

2 Deuxième partie, ManipSeqADN.c

A partir de maintenant, vous travaillerez avec plusieurs .c. Pour l'instant vous aurez ManipSeqSimple.c, ManipSeqADN.c et Main.c. Les fichiers .h vous sont donnés. Il y a un .h pour chaque .c et en plus un fichier codons.h. Il faudra écrire les fonctions dont les prototypes sont dans le .h dans le fichier .c portant le même nom. Vous écrirez votre fonction main pour tester vos fonctions dans Main.c. Vous pourrez compiler votre programme avec un Makefile si vous le souhaitez.

Voici la structure permettant de manipuler des séquences d'ADN qui est définie dans le fichier ManipSeqADN.h : Elle contient donc la séquence (seq), sa longueur (lg) et sa composition en GC.

```
typedef struct {
    char *seq;
    int lg;
    float GC;
} tySeqADN;
```

2.1 Fonctions de manipulation de la structures de données tySeqADN

Vous aurez besoin de fonctions pour créer (allouer) et détruire (désallouer) chacune de ces structures. Ces fonctions seront écrites dans le fichier ManipSeqADN.c, les prototypes vous sont donnés dans textttManipSeqADN.h.

Vous écrirez donc :

1. `tySeqADN* newSeqADN()` ; Cette fonction alloue de la mémoire pour un élément de type `tySeqADN` et met les champs à 0, -1 si ce sont des indices ou à NULL. Cette fonction n'allouera pas l'espace pour la séquence elle-même (champs `seq`) ; il sera alloué lors de la lecture de la séquence dans un fichier.
2. `tySeqADN* freeSeqADN(tySeqADN *pS)` ; Cette fonction libère l'espace alloué pour la séquence si nécessaire et l'élément de type `tySeqADN`.
3. `tySeqADN* complémentaire(tySeqADN *pS)` ; qui à partir d'une séquence donnée alloue et génère la séquence du brin complémentaire et le retourne. Elle utilisera la fonction `Nt.Complémentaire`.

2.1.1 Lecture d'une séquence au format fasta

Vous écrirez la fonction `tySeqADN *readFasta(char *nomFi)` ; qui permet de lire une séquence d'ADN (un génome) au format fasta. Vous utiliserez les fonctions `fgets` et `sscanf`. Comme vous ne connaissez pas *a priori* la longueur de la séquence, vous allouerez progressivement la chaîne de caractères pour la stocker.

Un modèle main vous est donné dans le fichier Main.c. Vous regardez cette fonction car elle permet de donner des arguments au programme créé. Le programme créé se nommera `Detect_ORF` et il prend comme argument le nom du fichier fasta contenant le génome.