INF442 – Projet pour la promo X2014 – sujet INF442-6

Un modèle simplifié de repliement des protéines

difficulté ***

contact: Philippe.Chassignet@polytechnique.edu

1 Préambule

Sur la page web qui accompagne ce sujet[1], on trouvera des pointeurs vers des informations détaillées sur la structure des protéines. Les quelques lignes d'explication qui suivent, suffisent cependant pour traiter le sujet.

Une protéine est une chaîne polymère d'acides aminés. Cette chaîne comporte de nombreuses libertés d'orientation et elle se replie pour donner une forme relativement compacte qui détermine en particulier sa fonction biologique. La figure 1 montre l'exemple d'une toute petite protéine qui est formée de 28 acides aminés et 479 atomes. Dans l'illustration de droite, le trait en gras passe par le premier atome de carbone de chaque acide aminé (dit carbone α), ce qui donne une idée du repliement.

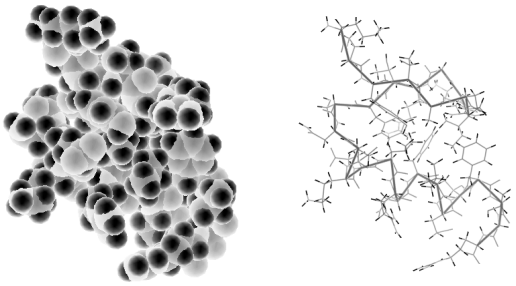


FIGURE 1 – Une petite protéine

Les liaisons covalentes déterminent la structure moléculaire, mais la forme globale d'une protéine résulte principalement des effets hydrophobes. Dans un milieu aqueux, le repliement tend a regrouper les acides aminés les plus hydrophobes (radicaux hydrocarbonés) à l'intérieur de la protéine et à placer les moins hydrophobes (radicaux polaires) en périphérie. C'est un enjeu important de la bioinformatique de pouvoir calculer la forme d'une protéine à partir de la séquence de ses acides aminés. Pour ce projet, nous allons nous placer dans un contexte très simplifié, le modèle HP, 2D et à maille carrée, connu aussi comme modèle de Dill [2].

2 Détail du sujet

Dans le modèle HP, les divers types d'acides aminés sont rangés en deux classes, selon la nature de leur radical, hydrophobe `H` ou polaire `P`. Une protéine composée de n acides aminés est ainsi décrite par une séquence $a_0\, a_1 \ldots\, a_{n-1}$, avec $a_i \in \{\mathtt{H}, \mathtt{P}\}\ \forall\, i \in \{0\,,\ \ldots\ ,\, n-1\}$. De plus, on considère que les acides aminés sont de même taille et sont contraints à se placer aux nœuds d'un réseau 2D à maille carrée. Dans les figures, comme l'exemple de la figure 2, les hydrophobes

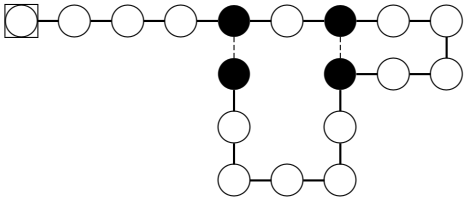
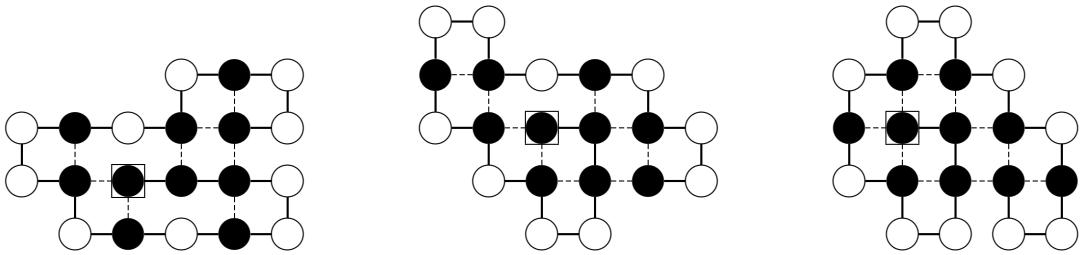


FIGURE 2 – Exemple de repliement 2D à maille carré pour la séquence `PPPPHPHPPPPHPPPHPH`

`H` sont marqués par des ● et les polaires `P` sont marqués par des ○. Le premier acide aminé de la séquence est entouré d'un carré pour le distinguer et le trait plein donne l'ordre de la séquence.

