



1-month Practical Course Genome Analysis (Integrative Bioinformatics & Genomics)

Lecture 4: Pair-wise (2) and Multiple sequence alignment

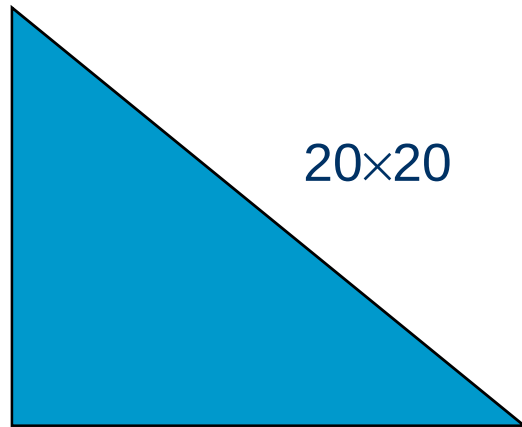
Centre for Integrative Bioinformatics VU (IBIVU)
Vrije Universiteit Amsterdam
The Netherlands

ibivu.nl

heringa@cs.vu.nl

Alignment input parameters

Scoring alignments



20x20

Amino Acid Exchange Matrix

A number of different schemes have been developed to compile residue exchange matrices

However, there are no formal concepts to calculate corresponding gap penalties

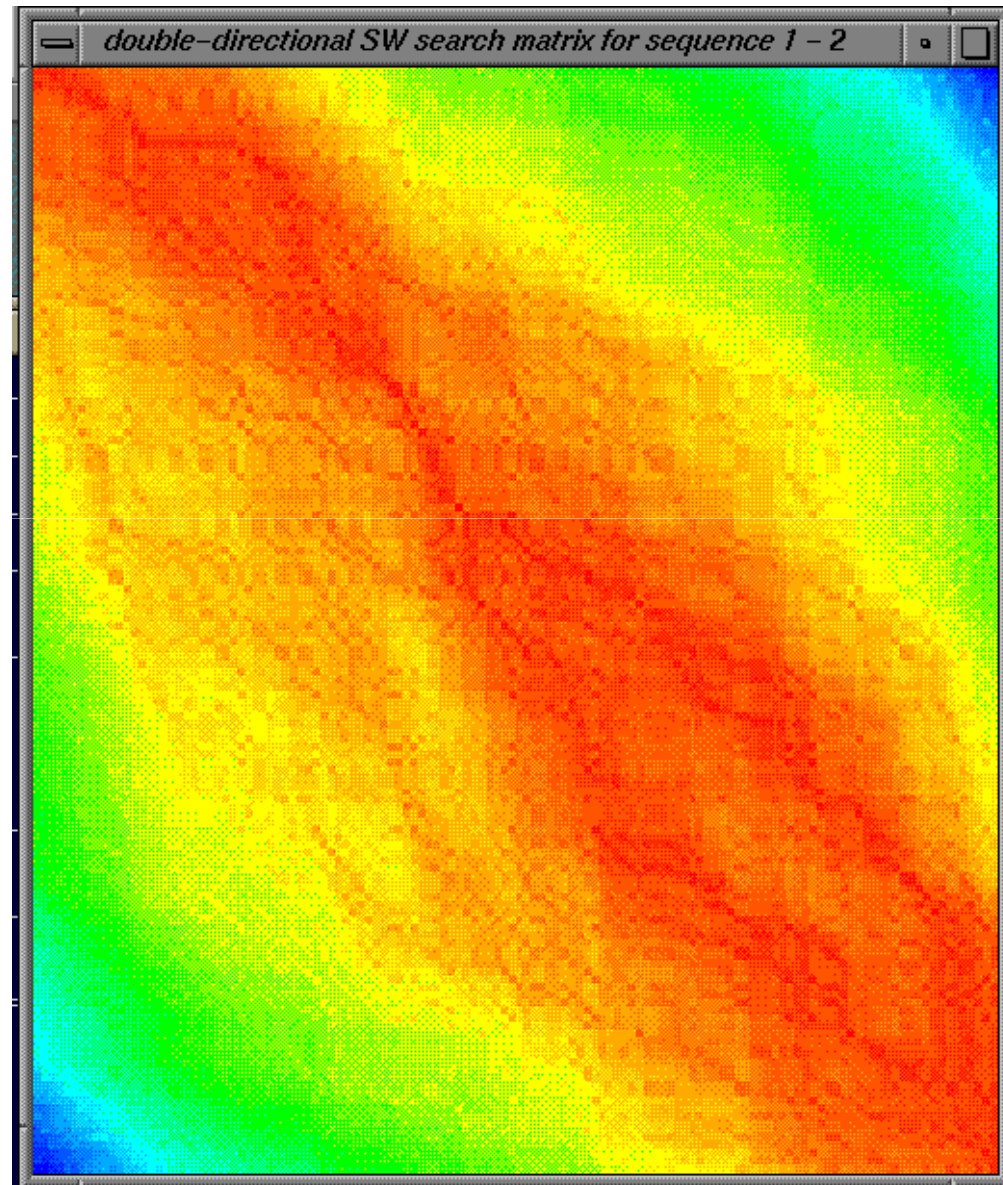
10

1

