CS4054
Bioinformatics

Spring 2025

Rushda Muneer

Strengthening Alignment Scoring

- Alignment score: Divided into three components:
  - match reward (+1)
  - mismatch penalty (-μ)
  - insertion/deletion penalty (-σ)

```
A T – G T T A T A
A T C G T – C – C
```

- For example, with the parameters μ = 1 and σ = 2

```
A  T  –  G  T  T  A  T  A
A  T  C  G  T  –  C  –  C
+1 +1 -2 +1 +1 -2 -1 -2 -1
```

- The alignment will be assigned a score of -4.

Global Alignment

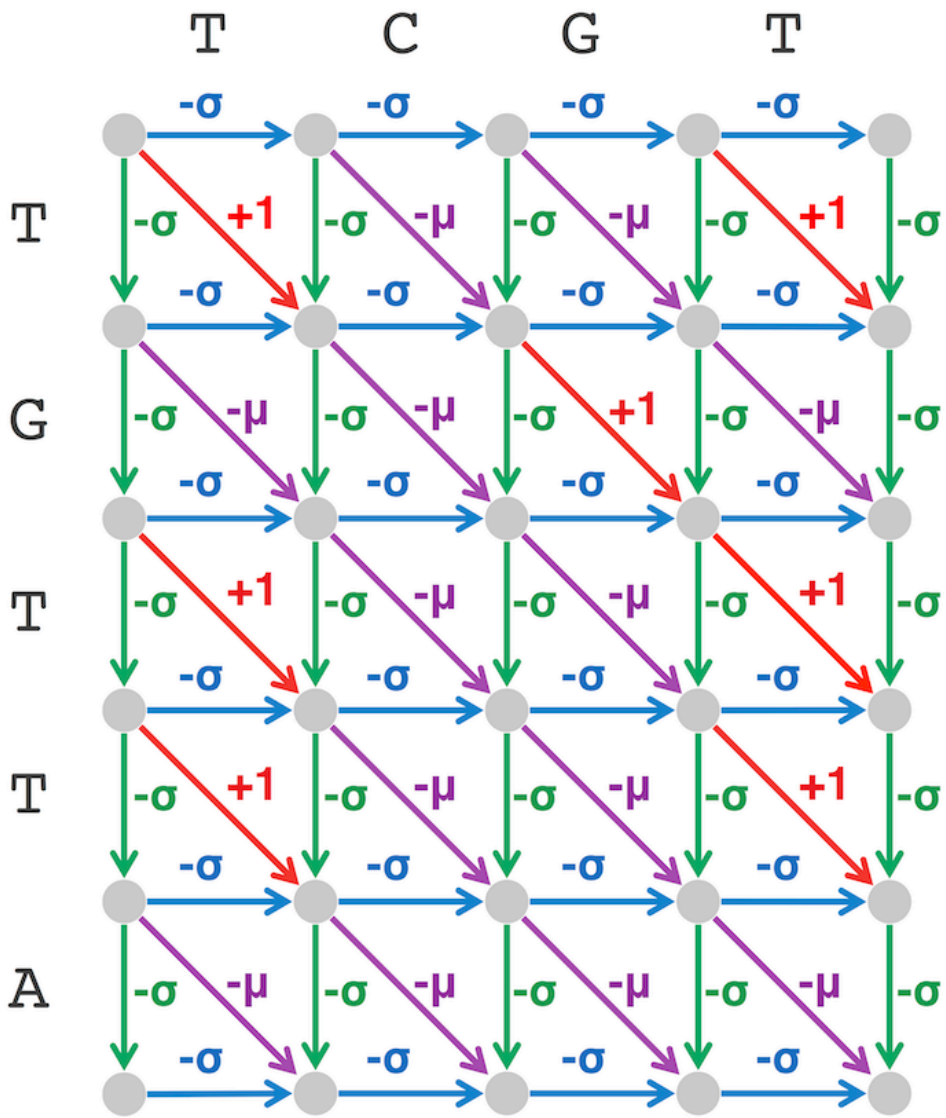
- **Global Alignment Problem:** Find a highest-scoring alignment of two strings.
  - **Input:** Two strings and numbers μ and σ .
  - **Output:** An alignment of the strings with maximum alignment score using these parameters.
- STOP: How can we modify the alignment network to solve this problem?

