

Assignment 02

Fundamentals of Bioinformatics

144133E – M.G.K.C.Piyarathna
B14
Faculty of IT
University of Moratuwa.

I have used following MATLAB Commands to both *Needleman-Wunch algorithm* and *Smith-Waterman algorithm*.

- Nargin – It is the number of Input Arguments
- Load(file name) – Load data from file name
- Length(x)- return the length of x
- Zeros(x,y)- create an array of zeros
- Strcat – string concatenation
- Disp – print output
- Max(A)- return the largest elements of A
- Cell - cell array is a data type with indexed data containers called cells. Each cell can contain any type of data
- bar3 - draw a 3 dimensional bar graph
- title –set title
- set(A,'B',C) - specifies a value for the property Name on the object identified by A
- xlabel- adds the label to the axes specified by Score
- pcolor - Pseudocolor plot-a rectangular array of cells with colors determined by C
- colormap - define the color scheme for many types of visualizations, such as surfaces and patches
- colorbar- showing color scale

Needleman-Wunch algorithm

NW algorithm consists of three steps.

- *Initialization of the score matrix*
A matrix $D(i,j)$ indexed by residues of each sequence is built recursively, such that
$$D(i,j) = \max \{ D(i-1,j-1) + s(x_i, y_j), D(i-1,j) + g, D(i,j-1) + g \}$$

Here $s(i,j)$ is the substitution score for residues i and j , and g is the gap penalty.
- *Calculation of scores and filling the trace-back matrix*
The traceback always begins with the last cell to be filled with the score (Bottom right cell). The traceback is completed when the first, top-left of the matrix is reached.
- *Deducing the alignment from the trace-back matrix*

Following is the code step by step. Each new step, there is a comment which explaining what going to do.

```
%144133E - M.G.K.C.Piyarathna
%Assignment 2 - BioInfomatics
%Implementing matlab code for Needleman_Wunsch Algorithm
function [] = needleman_wunsch(sequence_1, sequence_2)
%function is a group of statements that together perform a task. The name of
the file and of the function should be the same.
%I have input two parameters. Those are Sequences we have to provide
%Take in inputs (sequence_1, sequence_2)
%assign sequence_1 to variabe A
    A = sequence_1;
%assign sequence_2 to variabe A
    B = sequence_2;

%Computing the matrix (F)
%setting up variables
%Here I have used gap penalty = -2, match = 1, mismatch = -3
d = -2;%gap penalty
m = 2;%match
s = -3;%mismatch

column = length(A);%return the length of A and assign it to the column
row = length(B);%return the length of B and assign it to the row

%preallocating matrix F with the gap penalty as the top row and column
F = zeros(row+1,column+1); %create an array of Zeros/preallocating matrix
F all zeros
F(2:end,1) = d * (1:row)';
F(1,2:end) = d * (1:column);
%scores = zeros(row, col);
```

```

%Filling in the matrix
for i=2:row+1
    for j=2:column+1
        %j is going through the columns which is A
        %if the two positions match, the index in scores is listed as such
        if (A(j-1) == B(i-1))
            scores(A(j-1),B(i-1)) = m;
        else
            scores(A(j-1),B(i-1)) = s;
        end
        %Filling-in partial alignments here
        Match      = F(i-1,j-1) + scores(A(j-1),B(i-1));
        MismatchA   = F(i, j-1) + d;
        MismatchB   = F(i-1, j) + d;
        %computing the final score of the alignment and assigning it to F
        Temp = [Match MismatchA MismatchB];
        F(i,j) = max(Temp); % return the maximum of Temp
        % end
    end
end

%traceback part
Alignment_A = '';
Alignment_B = '';
i = length(B)+1; %row ----return the length of B and adding 1
j = length(A)+1; %col  ----return the length of A and adding 1

while (i>1 && j>1)
    Score = F(i,j);
    DIAG = F(i-1,j-1);
    %LEFT = F(i-1,j);
    UP    = F(i,j-1);

    %if scores are equal to the diagonal, there is no gap.
    if (Score == DIAG + scores(A(j-1),B(i-1)))
        Alignment_A = strcat(Alignment_A, A(j-1));%String concatenation
        Alignment_B = strcat(Alignment_B, B(i-1));%String concatenation
        %computes score, checks to see if the alignment are the same
        %characters

        i = i-1;
        j = j-1;

        %gap in sequence B
        elseif (Score == UP + d)
            Alignment_A = strcat(Alignment_A, A(j-1));%String concatenation
            Alignment_B = strcat(Alignment_B, '-');%String concatenation
            j = j-1;
            %gap in sequence A
        else
            Alignment_A = strcat(Alignment_A, '-');%String concatenation
            Alignment_B = strcat(Alignment_B, B(i-1));%String concatenation
            i = i-1;
        end
    end
end
%If at the end of one sequence fills in the rest with gaps
while(j>1)
    Alignment_A = strcat(Alignment_A, A(j-1));%String concatenation
    Alignment_B = strcat(Alignment_B, '-');%String concatenation
    j = j-1;
end

