

Présenté par :

**BELAKTIB Anas**

**M2-BI**

**Calcul de la surface accessible au solvant d'une protéine.**

## Introduction

La surface accessible (ASA) ou surface accessible par un solvant (SASA) est la surface d'une biomolécule qui est accessible à un solvant. La mesure de l'ASA se fait en Ångstroms <sup>2</sup>. L'ASA a été décrite pour la première fois par Lee et Richards en 1971 [1]

Le but de ce projet a été de développer un outil permettant de calculer la surface accessible absolue et relative à un solvant d'une protéine à partir de son fichier pdb. Une fois notre outil développé nous avons cherché à comparer nos résultats avec un outil déjà disponible DSSP.

## Matériels et méthodes

Le programme a été développé uniquement en python et nécessite le module Biopython pour être utilisé.

### Pipeline utilisée :

Afin d'utiliser notre programme il faut en premier lieu télécharger le pdb de la protéine qui nous intéresse et exécuter le programme main.py. Une fois exécuté il suffit d'indiquer l'emplacement du pdb de la protéine que l'on souhaite étudier. Ce programme constitue un pipeline en 3 parties (Figure 1) qui s'exécute consécutivement afin d'afficher les différents résultats sur le terminal

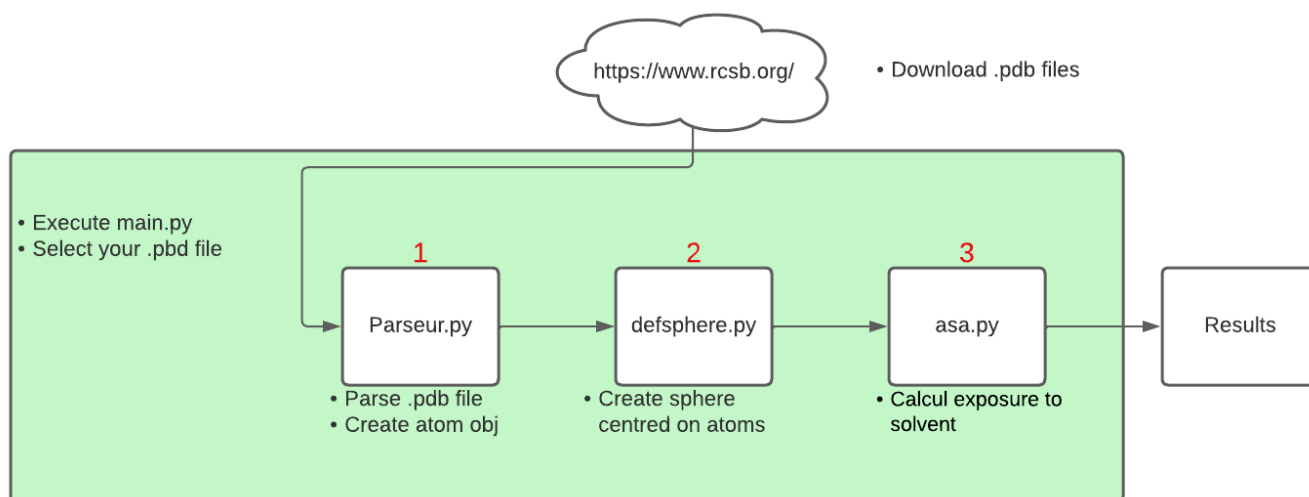


Figure 1: Pipeline du projet

### Extraire les informations des atomes présent dans des fichiers pdb (Figure 1-1) :

Tous les fichiers pdb que nous avons utilisés lors de cette étude sont issues du site de la RCBS Protein Data Bank [2].

Nous avons utilisé le module Biopython afin d'extraire les données de chaque atome. La sélection des atomes est réalisée par le programme afin de considérer les lignes du fichiers pdb commençant par ATOM et non HETAM. Pour les fichiers pdb contenant plusieurs modèles cristallographiques, notre programme va sélectionner les atomes constituant le premier modèle uniquement afin de ne pas fausser les résultats en prenant plusieurs fois le même atome. Cette étape est réalisée par le programme parseur.py.

