

## TD 2

### Travail sur les chaînes de caractères

Pour comprendre les éléments liés au traitement des chaînes de caractères, un exemple de séquence biologique mélangeant ARN et ADN et majuscules / minuscules a été discuté au tableau, pour estimer les approches « naïves » que l'on peut mettre en œuvre.

Par exemple, comment traduire en protéine une séquence d'ADN (trois lettres de l'ADN donnent 1 lettre en protéine, cf [https://fr.wikipedia.org/wiki/Code\\_génétique](https://fr.wikipedia.org/wiki/Code_génétique)).

Les instructions suivantes sont utiles pour travailler sur les chaînes de caractères.

```
var sequence = "ATGcggAAATCTTGUACCATUGA";
```

---

#### Éviter de chercher deux fois une lettre : `toLowerCase()`, `toUpperCase()`

Ici le principe est d'indiquer que lorsque l'on doit tester si une lettre est présente ou pas, il vaut mieux la transformer en majuscule ou minuscule pour n'effectuer qu'un des deux tests.

Il faut ainsi utiliser `texte.toLowerCase()` ou `texte.toUpperCase()`

---

#### Rechercher la position d'un caractère « `indexOf()` »

**Algorithme :** parcourir la chaîne du début à la fin et tester chaque position pour savoir si la lettre est trouvée. Si la position est trouvée, alors on s'arrête.

**Exemple javascript :** Déterminer la position du premier "U" (une thymine)

```
let positionU = sequence.indexOf("U");
```

**Remarque :** la séquence est numérotée à partir de zéro en informatique, mais qu'il faut ajouter 1 pour la position qui est affichée à l'utilisateur.

---

#### Rechercher la position d'un caractère « `search()` »

Cette méthode donne le même résultat (une position) que la recherche « `indexOf()` » mais il est possible d'utiliser une expression régulière ([https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Regular_Expressions)).

**Exemple :** la séquence a un « A » au début, suivi d'une lettre quelconque, suivi d'une autre lettre « A ».

```
let recherche = sequence.search('AAA');  
let recherche2 = sequence.search('A.A');
```

**Remarque :** on utilise le mot-clé « `let` » car on suppose que cette variable va être utilisée dans une fonction.

---

## Savoir si une chaîne de caractères contient un motif « match() »

Il est important en biologie de savoir identifier une séquence d'intérêt, par exemple un variant dans une séquence (comme pour la protéine Spike du SARS-Cov-2)

**Algorithme** : vérifier si un motif est présent, et ensuite vérifier si la position que l'on a trouvée est bien celle que l'on veut trouver

**Exemple** : mutation L452R du variant omicron, mais pas delta

Codon pour la leucine (L) : UUA, UUG, CUA, CUG, CUT, CUC

Codon pour l'arginine (R) : CGA, CGG, CGT, CGC

**Remarque** : dans la deuxième situation il n'y a qu'une lettre qui change (on parle de Single Nucléotide Polymorphism, que l'on prononce SNIP en Anglais)

[https://en.wikipedia.org/wiki/DNA\\_and\\_RNA\\_codon\\_tables](https://en.wikipedia.org/wiki/DNA_and_RNA_codon_tables)

Il faut donc rechercher l'un des 4 motifs avec une expression régulière (CG.) et vérifier qu'il est présent à la position 452.

A noter que la lettre « U » implique que l'on a un ARNm (celui du virus) et non l'ADN qui encode cet ARNm (qui lui a un « T »).

Ce n'est évidemment pas le plus efficace ...

---

## Vérifier le caractère à une position précise : charAt()

**Algorithme** : vérifier le caractère à une position spécifique, pour savoir si on a la lettre R ou la lettre L (dans ce cas dans un contexte protéique, cf après)

On peut par exemple vérifier l'acide aminé à la position 452 (variant omicron) ou à la position 614 (variant delta)

---

## Récupérer un élément de la séquence substr()

**Intérêt** : retrouver une sous-chaîne dans la chaîne de référence (cf fichier joint).  
Par exemple pour transformer l'ARNm (ou ADN) en protéine (3 lettres → 1 lettre)

**Remarque** : si on prévoit d'utiliser une forme de stockage des données de type « clé/valeur », on parle de dictionnaire en python ou de collection avec clés en javascript ([https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Keyed\\_collections](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Keyed_collections)), alors il est possible d'avoir une version très simple de la conversion d'un codon en acide aminés.