Sophie LEMATRE
M2 Bioinformatique

Repliement d'un modèle simplifié de protéine par un algorithme de Monte-Carlo et échange de répliques

Introduction

Le repliement des protéines en trois dimensions est aujourd'hui encore un problème. En effet, les méthodes expérimentales pour déterminer leur structure comme la cristallographie ou la résonance magnétique nucléaire restent assez longues et laborieuses. C'est pourquoi, il y a un réel besoin de programme de prédiction à partir de la séquence. Cependant, même simplifié à l'aide du modèle HP, c'est à dire que la séquence de la protéine ne sera composée que de h pour hydrophobe ou p pour polaire, et de l'utilisation d'un réseau pour la conformation de la protéine, le problème reste NP-difficile. C'est pourquoi les algorithmes stochastiques sont utilisés tels que ACO-HPPFP-3 utilisant la méthode ACO (Ant Colony Optimisation) ou PERM (pruned enriched Rosenbluth method) se basant sur la méthode de Monte Carlo. De plus, plusieurs autres algorithmes se sont basés sur la méthode de Monte Carlo, dont REMC (Replica Exchange Monte Carlo) qui permet de réaliser le travail sur un ensemble de réplique de conformation, à des températures différentes, sur un réseau, avec le modèle HP en deux dimension, modèle qui est implémenté ici.

Matériel et méthodes

Les mouvements

Afin d'essayer différentes conformations, les résidus de la protéine peuvent être déplacé selon deux grandes catégories de mouvements: VSHD et pull move, qui représentent une perturbation locale de la conformation.

La première catégorie de mouvement peut être divisée en trois sous-catégories: end-move, corner move et crankshaft move. Le premier permet de bouger un résidu en extrémité de chaîne autour du résidu connecté si des positions sont libres.
Le second permet, à partir d'une configuration en coude, de bouger le résidu formant l'angle en diagonal, permettant de faire tourner le coude de 90°. Le dernier mouvement permet quant à lui de bouger 2 résidus en même temps, en faisant pivoter une conformation en U de 180° si les positions sont libres.

La seconde catégorie de mouvement permet aussi de déplacer deux résidus. Si i-1, i et i+1 sont alignés et i+2 forme un coude, alors les résidus i et i+1 vont pouvoir se déplacer en diagonal sans que la chaîne ne soit brisée.

Le modèle HP

Le modèle hydrophobique-polaire permet de classer les acides aminés en deux classes, hydrophile ou polaire. La séquence n'est alors composée que de 'H' ou 'P'.

Calcul de l'énergie

L'énergie d'une conformation est calculée suivant le nombre de contact 'HH' topologique présent. C'est à dire, deux résidus 'H' l'un à côté de l'autre n'étant pas adjacent dans la séquence. Pour chacun de ces contacts, -1 est ajouté à l'énergie totale. Ainsi, plus il y a de contact 'HH' plus l'énergie est faible. De ce fait, il existe plus d'une conformation ayant l'énergie minimale qu'il est possible d'atteindre en utilisant un réseau.

Libraries

Le programme a été implémenté en python 3 orienté objet avec une classe *residu* contenant les coordonnées de ce dernier, son hydrophobicité, le résidu précédent et le suivant. Quant aux mouvements, une classe mère comporte les méthodes pour effectuer les changements de conformation tandis que les classes filles, étant chaque mouvement, comporte chacune sa propre méthode pour évaluer la possibilité de faire le mouvement et permet d'indiquer les positions auxquelles doivent se déplacer les résidus. De plus, une matrice permet de visualiser les conformations et de trouver les voisins libres de chaque résidu.

Pour se faire, plusieurs modules ont été nécessaire. Les modules *docopt 0.6.2* et *matplotlib 2.2.3* afin de traiter les arguments donnés par l'utilisateur et de créer une visualisation en trois dimension pour la structure de la protéine.

