Optimal Sequence Alig-ment

(low-budget production version)

Plan 9 from outer space
https://commons.wikimedia.org/w/index.php?curid=1793456

Overview

- The alignment problem
- The dynamic programming solution
- Pairwise alignment: exact global and local solutions
- Multiple alignment and the cost of perfection

Recap: protein scoring

$$\log\left(\frac{C_{a,b}}{f(a)f(b)}\right)$$

C matrix – scaled frequencies of change from amino acid *a* to amino acid *b* (based on observed changes in some set)

Expectation based solely on frequencies of amino acids (changes not favoured / disfavoured)

Better than random: ratio > 1
Random: ratio = 1
Worse than random: ratio < 1

http://www.acm.org/crossroads/xrds13-1/gfx/13-1-11/Image7-BLOSUM62.jpg

Recap: protein scoring

$$D_{a,b} = S \cdot \log\left(\frac{C_{a,b}}{f(a)f(b)}\right)$$

Better than random: ratio > 1
Random: ratio = 1
Worse than random: ratio < 1

Magic (why?)

log (why?)

Better than random: $D_{a,b} > 0$

Random: $D_{a,b} = 0$

Worse than random: $D_{a,b} < 0$

http://www.acm.org/crossroads/xrds13-1/gfx/13-1-11/Image7-BLOSUM62.jpg

PAM scoring matrix

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

PAM250 matrix (S = 2, log base 2)

http://www.acm.org/crossroads/xrds13-1/gfx/13-1-11/Image7-BLOSUM62.jpg

DNA matrix

- Something like this usually works:

	A	G	C	T
A	1	-1	-1	-1
G	-1	1	-1	-1
C	-1	-1	1	-1
T	-1	-1	-1	1

- Or this:

	A	G	C	T
A	1	0.5	-1	-1
G	0.5	1	-1	-1
C	-1	-1	1	0.5
T	-1	-1	0.5	1

Back to the alignment problem

Given a scoring scheme *S*

and a set of homologous sequences, uh, *S*

introduce gaps if necessary to generate an alignment (let's call it *S*) that optimizes the score

So let's make some alignments!

Sequence S_1: length *m*

Sequence S_2: length *n*

In total, there are $\binom{n+m}{m}$ possible alignments of these sequences

n = m = 2:
4!/2!2! = 6 possibilities

```
AB--   AB- AB- AB  A-B -AB
--CD   -CD C-D CD  -CD CD-
```

n = m = 10: 184,756 possible alignments

Alignment of 2 sequences, each 100 amino acids in length:

$= 9.05485147 \times 10^{58}$ possibilities

Brute force is *not* going to work here…

The Key to Alignment

- If we were given the **midpoint X** within an optimal alignment of S_1 and S_2, we could **split on X** and solve each problem independently

```
MEH..KNP..TYL
MDH..KQP..SYI
```

```
MEH..K
MDH..K
```

+

```
P..TYL
P..SYI
```

- But we **don't know** any X, so divide and conquer isn't going to work

However…

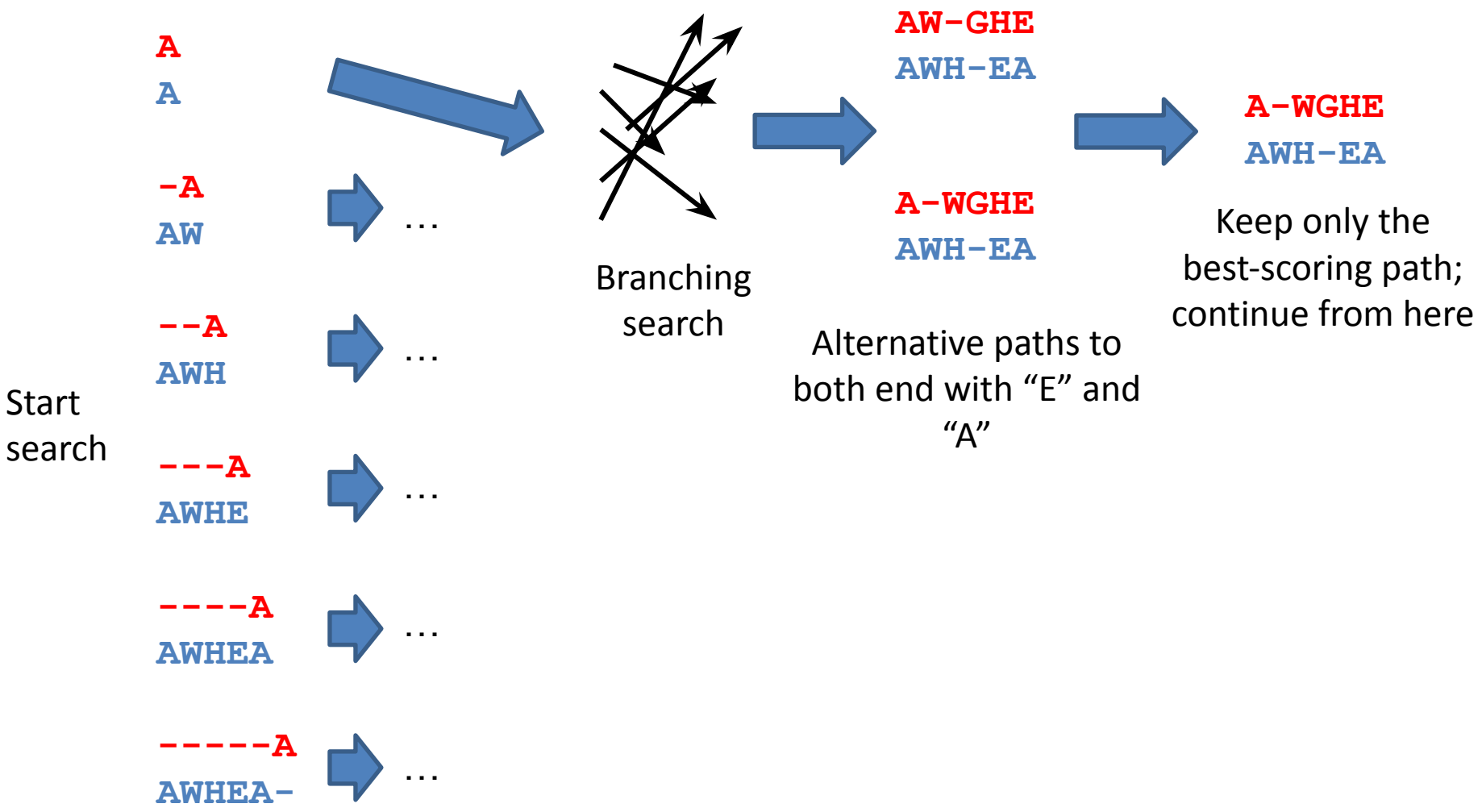
In searching for the best alignment:

