*le : 5 Janvier 2017*

UNIVERSITÉ DE MONTPELLIER

Bioinformatique
Connaissances
Données

# Projet structure de molécules

---

Projection d'alignements de séquences protéiques sur des structures 3D connues : développement de procédures en langage Jmol scripting

version Jmol utilisée 14.6.4

*Equipe M2 BCD: Audrey PASSERIEUX, Younes ZAHIDI, Abdoulaye DIALLO*
*Enseignant : Stefano TRAPANI*

# Sommaire

# Introduction

Dans ce compte rendu seront abordées les différentes étapes du projet que nous avons effectué pour le sujet qui nous a été fourni durant l'enseignement Structure de Molécules.
Il va y être explicité les différentes parties du projet, les solutions que nous y avons apportés ainsi que les problèmes rencontrés au cours du développement du script sur le logiciel JMOL (version 14.6.4). En plus de cela, un test a été appliqué pour chaque partie afin de vérifier son fonctionnement.

# Déroulement du projet

## Partie 1/4

Dans la première partie du projet il a été question de développer des fonctions en utilisant un script en langage Jmol afin de définir une nouvelle propriété atomique aux résidus « property_seqresno » d'une façon à pouvoir donner une nouvelle numérotation séquentielle aux résidus tels qu'ils apparaissent dans les enregistrements SEQRES du fichier PDB.

Pour cela il faut respecter certaines conditions dans la nouvelle numérotation, en comptant les résidus manquants ainsi que les résidus modifiés, la remise à zéro de la numérotation à la fin de chaque chaîne de la structure…

Nous disposons pour cela d'un fichier au format PDB « 1a2c.pdb » pour effectuer les tests de nos scripts. Ce fichier contient une structure qui représente beaucoup de difficultés pour l'exécution de notre script, elle contient quatre chaînes L, H, I, J dans lesquelles on retrouve des défis liés à:

- Des anomalies de numérotations : on retrouve au sein de la même chaîne des numéros de résidu par exemple:
  si on a : 1 , 2 , 2A , 2B , 3
  cela deviendra : 1 , 2 , 3 , 4 , 5
  Cela représente une difficulté pour réaliser notre script car pour la numérotation des résidus on va avoir affaire à deux types de données : des entiers, et des chaînes de caractères.
- Des résidus manquants (par exemple il manque les résidus 148 et 149 de la chaîne H) il faut en tenir compte dans notre nouvelle numérotation en réservant leur numéro respectifs.
- Des résidus non standards ou des résidus modifiés pour lesquels il faut aller chercher le résidu acide aminés standard auquel il correspond. Cette information peut être retrouvée dans la partie MODRES.

Le fichier de test 1a2c.pdb a été téléchargé sur le site de la banque de données de protéine PDB *http://www.rcsb.org/pdb/home/home.do* .

## a. Le script

Le fichier PDB est divisé en plusieurs parties dont deux nous intéressent:

- La partie « fileheader » ou entête du fichier contenant des informations sur la structure tel que le nom, la séquence « SEQRES », les résidus modifiés « MODRES », les chaînes qui constituent la structure …
- La deuxième partie « atomInfo » qui contient les informations sur les atomes composant les résidus qui sont représentés avec les modèles qu'on a, le numéro attribué à chaque résidu ainsi que la chaîne à laquelle il appartient, les coordonnées de chaque atome dans l'espace…

Pour répondre à la question posée lors de cette première partie, notre script va contenir **deux grandes parties** :

- **Récupérer la séquence de la structure:** On commence notre script par sélectionner l'entête du fichier PDB en utilisant la fonction «getProperty('fileHeader')» et les stocker dans une variable.
  Ensuite on va chercher dans la partie « SEQRES » de la variable en tête où l'on a stocké les identifiant des chaînes constituant la structure.
  Pour chaque chaîne, les acides aminés constituant sont concaténés dans un tableau qu'on a appelé « chaine_courante ».
  On a voulu en même temps afficher les résidus modifiés ainsi que les résidus d'origine y correspondant. Il a été pour cela nécessaire d'aller chercher les résidus modifiés dans la partie « MODRES ».

A ce niveau là, notre script nous permet d'obtenir un tableau pour chaque chaîne contenant sa séquence en acides aminés.