Global Alignment

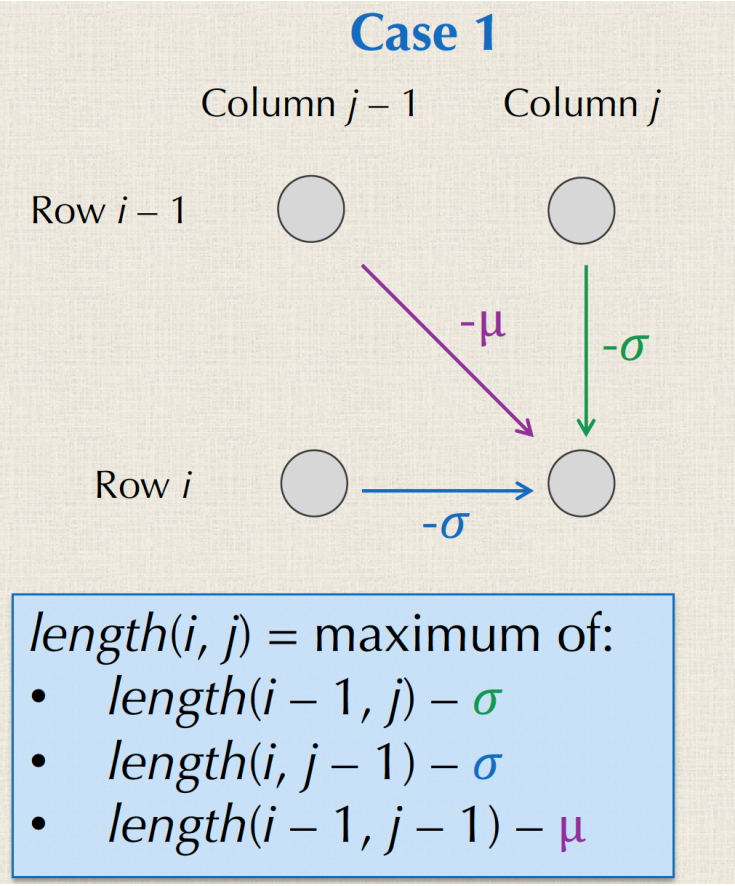
- Answer: Slight modification to alignment network

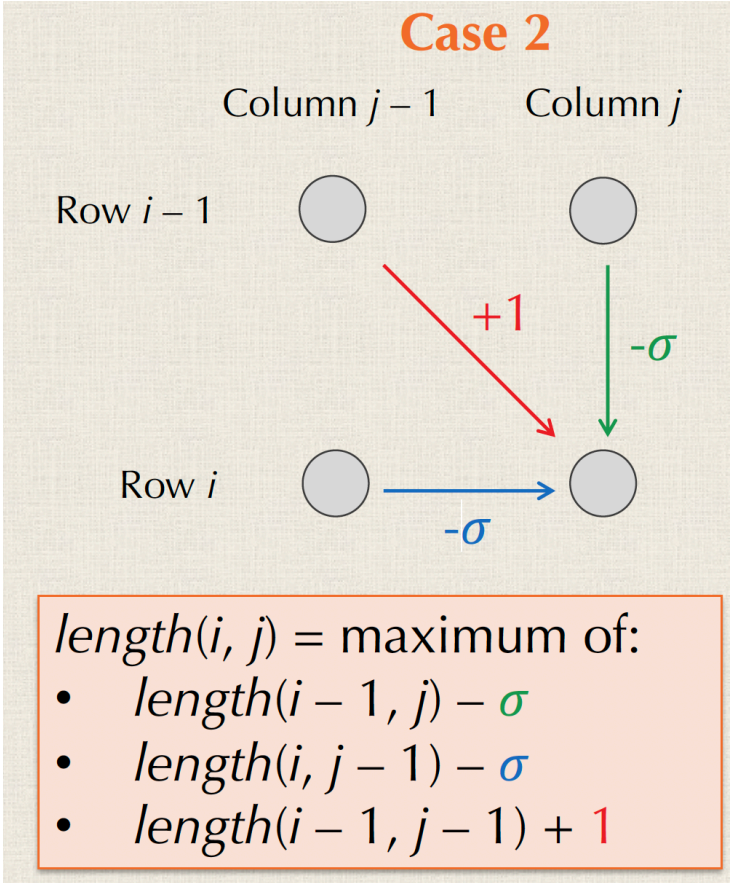
... a longest path will yield an alignment of maximum score!

