

TP 1 : calcul d'énergie

Le but de ce TP est de vous introduire les bases du calcul d'une énergie empirique d'un système d'une ou plusieurs molécule(s), utilisées classiquement dans les programmes de modélisation moléculaire (comme CHARMM ou GROMACS par exemple). Vous trouverez les fichiers et des informations supplémentaires sur la page : <http://www.dsimb.inserm.fr/~fuchs/M1BI/TP1/>

Rappel de cours :

Ces programmes utilisent un *champ de forces*, c'est à dire une équation pour le calcul d'énergie associée à un ensemble de paramètres définissant toutes les constantes nécessaires à ce calcul (voir Annexe A et la page web).

L'équation du calcul d'énergie s'écrit :

$$E = E_{\text{liée}} + E_{\text{non liée}} = (E_{\text{liaison}} + E_{\text{angle}} + E_{\text{dièdre}}) + (E_{\text{vdw}} + E_{\text{elect}})$$

Systeme(s) étudié(s) :

Dans un premier temps, nous avons choisi d'étudier une molécule très simple, la molécule d'eau. Ensuite, nous allons complexifier ce système en introduisant une puis plusieurs autres molécules d'eau en proposant différentes orientations entre ces molécules. Enfin, nous terminerons la séance sur l'étude d'un acide aminé dont la chaîne latérale est relativement courte. A chaque fois, nous vous demanderons de faire une description exhaustive de chaque terme d'énergie du système étudié et de calculer les différents termes énergétiques à l'aide d'un script Python. Au fur et à mesure, nous concevrons des fonctions qui nous aiderons à effectuer ce calcul sur n molécules.

1) Système d'une molécule d'eau

Les fichiers *water.pdb* et *water_bis.pdb* contiennent les coordonnées atomiques d'une molécule d'eau. Vous pouvez visualiser ces molécules avec Rasmol dont l'utilisation est détaillée en annexe et sur la page web.

1.1 Faire le bilan des termes d'énergie.

1.2 Concevoir la fonction *dist()* qui calcule une distance euclidienne en Python (avec comme arguments deux listes qui contiennent les coordonnées de chaque atome). Vérifier votre résultat avec Rasmol. Concevoir une autre fonction *angle()* qui calcule un angle (arguments : 3 atomes, représentés par 3 listes de coordonnées). Les étapes nécessaires à la construction de cette fonction sont décrites en annexe. Vérifier votre résultat avec Rasmol .

1.3 Calculer l'énergie potentielle de chaque molécule d'eau (*water.pdb* et *water_bis.pdb*).

1.4 Comment expliquez-vous les valeurs obtenues ? Commenter les différences observées.

2) Système de deux molécules d'eau

Nous rajoutons maintenant une deuxième molécule d'eau (fichier *water2.pdb*).

2.1 Faire le bilan des termes d'énergie décrivant ce système. Quels sont les termes supplémentaires qui apparaissent ?

2.2 Concevoir une fonction *ene_bond()* qui calcule l'énergie d'une liaison entre deux atomes (arguments : distance entre les deux atomes, longueur de liaison à l'équilibre et constante de force associée). En utilisant cette fonction, calculer la somme des énergies de liaison du système à l'aide d'une boucle (généralisable à n molécules d'eau).

2.3 De même concevoir une fonction *ene_angle()* qui calcule l'énergie d'un angle de liaison entre 3 atomes (arguments : angle calculé, valeur de l'angle à l'équilibre et constante de force). En utilisant cette fonction, calculer la somme des énergies des angles du système à l'aide d'une boucle (généralisable à n molécules d'eau).

2.4 Concevoir une fonction *ene_bonded_water()* qui calcule l'énergie liée d'une molécule d'eau (arguments : une molécule d'eau, soit une liste de listes de coordonnées). En utilisant cette fonction, calculer ensuite, avec une boucle, la somme des énergies liées d'un système à n molécules d'eau. Effectuer le calcul pour *water2.pdb*.

2.5 Concevoir une fonction *Lennard_Jones()* qui renvoie l'énergie de van der Waals entre deux atomes (arguments : constantes E_{ps} pour chaque atome, constantes R_{min} pour chaque atome et distance entre les deux atomes).

2.6 En utilisant la fonction *Lennard_Jones()*, calculer l'énergie associée aux interactions de van der Waals.

2.7 De même, concevoir une fonction *Coulomb()* qui renvoie l'énergie électrostatique entre deux atomes i et j (arguments : charge de i et j , et distance entre les deux atomes).