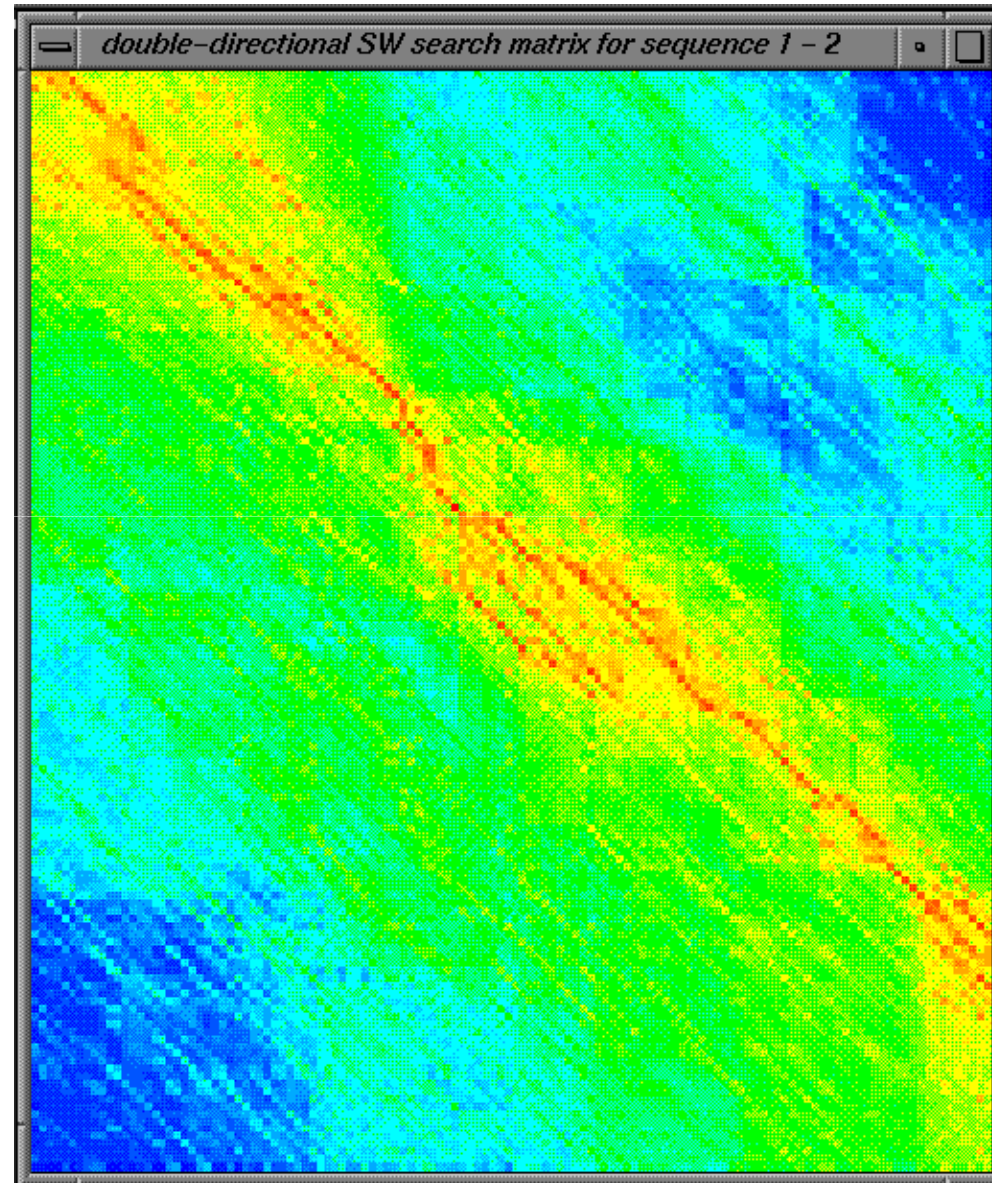
*Gap penalties (open,
extension)*

Empirically determined values are recommended for PAM250, BLOSUM62, etc.

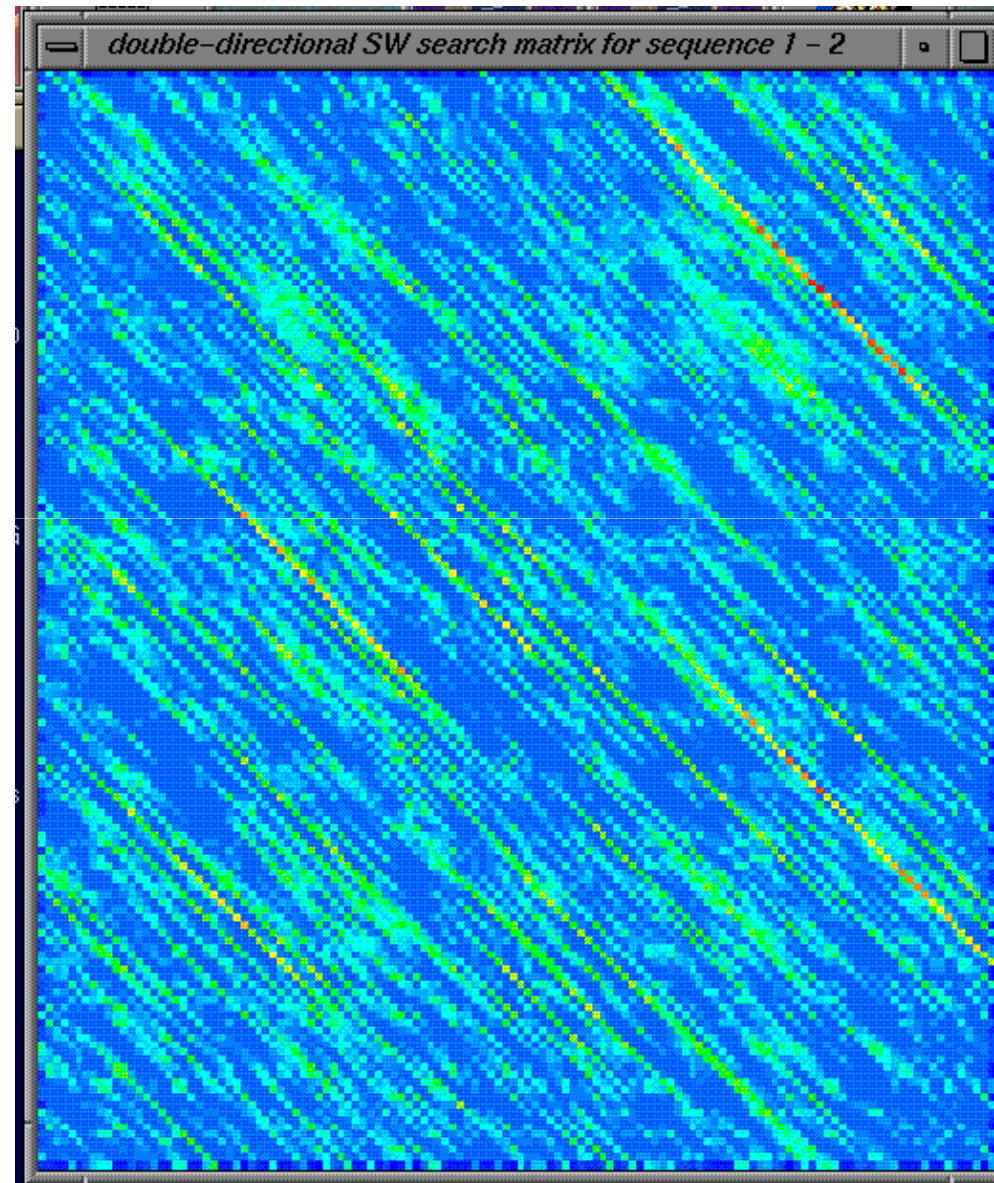
$M = \text{BLOSUM62}$, $P_o = 0$, $P_e = 0$



$M = \text{BLOSUM62}$, $P_o = 12$, $P_e = 1$



$M = \text{BLOSUM62}$, $P_o = 60$, $P_e = 5$



There are three kinds of alignments

- n Global alignment (preceding slides)
- n **Semi-global alignment**
- n Local alignment



Variation on global alignment

- n *Global* alignment: previous algorithm is called *global* alignment because it uses all letters from both sequences.

