Université Paris Cité

M2 Biologie Informatique

PROJET COURT PYTHON

CALCUL DE LA SURFACE ACCESSIBLE AU SOLVANT D'UNE PROTEINE

Pr Jean Christophe GELLY

Sarah BLANCHET-DEVERLY

08/07/2024

Lien GITHUB :
https://github.com/sarahblanchetdeverly/Projet_Court_Accessibilite_solvant_proteine

INTRODUCTION

La surface accessible au solvant des acides aminés des protéines est un paramètre crucial dans l'étude du repliement des chaînes polypeptidiques et pour le calcul de leur stabilité (1). Ce descripteur revêt une importance primordiale dans divers contextes, notamment pour prédire la structure des protéines.

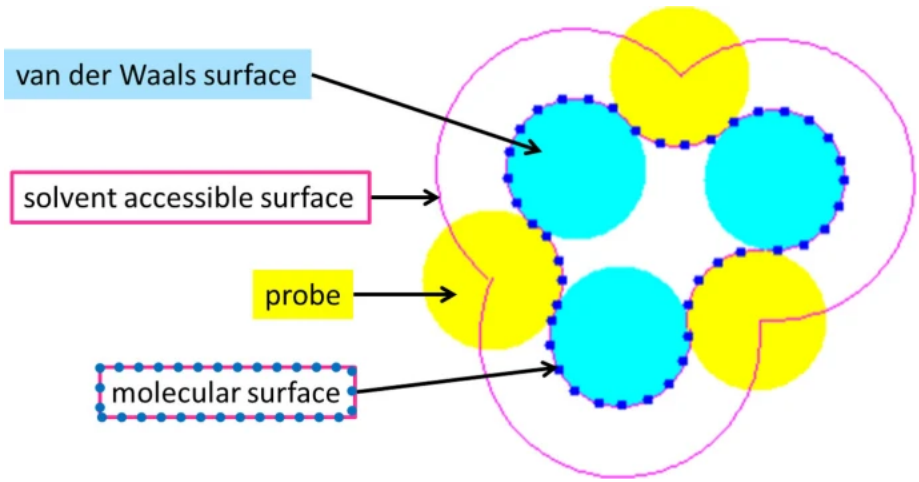
L'algorithme de Shrake et Rupley (2) est une méthode numérique utilisée pour calculer l'aire de surface accessible au solvant. Il crée un maillage de points équidistants autour de chaque atome d'une molécule, permettant ainsi de déterminer les points accessibles au solvant. Cette information est cruciale pour comprendre la fonction des protéines dont la séquence en acides aminés est connue, notamment dans la prédiction des interactions entre monomères.

La surface accessible au solvant est définie comme l'ensemble des positions où une sonde (représentée par une sphère) peut rouler sur la surface de la molécule. Chaque atome de la protéine est modélisé par une sphère de points N uniformément répartis à une distance égale au rayon de Van der Waals de l'atome. Ainsi, les protéines sont représentées comme un ensemble de sphères de rayon de Van der Waals solvatées.

Le projet vise à développer un programme capable de calculer la surface accessible au solvant (absolue et relative) à partir des coordonnées d'une protéine issues d'un fichier de la Protein Data Bank (PDB). Pour ce faire, l'insuline humaine (PDB 3i40) a été sélectionnée comme modèle d'étude. L'insuline est une petite protéine composée de deux chaînes polypeptidiques (chaîne A et chaîne B), reliées par des ponts disulfure (figure 2). La forme active de l'insuline est cruciale pour son interaction avec le récepteur de l'insuline à la surface des cellules.

En somme, ce projet vise à utiliser l'algorithme de Shrake-Rupley pour mieux comprendre comment la surface accessible au solvant de l'insuline humaine contribue à sa fonction biologique, en particulier dans ses interactions avec d'autres molécules.

Figure 1: Surface accessible au solvant d'une protéine, issue de Proteome Science, Heng Yang et al 2012 (3)



Légende :La surface de van der Waals (en cyan) est obtenue en faisant l'union des surfaces sphériques des atomes définies par le rayon de van der Waals de chaque atome. La surface accessible aux solvants (en rose) est définie par le chemin tracé par le centre d'une sonde (en jaune) enroulée autour de la protéine.