Monte Carlo

La méthode utilisée ici est principalement basé sur la méthode de Monte Carlo, qui est une méthode stochastique, c'est à dire qu'elle est basée sur le principe de l'aléatoire dépendant du temps. Un résidu est d'abord choisi aléatoirement. Si il se trouve en extrémité de chaîne alors le mouvement end move est appliqué. Sinon, un chiffre est tiré aléatoirement dans une loi uniforme de 0 à 1. Si le chiffre tiré est inférieur à 0.4, paramètre donné dans l'article permettant de ne pas perdre de performance, alors le mouvement pull move est appliqué. Sinon, un second chiffre est tiré aléatoirement, permettant de déterminer si le mouvement sera un corner move ou un crankshaft move suivant une probabilité de 0.5. Si au cours de ce mouvement, l'énergie obtenue est inférieure à la conformation précédente alors la conformation est gardée. Sinon, suivant une probabilité de transition, la conformation peut tout de même être gardée.

$$P = e^{\frac{-\Delta E}{T}}$$

Cette particularité permet de ne pas rester bloquer sur une solution qui pourrait être optimale localement mais de rechercher une solution qui pourrait être optimale le plus globalement possible.

Cette méthode va être appliquée sur une liste de réplicats de conformation, avec chacun une température différente. En effet, plus la température est élevée, plus il est possible d'accepter des solutions défavorables.

De plus, les réplicats sont de temps à autre changer d'environnement. En effet, chaque indice de la liste est associée à une température. De ce fait, 2 conformations à des indices voisins peuvent interchanger d'indice et donc de température, suivant la probabilité suivante:

$$P = e^{-(\frac{1}{Tj} - \frac{1}{Ti}) * (E(ci) - E(cj))}$$

Cette méthode est répétée jusqu'à ce qu'une conformation ait atteinte l'énergie minimale demandée ou lorsque le nombre d'étapes maximum est atteint, nombre étant 500 par défaut.

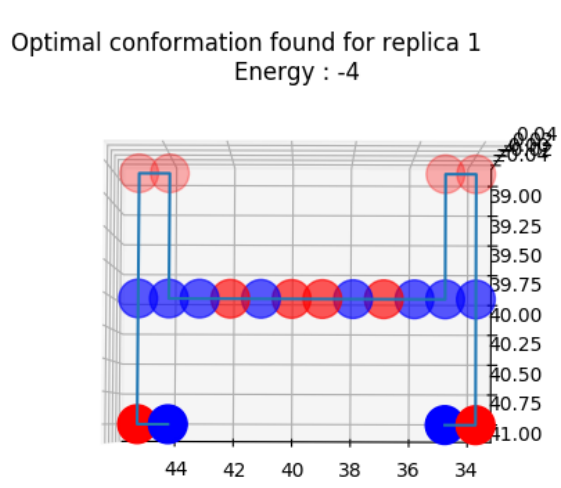
Visualisation

Les conformations obtenues pour chaque réplicat sont ensuite représentées sur matplotlib 3D et sauvegardées dans un dossier. Les résidus hydrophiles sont représentés en bleu tandis que les polaires sont en rouge. L'énergie la plus faible trouvée s'affiche sur le terminal avec son temps d'exécution.

Résultats

Pour ce résultat, le benchmark standard des séquences avec le modèle HP a été utilisé, notamment la première séquence "$(HP)_2PH_2PHP_2HPH_2P_2HPH$" avec un total de 20 résidus.

Le programme a été lancé suivant deux modes: VSHD et MIXE. Pour le VSHD, seul les mouvements de cette catégorie ont été utilisés tandis que MIXE utilise les mouvements de VSHD et de pull move. L'énergie minimale attendue était de -9 et le nombre d'étapes maximum a été fixé à 500. Les conformations sont représentées sur la figure 1. L'énergie atteinte est plus élevée que celle attendue.