```

```

end
while(i>1)
    Alignment_A = strcat(Alignment_A, '-');%String concatenation
    Alignment_B = strcat(Alignment_B, B(i-1));%String concatenation
    i = i-1;
end

%displays the alignment score and alignments
disp('Score Matrix');%print output
i = 1;
while(i<=row+1)
    disp(F(i,1:end));%print output
    i=i+1;
end
matches = 0;
for i=1:length(Alignment_A)
    if( Alignment_A(i) == Alignment_B(i))
        matches = matches + 1;
    end
end
disp (strcat('Number of Matches :',num2str(matches)));%String concatenation
%num2str- converts the array A into a string
disp('Final alignment for given two sequences');%print output

disp(Alignment_A);%print output
disp(Alignment_B);%print output

```

```

1 %14133E - M.G.K.C.Piyarathna
2 %Assignment 2 - Bioinformatics
3 %Implementing matlab code for Needleman_Wunsch Algorithm
4 function [] = needleman_wunsch(sequence_1, sequence_2)
5 %function is a group of statements that together perform a task.The name of the file and of the function should be the same.
6 %I have input two parameters.Those are Sequences we have to provide
7 %Take in inputs (sequence_1, sequence_2)
8 %assign sequence_1 to variable A
9 A = sequence_1;
10 %assign sequence_2 to variable B
11 B = sequence_2;
12
13 %Computing the matrix (F)
14 %setting up variables
15 %Here I have used gap penalty = -2, match = 1, mismatch = -3
16 d = -2;%gap penalty
17 m = 2;%match
18 s = -3;%mismatch
19
20 column = length(A);%return the length of A and assign it to the column
21 row = length(B);%return the length of B and assign it to the row
22
23
24 %preallocating matrix F with the gap penalty as the top row and column
25 F = zeros(row+1,column+1); %create an array of Zeros/preallocating matrix F all zeros
26 F(2:end,1) = d * (1:row)';
27 F(1,2:end) = d * (1:column);
28 %scores = zeros(row, col);
29 %Filling in the matrix
30 for i=2:row+1

```

```

84 %If at the end of one sequence fills in the rest with gaps
85 while(j>1)
86     Alignment_A = strcat(Alignment_A, A(j-1));%String concatenation
87     Alignment_B = strcat(Alignment_B, '-');%String concatenation
88     j = j-1;
89 end
90 while(i>1)
91     Alignment_A = strcat(Alignment_A, '-');%String concatenation
92     Alignment_B = strcat(Alignment_B, B(i-1));%String concatenation
93     i = i-1;
94 end
95 %displays the alignment score and alignments
96 disp('Score Matrix');%print output
97 i = 1;
98 while(i<=row+1)
99     disp(F(i,1:end));%print output
100    i=i+1;
101 end
102 matches = 0;
103 for i=1:length(Alignment_A)
104     if( Alignment_A(i) == Alignment_B(i))
105         matches = matches + 1;
106     end
107 end
108 disp (strcat('Number of Matches :',num2str(matches)),'%String concatenation %num2str- converts the array A into a string
109 disp('Final alignment for given two sequences');%print output
110
111 disp(Alignment_A);%print output
112 disp(Alignment_B);%print output

