



Alignment problem

Alignment problem

x:	A	T	C	T	C	A	G
y:	T	G	C	A	T	A	

- Let's assume we have two sequences of strings composed by any sequence of the following four characters: 'A', 'T', 'G', and 'C'.
 - Adenine (A) - Thymine (T)
 - Guanine (G) - Cytosine (C)
- You would like to align those two sequences by inserting gaps or admitting differences.
 - Any time you have a gap, an '_' is inserted in one of the sequences. Any '_' adds a cost of 2 units to the final solution.
 - Any time you allow a difference, replace the two characters with a '*'. Any '*' adds a cost of 5 units to the final solution (10 on both strings).
- Given two strings, $X = x_1 x_2 \dots x_m$, and $Y = y_1 y_2 \dots y_n$, write the algorithm that outputs the minimum cost-aligned strings.
- The two aligned strings generated have to have the same length.

Examples

Gap Cost = 2, replace Cost = 5+5

- Input : X = CG, Y = CA,
- Output : X = CG_, Y = C_A, final cost = 4
- Input : X = AGGGCT, Y = AGGCA,
- Output : X = AGGGC*, Y = A_GGC*, final cost = 12

- Input : X = CG, Y = CA,
- Output : X = C*, Y = C*, final cost = 10

Challenge duties

- Provide a link to a Google Colab where you show your findings.
- Provide a short video where you present your findings.
- One additional point for:
 - Given two numbers n and m :
 - generate two strings composed of 'A', 'T', 'G', and 'C' characters that produce the maximum cost
 - The gap and the replacement costs have to be parametric