Liubov Shilova (7003130)

I finally got my MN!!!

Exercise 1.

Theory (just for myself, can be skipped): For each pair (i, j) of possible cut positions of s_p and s_q we define the pairwise additional cost with respect to c by:

Csp,sq [i, j] := min{c(A++B) | A ∈ A(α i sp , α j sq), B ∈ A(σ i sp , σ j sq)} −c(A ∗),

where A(a, b) denote the set of all possible alignments of two sequences a and b.

The matrix $Cs_p,s_q := (C_{i,j})\ 0 \le i \le |s_p|,\ 0 \le j \le |s_q|$ is called the additional cost matrix of s_p and s_q with respect to c. The additional cost matrices can be easily computed using the "forward" and "reverse" pairwise alignment matrices, that is

$$C_{s,t}[i,j] = D^f_{s,t}[i,j] + D^r_{s,t}[i,j] - c_{opt}(s,t)$$

where

$$c_{opt}(s,t) = D^f_{s,t}[|\,s\,|,|\,t\,|] = D^r_{s,t}[0,0],$$

by definition.

The exercise itself:

1. Compute forward matrix
2. Compute reverse.
3. Compute (1) + (2) = T. and from each cell do (– optimal cost), which is stored in the right bottom of (1)

s = ACCG and t = TACG

D

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

D^{rev}

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

Cs_p,s_q

	A	C	C	G	e
T	0	1	2	4	6
A	0	0	1	3	5
C	2	0	0	2	4
G	4	2	0	0	2
e	6	4	2	1	0

Exercise 2

Like a normal clustering algorithm:

1. Find two closest sequences. Make the alignment. Then treat it as a
2. We can treat the alignment from (1) as a string and align other strings (or alignments) to it.
3. Again find the two closest strings or alignments and align them to each other.
4. Repeat.

The tree: each leaf is a string; each inner node is an alignment.

If we have k strings with length at most n:

Time complexity: O (n^2k^2)

Space complexity: nk (maybe not?)