RNA Sequence Comparison by Needleman-Wunsch Algorithm

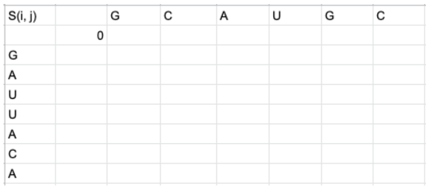
Project Description

Recently, coronavirus is exploding in China, and has a trend of spreading over the world. The rapid mutation and unavailable targeted drugs determine the high order of severity. Due to the threat from it, identifying mutation in RNA sequences of Coronavirus seems a great step on how to conquer this problem.

Ribonucleic acid (RNA) is a polymeric molecule essential in various biological roles in coding decoding, regulation and expression of genes. It is assembled as a single strand chain of nucleotides conveying genetic information using nitrogenous bases of guanine (G), uracil (U), adenine (A) and Cytosine (C) in certain orders. The purpose of the project is to take input of two sequences (consist of 'GUAC') of RNA from the user, calculate the match score and analyze the relativity between two sequences by using Needleman-Wunsch Algorithm. The software output will give the details about the relativity of two RNA sequences by showing the rearranged RNA sequences and calculate a two-dimensional matrix based on Needleman-Wunsch Algorithm.

Algorithm Introduction

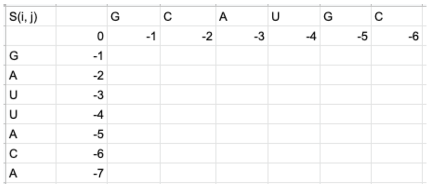
The Needleman-Wunsch Algorithm is used to generate analysis matrix which includes two DNA sequences using dynamic programing. Take 2 sequences: "GCAUGC", "GAUUACA", as example, as shown in figure1, two sequences will be labelled on both horizontal axis and vertical axis. Assuming the initial gap penalty is 1, the algorithm will calculate each individual value for first column and first row. The default cell (0, 0) is 0.



S(i, j)		G	C	A	U	G	C
	0						
G							
A							
U							
U							
A							
C							
A							

Figure 1

1. Matrix initialization
   Giving sequence $A = a_1, a_2, \dots, a_n$ and $B = b_1, b_2, \dots, b_m$, then create a matrix of size (n+1) times (m+1). Then fill the first column and first row with gap penalty, the result will be like figure2.



S(i, j)		G	C	A	U	G	C
	0	-1	-2	-3	-4	-5	-6
G	-1						
A	-2						
U	-3						
U	-4						
A	-5						
C	-6						
A	-7						

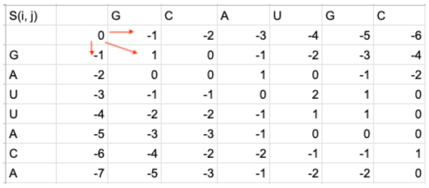
Figure 2

2. Charging matrix

Suppose the value of matrix is matrix S, the value of each cell start at (1, 1) is calculated as following:

$$S\ (i,j) = \ max \begin{cases} S(i-1,j-1) + s(a_i,b_j) \\ \qquad S(i-1,j) - d \\ \qquad S(i,j-1) - d \end{cases}$$

$S(i-1,j-1)$ is the value of matrix on the up-left corner.
$S(i-1,j)$ is the value of matrix on the left of $S\ (i,j)$.
$S(i,j-1)$ is the value of matrix on the above of $S\ (i,j)$.
$s(a_i,b_j)$ is the substitution matrix value in residue i in sequence A and j in sequence B.
d is the gap penalty.
We take match score 1, mismatch score 1, gap penalty 1 as an example, the result will be like Figure3:

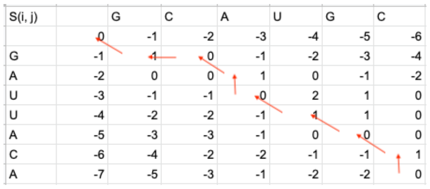


S(i, j)		G	C	A	U	G	C
	0	-1	-2	-3	-4	-5	-6
G	-1	1	0	-1	-2	-3	-4
A	-2	0	0	1	0	-1	-2
U	-3	-1	-1	0	2	1	0
U	-4	-2	-2	-1	1	1	0
A	-5	-3	-3	-1	0	0	0
C	-6	-4	-2	-2	-1	-1	1
A	-7	-5	-3	-1	-2	-2	0

Figure 3

3. Traceback and determine result

After the score matrix is filled, starting traceback from the right bottom corner, $S\ (n,m)$ to the original starting point $s(0,0)$. There are still 3 directions, up-left, left, and up. Comparing scores from all these directions, if the max is on the left, insert a gap, "-" vertically, if the max is on the up, insert a gap horizontally, if the max is on the up-left, no gap is imported. The trackback process is shown in Figure4.

