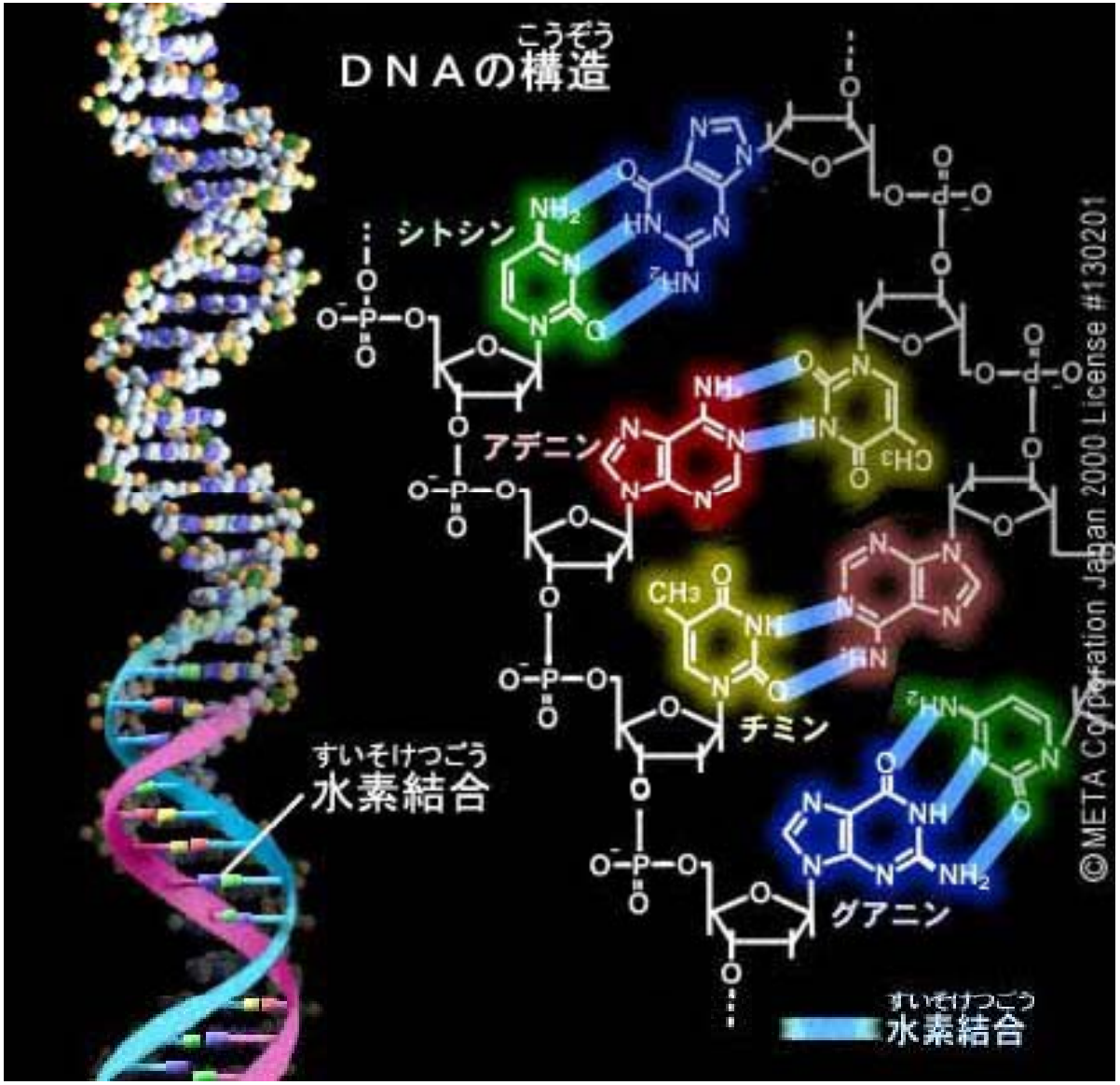
今期から始まった遠隔授業の受講者の方々にも対応するため、早めにファイルを置いていますが、授業直前に内容を修正する可能性があります。2010年4月19日

#2
Sequence Alignment

Topics:

- Sequence alignment
- Alignment score
- Sequence alignment by DP (Dynamic Programming)
- Global alignment versus Local alignment
- Local sequence alignment by DP
- Gap penalty: Linear gap and Affine gap

Deoxyribonucleic Acid (DNA)



four types of nucleobases

C: **Cytosine** *(シトシン)*

A: **Adenine** *(アデニン)*

T: **Thymine** *(チミン)*

G: **Guanine** *(グアニン)*

base complimentarity

A ··· T

G ···C

From : IPA Educational Material Archive, http://www2.edu.ipa.go.jp/gz/

base complimentarity and hydrogen bonds

A = T

two
hydrogen bonds

G ≡ C

three
hydrogen bonds

Purine base
(larger)

Pyrimidine base
(smaller)

What is Sequence Alinment?

You can find similar biological sequences, among human, mouse, cow, etc.

An alignment shows you corresponding elements in a vertical row.