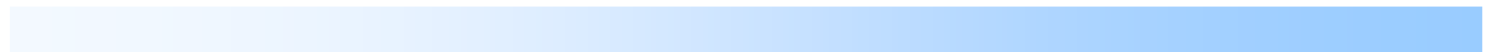
CAGCACTTGGATTCTCGG

CAGC - - - - - G - T - - - - GG

- n *Semi-global* alignment: uses all letters but does not penalize for end gaps

CAGCA-CTTGGATTCTCGG

- - - CAGCGTGG - - - - -



Semi-global alignment

- n Global alignment: all gaps are penalised
- n Semi-global alignment: N- and C-terminal (5' and 3') gaps (end-gaps) are not penalised

MSTGAVLIY - - TS - - - - -
- - - GGILLFHRTSGTSNS

End-gaps

End-gaps

A diagram illustrating semi-global alignment between two protein sequences. The first sequence is "MSTGAVLIY - - TS - - - - -" and the second is "- - - GGILLFHRTSGTSNS". Red arrows point to the leading gaps in each sequence, labeled "End-gaps" in blue italicized text. A light blue horizontal bar is at the bottom.

Semi-global alignment

Applications of *semi-global*:

- Finding a gene in genome
- Placing marker onto a chromosome
- One sequence much longer than the other



Risk: if gap penalties high -- really bad alignments for divergent sequences



Protein sequences have N- and C-terminal amino acids that are often small and hydrophilic

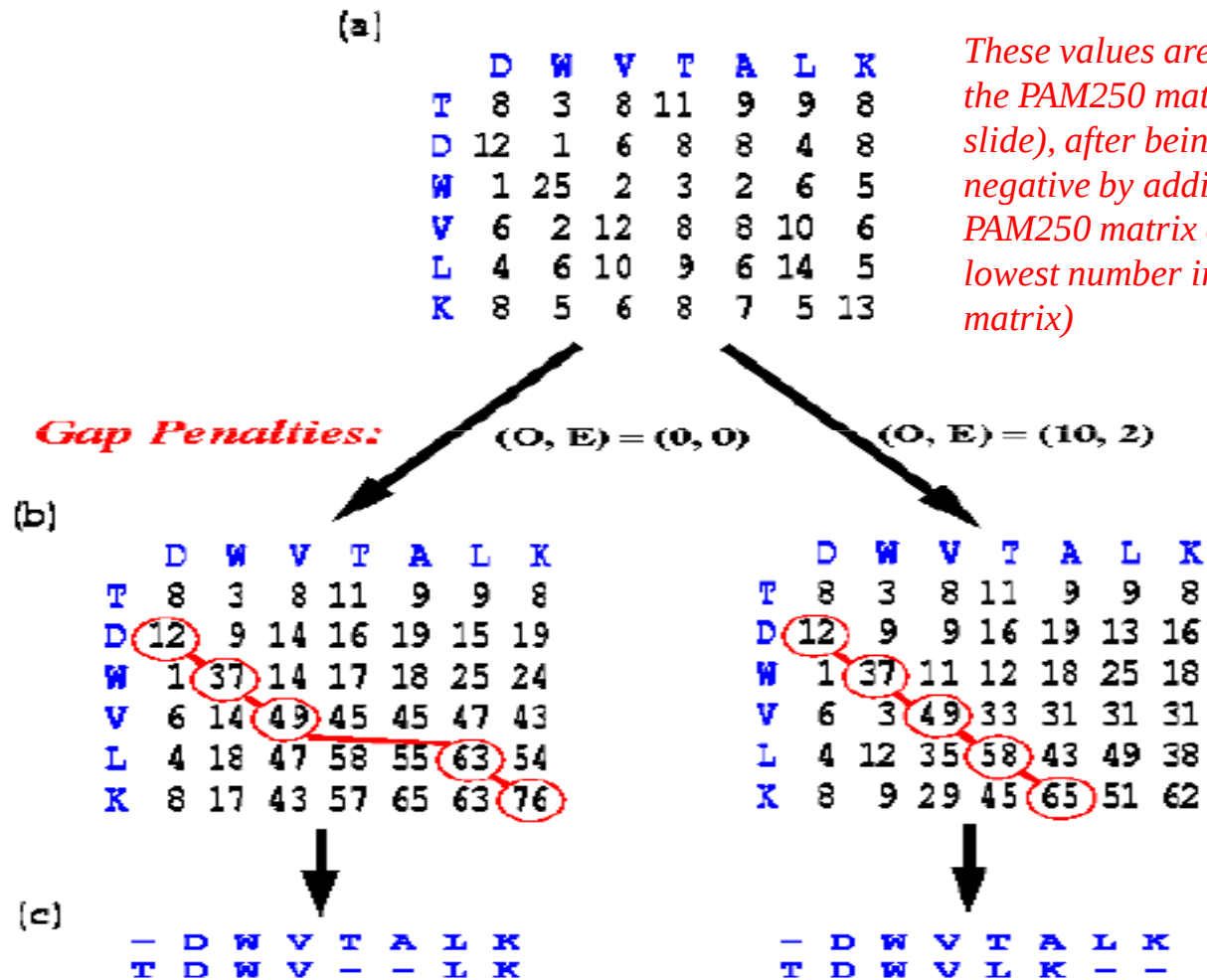
Semi-global alignment

- n Ignore 5' or N-terminal end gaps
 - First row/column set to 0
- n Ignore C-terminal or 3' end gaps
 - Read the result from last row/column (select the highest scoring cell)

