CSE 549

Lecturer: Sael Lee

AMINO ACID SEQUENCE ALIGNMENT

Slides provided by courtesy of Dr. D. Kihara @ Purdue

KEY ISSUES IN SEQUENCE ALIGNMENT

- What sort of alignment should be considered?
- What scoring system should be used to rank alignment ?
- What algorithm should be used to find optimal (or good) scoring alignments ?
- What statistical methods should be used to evaluate the significance of an alignment score?

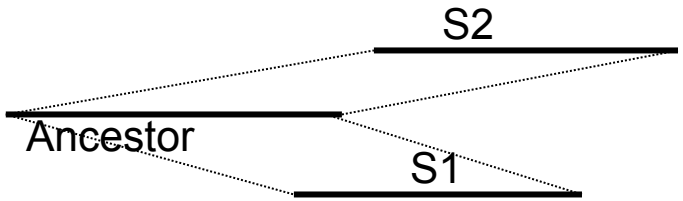
ALIGNING SEQUENCES TO FIND HOMOLOGY

- Homologous sequences
  - Related by evolution (common ancestors)

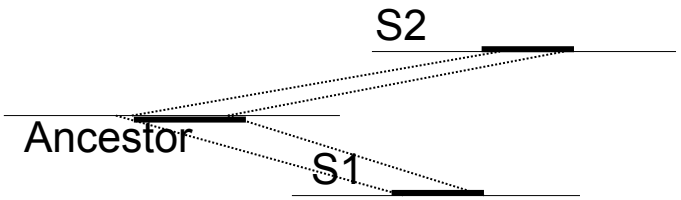
- Alignment of homologous sequences
  - Identifying relationship between the sequence elements
  - Match up characters coming from same characters in ancestor

TYPES OF ALGINMENT

- Global Alignment
  - Assuming that the *complete* sequences are the results of evolution from the same ancestor sequence



- Local Alignment
  - Align segments of the sequences so that the segments are evolutionarily related



ALIGNMENT SCORE

SCORING (1)

- Match – mismatch
  - Match : +1, mismatch: 0
  - Identity matrix (often used for DNA sequences)

DNA

	a	c	g	t
a	1	0	0	0
c	0	1	0	0
g	0	0	1	0
t	0	0	0	1

Amino acid

	A	R	N	D	…
A	1	0	0	0	0
R	0	1	0	0	0
N	0	0	1	0	0
D	0	0	0	1	0
…	0	0	0	0	..

ALIGNMENT SCORE

- Add up the terms (assume independence).
- DNA

```
atgatcaagtactttaagaagcagaagcggc
|||||  |||  ||||||||||  ||  |||  |||
atgataaagcactttaagaaacaaaagaggc
```

  - 26 matches / 31 nt (= 83.9%) **(identity)**
- Protein

```
SSWRVISSIEQKTER
.:::.:::::::.:
ASWRILSSIEQKEEA
```

  - 10 matches / 15 aa (= 66.7%) **(identity)**

SCORING (2)

- Amino acid **substitution (similarity) matrix**
  - Counting similarity of amino acids
  - Analyze statistics of known alignments
  - PAM, BLOSUM series, matrices specific for a certain type of proteins, e.g. membrane proteins

	A	R	N	D	…
A	5	-2	-1	-2	
R	-2	7	0	-1	
N	-1	0	6	2	
D	-2	-1	2	7	
…					..

(BLOSUM45)

SCORING MATRICES FOR PROTEIN SEQUENCE ALIGNMENT

- Define scores for amino acid pairs in sequence alignments
- Reflect “similarity” of amino acid residues
- Most often a substitution matrix is used.
- *Amino acid substitution matrix* is not necessarily symmetric,
  - Reflecting the difference of the mutation probability of A > B from B > A (A, B: two different amino acids)
  - Correspond to the logarithm of the relative likelihood that the sequences are related, compared to being unrelated.

BLOSUM45 MATRIX

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-2	-2	0
R	-2	7	0	-1	-3	1	0	-2	0	-3	-2	3	-1	-2	-2	-1	-1	-2	-1	-2
N	-1	0	6	2	-2	0	0	0	1	-2	-3	0	-2	-2	-2	1	0	-4	-2	-3
D	-2	-1	2	7	-3	0	2	-1	0	-4	-3	0	-3	-4	-1	0	-1	-4	-2	-3
C	-1	-3	-2	-3	12	-3	-3	-3	-3	-3	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1
Q	-1	1	0	0	-3	6	2	-2	1	-2	-2	1	0	-4	-1	0	-1	-2	-1	-3
E	-1	0	0	2	-3	2	6	-2	0	-3	-2	1	-2	-3	0	0	-1	-3	-2	-3
G	0	-2	0	-1	-3	-2	-2	7	-2	-4	-3	-2	-2	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	0	-3	1	0	-2	10	-3	-2	-1	0	-2	-2	-1	-2	-3	2	-3
I	-1	-3	-2	-4	-3	-2	-3	-4	-3	5	2	-3	2	0	-2	-2	-1	-2	0	3
L	-1	-2	-3	-3	-2	-2	-2	-3	-2	2	5	-3	2	1	-3	-3	-1	-2	0	1
K	-1	3	0	0	-3	1	1	-2	-1	-3	-3	5	-1	-3	-1	-1	-1	-2	-1	-2
M	-1	-1	-2	-3	-2	0	-2	-2	0	2	2	-1	6	0	-2	-2	-1	-2	0	1
F	-2	-2	-2	-4	-2	-4	-3	-3	-2	0	1	-3	0	8	-3	-2	-1	1	3	0
P	-1	-2	-2	-1	-4	-1	0	-2	-2	-2	-3	-1	-2	-3	9	-1	-1	-3	-3	-3
S	1	-1	1	0	-1	0	0	0	-1	-2	-3	-1	-2	-2	-1	4	2	-4	-2	-1
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-1	-1	2	5	-3	-1	0
W	-2	-2	-4	-4	-5	-2	-3	-2	-3	-2	-2	-2	-2	1	-3	-4	-3	15	3	-3
Y	-2	-1	-2	-2	-3	-1	-2	-3	2	0	0	-1	0	3	-3	-2	-1	3	8	-1
V	0	-2	-3	-3	-1	-3	-3	-3	-3	3	1	-2	1	0	-3	-1	0	-3	-1	5

ALIGNMENT SCORE: SMITH-WATERMAN SCORE

- BLOSUM45, Gap penalty: -12/-2
- Add up each term.
- Sequence identity: 15/29 = 51.7%
- Smith-Waterman Score: 63

```
SSWRVISSIEQKTER--NEKKQQ-GKEYR
.:::.::::::::: :   :.  .  .   ::::
ASWRILSSIEQKEEAKGNDVSVKRIKEYR
```

1+4+15+7+3+2+4+4+5+6+…. +6-2-12-2+6… … -12…

GAPLESS ALIGNMENT

- Gaps not allowed in the middle
  - Scan one sequence along the other one
- Number of possible alignments
  - Sequence length: m, n

  - m + n + 1 , If m=n, 2n+1; i.e O(n)
- Application: finding a known motif in a sequence
- How to choose the “best” alignment?
  - Scoring scheme

ALIGNMENT WITH GAPS

- Scoring: AA matrix + gap penalty
- Gap penalty for a gap of length g:
  - Linear model: -gd (d : gap penalty, d>0)
  - Affine model: -d – (g-1)e

