# UE Bioinformatique BISM
# Projet Needleman-Wunsch - L3 BISM
# 2019-2020

**À rendre le 31/03/2020, heure limite 23:59.**
**-1 point pour chaque jour de retard.**
**Deposer vos fichiers sur CLaroline dans le zone de depot "rendu_NW" dans une archive zip nommée nom_prenom_NW.zip. Chacun des noms de vos fichier devra contenir votre prénom et votre nom.**

**Objectif: implémentez l'algorithme de Needleman-Wunsch en python.**

**Règles à suivres :**

1. Les deux séquences à aligner se trouveront dans deux fichiers au format fasta (un par séquence). Il faudra donc lire et stocker les séquences de ces fichiers.

2. Votre algorithme doit être capable d'aligner des séquences nucléotidiques composées de `A, a, C, c, G, g, T, t, U, u`.

3. Par défaut, votre matrice de substitution doit : donner un score de 2 à un match, un score de 1 à un mismatch purine/purine (A et G) ou pyrimidine/pyrimidine (C et T) et retirer une penalité de -1 aux autres mismatches. *Vous pouvez en plus donner la possibilité à l'utilisateur d'entrer sa propre matrice de substitution, au format de votre choix.*

4. Par défaut, une ouverture de gap doit retirer une penalité de -10. Ce paramètre doit pouvoir être modifié.

5. Par défaut, une extension de gap doit retirer une penalité de -1. Ce paramètre doit pouvoir être modifié.

6. L'algorithme doit trouver au moins un des meilleurs alignements.

7. Cet alignement doit être affiché avec les informations suivantes au moins : les séquences, les symboles d'alignement, le score final de l'alignement et le nombre de match/mismatch/gap. Les séquences et les symboles d'alignements peuvent par exemple ressembler à ça :

```
ATGGCGT
|||: ||
ATGA-GT
```

*Ceci n'est qu'un exemple, vous pouvez proposer votre propre affichage tant que celui-ci est claire et que toutes les informations requises y figure.*

8. Votre algorithme doit retourner les matrices de score et de traceback au moins.
9. Votre code doit être fait pour faciliter sa compréhension par d'autres programmeurs. C'est à dire, clair (nommage rationnel des variables) et commenté : le rôle et les entrées/sorties de chaque fonction, classe, méthode, paramètre ou partie de code compliqué doit être précisé.
10. Votre code doit être facile à exécuter pour quelqu'un d'extérieur (par exemple, un utilisateur qui n'a jamais programmé en python, mais qui veux aligner deux séquences).
11. Vous devez créer un fichier `README` qui explique ce que fait votre algorithme et quels sont les paramètres modifiables par l'utilisateur ainsi que leurs valeurs par défaut. Ce fichier doit aussi contenir un exemple fonctionnel d'utilisation de votre script python, ainsi qu'une explication de ce qui est affiché et retourné.

**Fonctionnalité à ajouter:**

1. L'utilisateur doit pouvoir réaliser un alignement protéique. Vous devez déterminer quelle sera alors la matrice de substitution à utiliser.
2. L'alignement protéique doit être fonctionnel.

**Fonctionnalitées bonus:**

1. L'utilisateur doit pouvoir spécifier si il préfère réaliser un alignement global (Needleman-Wunsch) ou local (Smith-Waterman). Quelles sont les modifications à apporter à votre code pour que celui-ci réalise un alignement local ?
2. Implémenater ces modifications.
3. Trouver l'ensemble des alignements optimaux (càd retourner les ex aequo si il y en a).