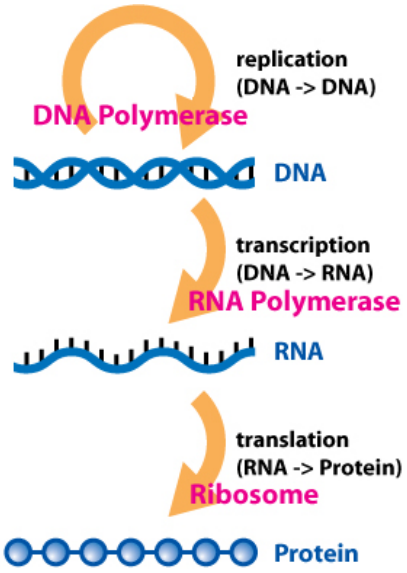
TP 6
Codage en génétique



Dans ce TP, les fonctions programmées vont permettre de reproduire le dogme central de la biologie moléculaire :

Les molécules représentées dans ce schéma sont :

- l'**ADN** : support stable et transmissible de l'information génétique.

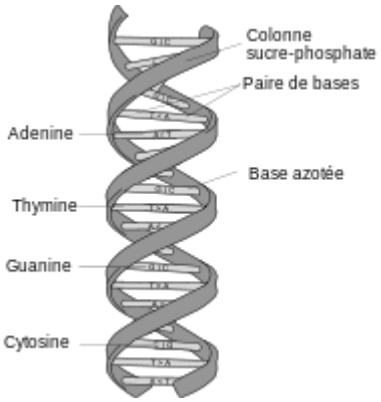
Il est formé des 4 nucléotides suivants (appelés aussi bases) :

A = adénine, T = thymine, G = guanine et C = cytosine.

Dans les cellules vivantes, l'ADN est sous la forme double brin, c'est-à-dire que 2 séquences ADN se font face.

Une séquence est lue de gauche à droite et l'autre de droite à gauche.
De plus, les bases complémentaires l'une de l'autre se font face :

A et T sont complémentaires
G et C sont complémentaires.



Un brin est donc complémentaire et inversé par rapport à l'autre.

- l'**ARN** : support temporaire permettant l'expression de l'information génétique.

Il est formé des 4 nucléotides suivants :

A = adénine, U = uracile, G = guanine et C = cytosine

- les **protéines** : outils de la cellule (enzymes, transporteurs, etc.).

Elles sont formées de 20 acides aminés différents.

Les processus du dogme central de la biologie moléculaire, réalisés par les cellules sont les suivants :

- la **transcription** : certaines parties spécifiques de l'ADN sont transcrites en ARN. La transcription consiste en l'assemblage de nucléotides ARN en suivant le modèle ADN et en prenant les bases complémentaires à savoir :
  le A de l'ADN est remplacé par un U dans l'ARN,
  le T par un A, le G par un C et le C par un G.

- la **traduction** : les ARN messagers sont traduits en protéines.

Le passage d'une séquence ARN composée de 4 nucléotides à une séquence protéique composée de 20 acides aminés, se fait à l'aide du code génétique.

Dans ce code, chaque mot de 3 bases, appelé codon, correspond à un acide aminé. Il est possible de construire 4^3 = 64 codons différents à l'aide des 4 bases.
Ce code est donc dégénéré : plusieurs codons correspondent au même acide aminé.
Les codons sont lus sans chevauchement, les uns à la suite des autres.

- la **réplication** : l'ADN de chaque brin d'une double hélice est recopié de telle sorte que deux nouvelles doubles hélices sont produites, identiques à l'unique double hélice qui a servi de modèle.

De simples chaînes de caractères permettent de représenter les séquences biologiques et les fonctions programmées vont reproduire les processus.

Les séquences ADN

Question 1

Réalisez une fonction nommée `**estADN**` qui vérifie si la chaîne de caractères passée en paramètre ne contient aucun autre caractère que les quatre bases A, C, G et T.
Cette fonction renvoie la valeur `**True**` si tel est le cas, et la valeur `**False**` dans le cas contraire. De plus, elle renvoie `**True**` si la chaîne est vide.

Exemples :

```
>>> estADN('ATGCGATC')
True
>>> estADN('ACKT')
False
>>> estADN('ACTK')
False
>>> estADN("")
True
```

Question 2

Il est possible de générer aléatoirement une séquence ADN. La version naïve suppose que les 4 bases ont la même probabilité d'apparaître à une position donnée.

Réalisez une fonction nommée `**genereADN**` qui renvoie une séquence ADN générée aléatoirement et dont la taille est passée en paramètre.