- Exercise: What is the recurrence relation?



Two Cases: Mismatch vs. Match





Further Strengthening Scoring with a Scoring Matrix

- Biologists have further refined this cost function to allow for the fact that some mutations may be more likely than others, which calls for mismatches and indel penalties that differ depending on the specific symbols involved.
- Scoring matrix:
- Penalizes indels and matches/mismatches differently depending on individual symbols.

	A	C	G	T	–
A	+1	$-\mu$	$-\mu$	$-\mu$	$-\sigma$
C	$-\mu$	+1	$-\mu$	$-\mu$	$-\sigma$
G	$-\mu$	$-\mu$	+1	$-\mu$	$-\sigma$
T	$-\mu$	$-\mu$	$-\mu$	+1	$-\sigma$
–	$-\sigma$	$-\sigma$	$-\sigma$	$-\sigma$	

Strengthening Global Alignment

- **Global Alignment Problem:** Find a highest-scoring alignment of two strings.
- **Input:** Two strings and a scoring matrix.
- **Output:** An alignment of the strings with maximum alignment score according to the scoring matrix.
- Every edge simply gets weighted with the cost of the corresponding scoring matrix value.

Summarizing our Global Alignment Algorithm

1. Form a 2-D array using the recurrence relation for dynamic programming.
2. Create array containing “backtracking pointers”.
3. After reaching the sink, backtrack to source to produce a maximum-weight path.
4. Infer the alignment corresponding to this path.

Limitations of global alignment

- Analysis of **homeobox genes** offers an example of a problem for which global alignment may fail to reveal biologically relevant similarities.
- Homeobox genes are long, and they differ greatly between species, but an approximately 60 amino acid-long region in each gene, called the **homeodomain**, is highly conserved.
- Global alignment seeks similarities between two strings across their entire length
- However, when searching for homeodomains, we are looking for smaller, *local* regions of similarity and do not need to align the entire strings.

Limitations of global alignment

- For example, the global alignment between the two sequences below has 22 **matches**, 18 indels, and 2 mismatches, resulting in the score 22 − 18 - 2 = 2 (if σ = μ = 1).

```
GCC-C-AGTC-TATGT-CAGGGGGCACG--A-GCATGCACA-
GCCGCC-GTCGT-T-TTCAG----CA-GTTATGT-T-CAGAT
```

- However, these sequences can be aligned differently (with 17 matches and 32 indels) based on a highly conserved interval represented by the substrings CAGTCTATGTCAG and CAGTTATGTTCAG:

```
---G----C-----C--CAGTCTATG-TCAGGGGGCACGAGCATGCACA
GCCGCCGTCGTTTTCAGCAGT-TATGTTCAG-----A------T-----
```

