Assignment on Dynamic Programming

CSE106: Data Structures and Algorithms I Sessional

Sequence Alignment using Dynamic Programming

Introduction

In molecular biology, comparing DNA sequences is crucial for understanding evolutionary relationships, identifying conserved genes, and detecting mutations. However, two sequences may not match exactly due to biological changes such as:

- **Substitution**: one base is replaced by another
- **Insertion**: new base(s) are added
- **Deletion**: base(s) are lost

These changes are modeled computationally using **sequence alignment**, where we try to align two sequences by introducing gaps and scoring matches, mismatches, and gaps appropriately. The goal is to find the alignment with the maximum possible score.

Alignment Types

There are two major types of pairwise sequence alignment:

Global Alignment

- Aligns the entire length of both sequences, from start to end.
- Finds the maximum score alignment while considering the entire sequences.

Local Alignment

- Finds the most similar subsequences between the two sequences.
- The alignment may start and end at any position in the sequences.
- Identifies the highest scoring alignment between subsequences, ignoring poorly matching regions.

Scoring Scheme

Your code should use variables for the scoring parameters, which are to be taken as input from the user:

- `match_score` (e.g., +1)
- `mismatch_penalty` (e.g., -1)
- `gap_penalty` (e.g., -2)

Your implementation for both global and local alignment should prompt the user to enter these values before running the algorithm.

Example of Aligned Sequences

Here is an example alignment, showing various operations using a scoring scheme where `match_score = +1`, `mismatch_penalty = -1`, and `gap_penalty = -2`:

```
A:  A   G   A   C   T
B:  A   -   A   G   T
```

A	B	Operation	Score	Biological Meaning
A	A	Match	+1	Conserved base
G	–	Gap	–2	Insertion or deletion
A	A	Match	+1	Conserved base
C	G	Mismatch	–1	Substitution mutation
T	T	Match	+1	Conserved base

Total alignment score $= +1 - 2 + 1 - 1 + 1 = \mathbf{0}$

Your Tasks

You will take two DNA sequences (strings made up of 'A', 'T', 'C', 'G') as input from the user and three integers for scoring:

- match score
- mismatch penalty
- gap penalty

You will solve two problems.

Part 1: Global Alignment

Solve the global alignment problem using dynamic programming to find the optimal (maximum score) alignment between the two complete sequences.

Your solution should output:

1. The aligned sequences with appropriate gaps
2. The maximum alignment score achieved

Part 2: Local Alignment

Solve the local alignment problem using dynamic programming to find the highest scoring alignment between subsequences of the two sequences.

Your solution should output:

1. The aligned subsequences with appropriate gaps
2. The maximum alignment score achieved

Note:

- Local alignment can be solved in $O(n \times m)$ time using dynamic programming, where n and m are the lengths of the two sequences.
- Unlike global alignment, local alignment excludes negative-scoring regions from the final alignment.

Sample Input and Output

Sample Input

```
Enter first sequence: AGACTAGTTAC
Enter second sequence: CGAAGTT
Enter match score: 1
Enter mismatch penalty: -1
Enter gap penalty: -2
```

Sample Output for Global Alignment

```
Global Alignment:
AGACTAGTTAC
CGA--AGTT--

Maximum Score: -3
```

Sample Output for Local Alignment

```
Local Alignment:
AGTT
AGTT

Maximum Score: 4
```