Rapport Projet Court :

Conception d’un programme de threading par double programmation dynamique

Auteurs :

Estelle Mariaux : [estelle.mariaux@hotmail.fr](mailto:estelle.mariaux@hotmail.fr)

Théo Ferreira : [theo.ferreira.med@gmail.com](mailto:theo.ferreira.med@gmail.com)

Contact :

Jean-Christophe Gelly : [jean-christophe.gelly@univ-paris-diderot.fr](mailto:jean-christophe.gelly@univ-paris-diderot.fr)

Année Universitaire : 2020 – 2021

Table des matières

**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.

A screenshot of a cell phone

Description automatically generated(Img : 10.1126/science.1065659)

La modélisation de cette structure tridimensionnelle peut être faite par : (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 basé 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. Il existe points déterminants majeurs ; l’existence d’une bibliothèque exhaustive de repliement et un outil précis 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) (Jones 1998).

**Matériels & Méthodes**

**Environnement informatique**

Le programme a été codé en python 3, le programme est constitué de X fichiers (main.py, xx.py, xx.py) et fonctionne sous une environnement Linux. Un environnement conda spécifique a été utilisé pour améliorer au maximum la reproductibilité et les modules à installer son disponible dans le fichier XXX.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 de temps.

**Gestion du code**

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

**Structure du code**

Choix d’une séquence à aligner :

Choix d’une séquence protéique courte permettant de réduire les temps de calcul (ß-hairpin XX), 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 (n°XX) pour réduire les temps de calcul mais une banque de donnée plus complète est disponible <https://zhanglab.ccmb.med.umich.edu/library> (iTASSER). 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 résidus liés.

**Création matrice de distance pour une structure** : distance entre aa

**Calcul energie en fonction de la distance**

* Dope potentiel energie explication de 0,5 en 0,5 jusqu’à environ 15 … : only CA-CA dict res-res clef renvoyant au 30 enrgies

Matrice de bas niveau : on somme les scores trajet optimal, algorithme de Needleman-Wunsch – programmation dynamique 1

Incorporation matrice de haut niveau – programmation dynamique 2

A screenshot of text

Description automatically generated

**Résultats :**

**Pour une seq de x AA (nom) alignée sur structure (nom) on obtient x matrice avec score pour le chemin max**

**Conclusion :**

Same sequence fragments can adopt different structures depending on the environment.

Def programmation dynamique

En [informatique](https://fr.wikipedia.org/wiki/Informatique), la **programmation dynamique** est une méthode [algorithmique](https://fr.wikipedia.org/wiki/Algorithme) pour résoudre des [problèmes d'optimisation](https://fr.wikipedia.org/wiki/Optimisation_(math%C3%A9matiques)). Le concept a été introduit au début des années 1950 par [Richard Bellman](https://fr.wikipedia.org/wiki/Richard_Bellman)[1](https://fr.wikipedia.org/wiki/Programmation_dynamique#cite_note-Cormen359-1). À l'époque, le terme « programmation » signifie planification et ordonnancement[1](https://fr.wikipedia.org/wiki/Programmation_dynamique#cite_note-Cormen359-1). La programmation dynamique consiste à résoudre un problème en le décomposant en sous-problèmes, puis à résoudre les sous-problèmes, des plus petits aux plus grands en stockant les résultats intermédiaires

= une méthode d’optimisation des processus de décisions séquentielles. S’appuie sur l’algorithme de Bellman