```

Output

Functions which have built in matlab can be called from scripts, the command window, or from the functions. Here, I have demonstrated each code using command window.

```

>> sequence_1='CGTGAATTCAT';
>> sequence_2='GACTTAC';
>> needleman_wunsch(sequence_1, sequence_2)
Score Matrix
    0    -2    -4    -6    -8   -10   -12   -14   -16   -18   -20   -22
   -2    -3     0    -2    -4    -6    -8   -10   -12   -14   -16   -18
   -4    -5    -2    -3    -5    -7    -9   -11   -13   -15   -17   -19
   -6    -7    -4    -5    -6    -8   -10   -12   -14   -16   -18   -20
   -8    -9    -6    -7    -8   -10   -12   -14   -16   -18   -20   -22
  -10   -11    -8    -9   -10   -12   -14   -16   -18   -20   -22   -24
  -12   -13   -10   -11   -12   -14   -16   -18   -20   -22   -24   -26
  -14   -15   -12   -13   -14   -16   -18   -20   -22   -24   -26   -28
  -16   -17   -14   -15   -16   -18   -20   -22   -24   -26   -28   -30
  -18   -19   -16   -17   -18   -20   -22   -24   -26   -28   -30   -32
  -20   -21   -18   -19   -20   -22   -24   -26   -28   -30   -32   -34
  -22   -23   -20   -21   -22   -24   -26   -28   -30   -32   -34   -36
  -24   -25   -22   -23   -24   -26   -28   -30   -32   -34   -36   -38
  -26   -27   -24   -25   -26   -28   -30   -32   -34   -36   -38   -40
  -28   -29   -26   -27   -28   -30   -32   -34   -36   -38   -40   -42
  -30   -31   -28   -29   -30   -32   -34   -36   -38   -40   -42   -44
  -32   -33   -30   -31   -32   -34   -36   -38   -40   -42   -44   -46
  -34   -35   -32   -33   -34   -36   -38   -40   -42   -44   -46   -48
  -36   -37   -34   -35   -36   -38   -40   -42   -44   -46   -48   -50
  -38   -39   -36   -37   -38   -40   -42   -44   -46   -48   -50   -52
  -40   -41   -38   -39   -40   -42   -44   -46   -48   -50   -52   -54
  -42   -43   -40   -41   -42   -44   -46   -48   -50   -52   -54   -56
  -44   -45   -42   -43   -44   -46   -48   -50   -52   -54   -56   -58
  -46   -47   -44   -45   -46   -48   -50   -52   -54   -56   -58   -60
  -48   -49   -46   -47   -48   -50   -52   -54   -56   -58   -60   -62
  -50   -51   -48   -49   -50   -52   -54   -56   -58   -60   -62   -64
  -52   -53   -50   -51   -52   -54   -56   -58   -60   -62   -64   -66
  -54   -55   -52   -53   -54   -56   -58   -60   -62   -64   -66   -68
  -56   -57   -54   -55   -56   -58   -60   -62   -64   -66   -68   -70
  -58   -59   -56   -57   -58   -60   -62   -64   -66   -68   -70   -72
  -60   -61   -58   -59   -60   -62   -64   -66   -68   -70   -72   -74
  -62   -63   -60   -61   -62   -64   -66   -68   -70   -72   -74   -76
  -64   -65   -62   -63   -64   -66   -68   -70   -72   -74   -76   -78
  -66   -67   -64   -65   -66   -68   -70   -72   -74   -76   -78   -80
  -68   -69   -66   -67   -68   -70   -72   -74   -76   -78   -80   -82
  -70   -71   -68   -69   -70   -72   -74   -76   -78   -80   -82   -84
  -72   -73   -70   -71   -72   -74   -76   -78   -80   -82   -84   -86
  -74   -75   -72   -73   -74   -76   -78   -80   -82   -84   -86   -88
  -76   -77   -74   -75   -76   -78   -80   -82   -84   -86   -88   -90
  -78   -79   -76   -77   -78   -80   -82   -84   -86   -88   -90   -92
  -80   -81   -78   -79   -80   -82   -84   -86   -88   -90   -92   -94
  -82   -83   -80   -81   -82   -84   -86   -88   -90   -92   -94   -96
  -84   -85   -82   -83   -84   -86   -88   -90   -92   -94   -96   -98
  -86   -87   -84   -85   -86   -88   -90   -92   -94   -96   -98   -100
  -88   -89   -86   -87   -88   -90   -92   -94   -96   -98   -100   -102
  -90   -91   -88   -89   -90   -92   -94   -96   -98   -100   -102   -104
  -92   -93   -90   -91   -92   -94   -96   -98   -100   -102   -104   -106
  -94   -95   -92   -93   -94   -96   -98   -100   -102   -104   -106   -108
