Introduction to Scientific Computing
Report Practical 1
Thijs Baksteen s3145034
Phil Oetinger s2966018
Mar 1, 2017

Assignment 1 Basic Needleman-Wunsch Algorithm

a. The complete code of `nw1.m` is found in Appendix A on page 4. The main structure of the program is based on Algorithm 2.1 in the syllabus, however, instead of implementing the scoring matrix w as a matrix, Thijs has instead opted to use an equality check on elements of the strings to find the correct cost.

b. Running `nw1.m` with `nw_test1.txt` as input and $p = 0, q = 4, g = 5$ results in the output

```
D =

     0     5    10    15
     5     4     9    10
    10     9     8     9
    15    10    13    12
    20    15    14    17
    25    20    15    18
    30    25    20    15
    35    30    25    20
```

which is identical to the matrix in the syllabus.

Breakdown of tasks: Thijs 100%.

Assignment 2 Needleman-Winsch with Predecessors

a. Phil continued from Thijs's work, and started working to create a matrix P. Using the finding of a match or gap creation during the processing of matrix D, a symbol was added to matrix P with a priority of '\' > '|' > '−'. These symbols were selected according to which path is taken according to the minimum values at each point in the loop from the first assignment; \ was inserted in the case the entry to the northwest was chosen, | was inserted in the case the entry to the north was chosen, and − for the entry to the west. However, if both \ and − were predecessors \ was chosen. This would create the correct matrix following from the current location's predecessor with a northwest predecessor taking preference.

b. The resulting output was identical to p.26:

```
D =

     0     5    10    15
     5     4     9    10
    10     9     8     9
    15    10    13    12
    20    15    14    17
    25    20    15    18
    30    25    20    15
    35    30    25    20

P =

*---
|\\\
|\\\
|\\\
|\\\
||\\
|||\
|||\
```

Breakdown of tasks: Phil 100%.

Assignment 3 Needleman-Wunsch with Optimal Alignment

a. The last part was a 50/50 split by both Thijs and Phil to create a backwards walking loop that would create the three new strings `l_al`, `s_al`, and `t_al`. These three strings were created as empty strings, and were appended by updated the vector with the appropriate letters or `-` if the letter was to be skipped. This was done in a separate loop after the loop described in the first two assignments. The program starts at the bottom right entry of D and P, which represents the alignment for the entire string. The program then traverses the matrices according to the contents of P. As a `|` or `-` in P marks an insertion or deletion, at these points one of `s_al` and `t_al` is prepended with a `-`, and the other with the previous letter in the string. In other cases, both strings are prepended with the appropriate letter. Matches were similarly marked in `l_al` with `|` and non-matches with a space.

b. The resulting output of our code was:

```
GGAATGG
   || |
---AT-G
```

This was the correct output, so we moved on to file manipulation.

c. For the last part, the thing we had to find ourselves was code to output a matrix to a file properly. We attempted to use dlmwrite; however, this method did not properly format the matrix D in our output file, so we resorted to using for loops to output each line of the matrix D. The resulting output is available in appendix D.

Breakdown of tasks: Program design and implementation was 50% each as we thought about and created the code together.

The requested files `nw1.m`, `nw2.m`, `nw3.m`, and `nw3-output.txt` are contained in the subdirectory `results` of this directory.

A Appendix: Matlab code of `nw1.m`

```
% Assignment 1 - CS7
% Thijs Baksteen s3145034
% Phil Oetinger  s2966018
in=fopen('nw_test1.txt');
s=fgetl(in);
t=fgetl(in);
fclose(in);
len_s = length(s);
len_t = length(t);
p = 0;
q = 4;
g = 5;
D = [];
for i = 1:len_s+1
  D(i,1) = g*(i-1);
end
for j = 1:len_t+1
  D(1,j) = g*(j-1);
end
for i = 2:len_s+1
  for j = 2:len_t+1
    m = D(i-1,j-1);
    if s(i-1) == t(j-1)
      m += p;
    else
      m += q;
    end
    d = D(i-1,j) + g;
    d2 = D(i,j-1) + g;
    D(i,j) = min([m,d,d2]);
  end
end
D
```

