

Goal

Scorites	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Carenum	C	T	T	G	T	C	G	T	C	C	G	-	-	-	G	T	T	C
Pasimachus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Pheropsophus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	C	T	T	C
Brachinus armiger	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Brachinus hirsutus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Aptinus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	C	T	T	C
Pseudomorpha	C	T	T	G	T	C	G	T	C	C	G	-	-	-	C	T	T	C

- Bioinformatics is related to the use of computers to solve biological problems
- Molecular biology is dependent on computers
- Genetics databases hold large amounts of raw data
- Strands of genetic material (e.g., DNA) are sequences of small elements called nucleotides (strings)
- DNA consists of two strands of adenine (A), cytosine (C) thymine (T) and guanine (G) nucleotides
- The aim is to find the ***longest common subsequence (LCS)*** of two DNA sequences
 - **NOTE:** The characters in a subsequence do not need to be continuous (e.g., ADE is a subsequence, not a substring, of ABCDE)

Data for a specific problem

Scorites	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Carenum	C	T	T	G	T	C	G	T	C	C	G	-	-	-	G	T	T	C
Pasimachus	C	T	T	G	T	C	G	T	C	C	G	T	-	-	-	T	T	C
Pheropsophus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	-	T	T	C
Brachinus armiger	C	T	T	G	T	C	G	T	C	C	G	-	-	-	-	T	T	C
Brachinus hirsutus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	-	T	T	C
Aptinus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	-	T	T	C
Pseudomorpha	C	T	T	G	T	C	G	T	C	C	G	-	-	-	-	T	T	C

- Consider the following two DNA sequences:
 - $S1 = \text{GCCCTAGCG}$
 - $S2 = \text{GCGCAATG}$
- An LCS of the previous sequences is GCCAG
- It is **an** LCS, rather than **the** LCS
- That is, it could be more than one solution
- How would you calculate an LCS using a *recursive technique*? and using dynamic programming?

Strategy (recursively)

Scorites	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Carenum	C	T	T	G	T	C	G	C	C	G	-	-	-	-	G	T	T	C
Pasimachus	C	T	T	G	T	C	G	C	C	G	T	-	-	-	G	T	T	C
Pheropsophus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	G	T	T	C
Brachinus armiger	C	T	T	G	T	C	G	-	-	-	-	-	-	-	T	T	T	C
Brachinus hirsutus	C	T	T	G	T	C	G	-	-	-	-	-	-	-	T	T	T	C
Aptinus	C	T	T	G	T	C	G	-	-	-	-	-	-	-	G	T	T	C
Pseudomorpha	C	T	T	G	T	C	G	-	-	-	-	-	-	-	G	T	T	C

- Consider the following two DNA sequences:
 - $S1 = \text{GCCCTAGCG}$
 - $S2 = \text{GCGCAATG}$
- How could we compute an LCS recursively?
 - $C1$ is the right-most element of $S1$
 - $C2$ is the right-most element of $S2$
 - $S1'$ is $S1$ without $C1$
 - $S2'$ is $S2$ without $C2$
- We have three recursive problems:
 - $L1 = \text{LCS}(S1', S2)$
 - $L2 = \text{LCS}(S1, S2')$
 - $L3 = \text{LCS}(S1', S2')$
- The solution would be whichever of these is the longest:
 - $L1$
 - $L2$
 - $L3 + 1$ if $C1$ is equals to $C2$, or $L3$ if $C1$ is not equal to $C2$
- The base case is:
 - Whenever $S1$ or $S2$ is a zero-length string. Then, the LCS of $S1$ or $S2$ is 0

Strategy (I)

- From top to bottom and from left to right
- Each cell contains the length of an LCS of the two strings prefixed up to that row and column

[illegible]

Soarites	C	T	T	A	G	T	T	G	T	A	C	C	A	-	-	-	A	T	T	T	A	C			
Carenum	C	T	T	A	G	T	T	G	T	A	C	C	A	C	-	-	T	A	C	-	T	T	A	C	
Pasinachus	C	T	T	A	G	T	T	G	T	A	C	C	A	C	-	-	-	-	A	G	T	T	T	A	C
Pheroposphus	C	T	T	A	G	T	T	G	T	T	A	C	C	A	C	-	-	-	A	T	T	T	A	C	
Brachinus armiger	C	T	T	A	G	T	T	G	T	A	C	C	A	C	-	-	-	-	A	T	T	T	T	A	C
Brachinus hirsutus	C	T	T	A	G	T	T	G	T	A	C	C	A	C	-	-	-	-	T	T	T	T	A	C	
Aptinus	C	T	T	A	G	T	T	G	T	A	C	C	A	C	-	-	-	-	A	T	T	T	A	C	
Pseudomorpho	C	T	T	A	G	T	T	G	T	A	C	C	A	C	-	-	-	-	A	T	T	T	A	C	

Strategy (II) – Base case

- Like the base case of the recursive solution

		G	C	C	C	T	A	G	C	G
	0	0	0	0	0	0	0	0	0	0
G	0									
C	0									
G	0									
C	0									
A	0									
A	0									
T	0									
G	0									