- Start at the beginning of the sequences and consider **every possible X**

- BUT -

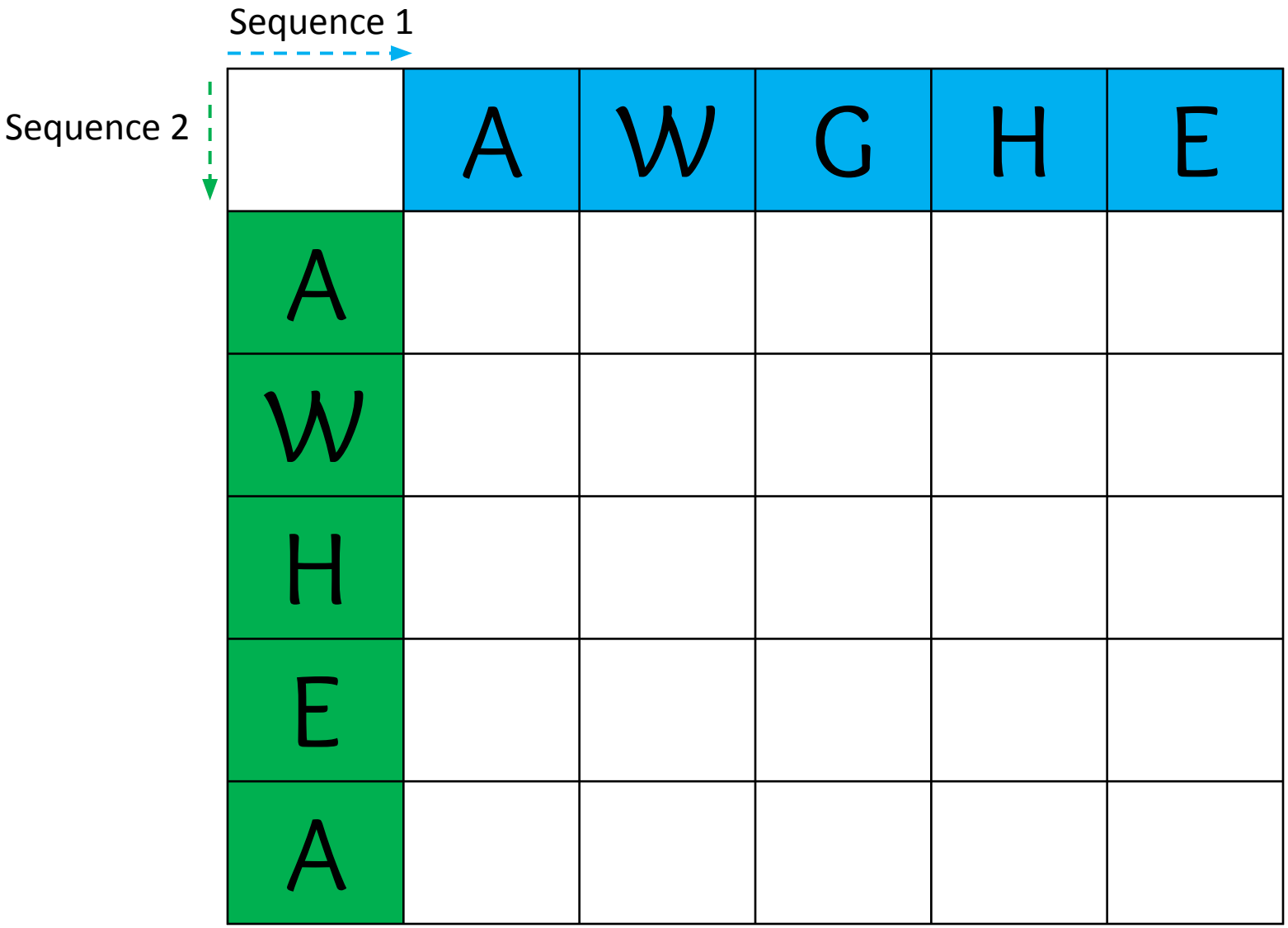
- Store only the best path (series of matches and gaps) that leads us to X

Consider an alignment of AWGHE vs AWHEA:



= **Dynamic Programming**

Consider an alignment of AWGHE vs AWHEA:



Every possible X

AWGHE vs. AWHEA	A	W	G	H	E
A	Best →(A,A)				
W					
H					
E		Best → (E,W)			
A					Best → (A,E)

Filling the matrix

We need our substitution matrix S and gap penalty scheme G

(we'll start with a linear gap penalty G = -gd)