	-	G	A	G	T	G
-	0	0	0	0	0	0
A	0	-1	1	-1	-1	-1
G	0	1	-2	2	0	-2
T	0	-1	0	0	3	1

Semi-global dynamic programming

- two examples with different gap penalties -



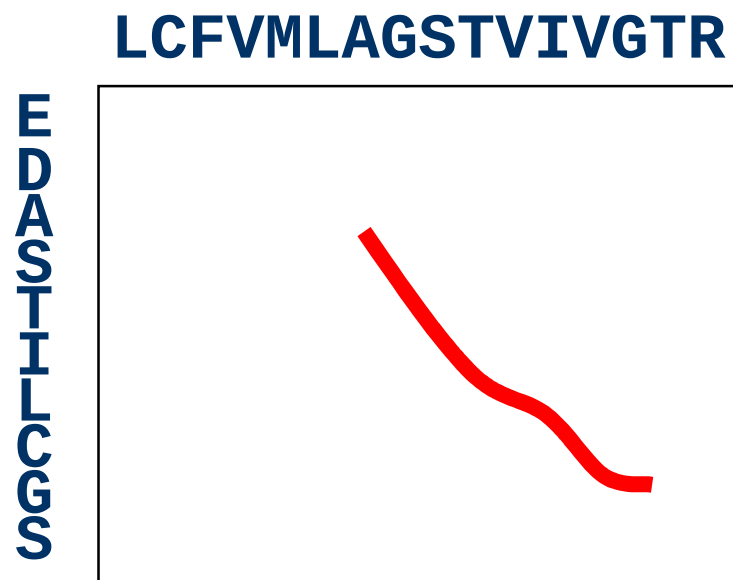
There are three kinds of alignments

- n Global alignment
- n Semi-global alignment
- n **Local alignment**



Local dynamic programming

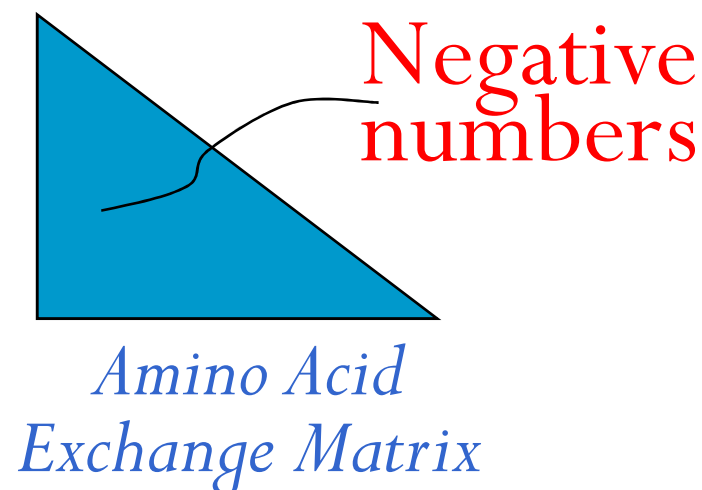
(Smith & Waterman, 1981)



Search matrix



AGSTVIVG
A-STILCG

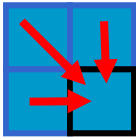


*Gap penalties
(open, extension)*

Local dynamic programming (Smith and Waterman, 1981)

basic algorithm

$$H(i,j) = \text{Max} \left\{ \begin{array}{ll} H(i-1,j-1) + S(i,j) & \text{diagonal} \\ H(i-1,j) - g & \text{vertical} \\ H(i,j-1) - g & \text{horizontal} \\ 0 & \end{array} \right.$$



The diagram shows a 2x2 grid of cells. The top-left cell is labeled $i-1$ and $j-1$. The bottom-left cell is labeled i and $j-1$. The bottom-right cell is labeled i and j . Red arrows indicate the possible transitions: a diagonal arrow from the top-left to the bottom-right, a vertical arrow from the top-left to the bottom-right, and a horizontal arrow from the top-left to the bottom-right. The bottom-right cell is highlighted with a black border.

Example:

local alignment of two sequences

n Align two DNA sequences:

- GAGTGA
- GAGGCGA (note the length difference)

n Parameters of the algorithm:

- **Match:** $\text{score}(A,A) = 1$
 - **Mismatch:** $\text{score}(A,T) = -1$
 - **Gap:** $g = -2$
- $$M[i, j] = \max \begin{cases} M[i-1, j-1] + 1 \\ M[i, j-1] - 2 \\ M[i-1, j] - 2 \\ 0 \end{cases}$$

The algorithm. Step 1: init

n Create the matrix

n Initiation

- No beginning row/column
- Just apply the equation...

$$M[i, j] = \max \begin{cases} M[i-1, j-1] \pm 1 \\ M[i, j-1] - 2 \\ M[i-1, j] - 2 \\ 0 \end{cases}$$

	$j \rightarrow$	1	2	3	4	5	6
$i \downarrow$		G	A	G	T	G	A
1	G						
2	A						
3	G						
4	G						
5	C						
6	G						
7	A						

The algorithm. Step 2: fill in

- n Perform the forward step...

$$M[i, j] = \max \begin{cases} M[i-1, j-1] \pm 1 \\ M[i, j-1] - 2 \\ M[i-1, j] - 2 \\ 0 \end{cases}$$

	$j \rightarrow$	1	2	3	4	5	6
$i \downarrow$		G	A	G	T	G	A
1	G	1	0	1	0	1	0
2	A	0	2	0	0	0	2
3	G	1	0	3	1	1	0
4	G	1	0	1	2		
5	C						
6	G						
7	A						

The algorithm. Step 2: fill in

- n Perform the forward step...

$$M[i, j] = \max \begin{cases} M[i-1, j-1] \pm 1 \\ M[i, j-1] - 2 \\ M[i-1, j] - 2 \\ 0 \end{cases}$$

	$j \rightarrow$	1	2	3	4	5	6
$i \downarrow$		G	A	G	T	G	A
1	G	1	0	1	0	1	0
2	A	0	2	0	0	0	2
3	G	1	0	3	1	1	0
4	G	1	0	1	2	2	0
5	C	0	0	0	0	1	1
6	G	1	0	1	0	1	0
7	A	0	2	0	0	0	2

The algorithm. Step 2: fill in

n We're done

$$M[i, j] = \max \begin{cases} M[i-1, j-1] \pm 1 \\ M[i, j-1] - 2 \\ M[i-1, j] - 2 \\ 0 \end{cases}$$

n Find the highest cell anywhere in the matrix

	$j \rightarrow$	1	2	3	4	5	6
$i \downarrow$		G	A	G	T	G	A
1	G	1	0	1	0	1	0
2	A	0	2	0	0	0	2
3	G	1	0	3	1	1	0
4	G	1	0	1	2	2	0
5	C	0	0	0	0	1	1
6	G	1	0	1	0	1	0
7	A	0	2	0	0	0	2

The algorithm. Step 3: trace back

- n Reconstruct path leading to highest scoring cell
- n Trace back until zero or start of sequence: alignment path can begin and terminate anywhere in matrix
- n Alignment: GAG
GAG