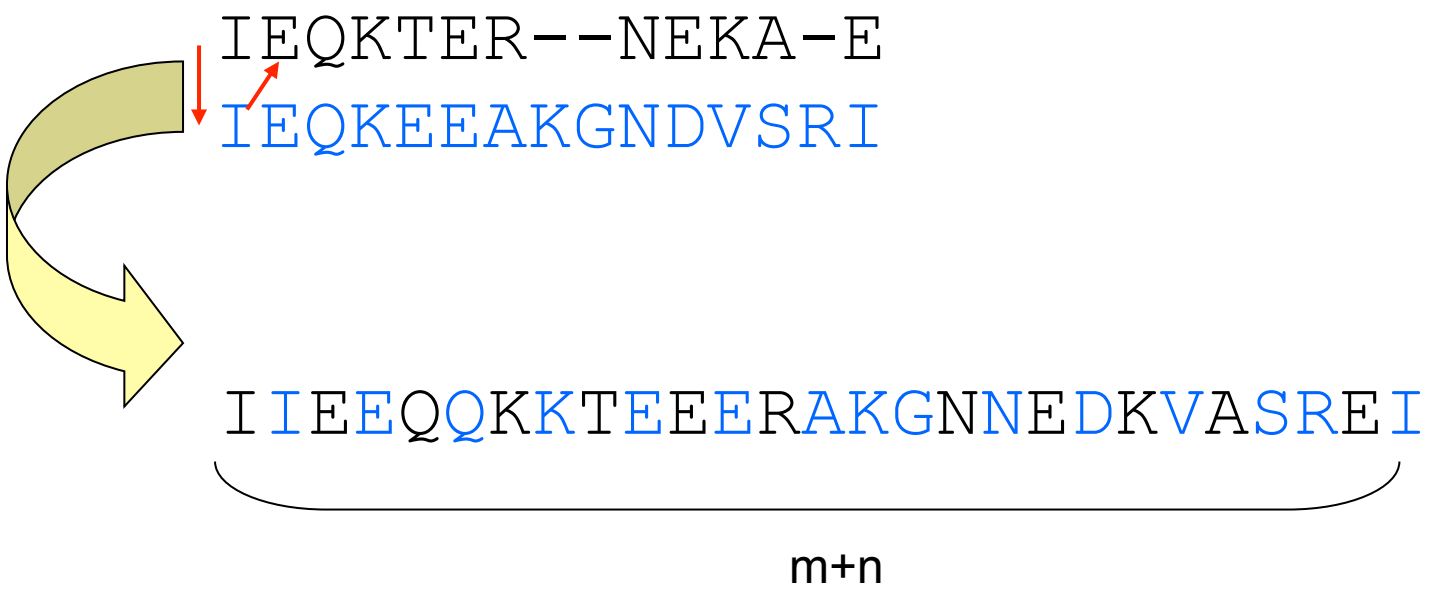
    (d: opening penalty, e: extension penalty. d > e > 0)
- Number of possible alignments

$\binom{m+n}{m}$ If m=n, $\binom{m+n}{m} = \binom{2n}{n} = \frac{(2n)!}{(n!)^2} \cong \frac{2^{2n}}{\sqrt{2\pi n}}$ i.e. $O(4^n)$

- Algorithmic challenge: Given AA matrix, gap penalty, find th e alignment with the best score.

COMPLEXITY OF AN ALIGNMENT WITH GAPS

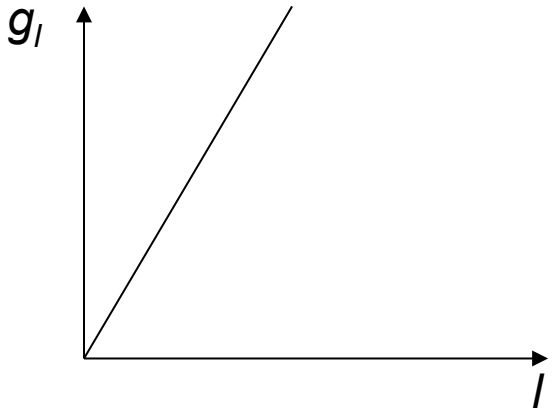
- Why $\binom{m+n}{n}$?



LINEAR AND AFFINE GAP PENALTIES

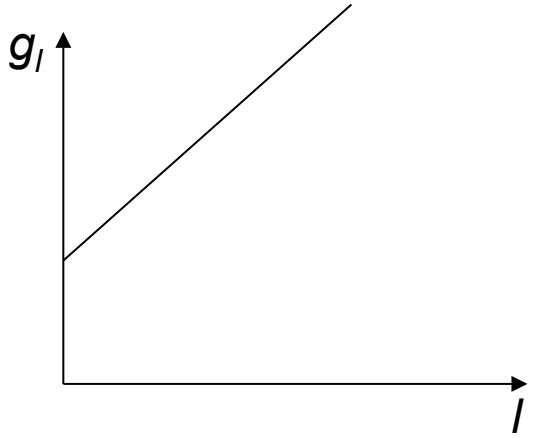
- Linear:

$$g_l = g \cdot l$$



- Affine:

$$g_l = g_{open} + (l-1) \cdot g_{extend}$$



GLOBAL ALIGNMENT

FINDING THE HIGHEST SCORING ALIGNMENT

- Problem:
  - Given two sequences, a scoring matrix, and a gap penalty, find the alignment with the highest score
- Large number of possible alignments
  - Cannot generate all and score them to find the best
  - Algorithm: **dynamic programming (DP) algorithm (Needleman-Wunsch Algorithm)**

INDEPENDENCE BETWEEN SUB-ALIGNMENTS

- Observations:
  - The score of the alignment up to and including character *i* from *q* and character *j* from *d* is independent of how the rest of the sequences are aligned
  - The best solution to *(i,j)* can be "locked", its score recorded in $H_{i,j}$
    - *Where* $H_{m,n}$ is the score of the best global alignment
  - Amenable to Dynamic Programming

RECURRENCE RELATION IN DP

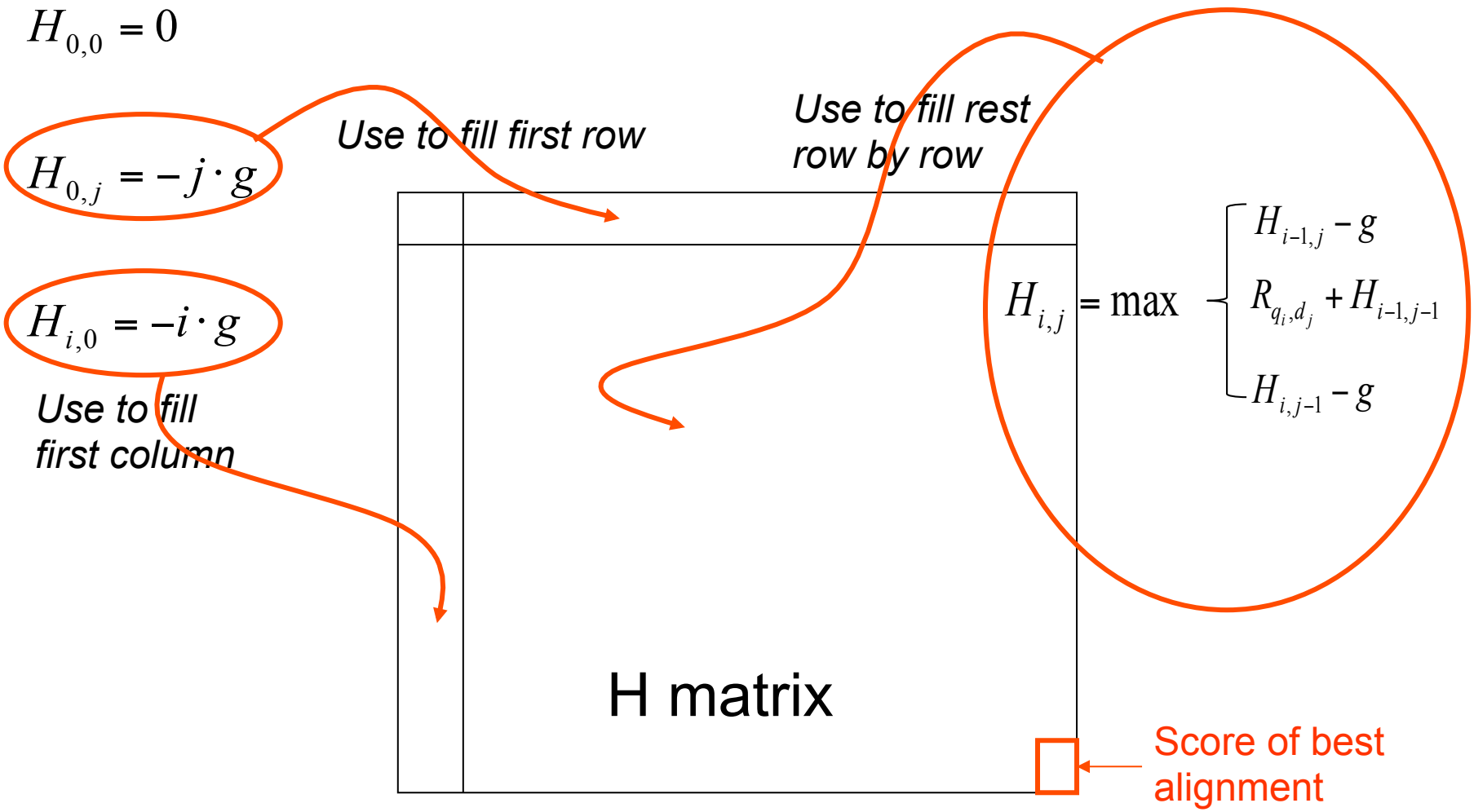
- Assume that, $H_{i-1,j-1}$, $H_{i-1,j}$, $H_{i,j-1}$ are known