### Création d'une sphère correspondant à chaque atome (Figure 1-2) :

Afin de créer une sphère représentant au mieux nos atomes, nous avons utilisé l'algorithme de Fibonacci. Ce dernier permet de distribuer des points rapidement, uniformément sur une sphère. Afin d'avoir une sphère avec des résultats concluants il faut prendre à minima 92 points la constituant d'après l'article de Shrake de 1973 [3].

Cette sphère une fois créée a été agrandie selon le rayon de nos différents atomes, obtenu d'après l'article de Shrake de 1973, auxquels nous avons ajouté le rayon du solvant utilisé (ici rayon du solvant = 1.4 Å) puis le centre de la sphère a été calibré sur les coordonnées de chacun de nos atomes. Chacune de ces sphères possèdent une surface différente selon l'atome qu'elle représente). Cette étape est réalisée par le programme defsphere.py.

Les valeurs de rayons utilisés sont les suivants : rayon\_Vdw = {"C" : 3.1, "O" : 2.8, "N" : 2.9, "S" : 3.25, "CU" : 2.8, "CL" : 2.9, "H" : 2.5, "P" : 3.2}

### Calcul de la surface accessible absolue au solvant ASA (Figure 1-3) :

Pour chacun de nos atomes nous avons cherchés les atomes voisins c'est à dire dont la distance entre les deux est inférieur à 10 Å.

En prenant par exemple 2 atomes voisins A et B (Shrake-Rupley):

Pour chacun des points constituant les sphères représentant notre atome A, nous avons recherchés la distance entre les points constituant la sphère A et le centre de l'atome B. Si cette distance se trouvait inférieur au rayon de l'atome B, alors on considèrait ce point comme enfoui

En parcourant un à un les points constituant nos différentes sphères nous avons pu déterminer les points qui était enfouis de ceux qui sont accessible au solvant. En reportant le ratio de point exposé sur le nombre de point total et en utilisant la surface de l'atome nous avons pu obtenir la surface exposée par atome. Les surfaces par atomes ont été ensuite additionnés pour former une surface par résidu puis une surface totale exposé.

$$Surface\ exposé\ par\ atome = \frac{Nb\ points\ exposé}{Nb\ points\ total} * Surface\ de\ l'atome$$

Ces opérations nous ont permis d'obtenir la surface accessible totale, le pourcentage d'accessibilité et la surface accessible absolue par résidu.

### Calcul de la surface accessible relative au solvant RSA (Figure 1-3) :

La surface accessible relative au solvant (RSA) est le rapport entre la surface accessible au solvant et le MaxASA qui est une valeur en Å<sup>2</sup> qui dépend de la nature du résidu. Nous avons utilisé les valeurs de MaxASA publié par Tien et al en 2013 pour réaliser notre étude.

$$RSA = \frac{ASA}{MaxASA}$$

### Programmation orienté objet (Figure 2) :

Chaque atome a été associé à un objet python avec plusieurs. On va retrouver des attributs correspondant à l'identification de l'atome en jaune, des attributs sur les coordonnées

spatiales de l'atomes en vert ainsi que des attributs qui seront calculés afin d'obtenir notre surface accessible au solvant en bleu.