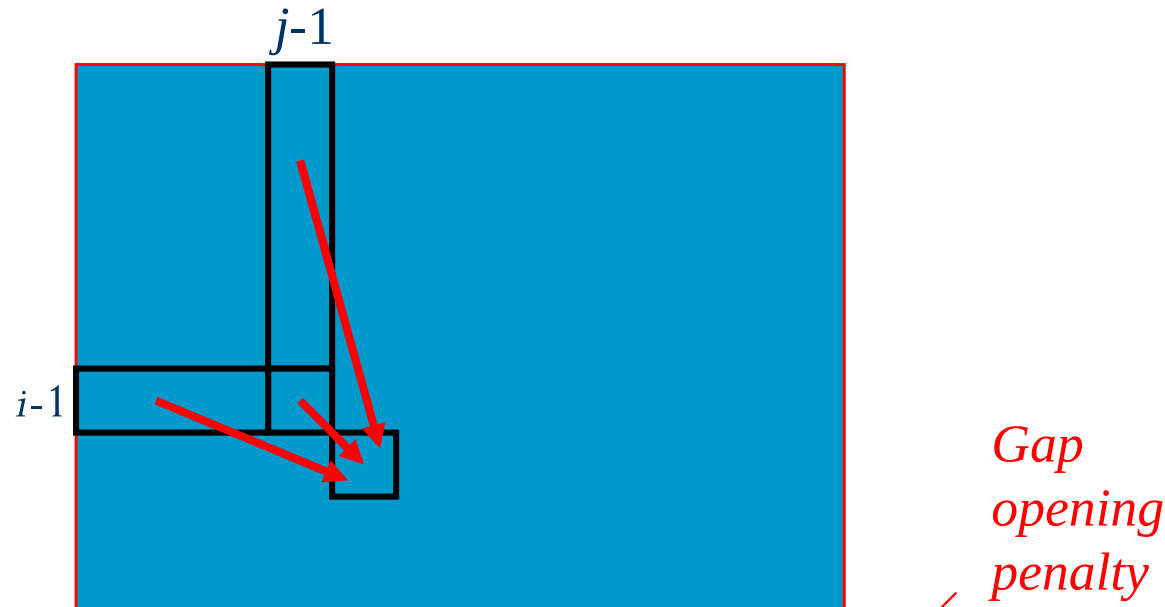
$$M[i, j] = \max \begin{cases} M[i-1, j-1] \pm 1 \\ M[i, j-1] - 2 \\ M[i-1, j] - 2 \\ 0 \end{cases}$$

	$j \rightarrow$	1	2	3	4	5	6
$i \downarrow$		G	A	G	T	G	A
1	G	1	0	1	0	1	0
2	A	0	2	0	0	0	2
3	G	1	0	3	1	1	0
4	G	1	0	1	2	2	0
5	C	0	0	0	0	1	1
6	G	1	0	1	0	1	0
7	A	0	2	0	0	0	2

Local dynamic programming

(Smith & Waterman, 1981; Gotoh, 1984)

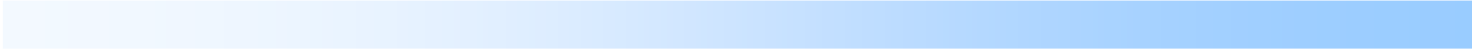
This is the general DP algorithm, which is suitable for linear, affine and concave penalties, although for the example here affine penalties are used



$$S_{i,j} = \text{Max} \begin{cases} S_{i,j} + \text{Max} \{ S_{0 < x < i-1, j-1} - P_i - (i-x-1)P_x \} \\ S_{i,j} + S_{i-1, j-1} \\ S_{i,j} + \text{Max} \{ S_{i-1, 0 < y < j-1} - P_i - (j-y-1)P_x \} \\ 0 \end{cases}$$

Gap extension penalty

Measuring Similarity

- n **Sequence identity** (number of identical exchanges per unit length)
 - n **Raw alignment score**
 - n **Sequence similarity** (alignment score normalised to a maximum possible)
 - n **Alignment score normalised** to a randomly expected situation (database/homology searching)
- 

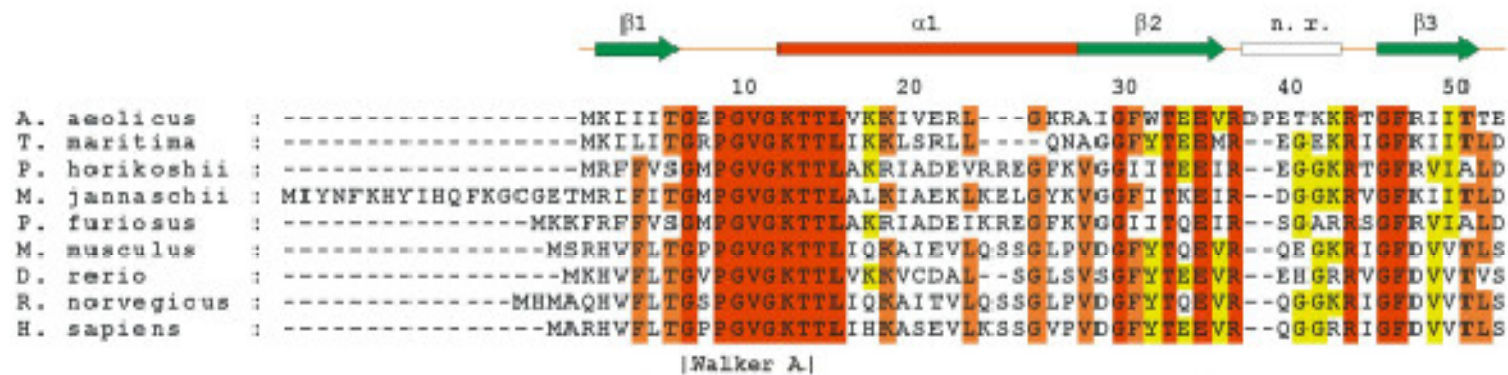
Pairwise alignment

- n Now we know how to do it:
- n How do we get a multiple alignment (three or more sequences)?
- n Multiple alignment: much greater combinatorial explosion than with pairwise alignment.....



Multiple alignment idea

- Take three or more **related** sequences and align them such that the greatest number of similar characters are aligned in the same column of the alignment.

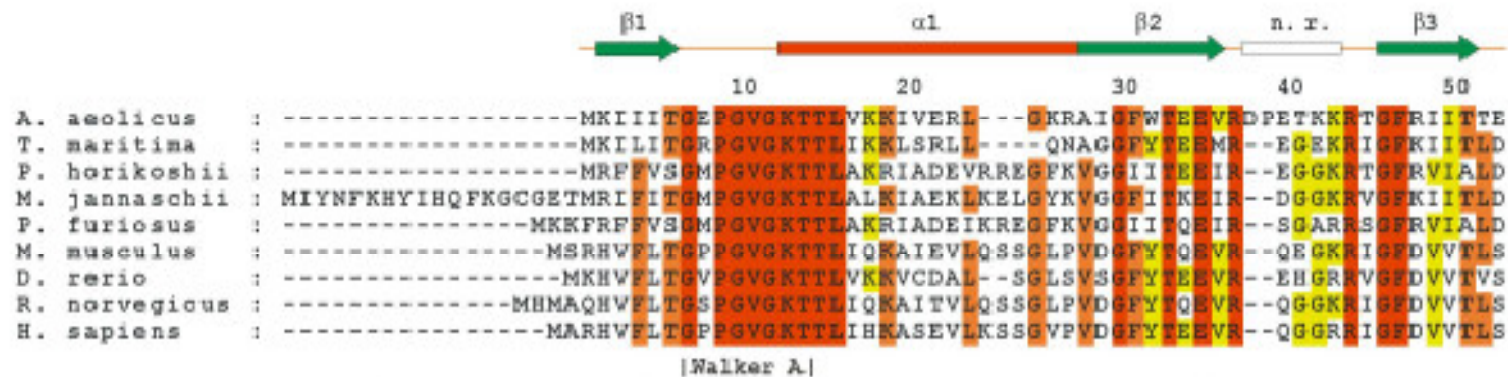


Ideally, the sequences are orthologous, but often include paralogues.