$H_{i-1,j}$	$q_{1..i-1}$ $d_{1..j}$	x_i -
$H_{i-1,j-1}$	$q_{1..i-1}$ $d_{1..j-1}$	x_i y_j
$H_{i,j-1}$	$q_{1..i}$ $d_{1..j-1}$	- y_j

$$H_{i,j} = \max \begin{cases} H_{i-1,j} - g \\ S(x_i, y_j) + H_{i-1,j-1} \\ H_{i,j-1} - g \end{cases}$$

Where g is the gap open penalty and $S(x_i, y_j)$ is the similarity score obtained from substitution matrix for residue type of x_i and y_j

CALCULATING SCORE OF BEST ALIGNMENT USING MATRIX



GLOBAL DP MATRIX, H(I,J)

$$H(i,j) = \max \begin{cases} H(i-1, j-1) + s(x_i, y_j) \\ H(i-1, j) - d \\ H(i, j-1) - d \end{cases}$$

BLOSUM45

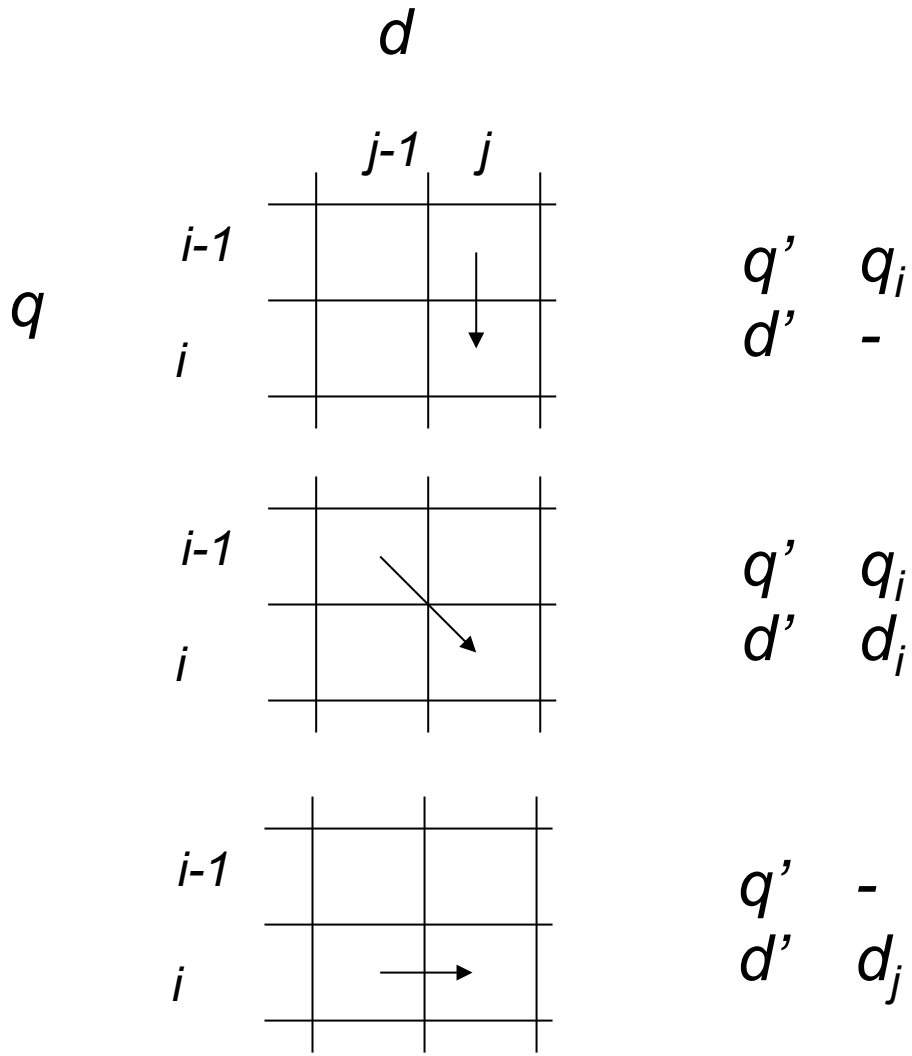
i

j		L	I	E	Y	G	D	A
	0	-8	-16	-24	-32	-40	-48	-56
V	-8	1	-24 -5 -7	…				
E	-16							
W	-24							
F	-32							
L	-40							

```
LI   LI-   LI
-V   --V   V-
  or    or
```

BLOSUM45
S(V,L) = 1
S(V,I) = 3
Gap = -8

TRACEBACK, ALIGNMENT



GLOBAL DP MATRIX, H(I,J)

Fill this table from top-left to bottom-right
Trace back to get the alignment!

		L	I	E	Y	G	D	A
	0	-8	-16	-24	-32	-40	-48	-56
V	-8	1	-5	-13	-21	-29	-37	-45
E	-16	-7	-2	1	-7	-15	-23	-31
W	-24	-15	-9	-5	4	-4	-12	-20
F	-32	-23	-15	-12	-2	1	-7	-14
L	-40	-27	-21	-17	-10	-5	-2	-8

LIEYGDA
-VEWF-L

TIME COMPLEXITY

- Sequences of lengths *n* and *m*

$$O(nm)$$

- Two sequences of length *l*

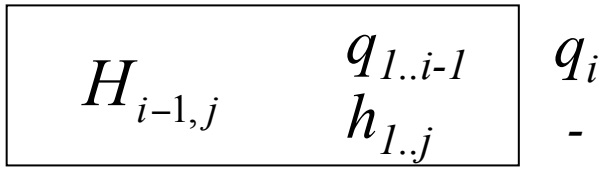
$$O(l^2)$$

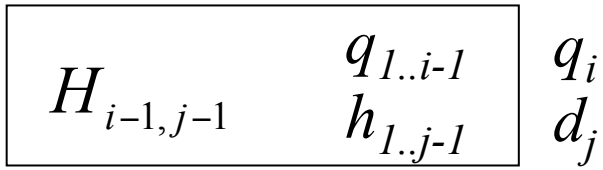
LOCAL ALGINMENT

THE LOCAL ALIGNMENT

- Aims to identify only very similar region of two protein sequences
- Should ignore negatively contributing suffixes of align ments
- Score of best local alignment – highest value in dyna mic programming matrix
- Alignment found by tracing back from maximum value until cell with value 0 (zero) has been reached