La mise en contact entre deux hydrophobes se traduit par une baisse d'énergie potentielle, alors que l'on néglige la variation d'énergie pour un contact qui implique au moins un polaire. Dans le modèle de Dill, la minimisation de l'énergie revient donc à maximiser le nombre des contacts, dans les directions du réseau, entre deux hydrophobes non consécutifs. Dans les figures, les segments en pointillé indiquent les contacts entre hydrophobes qui sont pris en compte pour évaluer la qualité du repliement. Ainsi, dans l'exemple de la figure 2, on n'a que 2 contacts.

La figure 3 montre un cas qui est déjà plus réaliste. Le repliement de droite donne le plus de contacts et il est donc considéré comme le plus stable. C'est en fait la seule solution optimale pour cet exemple, aux symétries et rotations près.



8 contacts 9 contacts 10 contacts

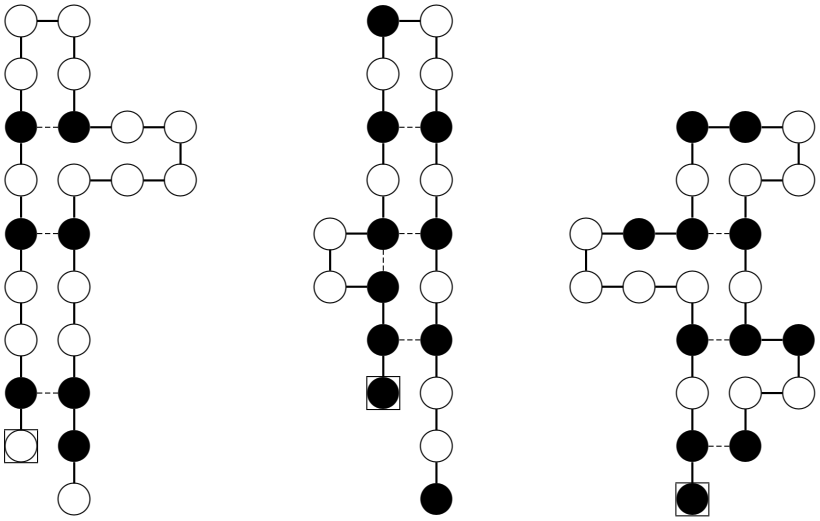
FIGURE 3 – Trois exemples de repliement pour la séquence `HHHPPHPHPHPPHPHPHPPH`

Le calcul du nombre maximal de contacts possibles, pour une séquence donnée, est un problème NP-difficile. On va s'intéresser à deux techniques de résolution. La première vise à trouver une solution approchée, très rapidement, même pour de grosses protéines. La seconde permet d'énumérer toutes les solutions optimales. L'exemple de la figure 3, avec 20 acides aminés, peut être traité en quelques millisecondes. Pour une trentaine d'acides aminés, il faut plutôt prévoir quelques minutes. Mais, en général, les protéines comportent quelques centaines d'acides aminés ...

Pour tout ce qui suit, on remarquera une propriété particulière de la maille carrée : si il y a un contact entre a_i et a_j, alors i et j sont de parités différentes.

2.1 Construction de solutions approchées en temps linéaire

On va construire ici des repliements qui ont la forme d'une "épingle à cheveux", comme illustré figure 4. Après avoir déterminé une position dans la séquence pour former le coude en $\cap$, en haut sur la figure, on descend deux branches en parallèle et on fait un pli d'un côté ou de l'autre pour compenser la différence de longueur de chemin entre des `H` appariables.



$n^{\mathcal{C}}_{\text{ref}} = 2\ ,\ n^{\mathcal{C}}_{\text{eff}} = 3$ $\quad$ $n^{\mathcal{C}}_{\text{ref}} = 3\ ,\ n^{\mathcal{C}}_{\text{eff}} = 4$ $\quad$ $n^{\mathcal{C}}_{\text{ref}} = 4\ ,\ n^{\mathcal{C}}_{\text{eff}} = 3$

FIGURE 4 – Exemples de solutions approchées

Soit $n^{\mathcal{P}}_{i,j}$, le nombre de H de rang pair dans la sous-séquence $a_i \dots a_j$. On définit de même $n^{\mathcal{I}}_{i,j}$ pour les H de rang impair.

Soit $n^{\mathcal{C}}(k) = \max(\min(n^{\mathcal{P}}_{0,k}, n^{\mathcal{I}}_{k+1,n-1}), \min(n^{\mathcal{I}}_{0,k}, n^{\mathcal{P}}_{k+1,n-1}))$ pour k variant de 0 à $n-1$. C'est le nombre maximal de couples possibles entre des H de rang pair d'un côté et de rang impair de l'autre, lorsque l'on coupe la séquence en $a_0 \dots a_k$ et $a_{k+1} \dots a_{n-1}$. Le choix des parités est donné par l'argument du max qui détermine le calcul de $n^{\mathcal{C}}(k)$. On peut montrer que $n^{\mathcal{C}}(k)$ passe par un maximum $n^{\mathcal{C}}_{\text{ref}}$ quand k varie de 0 à $n-1$ (en fait, ce maximum peut être atteint pour plusieurs positions k consécutives). On peut aussi montrer que $n^{\mathcal{C}}_{\text{ref}}$ est au minimum de l'ordre du quart du nombre optimal de contacts.