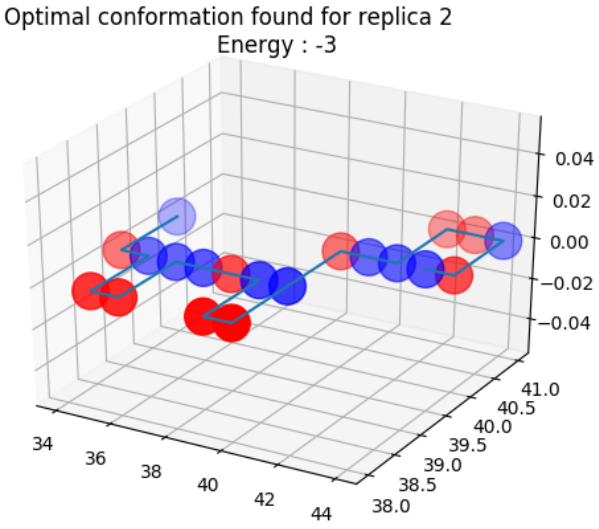


Fig. 1a Conformation du réplica 1 avec le mode VSHD: conformation ayant la plus faible énergie, -4, à la fin des 500 étapes. Les sphères rouges représentent les résidus polaires et les bleues représentent les hydrophobes.

Fig. 1b Conformation du réplica 2 avec le mode MIXE: conformation ayant la plus faible énergie, -3, à la fin des 500 étapes. Les sphères rouges représentent les résidus polaires et les bleues représentent les hydrophobes.

Avec ce benchmark standard, des conformations possibles ont déjà été trouvées (figure 2). Il est possible de remarquer, que la conformation trouvée par les mouvements VSHD ont permis d'avoir les extrémités de la protéine identique à la solution optimale.

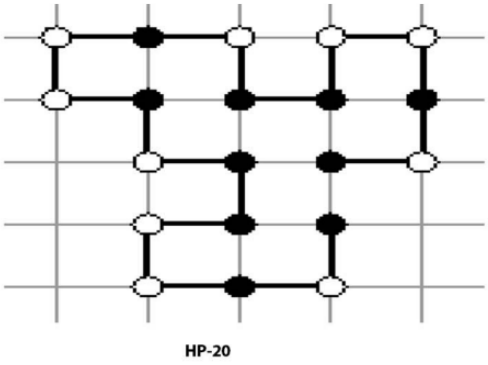


Fig. 2 Conformation optimale possible pour la séquence étudiée: Cette conformation permet d'avoir une énergie totale de -9 [1].

Cependant, le centre de la protéine est restée alignée, ce qui ne permet pas de contact 'HH' et donc une augmentation de l'énergie.

Concernant la conformation obtenue avec les mouvements mixe, les extrémités de la protéine ont une configuration différente mais le centre de la protéine a essayé de trouver une conformation plus adéquate permettant de diminuer son énergie.

Discussion

Ainsi, il est possible d'observer que les conformations trouvées sont différentes chacune des autres. Cela s'explique notamment la méthode aléatoire de Monte Carlo.

Il n'a pas été possible de trouver la conformation optimale ou tout du moins l'énergie minimale attendue. En effet, plusieurs paramètres peuvent être en cause de cette divergence. Tout d'abord la probabilité pour choisir entre le mouvement corner move et crankshaft, qui peut être modifiée au vu qu'aucune indication n'a été donnée.

De plus, le mouvement pull move a été implémenté dans les conditions les plus simples, mais ce mouvement peut s'élargir et déplacer plus que deux résidus. En effet, la protéine doit être capable, après un pull move, de se déplacer jusqu'à trouver une condition adéquate. Ce qui fait, qu'une conformation ne permettant pas de faire de pull move pour le moment pourrait s'y prêter.

Conclusion

Le programme permet de mettre en place la méthode de Monte Carlo, et donc de déterminer au mieux possible une solution optimale en deux dimensions. Cependant, le mouvement pull move est implémenté dans sa forme la plus simple et il serait bien de la complexifier. Néanmoins, les classes mouvements permettent d'intégrer facilement des nouveaux mouvements en créant pour chaque une nouvelle classe. Donc il est facilement possible de faire évoluer ce programme en l'agrémentant de nouveaux mouvements.