  -96   -97   -94   -95   -96   -98   -100   -102   -104   -106   -108   -110
  -98   -99   -96   -97   -98   -100   -102   -104   -106   -108   -110   -112
 -100  -101  -98   -99  -100  -102  -104  -106  -108  -110  -112  -114
 -102  -103  -100  -101  -102  -104  -106  -108  -110  -112  -114  -116
 -104  -105  -102  -103  -104  -106  -108  -110  -112  -114  -116  -118
 -106  -107  -104  -105  -106  -108  -110  -112  -114  -116  -118  -120
 -108  -109  -106  -107  -108  -110  -112  -114  -116  -118  -120  -122
 -110  -111  -108  -109  -110  -112  -114  -116  -118  -120  -122  -124
 -112  -113  -110  -111  -112  -114  -116  -118  -120  -122  -124  -126
 -114  -115  -112  -113  -114  -116  -118  -120  -122  -124  -126  -128
 -116  -117  -114  -115  -116  -118  -120  -122  -124  -126  -128  -130
 -118  -119  -116  -117  -118  -120  -122  -124  -126  -128  -130  -132
 -120  -121  -118  -119  -120  -122  -124  -126  -128  -130  -132  -134
 -122  -123  -120  -121  -122  -124  -126  -128  -130  -132  -134  -136
 -124  -125  -122  -123  -124  -126  -128  -130  -132  -134  -136  -138
 -126  -127  -124  -125  -126  -128  -130  -132  -134  -136  -138  -140
 -128  -129  -126  -127  -128  -130  -132  -134  -136  -138  -140  -142
 -130  -131  -128  -129  -130  -132  -134  -136  -138  -140  -142  -144
 -132  -133  -130  -131  -132  -134  -136  -138  -140  -142  -144  -146
 -134  -135  -132  -133  -134  -136  -138  -140  -142  -144  -146  -148
 -136  -137  -134  -135  -136  -138  -140  -142  -144  -146  -148  -150
 -138  -139  -136  -137  -138  -140  -142  -144  -146  -148  -150  -152
 -140  -141  -138  -139  -140  -142  -144  -146  -148  -150  -152  -154
 -142  -143  -140  -141  -142  -144  -146  -148  -150  -152  -154  -156
 -144  -145  -142  -143  -144  -146  -148  -150  -152  -154  -156  -158
 -146  -147  -144  -145  -146  -148  -150  -152  -154  -156  -158  -160
 -148  -149  -146  -147  -148  -150  -152  -154  -156  -158  -160  -162
 -150  -151  -148  -149  -150  -152  -154  -156  -158  -160  -162  -164
 -152  -153  -150  -151  -152  -154  -156  -158  -160  -162  -164  -166
 -154  -155  -152  -153  -154  -156  -158  -160  -162  -164  -166  -168
 -156  -157  -154  -155  -156  -158  -160  -162  -164  -166  -168  -170
 -158  -159  -156  -157  -158  -160  -162  -164  -166  -168  -170  -172
 -160  -161  -158  -159  -160  -162  -164  -166  -168  -170  -172  -174
 -162  -163  -160  -161  -162  -164  -166  -168  -170  -172  -174  -176
 -164  -165  -162  -163  -164  -166  -168  -170  -172  -174  -176  -178
 -166  -167  -164  -165  -166  -168  -170  -172  -174  -176  -178  -180
 -168  -169  -166  -167  -168  -170  -172  -174  -176  -178  -180  -182
 -170  -171  -168  -169  -170  -172  -174  -176  -178  -180  -182  -184
 -172  -173  -170  -171  -172  -174  -176  -178  -180  -182  -184  -186
 -174  -175  -172  -173  -174  -176  -178  -180  -182  -184  -186  -188
 -176  -177  -174  -175  -176  -178  -180  -182  -184  -186  -188  -190
 -178  -179  -176  -177  -178  -180  -182  -184  -186  -188  -190  -192
 -180  -181  -178  -179  -180  -182  -184  -186  -188  -190  -192  -194
 -182  -183  -180  -181  -182  -184  -186  -188  -190  -192  -194  -196
 -184  -185  -182  -183  -184  -186  -188  -190  -192  -194  -196  -198
 -186  -187  -184  -185  -186  -188  -190  -192  -194  -196  -198  -200
 -188  -189  -186  -187  -188  -190  -192  -194  -196  -198  -200  -202
 -190  -191  -188  -189  -190  -192  -194  -196  -198  -200  -202  -204
 -192  -193  -190  -191  -192  -194  -196  -198  -200  -202  -204  -206
 -194  -195  -192  -193  -194  -196  -198  -200  -202  -204  -206  -208
 -196  -197  -194  -195  -196  -198  -200  -202  -204  -206  -208  -210
 -198  -199  -196  -197  -198  -200  -202  -204  -206  -208  -210  -212
