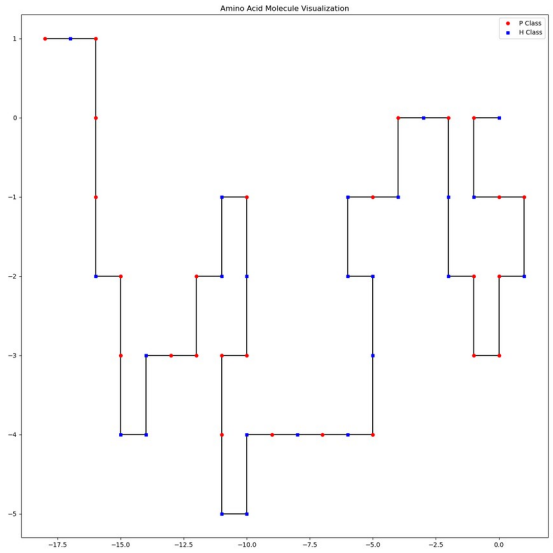
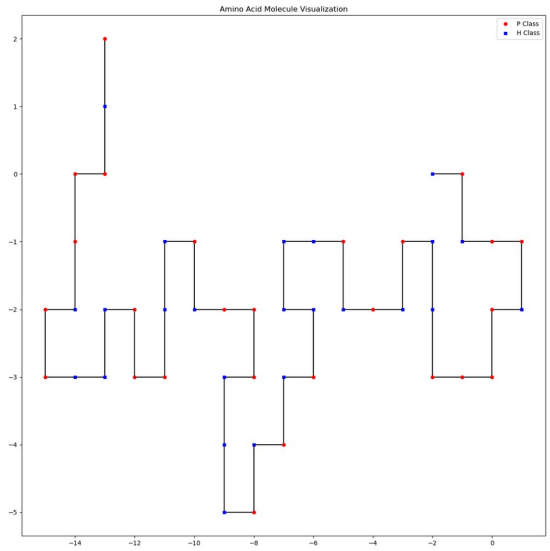
M2 BioInformatique
BI
Année 2023/2024

Compte rendu de projet :

Repliement de protéine ab initio selon la méthode de Monte Carlo





SALAUN Nicolas
N°Etudiant 22200082

lien du projet GitHub :
https://github.com/n-salaun/SALAUN_monte_carlo_protein_folding

Introduction

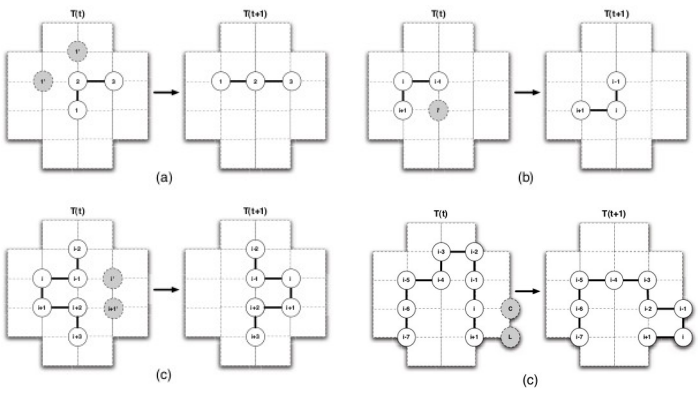
Les protéines, composées d'acides aminés, sont des éléments fondamentaux de la biologie. Elles jouent un rôle essentiel dans la structure, la fonction et la régulation du vivant, à travers de multiples fonctions. La structure d'une protéine est intimement liée à sa fonction et sa compréhension est donc cruciale pour élucider leurs fonctions biologiques, pour comprendre leurs interactions et concevoir par exemple des médicaments ciblés. Il existe de nombreuses méthodes expérimentales permettant de résoudre ces conformations, tel que la RMN, ou encore la cristallographie à rayon X. Cependant, ces méthodes possèdent de nombreuses limitations, telles la rigidité pour la cristallographie, ou encore la taille de la séquence pour la RMN, sans parler du coût de ces deux méthodes. C'est pourquoi la modélisation et la prédiction du repliement des protéines sont devenues des domaines de recherche majeurs en biologie structurale et en bioinformatique.

Il existe de nombreuses techniques de modélisations informatique, parmi lesquelles nous pouvons retrouver la modélisation par homologie, la dynamique moléculaire ou encore la modélisation ab initio. Chacune de ces méthodes ont leurs avantages et leurs inconvénients, la modélisation par homologie par exemple, permet de replier efficacement une protéine selon une conformation connue d’une séquence similaire, ce qui implique que cette méthode ne convient pas aux protéines ayant peu d’homologues connus.