Bibliographie

1. Cox GA, Mortimer-Jones TV, Taylor RP, Johnston RL. Development and optimisation of a novel genetic algorithm for studying model protein folding. Theor Chem Acc. 2004;112. doi:10.1007/s00214-004-0601-4

Annexes

Le programme est orienté objet (figure S1) avec l'objet *residu* qui contient les coordonnées du résidu, son hydrophobicité et les résidus suivant et précédent.

La classe *movement* est la classe mère de toutes les autres classes mouvements, et contient les méthodes pour faire les changements de conformation et totaliser l'énergie des nouvelles conformations. Pour chaque mouvement est créée une classe fille, ayant comme méthode *mutation()* permettant de vérifier que la conformation de la protéine se prête au mouvement et si c'est le cas, de trouver les positions libres auxquelles le ou les résidus doivent se déplacer et s'il y a plusieurs possibilités, elle en choisit une aléatoirement.

Cette configuration permet d'ajouter facilement de nouveaux mouvements en créant une classe fille de la classe *movement* et en lui intégrant la méthode *mutation()* cherchant si le mouvement peut s'exécuter.

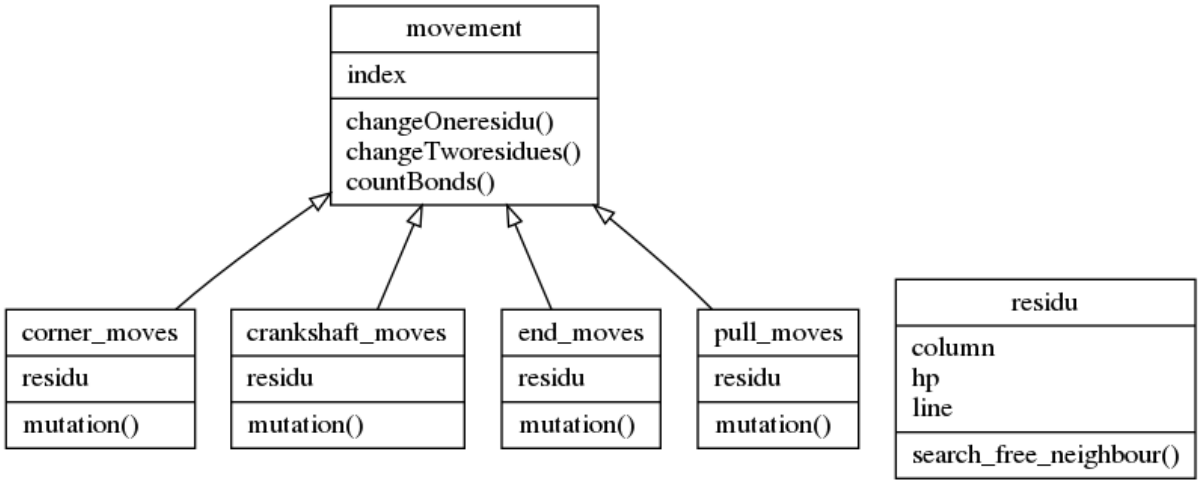


Fig. S1 Schéma UML simplifié: À la classe *residu* est ensuite ajouté un pointeur vers le résidu suivant et précédent. Il est assez facile d'implémenter de nouveaux mouvements.

Le programme suit le principe de la méthode de Replica Exchange Monte Carlo (figure S2) en intégrant une liste de réplicats ayant chacun une température différente. La liste contient un ensemble de dictionnaire recensant la liste d'objet *residu*, le réseau *structure_grid* étant une liste de liste initialisée à -1 et remplie avec les indices des résidus, la température et l'énergie associée à la conformation.

Ainsi, lorsqu'il y a un échange de température, ce sont la liste de *residu*, le réseau et l'énergie associée qui vont être interchangés car la température est associée à un indice de la liste et ne doit donc pas être interchangée.