 -200  -201  -198  -199  -200  -202  -204  -206  -208  -210  -212  -214
 -202  -203  -200  -201  -202  -204  -206  -208  -210  -212  -214  -216
 -204  -205  -202  -203  -204  -206  -208  -210  -212  -214  -216  -218
 -206  -207  -204  -205  -206  -208  -210  -212  -214  -216  -218  -220
 -208  -209  -206  -207  -208  -210  -212  -214  -216  -218  -220  -222
 -210  -211  -208  -209  -210  -212  -214  -216  -218  -220  -222  -224
 -212  -213  -210  -211  -212  -214  -216  -218  -220  -222  -224  -226
 -214  -215  -212  -213  -214  -216  -218  -220  -222  -224  -226  -228
 -216  -217  -214  -215  -216  -218  -220  -222  -224  -226  -228  -230
 -218  -219  -216  -217  -218  -220  -222  -224  -226  -228  -230  -232
 -220  -221  -218  -219  -220  -222  -224  -226  -228  -230  -232  -234
 -222  -223  -220  -221  -222  -224  -226  -228  -230  -232  -234  -236
 -224  -225  -222  -223  -224  -226  -228  -230  -232  -234  -236  -238
 -226  -227  -224  -225  -226  -228  -230  -232  -234  -236  -238  -240
 -228  -229  -226  -227  -228  -230  -232  -234  -236  -238  -240  -242
 -230  -231  -228  -229  -230  -232  -234  -236  -238  -240  -242  -244
 -232  -233  -230  -231  -232  -234  -236  -238  -240  -242  -244  -246
 -234  -235  -232  -233  -234  -236  -238  -240  -242  -244  -246  -248
 -236  -237  -234  -235  -236  -238  -240  -242  -244  -246  -248  -250
 -238  -239  -236  -237  -238  -240  -242  -244  -246  -248  -250  -252
 -240  -241  -238  -239  -240  -242  -244  -246  -248  -250  -252  -254
 -242  -243  -240  -241  -242  -244  -246  -248  -250  -252  -254  -256
 -244  -245  -242  -243  -244  -246  -248  -250  -252  -254  -256  -258
 -246  -247  -244  -245  -246  -248  -250  -252  -254  -256  -258  -260
 -248  -249  -246  -247  -248  -250  -252  -254  -256  -258  -260  -262
 -250  -251  -248  -249  -250  -252  -254  -256  -258  -260  -262  -264
 -252  -253  -250  -251  -252  -254  -256  -258  -260  -262  -264  -266
 -254  -255  -252  -253  -254  -256  -258  -260  -262  -264  -266  -268
 -256  -257  -254  -255  -256  -258  -260  -262  -264  -266  -268  -270
 -258  -259  -256  -257  -258  -260  -262  -264  -266  -268  -270  -272
 -260  -261  -258  -259  -260  -262  -264  -266  -268  -270  -272  -274
 -262  -263  -260  -261  -262  -264  -266  -268  -270  -272  -274  -276
 -264  -265  -262  -263  -264  -266  -268  -270  -272  -274  -276  -278
 -266  -267  -264  -265  -266  -268  -270  -272  -274  -276  -278  -280
 -268  -269  -266  -267  -268  -270  -272  -274  -276  -278  -280  -282
 -270  -271  -268  -269  -270  -272  -274  -276  -278  -280  -282  -284
 -272  -273  -270  -271  -272  -274  -276  -278  -280  -282  -284  -286
 -274  -275  -272  -273  -274  -276  -278  -280  -282  -284  -286  -288
 -276  -277  -274  -275  -276  -278  -280  -282  -284  -286  -288  -290
 -278  -279  -276  -277  -278  -280  -282  -284  -286  -288  -290  -292
 -280  -281  -278  -279  -280  -282  -284  -286  -288  -290  -292  -294
 -282  -283  -280  -281  -282  -284  -286  -288  -290  -292  -294  -296
 -284  -285  -282  -283  -284  -286  -288  -290  -292  -294  -296  -298
 -286  -287  -284  -285  -286  -288  -290  -292  -294  -296  -298  -300
 -288  -289  -286  -287  -288  -290  -292  -294  -296  -298  -300  -302
 -290  -291  -288  -289  -290  -292  -294  -296  -298  -300  -302  -304
 -292  -293  -290  -291  -292  -294  -296  -298  -300  -302  -304  -306
 -294  -295  -292  -293  -294  -296  -298  -300  -302  -304  -306  -308
 -296  -297  -294  -295  -296  -298  -300  -302  -304  -306  -308  -310
 -298  -299  -296  -297  -298  -300  -302  -304  -306  -308  -310  -312
 -300  -301  -298  -299  -300  -302  -304  -306  -308  -310  -312  -314
 -302  -303  -300  -301  -302  -304  -306  -308  -310  -312  -314  -316
 -304  -305  -302  -303  -304  -306  -308  -310  -312  -314  -316  -318
