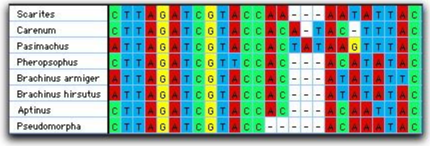
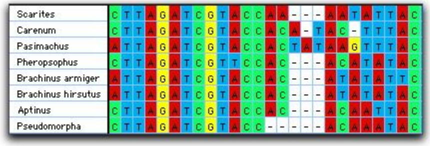
Longest common subsequence problem



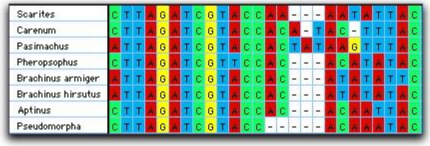
Goal

- Bioinformatics is related to the use of computers to solve biological problems
- Molecular biology is depedent 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)



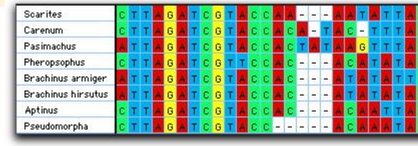
Data for a specific problem

- Consider the following two DNA sequences:
  - S1 = GCCCTAGCG
  - S2 = 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)

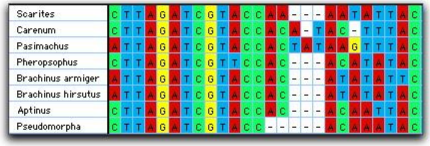
- Consider the following two DNA sequences:
  - S1 = GCCCTAGCG
  - S2 = GCGCAATG
- How could we compute an LCS recursively?
  - C1 is the right-most element of S1
  - C2 is the right-most element of S2
  - S1’ is S1 without C1
  - S2’ is S2 without C2
- We have three recursive problems:
  - L1 = LCS(S1’, S2)
  - L2 = LCS(S1, S2’)
  - L3 = 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

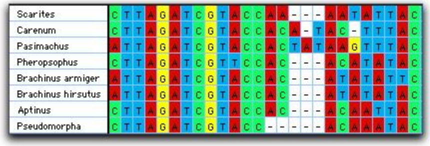
		G	C	C	C	T	A	G	C	G
G										
C										
G										
C						?				
A										
A										
T										
G										



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									

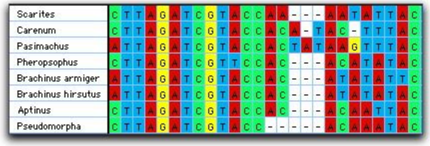


Strategy (III) – General case

		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
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						
A	0									
A	0									
T	0									
G	0									

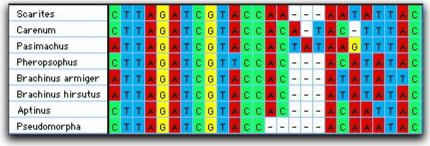
Examples of use

Longest common subsequence problem



Strategy (IV) – Fill in cell

		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



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

- 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