**Dynamic Programming in Alignment**

**Why Dynamic Programming?**

* Sequence alignment requires testing many possibilities → too slow if done manually.
* DP breaks the problem into smaller subproblems and builds up the best solution.

**Analogy**: Imagine finding the shortest path in a maze. Instead of checking every possible route, you store the best path at each step → that’s DP.

**Needleman–Wunsch Algorithm (Global Alignment)**

**Steps:**

1. Create a matrix with one sequence across the top, one down the side.
2. Initialize first row/column with gap penalties.
3. Fill the matrix:
   * Match = +1, Mismatch = –1, Gap = –2 (example scoring).
   * Each cell = max of (diagonal + match/mismatch, left + gap, top + gap).
4. Trace back from bottom-right corner to get optimal alignment.

**Example**:

Seq1: ATG

Seq2: AG

Matrix filling

**Smith–Waterman Algorithm (Local Alignment)**

* Similar to Needleman–Wunsch, but:
  1. Initialize first row/column with **0** (no penalty at start).
  2. Negative values replaced with **0** (local alignment stops when similarity ends).
  3. Trace back from the **highest scoring cell** (not always bottom-right).

**Analogy**:

Global alignment = compare **entire essay**.

Local alignment = find **best matching paragraph** inside two essays.

**🧪 Lab Session: Manual DP Alignment**

**Objective**: Practice filling DP matrices.

1. Provide students with two short sequences:

Seq1: GATT

Seq2: GCT

* + Build DP matrix (3x4).
  + Fill scores step by step.
  + Trace optimal alignment.

Compare results for global vs local alignment.

Tool support: Use [EMBOSS Needle](https://www.ebi.ac.uk/Tools/psa/emboss_needle/) (global) and [EMBOSS Water](https://www.ebi.ac.uk/Tools/psa/emboss_water/) (local) to check answers.

**Quick Review Questions**

1. Why is dynamic programming needed for sequence alignment?
2. What’s the main difference between Needleman–Wunsch and Smith–Waterman?
3. In local alignment, why do we reset negative scores to zero?