For each possible X, consider the three immediate precursors

S = **PAM250**
g = **5**

AWGHE vs. AWHEA		A	W	G	H	E
	0					
A						
W						
H						
E						
A						

S = **PAM250**

g = **5**

-AWHEA
AWGHE

---AWHEA
AWGHE

-----AWHEA
AWGHE

AWGHE vs. AWHEA		A	W	G	H	E
	0	-5	-10	-15	-20	-25
A	-5					
W	-10					
H	-15					
E	-20					
A	-25					

AWHEA
-AWGHE

AWHEA
--AWGHE

AWHEA
---AWGHE

AWHEA
----AWGHE

AWHEA
-----AWGHE

insert gap in AWGHE

insert gap in AWHEA

S = **PAM250**
g = **5**

AWGHE vs. AWHEA		A	W	G	H	E
	0	-5	-10	-15	-20	-25
A	-5					
W	-10					
H	-15					
E	-20					
A	-25					

insert gap in AWGHE

insert gap in AWHEA

match

S(A,A) = 2

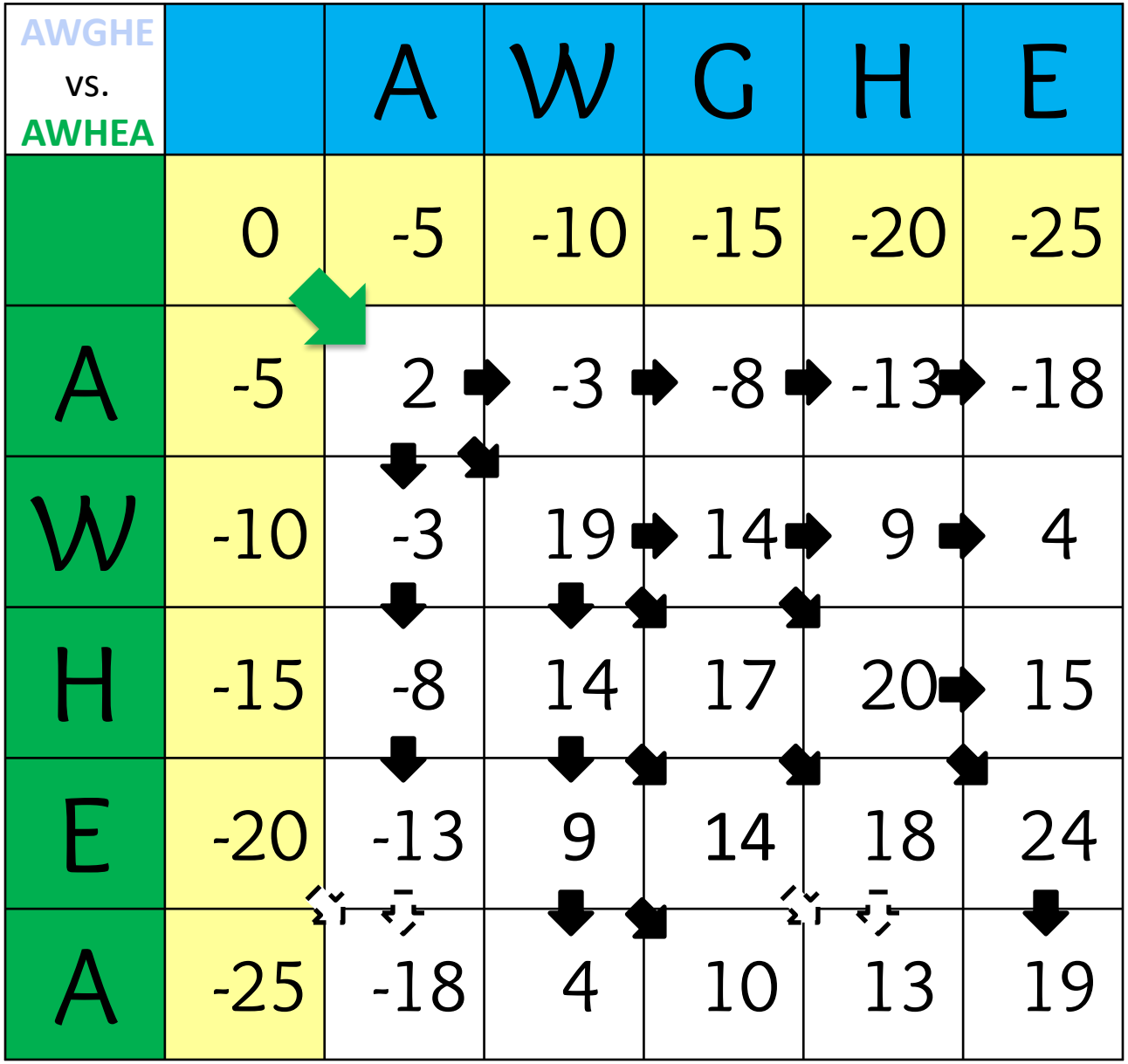
Therefore:
Insert -10
Insert -10
Match 2