DP RECURRENCE RELATION



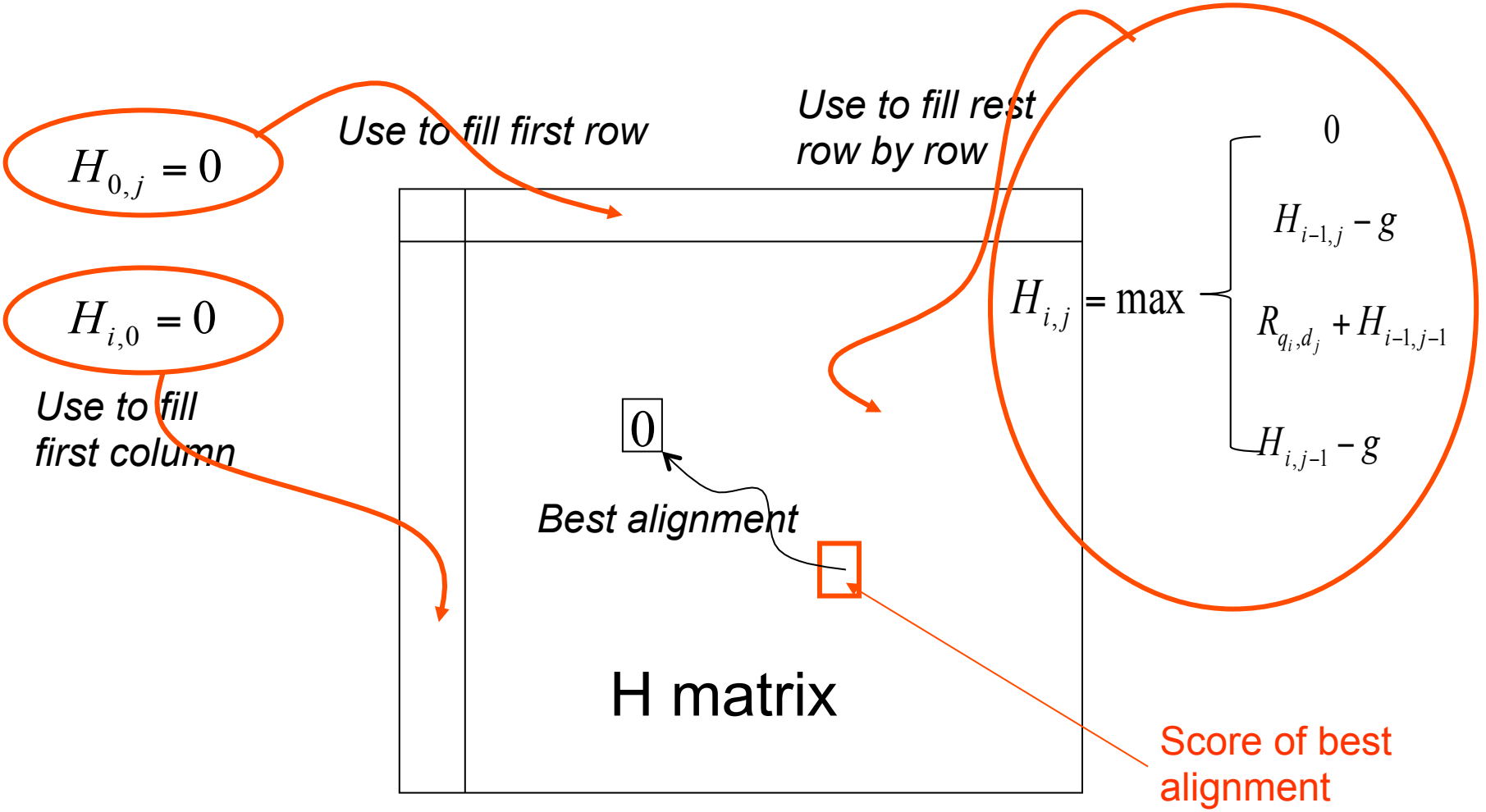


Empty alignment

$$H_{i,j} = \max \begin{cases} H_{i-1,j} - g \\ R_{q_i,d_j} + H_{i-1,j-1} \\ H_{i,j-1} - g \\ 0 \end{cases}$$

Effectively allows for removal of negatively contributing prefixes.

CALCULATING BEST LOCAL ALIGNMENT



EXAMPLE OF LOCAL DP MATRIX, H(I,J)

$$H(i,j) = \max \begin{cases} 0 \\ H(i-1, j-1) + s(x_i, y_j) \\ H(i-1, j) - d \\ H(i, j-1) - d \end{cases}$$

		L	I	E	Y	G	D	A
	0	0	0	0	0	0	0	0
V	0	1	3	0	0	0	0	0
E	0	0	0	9	1	0	2	0
W	0	0	0	1	12	4	0	0
F	0	1	0	0	4	9	1	0
L	0	5	3	0	0	1	6	0

BLOSUM45
Gap penalty = -8

IEY
VEW

TIME COMPLEXITY OF LOCAL ALIGNMENT

- Sequences of lengths *n* and *m*

$$O(nm)$$

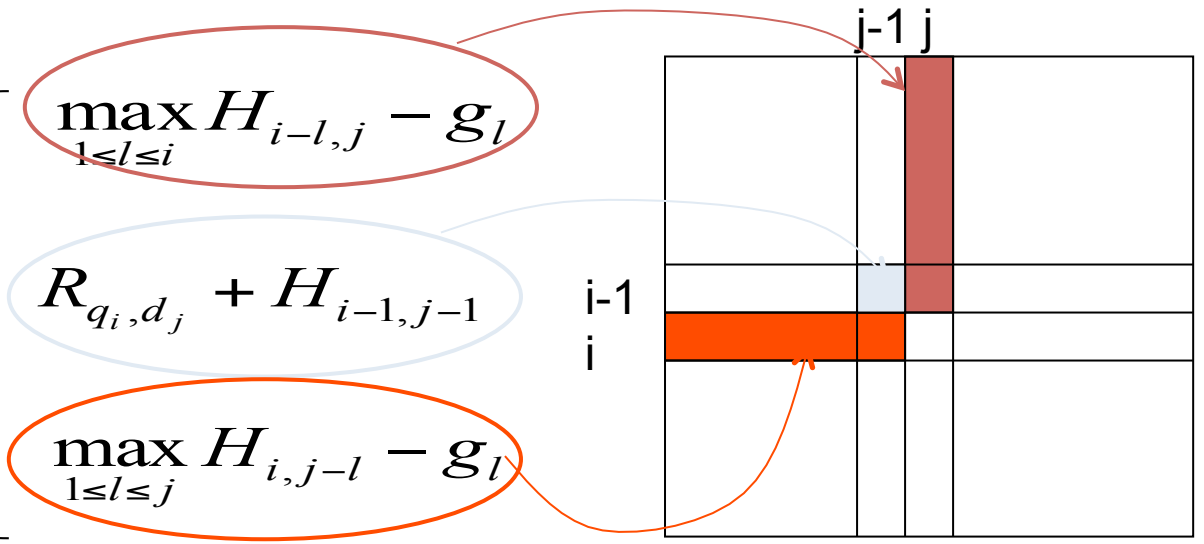
- Two sequences of length *l*

$$O(l^2)$$

DYNAMIC PROGRAMMING FOR GENERAL GAP PENALTY

- The recursion we have seen only works for linear gap penalties…
- For general penalty, we need to ”look further back”

$$H_{i,j} = \max \begin{cases} \max_{1 \le l \le i} H_{i-l,j} - g_l \\ R_{q_i,d_j} + H_{i-1,j-1} \\ \max_{1 \le l \le j} H_{i,j-l} - g_l \end{cases}$$



TIME COMPLEXITY OF ALIGNMENT WITH GAPS

- Sequences of lengths *n* and *m*

$$O(nm^2 + mn^2)$$

- Two sequences of length *l*

$$O(l^3)$$

CAN WE DO BETTER?