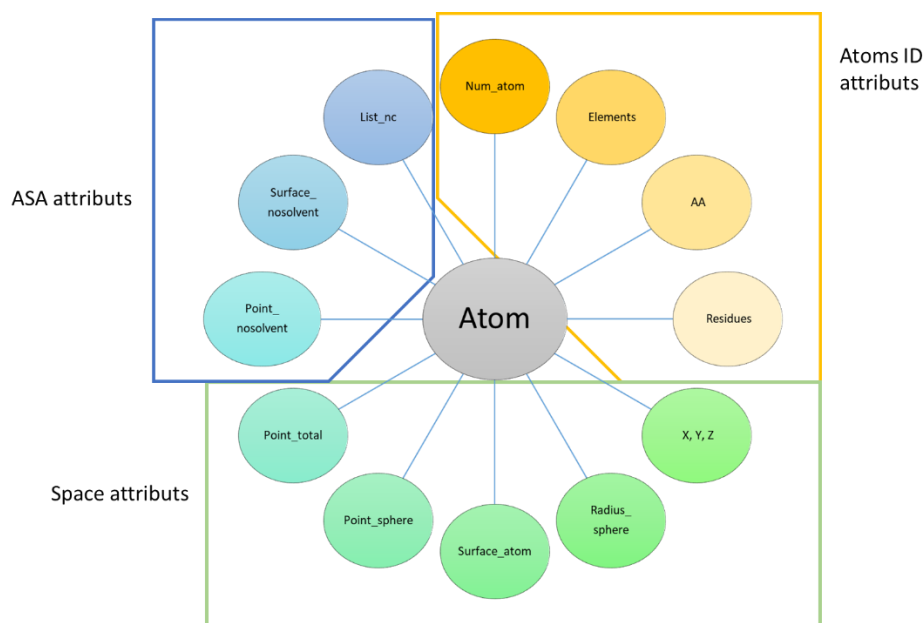


Figure 2: Utilisation de l'OOP

## Résultats

Le résultat obtenu pour une protéine est un ensemble d'information sur l'accessibilité au solvant par résidu ainsi qu'à l'échelle de la protéine dans sa globalité. On va y retrouver l'identifiant du résidu son accessibilité au solvant ainsi que l'accessibilité relative, le pourcentage d'accessibilité et la surface accessible totale. Pour la protéine dans notre exemple Figure 2, nous avons un pourcentage d'accessibilité de 5,9% et La surface accessible au solvant est de 9617 Å<sup>2</sup> et sa surface accessible relative est de 50%.

```
id residue:(92, 'PRO') - asa_abs (Å**2):53.81821397569194 - asa_rel (Å**2):0.3384793331804524
id residue:(93, 'GLU') - asa_abs (Å**2):224.0515585047771 - asa_rel (Å**2):1.0047155089900317
id residue:(94, 'LEU') - asa_abs (Å**2):42.69561007213468 - asa_rel (Å**2):0.2124159705081327
id residue:(95, 'LEU') - asa_abs (Å**2):1.312639365260779 - asa_rel (Å**2):0.00653054410577502
id residue:(96, 'LYS') - asa_abs (Å**2):215.96400628655752 - asa_rel (Å**2):0.9151017215532098

Le pourcentage d'accessibilité est de: 5.9437772667720195
La surface accessible au solvant est de 9617.304319950308Å**2
La surface accessible relative au solvant est de 50.69615441413572%
```

Figure 3: Exemple d'un résultat

Nous avons pu tester notre programme ainsi que le programme DSSP sur différents pdb : Nous avons comparé les résultats de la surface accessible totale en Å<sup>2</sup> entre nos 2 programme.

Fichier	Nb atomes	Résultat programme Å <sup>2</sup>	Résultat DSSP Å <sup>2</sup>	Différences
1bja.pdb	1423	9617.3	9721.4	-1.0%
1bjb.pdb	431	2450.5	2431.1	0.8%
1bjj.pdb	5837	37862.7	37435.6	1.1%

1bzb.pdb	362	3704.4	3839.8	-3.5%
1jhg.pdb	1806	8133.5	8222.1	1.0%
2ml6.pdb	2354	9958.1	10266.5	3.0%
6a5j.pdb	260	1596.43	1556.6	2.6%

Tableau 1: Tableau de comparaison de nos résultats avec le programme DSSP

## Discussion

### Résultats DSSP :

Nos résultats obtenus sont très proches de ceux du programme DSSP avec de légères différences. Ces différences peuvent s'expliquer par les valeurs utilisées pour les rayons d'atomes, la méthode de génération de sphère en effet nous ne savons pas comment le programme DSSP nous calcul les surface pour nos fichiers pdb l'utilisation d'arrondie pourra également être la cause de ces légères différences. Les différences entre les résultats DSSP et notre programme ne dépendent pas de la taille de nos protéines, en effet une très grosse protéine tel que la 1bjj.pdb qui est constitué de 5837 atomes à un écart de 1,1% entre nos deux programmes tandis qu'une petite protéine tel que la 6a5j qui est constitué de 260 atomes montre un écart de 2,6% entre nos deux programmes.