AWGHE vs. AWHEA		A	W	G	H	E
	0	-5	-10	-15	-20	-25
A	-5	2				
W	-10					
H	-15					
E	-20					
A	-25					

	A	W	G	H	E
A					
W		F(2,2)	F(2,3)		
H		F(3,2)	F(3,3) = ?		
E					
A					

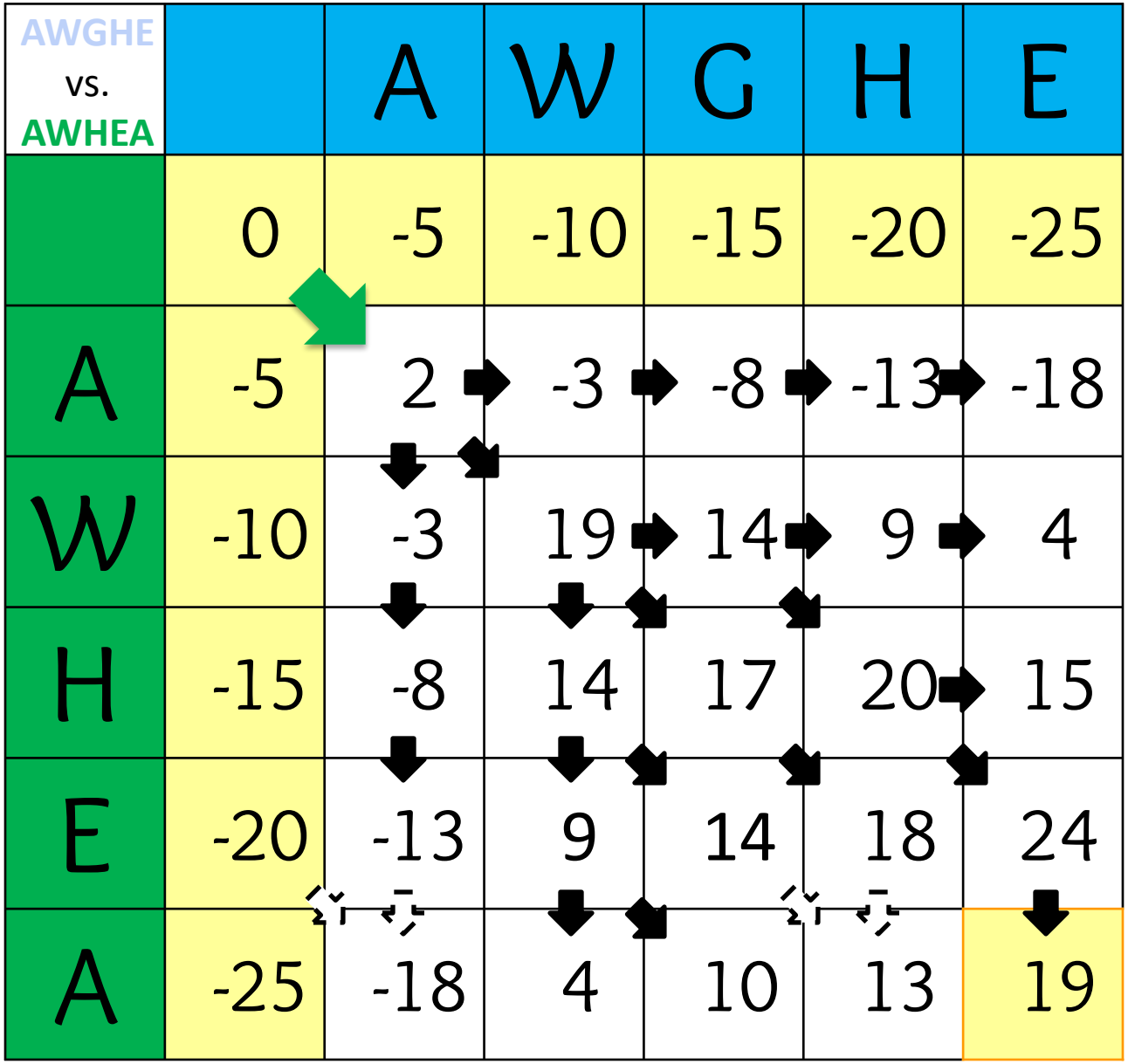
$$F(3,3) = \max \begin{cases} F(2,2) + S(G,H) & \text{match} \\ F(2,3) - d & \text{insert gap in AWGHE} \\ F(3,2) - d & \text{insert gap in AWHEA} \end{cases}$$

Remember paths INTO (not out of) each cell



Global Exact Alignment: Needleman-Wunsch

Since we have retained the best path to each F(x,y) in the matrix, we can trace back from F(m,n) to the origin and retrieve the optimal alignment path


AWGHE
vs.
AWHEA
A
W
G
H
E
0
-5
-10
-15
-20
-25
A
-5
2
-3
-8
-13
-18
W
-10
-3
19
14
9
4
H
-15
-8
14
17
20
15
E
-20
-13
9
14
18
24
A
-25
-18
4
10
13
19

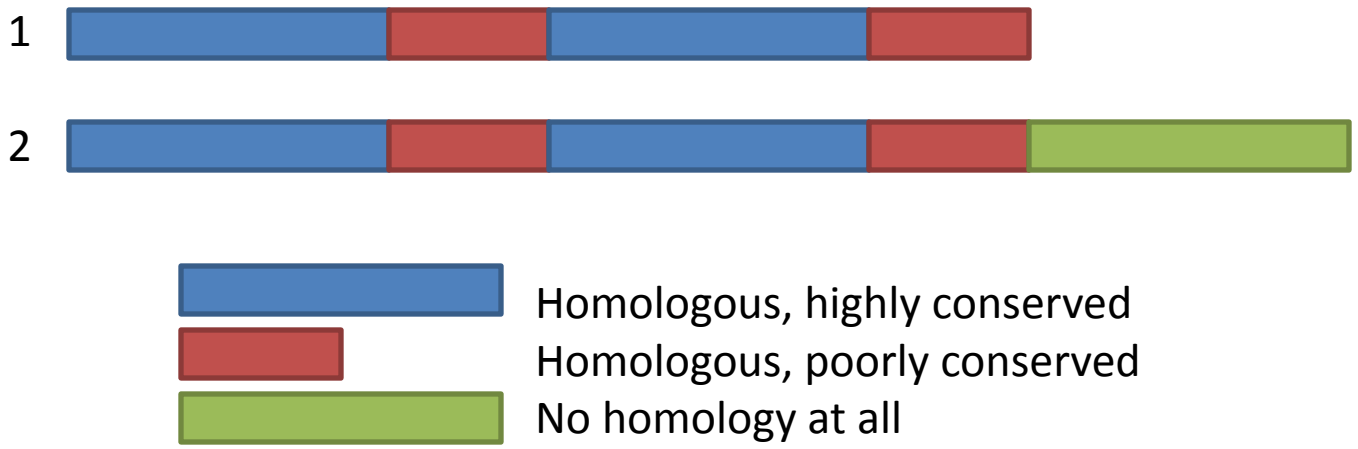
AWGHE−
AW−HEA

AWGHE vs. AWHEA		A	W	G	H	E
	0	-5	-10	-15	-20	-25
A	-5	2	-3	-8	-13	-18
W	-10	-3	19	14	9	4
H	-15	-8	14	17	20	15
E	-20	-13	9	14	18	24
A	-25	-18	4	10	13	19