On peut calculer les $n^{\mathcal{C}}(k)$ et $n^{\mathcal{C}}_{\text{ref}}$ en temps linéaire, par balayage de la séquence. On peut ensuite construire un repliement "en épingle à cheveux" qui réalise au moins $n^{\mathcal{C}}_{\text{ref}} - 1$ contacts, c'est-à-dire un contact pour chaque couple de H de bonne parité, sauf le couple le plus interne quand il est formé de deux noeuds consécutifs. Le nombre effectif de contacts ainsi réalisés, $n^{\mathcal{C}}_{\text{eff}}$, peut être supérieur au minimum prédit, parce que l'on aura incidemment construit des contacts supplémentaires, soit des contacts de parité inversée, soit des contacts internes aux plis latéraux.

Dans les exemples de la figure 4, $n^{\mathcal{C}}_{\text{ref}}$ vaut respectivement 2 (les pairs à gauche), 3 (les impairs à gauche) et 4 (les impairs à gauche). Dans le premier exemple, on observe un contact supplémentaire de parité inversée. Dans le second exemple, on observe un contact supplémentaire qui résulte de la manière particulière de faire un pli. Le troisième exemple est le cas où la paire la plus interne (en haut) ne compte pas comme contact car les deux H sont consécutifs dans la séquence.

On peut imaginer des ajustement locaux qui permettent d'augmenter un peu le nombre de contacts, mais ce ne sont que des cas particuliers et on restera assez loin de l'optimal qui, pour ces trois exemples, est respectivement de 6, 10 (voir figure 3) et 9 contacts.

2.2 Recherche exhaustive

2.2.1 Principe général

Le principe de base d'une recherche exhaustive consiste à construire une à une toutes les formes possibles et à évaluer le nombre de contacts pour chacune. Pour commencer, on place le premier acide aminé sur un nœud arbitraire. Pour éviter d'engendrer des solutions identiques

à une rotation près, on fixe aussi la position du second acide aminé, par exemple à droite du premier. Dans le même ordre d'idée, on mettra une contrainte sur le premier déplacement vertical pour éviter de construire des solutions symétriques.

Le principe général est le suivant :
- supposons les $i-1$ premiers acides aminés déjà placés, on doit maintenant placer le i-ème,
- on cherche un place libre autour de a_{i-1}, on y place alors a_i ,
- si $i = n-1$, on a une solution, sinon on poursuit récursivement pour placer a_{i+1}, ...
- on essaie ensuite de même, à tour de rôle, les autres positions possibles pour a_i ,
- quand on a essayé toutes les positions pour a_i , on revient à un autre choix pour a_{i-1} , ...

Pour espérer traiter des exemples non triviaux, on doit adapter diverses stratégies usuelles en optimisation combinatoire qui permettent de limiter le nombre de formes explorées.

2.2.2 Statégies d'accélération

Dès que l'on a une solution, la seule chose qui nous intéresse est de trouver des solutions qui égalent ou améliorent le nombre de contacts. Et, bien sûr, chaque fois que l'on trouve une solution qui améliore ce nombre cible, on actualise la cible pour la suite. En considérant le nombre de `H` qui restent à placer, on peut majorer le nombre de contacts encore réalisables et, connaissant le nombre de contacts déjà effectivement réalisés par le placement actuel de $a_0 \dots a_{i-1}$, on peut détecter si un certain choix pour a_i interdit d'égaler ou de dépasser le nombre cible. Dans ce cas, il est inutile de faire ce choix pour a_i et d'explorer l'ensemble des positions pour $a_{i+1} \dots a_{n-1}$ qui en découlent. Ce principe dit "branch and bound" permet à lui seul d'accélérer notablement la recherche des solutions optimales.

De plus, si on dispose à l'avance d'une solution approchée, du type de celles envisagées en 2.1, on peut initialiser la cible au nombre de contacts connus pour cette solution particulière. De telles stratégies sont nécessairement gagnantes en nombre de formes explorées. Mais elles ne seront effectivement utiles que si leur mise en œuvre ne fait pas perdre plus de temps que le temps gagné en évitant l'évaluation de certaines combinaisons.

Voici maintenant d'autres stratégies qui mènent aussi à une solution optimale, mais dont le résultat n'est pas garanti du point de vue du nombre de combinaisons explorées.

