TP de Manipulation de Fichiers PDB

Documentation concernant le format des fichiers PDB :
http://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html

Guide : Vous pouvez utiliser et modifier le programme *lecturePDB.py*. Vous testerez vos fonctions et programmes sur les PDBs suivants : 3PDZ et 1FCF.

A. Comparaison de deux structures protéiques

(1) Ecrire une fonction qui calcule le RMSD entre deux régions de deux structures de protéines *P1* et *P2* alignées. Les paramètres d'entrée sont:
- *P1 et P2* : deux listes d'atomes (C-alpha) représentant les deux protéines
- *sel_P1* et *sel_P2 :* deux listes de numéros de résidus représentant les régions d'intérêt. On suppose que la mise en correspondance a déjà été faite, et donc les deux listes ont la même longueur.

(2) Modifier le programme principal en conséquence.

(3) La structure 1FCF a été alignée sur 3PDZ avec Pymol, via la commande **align** (alignement basé sur la séquence). Télécharger le fichier correspondant *1fcf_aliSeq.pdb*. Calculer le RMSD entre *3pdz.pdb* et *1fcf_aliSeq.pdb*, en prenant comme sélections de résidus les résidus alignés dans la figure ci-dessous (3PDZ en rouge, 1FCF en bleu). Quelle est la valeur du RMSD ?

```
    21    26    31    36    41          46    51    56    61    66    71
----SVTGGVNTSVRHGGIYVKAVI-----PQGAAESDGRIHKGDRVLAVNGVSLEGATHKQ
 156   161   166   171   176   181   186   191   196   201   206   211
GTAGSVTG-VGLEITYDGGSGKDVVVLTPAPGGPAEKAGA-RAGDVIVTVDGTAVKGLSLYD
```

(4) Ouvrir les 2 fichiers PDB dans Pymol. Aligner 1FCF sur 3PDZ avec la commande **super** (alignement structural). Que constate-t-on ? Quelle est la valeur de RMSD correspondante ? Sélectionner les résidus 21 de 3PDZ et 159 de 1FCF pour les visualiser sur les structures. Sont-ils superposés ? Que peut-on en déduire ?

B. Cartes de contacts

(1) Ecrire une fonction qui calcule toutes les distances entre deux sélections de résidus dans une protéine, en considérant uniquement les C-alpha, et affiche le résultat sous forme de matrice colorée.

Aide : Pour l'affichage de la matrice, le module pylab et la fonction pcolor peuvent être utilisés.

(2) Proposer une mesure de dissimilarité entre deux cartes de contact et écrire une fonction qui calcule la dissimilarité entre deux cartes de contact données en entrée.

C. Variance circulaire

La variance circulaire est une mesure géométrique qui rend compte de l'enfouissement des résidus dans les structures de protéines. Etant donné un atome *i*, la variance circulaire de *i* est égale à 1 moins la résultante des vecteurs qui partent de *i* et pointent vers les autres atomes de la protéine, dans un rayon r_c :

$$CV(i) = 1 - \frac{1}{n_i} \left| \sum_{j \neq i, r_i \leq r_c} \frac{\vec{r_{ij}}}{\| \vec{r_{ij}} \|} \right|$$

Si la résultante est nulle, alors l'atome est enfoui dans la protéine ; si la résultante est grande, alors l'atome est protubérant. Pour déterminer le niveau d'enfouissement d'un résidu, on calcule la moyenne des variances circulaires des atomes qui le composent.

(1) Ecrire une fonction qui calcule la valeur CV pour chaque résidu d'une protéine, étant donné un rayon donné en entrée. Typiquement, le rayon doit être supérieur ou égal à 20 Å.

(2) Ecrire une fonction qui détermine les x% de résidus les plus enfouis, et les x% les plus protubérants d'une protéine, étant donné un rayon donnée en entrée.

(3) Ecrire une fonction qui lit un fichier PDB et des valeurs associées à chaque résidu ou chaque atome, et écrit un fichier PDB avec les valeurs données en entrée dans la colonne des B-facteurs (12ème colonne). Utiliser cette fonction pour visualiser les valeurs de CV sous pymol

(4) Comparer les valeurs CV des chaînes A et B de la structure 2BBM, quand on considère chaque chaîne séparément, ou bien le complexe entier.

D. Champ de force

Un champ de force est une expression analytique (ou fonctionnelle) qui représente les interactions inter-atomiques d'un système moléculaire. Il permet d'estimer l'énergie mécanique moléculaire d'une protéine ou d'un complexe, dans le vide ou solvaté.

Pour une protéine *P*, dans le vide, l'énergie est exprimée comme :
$E_{tot}(P) = E_{bonded}(P) + E_{non-bonded}(P, P) = E_{int}(P)$

Où E_{bonded} correspond à l'énergie d'interaction entre les atomes liés covalemment et $E_{non-bonded}$ correspond à l'énergie d'interaction entre les atomes non liés (distant de plus de 4 atomes).

