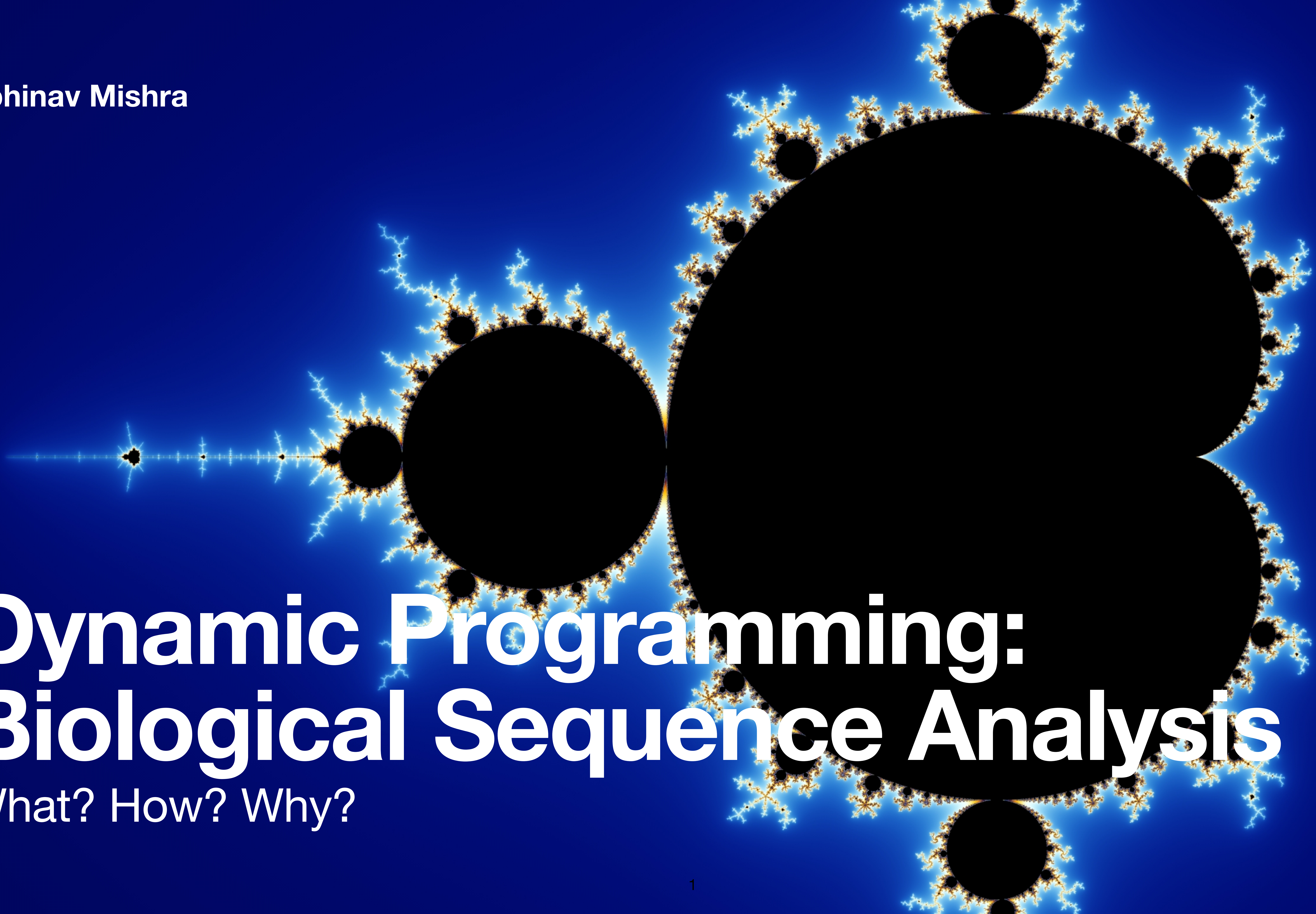
Credits: Wikipedia, Wolfgang Beyer

Abhinav Mishra



Dynamic Programming: Biological Sequence Analysis

What? How? Why?

“The problem is not to be considered solved in the mathematical sense until the structure of the optimal policy is understood.”

Richard Bellman (pg. ix, ‘Dynamic Programming’, 1957)

Biological Problem

Pairwise Sequence Alignment

Input: two sequences (DNA/Protein)

Output: alignment score (Similarity)

Goal: To optimally align the two sequences to maximise their similarity.

Why do we need an algorithm for this ?

Reducing the computational time, and cost.

Biological Problem

DP Algorithm//Initialisation

We have the two sequences ‘x’, and ‘y’.

Length of sequence ‘x’, and ‘y’ is “M”, and “N”, respectively.

So, the i_{th} residue in ‘x’ is x_i and the j_{th} residue in ‘y’ is y_j.

Parameters for scoring

- Scoring matrix $\sigma(a, b)$
- Gap Penalty γ

Biological Problem

DP Algorithm//Recursive Definiton

$$S(i,j) = \overbrace{S(M-1,N-1)+\sigma(i,j),\quad S(M-1,N)+\gamma,\quad S(M,N-1)+\gamma}^{max}$$

Case I: Residue x_i and y_j aligns to each other.