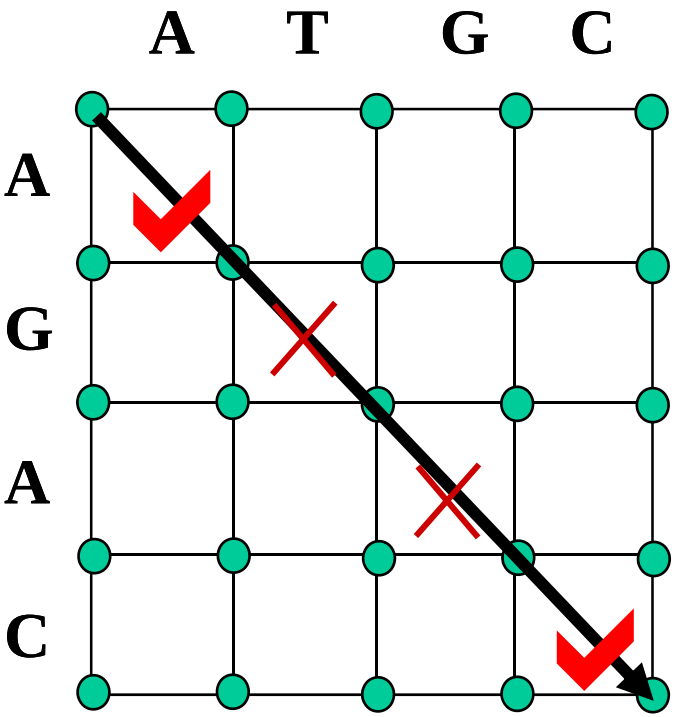
```
>BC116084|BC116084.1 Bos taurus similar to Aldehyde dehydrogenase,
            mitochondrial precursor (ALDH class 2) (ALDH1) (ALDH-E2),
            mRNA (cDNA clone MGC:138997 IMAGE:8094240), complete cds.
          Length = 2068

 Score =  127 bits (64), Expect = 6e-27
 Identities = 100/112 (89%)
 Strand = Plus / Plus


Query: 10   gggtcaactgctatgatgtgtttggagcccagtcacccttttggtggctacaagatgtcgg 69
            |||||||||||||||| |||||||| |||||||| || || ||||||||||||| |||| |
Sbjct: 1499 gggtcaactgctatgacgtgtttggggcccagtcgccgttcggtggctacaagctgtctg 1558


Query: 70   ggagtggccgggagttgggcgagtacgggctgcaggcatacactgaagtgaa 121
            ||||||||||||||| |||| || ||||||||||||||| ||||| ||||||||
Sbjct: 1559 ggagtggccgggagctggggggaatacgggctgcaggcgtacaccgaagtgaa 1610
```

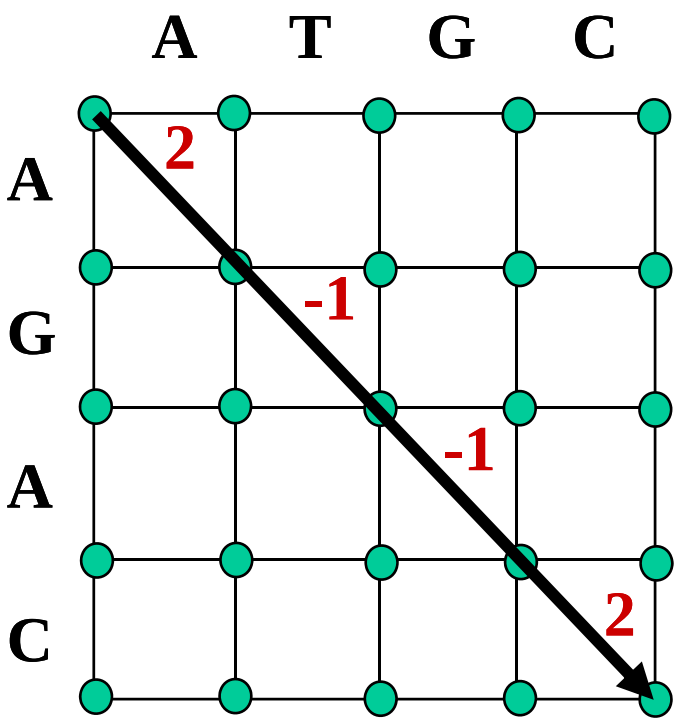
Bovine has *1 gene

In this example, Query (input) is human ALDH2 gene sequence, and Sbjct is a similar sequence found from the database.

Cow: (Latin) Bovine, (scientific name) Bos taurus.

Sequence Comparison (no gap)





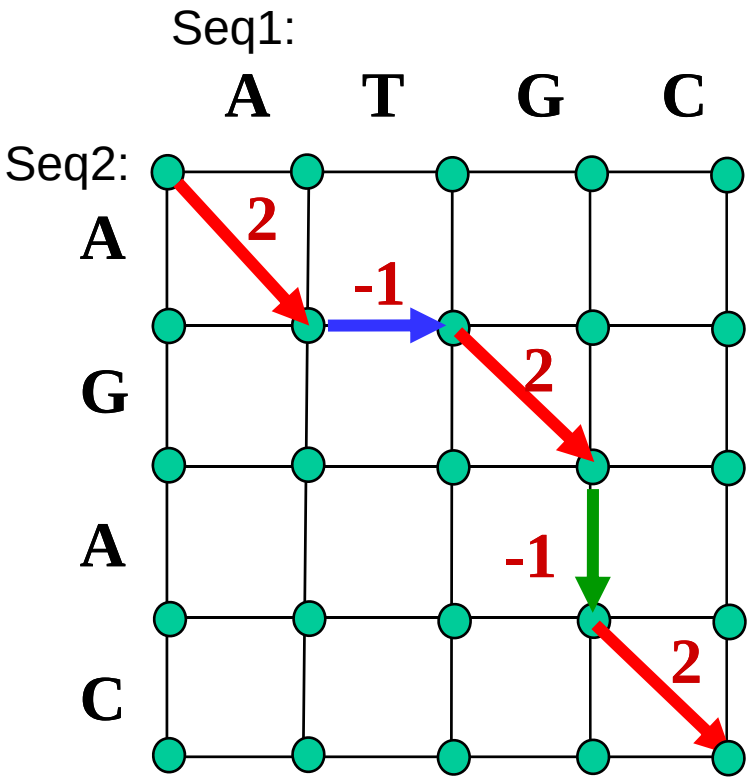
Match or Unmatch

✔ ✔ = 2

Similarity scoring

2 + (-1) + (-1) + 2 = 2

Sequence alignment



allowing "gap(s)"

