Journal of Computational Biology (1) 2021 01-10 Submitted 09/29; Published 09/29

First CSCI-551 Assignment, Supplemented by Sung (2009)

Joshua Ryan Steenson joshua.ryan.steenson@gmail.com
Jordan Dood jordan.dood7@gmail.com
Susan McCartney susanmccartney12@gmail.com

Editor: Joshua Ryan Steenson, Jordan Dood, and Susan McCartney

First Problem

The **Longest Common Subsequence (LCS)** problem is a special case of the global alignment problem that can be solved with a similar dynamic programming approach as the original overarching global alignment problem for sequencing (**Sung (2009)**). Evidently, both dynamic programming approaches are not the same because of the different score functions that are implemented.

For the **LCS** global alignment variant problem, given two sequences $\boldsymbol{X}$ and $\boldsymbol{Y}$, a sequence $\boldsymbol{Z}$ is said to be a common subsequence of $\boldsymbol{X}$ and $\boldsymbol{Y}$ if $\boldsymbol{Z}$ is a subsequence of both $\boldsymbol{X}$ and $\boldsymbol{Y}$ (**Sung (2009)**). With the unique scoring function for this special variant of the global alignment problem, when a match is obtained, we obtain a score of one. When there is a mismatch, there is a score of $-\infty$ because a mismatch is not tolerated for this global alignment variant (**Sung (2009)**). Lastly, we obtain a score of zero when an insertion operation or a deletion operation must be performed. Ultimately, the objective of the **LCS** is to find $\boldsymbol{Z}$ such that it is of the maximum length (**Sung (2009)**).

$\delta(x, x) = 1$
$\delta(x, y) = -\infty$, where $x \neq y$
$\delta(-, y) = \delta(x, -) = 0$

$$\mathbf{LCS}[i, j] = \max \begin{cases} \mathbf{LCS}(S[i-1], T[j]) + \delta(S[i], T[j]) \\ \mathbf{LCS}(S[i-1], T[j-1]) + \delta(-, T[j]) \\ \mathbf{LCS}(S[i], T[j-1]) + \delta(S[i], -) \end{cases}$$

```
if M[i, j] = ∅ then
    Return M[i, j] ;                    /* Let M = [n, m] (Dynamic Case) */
else
    if j > i then
        x = 0 ;                                          /* Recursive Case */
    else
        x = LCS[i, j] ;                                  /* Recursive Case */
    end
    M[i, j] = x;
    Return x;
end
```

Algorithm 1: LCS Pseudocode (LCS(S, T)) (Sung (2009))

Second Problem

Given two sequences, $\boldsymbol{S}$ and $\boldsymbol{T}$ (not necessarily the same length), let $\boldsymbol{G}$, $\boldsymbol{L}$, and $\boldsymbol{H}$ be the score of an optimal global alignment, an optimal local alignment, and an optimal global alignment without counting the beginning space of $\boldsymbol{S}$ and end space of $\boldsymbol{T}$, respectively (Sung (2009)).

1. Give an example of $\boldsymbol{S}$ and $\boldsymbol{T}$ so that all three scores $\boldsymbol{G}$, $\boldsymbol{L}$, and $\boldsymbol{H}$ are different (Sung (2009)).

Global Alignment Score	**9**
Semi-Global Alignment Score	**12**
Local Alignment Score	**13**

S	T
A	**A**
T	**G**
C	**C**
C	**A**
G	**T**
A	**G**
A	**C**
C	**A**
A	**A**
T	**T**
C	**A**
C	
A	
A	
T	
C	
G	
A	
A	
G	
C	

Global Alignment (G)

Match	2
Mismatch	−1
Insertion/Deletion	−1
S	T
A	A
T	–
C	–
C	–
G	G
A	–
A	–
C	C
A	A
T	T
C	G
C	C
A	A
A	A
T	T
C	–
G	–
A	A
A	–
G	–
C	–

Semi-Global Alignment (H)

{}	No Penalty
Match	**2**
Mismatch	**−1**
Insertion/Deletion	**−1**
S	***T***
A	**A**
T	**–**
C	**–**
C	**–**
G	**G**
A	**–**
A	**–**
C	**C**
A	**A**
T	**T**
C	**G**
C	**C**
A	**A**
A	**A**
T	**T**
C	**–**
G	**–**
A	**A**
$\overbrace{\text{A}}$	$\overbrace{\text{–}}$
G	**–**
$\underbrace{\text{C}}$	$\underbrace{\text{–}}$

Local Alignment (L)

Match	2
Mismatch	$-\infty$
Insertion/Deletion	-1
S	***T***
A	A
–	G
C	C
A	A
T	T
–	G
C	–
C	C
A	A
A	A
T	T

2. Prove or disprove the statement stating that $L \geq H \geq G$ (**Sung (2009)**).

   Given two strings of arbitrary length, $\boldsymbol{S}$ and $\boldsymbol{T}$, where $|S| = |T|$, the string can be scored as follows. The score for mismatch is presented the way it is because mismatch is not allowed in local alignment. Moreover, the score for insertion or deletion is presented the way it is unless the site for scoring in the strings is at the beginning of $\boldsymbol{S}$ or at the end of $\boldsymbol{T}$.