 -306  -307  -304  -305  -306  -308  -310  -312  -314  -316  -318  -320
 -308  -309  -306  -307  -308  -310  -312  -314  -316  -318  -320  -322
 -310  -311  -308  -309  -310  -312  -314  -316  -318  -320  -322  -324
 -312  -313  -310  -311  -312  -314  -316  -318  -320  -322  -324  -326
 -314  -315  -312  -313  -314  -316  -318  -320  -322  -324  -326  -328
 -316  -317  -314  -315  -316  -318  -320  -322  -324  -326  -328  -330
 -318  -319  -316  -317  -318  -320  -322  -324  -326  -328  -330  -332
 -320  -321  -318  -319  -320  -322  -324  -326  -328  -330  -332  -334
 -322  -323  -320  -321  -322  -324  -326  -328  -330  -332  -334  -336
 -324  -325  -322  -323  -324  -326  -328  -330  -332  -334  -336  -338
 -326  -327  -324  -325  -326  -328  -330  -332  -334  -336  -338  -340
 -328  -329  -326  -327  -328  -330  -332  -334  -336  -338  -340  -342
 -330  -331  -328  -329  -330  -332  -334  -336  -338  -340  -342  -344
 -332  -333  -330  -331  -332  -334  -336  -338  -340  -342  -344  -346
 -334  -335  -332  -333  -334  -336  -338  -340  -342  -344  -346  -348
 -336  -337  -334  -335  -336  -338  -340  -342  -344  -346  -348  -350
 -338  -339  -336  -337  -338  -340  -342  -344  -346  -348  -350  -352
 -340  -341  -338  -339  -340  -342  -344  -346  -348  -350  -352  -354
 -342  -343  -340  -341  -342  -344  -346  -348  -350  -352  -354  -356
 -344  -345  -342  -343  -344  -346  -348  -350  -352  -354  -356  -358
 -346  -347  -344  -345  -346  -348  -350  -352  -354  -356  -358  -360
 -348  -349  -346  -347  -348  -350  -352  -354  -356  -358  -360  -362
 -350  -351  -348  -349  -350  -352  -354  -356  -358  -360  -362  -364
 -352  -353  -350  -351  -352  -354  -356  -358  -360  -362  -364  -366
 -354  -355  -352  -353  -354  -356  -358  -360  -362  -364  -366  -368
 -356  -357  -354  -355  -356  -358  -360  -362  -364  -366  -368  -370
 -358  -359  -356  -357  -358  -360  -362  -364  -366  -368  -370  -372
 -360  -361  -358  -359  -360  -362  -364  -366  -368  -370  -372  -374
 -362  -363  -360  -361  -362  -364  -366  -368  -370  -372  -374  -376
 -364  -365  -362  -363  -364  -366  -368  -370  -372  -374  -376  -378
 -366  -367  -364  -365  -366  -368  -370  -372  -374  -376  -378  -380
 -368  -369  -366  -367  -368  -370  -372  -374  -376  -378  -380  -382
 -370  -371  -368  -369  -370  -372  -374  -376  -378  -380  -382  -384
 -372  -373  -370  -371  -372  -374  -376  -378  -380  -382  -384  -386
 -374  -375  -372  -373  -374  -376  -378  -380  -382  -384  -386  -388
 -376  -377  -374  -375  -376  -378  -380  -382  -384  -386  -388  -390
 -378  -379  -376  -377  -378  -380  -382  -384  -386  -388  -390  -392
 -380  -381  -378  -379  -380  -382  -384  -386  -388  -390  -392  -394
 -382  -383  -380  -381  -382  -384  -386  -388  -390  -392  -394  -396
 -384  -385  -382  -383  -384  -386  -388  -390  -392  -394  -396  -398
 -386  -387  -384  -385  -386  -388  -390  -392  -394  -396  -398  -400
 -388  -389  -386  -387  -388  -390  -392  -394  -396  -398  -400  -402
 -390  -391  -388  -389  -390  -392  -394  -396  -398  -400  -402  -404
 -392  -393  -390  -391  -392  -394  -396  -398  -400  -402  -404  -406
 -394  -395  -392  -393  -394  -396  -398  -400  -402  -404  -406  -408
 -396  -397  -394  -395  -396  -398  -400  -402  -404  -406  -408  -410
 -398  -399  -396  -397  -398  -400  -402  -404  -406  -408  -410  -412
 -400  -401  -398  -399  -400  -402  -404  -406  -408  -410  -412  -414
 -402  -403  -400  -401  -402  -404  -406  -408  -410  -412  -414  -416
 -404  -405  -402  -403  -404  -406  -408  -410  -412  -414  -416  -418
 -406  -407  -404  -405  -406  -408  -410  -412  -414  -416  -418  -420
 -408  -409  -406  -407  -408  -410  -412  -414  -416  -418  -420  -422
```