Scorites	C	T	T	G	A	T	C	G	T	C	C	A	-	-	-	T	C	T	T	T	G
Carenum	C	T	T	G	A	T	C	G	T	C	C	A	C	-	T	A	C	-	T	T	G
Psimachus	T	T	A	G	T	C	G	T	T	C	C	A	C	-	-	-	G	T	T	T	G
Pheropophus	C	T	T	G	A	T	C	G	T	T	C	C	A	-	-	-	C	T	T	T	G
Brachinus armiger	T	T	A	G	T	C	G	T	T	C	C	A	C	-	-	-	T	C	T	T	G
Brachinus hirsutus	C	T	T	G	A	T	C	G	T	T	C	C	A	-	-	-	T	C	T	T	G
Aptinus	C	T	T	G	A	T	C	G	T	T	C	C	A	-	-	-	C	T	T	T	G
Pseudomorpha	C	T	T	G	A	T	C	G	T	T	C	C	A	-	-	-	C	A	A	T	G

Examples of use

Longest common subsequence problem

Strategy (III) – General case

Scorites	C	T	T	G	A	T	C	G	T	C	C	A	-	-	-	T	C	T	T	T	G
Carenum	C	T	T	G	A	T	C	G	T	C	C	A	C	-	T	A	C	-	T	T	G
Psimachus	T	T	A	G	T	C	G	T	T	C	C	A	C	-	-	-	G	T	T	T	G
Pheropophus	C	T	T	G	A	T	C	G	T	T	C	C	A	-	-	-	C	T	T	T	G
Brachinus armiger	T	T	A	G	T	C	G	T	T	C	C	A	C	-	-	-	T	C	T	T	G
Brachinus hirsutus	C	T	T	G	A	T	C	G	T	T	C	C	A	-	-	-	T	C	T	T	G
Aptinus	C	T	T	G	A	T	C	G	T	T	C	C	A	-	-	-	C	T	T	T	G
Pseudomorpha	C	T	T	G	A	T	C	G	T	T	C	C	A	-	-	-	C	A	A	T	G

		G	C	C	C	T	A	G	C	G
	L3	0	L2	0	0	0	0	0	0	0
G	L1	0	?	1	1	1	1	1	1	1
C		0	1	2	2	2	2	2	2	2
G		0	1	2	2	2	2	3	3	3
C		0	1	2	3					
A		0								
A		0								
T		0								
G		0								

Examples of use

Longest common subsequence problem

Strategy (IV) – Fill in cell

Scarites	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Carenum	C	T	T	G	T	C	G	T	C	C	G	-	-	-	G	T	T	C
Pasimachus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Pteropsophus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	G	T	T	C
Brachinus armiger	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Brachinus hirsutus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	T	T	T	C
Aptinus	C	T	T	G	T	C	G	T	C	C	G	-	-	-	G	T	T	C
Pseudomorpha	C	T	T	G	T	C	G	T	C	C	G	-	-	-	G	T	T	C

		G	C	C	C	T	A	G	C	G
	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1
C	0	1	2	2	2	2	2	2	2	2
G	0	1	2	2	2	2	2	3	3	3
C	0	1	2	3	3	3	3	3	4	4
A	0	1	2	3	3	3	4	4	4	4
A	0	1	2	3	3	3	4	4	4	4
T	0	1	2	3	3	4	4	4	4	4
G	0	1	2	3	3	4	4	5	5	5

Examples of use

Longest common subsequence problem

Strategy (V). Tracking back to find an actual LCS

Scorites	C	T	T	G	T	C	G	T	C	C	-	-	-	T	T	T	C
Carenum	C	T	T	G	T	C	G	T	C	C	C	G	T	-	G	T	T
Pasimachus	C	T	T	G	T	C	G	T	C	C	C	G	T	-	G	T	T
Pheropsophus	C	T	T	G	T	C	G	T	C	C	C	G	T	-	G	T	T
Brachinus armiger	C	T	T	G	T	C	G	T	C	C	C	G	T	-	G	T	T
Brachinus hirsutus	C	T	T	G	T	C	G	T	C	C	C	G	T	-	G	T	T
Aptinus	C	T	T	G	T	C	G	T	C	C	C	G	T	-	G	T	T
Pseudomorpha	C	T	T	G	T	C	G	T	C	C	C	G	T	-	G	T	T

- We need to save a “pointer” to the previous cell
- We start from the last cell and we move forward the initial cell
- When the numerical value changes, we know that we have a new letter

		G	C	C	C	T	A	G	C	G
	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1
C	0	1	2	2	2	2	2	2	2	2
G	0	1	2	2	2	2	2	3	3	3
C	0	1	2	3	3	3	3	3	4	4
A	0	1	2	3	3	3	4	4	4	4
A	0	1	2	3	3	3	4	4	4	4
T	0	1	2	3	3	4	4	4	4	4
G	0	1	2	3	3	4	4	5	5	5