$2 + (-1) + 2 + (-1) + 2 = 4$

gap

Seq1: A T G — C …

Seq2: A — G A C …

Seq1:	A	T	G	—	C	C
Seq2:	A	—	G	A	C	G
position	*1*	*2*	*3*	*4*	*5*	*6*

1, 3, 5: ***Conserved = Match***
2: ***Deletion (Seq1 → Seq2)***
4: ***Insertion (Seq1 → Seq2)***
6: ***Mutation = Unmatch***

Alignment score

alignment

```
A T T G C T T A C G T
| | . . | | | | | |
A T A A C - - - C G T
```

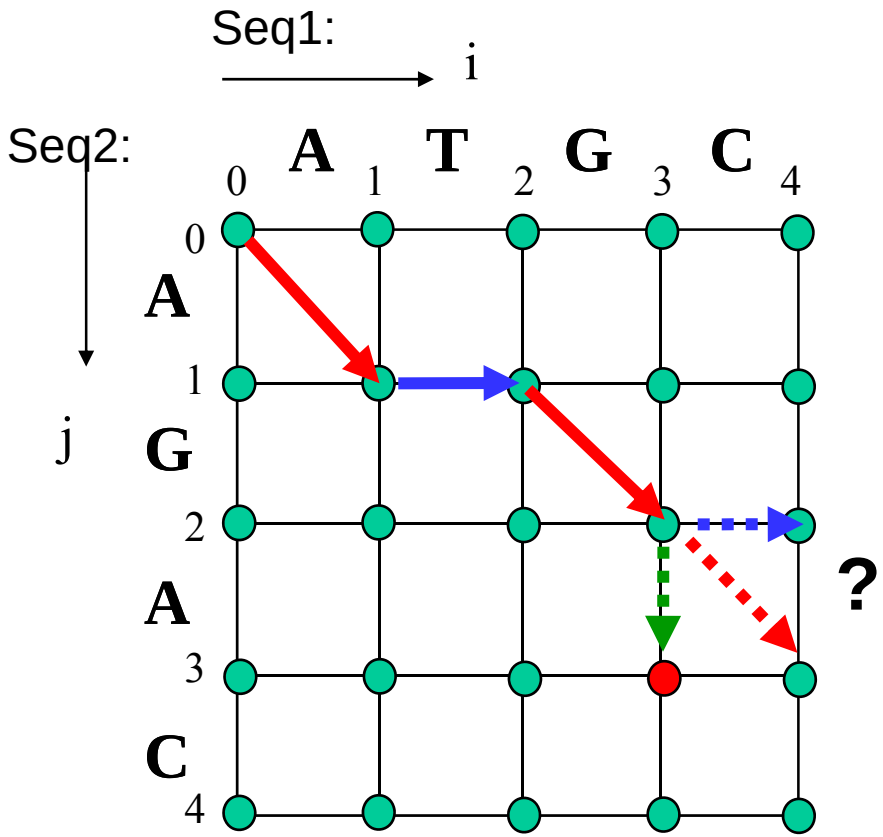
2 + 2 + (−1) + (−1) + 2 + (−1) + (−1) + (−1) + 2 + 2 + 2 = +7

Example score for DNA

match:	+2
unmatch:	-1
gap:	-1

Scoring values are subject to change, depending to the purpose of study, and/or nature of subjects.

as an optimal path problem



Start: Top Left (0,0)
End: Bottom Right (m,n)

Three directions for a move.
(No backward move. No closed cycle)

deletion (from Seq1)

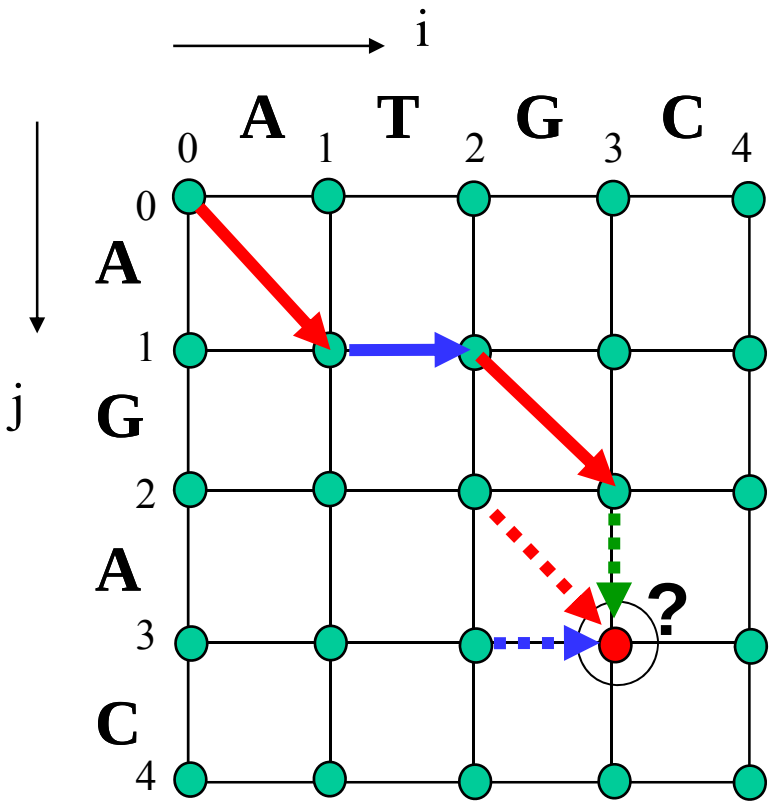
insertion (from Seq1)

conserved, or mutation

Search a path with **Maximum score**

Simple tree search algorithm may take $3^N \sim 3^{2N}$ time... (!?)