Dans le cas de la modélisation ab initio, elle se distingue par sa capacité a se détacher de possibles conformations préexistantes, et de prédire la structure d'une protéine à partir de sa séquence d'acides aminés et des principes physico-chimiques guidant le repliement des protéines. Cependant, elle comporte également des avantages et des inconvénients significatifs. En effet, son applicabilité universelle, et sa capacité à explorer des structures originales, se solde par une complexité calculatoire, et une demande en ressource informatique considérable.

Le but de ce projet est de tenter de reproduire en partie des résultats obtenus dans l’article scientifique de Thachuk, Shmygelska et Hoos, intitulé "A replica exchange Monte Carlo algorithm for protein folding in the HP model,". Cette recherche vise à développer une méthode de modélisation informatique du repliement des protéines en utilisant le modèle HP, une approche simplifiée pour représenter les protéines selon deux statuts, hydrophobes, ou polaires. Un tel modèle permet d’ignorer les détails complexes induits par les interactions des différents acides aminés entre eux, et permet de réduire la complexité du problème tout en proposant une solution cohérente.

L'article de Thachuk et al. explore comment utiliser un algorithme de Monte Carlo avec réplica pour échantillonner efficacement ces différents mouvements et rechercher la conformation la plus stable de protéines dans le modèle HP. Cette approche algorithmique permet d'explorer l'espace des conformations de manière aléatoire, mais plus efficace. Les auteurs ont déterminé quatre mouvements possibles, le *end move,* le *corner move*, le *crankshaft,* et le *pull move*.



Matériel et Méthode :

Pour ce projet, nous avons réalisé le programme sur le langage de programmation python, avec les librairies hors base de python étant matplotlib, argparse et tqdm.

Notre approche pour la réalisation du code a été similaire à une approche bottom-up classique. Nous nous sommes tout d'abord concentrés sur la lecture d'un fichier FASTA afin de le transformer en une séquence HP utilisable par notre programme. Nous avons ensuite entamé la création de la classe principale regroupant les acides aminés, avec leur type (Hydrophobe ou Polaire), et leurs coordonnées.

Une fois nos acides aminés créés, il a fallu les initialiser, en leur donnant une conformation de départ. Pour cela, nous avons opté pour deux méthodes, une initialisation linéaire, plaçant les acides aminés sur une même ligne, et une méthode aléatoire. Cette dernière doit être préférée, car elle permet d'avoir une conformation de base permettant un repliement plus efficace, en permettant plus de mouvements.

Nous avons ensuite procédé à la création de différentes fonctions générales, permettant par exemple de visualiser les acides aminés. Ces fonctions ont été réalisées en amont des mouvements, afin de pouvoir corriger après visualisation les potentielles erreurs implémentées par les mouvements.

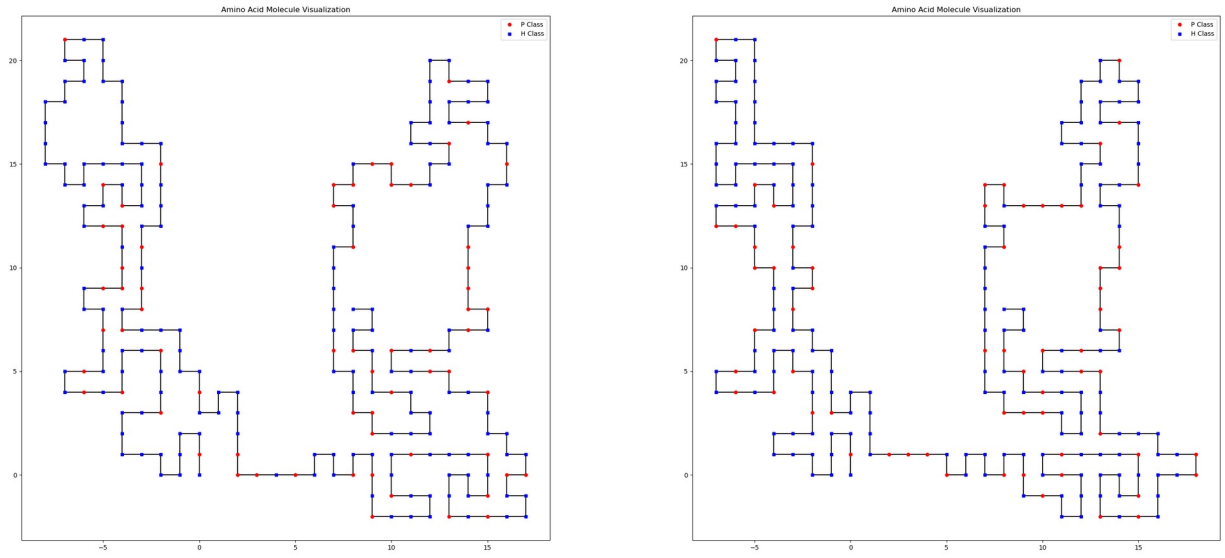
Cette étape ainsi réalisée, nous nous sommes donc intéressés à la création des différents mouvements. Nous avons pu implémenter les trois mouvements de VSHD (end, corner et crankshaft), mais l'implémentation du pull move comporte quelques erreurs critiques avec une réussite de mouvement sporadique, et nous avons préféré ne pas l'utiliser dans le rendu final du projet.