Smith-Waterman algorithm

The basis of a Smith-Waterman search is the comparison of two sequences $A = (a_1 a_2 a_3 \dots a_n)$ and $B = (b_1 b_2 b_3 \dots b_m)$

The Smith-Waterman algorithm uses individual pair-wise comparisons between characters as:

$$H_{ij} = \max \begin{aligned} & H_{i-1,j-1} + s(a_i, b_j), \\ & \max_k \{H_{i-k,j} - W_k\}, \\ & \max_l \{H_{i,j-l} - W_l\}, \\ & 0 \end{aligned}$$

Following is the code step by step. Each new step, there is a comment which explaining what going to do.

```
%144133E - M.G.K.C.Piyarathna
%Assignment 2 - BioInfomatics
%Implementing matlab code for Smith Waterman Algorithm
function [] =smith_waterman(sequence_1,sequence_2)
%function is a group of statements that together perform a task.The name of
the file and of the function should be the same.
%I have input two parameters.Those are Sequences we have to provide
%Take in inputs (sequence_1, sequence_2)
%assign sequence_1 to variabe A
A = sequence_1;
%assign sequence_2 to variabe A
B = sequence_2;

%Here I have used gap penalty = -2,match = 2, mismatch = -3
d = -2;%gap penalty
m = 2;%match
s = -1;%mismatch

col = length(A);%return the length of A and assign it to the column
row = length(B);%return the length of B and assign it to the row
% rows|columns

%preallocating matrix F all zeros
F = zeros(row+1,col+1); %create an array of Zeros/preallocating matrix F
all zeros

%Filling in the matrix
for i=2:row+1
    for j=2:col+1
        %j is going through the columns which is A
        %if the two positions match,scoring is done accordingly

        if (A(j-1) == B(i-1))
            scores(A(j-1),B(i-1)) = m;
        else
            scores(A(j-1),B(i-1)) = s;
        end
        %Filling-in partial alignments here
```

```

        Diagonal      = F(i-1,j-1) + scores(A(j-1),B(i-1));
        Left = F(i, j-1) + d;
        Right = F(i-1, j) + d;
        %computing the final score of the alignment and assigning it to F
        Temp = [Diagonal Left Right 0];
        F(i,j) = max(Temp);% return the maximum of Temp
        % end
    end
end
%traceback part
%find max element
[rr,cc]=(find(F==max(F(:)))));

Alignment_A = '';
Alignment_B = '';
i = rr; %traceback starts with the maximum element of F
j = cc; %max column
TotalScore = 0;
while (F(i,j) ~= 0 )
    Score = F(i,j);
    DIAG = F(i-1,j-1);
    %LEFT = F(i-1,j);
    UP    = F(i,j-1);

    %if scores are equal to the diagonal, there is no gap.
    if (Score == DIAG + scores(A(j-1),B(i-1)))
        Alignment_A = strcat(Alignment_A, A(j-1));
        Alignment_B = strcat(Alignment_B, B(i-1));
        %computes score, checks to see if the alignment are the same
        %characters
        if (A(j-1) == B(i-1))
            TotalScore = TotalScore + 1;
        else
            %mismatch
            TotalScore = TotalScore + s;
        end
        i = i-1;
        j = j-1;

        %gap in sequence B
    elseif (Score == UP + d)
        Alignment_A = strcat(Alignment_A, A(j-1));%String concatenation
        Alignment_B = strcat(Alignment_B, '-');%String concatenation
        j = j-1;

        %gap in sequence A
    else
        Alignment_A = strcat(Alignment_A, '-');%String concatenation
        Alignment_B = strcat(Alignment_B, B(i-1));%String concatenation
        i = i-1;
    end
end

%displays the alignment score and alignments
disp('Score Matrix');%print output
i = 1;
while(i<=row+1)
    disp(F(i,1:end));%print output
    i=i+1;
end

```



```

end
matches = 0;
for i=1:length(Alignment_A)
    if( Alignment_A(i) == Alignment_B(i))
        matches = matches + 1;
    end
end
matches = 0;
for i=1:length(Alignment_A)
    if( Alignment_A(i) == Alignment_B(i))
        matches = matches + 1;
    end
end
disp (strcat('Number of Matches :',num2str(matches)));%String concatenation
--print output %num2str- converts the array A into a string

disp('Final alignment for given two sequences');%print output

disp(Alignment_A);%print output
disp(Alignment_B);%print output

```

Editor - C:\Users\Kasun Chinthaka\Downloads\smith_waterman.m

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Comment % % % % Breakpoints Run Run and Advance Run and Time

FILE NAVIGATE EDIT BREAKPOINTS RUN

Untitled smith_waterman.m needleman_wunsch.m

```
1 %144133E - M.G.K.C.Piyarathna
2 %Assignment 2 - BioInformatics
3 %Implementing matlab code for Smith Waterman Algorithm
4 function [] =smith_waterman(sequence_1,sequence_2)
5 %function is a group of statements that together perform a task.The name of the file and of the function should be the same.
6 %I have input two parameters.Those are Sequences we have to provide
7 %Take in inputs (sequence_1, sequence_2)
8 %assign sequence_1 to variable A
9 A = sequence_1;
10 %assign sequence_2 to variable A
11 B = sequence_2;
12
13 %Here I have used gap penalty = -2,match = 2, mismatch = -3
14 d = -2;%gap penalty
15 m = 2;%match
16 s = -1;%mismatch
17
18 col = length(A);%return the length of A and assign it to the column
19 row = length(B);%return the length of B and assign it to the row
20 % rows\columns
21
22 %preallocating matrix F all zeros
23 F = zeros(row+1,col+1); %create an array of Zeros/preallocating matrix F all zeros
24
25 %Filling in the matrix
26 for i=2:row+1
27     for j=2:col+1
28         %j is going through the columns which is A
29         %if the two positions match,scoring is done accordingly
30
```

smith_waterman Ln 4 Col 28

Editor - C:\Users\Kasun Chinthaka\Downloads\smith_waterman.m

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Comment % % % % Breakpoints Run Run and Advance Run and Time