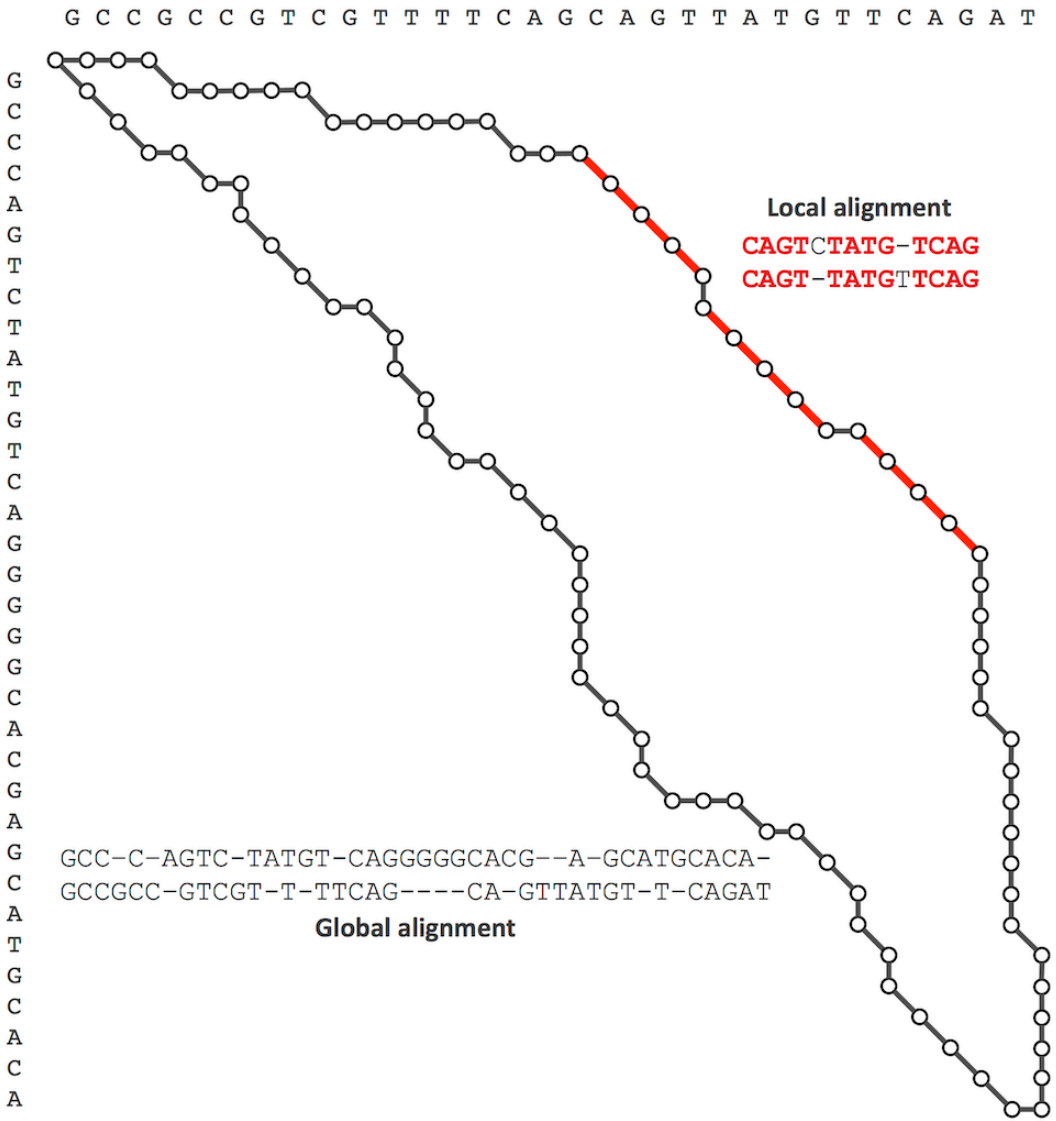
- This alignment has fewer matches and a lower score of 17 − 32 = −15, even though the conserved region of the alignment contributes a score of 12 - 2 = 10, which is hardly an accident.

From Global to Local Alignment

- Real genes have variable and conserved regions;
- When biologically significant similarities are present in some parts of sequences *v* and *w* and absent from others, biologists attempt to ignore global alignment
- Instead they align *substrings* of *v* and *w*, which yields a **local alignment** of the two strings.
- The problem of finding substrings that maximize the global alignment score over all substrings of *v* and *w* is called the **Local Alignment Problem**.

Visualizing Local Alignments

- Local alignments may be well away from “main diagonal” because they have a lot of indels on ends of the alignment.

