

Alignement de séquences

1. Présentation du sujet

On dispose de deux chaînes de caractères : A , qui vaut `INFORMATIQUE`, et B , qui vaut `NUMERIQUE`. On aimerait mettre ces deux chaînes de caractères en correspondance de la manière suivante :

- On place les 2 chaînes l'une en dessous de l'autre;
- Si les derniers caractères des deux chaînes coïncident, alors on passe aux caractères suivants;
- Sinon, on va ajouter un trou dans une des deux chaînes, symbolisé par un - et on passe aux caractères suivant.

Voici un exemple d'alignement optimal :



Dans cette situation on a besoin de 9 tirets, pas moins.

L'objectif est d'aligner le maximum de lettres (donc de mettre le moins de - possible). Ce n'est pas un problème simple, surtout quand les chaînes sont longues, comme pour les séquences d'ADN par exemple :

```
TTCACCAAGAAAAGA - -ACACGGTAGTTA-CGAG - -TCCAATATT-GTTAA - -ACC - -G
TTCA - C -GAAAA -AGTA-ACGG - -G - -CCGA-TCTCCAATA - -AG - T - -GCGACCGAG
```

2. Résolution par une méthode récursive

Le principe est présenté dans la vidéo suivante :

Pour les mots GRAS et GERS, l'arbre obtenu est le suivant :



Après application d'une méthode dynamique (**Top Down**) (c'est-à-dire récursive avec mémorisation), on obtient le graphe suivant :



3. Résolution par une méthode itérative

3.1. Activité débranchée

Considérons les deux chaînes de caractères `GENOME` et `ENORME`. Afin de chercher le nombre minimal d'insertion à effectuer, nous allons compléter le tableau suivant, avec la convention suivante : à l'intersection de la colonne `N` et de la première ligne `E` se trouve le nombre minimal d'insertion nécessaire pour aligner les chaînes `GEN` et `E`, c'est-à-dire 2 tirets. Par convention la première ligne et la première colonne correspondent à une chaîne vide.

?

Question

Enoncé

En quoi la méthode ci-dessus est-elle une méthode itérative (**Bottom Up**) ?

Réponse

A venir !

3.2. Application en Python

L'activité est disponible sous la forme d'un `notebook capytale`