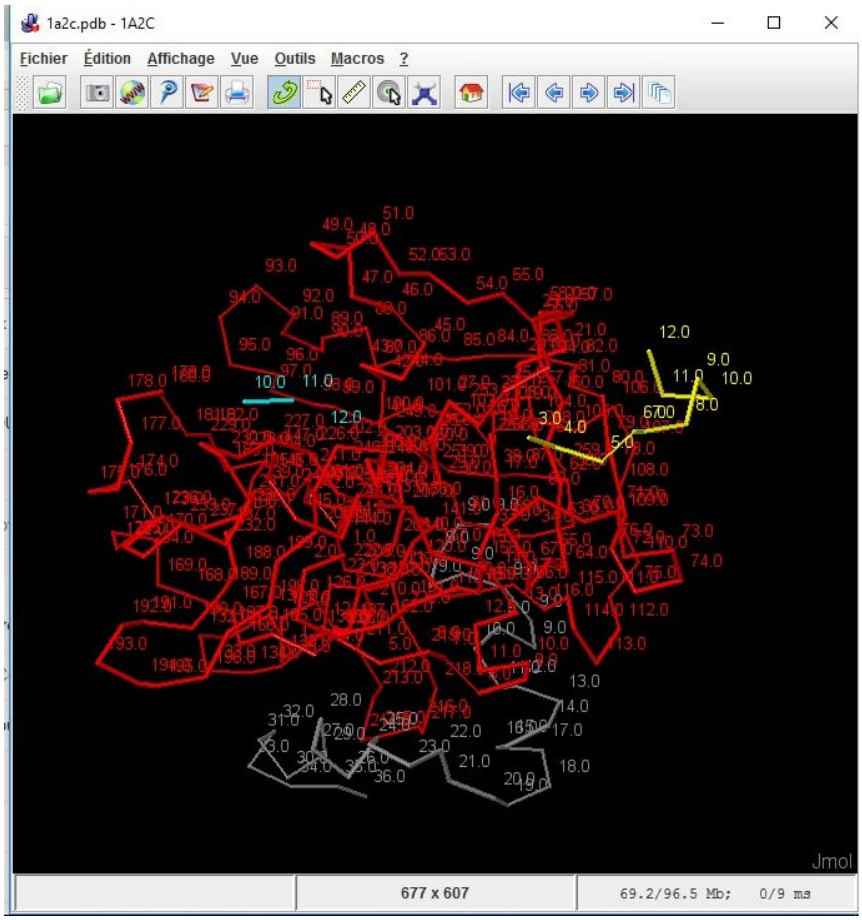
- **Attribuer une nouvelle propriété atomique « property_seqresno » :** Dans cette deuxième partie, pour chaque chaîne on se sert de la fonction " getProperty('chainInfo') " de jmol qui nous renvoie une structure qui contient les informations atomiques de tous les résidus de la structure macromoléculaire.

Ainsi nous obtenons d'une part un tableau associatif des résidus de chaque chaîne (SEQRES) et d'autre part un second tableau associatif des résidus de la structure PDB. En parcourant ces deux tableaux simultanément, on arrive à comparer séquentiellement les résidus et à définir ainsi une propriété "property_seqresno" (numérotation numérique et continue des résidus) en fonction des cas.

## b. Résultats obtenus sur modèles réels

Afin d'avoir un aperçu de l'usage du property_seqresno, nous avons utilisé le fichier 1a2c.pdb et avons affiché pour nos chaînes les property_seqresno. On peut voir sur la *capture 1* ci-dessous, la nouvelle numérotation de la structure qui apparaît sur l'ensemble

des chaînes. On peut voir pour chaque chaîne que la numérotation commence à 1 et qu'elle prend en compte les résidus manquants et les résidus modifiés.




*Capture 1 : Affichage de property_seqresno obtenu sur la protéine 1a2c; Chaîne 'H' en rouge ; 'I' en jaune ; 'J' en cyan ; 'L' en gris.*

*Code utilisé dans la console Jmol pour afficher 1a2c pour la Capture 1 :*

```
script projet_Diallo_Passerieux_Zahidi.jmol
load 1a2c.pdb;
initialization();numbering();
select all; backbone -0.2
select {chain='H'}; color red;
select {chain='J'}; color yellow;
select {chain='I'}; color cyan;
select all; select{atomname='CA'};
label %[property_seqresno]
```

# Partie 2/4

Dans cette partie, nous disposons d'un résultat d'alignement BLAST d'une séquence requête contre une molécule (accession) de la PDB. Le but ici est de faire correspondre les position des alignements à la nouvelle numérotation qu'on a effectuée dans la partie 1.

De ce fait, on devra définir pour chaque résidu de la structure alignée contre la structure requête une nouvelle propriété atomique « **property_qresno** » telle que :
Si le résidu ne fais pas parti du segment aligné la propriété property_qresno vaut "-1".
Si le résidu est aligné avec un indel (gap), property_qresno = 0 .