Figure 2: Représentation de la protéine 3I40 correspondant à l'insuline humaine (issue de la PDB)

MATERIEL ET METHODES :

1) SCRIPT :

Librairies python :

Ce programme a été codé en Python 3 et réalisé dans un environnement Linux.
Les librairies utilisées ont été les suivantes :

- *math :* pour les fonctions mathématiques de base comme les calculs de distances et de surfaces.
- *pathlib :* pour la manipulation des chemins de fichiers de manière portable.
- *sys :* pour récupérer les arguments de la ligne de commande.
- *collections :* Pour utiliser defaultdict, un dictionnaire avec des valeurs par défaut automatiques.

Données :

Ce code a été testé sur le pdb de la protéine : 3i40 correspondant à l'insuline humaine extraite de la Protein Data Bank.
Le script utilise un dictionnaire implémenté « VanderWaalR » contenant les rayons de Van der Waals pour différents atomes. Ces rayons sont essentiels pour modéliser les atomes comme des sphères et pour calculer la surface accessible au solvant.
Lorsque le rayon de Van der Waals d'un élément était inconnu, une valeur par défaut de 2 a été attribuée.

Fonctions utilitaires :

Les fonctions suivantes ont été créées :

- sauvegarde_points : enregistre une liste de points dans un fichier CSV.
- sauvegarde_atomes : enregistre les informations des atomes dans un fichier CSV.
- lire_pdb : lit un fichier PDB et extrait les coordonnées des atomes, les rayons de Van der Waals, et prépare les sphères atomiques pour les calculs ultérieurs.

Fonctions mathématiques

- calculer_surface **:** calcule la surface d'une sphère donnée son rayon.
- calculer_distance **:** Calcule la distance euclidienne entre deux points.
- sphere_unite : génère des points uniformément distribués sur une sphère unitaire en utilisant l'algorithme de Saff et Kuijlaars.
- mise_a_echelle_points : met à l'échelle et translate les points de la sphère unitaire pour correspondre au rayon et à la position d'un atome.

Calculs de l'accessibilité des atomes et résidus

- marquer_points_accessibles : détermine si chaque point de la sphère est accessible en vérifiant les distances avec les autres atomes.
- calculer_surface_accessible **:** calcule la surface accessible d'un atome en fonction des points accessibles.
- calculer_accessibilite **:** intègre les deux fonctions précédentes pour déterminer l'accessibilité d'un atome.

- calculer_accessibilite_residu : calcule l'accessibilité d'un résidu entier en additionnant les surfaces accessibles de ses atomes constituants.

Programme principal :

Le script principal exécute les étapes suivantes :

- Lecture des arguments : Récupère les arguments de la ligne de commande, incluant le nombre de points de la sphère, le nombre d'atomes proches à considérer, et le fichier PDB à analyser.
- Génération des points de la sphère unitaire : utilise sphere_unite pour générer des points uniformément distribués sur une sphère unitaire.
- Lecture du fichier PDB : utilise lire_pdb pour extraire les informations atomiques et préparer les sphères atomiques.
- Calcul des surfaces accessibles : pour chaque atome, calcule la surface accessible au solvant en marquant les points accessibles et en calculant la surface accessible.
- Sauvegarde des résultats : les résultats sont exportées et préparées dans un dataframe pour être ultérieurement enregistrés dans un tableau excel

Exécution du script dans le terminal :

Pour exécuter le script, il faut utiliser la commande suivante dans le terminal :
python3 script.py <nombre_points_sphere> <nombre_atomes_proches> <fichier_pdb>

Exemple : python3 projet_court_proteine.py 100 5 3i40.pdb

2) NACCESS :

Naccess est un logiciel gratuit fonctionnant sur Linux et Windows. Il permet d'obtenir la surface atomique et résiduelle des protéines.
Nos résultats ont été comparés avec ceux trouvés par Naccess pour une même protéine