Local Exact Alignment: Smith-Waterman

- Only return ‘good’ sub-alignments of the whole problem

- Useful, for instance, when

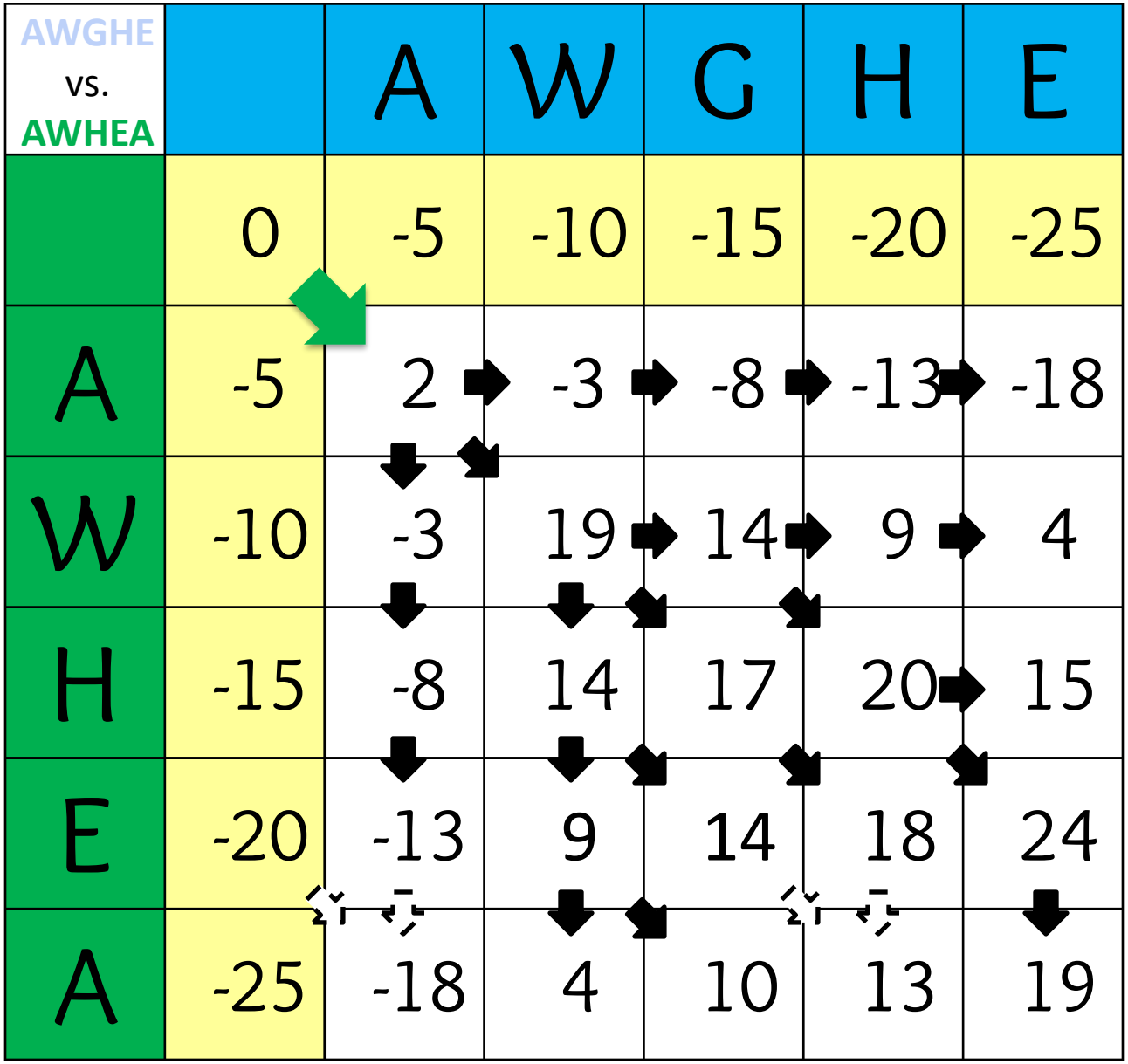


	A	W	G	H	E
A					
W		F(2,2)	F(3,2)		
H		F(2,3)	F(3,3) = ?		
E					
A					

$$F(3,3) = \max \begin{cases} F(2,2) + S(G,H) & \text{match} \\ F(3,2) - d & \text{insert gap in AWGHE} \\ F(2,3) - d & \text{insert gap in AWHEA} \\ \mathbf{0} & \text{Nothing is particularly good} \end{cases}$$

This is Needleman-Wunsch again

AWGHE–
AW–HEA



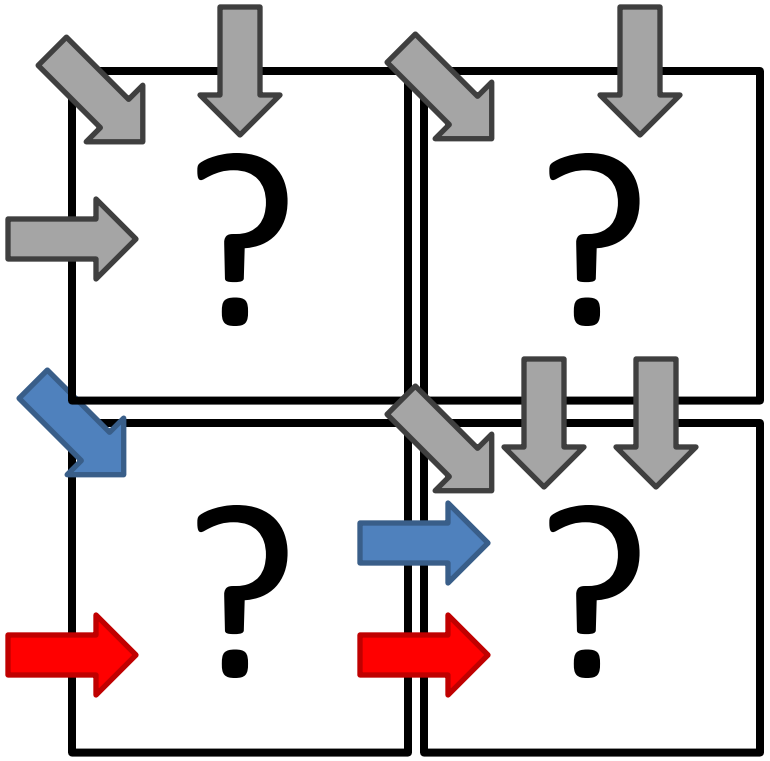
Slightly modified (non-trivial) S-W example