Pour mesurer la similarité on doit définir également une autre nouvelle propriété atomique "**property_qsimil**" telle que:

- property_qsimil = -1 si le résidu est non aligné.
- property_qsimil = 0 si le résidu aligné est dissimilaire.
- property_qsimil = 1 si le résidu aligné est similaire.
- property_qsimil = 2 si le résidu aligné est identique.

On commence d'abord par chercher les fichiers PDB des différentes structures qu'on va étudier dans cette partie à savoir : 2kfu, 2pie, 3qgs, 2fez, 3oun. (*http://www.rcsb.org/pdb/home/home.do* ).

## a. Le script

Après avoir chargé sur jmol le fichier résultat de l'alignement BLAST, on définit comme précédemment la propriété « property_seqresno » pour les résidus de chacune des chaînes de la structure alignée avec notre structure requête.

Ensuite à partir de la fonction « getProperty('chainInfo') » on parcourt les résidus de chaque chaîne en stockant au passage leur nouvelle numérotation (property_seqresno).

Ainsi nous avons adapté la numérotation des positions d'alignements en faisant la soustraction de la position de début du HSP "hit" à la position de début du "HSP" requête, auquel on additionne la propriété property_seqresno.

Pour la similarité (property_qsimil), on parcourt juste le tableau contenant le résultat "hit" fourni et on en vérifie la valeur.

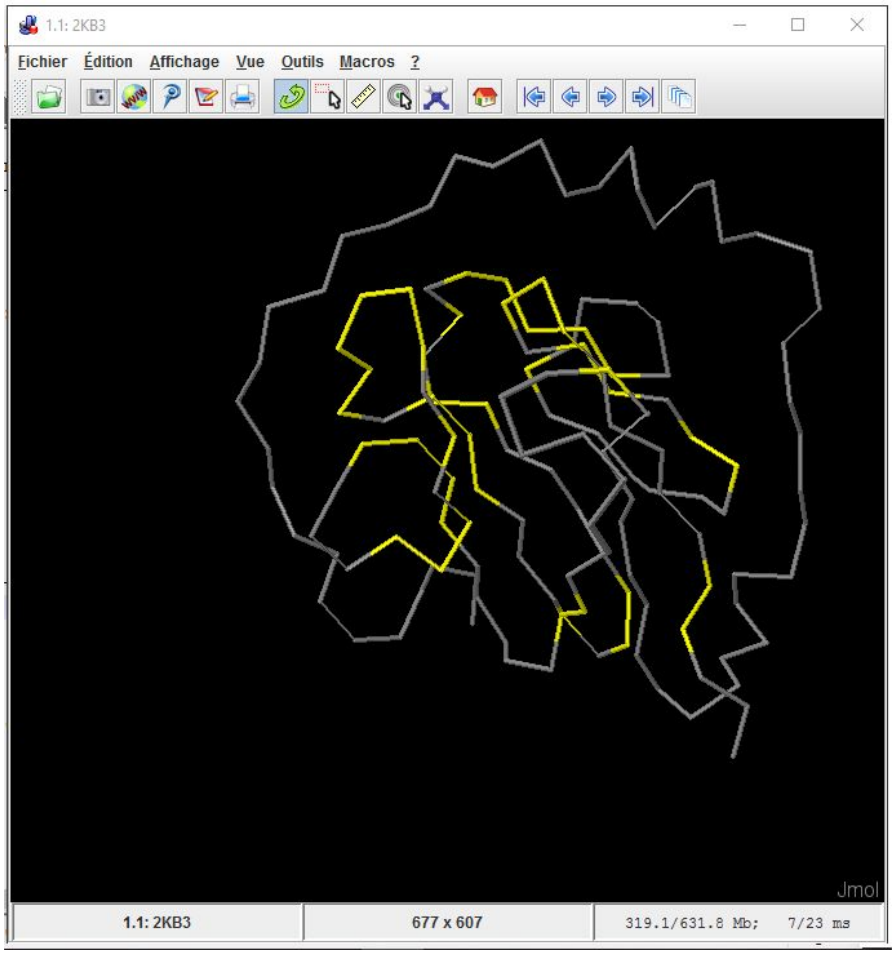
- "-" signifie que c'est un gap
- "+" la présence d'une similarité
- " " la présence d'une dissimilarité
- sinon ce sera une lettre qui indique que les résidus sont identiques.