Match	**2**
Mismatch	**-1 / $-\infty$**
Insertion/Deletion	**-1**

   Before we prove or disprove the statement stating that $L \geq H \geq G$, some truths must first be revealed. First, semi-global alignment will contain the local alignment. Any further alignment that can be accomplished would be attributed to the fact that the local alignment is the maximum of the two substrings of $\boldsymbol{S}$ and $\boldsymbol{T}$. In so doing, a different higher score is created. Second, semi-global alignment may have internal apertures, which will only ever lower the overall alignment score. Therefore, the possibility of the local alignment score being the highest alignment score presents itself. The global alignment score would either be equal to the semi-global alignment score, or if there are insertions at the beginning of $\boldsymbol{S}$ or at the end of $\boldsymbol{T}$, then the semi-global alignment score would increase because it would not be penalized for the insertions or deletions at the beginning of $\boldsymbol{S}$ or at the end of $\boldsymbol{T}$.

   Proof. We must first consider the case of $H \geq G$. Within this case, consider the two cases where there exists or does not exist any leading blanks to $\boldsymbol{T}$ or trailing blanks of $\boldsymbol{S}$, hereafter referred to collectively as leading blanks.

In the latter case, because semi-global alignment and global alignment are nearly identical except when scoring such leading blanks, in any situation where there are no leading blanks in the optimal alignment, semi-global alignment and global alignment agree perfectly and thus have equal alignment scores. In the second case, where the leading blanks occur, they are penalized in global alignment but not in semi-global alignment. Therefore, in this case, the score for global alignment would suffer from the penalization of the leading blanks and its score would be less than that of the semi-global alignment score. Therefore $H \geq G$.

Next, we must consider the case of $L \geq H$. Within this case, consider the following two cases. In the first case, the semi-global alignment score and the local alignment score agree perfectly with reference to the optional alignment. The first case might be the case if $\boldsymbol{S}$ and $\boldsymbol{T}$ are optimal. In this case, the scores for both alignments would be equal, assuming that matches and apertures are scored the same between the algorithms. In the second case, where the semi-global alignment and the local alignment are not the same, it is clear that the local alignment will, by definition, be the optimal alignment of the substrings of the inputs. The same substring alignment will also be contained within the semi-global alignment because it is considered to be the optimal solution to the subproblem. Any additional components of the semi-global alignment will clearly result in overall penalization. Otherwise, they would have been included in the aforementioned optimal substring alignment. Thus, the score of the semi-global alignment in this second case must be strictly less than that of the local alignment. Therefore, $L \geq H$.

Therefore, $L \geq H \geq G$, by the transitive property of inequality. $\square$

Third Problem

$$V[i,j] = \max \begin{cases} V[i-1,j-1] + \delta(S[i], T[j]) \\ V[i,j-1] + \delta(-, T[j]) \\ V[i-1,j] + \delta(S[i], -) \end{cases}$$

$\delta(x,x) = 2$	$V(0,0) = 0$
$\delta(x,y) = -1$, where $x \neq y$	$V(0,j) = -j$
$\delta(-,x) = \delta(x,-) = -1$	$V(i,0) = -i$

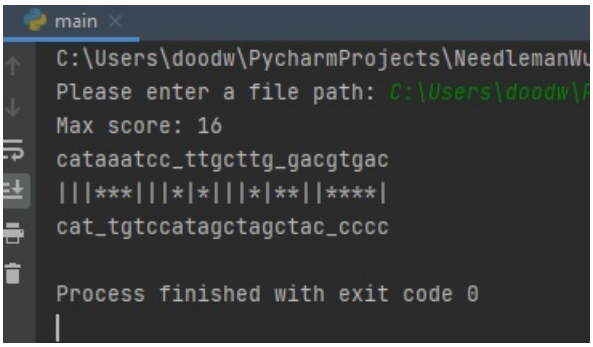

Figure 1: First Output

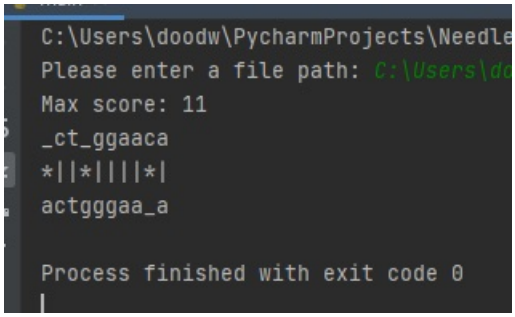

Figure 2: Second Output

References

W. Sung. *Algorithms in Bioinformatics: A Practical Introduction.* CRC Press, 2009.