Complexité des Algorithmes

Création et Remplacement

Pour les méthodes de **création et de remplacement**, nous parcourons une boucle de taille `N` afin de générer aléatoirement des caractères dans la liste de données (liste de nucléotides). La complexité de ces opérations est donc `O(N)`.

Il en va de même pour les **opérations de suppression et d'insertion**.

Comparaison

Pour la **comparaison par remplacement**, nous parcourons deux listes de même taille et faisons un parcours par indice. La complexité reste `O(N)`, où `N` correspond à la longueur de la chaîne de caractères.

Distance de Levenshtein

Pour le calcul de la **distance de Levenshtein**, nous utilisons une matrice inspirée de la page Wikipédia sur l'algorithme. Cette matrice est remplie grâce à deux boucles imbriquées parcourant respectivement `n` et `m`, ce qui donne une complexité de `O(n * m)`, `n` et `m` représentant les longueurs des deux chaînes ou fichiers comparés.

Phylogénie

L'accès aux données se fait directement grâce à la programmation orientée objet, ce qui rend les opérations courantes de complexité `O(1)`, à l'exception de la **création**, qui nécessite la lecture complète du fichier et a donc une complexité `O(n)`.

Fonctions Distance et Espèces

Les fonctions **distance**, **espèces** sont récursives et ont une complexité `O(n * m`, où `m` est le nombre d'espèces considérées.