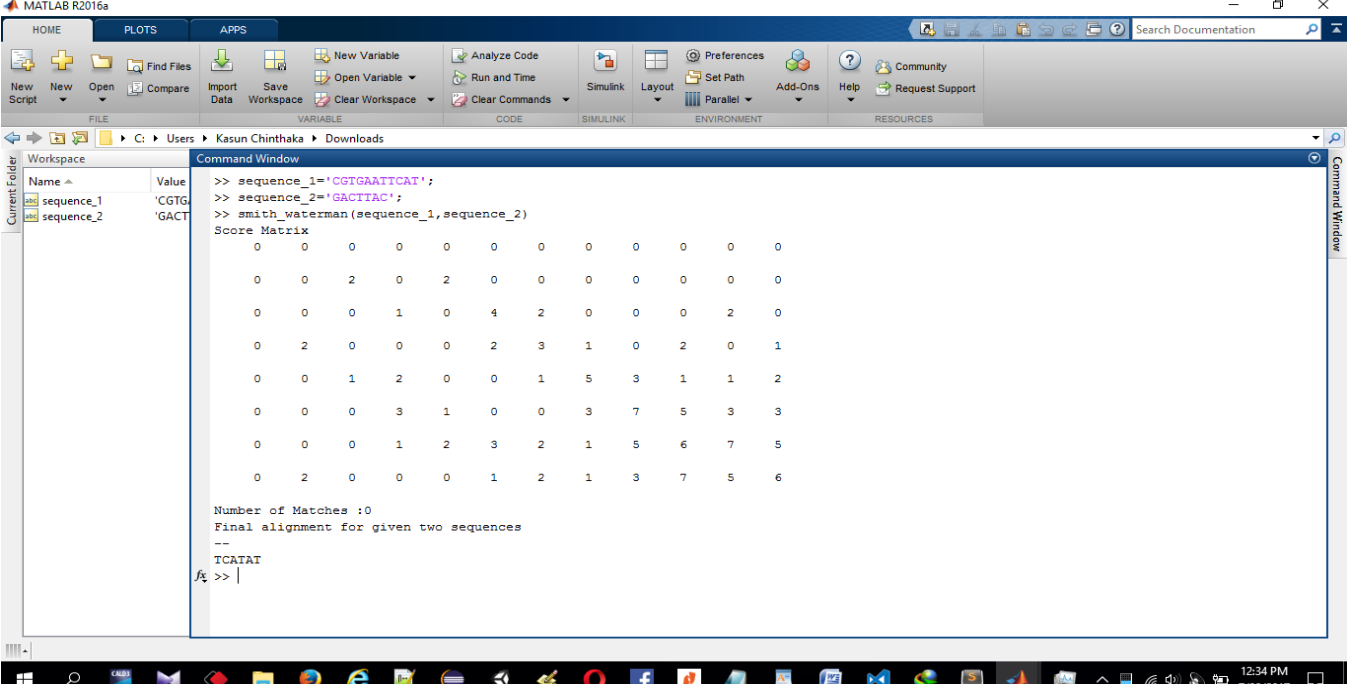
FILE NAVIGATE EDIT BREAKPOINTS RUN

Untitled smith_waterman.m needleman_wunsch.m

```
86 i = i-1;
87 end
88 end
89
90
91 %displays the alignment score and alignments
92 disp('Score Matrix');%print output
93 i = 1;
94 while(i<=row+1)
95     disp(F(i,1:end));%print output
96     i=i+1;
97 end
98 matches = 0;
99 for i=1:length(Alignment_A)
100     if( Alignment_A(i) == Alignment_B(i))
101         matches = matches + 1;
102     end
103 end
104 matches = 0;
105 for i=1:length(Alignment_A)
106     if( Alignment_A(i) == Alignment_B(i))
107         matches = matches + 1;
108     end
109 end
110 disp (strcat('Number of Matches :',num2str(matches)),'%String concatenation --print output %num2str- converts the array A into a string
111
112 disp('Final alignment for given two sequences');%print output
113
114 disp(Alignment_A);%print output
115 disp(Alignment_B);%print output
```

smith_waterman Ln 4 Col 28

Output



The image shows the MATLAB R2016a interface. The Command Window displays the following output:

```
>> sequence_1='CGTGAATTCAT';
>> sequence_2='GACTTAC';
>> smith_waterman(sequence_1,sequence_2)
Score Matrix
    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    2    0    2    0    0    0    0    0    0    0
    0    0    0    1    0    4    2    0    0    0    2    0
    0    2    0    0    0    2    3    1    0    2    0    1
    0    0    1    2    0    0    1    5    3    1    1    2
    0    0    0    3    1    0    0    3    7    5    3    3
    0    0    0    1    2    3    2    1    5    6    7    5
    0    2    0    0    0    1    2    1    3    7    5    6

Number of Matches :0
Final alignment for given two sequences
---
TCATAT
^>> |
```

The Workspace window shows two variables: sequence_1 with value 'CGTGAATTCAT' and sequence_2 with value 'GACTTAC'.