La difficulté qu'on a rencontré lors de cette partie est le problème des indel (gap) que l'on n'arrive pas à prendre en compte.

## b. Résultats obtenus sur des modèles réels

Une fois notre script établi nous avons pu le tester sur les différents résultats d'alignements blast qui nous ont été fournis.

Nous avons décidé pour cela d'utiliser le fichier **2KB3.pdb** sur lequel est appliqué le property_qresno afin de déterminer les parties qui s'alignent avec le blast ( 0 correspondant à un résidu aligné et -1 à un résidu non aligné) et utiliser le fichier **BLAST_result_02.jmol** où nous avons appliqué le property_qsimil (revoyant les acides aminés similaires, identiques et les gap) . Pour notre capture, nous avons décidé de représenter uniquement les property_qsimil = 2 (résidus identiques) , qui, sur la représentation 3D sont en jaune.
Les parties ne correspondant pas à property_qsimil = 2 , et donc représentant le reste de la chaîne 2KB3 sont de couleur grise.




*Capture 2 : Application du property_qsimil ainsi que du property_qresno afin de déterminer les acides aminés identiques entre 2KB3 sur le "BLAST_result_02.jmol". En jaune les a.a[1] identiques; en gris le reste de la structure 2KB3.*

[1] a.a = acides aminés

*Code effectué pour obtenir capture 2 :*

```
script projet_Diallo_Passerieux_Zahidi.jmol
load 2kb3.pdb;
initialization(); numbering(); alignment();
select all; backbone -0.2
select{atomname='CA' and property_qsimil=2};
color yellow;
```

# Partie 3/4

Dans la troisième partie il est question de rédiger un script afin de superposer deux structures à partir de l'alignement de chacune des séquences avec la même séquence requête. C'est à dire que l'on a deux structures A et B, le résultat de l'alignements *"BLAST"* de ces structures contre une séquence d'intérêt. Le but est de superposer A et B en se basant sur les régions de l'alignement similaires entre A et B. A partir de là, il est alors possible de créer des paires de type {Ai,Bi} avec i = 1...n, correspondant aux aa alignés avec la séquence requête.
Afin de réaliser cette partie il faut tout d'abord choisir quels résidus nous voulons utiliser pour nos comparaisons. Pour cela plusieurs solutions: la première, est de choisir de prendre en compte la totalité des résidus alignés y compris les dissimilaires, la deuxième étant de considérer seulement les résidus identiques et similaires et la troisième de ne considérer que les résidus identiques (ces paramètres sont définis par property_qsimil dans la partie 2).

Passons maintenant au déroulement de notre projet. Dans notre cas, nous avons choisi de prendre en compte les résidus identiques et similaires.

## a. Script

La script à été mis en place de la manière suivante: on commence par charger les fichiers PDB ainsi que les alignements BLAST de nos deux structures que nous souhaitons superposer. On fait alors tourner le script mis en place dans la partie 1 et celui de la partie 2 afin de définir les propriétés "property_seqresno", "property_qresno" et "property_qsimil" pour chacunes des structures. On crée ensuite un tableau dans lequel on stocke les valeurs property_qresno des résidus situés au niveau de la région alignée avec la séquence requête, à savoir les résidus ayant une propriété "property_qsimil" d'une valeur 0 , 1 ou 2.
Ceci permet d'obtenir la liste des "property_qresno" des a.a. alignés pour chaque structure.
Par contre nous obtenons un résultat non valide de la fonction "compare" que nous ne sommes pas parvenu à corriger. Les paramètre Ai et Bi correspondant à la position des a.a.

La commande utilisée est la suivante:

```
compare{1.1}{1.2} SUBSET {*.CA} ATOMS {Ai} {Bi} ROTATE TRANSLATE
```

### b. Résultats obtenus sur modèles réels

**RMSD NaN --> NaN Angstroms**

## Partie 4/4

Cette partie consistait à faire des tests sur plusieurs types de données qui ont été fournies au préalable. Chose que nous avons faite pour les parties 1, 2 et 3. Le script reste fonctionnel même s'il existe des améliorations que nous aurions pu apporter. Notamment sur la gestion des gaps dans les alignements.

# Conclusion

Ce projet très intéressant nous a permis d'intégrer une nouvelle thématique dans notre discipline qui est la Bioinformatique : la modélisation de molécules biochimiques. Nous avons pu apprendre le stockage des données dans des bases de données de molécules, la structure des fichiers pdb ainsi que le fonctionnement du logiciel JMOL et son langage de script. Ceci nous a permis de prendre en main le problème et à bien mener ce projet.