BIOL 4540 Sequence Analysis
Homework 2 -- Due monday Oct 2

Starting with your DP alignment program from HW1, extend program in four ways:
(1) use BLOSUM,
(2) use affine gap penalty,
(3) do local alignment, and
(4) output alignment in blocks.

Replace the pseudocode with real code. WARNING: There may be unintentional bugs in the pseudocode! I have looked it over but nobody is perfect. Your best defense against bugs is to understand what you are doing and not to simply slavishly translate the pseudocode into code.

Test your code on the sequences provided under HW2 of www.bioinfo.rpi.edu/bystrc/courses/biol4540/homework.html

```plaintext
## Read input file names and parameters from stdin or command line.
## (Don't hard wire, except during debugging)
#------------------------------------------------------------#
## Read sequence 1 and sequence 2, FASTA format.
aa1 = read(seq1file)
aa2 = read(seq2file)
#------------------------------------------------------------#
## Read in the BLOSUM62 matrix
## (or any other 20x20 substitution matrix), ordered by the
## amino acids 1-letter code.
BLOSUM(1:20,1:20) = read(blosumfile)
#------------------------------------------------------------#
## Write the function matchscore() to return the BLOSUM score
## integer function matchscore(i,j)
return BLOSUM(posit(aa1[i]), posit(aa2[j]))
## ...where aa1 is sequence 1, aa2 is sequence 2.
#------------------------------------------------------------#
## posit() is a function returning the
## position of the amino acid character in constant
## aastring="ACDEFGHIKLMNPQRSTVWY"
integer function posit(aa)
return index(aastring, aa)   // this works in fortran.
// Returns 0 if not a AA (error!)
#------------------------------------------------------------#
## getmaxarrow() is the DP summing forward function
## using affine gap penalty.
real function getmaxarrow(matrix, i,j, gop, gep, fromi, fromj)
For all positions in the row (i-1, j=1:j-1) and in the column
```
(i=1:i-1, j-1) sum the matrix value and the appropriate affine gap penalty. Keep the highest score and return it as the return value. Also return fromi and fromj as the positions to trace back to. If zero is the highest score then return 0 and trace back to (0,0). Watch out for off-by-one errors. Make sure this function works properly for all values of i and j, including i=1 and j=1. Refer to Lecture 4, Slide 22 and Slide 12.


For i=1,sizeof(aa1)
For j=1,sizeof(aa2)
    matrix(i,j) = matchscore(i,j) +
    getmaxarrow(matrix,i,j,gop,gep,fromi,fromj)
    traceback(i,j) = (fromi, fromj)

## MAIN function. Part 2. Traceback. From the end, traceback to maximum, then from there trace back to beginning (0,0).

nextmatch = (fromi, fromj) = getmaximum(matrix)
i = fromi
j = fromj
while (i > 0)
    nextmatch += (fromi, fromj) = traceback(i,j)  // += means append
    // to nextmatch, which is a list.
    (i,j) = (fromi, fromj)

## calculate the number of alignment columns
ncolumns = sizeof(aa1) + sizeof(aa2) - sizeof(nextmatch)
## ...where sizeof(nextmatch) is the number of match columns.
## Calculate the number of blocks in the alignment
## if each block is 50 columns.
 nblocks = ceiling(ncolumns/50)
## Create output strings.
k = sizeof(nextmatch)
(fromi, fromj) = (0,0)
outputstr1 = "" /* empty string*/
outputstr2 = "" /* empty string*/
while (k > 0)

(toi,toj) = nextmatch(k)
outputstr1 += aa1[fromi+1:toi-1] /* if not empty set, gapped */
// positions in aa1
outputstr1 += gaps( toj - fromj - 1) /* if > 0, write this many */
"-" for gapped position in aa2
outputstr1 += aa1[toi] /* match position*/
outputstr2 += aa2[fromj+1:toj-1] /* if not empty set, gapped */
//positions in aa2
outputstr2 += gaps( toi - fromi - 1) /* if > 0, write this many "-" */
for gapped position in aa1
outputstr2 += aa2[toj] /* match position*/

k-- /* go to next match*/
## Tack on the unaligned sequence at the end if both sequences.
outputstr1 += aa1[toi+1:nres21] += gaps(nres2-toj-1)
outputstr2 += gaps(nres1-toi-1) += aa2[toj+1:nres2]
## Output in blocks of 50 characters with a space line
For ibl = 1,nblocks

print outputstr1[ (ibl-1)*50 + 1: ibl*50) /* print a substring*/
print outputstr2[ (ibl-1)*50 + 1: ibl*50)
print empty line

==============================================================================
Files you need are in
www.bioinfo.rpi.edu/bystrc/courses/biol4540/homework.html
under HW2

Use smallseq1, smallseq2 to debug. You should get this alignment no matter
what gap penalty you pick.

seq1 EFGHIKLM
seq2 EFGHIKLM

If you didn't get this alignment, debug the program by writing out
(for example) matrix, traceback, etc. Here's my debugging output for
gop=10 and gep=5. I wrote out the BLOSUM matrix, the alignment score
and the alignment matrix after the forward sum. These are the correct
numbers for smallseq1, smallseq2 given gop=10, gep=5. i is row, j is column.

20 lines read from blosum.dat

Here's what you should get for the two sequences used in HW1. gop=5, gep=2

Your results may vary slightly.

Assignment:

On the homework web page, click "program" and upload your program.
Click "results" and upload a plain-text file (no PDF or Word files) with the following in it:
1) Alignment output for the sequences linked under "1fab" and "2fab" using gop=10 and gep=1.
2) Alignment output for the sequences linked under "hydrolase1" and "hydrolase2" using gop=5
and gep=0.