Sequence alignment by DP



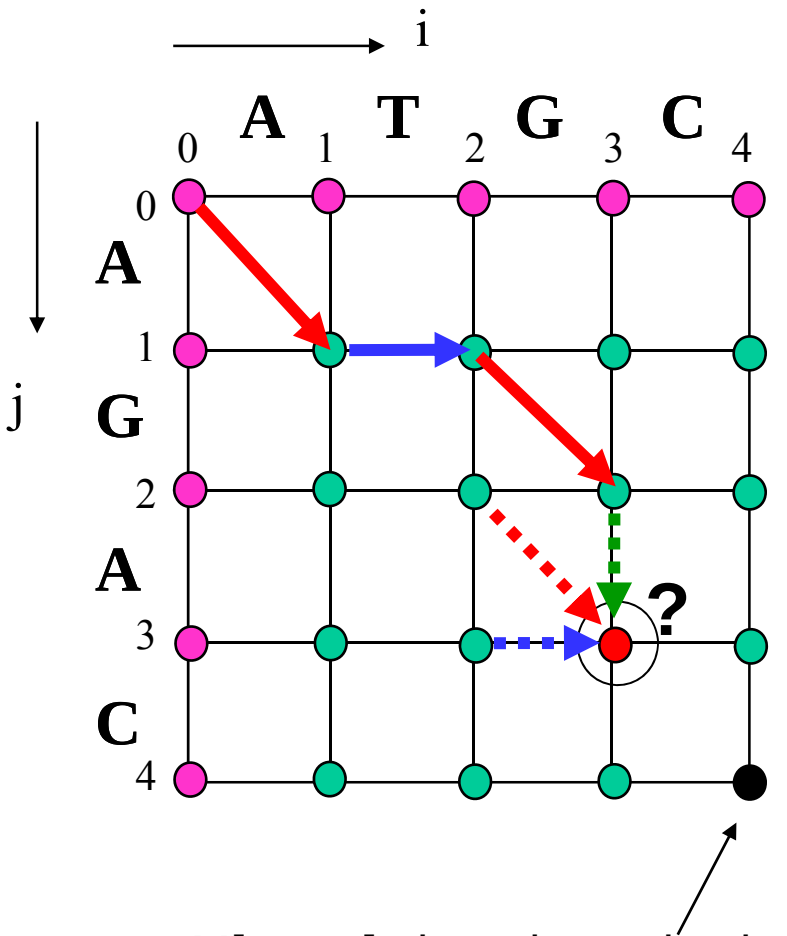
for i>0, j>0,

$$M[i,j] \leftarrow \max \begin{cases} M[i-1,j] + w \\ M[i,j-1] + w \\ M[i-1,j-1] + S[i,j] \end{cases}$$

where S[i, j] is the match/unmatch score,
w is gap penalty constant (<= 0), and
M[i, j] is **accumulated score** until [i,j].

Boundary setting.
M [i, 0] ← 0, M [0, j] ← 0
(note: here "outgap" has no penalty)
w = 0 for last column and last row.

Implementation



M[m, n] gives the optimal score for the total alignment

(1) Initialization step

Set zero on the nodes colored in pink.

$$M[i, 0] \leftarrow 0, \quad \text{for } i=0, m$$
$$M[0, j] \leftarrow 0, \quad \text{for } j=0, n$$

(2) Matrix Fill step

For all other nodes, calculate M[i, j] value (accumulated score) from Top Left toward Bottom Right value M[m.n].

(optional:) On each node, keep a pointer record to show how (from which neighbor) the accumulated score was calculated.

(3) Trace Back step

Now M[m,n] shows the optimal score.

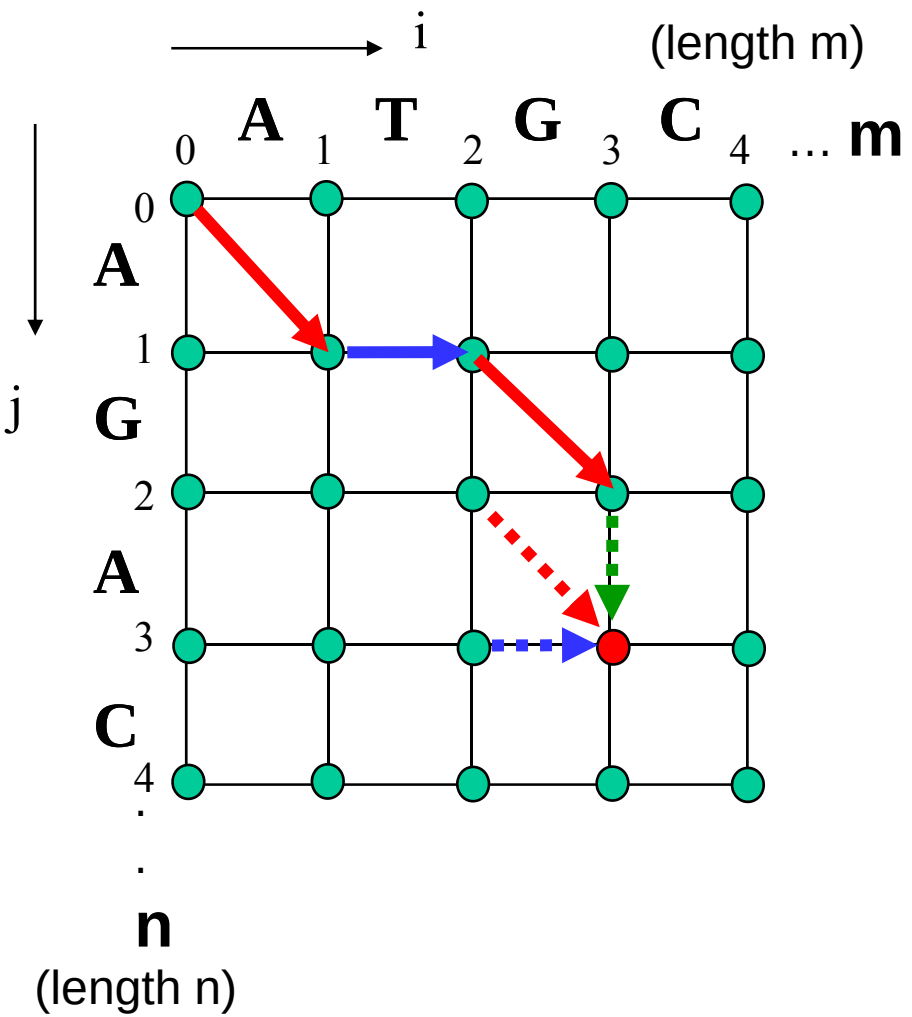
In order to find the corresponding optimal path, follows the pointers (if available) or recalculate the value and find the path, toward any of initial nodes colored in pink.