2.8 En utilisant la fonction *Coulomb()*, calculer l'énergie associée aux interactions électrostatiques.

2.9 Concevoir les fonctions *Lennard_Jones_2waters()* et *Coulomb_2waters()* qui calculent respectivement l'énergie de van der Waals et l'énergie électrostatique entre deux molécules d'eau (arguments : deux molécules d'eau représentées sous forme de listes de listes de coordonnées). Implémenter ensuite la fonction *calc_ene()* qui calcule l'énergie potentielle du système en appelant les fonctions *ene_bonded_water()*, *Lennard_Jones_2waters()* et *Coulomb_2waters()*. Effectuer le calcul pour le système étudié.

3) Système de trois molécules d'eau

Nous avons maintenant deux systèmes de trois molécules d'eau (fichiers *water3.pdb* et *water3_bis.pdb*). Le but de cette étape est d'automatiser le calcul d'énergie en vue de le généraliser à n molécules d'eau. Dans un premier temps, on se concentrera sur *water3.pdb*.

3.1 Calculer les termes d'énergie liée comme dans le 2).

3.2 Calculer l'énergie associée aux interactions non liées : utiliser pour cela une double boucle imbriquée sur chaque paire de molécule d'eau. Dans le corps de cette double boucle, il suffira de calculer les interactions sur chaque paire d'atomes entre les deux molécules d'eau correspondantes.

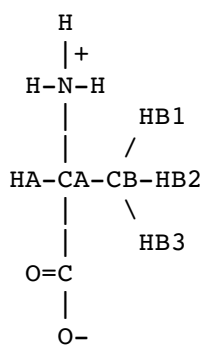
3.3 Calculer l'énergie potentielle du système.

3.4 Refaire le calcul pour le fichier *water3_bis.pdb*. Comment expliquer les valeurs obtenues ? Commenter les différences observées.

3.5 Ouvrir maintenant ces deux fichiers sous Rasmol. Quelles remarques pouvez-vous faire sur ces deux systèmes ? Tester maintenant votre programme avec le fichier *water10.pdb*.

4) Système d'une alanine

Nous vous proposons d'appliquer ce calcul d'énergie sur un acide aminé *simple* l'alanine (cf fichier *ala.pdb*). Voici un rappel de sa structure chimique :



4.1 Regarder la structure avec Rasmol et faire le bilan de tous les termes d'énergie décrivant cette molécule.

4.2 Quel type de terme supplémentaire apparaît maintenant ? Comment feriez-vous pour le calculer ?

ANNEXES

A Paramètres du champ de forces pour une molécule d'eau (Charmm) :

- Energie de liaison :
Distance à l'équilibre OH-H : $l_{eq} = 0,9572 \text{ \AA}$
Constante de force : $kbond = 450,000 \text{ kcal.mol}^{-1}.\text{\AA}^{-2}$
- Energie d'Angle :
Angle à l'équilibre : $theta_{eq} = 104,5200 \text{ deg}$
Constante de forces: $kangle = 55,000 \text{ kcal.mol}^{-1}.\text{rad}^{-2}$
- Energie de Lennard-Jones :
Profondeur du puits pour les atomes i ou j (en kcal.mol^{-1}) = Eps
Distance à laquelle l'énergie de LJ est minimale (en \AA) = Rmin
EpsHH = 0,0460 kcal.mol^{-1} EpsOH = 0,0836 kcal.mol^{-1} EpsOO = 0,1521 kcal.mol^{-1}
RminHH = 0,4490 \AA RminOH = 1,9927 \AA RminOO = 3,5364 \AA
- Energie de Coulomb
Constante diélectrique : $EPS0 = 1$ (pour le vide)
Charge partielle (unité e , avec e charge électronique = $1,6 \times 10^{-19}$ coulomb)
 $Q(\text{OH}) = -0,834$; $Q(\text{H}) = 0,417$
Facteur de conversion $e : f = 1/(4 \times \pi \times EPS0) = 332,0716 \text{ kcal.\AA.mol}^{-1}e^{-2}$

