

MASTER BIBS M1 - 2024
Programmation Python Séance 2

Objectifs : manipulation des dictionnaires. Organisation et manipulation de structures de données complexes. Manipulation des fonctions et modules. Application à la prise en main du « parser » de fichiers PDBs.

Exercices de base

Ouvrez le fichier data1.txt. Comment est-il formaté, comment sont séparées les colonnes ?

1/ Ecrire un programme qui lit le fichier et crée un dictionnaire où chaque clé correspond à l'identifiant des gènes (colonne 1) et pointe vers l'identifiant du chromosome (colonne 2).

2/ Modifiez votre script pour que chaque clé pointe vers la localisation cellulaire du produit protéique de chaque gène (colonne 4).

3/ Créez trois listes correspondant aux différentes localisations cellulaires possibles (cytosol, nucleus et cell membrane). Modifiez ensuite votre script de façon à parcourir le dictionnaire que vous venez de créer et stocker les gènes (i.e. leur identifiant) en fonction de la localisation de leur produit protéique. Pour cela, on vous demande de stocker dans les trois listes que vous venez de créer, les identifiants des gènes correspondants d'après leur localisation.

4/ Créez un dictionnaire de dictionnaire (pas manuellement mais en « parsant » automatiquement votre fichier) où chaque clé correspond à l'identifiant du gène et pointe vers un sous-dictionnaire lui-même composé de quatre clés « Expression », « Localisation », « fonction » et « Chromosome ». Ces clés pointeront vers les valeurs correspondantes dans le tableau.

Votre dictionnaire aura cette forme :

```
d_genes = {  
    "gene283245": {"Expression": 0.8, "Localization": "cytosol", "function":  
        "nuclease", "Chromosome": "chr_V"},
```

```
"gene10098": {"Expression": 0.9, "Localization": "nucleus", "function":  
"Histidine kinase", "Chromosome": "chr_V"}  
...  
}
```

Manipulation du « parser » de fichiers PDBs

A partir de la fonction `parsePDBMultiChains()` et du fichier PDB `1brs.pdb`, créez un dictionnaire `dPDB` contenant les coordonnées atomiques de toutes les chaînes de `1brs.pdb`

Affichez le nombre de chaînes, le nombre de résidus total et le nombre de résidus par chaîne.

Combien y a-t-il de Lysines au total ? Pour chaque chaîne ?

Même question pour le nombre de Glutamines.

Ecrivez une fonction permettant de renvoyer le nombre d'un type d'acide aminé par chaîne (type d'aa à rentrer en argument).

Manipulation des coordonnées 3D

Conseil : utilisez pymol pour vérifier constamment que vos calculs sont corrects. C'est un très bon moyen pour déboguer !

Note : toutes vos fonctions seront stockées dans un module que vous créerez et que vous pourrez réutiliser par la suite. Donc pensez « flexible », codez « robuste » ;)

Ecrivez une fonction permettant de calculer la distance entre deux atomes.

Ecrivez une fonction utilisant la précédente, qui permet de calculer la distance entre deux résidus. Cette fonction utilisera la structure de `dPDB` et permettra de choisir entre deux modes de calcul : calcul de la distance la plus courte entre les deux résidus ou calcul de la distance entre les barycentres des deux résidus. Le calcul du barycentre d'un résidu se fera par une fonction indépendante qui prend en argument un résidu avec la structure de `dPDB` et qui renvoie les coordonnées (x,y,z) du barycentre.

Pour chaque chaîne identifiez la paire de résidus les plus éloignés et affichez la distance correspondante. Qu'en déduire à propos des différentes chaînes de `1brs` ?