RESULTATS

En prenant comme exemple la protéine 3i40.pdb avec les paramètres suivants en entrée :
Nombre de points sur la sphère = 100
Nombre d'atomes les plus proches = 5
On obtient une surface totale de la protéine de **13 993.6 A°** , une surface accessible **de 3 378.3 A°** et un pourcentage d'accessibilité au solvant de **24.1%**

Lorsqu'on compare nos résultats avec ceux retrouvés par Naccess, on constate qu'ils sont relativement proches : en effet la surface accessible de tous les atomes de cette même protéine est calculée à **3 368.8 A°.**

On obtient également le détail de la surface accessible par résidu (table 1).
Les résultats sont similaires pour certains résidus et diffèrent pour d'autres par rapport à ceux générés avec Naccess.

Par exemple pour le premier résidu : GLY 1 on a une surface accessible totale de 36,7 avec notre programme vs 41,02 obtenue avec Naccess, ce qui est relativement proche.
En revanche pour le dernier résidu : ALA 30 on a une surface accessible de 56,99 dans notre programme vs 96,91 obtenue avec Naccess.

Table 1 : Accessibilité au solvant pour chaque résidu protéique. Surface totale, surface accessible totale et pourcentage d'accessibilité au solvant.

Résidu	Surface totale	Surface accessible totale	% Accessibilité au solvant
GLY1	131,86	36,7	27,83
ILE2	277,12	67,16	24,24
VAL3	240,81	63,65	26,43
GLU4	298,87	61,51	20,58
GLN5	300,03	73,93	24,64
CYS6	208,89	44,88	21,49
CYS7	417,78	100,72	24,11
THR8	233,52	57,27	24,52
SER9	394,42	93,8	23,78
ILE10	277,12	75,01	27,07
CYS11	208,89	48,14	23,05
SER12	197,21	50,68	25,7
LEU13	277,12	73,28	26,44
TYR14	698,36	119,53	17,12
GLN15	300,03	74,62	24,87
LEU16	277,12	66,11	23,86
GLU17	298,87	68,85	23,04
ASN18	263,72	61,8	23,44
TYR19	415,11	89,76	21,62
CYS20	208,89	50,88	24,36
ASN21	292,75	74,85	25,57
PHE1	386,08	101,58	26,31
VAL2	240,81	64	26,58
ASN3	263,72	66,9	25,37
GLN4	300,03	73,87	24,62
HIS5	337,51	81,07	24,02
LEU6	277,12	66,53	24,01
GLY8	131,86	31,48	23,87
HIS10	337,51	79,9	23,67
LEU11	277,12	70,01	25,26
VAL12	240,81	61,64	25,6
GLU13	298,87	73,47	24,58
ALA14	168,17	44,68	26,57
LEU15	277,12	68,56	24,74
TYR16	415,11	95,42	22,99
LEU17	277,12	70,52	25,45

VAL18	240,81	63,26	26,27
CYS19	208,89	47,81	22,89
GLY20	131,86	32,6	24,73
GLU21	298,87	73,48	24,58
ARG22	367,7	88,12	23,97
GLY23	131,86	32,48	24,63
PHE24	386,08	93,77	24,29
PHE25	386,08	91,91	23,81
TYR26	415,11	94,31	22,72
THR27	233,52	59,13	25,32
PRO28	240,81	60,29	25,04
LYS29	307,32	81,39	26,48
ALA30	197,21	56,99	28,9

DISCUSSION

Pour conclure, l'algorithme de Shrake et Rupley que nous avons implémenté dans notre code semble produire des résultats comparables à ceux obtenus avec Naccess. Cependant, notre programme présente plusieurs limitations significatives. Tout d'abord, nous avons calculé la surface accessible totale pour tous les atomes d'un résidu sans distinguer s'ils sont polaires ou non polaires, ni considérer leur position sur la chaîne (principale ou non). De plus, lors du calcul final de la surface accessible totale en incluant tous les atomes, nous n'avons pas examiné en détail les chaînes d'insuline. En effet, l'insuline est composée de deux chaînes distinctes (chaîne A et chaîne B), et il serait pertinent d'étudier la contribution de chaque chaîne de manière indépendante à l'accessibilité de la protéine, à l'instar de ce qui est réalisé dans Naccess.