B Rappels de mathématiques

- Calcul d'un angle :
Rappel : produit scalaire (<http://hyperphysics.phy-astr.gsu.edu/hbase/vsca.html>)
Fonction *angle()* : on lui passe en argument les coordonnées des 3 atomes (OH2, H1, H2) sous forme d'une liste de listes de coordonnées :
[[OH2x, OH2y, OH2z], [H1x, H1y, H1z], [H2x, H2y, H2z]]
Par exemple, pour obtenir l'angle θ (angle H1-OH2-H2) dans une molécule d'eau, il faut combiner les deux expressions suivantes :
(1) $v(A).v(B) = ||A||.||B|| \cos \theta$
(2) $v(A).v(B) = AxBx + AyBy + AzBz$
où $v(A)$ et $v(B)$ indiquent les vecteurs A et B, $||A||$ et $||B||$ correspondent à la norme de ces vecteurs, et (Ax,Ay,Az) et (Bx,By,Bz) sont les coordonnées de ces vecteurs.
- Calcul d'un dièdre :
Rappel : produit vectoriel (<http://hyperphysics.phy-astr.gsu.edu/hbase/vvec.html>)

C Quelques conseils pour calculer les énergies des molécules d'eau en Python

- Représentation du système étudié :

Chaque molécule d'eau est représentée par un tableau (*MLC*) à deux dimensions (liste de listes). Les atomes (OH2, H1 et H2) constituent la 1^{re} dimension et les coordonnées de chaque atome constituent la 2nd :

```
[ [OH2x, OH2y, OH2z], [H1x, H1y, H1z], [H2x, H2y, H2z] ]
```

Le système complet (plusieurs molécules d'eau, *LISTMLC*) peut être modéliser par un tableau à 3 dimensions (liste de listes de listes de coordonnées ou liste de molécules).

- Autres fonctions utiles (fournies):

```
# cette fonction prend en argument les lignes d'un fichier pdb
# et renvoie les coordonnees des 3 atomes de la molecule d'eau
# specifiee par waternb
# pdblines est un liste de lignes
# waternb est une chaine de caracteres
def get_water_coordinates(pdblines,waternb):
    ...
```

```
# fonction renvoyant le produit scalaire
# de 2 vecteurs (de 3 coordonnees)
def scalar(A,B):
    return (A[0]*B[0]) + (A[1]*B[1]) + (A[2]*B[2])
```

```
# fonction renvoyant la norme d'un vecteur
# (de 3 coordonnees)
def norme(A):
    return math.sqrt(A[0]**2+A[1]**2+A[2]**2)
```

```
# fonction qui transforme des radians en degres
def rad2deg(ang):
    return ang*(180/math.pi)
```

```
# fonction qui transforme des degres en radians
def deg2rad(ang):
    return ang*(math.pi/180)
```

D Rappels de Rasmol

<http://rasmol.org/doc/rasmol.html>

http://www.chemistry.wustl.edu/~taylor/RasMol_Ref.pdf

- Représentation de la molécule :

cartoon <number>

backbone <value>

également : *cpk*, *wireframe*, *ribbons* ...

- Choix de couleurs :

color {<object>} <color>

atom object = cpk, amino, chain, group, structure, charge

couleurs prédéfinies = cyan, greenblue, orange, red, violet, yellow, green, magenta, purple, redorange, white

ex : *color cpk*

(Carbon=light grey, Oxygen=red, Hydrogen =white, Nitrogen=light blue, Phosphorus=orange)

color green

Modifier la couleur de fond : *background* <color>

- Calcul de distance, angle, dièdre :

Activer l'une des options :

set picking distance

set picking angle

set picking torsion

Puis, cliquer sur le nombre d'atomes nécessaires au calcul

- Sélection

select {<expression>}

ex : *select all*

select hydrogen

select Arg

- Affichage

label {<string>}

E Rappel sur le format d'un fichier PDB

Dans un fichier PDB, le format des lignes qui comportent les coordonnées atomiques est indiqué ici :

<http://www.wwpdb.org/documentation/format32/sect9.html>

et rappelé ici :

http://www.dsimb.inserm.fr/~poulain/misc/pdb_format.pdf

Dans le cas des molécules du TP (par exemple *water.pdb*) :

1	2	3	4	5	6	7	8	9	10
ATOM	1	OH2	TIP3	1	-0.765	-0.097	0.010	1.00	0.00
ATOM	2	H1	TIP3	1	-0.222	0.691	0.024	1.00	0.00
ATOM	3	H2	TIP3	1	-0.138	-0.819	-0.035	1.00	0.00

champ n°1 : ATOME ou HETATOM (hétéro-atome)

champ n°2 : numéro de l'atome

champ n° 3 : nom d'atome

champ n° 4 : nom de la molécule

champ n° 5 : numéro de la molécule

champ n° 6, 7, 8 : coordonnées x, y, z

champ n° 9 : occupation

champ n° 10 : facteur de température