Example (Global)

ex.)

		A	T	G	C
	0	0	0	0	0
A	0	2	1	0	−1
G	0	1	1	3	2
C	0	0	0	2	5

Time and Space complexitiy



DP (Dynamic Programming)

Time complexity:

O (**m n**)

.. Matrix fill step O (**m n**)

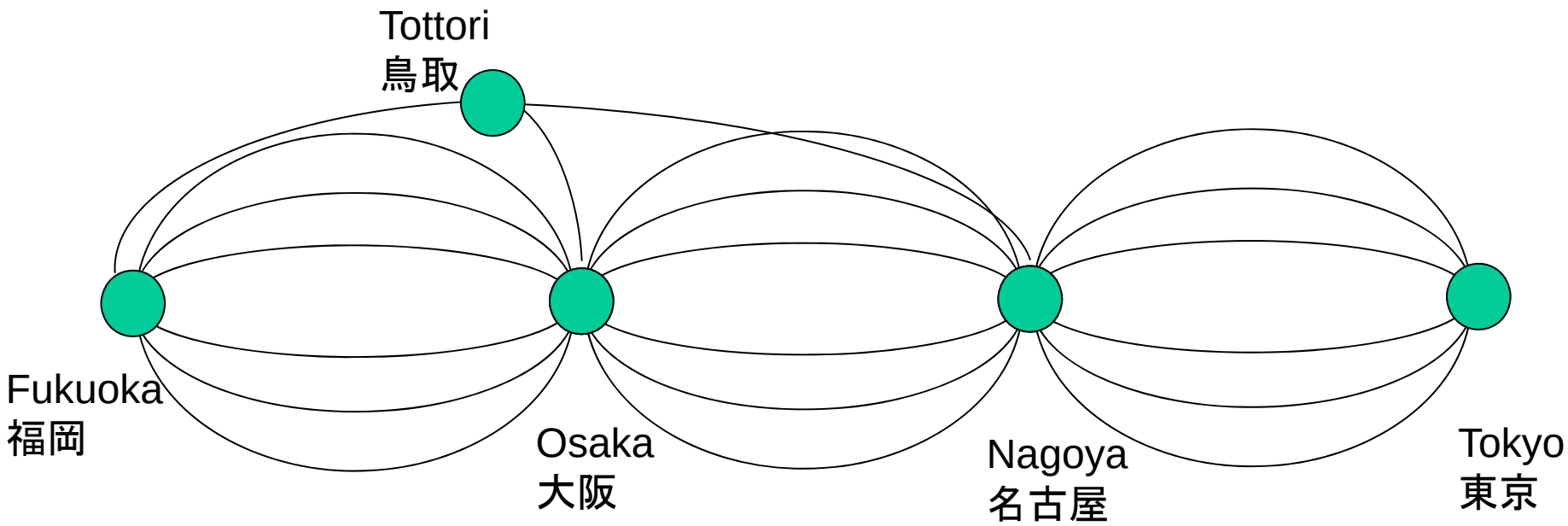
.. Traceback O (**m + n**)

Space complexity:

(memory requirement)

O (**m n**)

DP: Dynamic Programming
動的計画法



- Solve a large-scale problem, by dividing it into smaller subproblems. (optimal solutions of subproblems can be used to find the optimal solutions of the overall problem）

- **By storing and reusing the optimal solutions of subproblems,** DP can avoid redundant calculations. (This is the origin of DP's speed.)

Global vs. Local

```
A   T   T   G   C   T   C   G   T   T   G   G   A
|   .   .   .   |       |   |   |   .           |
A   A   A   A   C   -   C   G   T   A   -   -   A
+2  -1  -1  -1  +2  -1  +2  +2  +2  -1  -1  -1  +2
```

Global Alignment = (+2) x 6 + (-1) x 7 = +5

大域的アラインメント

```
-   -   -   -   C   T   C   G   T   -   -   -   -
                |       |   |   |
-   -   -   -   C   -   C   G   T   -   -   -   -
0   0   0   0   +2  -1  +2  +2  +2  0   0   0   0
```

Local Alignment = (+2) x 4 + (-1) x 1 = +7

局所的アラインメント

Global Sequence Alignment by DP

- Saul Needleman, Christian Wunsch : "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of Molecular Biology*, 48: 443-453 (1970). **(Needleman-Wunsch algorithm)**

Suitable when similarity can be found on **whole region** of sequences.

Local Sequence Alignment by DP

- Temple Smith, Michael Waterman: "Identification of Common Molecular Subsequences", *Journal of Molecular Biology*, 147: 195-197 (1981). **(Smith-Waterman algorithm)**

Suitable when similarity can be found only on **partial subregion(s)** of sequences.

Complexity of Local Alignment?

In local alignment, we should search the optimal solution considering combination of all possible sub-sequences from (Seq1, Seq2) ...

