Individual Assignment 2 - Inês Tavares   up201706579

Seg 1: CGCTTA   Seg 2: ACCTAA   Match: 3   Mismatch: -1   Gap: -3
TASK 1
Global alignment

Score Matrix
Seq2

	gap	A	C	C	T	A	A
gap	0	-3	-6	-9	-12	-15	-18
C	-3	-1	0	-3	-6	-9	-12
G	-6	-4	-2	-1	-4	-7	-10
C	-9	-7	-1	1	-2	-5	-8
T	-12	-10	-4	-2	4	1	-2
T	-15	-13	-7	-5	1	3	0
A	-18	-12	-10	-8	-2	4	6

Seg1 (left label)

← best alignment score

Similarity Matrix

	A	C	C	T	A	A
C	-1	3	3	-1	-1	-1
G	-1	-1	-1	-1	-1	-1
C	-1	3	3	-1	-1	-1
T	-1	-1	-1	3	-1	-1
T	-1	-1	-1	3	-1	-1
A	3	-1	-1	-1	3	3

0	-3
-3	

↓ $-3 + g = -3 + (-3) = -6$ ↘ $0 + (-1) = -1$

→ $-3 + g = -3 + (-3) = -6$ $\max(-6, -6, -1) = -1$

Traceback Matrix
Seq2

	gap	A	C	C	T	A	A
gap	0	-3	-6	-9	-12	-15	-18
C	-3	-1	0	-3	-6	-9	-12
G	-6	-4	-2	-1	-4	-7	-10
C	-9	-7	-1	1	-2	-5	-8
T	-12	-10	-4	-2	4	1	-2
T	-15	-13	-7	-5	1	3	0
A	-18	-12	-10	-8	-2	4	⑥

Seg1 (left label)

Score of the best alignment: $-1 + (-2) + 1 + 4 + 3 + 3 = 14$
Best alignment:

ACCTAA
CGCTTA

best score: 6

∴ There are not multiple best alignment

Task2
Local Alignment

Score Matrix

Seq2

	gap	A	C	C	T	A	A
gap	0	0	0	0	0	0	0
C	0	0	3	3	0	0	0
G	0	0	0	2	2	0	0
C	0	0	3	3	0	0	0
T	0	0	0	0	6	3	0
T	0	0	0	0	3	5	2
A	0	3	0	0	0	6	8

Seq₁ (row labels), ← best alignment score (circled 8)

Similarity Matrix

	A	C	C	T	A	A
C	-1	3	3	-1	-1	-1
G	-1	-1	-1	-1	-1	-1
C	-1	3	3	-1	-1	-1
T	-1	-1	-1	3	-1	-1
T	-1	-1	-1	3	-1	-1
A	3	-1	-1	-1	3	3

$$S_{1,1} = \max(S_{0,0} + sm(C,A), \; S_{0,1} + g, \; S_{1,0} + g, \; 0) = \max(-1, -3, -3, 0) = 0$$

Traceback Matrix

Seq2

	gap	A	C	C	T	A	A
gap	0	0	0	0	0	0	0
C	0	0	3	3	0	0	0
G	0	0	0	2	2	0	0
C	0	0	3	3	0	0	0
T	0	0	0	0	6	3	0
T	0	0	0	0	3	5	2
A	0	3	0	0	0	6	8

Seq₁ (row labels)

Score of the best alignment: 0 + 3 + 3 + 6 + 5 + 8 = 25

Best alignment:

A - C A A A

C G C T T A

best score: 6

∴ There are not multiple best alignment, although it could occur.

```python
def global_and_local_alignment(s1,s2,match=3,mismatch=-1,gap=-3):
    #Function to obtain the global and local alignment

    #Global alignment
    global_st = needleman_Wunsch(s1, s2,
                                  create_submat(match, mismatch, "ACGT"), gap)
    global_S = global_st[0]  # Score matrix
    global_T = global_st[1]  # Traceback matrix
    global_best_score=global_best_score = global_S[len(s1)][len(s2)]
    global_alignment = recover_align(global_T, s1, s2)  # Optimal alignment


    # Local Alignment
    local_st = smith_Waterman(s1, s2,
                              create_submat(match, mismatch, "ACGT"), gap)
    local_S = local_st[0]  # Score matrix
    local_T = local_st[1]  # Traceback matrix
    local_best_score = local_st[2]  # Best score
    local_alignment = recover_align_local(local_S, local_T, s1, s2)  # Optimal alignment
    return ("Global", global_S, global_T, global_best_score, global_alignment),("Local", local_S,
local_T, local_best_score, local_alignment)


# Test the alignment_info function
if __name__ == "__main__":
    s1 = "CGCTTA"
    s2 = "ACCTAA"
    match = 3
    mismatch = -1
    gap = -3

    global_info,local_info = global_and_local_alignment(s1, s2, match, mismatch, gap)
    # Global Alignment
    print("Global Alignment:")
    alignment_type, score_matrix, traceback_matrix, best_score, optimal_alignment = (
        global_info
    )
    print("Alignment Type:", alignment_type)
    print("Score Matrix:")
    for row in score_matrix:
        print(row)
    print("Traceback Matrix:")
    for row in traceback_matrix:
        print(row)
    print("Best Score:", best_score)
    print("Optimal Alignment:")
    for line in optimal_alignment:
        print(line)

    # Local Alignment
    print("\nLocal Alignment:")
    alignment_type, score_matrix, traceback_matrix, best_score, optimal_alignment = (local_info)
    print("Alignment Type:", alignment_type)
    print("Score Matrix:")
    for row in score_matrix:
        print(row)
    print("Traceback Matrix:")
    for row in traceback_matrix:
        print(row)
    print("Best Score:", best_score)
    print("Optimal Alignment:")
    for line in optimal_alignment:
        print(line)
```