Rapport projet : Calcul de la surface accessible au solvant d'une protéine

GOULANCOURT Rebecca

Lien Github : https://github.com/Rebbekkah/Projet.git

I. Introduction

Dans ce projet nous chercherons grâce à un script en Python à déterminer la surface accessible, absolue et relative, d'une protéine par le solvant à partir de ses coordonnées qui seront extraites d'un fichier PDB. Le but de ce projet est d'avoir une idée de la structure de la protéine, de l'exposition ou l'enfouissement de ses résidus mais aussi de pouvoir déterminer avec quels résidus le solvant pourrait rentrer en contact, ainsi cela pourrait permettre de comprendre les perturbations structurales liées au solvant.
Le but final serait d'obtenir une surface exposée au solvant, ou surface accessible. À l'issue du calcul de la surface accessible au solvant nous pourrons comparer nos résultats avec ceux obtenus grâce à FreeSASA.
Nous prendrons comme référence de solvant une molécule d'eau (H_20) dont le rayon est de 1.4 $Ä^2$.

II. Matériel et méthodes

1) Matériel

- La Proteine Data Bank (PDB)
-

La PDB est une banque de données protéique permettant de regrouper toutes les informations obtenues sur ces protéines, comme par exemple leurs structures 3D. Toutes ces informations peuvent être téléchargées en format .pdb (*identifiant_de_la_protéine.pdb*). Dans notre cas c'est en téléchargeant ce fichier PDB que l'on peut récupérer les coordonnées atomiques d'une protéine d'intérêt.

- Python 3.9 et les bibliothèques

La version de python que nous avons utilisée est la version 3.9. Les bibliothèques (ou encore modules ou librairies) sont des fonctions open sources téléchargeables à l'aide de la commande « import ». Elles permettent de palier des problèmes liés à un code trop lourd à écrire en important un code déjà écrit. Dans notre cas nous avons importé les modules pandas, numpy, sys, scipy, matplotlib, mpl_toolkits et collections.
La bibliothèque pandas permet la manipulation et l'analyse de données et numpy aide à la manipulation des vecteurs, matrices et tableaux. Ces deux bibliothèques sont complémentaires et seront clés dans notre script car les données et output seront lus sous forme de DataFrame.
Le module sys permet d'accéder à des variables de l'interpréteur, dans notre cas le nom du fichier en entrée tandis que scipy (extension de numpy) est un ensemble de bibliothèques servant à résoudre des formules et problèmes mathématiques.

La librairie matplotlib est une librairie complète donnant accès à des représentations graphiques, alors que mpl_toolkits permet la création de graphes complexes en 3 dimensions comme par exemple des sphères. Enfin le module collection permet rapidement de créer un « dictionnaire de dictionnaire ».

- Github

Pour ce qui est de la gestion du script celui-ci a été régulièrement déposé sur Github. C'est une plateforme essentiellement pour la collaboration où l'on peut s'inscrire pour déposer ou avoir accès à des fichiers.

- Environnement utilisé

Pour ce projet j'ai utilisé l'environnement de base auquel j'ai installé les librairies citées plus haut lorsqu'elles n'étaient pas déjà présentes.

- FreeSASA

FreeSASA est une librairie écrite en langage C et accessible par des commandes en C++ dans le terminal. Elle permet d'obtenir la surface accessible d'un solvant de molécules biologiques par l'intermédiaire de la méthode de Lee & Richard's. Une démo est disponible sur le site officiel de FreeSASA (https://freesasa.github.io/) où l'on peut rentrer l'identifiant PDB d'une protéine et obtenir en output sa surface accessible au solvant. Sinon le logiciel peut être installé en suivant le guide du lien. Nous pouvons ainsi comparer des résultats avec ceux obtenus via FreeSASA.

2) Méthodes

En premier lieu nous allons choisir dans la banque PDB une molécule. Ensuite il faudra récupérer le fichier correspondant en format .pdb et le stocker dans le même dossier que le script. Pour le faire tourner il faudra l'appeler et ensuite passer en argument l'identifiant de la molécule étudiée (en format .pdb) comme ceci : python3 script.py *identifiant_pdb*.pdb.