A naïve approach:

Seq1

A T T G C T C G T T G G A ... (length m)

generation partial seq. → $O(\mathbf{m}^2)$ **sub-sequences**

Seq2

A A A A C C G T A A... (length n)

generation partial seq. → $O(\mathbf{n}^2)$ **sub-sequences**

Pairwise Alignment by Dynamic Programming

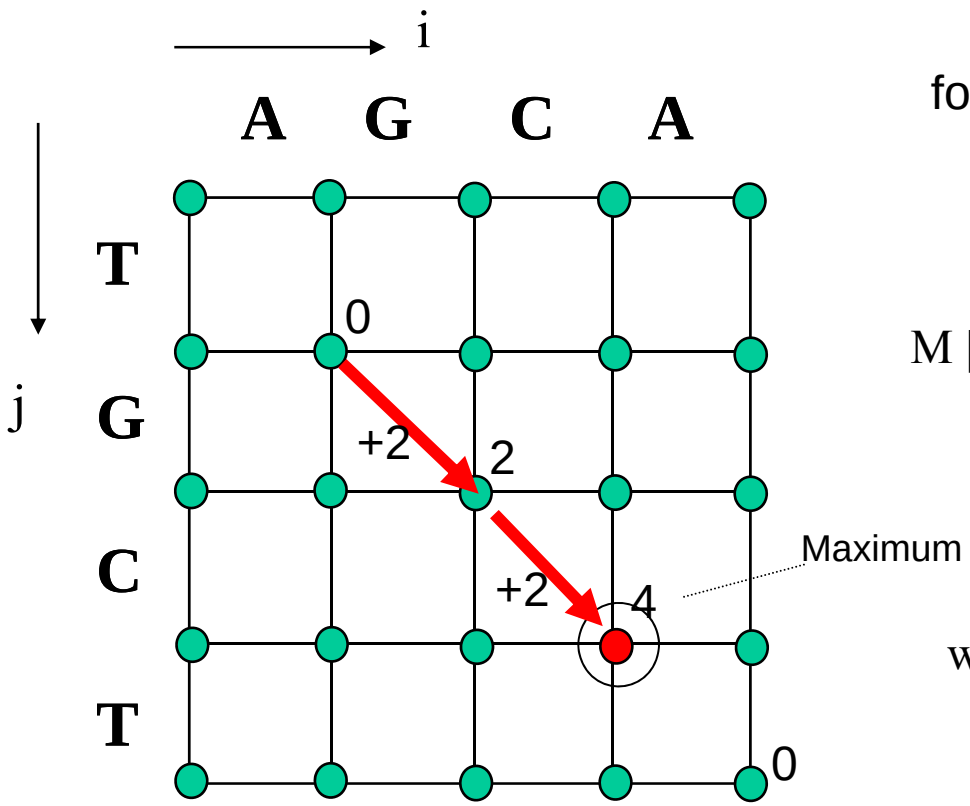
$O(\mathbf{m\ n})$

Choose optimal ?

Total: $O(\mathbf{m}^2) \times O(\mathbf{n}^2) \times O(\mathbf{m\ n})$ required ?!

Local alignment by DP

(Smith-Waterman algorithm)



for i>0, j>0,

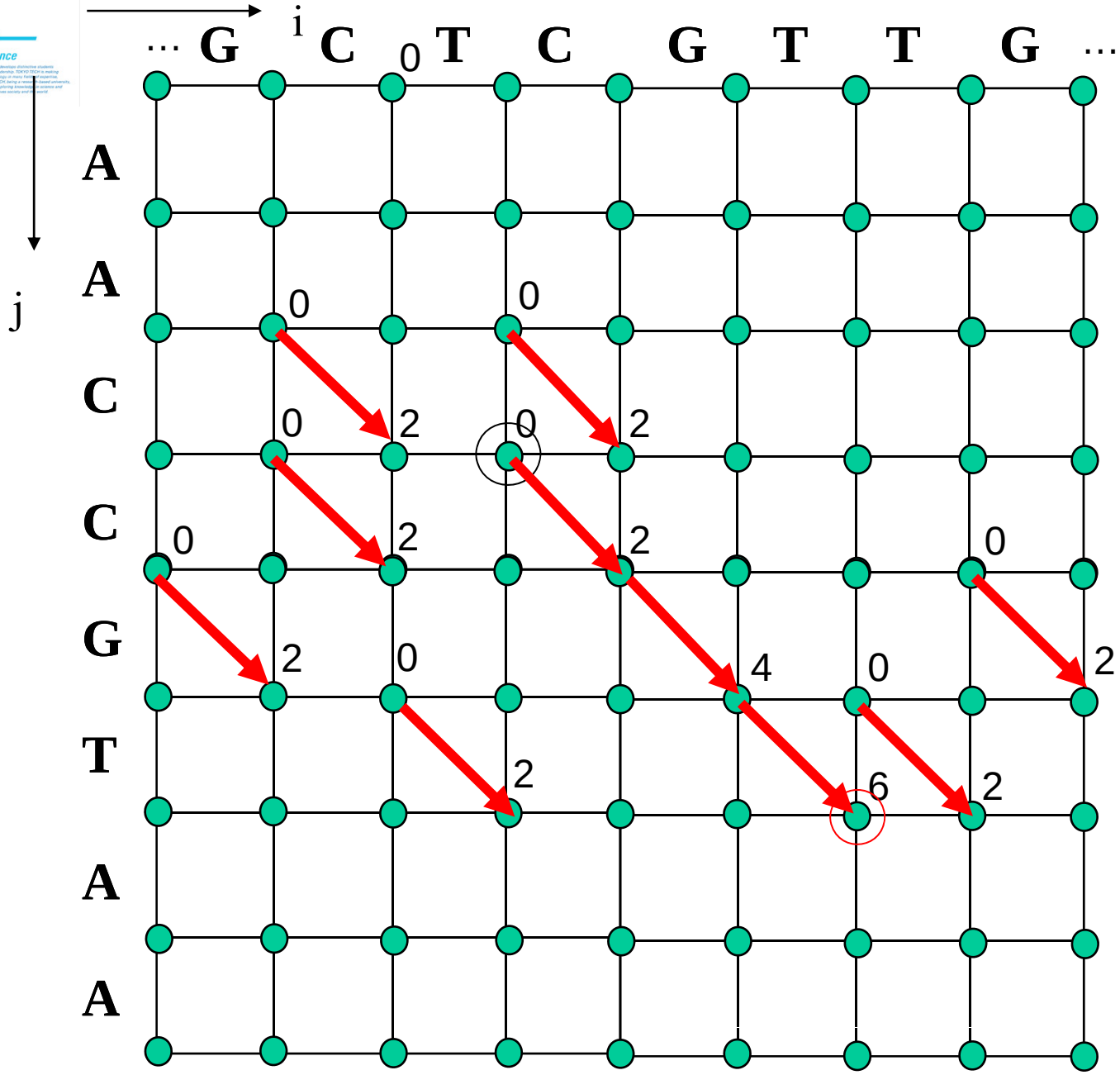
$$M[i,j] \leftarrow \max \begin{cases} M[i-1,j] + w \\ M[i,j-1] + w \\ M[i-1,j-1] + S[i,j] \\ \boxed{0 \quad \text{(discard \& reset)}} \end{cases}$$

where S[i, j] is match/unmatch score,
w is gap penalty constant (<= 0), and
M[i, j] is **accumulative score** until [i,j].

