DNA Alignment – Worksheet 2

*CDS 230*

*Spring 2018*

# Problem

Load and run the code presented in slide 4 of the video (Code 25.7 in the text).

def BruteForceSlide( mat, abet, seq1, seq2 ):

# length of strings

l1, l2 = len( seq1 ), len( seq2 )

# make new string with leader

t1 = len( seq2) \* '.' + seq1

lt = len( t1 )

answ = np.zeros( lt, int )

for i in range( lt ):

answ[i] = BlosumScore( mat, abet, t1[i:], seq2 )

return answ

import simplealign as sal

import blosum

s1 = 'RNDKPKFSTARN'

s2 = 'AAAAARNQKPKWWTATN'

v = sal.BruteForceSlide(blosum.BLOSUM50, blosum.PBET, s1, s2)

print(len(s2) - v.argmax())

s1 = 'AAAAAAARNDKPKFSTARN'

s2 = 'RNQKPKWWTATN'

v = sal.BruteForceSlide(blosum.BLOSUM50, blosum.PBET, s1, s2)

print(len(s2) - v.argmax())

print(s1)

print(7\* '.' +s2)

5-7

AAAAAAARNDKPKFSTARN

.......RNQKPKWWTATN

# Problem

Load and run the code presented in slide 14 of the video (Code 25.12 in the text).

import dynprog as dpg

s1 = 'IQIFSFIFRQEWNDA'

s2 = 'QIFFFFRMSVEWND'

scormat, arrow = dpg. ScoringMatrix (blosum. BLOSUM50, blosum. PBET, s1, s2)

def Backtrace (arrow, seq1, seq2):

st1, st2 = '',''

v, h = arrow. shape

ok = 1

v-=1

h-=1

while ok:

if arrow [v, h] == 0:

st1 += seq1 [v-1]

st2 += '-'

v -= 1

elif arrow [v, h] == 1:

st1 += '-'

st2 += seq2 [h-1]

h -= 1

elif arrow [v, h] == 2:

st1 += seq1 [v-1]

st2 += seq2 [h-1]

v -= 1

h -= 1

if v ==0 and h ==0:

ok = 0

#reverse the strings

st1 = st1 [::-1]

st2 = st2 [::-1]

return st1, st2

st1, st2 = dpg. Backtrace (arrow, s1, s2)

print(st1)

print(st2)

*IQIFSFIFRQ--EWNDA*

*-QIF-FFFRMSVEWND-*

# Problem

What is the purpose of the arrow matrix?

The arrow matrix is used to construct the aligned sequences. Once the scoring matrix is filled out, the optimal alignment is extracted using the arrow matrix. The arrow matrix is used to find which cell was influential in determining the value of the subsequent cell.