

KHADRAOUI Mohamed El Bachir

EL BAHRAOUI Imade

DOCUMENTATION TECHNIQUE

Le projet est constitué de 4 fichiers :

- main.f90 qui fait office de programme principal depuis lequel on fait les exécutions
- atom_type.f90 qui définit le type atom et qui définit les fonctions relatives à celui-ci
- molecule_type qui définit le type molecule et qui définit les fonctions relatives à celui-ci
- vdw_type qui définit le type vdw

atom_type.f90 : attributs :

- element : élément de l'atome
- coordinates : coordonnées de l'atome
- radius : rayon de l'atome

atom_type.f90 : fonctions :

- init_atom(a, elt, coordinates) : initialise a de type atome en prenant les paramètres element et coordinates
- print_atom(a) : affiche les coordonnées de l'atome a

molecule_type.f90 : attributs :

- atoms : tableau d'atomes
- nb_atoms : le nombres d'atomes de la molécule

molecule_type.f90 : fonctions :

- print_mol(m): affiche les coordonnées de chaque atome de la molécule m
- read(m, filename) : lit un fichier au format xyz et stocke les informations dans la molécule m
- set_radius(m, vdw_array) : définit chaque atome de la molécule avec le rayon approprié obtenu depuis un tableau Van Der Waals
- box(m, ligand, volume) : prend un ligand et une molécule et forme une boîte minimale contenant les deux et retourne le volume de cette boîte
- write(m, filename) : prend une molécule de type molecule et l'écrit dans un fichier au format xyz

vdw_type.f90 : attributs :

- atom_name : nom de l'élément atomique
- radius : rayon de l'élément atomique

vdw_type.f90 : fonctions :

- init_vdw(atvdw, name, rad) : initialise un objet atom_vdw
- print_vdw(atomvdwa) : affiche l'objet atom_vdw (élément + rayon)
- read_vdw_file(atomvdwa, fileName, vdw_atom_arrays) : lit un fichier au format vdw et stocke les informations d'atome et de rayon dans un tableau
- get_atom_radius(atomvdwa, atom_n, vdw_atom_array, actual_radius) : prend un nom d'élément d'atome et retourne son rayon

main.f90:

- Lecture des fichiers de ligand, molécule et VDW
- Vérification de la collision ou non entre le ligand et la molécule cible
- Création d'un set de ligands après application de matrices de rotation et de translation en utilisant rand() pour générer des deltas aléatoires pour la translation mais aussi pour faire varier l'angle teta de la rotation
- Vérification de la collision ou non entre le set de ligand et la molécule cible puis filtrage en gardant seulement les ligands pour lesquels il n'y a pas collision

- Recherche de la meilleure solution parmi les ligands sans collision en se basant sur le volume de boîte le plus petit
- Ecriture dans un fichier au format xyz du meilleur ligand

Temps d'exécution selon le nombre de ligands générés par rotation et translation :

Nombre de ligands générés	Temps d'exécution
10	0.01562s
20	0.03125s
50	0.04688
1000	0.3s
10000	3.03125s
100000	30.17s

KHADRAOUI Mohamed El Bachir a travaillé sur :

- atom_type
- molecule_type
- vdw_type
- Les déclarations initiales du main
- Le débogage du main

EL BAHRAOUI Imade a travaillé sur :

- Le calcul de distance
- Vérification de collision
- Définition de la box
- Recherche de la meilleure solution