structure. Un potentiel de statistique (fichier dope.par.txt http://www.dsimb.inserm.fr/~gelly/doc/dope.par) en fonction des couples d'acides aminés est associé à la distance correspondante dans la matrice de distance. Ceci constitue notre matrice de bas niveau. Pour rappel : le fichier dope.par.txt contient la valeur énergétique (potentiel statistique) pour une paire d'atomes spécifiques à une distance de bin donnée de 0,5 angströms de largeur.

3 – Dans une matrice de haut niveau sont incorporés tous les meilleurs chemins à travers chacune des matrices de bas niveau (score selon algorithme de Needlman et Wunsch) : double programmation dynamique.

Résultats

Ici, un exemple de matrice de bas niveau que renvoie le programme.

Pour un résidu de glutamine issu de la séquence à aligner en position 4 sur la structure du modèle, nous obtenons un score optimal de -5,52. Pour rappel nous cherchons le score minimum.

```
The fixed residue is GLU in position 4
[[ 0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.  ]
 [ 0.   -0.13 -0.17 -1.53   nan   nan   nan   nan   nan   nan   nan   nan]
 [ 0.   -0.13 -0.35 -1.81   nan   nan   nan   nan   nan   nan   nan   nan]
 [ 0.   -0.16 -0.35 -2.05   nan   nan   nan   nan   nan   nan   nan   nan]
 [ 0.     nan   nan   nan -2.05   nan   nan   nan   nan   nan   nan   nan]
 [ 0.     nan   nan   nan   nan -2.81 -2.81 -2.81 -2.81 -2.81 -2.81 -2.81]
 [ 0.     nan   nan   nan   nan -2.81 -3.57 -3.57 -3.57 -3.57 -3.57 -3.57]
 [ 0.     nan   nan   nan   nan -2.81 -4.08 -4.84 -4.84 -4.84 -4.84 -4.84]
 [ 0.     nan   nan   nan   nan -2.81 -4.08 -5.06 -5.06 -5.22 -5.22 -5.22]
 [ 0.     nan   nan   nan   nan -2.81 -4.08 -5.06 -5.06 -5.37 -5.37 -5.37]
 [ 0.     nan   nan   nan   nan -2.81 -4.08 -5.06 -5.06 -5.52 -5.52 -5.52]]
```

Ici, sont aussi représentés des exemples supplémentaires pour illustrer les différents scores que nous pouvons obtenir.

```
For the fixed residue is CYS in position 1, the optimized score is -3.01
For the fixed residue is CYS in position 4, the optimized score is -7.35
For the fixed residue is CYS in position 7, the optimized score is -6.56
-----
For the fixed residue is THR in position 1, the optimized score is -2.41
For the fixed residue is THR in position 4, the optimized score is -5.02
For the fixed residue is THR in position 7, the optimized score is -4.53
-----
For the fixed residue is TRP in position 1, the optimized score is -2.65
For the fixed residue is TRP in position 4, the optimized score is -4.64
For the fixed residue is TRP in position 7, the optimized score is -4.03
```

Conclusion

Au terme de ce projet nous avons réussi à implémenter jusqu'à la création d'une matrice de bas niveau et du calcul du score du chemin optimal (programmation dynamique).

Toutes les possibilités d'alignement sont calculées, or nous aurions pu ne pas tenir compte des alignements obtenant un score trop élevé pour être envisagés [4]. De plus, notre séquence avait une longueur de 10 acides aminés et la structure avait 11 positions. Nous n'avons pas pris en compte les gaps dans l'alignement, ce qui amène *in fine* à un alignement incomplet.

Pour finir, nous n'avons pas pu créer les matrices de haut niveau.

BIBLIOGRAPHIE

1. Baker, D. & Sali, A. Protein Structure Prediction and Structural Genomics. *Science* **294**, 93–96 (2001).

2. Jones, D. THREADER: protein sequence threading by double dynamic programming. in vol. 32 285–311 (Elsevier, 1998).

3. Yang, J. *et al.* The I-TASSER Suite: protein structure and function prediction. *Nat Methods* **12**, 7–8 (2015).

4. Lathrop, R. H. & Smith, T. F. Global Optimum Protein Threading with Gapped Alignment and Empirical Pair Score Functions. *J Mol Biol* **255**, 641–665 (1996).