

## TP ALGO 9 : CORRIGÉ

## LECTURE D'UN FICHER TEXTE

```
import numpy as np
import matplotlib.pyplot as plt

# Brins d'ADN

fichierADN = open('hamming.txt', 'r')
brin1 = fichierADN.readline()
brin2 = fichierADN.readline()
fichierADN.close()

def repartition(brin) :
    """
    Renvoie la repartition en A,C,G,T de la chaine de caracteres brin
    """
    cA,cC,cG,cT = 0,0,0,0
    for lettre in brin :
        if lettre == 'A' :
            cA = cA+1
        elif lettre == 'C' :
            cC = cC+1
        elif lettre == 'G' :
            cG = cG+1
        elif lettre == 'T' :
            cT = cT+1
    return cA,cC,cG,cT

def GCcontent(brin):
    """Renvoie la teneur en GC de la chaine de caracteres brin"""
    cA,cC,cG,cT = repartition(brin)
    return 100*(cC+cG)/(cA+cC+cG+cT)

def Hamming(s,t) :
    """
    s et t sont deux chaines de caractères
    Renvoie le nombre de différences (dite distance de Hamming) entre les deux chaines
    """
    n=len(s)
    diff=0
    for i in range(n) :
        if s[i] != t[i] :
            diff = diff+1
    return diff
```

```
>>> repartition(brin1)
(223, 225, 249, 228)
>>> repartition(brin2)
(236, 213, 264, 212)
>>> GCcontent(brin1)
51.24324324324324
>>> GCcontent(brin2)
51.567567567567565
>>> Hamming(brin1,brin2)
465
```

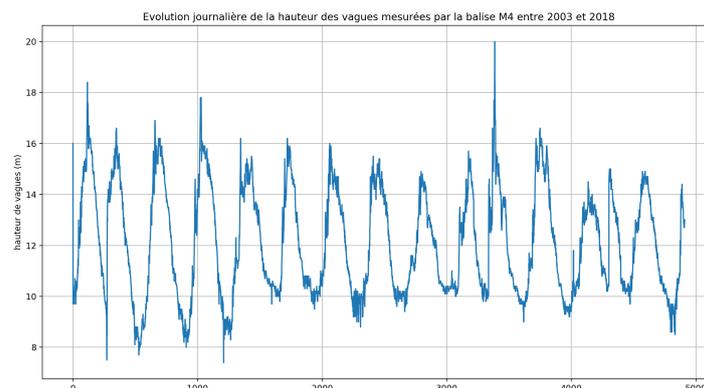
## ÉCRITURE D'UN FICHIER TEXTE

```
def compteRendu(s,t) :
    '''
    s et t sont deux chaines de caractères
    Genere un compte-rendu (repartition, teneur en GC, distance de Hamming) a partir de
    ces deux brins d ADN
    '''
    teneur1 = GCcontent(s)
    teneur2 = GCcontent(t)
    d = Hamming(s,t)
    monResume = open('CR.txt','w')
    entete1 = '*Premier brin*'+'\n'
    message1 = 'Teneur en GC : '+str(teneur1)+'\n'
    entete2 = '*Deuxième brin*'+'\n'
    message2 = 'Teneur en GC : '+str(teneur2)+'\n'
    entete3 = '*Nombre de mutations*'+'\n'
    message3 = 'Distance de Hamming entre les deux brins : '+str(d)
    monResume.write(entete1+message1+entete2+message2+entete3+message3)
    monResume.close()
```

## EXTRAIRE DES DONNÉES D'UN FICHIER TEXTE POUR LES REPRÉSENTER

```
def extraction1() :
    '''
    Renvoie la liste des hauteurs de vagues écrites dans le fichier hauteurs.txt
    '''
    f = open("hauteurs.txt", 'r')
    contenu = f.read()
    f.close()
    contenu = contenu.split(' ')
    L = [eval(val) for val in contenu]
    return L

def vagues() :
    ''' Affiche l'évolution journalière de la hauteur des vagues de 2003 à 2018 '''
    Y = extraction1()
    plt.plot(Y)
    plt.ylabel('hauteur de vagues (m)')
    plt.title('Evolution journalière de la hauteur des vagues mesurées par la balise M4
entre 2003 et 2018')
    plt.grid()
    plt.show()
```



```

def extraction2() :
'''
Renvoi :
la liste des précipitations écrites dans le fichier pluiesoleil.txt
et la liste des rayonnements solaires écrits dans le fichier pluiesoleil.txt
'''
f = open("pluiesoleil.txt", 'r')
ligne1 = f.readline()
ligne2 = f.readline()
ligne1 = ligne1.split(' ')
ligne2 = ligne2.split(' ')
L1 = [eval(val) for val in ligne1]
L2 = [eval(val) for val in ligne2]
f.close()
return L1,L2

def pluieSoleil() :
'''Affiche le rayonnement solaire en fonction des précipitations pluvieuses'''
X,Y = extraction2()
plt.plot(X,Y, 'o')
plt.grid()
plt.xlabel('Précipitations (mm)')
plt.ylabel('Rayonnement solaire (Wm-2)')
plt.title("Rayonnement solaire en fonction des précipitations au mois de janvier sur l'
archipel de Molène (période 1961-1990)")
plt.show()

```