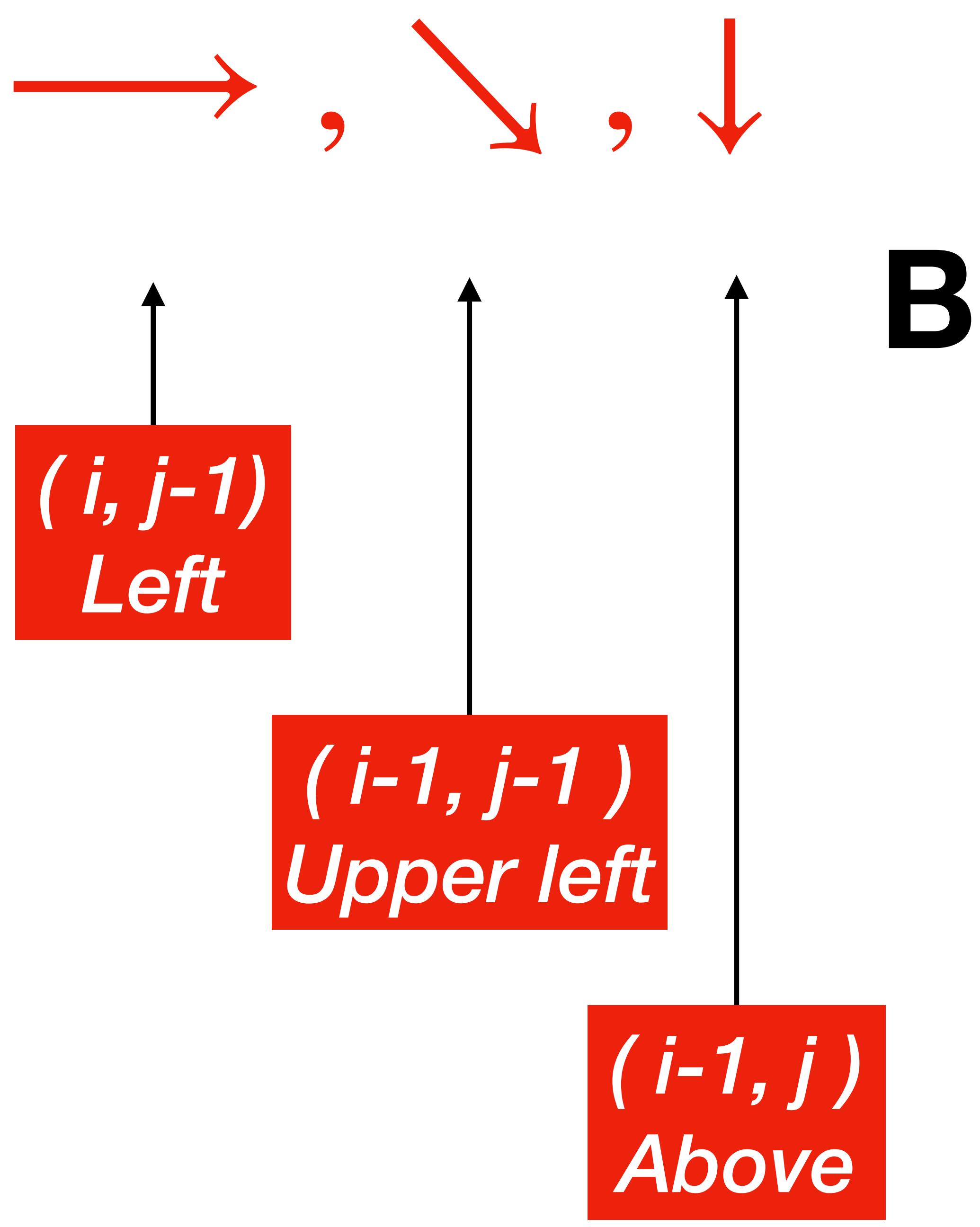
Case II: Residue x_i is aligned to a gap, and y_j appeared earlier in the alignment.

Case III: Residue y_j is aligned to a gap, and x_i appeared earlier in the alignment.

The optimal scores *S*(*i*, *j*) are tabulated in a two-dimensional matrix, with *i* running from 0...*M* and *j* running from 0…*N*.

Biological Problem

DP Algorithm//Optimal Score



$\longrightarrow, \searrow, \downarrow$

(i, j-1)
Left

(i-1, j-1)
Upper left

(i-1, j)
Above

Boundary Conditions $S(0,0) = 0 \quad S(i,0) = \gamma i \quad S(0,j) = \gamma j$

Iterate two nested loops $for(i = 0....M)$

$$for(i = 0....N)$$

Biological Problem

DP Algorithm//Traceback

Score of the optimal alignment $\boldsymbol{S_{xy} = S(M,N)}$

1. Start at cell (M,N).
2. Determine, and record which of the three cases (⟶ , ↘ , ↓) led to (M,N).
3. Follow the path back to the previous cell (one at a time).
4. Repeat until cell (0,0) is reached.
5. Retrieve the optimal path, i.e. optimal alignment.