- Yes (Affine Gap Algorithm 2).
- Using three matrices instead of one:
  - Matrix M, best score when S_i aligns with T_j.
  - Matrix I, best score when S_i aligns with a gap.
  - Matrix J, best score when a gap aligns with T_j.

AFFINE GAP ALGORITHM 2

M:

S/T		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	-12							
E	-14							
W	-16							
F	-18							
L	-20							

Gap:
Opening: -12
Extension: -2

$$M(i, j) = \max \begin{cases} M(i-1, j-1) + s(S_i, T_j) \\ I(i-1, j-1) + s(S_i, T_j) \\ J(i-1, j-1) + s(S_i, T_j) \end{cases}$$

S_i align with T_j
Si align with gap
gap align with T_j

AFFINE GAP ALGORITHM 2

Tj:

I:

Si:

S/T		L	I	E	Y	G	D	A
	0	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)
V	-12							
E	-14							
W	-16							
F	-18							
L	-20							

$$I(i, j) = \max \begin{cases} M(i-1, j) - d & S_i \text{ align with initial gap} \\ I(i-1, j) - e & S_i \text{ align with extension gap} \end{cases}$$

AFFINE GAP ALGORITHM 2

Tj:

J:

Si:

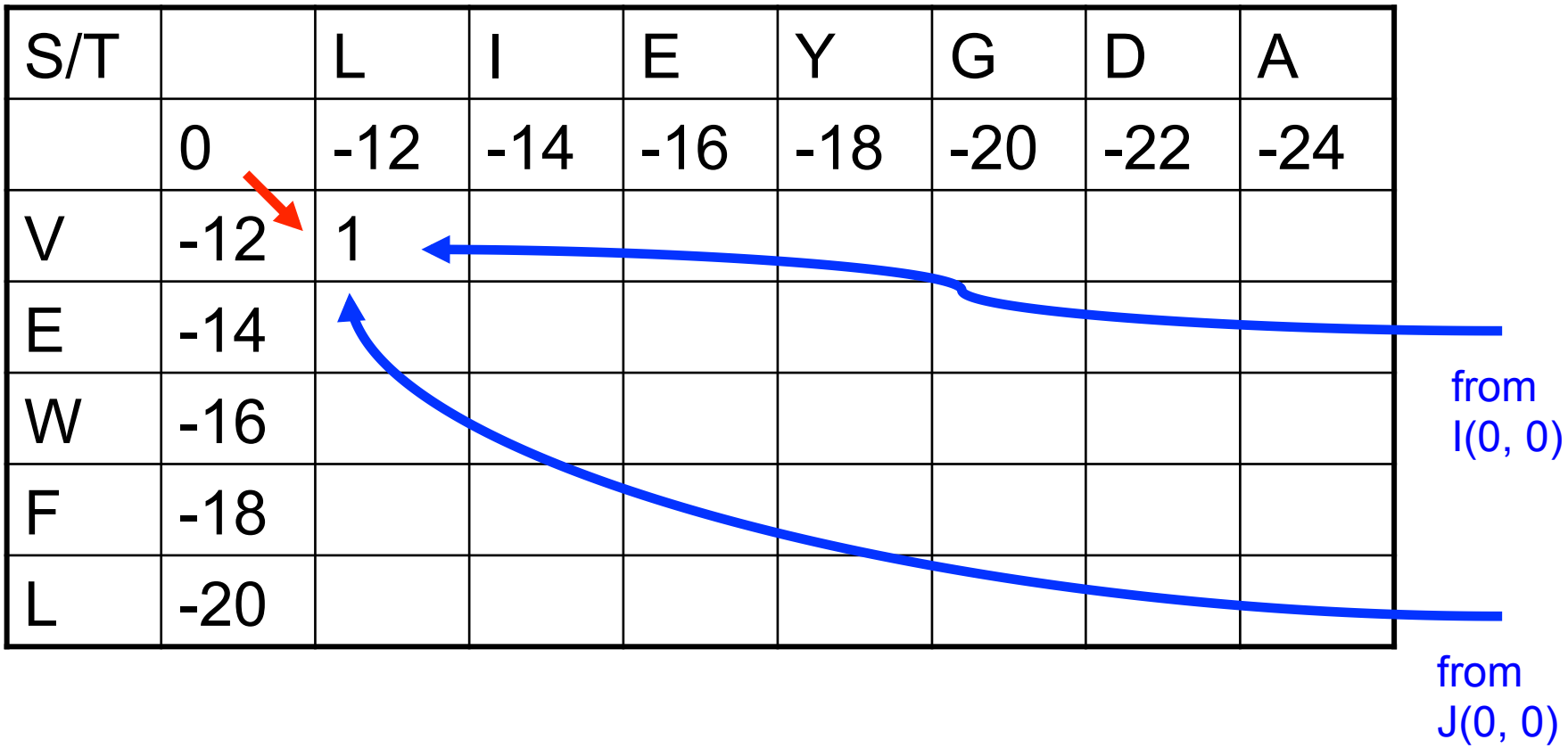
S/T		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	(-∞)							
E	(-∞)							
W	(-∞)							
F	(-∞)							
L	(-∞)							

$$J(i, j) = \max \begin{cases} M(i, j-1) - d & \text{initial gap align with } T_j \\ J(i, j-1) - e & \text{extension gap align with } T_j \end{cases}$$

AFFINE GAP ALGORITHM 2

M:

S/T		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	-12	1						
E	-14							
W	-16							
F	-18							
L	-20							



$$M(1, 1) = \max \begin{cases} 0 + 1 = 1 \\ 0 + 1 = 1 \\ 0 + 1 = 1 \end{cases}$$

AFFINE GAP ALGORITHM 2

I:

S_i/T_j		L	I	E	Y	G	D	A
	0	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)	(-∞)
V	-12	-24	-26	-28	-30	-32	-34	-36
E	-14	-11						
W	-16							
F	-18							
L	-20							

$$I(1, 1) = \max \begin{cases} 1 - 12 = -11 \\ -24 - 2 = -26 \end{cases}$$

from M(1, 1)

AFFINE GAP ALGORITHM 2

J:

S_i/T_j		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	(-∞)	-24	-11					
E	(-∞)	-26						
W	(-∞)	-28						
F	(-∞)	-30						
L	(-∞)	-32						

$$J(1, 1) = \max \begin{cases} 1 - 12 = -11 \\ -24 - 2 = -26 \end{cases}$$

from M(1, 1)

BLOSUM62 SCORE MATRIX

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

AFFINE GAP ALGORITHM 2

M:

		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	-12	1	-9	-16	-17	-21	-23	-22
E	-14	-15	-2	-4	-15	-17	-15	-20
W	-16	-16	-14	-5	-2	-17	-21	-18
F	-18	-16	-13	-16	-2	-5	-17	-18
L	-20	-14	-13	-16	-17	-6	-9	-17

I:

		L	I	E	Y	G	D	A
	0	(-)	(-)	(-)	(-)	(-)	(-)	(-)
V	-12	-24	-26	-28	-30	-32	-34	-36
E	-14	-11	-21	-30	-32	-33	-35	-34
W	-16	-13	-13	-16	-27	-29	-27	-32
F	-18	-15	-15	-17	-14	-29	-29	-30
L	-20	-17	-17	-19	-16	-17	-29	-30

J:

