Dynamic programming for k sequences

$V(i_1, i_2)$ = optimal alignment between $s_1[1..i_1]$ and $s_2[1..i_2]$

$$V(i_1, i_2) = \text{Max} \begin{cases} V(i_1-1, i_2-1) + \delta(s[i_1], s[i_2]) \\ V(i_1-1, i_2) + \delta(s[i_1], \_) \\ V(i_1, i_2-1) + \delta(\_, s[i_2]) \end{cases}$$

⇓ Align two sequences

$$V(i_1, i_2) = \text{Max}_{b_1, b_2 \in \{0,1\}^2 - \{(0,0)\}} \{ V(i_1-b_1, i_2-b_2) + \delta(s_1[i_1 b_1], s_2[i_2 b_2]$$

Assume $s_j[0] = \_$

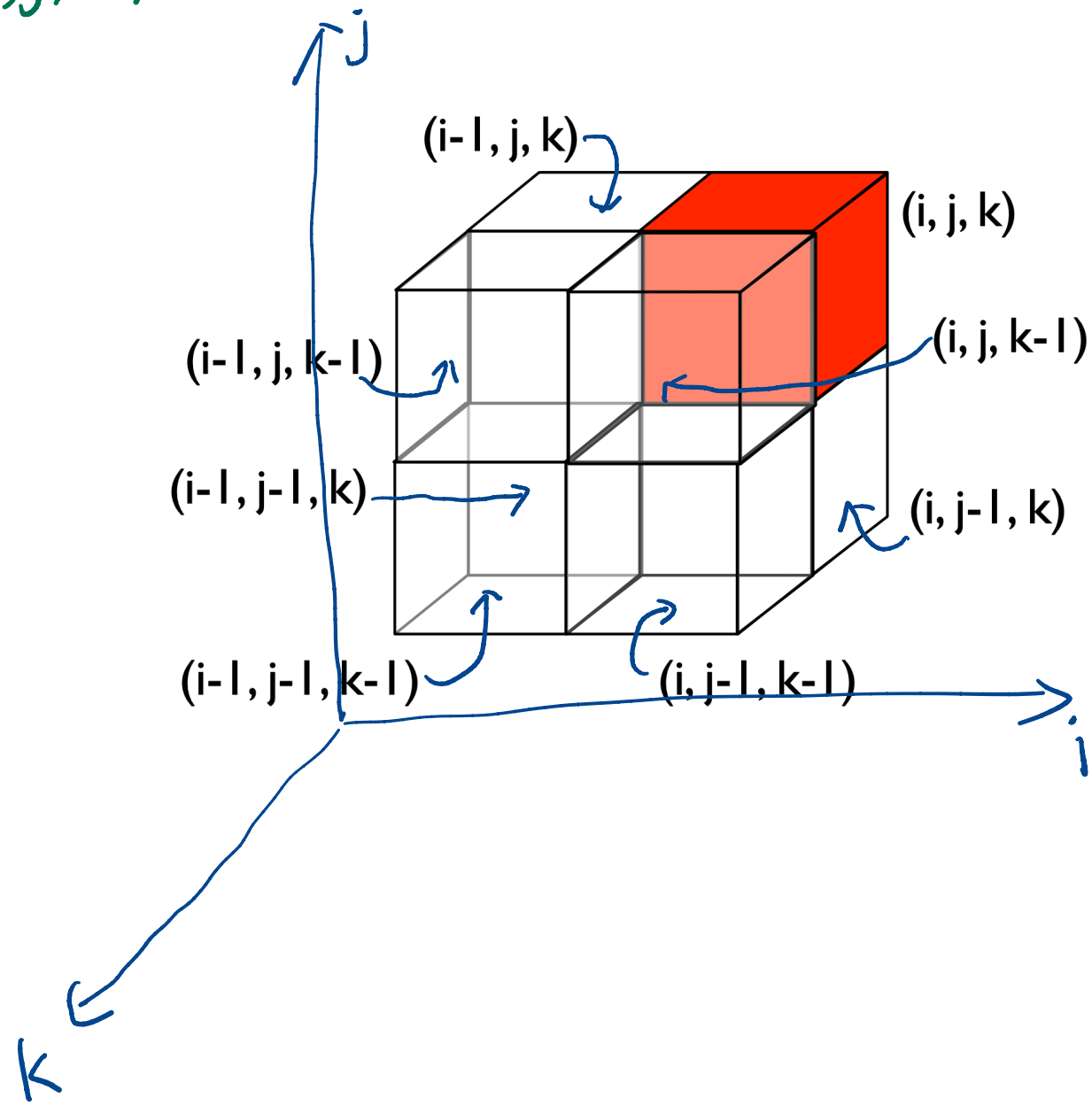
⇓ Align three sequences

$$V(i_1, i_2, i_3) = \text{Max}_{b_1, b_2, b_3 \in \{0,1\}^3 - \{(0,0)\}} \{ V(i_1-b_1, i_2-b_2, i_3-b_3) + \delta(s_1[i_1 b_1], s_2[i_2 b_2], s_3[i_3 b_3]) \}$$

⇓ Align k sequences

$$V(i_1, \ldots, i_k) = \text{Max}_{b_1 \ldots b_k \in \{0,1\}^k - \{(0,0)\}} \{ V(i_1-b_1, \ldots, i_k-b_k) + \delta(s_1[i_1 b_1], \ldots s_k[i_k b_k]) \}$$

For three sequences, there are 7 path to V(i,j,k)



when # sequences > 3, we cannot directly visualize the path

Example: Align 3 sequences

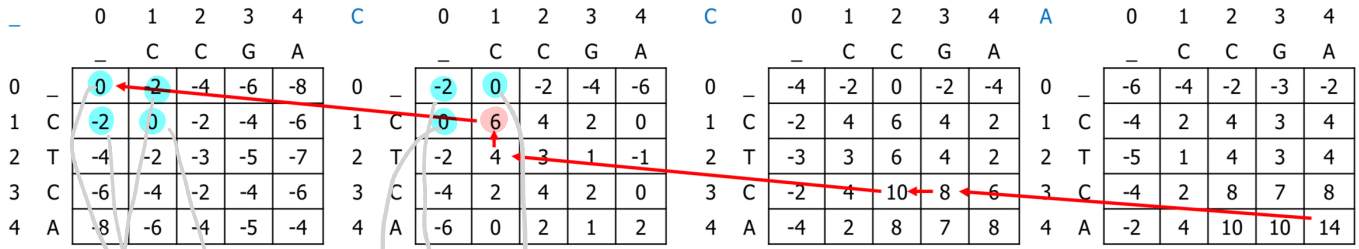