Question 6.

X = AGT
Y = AC

Scoring Scheme

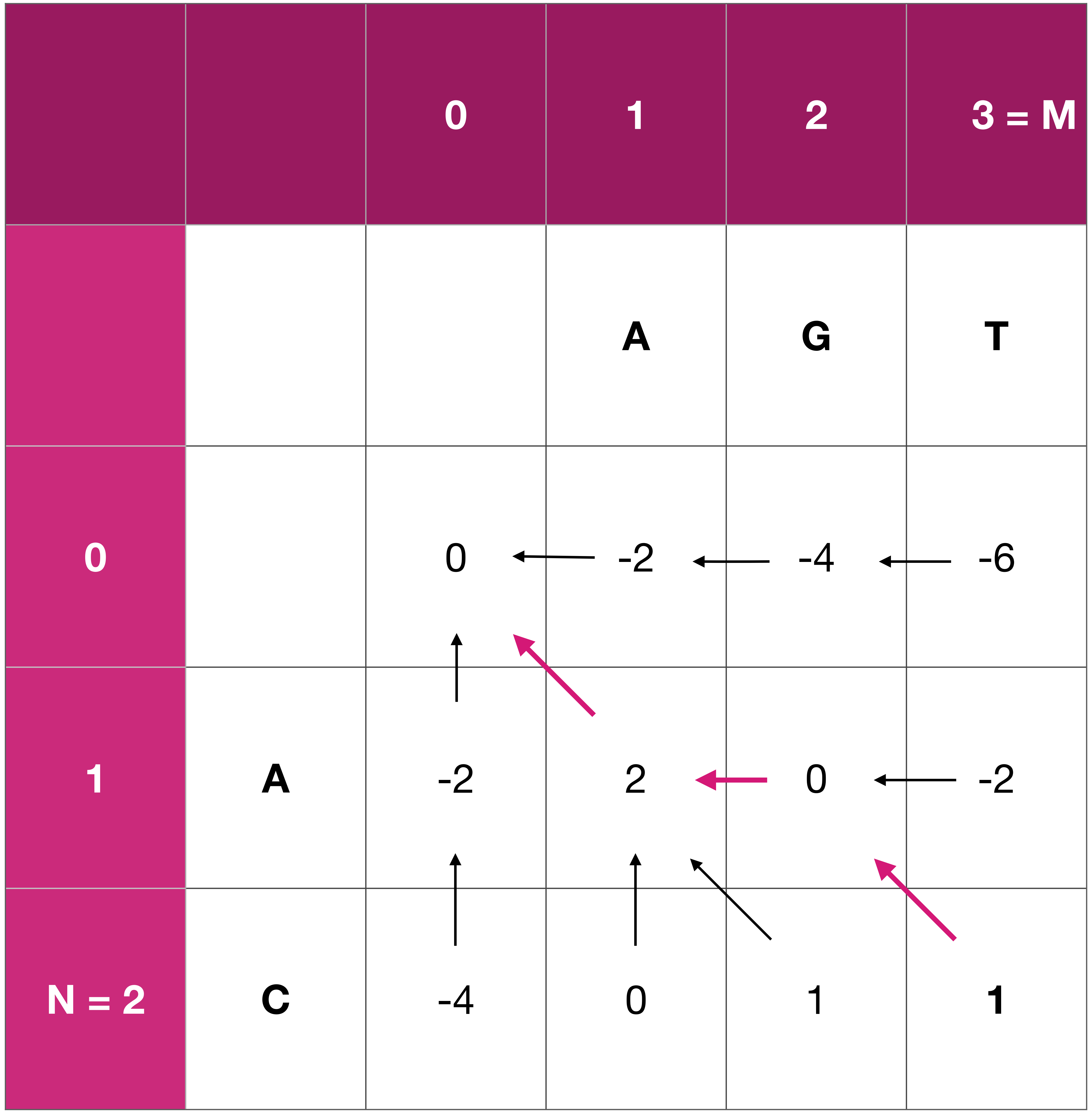
Match = +2,
Transitions: +1
Transversions: -1
Gap (INDELS) = -2

Optimal Alignment

```
A G T
|   ⋮
A - C
```

Alignment Score

$S_{xy} = 1$



		0	1	2	3 = M
			A	G	T
0		0	-2	-4	-6
1	A	-2	2	0	-2
N = 2	C	-4	0	1	**1**

“Mathematical Induction proves that we can climb as high as we like on a ladder, by proving that we can climb onto the bottom rung (the **basis**) and that from each rung we can climb up to the next one (the **step**).”

pg. 3, Concrete Mathematics, 1989

Reference

& Credit

Eddy, Sean R. "What is dynamic programming?." *Nature biotechnology* 22, no. 7 (2004): 909-910.

“Mandel zoom 00 mandelbrot set.jpg”, Created by Wolfgang Beyer with the program Ultra Fractal 3. / CC BY-SA (http://creativecommons.org/licenses/by-sa/3.0/).

Thank you for the opportunity!