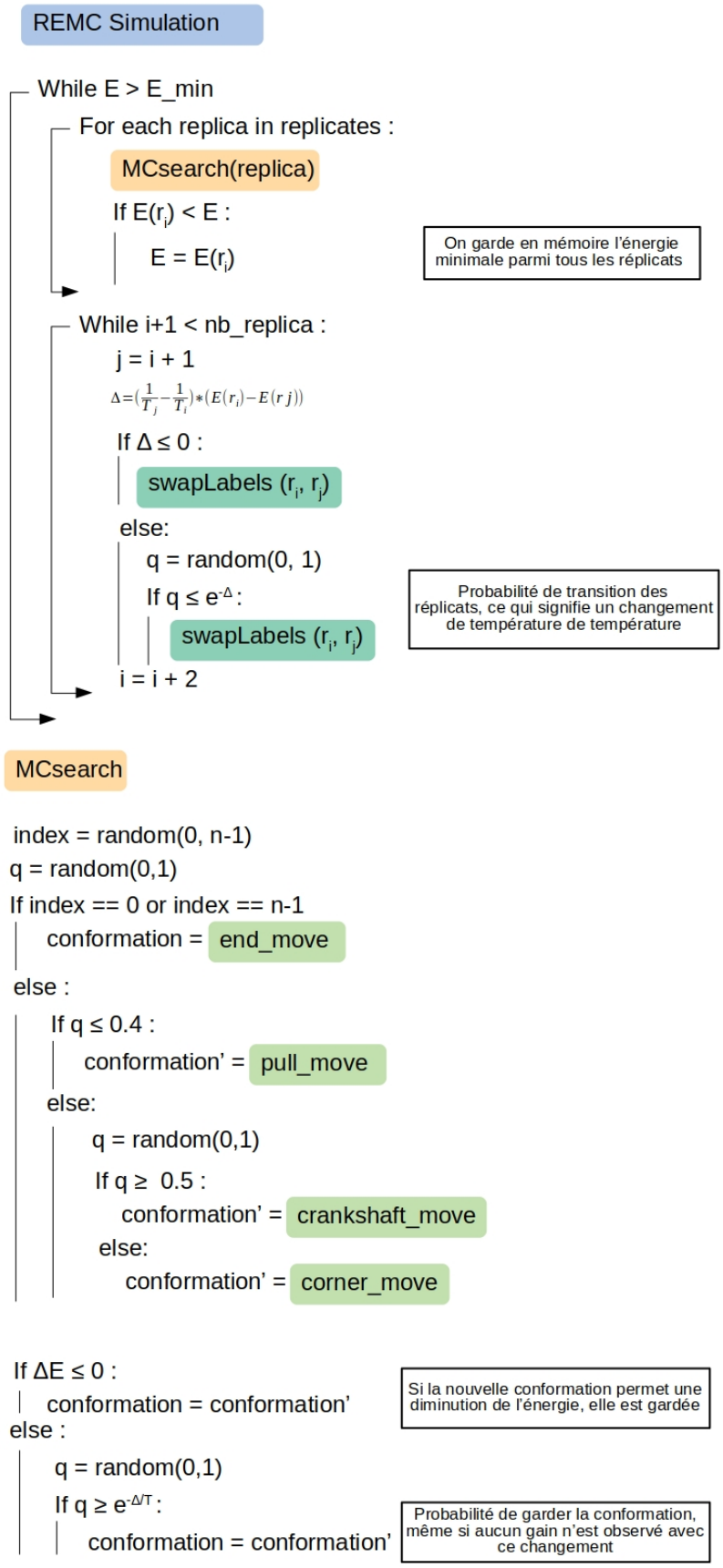


Fig. S2 Schéma de la méthode de REMC implémentée: Pseudo-code pour le REMC et la méthode de Monte Carlo. E_min est l'énergie minimale attendue, conformation' est la nouvelle conformation obtenue

Concernant les mouvements, les méthodes reposent principalement sur des conditions. En effet, il est nécessaire de vérifier que la protéine soit dans la bonne conformation pour effectuer le mouvement selon le résidu sélectionné et que les positions nécessaires soient libres pour effectuer le mouvement.