Pour un complexe formé par deux protéines *R* et *L*, dans le vide, l'énergie totale du système peut s'exprimer comme :

$$E_{tot}(R \bullet L) = E_{int}(R) + E_{int}(L) + E_{non-bonded}(R, L)$$

L'énergie d'interaction associée à la formation du complexe vaut :

$$\Delta E_{inter}(R \bullet L) = E_{tot}(R \bullet L) - E_{tot}(R) - E_{tot}(L) = E_{non-bonded}(R, L)$$

avec :

$$E_{non-bonded}(R, L) = \sum_{i,j} \frac{A_{ij}}{R_{ij}^8} - \frac{B_{ij}}{R_{ij}^6} + f \frac{q_i q_j}{20 R_{ij}}$$

où *i* et *j* sont les indices des atomes de *R* et *L*, et *f=332.0522*. Voir le fichier `ForceField.py` (cf Cornell *et al.* 1995) pour les paramètres.

(1) Ecrire une fonction qui calcule l'énergie d'interaction entre deux chaînes contenues dans un fichier PDB.

(2) Calculer l'énergie d'interaction entre la calmoduline et son peptide cible à partir du fichier 2BBM.

D. Elastic network model

Let us consider a molecular system with N atoms at an equilibrium position $q_0 \in \mathbb{R}^{3N}$. Let $V : \mathbb{R}^{3N} \mapsto \mathbb{R}$ be the potential energy of the molecular system. Let us also introduce $q \in \mathbb{R}^{3N}$, a small time-dependent displacement of the system around q_0. The potential energy V in the vicinity of q_0 can thus be given by its quadratic approximation, which allows to analytically solve the Newton's equation of motion,

$$M(\ddot{q} + \ddot{q_0}) + \nabla V(q_0 + q) \approx M\ddot{q} + Hq = 0,$$

where M is the diagonal mass matrix, and H is the Hessian matrix of the potential energy V evaluated at the equilibrium position q_0. We then compute the square matrix of eigenvectors L and the diagonal matrix of eigenvalues Λ of the mass-weighted Hessian $H_w = M^{-1/2} H M^{-1/2}$,

$$H_w = L \Lambda L^T.$$

Let us now introduce $\eta \in \mathbb{R}^{3N}$, a projection of q into the eigenspace of H_w, and $(\lambda_i)_{i=0\ldots3N}$, the diagonal values in Λ. Then, left multiplying the above mentioned equation of motion by $L^T M^{1/2}$ gives the following system of uncoupled equations,

$$\eta = L^T M^{1/2} q,$$
$$\ddot{\eta_i} + \lambda_i \eta_i = 0 \quad i = 1\ldots3N,$$

which can be solved analytically. The columns of the $M^{-1/2}L$ matrix are the Cartesian linear normal modes of the system. We should specifically mention that these normal modes are not generally orthogonal, unless all the masses in M are equal to each other.

Classical NMA methods can use any potential function, provided that it corresponds to the equilibrium position of the molecular system. A common practice is to use a harmonic potential, which applies between any two proximal pairs of atoms in the system. In other words, if two atoms are sufficiently close to each other, their interaction is modeled by a spring. The potential can then be expressed as

$$V(q) = \sum_{d_{ij}^0 < R_c} \frac{\gamma}{2}(d_{ij} - d_{ij}^0)^2,$$

where d_{ij} is the distance between the i^{th} and the j^{th} atoms, d_{ij}^0 is the reference distance between these atoms, as found in the original structure, γ is the spring constant, and R_c is a cutoff distance, typically between 3.5 and 15 Angstroms.

The Hessian matrix corresponding to this potential function is composed of the following super-elements,

$$H_{ij} = -\frac{\gamma}{(d_{ij}^0)^2} \begin{pmatrix} (x_{ij}^0)^2 & x_{ij}^0 y_{ij}^0 & x_{ij}^0 z_{ij}^0 \\ y_{ij}^0 x_{ij}^0 & (y_{ij}^0)^2 & y_{ij}^0 z_{ij}^0 \\ z_{ij}^0 x_{ij}^0 & z_{ij}^0 y_{ij}^0 & (z_{ij}^0)^2 \end{pmatrix} \quad i \neq j$$

$$H_{ii} = -\sum_{j \neq i} H_{ij},$$

Further simplification of the system can be achieved by considering only C-alpha carbons and by assuming that their fluctuations are isotropic and Gaussian. This way, the $3N \times 3N$ Hessian matrix is replaced by a $N \times N$ Kirchhoff matrix, with the elements:

$$\Gamma_{ij} = \begin{cases} -\gamma * & \text{if i} \neq \text{j and } d_{ij} \leq R_c \\ 0 & \text{if i} \neq \text{j and } d_{ij} > R_c \\ -\sum_{i \neq j} \Gamma_{ij} & \text{if i} = \text{j} \end{cases}.$$

(1) Write a function that creates an elastic network from a PDB structure. Each node is an alpha-carbon. Two nodes are connected if the corresponding alpha-carbons are closer than a cutoff value given as input. The function should create a new PDB file with only the C-alpha atoms and the connections saved using the CONECT keyword (see PDB documentation). It should output a Kirchhoff matrix representing the protein structure. By default you will consider a force constant of 1.0.

(2) Write a function that determines the eigenvectors of the Kirchhoff matrix, in other words, the normal modes of the protein structure. You can use :func:`scipy.linalg.eigh` from the Scipy module or :func:`numpy.linalg.eigh` from the numpy module to diagonalize the matrix.

(3) Write a function that deforms a structure along a normal mode up to a certain amplitude given as input. The function will output a multi-PDB file with 11 snapshots of the deformation trajectory (5 snapshots corresponding to the backward direction, the initial structure, and 5 snapshots corresponding to the forward direction).