Find the **largest** value in the matrix, and trace back from there to 0

HE
HE

AWGHE vs. AYHEA		A	W	G	H	E
	0	0	0	0	0	0
A	0	2	0	1	0	0
Y	0	0	2	0	0	0
H	0	0	0	0	6	1
E	0	0	0	0	1	10
A	0	2	0	1	0	5

Affine Gap Penalties



Opening a new gap
(cost = d)

Extending a gap
(cost = e)

A horizontal move now has two possible costs; we need to consider both alternatives

(and therefore store the best scores for each box given horizontal, vertical, or diagonal entry)

Significance of S-W Alignments

RANDOMIZE n times

Compute Z-score for each replicate

$$Z(A,B) = \frac{S(A,B) - \tilde{m}}{\tilde{\sigma}}$$

Curve = null model of Z-score fit to Gumbel extreme value distribution

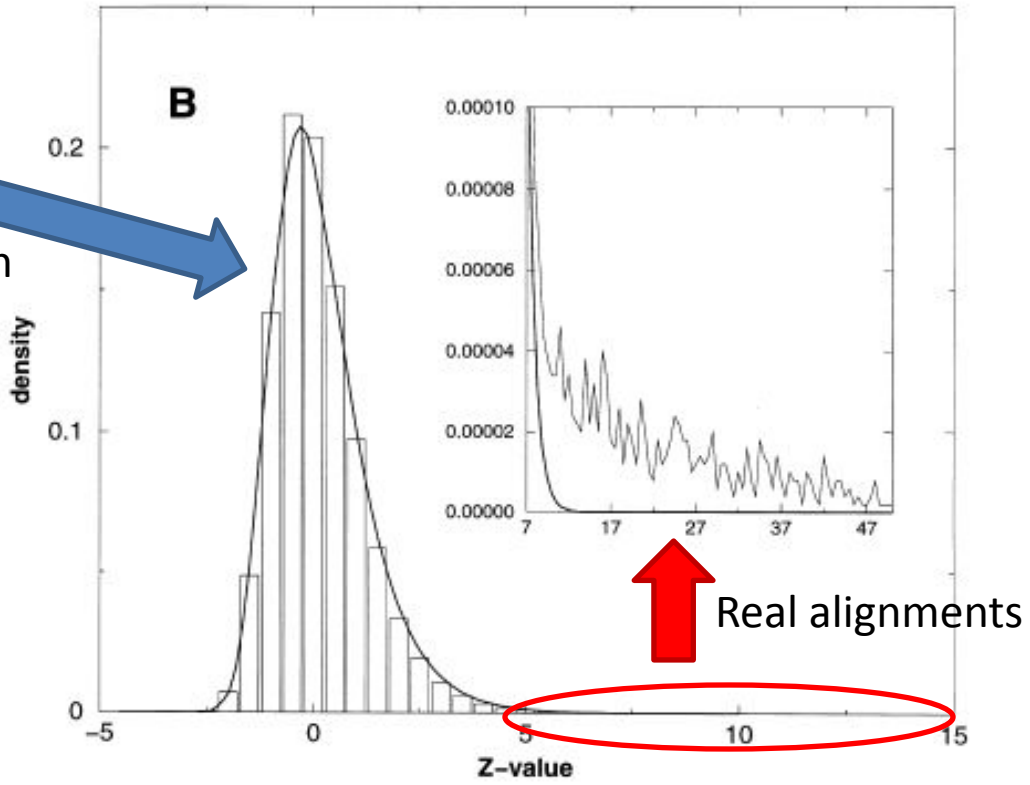


Fig. 6. Distribution of Z-values: (A) empirical distribution (rectangles) and Gumbel model (solide line) for quasi-real sequences. (Insert) the Gumbel model fits the experimental distribution for high Z-values. (B) empirical and Gumbel model for real sequences. (Insert) the Gumbel model (thick line) does not fit the experimental distribution (thin line) for high Z-values.

Comet et al., *Computers and Chemistry* (1999)

Alignment Complexity

- For each possible matching of a residue from sequence S_1 with a residue from S_2 , we need to carry out a constant number of computations and comparisons
- Total = 3 x *m* x *n*
- = O(*mn*)
- ~ O(n^2) if we assume *m* ≅ *n*

Multiple Sequence Alignment

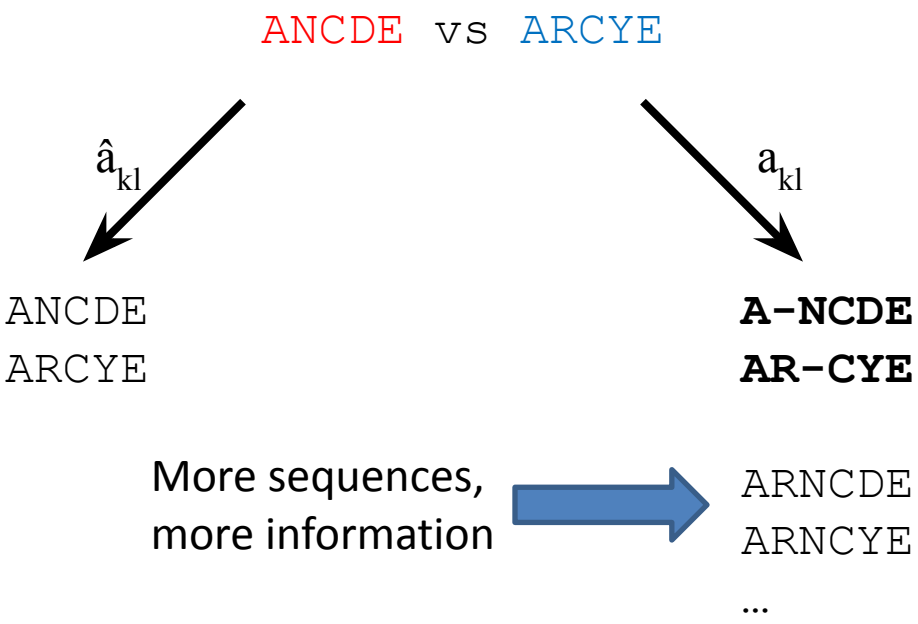
- In pairwise alignment, we are optimizing the score between two sequences