		L	I	E	Y	G	D	A
	0	-12	-14	-16	-18	-20	-22	-24
V	(-)	-24	-11	-13	-15	-17	-19	-21
E	(-)	-26	-17	-14	-16	-18	-20	-22
W	(-)	-28	-28	-26	-17	-14	-16	-18
F	(-)	-30	-28	-25	-27	-14	-16	-18
L	(-)	-32	-26	-25	-27	-29	-18	-20

V E W F - - L
L I E Y G D A

TIME COMPLEXITY

- Sequences of lengths *n* and *m*

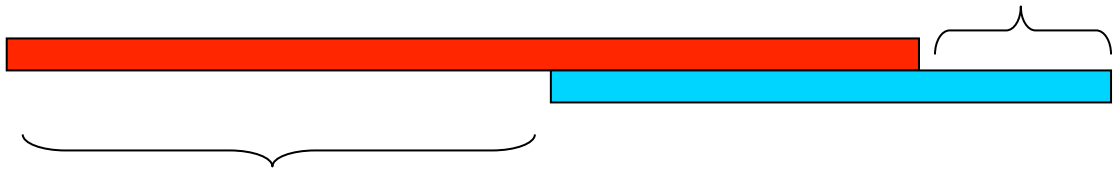
$$O(nm)$$

- Two sequences of length *l*

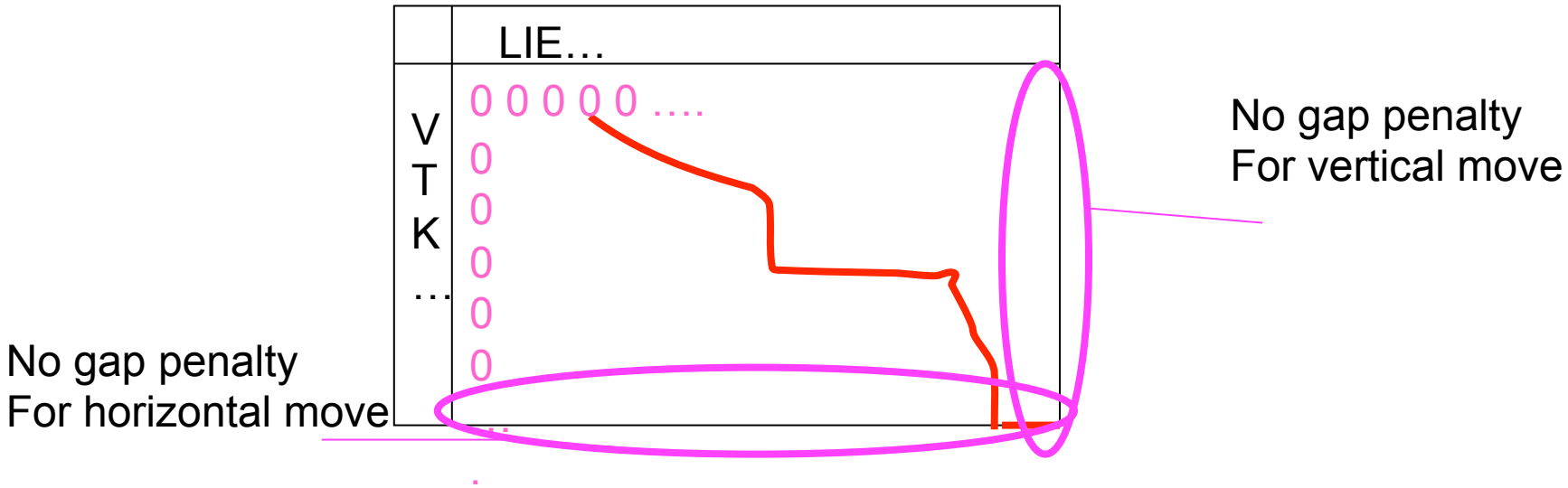
$$O(l^2)$$

ALIGNING SEQUENCES OF DIFFERENT LENGTH

- No penalty for “starting” and “ending” overhanging gaps



No gap penalty



REPEATED MATCHES

- Finding one or more non-overlapping copies of sections of one sequence in the other (motif, domains) (asymmetric)
- Only local matches scoring higher than a threshold T is considered
- Ex: Finding local matches of PAWHEAE in HEAGAWGHEE:
  - HEAGAWGHEE
  - HEA.AW-HE.

EXAMPLE: REPEATED MATCHES

i

The best sum of completed match score to the subsequence $x_{1..i}$ assuming x_i is in an unmatched region

		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	1	1	1	1	1	3	9
P	0	0	0	0	1	1	1	1	1	3	9
A	0	0	0	5	1	6	1	1	1	3	9
W	0	0	0	0	2	1	21	13	5	3	9
H	0	10	2	0	1	1	13	19	23	15	9
E	0	2	16	8	1	1	5	11	19	29	21
A	0	0	8	21	13	6	1	5	11	21	28
E	0	0	6	13	18	12	4	1	5	17	27

j

9

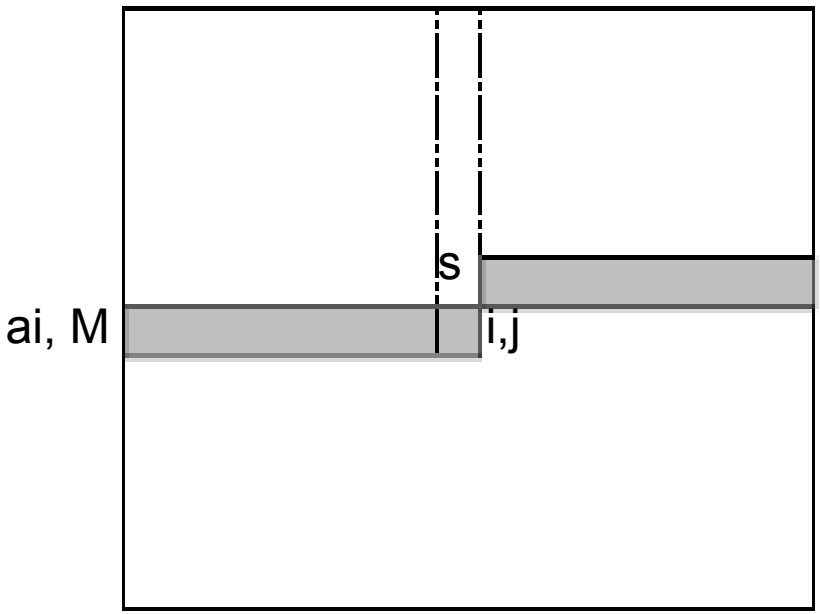
T=20

REPEATED MATCHES: ALGORITHM

$$H(i,0) = \max \begin{cases} H(i-1,0) \\ H(i-1,j) - T, \ \ \mathrm{j} = 1,..\mathrm{m} \end{cases}$$

$$H(i,j) = \max \begin{cases} H(i,0) \\ H(i-1,j-1) + s(x_i, y_j) \\ H(i-1,j) - d \\ H(i,j-1) - d \end{cases}$$

LINEAR SPACE COMPUTATION OF THE OPTIMAL ALIGNMENT SCORE (ONLY)



First row (M=0)

Gap penalty

1. $S[0] \leftarrow 0$
2. **for** $j \leftarrow 1$ **to** N **do**
3. $\quad S[j] \leftarrow S[j-1] + \sigma(\begin{bmatrix} - \\ b_j \end{bmatrix})$
4. **for** $i \leftarrow 1$ **to** M **do**
5. $\quad s \leftarrow S[0]$
6. $\quad S[0] \leftarrow c \leftarrow S[0] + \sigma(\begin{bmatrix} a_i \\ - \end{bmatrix})$
7. $\quad$ **for** $j \leftarrow 1$ **to** N **do**
8. $\quad\quad c \leftarrow \max\{S[j] + \sigma(\begin{bmatrix} a_i \\ - \end{bmatrix}), s + \sigma(\begin{bmatrix} a_i \\ b_j \end{bmatrix}), c + \sigma(\begin{bmatrix} - \\ b_j \end{bmatrix})\}$
9. $\quad\quad s \leftarrow S[j]$
10. $\quad\quad S[j] \leftarrow c$
11. write "Maximum alignment score is" $S[N]$