$M[i,0] \leftarrow 0$
$M[0,j] \leftarrow 0$

After the Matrix fill step ends, maximum value in M[x,y] should be searched all over the matrix (not simply taken from bottom right). (If maximum value is always recorded at the Matrix fill, then no need to search again.) The max. value node is the "tail". Trace back to a node with 0, which is the "head" of alignment.

Example (Local)

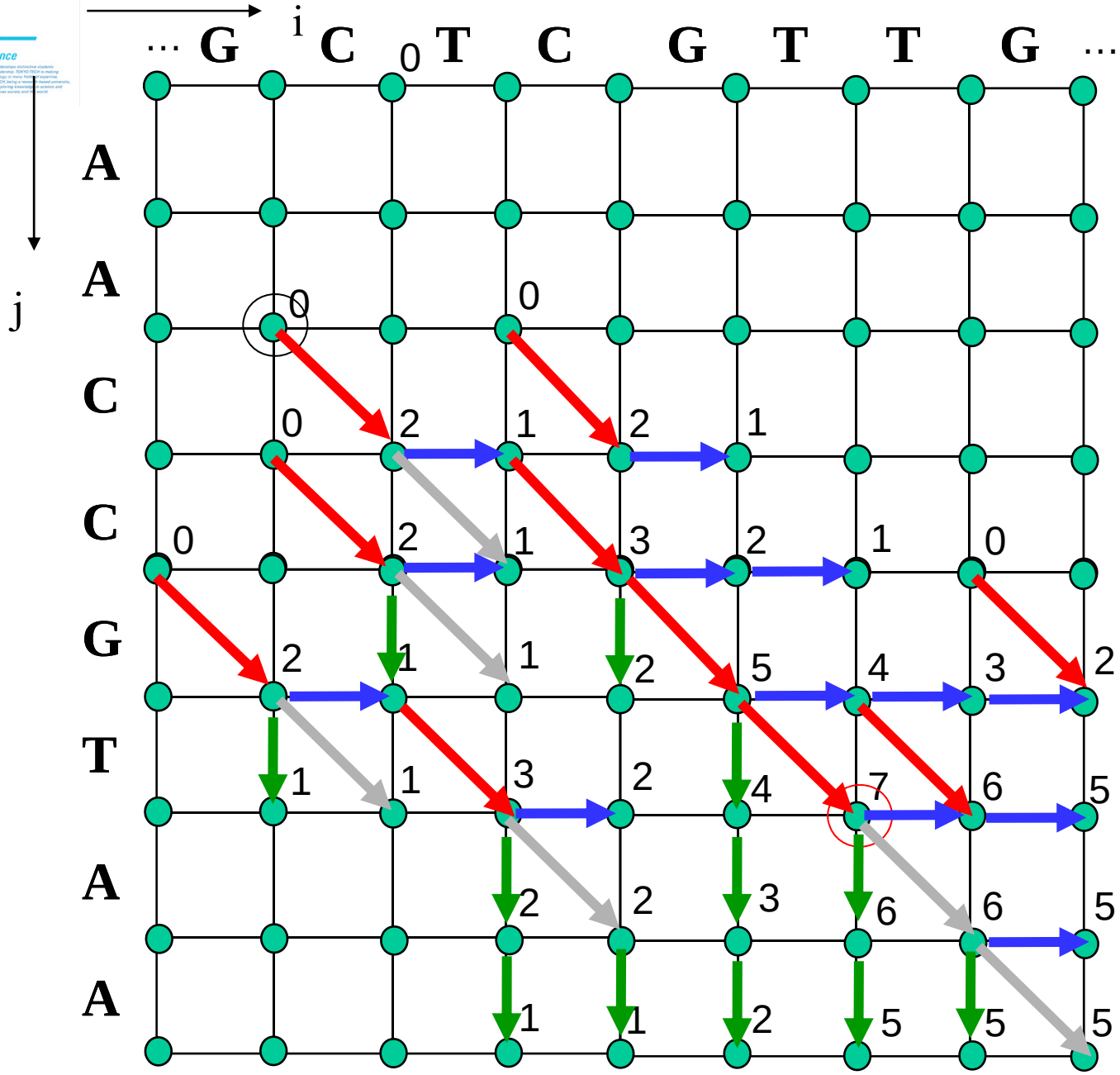



```
---CGT--
   |||
---CGT--
```

Max=6

Match = +2 (*for simplicity*, Mismatch = -99999, Gap = -99999)

Example (Local)