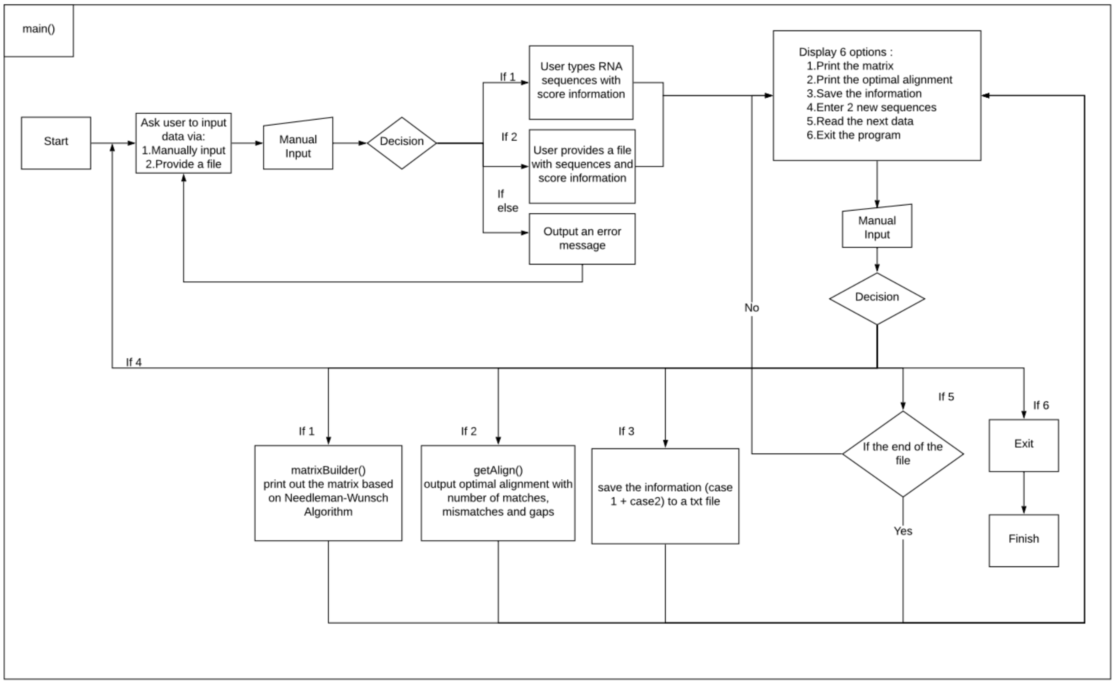


S(i, j)		G	C	A	U	G	C
	0	-1	-2	-3	-4	-5	-6
G	-1	1	0	-1	-2	-3	-4
A	-2	0	0	1	0	-1	-2
U	-3	-1	-1	0	2	1	0
U	-4	-2	-2	-1	1	1	0
A	-5	-3	-3	-1	0	0	0
C	-6	-4	-2	-2	-1	-1	1
A	-7	-5	-3	-1	-2	-2	0

Figure 4

The result after this would be: GAUUA-CA
G-CAUGC-

Flow chart


main()
Start
Ask user to input data via:
1.Manually input
2.Provide a file
Manual Input
Decision
If 1
User types RNA sequences with score information
If 2
User provides a file with sequences and score information
If else
Output an error message
Display 6 options :
1.Print the matrix
2.Print the optimal alignment
3.Save the information
4.Enter 2 new sequences
5.Read the next data
6.Exit the program
Manual Input
Decision
No
If 4
If 1
If 2
If 3
If 5
If 6
matrixBuilder()
print out the matrix based on Needleman-Wunsch Algorithm
getAlign()
output optimal alignment with number of matches, mismatches and gaps
save the information (case 1 + case2) to a txt file
If the end of the file
Yes
Exit
Finish

Explanation

1. Start the program.
2. Ask the user to select the way to input data：
   a. If choose 1, the user will input the data manually.
   b. If choose 2, the user will input the data from file by typing filename. (currently we only have one file “data1” in ./data)
   c. If else, the program will output an error message and asks user to make a new decision.
3. After inputs are collected, and the user needs to choose what kinds of results he/she wants repeatedly until exit.
   a. If choose 1, the program will call matrix_builder function to modify a matrix based on Needleman-Wunsch Algorithm, then output the matrix.
   b. If choose 2, the program will call get_align function to output optimal alignment of two RNA sequences along with number of matches, gaps and alignment length.
   c. If choose 3, the program will save all the information above to a text file.
   d. If choose 4, the program will be back to step 1.
   e. If choose 5, the program will read the next inputs from file. If it reaches the end of the file, the program will output an error message ask user to re-choose.
   f. If choose 6, the program will exit.
   g. If else, the program will output an error message and asks user to make a new decision.

Work Distribution

Yipang: matrix_builder function
check_input function
unit tests for above function

Siyu: main function
get_align function
get_max function
unit tests for above function