COMP561-A1_NeedlemanWunch Modified Implementation (python)

October 27, 2019

[2]:
```
import numpy as np
import string
import pandas as pd# implement multi-gap free# 4 inputs: fasta file; score for␣
 ↪match; score forr mismatch; gap penalty bdef
def main():
    fasta = open('/root/comp401/hw1_medium.txt','r')
    mgf(fasta,1,-1,-1)
def SuborMatch(ms,mms,x,y):
    if x is y:
        return ms #match score
    else:
        return mms#mismatch score
def mgf(fasta,ms,mms,b):
    l1 = fasta.readline()
    S = list((l1.strip()).split(' ')[1])
    l2 = fasta.readline()
    T = list(l2.split(' ')[1])
    m = len(S)
    n = len(T)
    X= np.full((m+1,n+1),float("-inf"))
    I = np.full((m+1,n+1),float("-inf"))
    D = np.full((m+1,n+1),float("-inf"))
    align = np.full((m+1,n+1),'n')
    X[0,0]=0
    X[0,1]=b
    X[1,0]=b
    I[0,1]=b
    I[1,0]=b
    D[0,1]=b
    D[1,0]=b
    I[0,0]=0
    D[0,0]=0
    for i in range(1,m+1,1):
        for j in range(1,n+1,1):
            I[i,j] = max(X[i-1,j]+b,D[i-1,j]+b)#b
```

```
            D[i,j] = max(X[i,j-1]+b,I[i,j-1]+b)#b
            s=SuborMatch(1,-1,S[i-1],T[j-1])
            X[i,j] = max(X[i-1,j-1]+s,I[i-1,j-1]+s, D[i-1,j-1]+s)
            if (X[i,j]==X[i-1,j-1]+s):
                align[i,j] = 'x'
            elif (X[i,j]==I[i-1,j-1]+s):
                align[i,j] = 'd'
            else:
                align[i,j] = 'i'
#     print(X)
#     print(I)
#     print(D)
#     print(S)
#     print(align)
    s_inAlign = []
    t_inAlign = []
    i =m
    j =n
    score = max(X[i,j],I[i,j],D[i,j])
    while (i >=0 and j>=0):
        if align[i,j]=='x':
    #         print ("yes")
            s_inAlign.append(S[i-1])
            t_inAlign.append(T[j-1])
            i=i-1
            j=j-1
        elif align[i,j]=='i':
            t_inAlign.append(T[j-1])
            s_inAlign.append('-')
            j=j-1
        elif align[i,j]=='d':
            s_inAlign.append(S[i-1])
            t_inAlign.append('-')
            i=i-1
        else:
            break
    s_inAlign.reverse()
    t_inAlign.reverse()
    align_s = string.join(s_inAlign)
    align_t = string.join(t_inAlign)
#     with open ('/root/comp401/hw1_feedback_q3.txt','a') as f:
#         f.write("score for long run is"+str(score)+'\n')
#         f.write(align_s+'\n')
#         f.write(align_t+'\n')
    print (score)
    print (align_s)
    print (align_t)
```

```
main()
```

```
314.0
A T G G A T T T A T C T G C T C T T C G C G - T - T G A A G A A G T A C A A A A
T G T C A T T A A T G C T A T G C A G A A A A T C T T A G A G T G T C C C A T C
T G T C T G G A G T T G A T C A A G G A A C C T G T C T C C A C A A A G T G T G
A C C A C A T A T T T T G C A A A T T T T G C A T G C T G A A A C T T C T C A A
C C A G A A G A A A A G G G C C T T C A C A G T G T C C T T T A T G T A A G A A T
G A T A T A A C C A A A A A G G A G C C T A C A A G A A A G T A C G A G A T T T A
G T C A A C T T G T T G A A G A G C T A T T G A A A A T C A T T T G T - G C T T
T T C A G C T T G A C A C A G G T T T G G A G T - A T G C A A A C A G C T A T A
A T T T T G C A A A A A A A G G - A A A A T A A C T C T C C T G A A C A T C T A A
A A G A T G A A G T T T C T A T C A T C C A A A A G T A T G G G C T A C A G A A A
C C G T G C C A A A A A G A C T T C T A C A G A G T G - A A C C C G A A A A T C -
C T T C C T T G C A G G A A A C C A G T C T C A G - T G T C C A A C T C T C T A
A C C T T G G A A C - T G T G A G A A C T C T G A - G G - A - C
A T G G A T T T A T C T G C C - G T C C - A A A T T C A A G A A G T A C A A A A A
T G T C C T T C A T G C T A T G C A G A A A A T C T T A G A G T G T C C G A T C
T G T T T G G A A C T G A T C A A A G A A C C T G T T T C C A C A A A G T G T G
A C C A C A T A T T T T G C A A A T T T T G T A T G C T G A A A C T T C T T A A
C C A G A A G A A A A G G G C C T T C A C A A T G T C C T T T G T G T A A G A A T
G A G A T A A C C A A A A A G G A G C C T A C A G G G A A G C A C A A G G T T T A
G T C A G C T T G C T G A A G A G C T G C T G A G A A T A - A T G G C T G C T T
T T G A G C T T G A C A C G G G A A T G C A G C T T - A C A A A T G G T T T T A
G T T T T T C A A A A A A A G A G A - A A T A A T T C T T G T G A G C G T T T G A
A T G A G G A G G C G T C G A T C A T C C A G A G C G T G G G C T A C C G G A A
C C G T G T C A G A A G G C T T C C C C A G G - T C G A A C C T G G A A A T G C
C A - C C T T G A A G G A - C - A - G C C T A - G G T G T C C A G C T G T C T A
A C C T T G G A A T C G - T G A G A T C A G T G A A G A A A A A
```