Le code devra être construit de sorte à respecter les étapes clés présentées ci-dessous.
i) Parsing des fichiers PDB afin de récupérer les coordonnées atomiques de chaque atome présent dans la protéine.

ii) Construire une matrice de distance des atomes deux à deux de la protéine à partir de leurs coordonnées (méthode euclidienne). On obtiens alors une matrice de la distance séparant chaque atome. Ensuite on définit un seuil à partir duquel les atomes ne seront plus considérés comme voisins l'un de l'autre. Ce seuil est définit essentiellement par le plus grand Rayon de Wan der Waals atomique (ici le phosphore P) tel que : seuil = $rayon_{WdW\_max}$ *2 +1.4. Si la distance séparant les couples d'atomes est inférieur au seuil alors on pourra les considérer voisins.

iii) On récupère ensuite pour chaque atome une liste de ses voisins et on en fait un « dictionnaire de listes ». Ensuite on place une sphère sur l'atome étudié tel que : $rayon_{sphère} = rayon_{VanDerWaals\_atome.}$

Pour cela il faut créer une nouvelle matrice qui contient de nouvelles coordonnées tel que :

Atome	Coordonnées initiale	Coordonnées sur la sphère
0	$\{x_0, y_0, z_0\}$	$\{x_0 + r_{WdW\_atome}, y_0 + r_{WdW\_atome}, z_0 + r_{WdW\_atome}\}$
1	$\{x_1, y_1, z_1\}$	$\{x_1 + r_{WdW\_atome}, y_1 + r_{WdW\_atome}, z_1 + r_{WdW\_atome}\}$
…	$\{x_n, y_n, z_n\}$	$\{x_n + r_{WdW\_atome}, y_n + r_{WdW\_atome}, z_n + r_{WdW\_atome}\}$

Figure 1 : Tableau récapitulatif de la méthode pour obtenir les coordonnées de la sphère.

On dispose alors d'une sphère de N points de coordonnées {x, y, z} centrée sur l'atome étudié.

v) Pour pouvoir déterminer la surface accessible au solvant il va falloir étudier chaque point de la sphère et définir si ces points sont en contact avec un des atomes voisins. Si c'est le cas alors on considérera que ce point précis n'est pas exposé et le retirer puis passer au point suivant. Sinon il s'agira de regarder si la distance entre le point de la sphère et le voisin est supérieur au diamètre de la molécule d'eau. Pour cela on utilisera la formule présentée en figure 3.

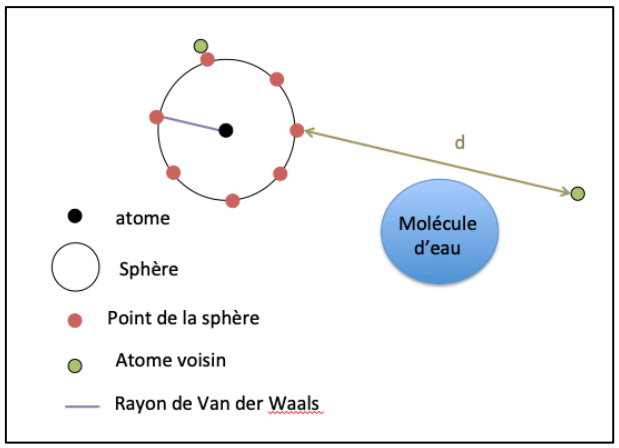

Figure 2 : Schéma explicatif du raisonnement.

$$d = \sqrt{(x_{sphère} - x_{atome\_voisin})^2 + (y_{sphère} - y_{atome\_voisin})^2 + (z_{sphère} - z_{atome\_voisin})^2}$$

Figure 3 : Formule pour calculer la distance entre un atome et la sphère de l'atome voisin.