## Annexes

### Utilisation du programme :

Pour exécuter le programme il suffit dans votre terminale d'appeler le programme « main.py » sous le langage python.

```
PS C:\Users\belak\Desktop\M2\projet1\nouveaurepo\bin> python .\main.py
```

#### Annexe 1: Exécution du programme

Un message va alors s'affiché afin de vous demander quelle protéine en question voulez – vous étudier, il suffira alors de lui insérer l'emplacement du fichier. Dans l'exemple, nous avons choisis de traiter la protéine 1bja.pdb.

```
Entrer l'emplacement de votre fichier. pdb:../data/1bja.pdb
```

#### Annexe 2: Sélection du fichier .pdb

Quelques secondes plus tard, nos résultats s'affichent sur le terminal.

```
id residue:(88, 'GLN') - asa_abs (Å**2):190.78345821311058 - asa_rel (Å**2):0.8479264809471582
id residue:(89, 'GLU') - asa_abs (Å**2):217.5921708270266 - asa_rel (Å**2):0.9757496449642448
id residue:(90, 'ASN') - asa_abs (Å**2):81.20060872635044 - asa_rel (Å**2):0.41641337808384843
id residue:(91, 'ALA') - asa_abs (Å**2):10.337205740159808 - asa_rel (Å**2):0.08013337783069618
id residue:(92, 'PRO') - asa_abs (Å**2):53.81821397569194 - asa_rel (Å**2):0.3384793331804524
id residue:(93, 'GLU') - asa_abs (Å**2):224.0515585047771 - asa_rel (Å**2):1.0047155089900317
id residue:(94, 'LEU') - asa_abs (Å**2):42.69561007213468 - asa_rel (Å**2):0.2124159705081327
id residue:(95, 'LEU') - asa_abs (Å**2):1.312639365260779 - asa_rel (Å**2):0.00653054410577502
id residue:(96, 'LYS') - asa_abs (Å**2):215.96400628655752 - asa_rel (Å**2):0.9151017215532098

Le pourcentage d'accessibilité est de: 5.9437772667720195
La surface accessible au solvant est de 9617.304319950308Å**2
```

#### Annexe 3: Aperçue des résultats

### Difficultés rencontrées :

Le programme a rencontré plusieurs difficultés durant son élaboration. La première version du programme créait un document texte contenant les informations de chaque atome par ligne. Le programme était assez long à s'exécuter car pour chaque opération une recherche dans un document devait être réalisée, c'est -à -dire ouverture puis fermeture du fichier texte. Afin de remédier à ce problème, nous avons préféré travailler avec une liste de listes. Chacune de ces listes correspondaient à un atome et chaque élément de la liste était une information sur l'atome. Un gain de temps considérable a été gagné mais utilisé des listes n'est pas optimale dans ce genre de situation. Finalement, nous avons décidé de représenter nos atomes comme des objets. Ainsi chaque donnée qui nous intéresse concernant un atome peut être stocké dans un attribut. Ce qui nous a donné un code plus lisible et plus organisé facile à comprendre car les noms d'attributs correspondent à leur contenu.

### Bibliographie

[1] Lee, B; Richards, FM. (1971). "The interpretation of protein structures: estimation of static accessibility" . *J Mol Biol*. 55 (3): 379–400. doi:10.1016/0022-2836(71)90324-X.

[2] <https://www.rcsb.org/>

[3] Shrake, A; Rupley, JA. (1973). "Environment and exposure to solvent of protein atoms. Lysozyme and insulin". *J Mol Biol* 79 (2): 351–71. doi:10.1016/0022-2836(73)90011-9.

[4] Tien, et al.. (2013). "Maximum allowed solvent accessibilities of residues in proteins". *PLOS ONE*. 8 (11): e80635. doi:10.1371/journal.pone.0080635