Gap(l) = -g*l

Only a single array S[0..N] is needed

LINEAR SPACE ALIGNMENT ALGORITHM (HIRSHBERG)

- Divide the DP matrix into half (middle row = int (M/2)
- Forward (row:1>M/2) & Backward (M>M/2) score-only pass
- Add the scores from the two passes on the middle raw
- Sweep along the middle raw and find the point with the highest score
- Divide the matrix into half
- Compute recursively
- Let T be the size of the original, then total size of the problem is at most T+(1/2)T +(1/4)T+... = 2T

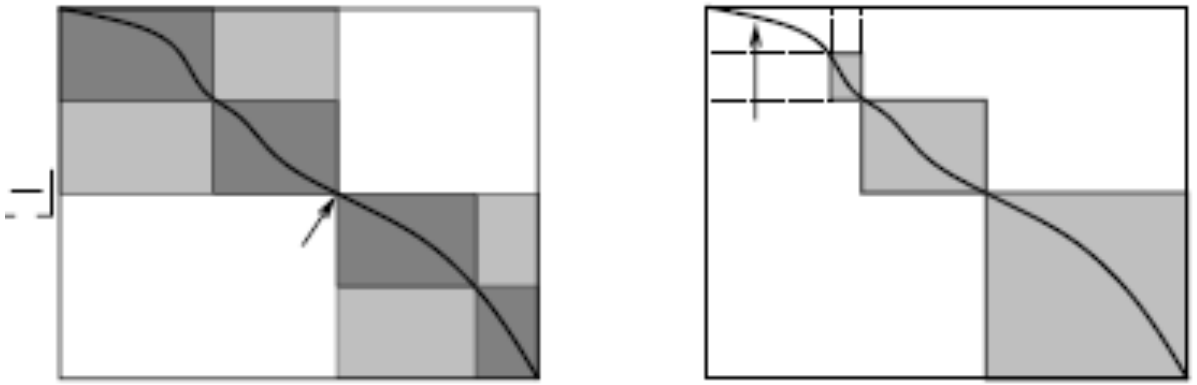
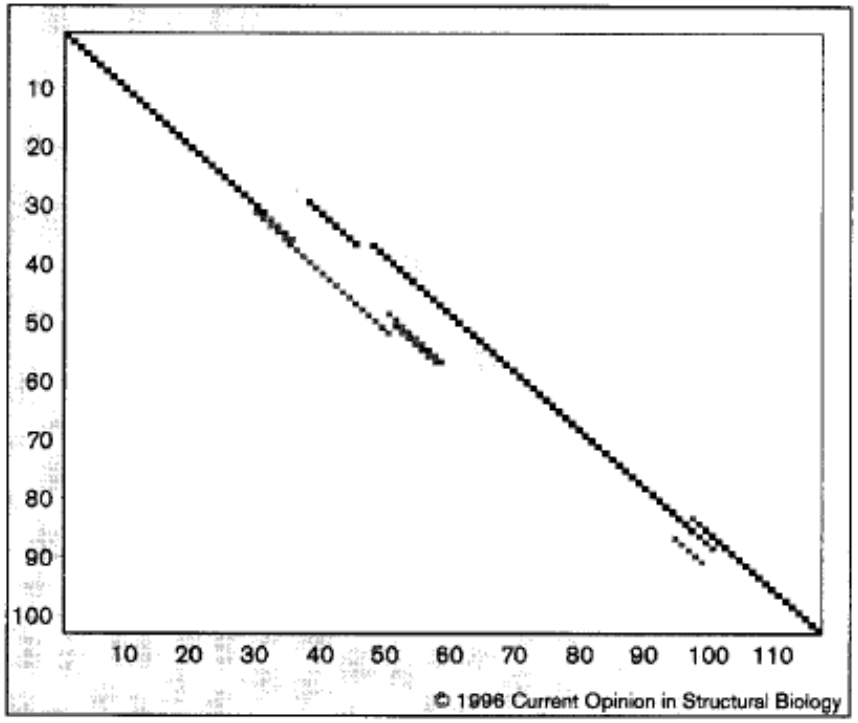


FIG. 8. (A) The two subproblems and four subsubproblems in Hirschberg's linear-space alignment procedure. (B) Snapshot of the execution of Hirschberg's algorithm. Shaded areas indicate problems remaining to be solved.

SUBOPTIMAL ALIGNMENTS

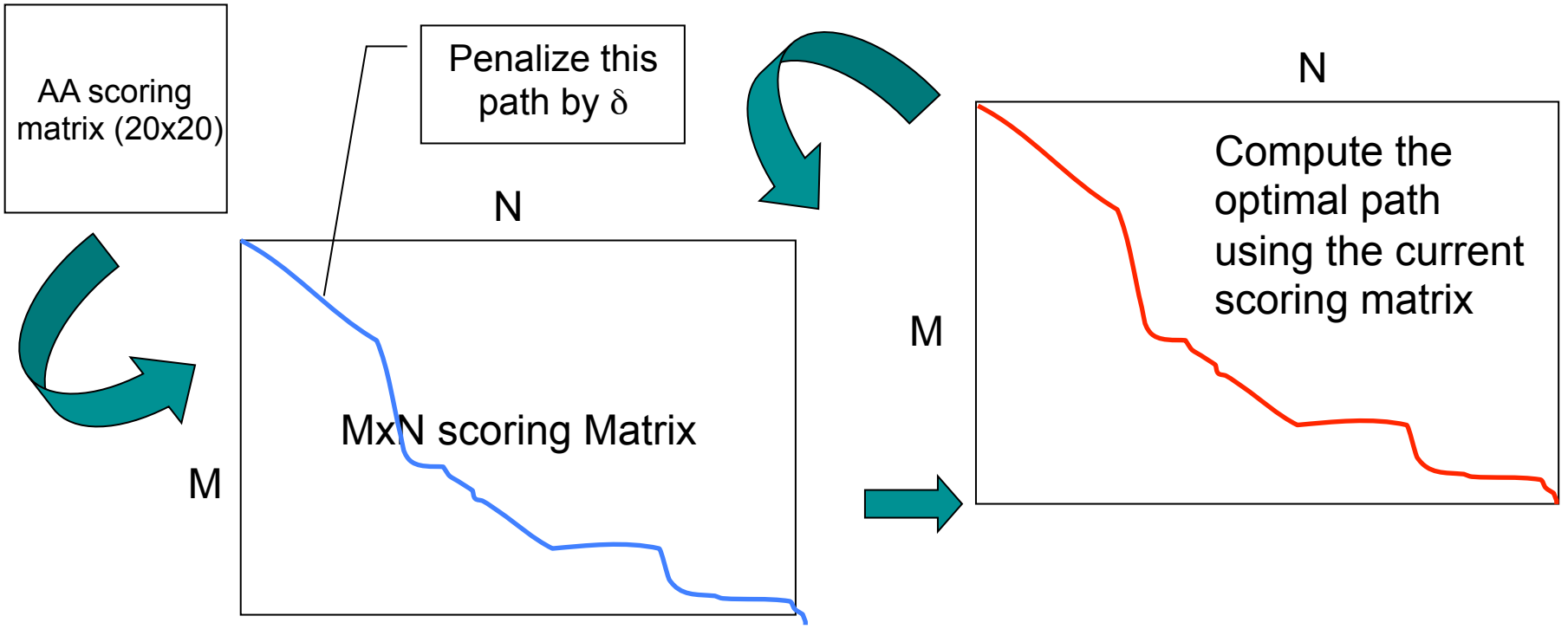


A dot plot depicting suboptimal points for the comparison of heavy (horizontal axis) and light (vertical axis) chain of the Fab antibody variable region. The best alignment is indicated in black, the second best alignment is dark gray and the third best is in light gray. Residue pairs on alignments scoring up to five points less than the optimum are shown. The lighter shade of gray is associated with points on alignments of a lower score.