Si le nombre de contacts cible est surévalué, l'exploration peut être considérablement écourtée pour conclure qu'il n'y a pas de solution réalisant ce nombre cible. Dans certains cas, il peut être ainsi efficace de partir d'un majorant raisonnable et de recommencer une recherche exhaustive en faisant décroître à chaque fois la cible, jusqu'à trouver le nombre optimal de contact. On peut facilement donner une majoration du nombre de contacts en fonction de $n^{\mathcal{P}}_{0,n-1}$ et $n^{\mathcal{I}}_{0,n-1}$.

Parfois aussi, le calcul peut être plus rapide si on considère la séquence renversée, c'est-à-dire que l'on place les acides aminés dans l'ordre $a_{n-1} \dots a_0$. Cela relève d'un principe général qui préconise d'appliquer d'abord les contraintes les plus fortes. On pourra essayer de justifier des heuristiques qui permettraient de décider a priori, pour une séquence donnée, si telle ou telle stratégie semble préférable. On validera alors expérimentalement.

3 Travail demandé

Il s'agira avant tout de programmer les méthodes exposées en 2.1 et 2.2. On peut facilement distribuer le calcul à l'aide de MPI. En remarquant que la méthode de recherche exhaustive revient à parcourir un arbre, il suffit de confier l'exploration de différents sous-arbres à autant de nœuds de calcul. La chose un peu délicate sera de propager correctement à tous les nœuds chaque amélioration du meilleur nombre de contacts connu.

Pour visualiser les résultats, on devra également produire une représentation graphique. On se contentera d'un graphisme rudimentaire, dans le style des figures qui sont présentées ci-dessus.

Au choix, la visualisation sera intégrée au programme ou se fera par l'écriture de fichiers dans un format reconnu par un utilitaire graphique existant, par exemple au format `svg` ou `latex`.

Des exemples à traiter sont donnés sur la page web [1]. On y trouvera aussi le nombre de solutions optimales attendues et des temps de calculs indicatifs pour chacun de ces exemples.

Il est conseillé de concevoir un programme qui peut fonctionner selon deux modes :

1. recherche du nombre optimal de contacts et, en option, affichage d'une solution optimale,
2. collecte de toutes les solutions optimales et, en option, affichage une par une.

4 Extensions possibles

Voici deux pistes d'extension du sujet. La première est la plus intéressante dans le cadre d'un projet INF442. La seconde peut se décomposer en une suite de petites évolutions qui demandent chacune relativement moins de travail.

4.1 Algorithmes génétiques

Pour traiter le cas de séquences relativement longues, il faut utiliser des algorithmes qui visent à obtenir assez rapidement une solution intéressante, mais sans chercher à atteindre l'optimal ni forcément le connaître. On pourra expérimenter en particulier l'utilisation des algorithmes génétiques ou autres [3, 4, 5, 6]. Ceux-ci peuvent faire facilement l'objet d'un calcul distribué.

4.2 Pertinence biologique

D'autres améliorations sont envisageables pour atteindre des résultats plus réalistes [5]. On peut par exemple considérer un modèle d'interaction un peu plus fin que HP, par exemple le modèle dit HPNX (Hydrophobe, Positif, Négatif et X pour autres).

Le passage à un modèle en 3D pose le problème de la visualisation du résultat. Le cas de la maille cubique qui conserve beaucoup de propriétés de la maille carrée, est assez facile. Une maille beaucoup plus réaliste serait la maille cubique faces centrées.

Références

[1] `http://www.enseignement.polytechnique.fr/profs/informatique/Philippe.Chassignet/14-15/REPLIEMENTS/index.html`

[2] Dill K.A. : "Theory for the folding and stability of globular proteins", Biochemistry, Vol. 24, No. 6, March 1985, pp. 1501-1509.

[3] Unger R., Moult J. : "Genetic Algorithms for Protein Folding Simulations", J. Mol. Bio., No. 231, 1993, pp. 75-81.

[4] Hoque T., Chetty M., Lewis A., Sattar A. "DFS Based Partial Pathways in GA for Protein Structure Prediction" in "Pattern Recognition in Bioinformatics", LNBI, Vol. 5265, 2008, pp. 41-53.

[5] Hoque T., Chetty M., Lewis A., Sattar A. "Genetic Algorithm in Ab Initio Protein Structure Prediction Using Low Resolution Model : A Review" in "Biomedical Data and Applications", SCI, Vol. 224, 2009, pp. 317-342.

[6] Bornberg-Bauer E. : "Chain growth algorithms for HP-type lattice proteins", in proc. RECOMB'97, 1997, pp. 47-55