TP6 – Recherche de chaînes de caractères

Le " find "

La méthode **find**, que nous avons présentée lors du premier TP, permet de rechercher une sous-chaîne dans une chaîne. Elle est très utile comme nous allons le voir au cours de ce TP :

- La méthode **find** s'applique sur des variables contenant des chaînes de caractères,

- La méthode **find** nécessite deux paramètres : le premier est la chaîne de caractères à rechercher ; le second le départ de la recherche. Si le second paramètre n'est pas spécifié, Python considère que la recherche commence à la position 0 (comme nous l'avons vu lors du premier TP),

- **find** renvoie la position du premier caractère de la première occurrence de la sous-chaîne rencontrée – à partir de la position spécifiée – dans la chaîne considérée ; la valeur de retour de la méthode **find** est –1 si la recherche n'est pas fructueuse.

- Exemple :

```
ch= "atggcatgtgg"

pos = ch.find("atg", 3)

print pos       # à l'écran s'affiche 5
```

Exercices illustrant la méthode find

Exercice 1 :

Ecrire un programme permettant de rechercher la position de la première occurrence (si elle existe) d'une courte séquence nucléique dans une séquence stockée dans un fichier. L'utilisateur sera invité à saisir la séquence à rechercher ainsi que le nom du fichier stockant la séquence à cribler.

Exercice 2 :

Ecrire un programme permettant de rechercher la position de **toutes les occurrences** d'une courte séquence nucléique dans une séquence stockée dans un fichier. Comme précédemment, l'utilisateur sera invité à saisir la séquence à rechercher ainsi que le nom du fichier stockant la séquence à cribler.

Exercice 3 :

Reprendre l'exercice précédent et générer un fichier (dont le nom sera spécifié par l'utilisateur) contenant les différentes positions des occurrences trouvées.

Les limites du " find " en biologie

En biologie, les motifs sont souvent " dégénérés " (flous) : leur " expression " sur une séquence peut varier autour d'une forme " canonique ". C'est le cas des régions promotrices, que l'on trouve en amont du codon d'initiation des gènes codant pour des protéines, qui sont impliquées dans la transcription. Dans ces conditions, la méthode `find` n'est pas d'une grande utilité.

L'objectif de la seconde partie de ce TP et du TP7 à venir va être de développer un programme permettant d'incorporer de la souplesse dans la recherche d'un motif. L'exercice qui suit a pour seul objectif de recoder un `find` offrant la possibilité, on le verra au cours du TP7, d'introduire " de la souplesse ".

Exercice 4 :

Ecrire un programme permettant de rechercher toutes les occurrences (et la position de chaque occurrence) d'une courte séquence nucléique dans une séquence nucléique stockée dans un fichier ***sans utiliser la méthode `find`***.

L'utilisateur sera invité à saisir la séquence à rechercher ainsi que le nom du fichier stockant la séquence à scanner. En sortie, on génèrera un fichier dans lequel on stockera les différentes positions trouvées.

Exercice complémentaire

Ecrire un programme calculant la longueur moyenne des séquences contenues dans un fichier (qui contient une séquence par ligne) ainsi que le nombre de séquences contenues dans ce fichier. Le nom de ce fichier devra être spécifié par l'utilisateur.