- When aligning 3 or more sequences, instead optimize the **sum of pairs** score:

```
1   N
2   Q
3   Q
4   D
```

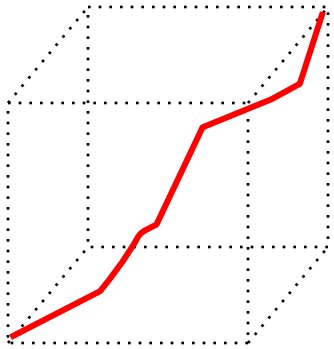
$$SP(N,Q,Q,D) = \begin{array}{l} 2 \times S(N,Q) \\ + 2 \times S(D,Q) \\ + S(Q,Q) \\ + S(N,D) \end{array}$$

The best alignment between a **pair of sequences** may not appear in the optimal **multiple alignment**



Multiple Sequence Alignment

- Dynamic programming on k sequences, each of length n requires construction of a k-dimensional matrix with n^k entries



- = $O(n^k)$

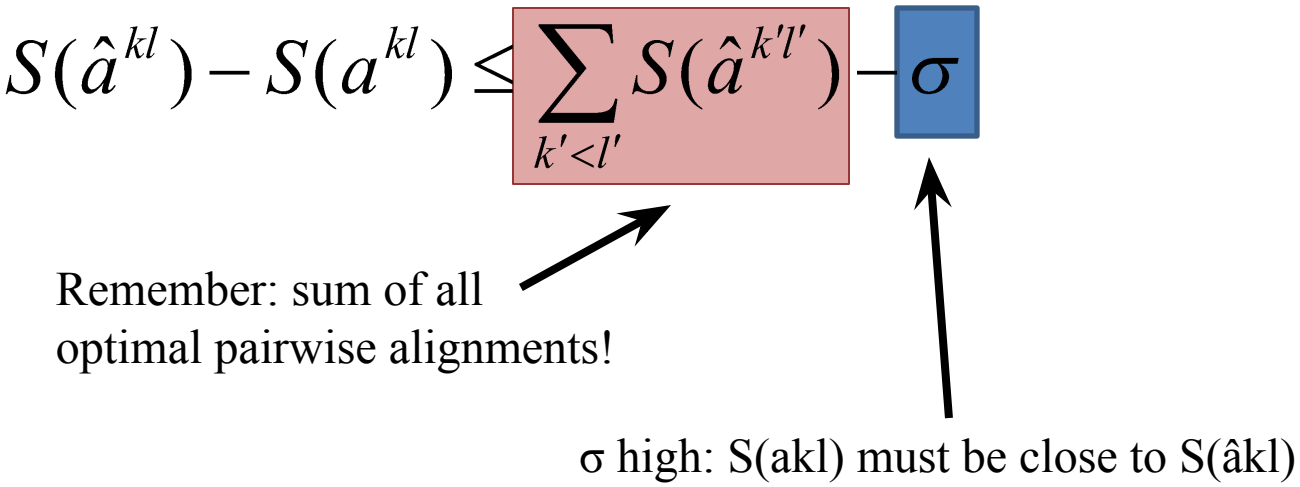
- Therefore **exponential** in the number of sequences!

MSA (Carrillo and Lipman, 1988)

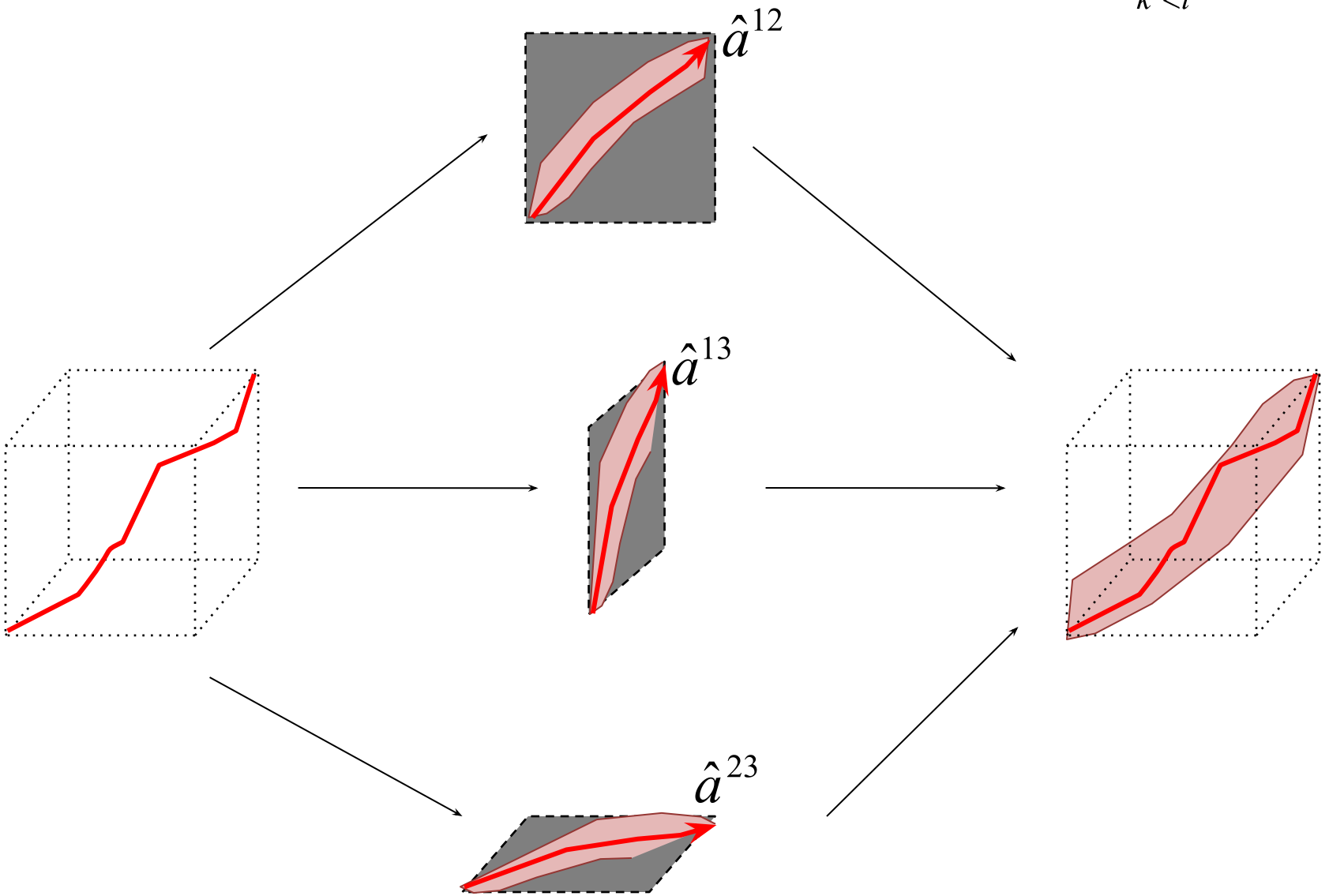
- The score of the optimal multiple alignment S(a) can be no greater than the sum of optimal pairwise alignments S($\hat{a}^{kl}$)