Scoring a multiple alignment

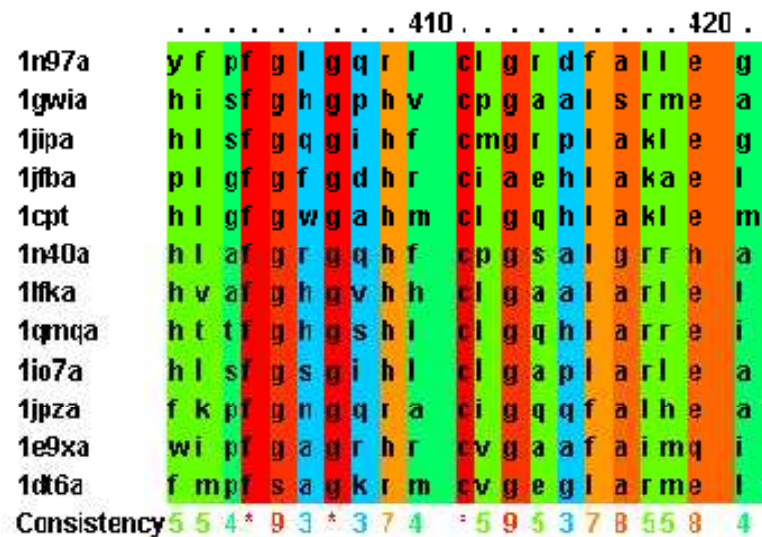
- You can score a multiple alignment by taking all the pairs of aligned sequences and adding up the pairwise scores:

$$S_{a,b} = \sum_i s(a_i, b_j) - \sum_k N_k \bullet gp(k)$$

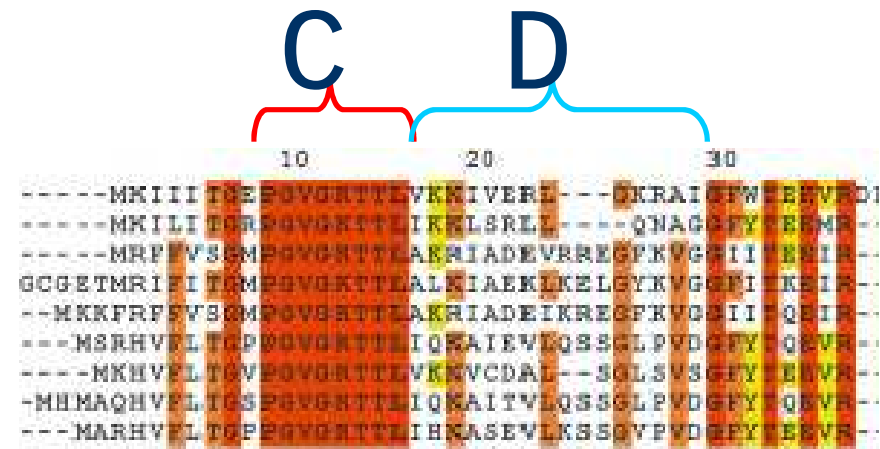


- This is referred to as the Sum-of-Pairs score

Information content of a multiple alignment



F



What to ask yourself

- What program to use to get a multiple alignment?
(three or more sequences)
- What is our aim?
 - Do we go for max accuracy?
 - Least computational time?
 - Or the best compromise?
- What do we want to achieve each time?



Multiple alignment methods

- q **Multi-dimensional dynamic programming**

- > extension of pairwise sequence alignment.

- q **Progressive alignment**

- > incorporates phylogenetic information to guide the alignment process

- q **Iterative alignment**

- > correct for problems with progressive alignment by repeatedly realigning subgroups of sequence



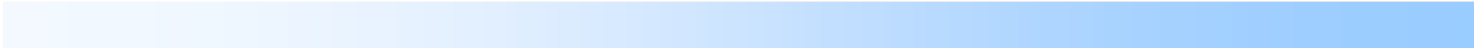
Exhaustive & Heuristic algorithms

- Exhaustive approaches
 - Examine all possible aligned positions simultaneously
 - Look for the optimal solution by (multi-dimensional) DP
 - Very (very) slow
- Heuristic approaches
 - Strategy to find a near-optimal solution (by using rules of thumb)
 - Shortcuts are taken by reducing the search space according to certain criteria
 - Much faster

