

Ps5: Needleman-Wunch DNA analysis programs

Assignment Description:

Ps5 tasked us to create a program that checks for the similarity of 2 strings using the Needleman-Wunch algorithm to obtain the edit distance for 2 strings of DNA. This edit distance is based on penalty values from a 2D array of integers that follows the following rules:

- 2pt penalty for gap
- 1pt penalty for different characters
- 0pt penalty for same character

This assignment is able to determine the edit distance for string of the ecoli genome of up to 28000 nucleotide pairs. The program runs until the array hits [0][0] where it checks the value in [0][0] which is the edit distance for the 2 strings.

Key Concepts and Algorithms:

The key algorithm for the assignment is the Needleman-Wunch algorithm. This algorithm follows the following procedure for an NxM matrix.

- Start with $[N][M] = 0$
- Add 2 to the previous element for the right column and bottom row.
- For the remaining columns and rows choose the smallest penalty:
 - $[N][M+1]$ or $[N+1][M] + 2$ to add a gap
 - $[N+1][M+1] + \text{penalty value}$

This algorithm produces the edit distance and the aligned strings.

What I learned in this assignment:

I learned how the Needleman-Wunch algorithm works and how to take a mathematical algorithm and adapt it into computer code. This assignment taught me how to take algorithms that I use in my math classes and figure out how to implement them in a program to compute the result automatically.

Ps5: Screen shot

```
TG1
AC1
AT1
TG1
CG1
TC1
GA1
CG1
CA1
CA1
GC1
CG1
CT1
GG0
AC1
TC1
TG1
TA1
TT0
GT1
CT1
TA1
GC1
CC0
GC1
TT0
T - 2
G - 2
C - 2
G - 2
T - 2
A - 2
A - 2
T - 2
- 2
- 2
- 2
- 2
- 2
T1
G1
C1
- 2
- 2
- 2
T1
T1
T1
C1
G1
C1
A1
T1
A1
Time : 0.197967
Edit distance = 160
slnogSlnog-Laptop:~/ps5$
```

Ps5 Source Code: Makefile

```
1 CC= g++
2 CFLAGS= -g -O3 -Wall -Werror -std=c++0x
3 DEPS= -lsfml-system
4
5 all:    ED
6
7 ED: main.o ED.o
8     $(CC) main.o ED.o -o ED $(DEPS)
9
10 main.o: main.cpp ED.hpp
11     $(CC) -c main.cpp ED.hpp $(CFLAGS)
12
13 ED.o:    ED.cpp ED.hpp
14     $(CC) -c ED.cpp ED.hpp $(CFLAGS)
15
16 clean:
17     rm *.o
18     rm *.gch
19     rm ED
```

Ps5 Source Code: main.cpp

```
1 #include "ED.hpp"
2
3 int main(int argc, char* argv[]){
4     sf::Clock clock;
5     sf::Time time;
6
7     std::string a;
8     std::string b;
9     std::cin >> a >> b;
10
11     ED obj(a, b);
12     int dist = obj.OptDistance();
13     std::string align = obj.Alignment();
14     std::cout << "Edit distance = " << dist << std::endl;
15     std::cout << align << std::endl;
16
17     time = clock.getElapsedTime();
18     std::cout << "Time : " << time.asSeconds() << std::endl;
19     std::cout << "Edit distance = " << dist << std::endl;
20 }
```

```
21     return 0;
22 }
```

Ps5 Source Code: ED.hpp

```
1  #ifndef ED_HPP
2  #define ED_HPP
3
4  #include <iostream>
5  #include <sstream>
6  #include <stdexcept>
7  #include <string>
8  #include <vector>
9  #include <SFML/System.hpp>
10
11 class ED{
12 public:
13     ED(std::string a, std::string b);
14     static int penalty(char a, char b){
15         if(a == b){
16             return 0;
17         }
18         else{
19             return 1;
20         }
21     }
22     static int min(int a, int b, int c){
23         int minimum = 999999;
24         if (a < minimum){
25             minimum = a;
26         }
27         if (b < minimum){
28             minimum = b;
29         }
30         if (c < minimum){
31             minimum = c;
32         }
33         return minimum;
34     }
35     int OptDistance();
36     std::string Alignment();
37 private:
38     std::string A;
39     std::string B;
```

```

40     std::vector<std::vector<int> > matrix;
41 };
42
43 #endif

```

Ps5 Source Code: ED.cpp

```

1  #include"ED.hpp"
2
3  ED::ED(std::string a, std::string b){
4      A = a;
5      B = b;
6      A = A + ' ';
7      B = B + ' ';
8  }
9
10 int ED::OptDistance()
11 {
12     int i = A.length();
13     int j = B.length();
14     int k;
15     int l;
16     for(k = 0; k <= j; k++)
17     {
18         std::vector<int> temp;
19         matrix.push_back(temp);
20
21         for(l = 0; l <= i; l++)
22         {
23             matrix.at(k).push_back(0);
24         }
25     }
26     for(k = 0; k <= j; k++){
27         matrix[k][i] = (2 * j) - (2 * k);
28     }
29     for(l = 0; l <= i; l++){
30         matrix[j][l] = (2 * i) - (2 * l);
31     }
32
33     for(k = j - 1; k >= 0; k--){
34         for(l = i - 1; l >= 0; l--){
35             matrix[k][l] = min(matrix[k+1][l] + 2, matrix[k][l+1] + 2, matrix[k+1][l+1]
36         }
37     }

```

```

38
39     return matrix[0][0];
40 }
41
42
43 std::string ED::Alignment(){
44     std::string temp;
45     int j = 0;
46     int i = 0;
47     int counter = 0;
48     int counterB = 0;
49     int path;
50     temp += A[counter];
51     temp += ' ';
52     temp += B[counterB];
53     temp += ' ';
54     if(A[counter] == B[counterB])
55         temp += '0';
56     else
57         temp += '1';
58     temp += '\n';
59     counter++;
60     counterB++;
61     while((unsigned)i < A.length() && (unsigned)j < B.length()){
62         path = min(matrix[i+1][j], matrix[i][j+1], matrix[i+1][j+1]);
63         if (path == matrix[i+1][j+1]){
64             temp += A[counter];
65             temp += ' ';
66             temp += B[counterB];
67             temp += ' ';
68             if(A[counter] == B[counterB])
69                 temp += '0';
70             else
71                 temp += '1';
72             temp += '\n';
73             i++;
74             j++;
75             counter++;
76             counterB++;
77             path = -1;
78         }
79         else if (path == matrix[i][j+1]){
80             temp += A[counter];
81             temp += ' ';
82             temp += '-';
83             temp += ' ';

```

```
84     temp += '2';
85     temp += '\n';
86     j++;
87     counter++;
88     path = -1;
89 }
90 else if (path == matrix[i+1][j]){
91     temp += A[counter];
92     temp += ' ';
93     temp += '-';
94     temp += ' ';
95     temp += '2';
96     temp += '\n';
97     i++;
98     counter++;
99     path = -1;
100 }
101
102 }
103 return temp;
104 }
```