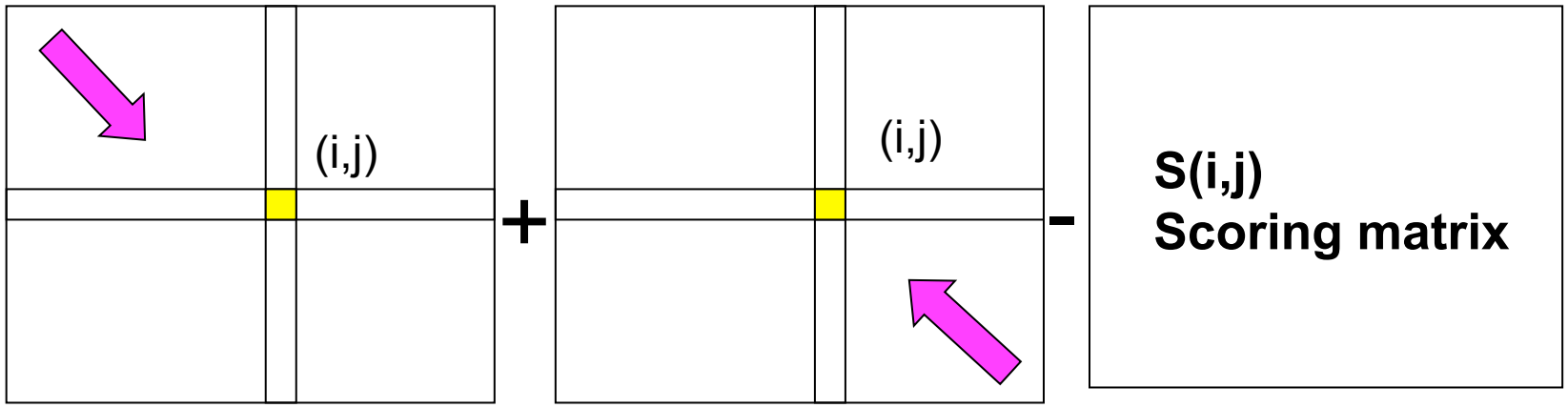
- So far we focused on obtaining the best scoring alignment
- Suboptimal alignments: The second best, the n-th best alignment
- Why?
  - The mathematically best alignment is not necessarily biologically correct

STERNBERG'S IDEA (1991)

- After the optimal alignment is computed, the path in the DP matrix is penalized:



VINGRON & ARGOS IDEA (1990)



D^+: Forward Matrix

Score at (i,j): The best score of alignment from (0,0) to (i,j)

D^- : Backward Matrix

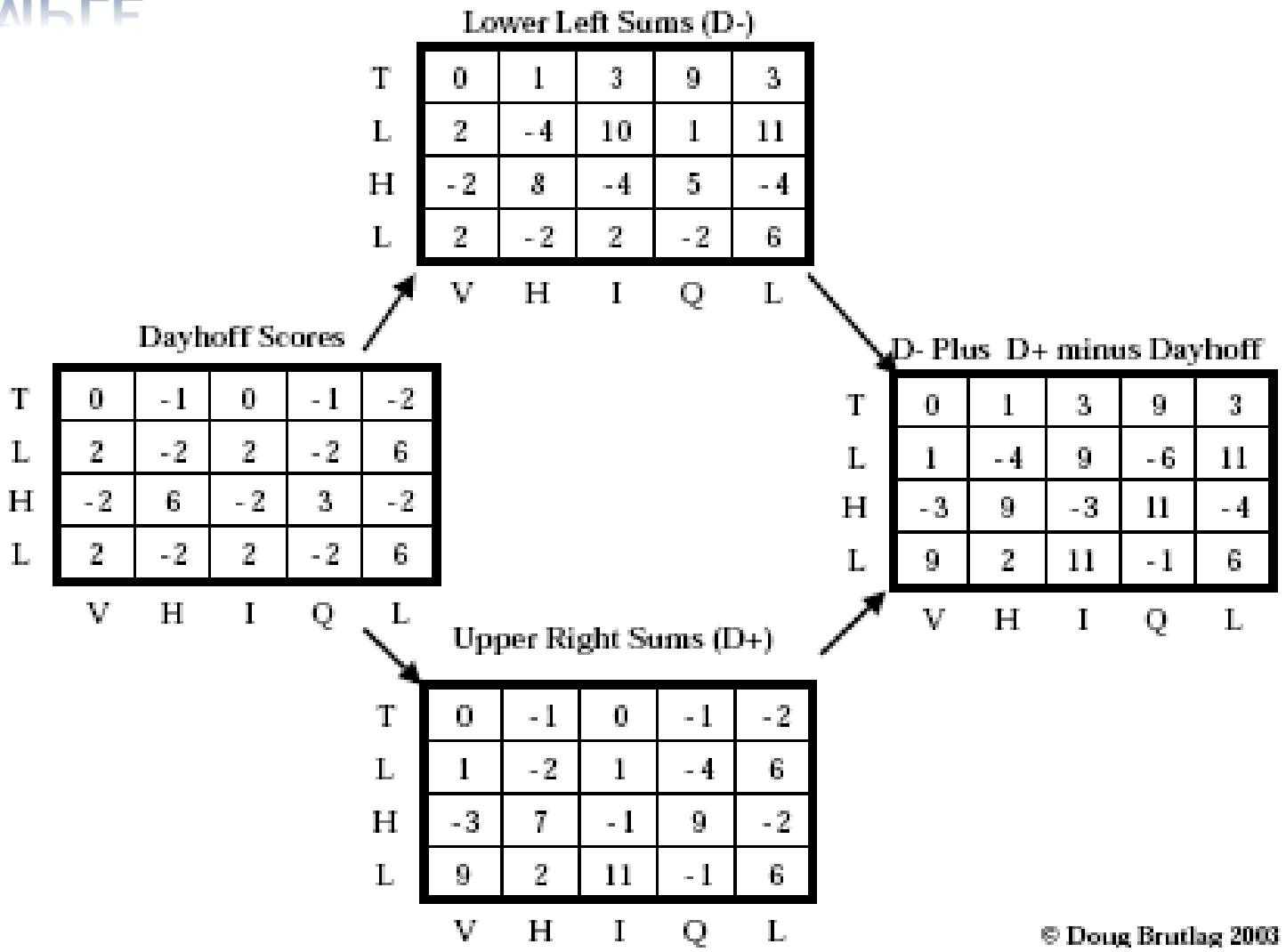
Score at (i,j): The best score of alignment from (M,N) to (i,j)

$D^+ + D - S$

= The best score of the alignment which goes through (i,j)

Compute the path (forward & backwards)

EXAMPLE



REFERENCES

- Biological Sequence Analysis, R. Durbin, S. Eddy, A. Krogh, G. Mitchison, Cambridge press
- “Recent developments in linear-space alignment methods: a survey”. Chao KM, Hardison RC, Miller W., J Comput Biol. 1(4):271-91, 1994.
- “A simple method to generate Non-trivial alternate alignments of protein sequences”, Saqi M.A.S., Sternberg, M.J.E. J. Mol. Biol 219: 727-732, 1991.
- “Determination of reliable regions in protein sequence alignments”, Vingron M., Argos, P. Prot. Engineering, 3: 565-569, 1990.