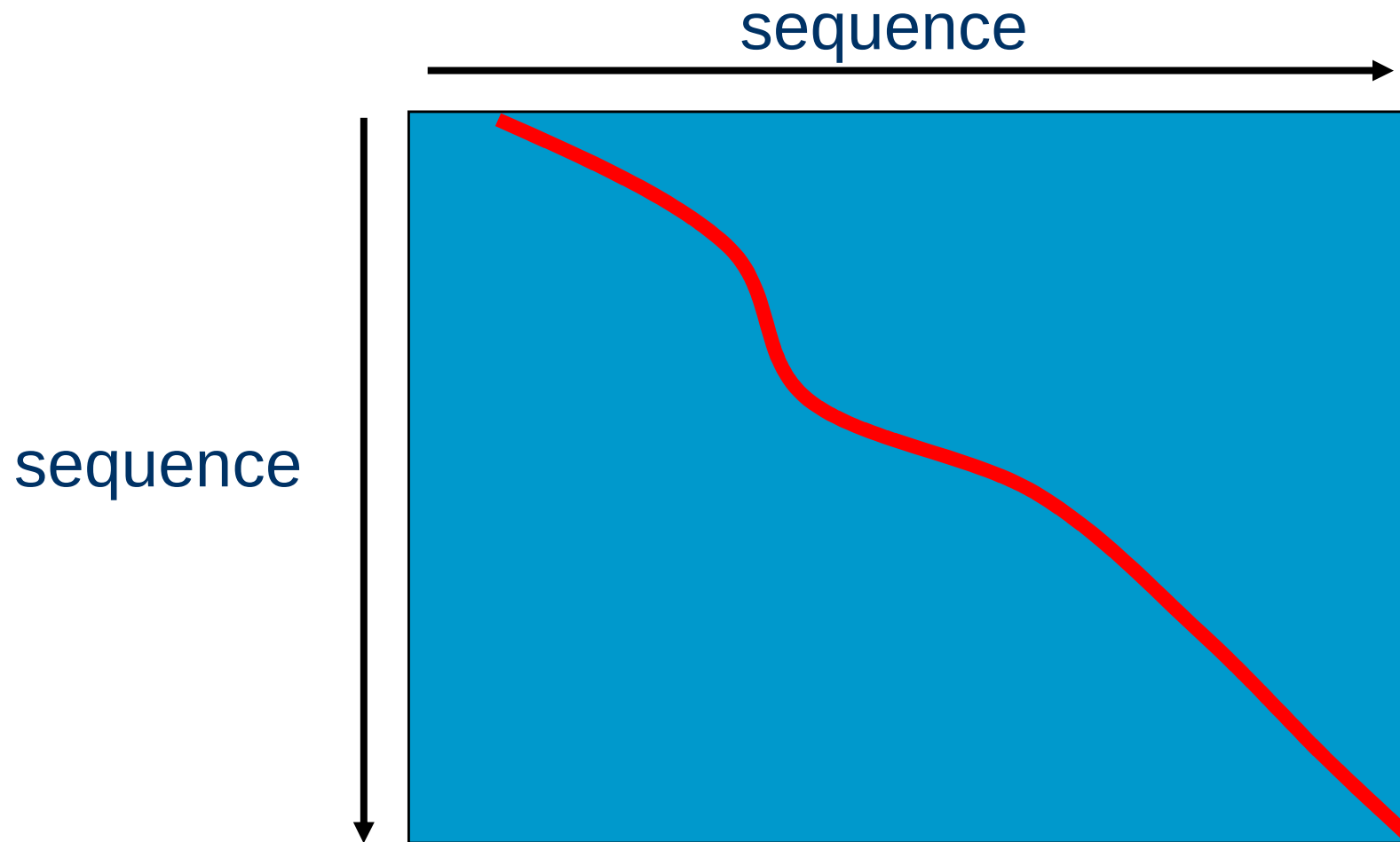
Simultaneous multiple alignment

Multi-dimensional dynamic programming

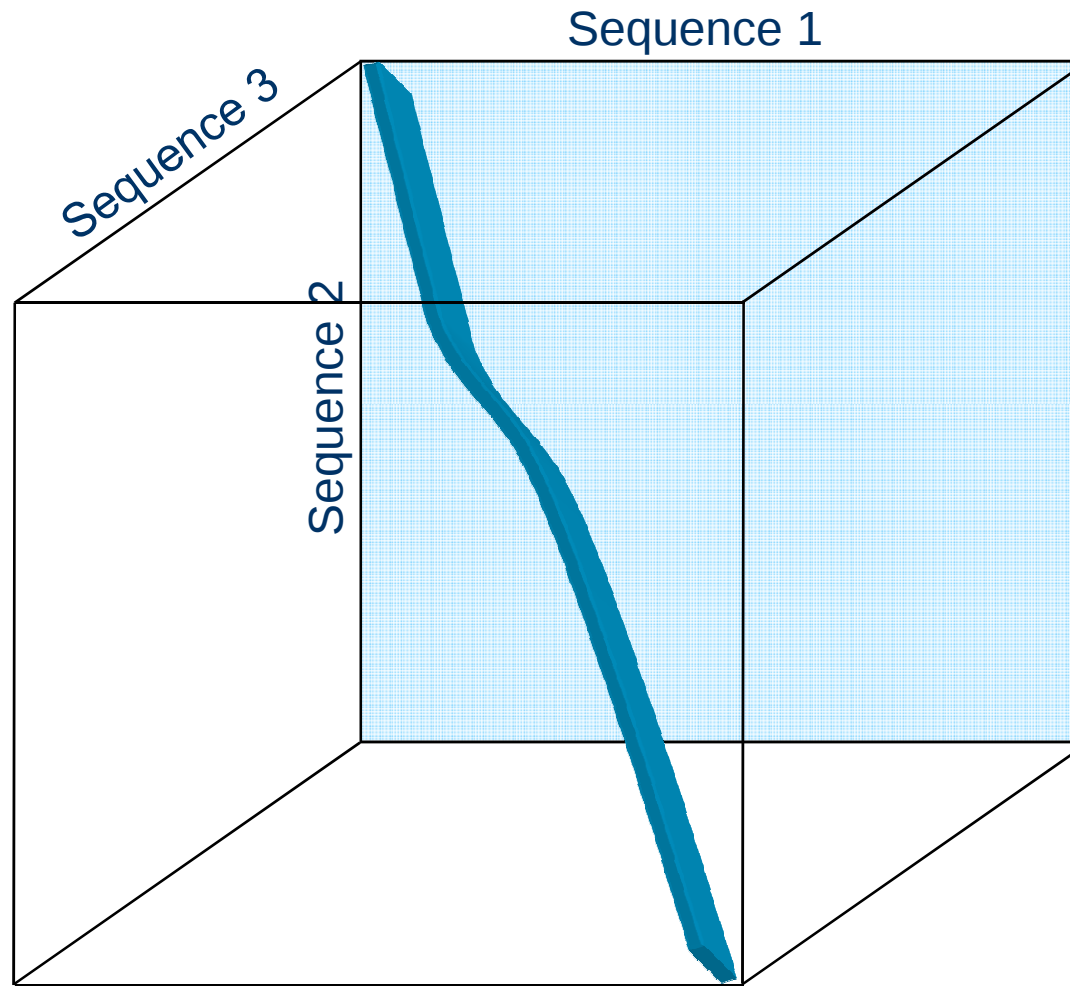
Combinatorial explosion

- q DP using two sequences of length n
 Θn^2 comparisons
 - q Number of comparisons increases exponentially
 Θ i.e. n^N where n is the length of the sequences, and N is the number of sequences
 - q Impractical even for small numbers of short sequences
- 