Ainsi si cette distance est supérieure au diamètre de la molécule d'eau (2*r = 2.8 Ä) alors on peut considérer que l'espace entre l'atome et son voisin est accessible par le solvant. Il faut ensuite faire une translation de la sphère sur le prochain atome et recommencer le processus. Si l'on observe que sur une sphère il y a par exemple 30 points sur 100 qui sont accessibles au solvant alors on peut conclure qu'il y a 30% de surface relative accessible. Pour obtenir le pourcentage de la surface absolue il suffira donc d'additionner les surfaces accessibles relatives.

Il s'agira en dernier lieu de calculer l'aire accessible localement grâce à la formule $aire_{sphère} = 4\pi r^2$, puis l'aire absolue qui correspond à la somme des aires locales.
À la suite de tout cela nous pourrons comparer les résultats obtenus avec le logiciel FreeSASA.

III. Résultats et conclusion

Le code n'étant pas fini, nous n'avons pas de résultats concrets. Toutefois nous avons réussi à placer les sphères aux coordonnées {0, 0, 0} mais pas à les centrer sur chacun des atomes et obtenir les coordonnées de chacun des points de la sphère. La fonction Sphère(*number_points, coord*) sera présente dans le code comme exemple de code mais ne sera pas utilisable car elle ne renvoie pas le bon résultat, nous mettrons donc en commentaire la ligne de commande permettant de renvoyer le faux résultat.

IV. Discussion

Nous pouvons émettre l'hypothèse que plus il y aura de points sur la sphère plus les résultats seront précis car un nombre de points élevé permet de couvrir et calculer sur une plus grande surface et éviter de biaiser les résultats lors des calculs de surface accessibles. Voici des exemples de sphères obtenues avec différents nombres de points :

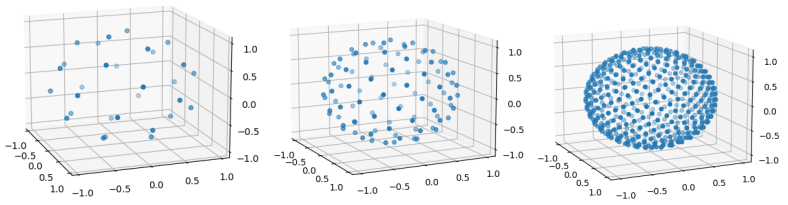

Figure 4 : Exemples de sphères générées traçables autour des atomes pour N = 30, N = 100 et N = 500.

Nous pouvons aussi nous questionner sur le solvant pris en référence. En effet le rayon de la molécule d'eau a été choisi selon la méthode de Pauling où r_{eau} = 1.4 Ä tandis que selon la méthode de Bondi il est de 1.715Ä. Donc les résultats peuvent varier selon les sources.

Aussi avec plus de temps nous aurions pu achever le projet et obtenir une surface accessible au solvant, mais aussi tracer des graphes comme par exemple de la surface accessible au solvant en fonction du numéro atomique, comme présenté ci-dessous.

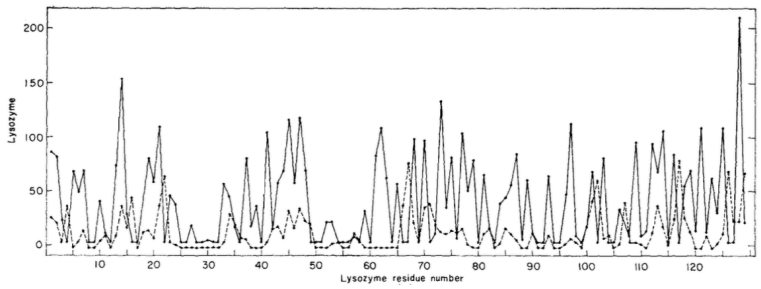

Figure 5 : Exemple de graphe de la surface exposée du lysozyme en fonction du numéro de résidus.

De même avec plus de temps, nous aurions pu écrire un code plus propre, compact et mieux écrit.