$$\sum_{k<l} S(a^{kl}) \leq \sum_{k<l} S(\hat{a}^{kl})$$

- If we can establish a lower bound σ on the multiple alignment score, then we constrain each S(a^{kl}):



Constrain each pairwise alignment to score no less than $\sigma + S(\hat{a}^{kl}) - \sum_{k'<l'} S(\hat{a}^{k'l'})$



So we need all optimal pairwise alignments

We also need σ. Where can we find it?

Types of multiple alignment

A. Block alignment

```
VRALFDF KGDILRI WWNA GMIPVPYV
FVALYDF KGEKLRV WCEA GWVPSNYI
VQALFDF RGDFIHV WWKG GMFPRNYV
VVALYDY KGDEYFI WWRA GYIPSNYV
FRAMYDY DGDAIIN WMYG GMLPANYV
VKALFDY KSAIIQN WWRG LWFPSNYV
YRALYDY LGDILTV WLNG GDFPGTYV
```

B. Segment alignment

```
EYVRALFDFNGNDEEDLPFKKGDILRIRDKPEEQ..............WWNAEDSEGKR.GMIPVPYVEK
NLFVALYDFVASGDNTLSITKGEKLRVLGYNHNGE..............WCEAQTKNGQ..GWVPSNYITP
TYVQALFDFDPQEDGELGFRRGDFIHVMDNSDPN..............WWKGACHGQT..GMFPRNYVTP
KKVVALYDYMPMNANDLQLRKGDEYFILEESNLP..............WWRARDKNGQE.GYIPSNYVTE
KIFRAMYDYMAADADEVSFKDGDAIINVQAIDEG..............WMYGTVQRTGRTGMLPANYVEA
CAVKALFDYKAQREDELTFIKSAIIQNVEKQEGG..............WWRGDYGGKKQ.LWFPSNYVEE
YQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFSDGQEARPEEIGWLNGYNETTGERGDFPGTYVEY
```

C. Local alignment

```
.........................aeyVRALFDFngndeedlpfkKGDILRIrdkpeeq..............WWNAedsegkr.GMIPVPYVek..........
.........................nlFVALYDFvasgdntlsitKGEKLRVlgynhnge..............WCEAqtkngq..GWVPSNYItpvns.......
lvdyhrstsvsrnqqiflrdieqvpqqptyVQALFDFdpqedgelgfrRGDFIHVmdnsdpn..............WWKGachgqt..GMFPRNYVtpvnrnv.....
.....................gsmstselkkVVALYDYmpmnandlqlrKGDEYFIleesnlp..............WWRArdkngqe.GYIPSNYVteaeds......
.......................tagkiFRAMYDYmaadadevsfkDGDAIINvqaideg..............WMYGtvqrtgrtGMLPANYVeai.........
.....................gsptfkcaVKALFDYkaqredeltfiKSAIIQNvekqegg..............WWRGdyggkkq.LWFPSNYVeemvnpegihrd
.........................gyqYRALYDYkkereedidlhLGDILTVnkgslvalgfsdgqearpeeigWLNGynettgerGDFPGTYVeyigrkkisp..
```

D. Global alignment

```
.........................AEYVRALFDFNGNDEEDLPFKKGDILRIRDKP..............EEQWWNAEDS.EGKRGMIPVPYVEK..........
.........................NLFVALYDFVASGDNTLSITKGEKLRVLGYN..............HNGEWCEAQTK..NGQGWVPSNYITPVNS.......
LVDYHRSTSVSRNQQIFLRDIEQVPQQPTYVQALFDFDPQEDGELGFRRGDFIHVMDNS..............DPNWWKGACH..GQTGMFPRNYVTPVNRNV.....
.....................GSMSTSELKKVVALYDYMPMNANDLQLRKGDEYFILEES..............NLPWWRARDK.NGQEGYIPSNYVTEAEDS......
.......................TAGKIFRAMYDYMAADADEVSFKDGDAIINVQAI..............DEGWMYGTVQRTGRTGMLPANYVEAI.........
.....................GSPTFKCAVKALFDYKAQREDELTFIKSAIIQNVEKQ..............EGGWWRGDYG.GKKQLWFPSNYVEEMVNPEGIHRD
.........................GYQYRALYDYKKEREEDIDLHLGDILTVNKGSLVALGFSDGQEARPEEIGWLNGYNETTGERGDFPGTYVEYIGRKKISP..
```

From Lecompte et al. (2001) *Gene*

Summary

- Dynamic programming allows the calculation of optimal pairwise alignments (for a given scoring scheme!)
- As soon as we go from 2 to >2 sequences, the exponential time complexity of the algorithm makes it impractical
- Need heuristics!