B Appendix: Matlab code of nw2.m

```
% Assignment 1 - CS7
% Thijs Baksteen s3145034
% Phil Oetinger  s2966018
in=fopen('nw_test1.txt');
s=fgetl(in);
t=fgetl(in);
fclose(in);
len_s = length(s);
len_t = length(t);
p = 0;
q = 4;
g = 5;
D = [];
P = char([]);
for i = 1:len_s+1
  D(i,1) = g*(i-1);
  P(i,1) = '|';
end
for j = 1:len_t+1
  D(1,j) = g*(j-1);
  P(1,j) = '-';
end
P(1,1)= '*';
for i = 2:len_s+1
  for j = 2:len_t+1
    % m=match, d=delete, d2=delete2
    m = D(i-1,j-1);
    if s(i-1) != t(j-1)
      m += q;
    else
      m += p;
    end
    d = D(i-1,j) + g;
    d2 = D(i,j-1) + g;
    D(i,j) = min([m,d,d2]);
    if min([m,d,d2]) == m
      P(i,j) = '\';
    elseif min([m,d,d2]) == d
      P(i,j) = '|';
    else
      P(i,j) = '-';
    end
  end
end
D
P
```

C Appendix: Matlab code of nw3.m

```
% Assignment 1 - CS7
% Thijs Baksteen s3145034
% Phil Oetinger  s2966018
in=fopen('nw_test1.txt');
s=fgetl(in);
t=fgetl(in);
fclose(in);
len_s = length(s);
len_t = length(t);
p = 0;
q = 4;
g = 5;
D = [];
P = char([]);
for i = 1:len_s+1
  D(i,1) = g*(i-1);
  P(i,1) = '|';
end
for j = 1:len_t+1
  D(1,j) = g*(j-1);
  P(1,j) = '-';
end
P(1,1)= '*';
l_al = '';
for i = 2:len_s+1
  for j = 2:len_t+1
    match = D(i-1,j-1);
    if s(i-1) != t(j-1)
      match += q;
    else
      match += p;
    end
    delete = D(i-1,j) + g;
    delete2 = D(i,j-1) + g;
    D(i,j) = min([match,delete,delete2]);
    if min([match,delete,delete2]) == match
      P(i,j) = '\';
    elseif min([match,delete,delete2]) == delete
      P(i,j) = '|';
    else
      P(i,j) = '-';
    end
  end
end
i = len_s+1;
j = len_t+1;
s_al = "";
t_al = "";
l_al = "";
while i != 1 || j != 1
  c = P(i,j);
  if (c == ’\’)
    i -= 1;
    j -= 1;
    t_al = [t(j), t_al];
    s_al = [s(i), s_al];
    if (s(i) == t(j))
      l_al = [’|’, l_al];
    else
      l_al = [’ ’, l_al];
    end
  elseif (c == ’|’)
    i -= 1;
    l_al = [’ ’, l_al];
    s_al = [s(i), s_al];
    t_al = [’-’, t_al];
  elseif (c == ’-’)
    j -= 1;
    l_al = [’ ’, l_al];
    t_al = [t(j), t_al];
    s_al = [’-’, s_al];
  end
end
output=fopen(’nw3-output.txt’, ’w’);      % open file
fprintf(output,’Name: Philip Oetinger, Thijs Baksteen\n’); % enter your name(s)
fprintf(output,’IBC, Practical 3\n\n’);
fprintf(output,’\n\nString s:\n’);
for i=1:length(s)
  fprintf(output,’%s’,s(i));
end
fprintf(output,’\n\nString t:\n’);
for i=1:length(t)
  fprintf(output,’%s’,t(i));
end

fprintf(output,’\n\nMatrix D:\n\n’);
for i=1:len_s+1
  for j=1:len_t+1
    fprintf(output, "%4d", D(i,j));
  end
  fprintf(output, "\n");
end
fprintf(output,’\n\nMatrix P:\n\n’);
dlmwrite(output,P,’’);
fprintf(output,’\n\nAlignment:\n\n’);
fprintf(output, "\n%s\n", s_al);
fprintf(output, "%s\n", l_al);
fprintf(output, "%s\n", t_al);
fclose(output);
```

D Appendix: output file `nw3output.txt`

```
Name: Philip Oetinger, Thijs Baksteen
IBC, Practical 3



String s:
GGAATGG

String t:
ATG

Matrix D:

   0   5  10  15
   5   4   9  10
  10   9   8   9
  15  10  13  12
  20  15  14  17
  25  20  15  18
  30  25  20  15
  35  30  25  20


Matrix P:

*---
|\\\
|\\\
|\\\
|\\\
||\\
|||\
|||\


Alignment:


GGAATGG
   || |
---AT-G
```