La dernière étape algorithmique a été de réaliser les choix de mouvements. Pour cela, le programme choisit parmi tous les acides aminés une instance d'un acide aminé, et l'on vérifie tous les mouvements possibles que peut faire ce dernier. Si aucun mouvement n'est possible, nous procédons au tirage aléatoire d'un autre acide aminé, mais si un ou plusieurs choix est possible, nous en sélectionnant un aléatoirement et l'appliquons. Si la nouvelle conformation a une énergie plus faible que la précédente, le mouvement est accepté. Une option a été créée afin de pouvoir choisir une température, permettant de sortir la molécule d'un puits local d'énergie en acceptant des mouvements légèrement défavorables, cependant cette option semble accepter des mouvements trop défavorables, et ne devrait pas être utilisée.

Enfin, les implémentations de qualité de vie ont été implémentées telles que le parseur d'arguments argparse, afin d'offrir différentes options à l'utilisateur tel que le nombre d'itération, le fichier FASTA d'entrée, ou le mode d'initialisation.

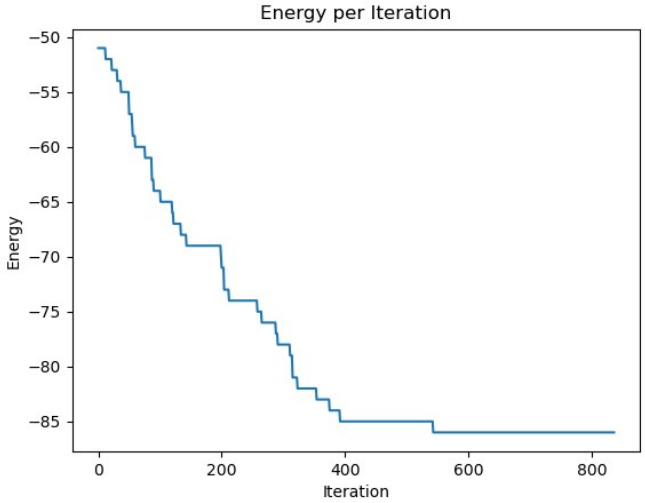
Résultats:

Nous pouvons ici voir un exemple de repliement de protéine, pour une protéine d'environ 230 acides aminés, avec 10 000 itérations pour un temps de calcul de cinq minutes.



Repliement d'une protéine de 230 aa, a gauche la position de départ, a droite celle optimisée

Ici, l'énergie de départ est de -50, et après repliement de la protéine avec le programme, nous obtenons une énergie de -86. Cette implémentation est sans recuit simulé, et est sans doute situé dans un puits d'énergie. Le graphique représentatif de l'énergie de ce repliement est le suivant.



Graphique représentatif de l'énergie de la protéine en fonction du nombre d'itérations (divisé par 10)

Conclusion :

Lors de ce projet, nous avons réalisé un programme fonctionnel de repliement de protéine selon l'article scientifique de Thachuk, Shmygelska et Hoos, "A replica exchange Monte Carlo algorithm for protein folding in the HP model,". Le programme possède certaines limites et des améliorations potentielles sont à envisager, mais le repliement effectif de la protéine est bien présent.

Parmi les améliorations possibles, nous pouvons bien évidemment parler du pull move, qui est, selon les auteurs à lui seul suffisant pour replier une protéine efficacement. En effet, ce mouvement permet d'améliorer la variabilité des mouvements.

Une autre amélioration notable serait d'implémenter correctement le recuit simulé, afin de pouvoir sortir de possibles puits d'énergie, et d'ainsi atteindre une conformation idéale en acceptant des mouvements légèrement défavorables.

Une implémentation moins nécessaire, mais intéressante serait de pouvoir observer le repliement de la protéine à travers un film de chaque frame après un mouvement. Cette implémentation a été essayée avec Animations de matplotlib, mais sans succès.

En conclusion, ce projet, malgré des limitations et des améliorations notables, nous a permis d'obtenir le repliement de protéine selon une méthode ab initio suivant la méthode de Monte Carlo.