Exemple :

```
>>> genereADN(10)
'ACGCCGACTA'
```

La transcription

Pour rappel, dans l'ADN, les bases A et T sont complémentaires, ainsi que les bases G et C. Pour passer de l'ADN à l'ARN, le A est transformé en U, le T en A, le G en C et le C en G.

Question 3

Réalisez une fonction nommée `**baseComplementaire**` qui renvoie la base complémentaire de la base passée en paramètre, selon le type de séquence demandé en sortie qui peut être soit `'**ADN**'`, soit `'**ARN**'`.
Les contraintes d'utilisation de cette fonction sont que le premier paramètre est bien l'une des quatre bases de l'ADN (`A`, `T`, `G` ou `C`) et le deuxième est la chaîne soit `'**ADN**'`, soit `'**ARN**'`.

Exemples :

```
>>> baseComplementaire('G', 'ADN')
'C'
>>> baseComplementaire('A', 'ARN')
'U'
```

Question 4

Réalisez une fonction nommée `**transcrit**` qui renvoie l'ARN construit à partir de la sous-séquence d'ADN comprise entre les deux positions passées en paramètre, incluses. La première base de la séquence étant numérotée 1.
Pour mémoire, cet ARN est la séquence complémentaire de la portion d'ADN transcrite.
La fonction `**baseComplementaire**` doit être utilisée.

Exemple :

```
>>> transcrit('TTCTTCTTCGTACTTTGTGCTGGCCTCCACACGATAATCC', 4, 23)
'AAGAAGCAUGAAACACGACC'
```

La traduction

Question 5

Réalisez une fonction nommée `**codeGenetique**` qui renvoie l'acide aminé (sous la forme du nom abrévié EN 1 lettre) correspondant au codon passé en paramètre, ou pour les codons Stop.
Vous pouvez copier-coller le code génétique donné ci-dessous afin de faciliter l'écriture du code. N'oubliez pas les conditions d'utilisation !

Le code génétique :

'UUU', 'UUC' : 'F'
'UUA', 'UUG', 'CUU', 'CUC', 'CUA', 'CUG' : 'L'
'AUU', 'AUC', 'AUA' : 'I'
'AUG' : 'M'
'GUU', 'GUC', 'GUA', 'GUG' : 'V'
'UCU', 'UCC', 'UCA', 'UCG', 'AGU', 'AGC' : 'S'
'CCU', 'CCC', 'CCA', 'CCG' : 'P'
'ACU', 'ACC', 'ACA', 'ACG' : 'T'
'GCU', 'GCC', 'GCA', 'GCG' : 'A'
'UAU', 'UAC' : 'Y'
'UAA', 'UAG', 'UGA' : '*'
'CAU', 'CAC' : 'H'
'CAA', 'CAG' : 'Q'
'AAU', 'AAC' : 'N'
'AAA', 'AAG' : 'K'
'GAU', 'GAC' : 'D'
'GAA', 'GAG' : 'E'
'UGU', 'UGC' : 'C'
'UGG' : 'W'
'CGU', 'CGC', 'CGA', 'CGG', 'AGA', 'AGG' : 'R'
'GGU', 'GGC', 'GGA', 'GGG' : 'G'

Question 6

Réalisez une fonction nommée `**traduit**` qui construit la séquence protéique obtenue par la traduction de la séquence ARN passée en paramètre. Cette traduction se fait à partir du premier nucléotide de la séquence ARN et utilise la fonction `**codeGenetique**`.

Exemple :

```
>>> traduit('AAUUCAAUUUAA')
'NSI*'
>>> traduit('AUGCGAAGCCGAAAGAACACCGGCUAA')
'MRSRKNTG*'
```

La réplication

Question 7

Réalisez une fonction nommée `**replique**` qui construit la séquence ADN complémentaire et inversée de celle passée en paramètre.
Pour cela, cette fonction fait appel à la fonction `**baseComplementaire**`. Une contrainte d'utilisation de cette fonction est que son paramètre est bien une séquence ADN.

Exemple :

```
>>> replique('ACTG')
'CAGT'
```

Bilan

Afin de s'assurer que les fonctions programmées ont le comportement attendu, vous avez normalement écrit des doctests pour chaque fonction, à part pour **genereADN** qui renvoie une chaîne de caractères aléatoire.
Il est également possible de finir cet exercice par l'écriture d'un menu interactif qui permet d'utiliser les fonctions programmées. Ce menu sera programmé dans le main et comportera des vérifications de saisie.