$S_1 = CTCA$

$S_2 = CCGA$

$S_3 = CCA$

Score matrix

	_	A	C	G	T
_	0	-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2



_		0	1	2	3	4
		_	C	C	G	A
0	_	0	-2	-4	-6	-8
1	C	-2	0	-2	-4	-6
2	T	-4	-2	-3	-5	-7
3	C	-6	-4	-2	-4	-6
4	A	-8	-6	-4	-5	-4

C		0	1	2	3	4
		_	C	C	G	A
0	_	-2	0	-2	-4	-6
1	C	0	6	4	2	0
2	T	-2	4	3	1	-1
3	C	-4	2	4	2	0
4	A	-6	0	2	1	2

C		0	1	2	3	4
		_	C	C	G	A
0	_	-4	-2	0	-2	-4
1	C	-2	4	6	4	2
2	T	-3	3	6	4	2
3	C	-2	4	10	8	6
4	A	-4	2	8	7	8

A		0	1	2	3	4
		_	C	C	G	A
0	_	-6	-4	-2	-3	-2
1	C	-4	2	4	3	4
2	T	-5	1	4	3	4
3	C	-4	2	8	7	8
4	A	-2	4	10	10	14

$$
\begin{aligned}
6 = \max \{ & \\
& -2 + \delta(\_\,\_\,C) \\
& 0 + \delta(\_\,C\,C) \\
& 0 + \delta(C\,\_\,C) \\
& 0 + \delta(\_\,\_\,C) \\
& -2 + \delta(C\,\_\,C) \\
& -2 + \delta(\_\,C\,C) \\
& 0 + \delta(C\,C\,C)
\end{aligned}
$$

Complexity

Time:

Entries: $n_1 n_2 \cdots n_k$ → k sequences

Each entry: $2^k k^2$ → compute the cost

→ compute the max

Total running time:

$$O(2^k k^2 n_1 n_2 \cdots n_k)$$

Space:

$$O(n_1 n_2 \cdots n_k)$$