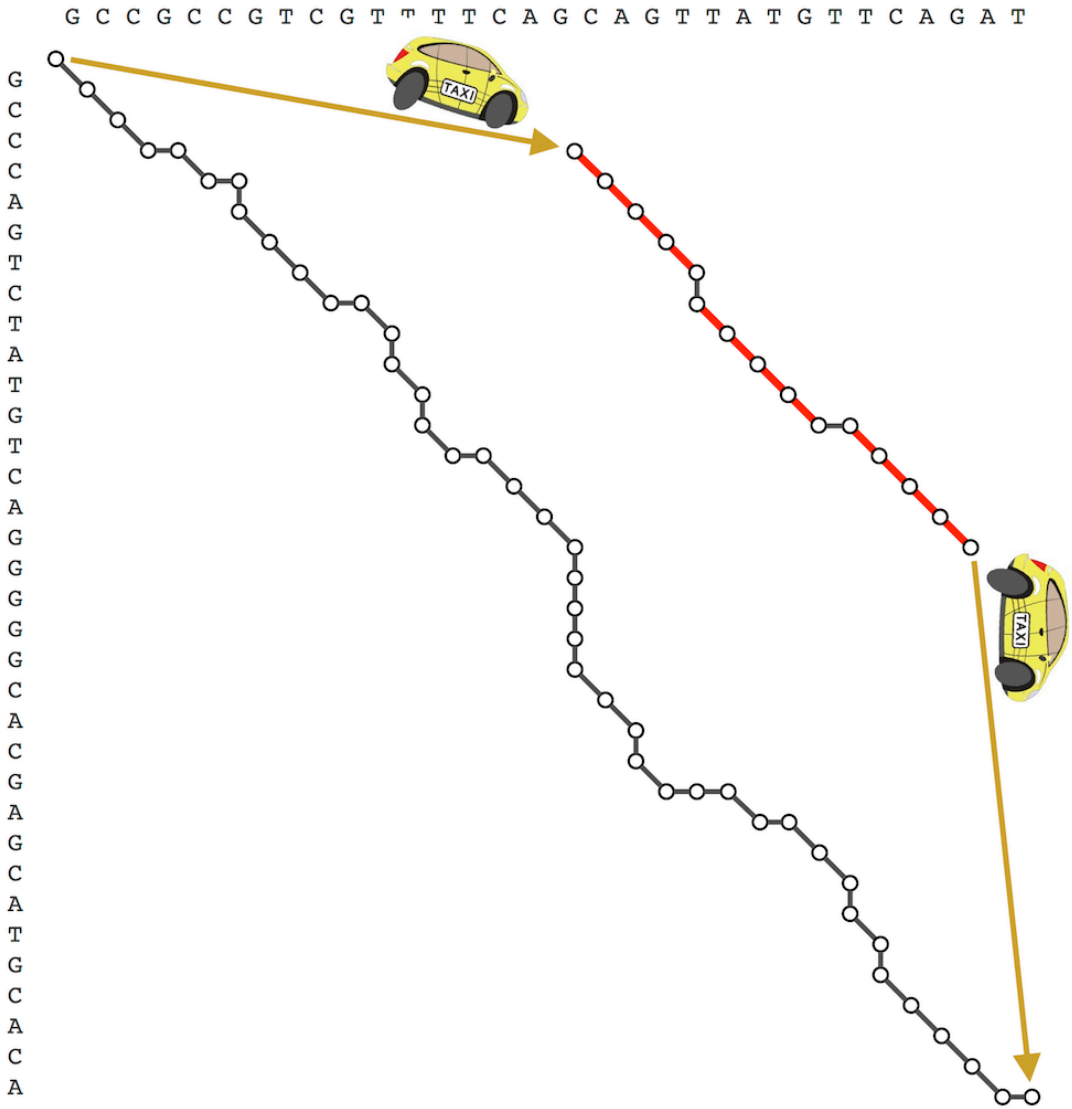


Local Alignment

- **Local Alignment Problem:**
- **Input:** Two strings v and w and a scoring matrix.
- **Output:** Substrings of v and w whose best global alignment score is maximized over all substrings.