De ce fait, les mouvements *crankshaft* et *pull move* nécessitent une longue liste de conditions qui été optimisées dans la limite du possible pour qu'elles restent compréhensibles. En effet, la conformation attendue peut être horizontale ou verticale, de ce fait une boucle a été implémentée afin de pouvoir changer les attributs colonne et ligne facilement sans avoir redondance.

Pour *pull move*, les résidus impliqués peuvent être i-1, i, i+1 et i+2 ou i-2, i-1, i et i+1, i étant l'indice du résidu choisi aléatoirement. Il est d'abord nécessaire de regarder si trois résidus sont alignés. Si tel est le cas, et qu'ils sont adjacents sur la séquence il faut regarder s'il y a un résidu adjacent sur la séquence, qui permet de former un coude avec *__find_ngh_occupied()*. Il faut donc regarder aux deux extrémités de la séquence pour savoir où se situe le coude. Une fois fait et trouvé, il faut vérifier que les deux positions suivies en-dessous de ce coude soient libres pour pouvoir effectuer le déplacement.

Pour *crankshaft*, les résidus impliqués peuvent être i-2, i-1, i et i+1 ou i-1, i, i+1 et i+2, ce qui augmente le nombre de cas à regarder. La conformation en U recherchée trouvée, il est nécessaire de regarder si les voisins en face sont libres afin de pouvoir effectuer le déplacement.

Ces deux mouvements ont été les plus difficiles à implémenter en raison des différents cas et conditions possibles. En effet, il est important qu'il ne puisse pas y avoir d'exception, sinon la protéine ne suit plus un réseau et des liaisons se croisent ou sont en diagonales.

Concernant les mouvements pull move, d'après l'article il est possible de lancer le programme avec ce seul mouvement. Cependant, la protéine est linéaire à l'origine, ce qui fait qu'elle ne peut être en condition pour effectuer un pull move. C'est pourquoi, il n'a été présenté que des résultats avec les mouvements VSHD ou mixe.

Le tout est ensuite visualisé sur matplotlib 3D, permettant de tourner autour de la conformation.

Le programme doit être lancé en ligne de commandes avec plusieurs arguments obligatoires et certains arguments optionnels qui prennent une valeur par défaut.

./main.py -s <seq> -e <x> -m <move>
./main.py -s <seq> -e <x> -m <move> [-p <nb_steps>] [-t <temp>] [-r <nb_rep>]

./main.py -s hphpphhphpphphhpphph -e 9 -m MIXE

La séquence doit être rentrée en ligne de commande, suivant le modèle hydrophobic-polar. Une énergie minimale à atteindre est attendue, ainsi que le mouvement désiré.

Le nombre d'étapes maximum est par défaut à 500, la température minimale est elle à 160 et le nombre de réplicats à 5. Mais il est possible de les changer si l'utilisateur le désire en ligne de commande.

La sortie est la suivante (figure S3), avec la plus basse énergie trouvée, le temps d'exécution et un rappel de la légende pour les figures qui créées par matplotlib.

```
****************************************
  The best energy found is -3
****************************************
Execution time : 1.84 sec


Hydrophobic amino acid are blue
Polar amino acid are red
```

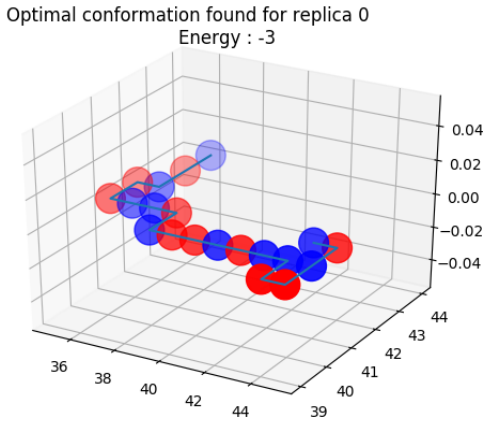


Fig. S3 Exemple de sortie du programme avec mouvement MIXE: À gauche est la sortie du terminal, donnant la plus basse énergie trouvée ainsi que le temps d'exécution. À droite, se trouve les images créées par matplotlib représentant les conformations pour chacun des réplicats.