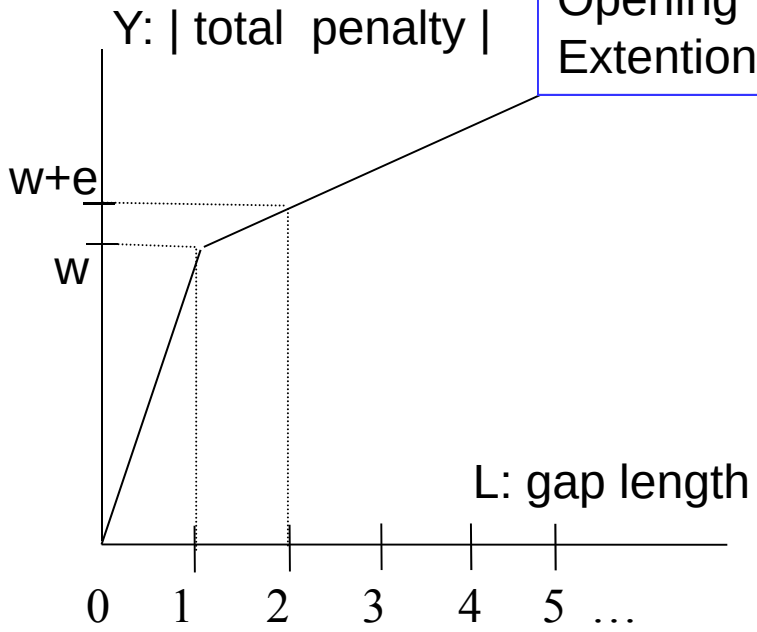

```
-CTCGT--
 | |||
-C-CGT--
```

Max=7

Match = +2, Mismatch = -1, Gap = -1 (Linear Gap)

Affine Gap

Opening gap (w) is heavy, while
Extention gap (e) is relatively lighter



additional matrices **Mx, My**
for recording consecutive gaps

for i>0, j>0,

$$M[i,j] \leftarrow \max \begin{cases} M_X[i,j] \\ M_Y[i,j] \\ M[i-1,j-1] + S[i,j] \end{cases}$$

$$M_X[i,j] \leftarrow \max \begin{cases} M[i-1,j] + w \\ M_X[i-1,j] + e \end{cases}$$

$$M_Y[i,j] \leftarrow \max \begin{cases} M[i,j-1] + w \\ M_Y[i,j-1] + e \end{cases}$$

<u>Affine gap:</u>

$Y = w + (L-1) \times e$

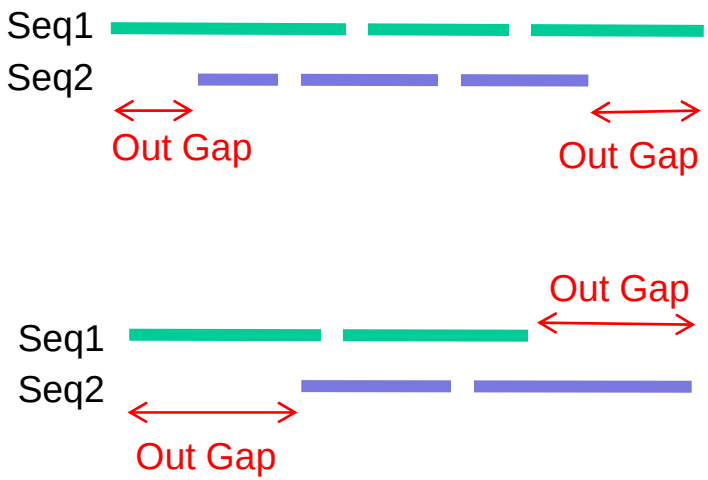
<u>Linear gap</u>:

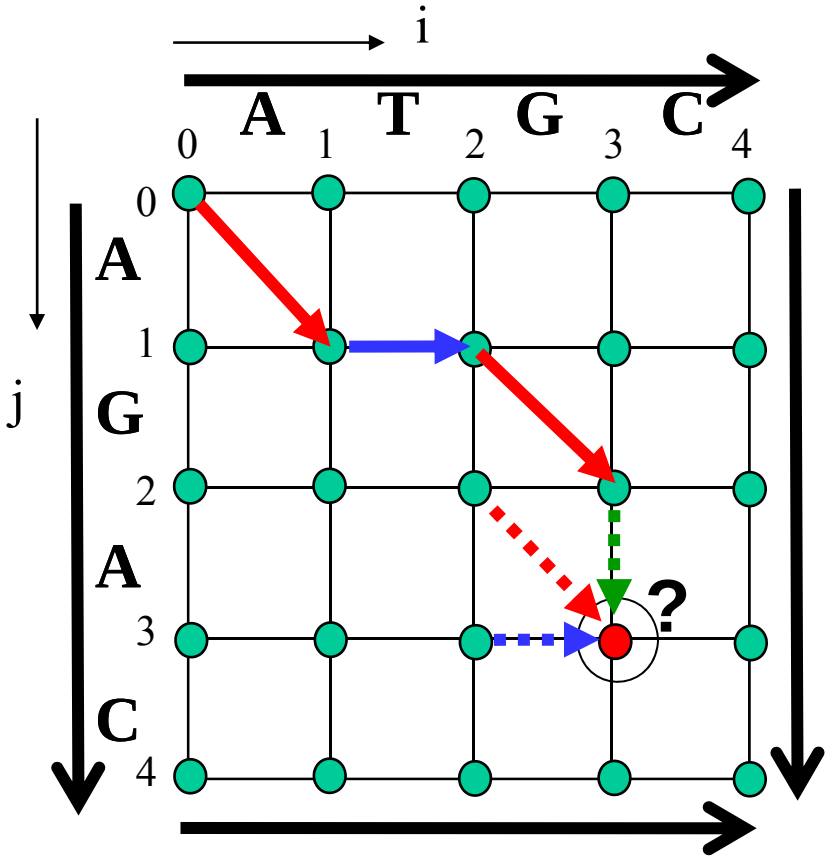
$Y = L \times w$

Seq1: **A T G G C …**
Seq2: **A – – – C …**

<u>Affine</u>: $Y = w + 2e$
<u>Linear</u>: $Y = 3w$

Out Gap





Outer trim area is called as an “Out Gap”.

Out Gap penalty is sometimes set to zero.
It means that any movement
on four edges (black arrow in the fig.) can be done
without penalty.

Summary of #2

- Sequence alignment
- Alignment score
- Sequence alignment by DP (Dynamic Programming)
  - Initialization step, Matrix fill step, Trace back step
- Global alignment versus Local alignment
- Local alignment by DP
- Gap penalty: Linear vs. Affine gap

Coming next:
#3 Multiple alignment