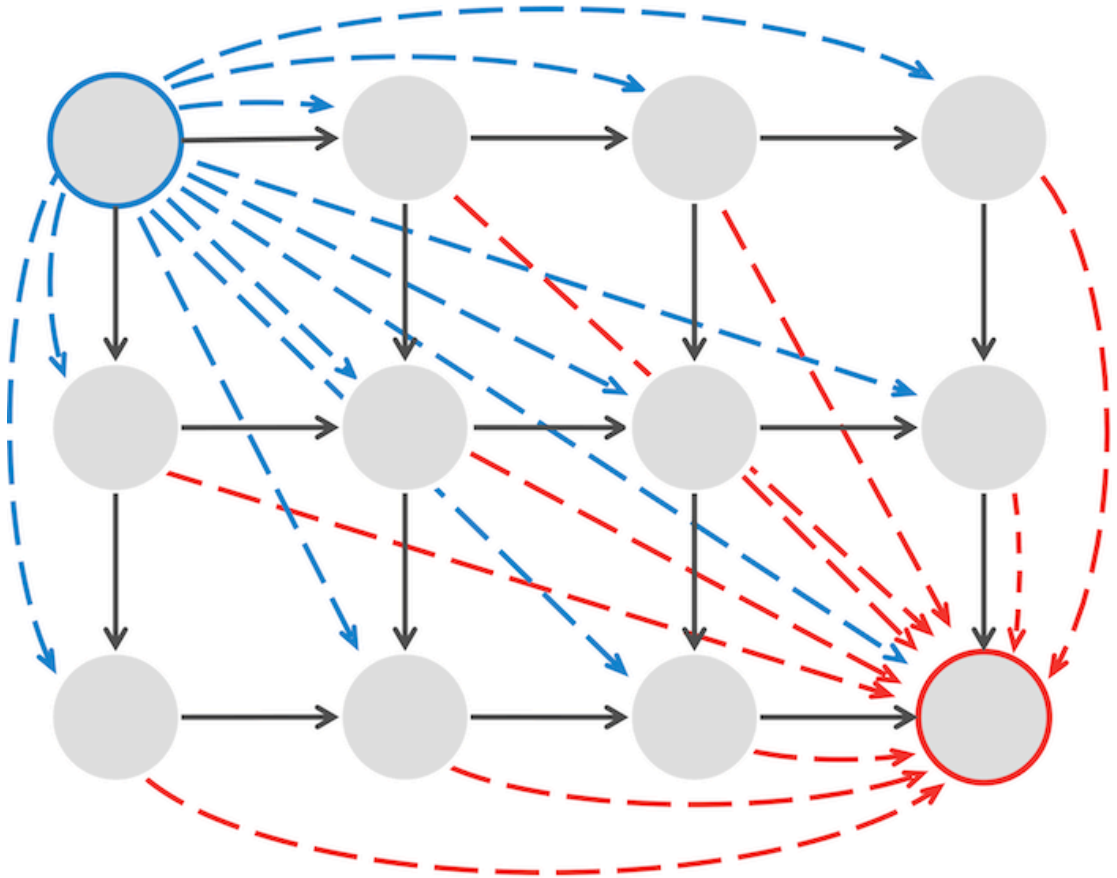
Free taxi rides in the alignment graph

- For a faster approach, imagine a “free taxi ride” from the source (0, 0) to the node representing the start node of the conserved interval.
- Imagine also a free taxi ride from the end node of the conserved interval to the sink.
- If such rides were available, then you could reach the starting node of the conserved interval for free, instead of incurring heavy penalties as in global alignment.



Local Alignment

- Connecting the source (0, 0) to every other node by adding a zero-weight edge (Blue)
- Connecting every node to the sink (*n*, *m*) by a zero-weight edge (Red)



Local Alignment

- What should be the Recurrence Relation for local alignment in the alignment graph?

$$s_{i,j} = \max \begin{cases} s_{i-1,j} & -\sigma \\ s_{i,j-1} & -\sigma \\ s_{i-1,j-1} & +1, \quad \text{if } v_i = w_j \\ s_{i-1,j-1} & -\mu, \quad \text{if } v_i \neq w_j \end{cases}$$

Global Alignment

$$s_{i,j} = \max \begin{cases} 0 & \\ s_{i-1,j} & + Score(v_i, -) \\ s_{i,j-1} & + Score(-, w_j) \\ s_{i-1,j-1} & + Score(v_i, w_j) \end{cases}$$

Local Alignment