Sequence-sequence alignment by Dynamic Programming



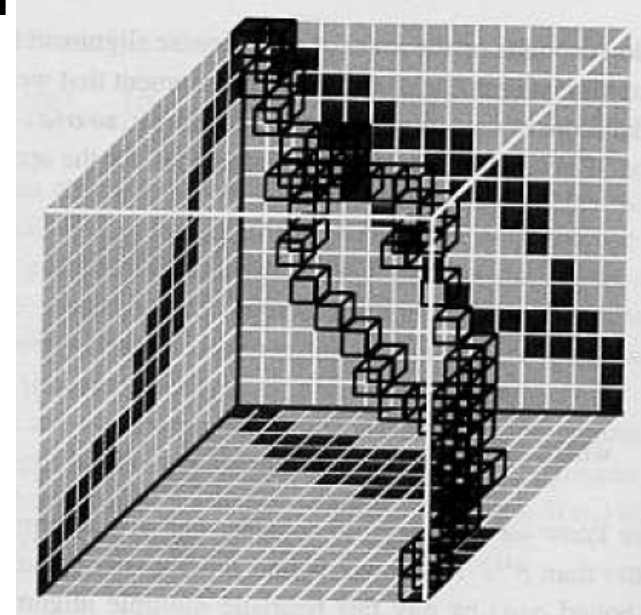
Multi-dimensional dynamic programming (Murata et al., 1985)



The MSA approach

Lipman et al. 1989

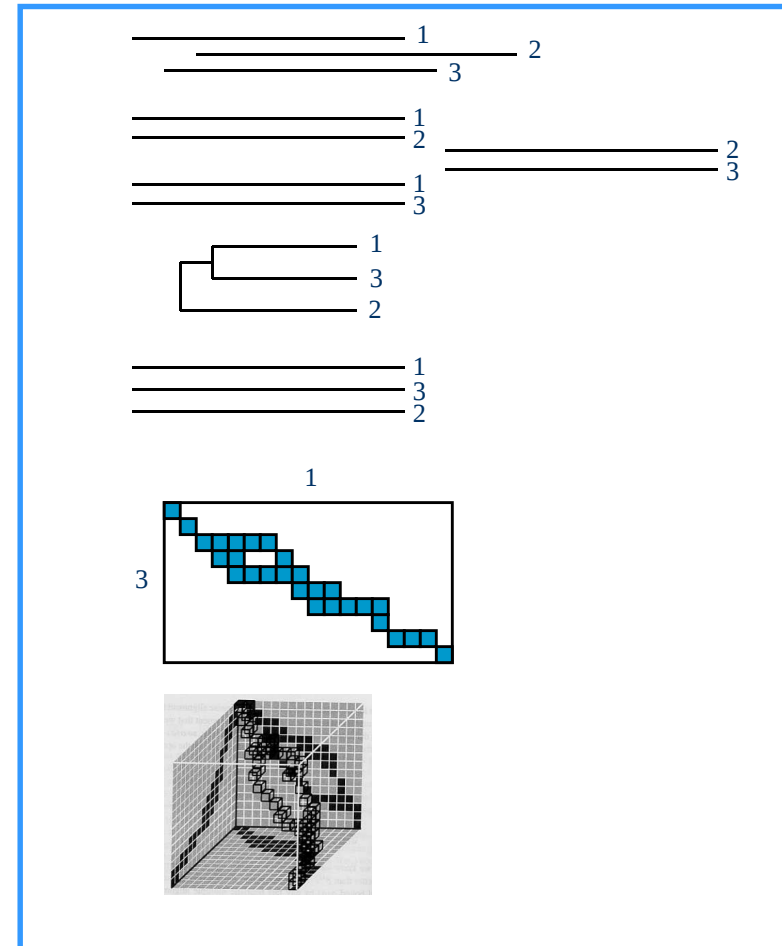
- n **Key idea**: restrict the computational costs by determining a minimal region within the n-dimensional matrix that contains the optimal path



The MSA method *in detail*

1. Let's consider 3 sequences
2. Calculate all pair-wise alignment scores by Dynamic programming
3. Use the scores to predict a tree
4. Produce a heuristic multiple align. based on the tree (quick & dirty)
5. Calculate maximum cost for each sequence pair from multiple alignment (upper bound) & determine paths with < costs.
6. Determine spatial positions that must be calculated to obtain the optimal alignment (intersecting areas or 'hypersausage' around matrix diagonal)
7. Perform multi-dimensional DP

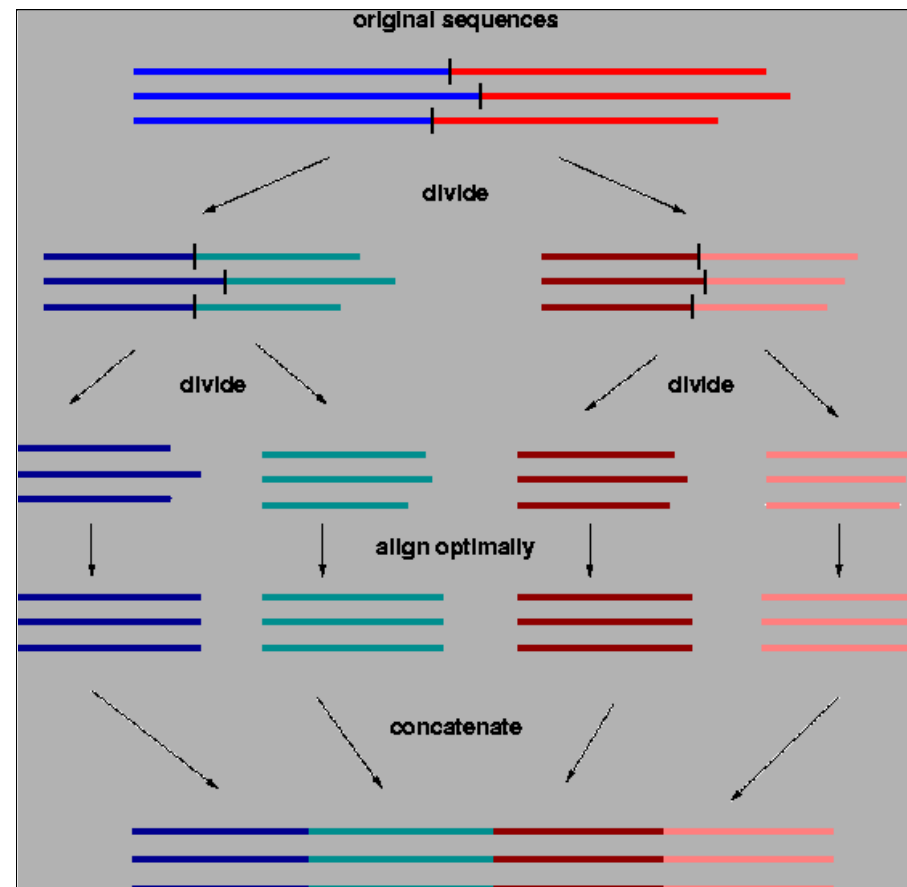
Note *Redundancy caused by highly correlated sequences is avoided*



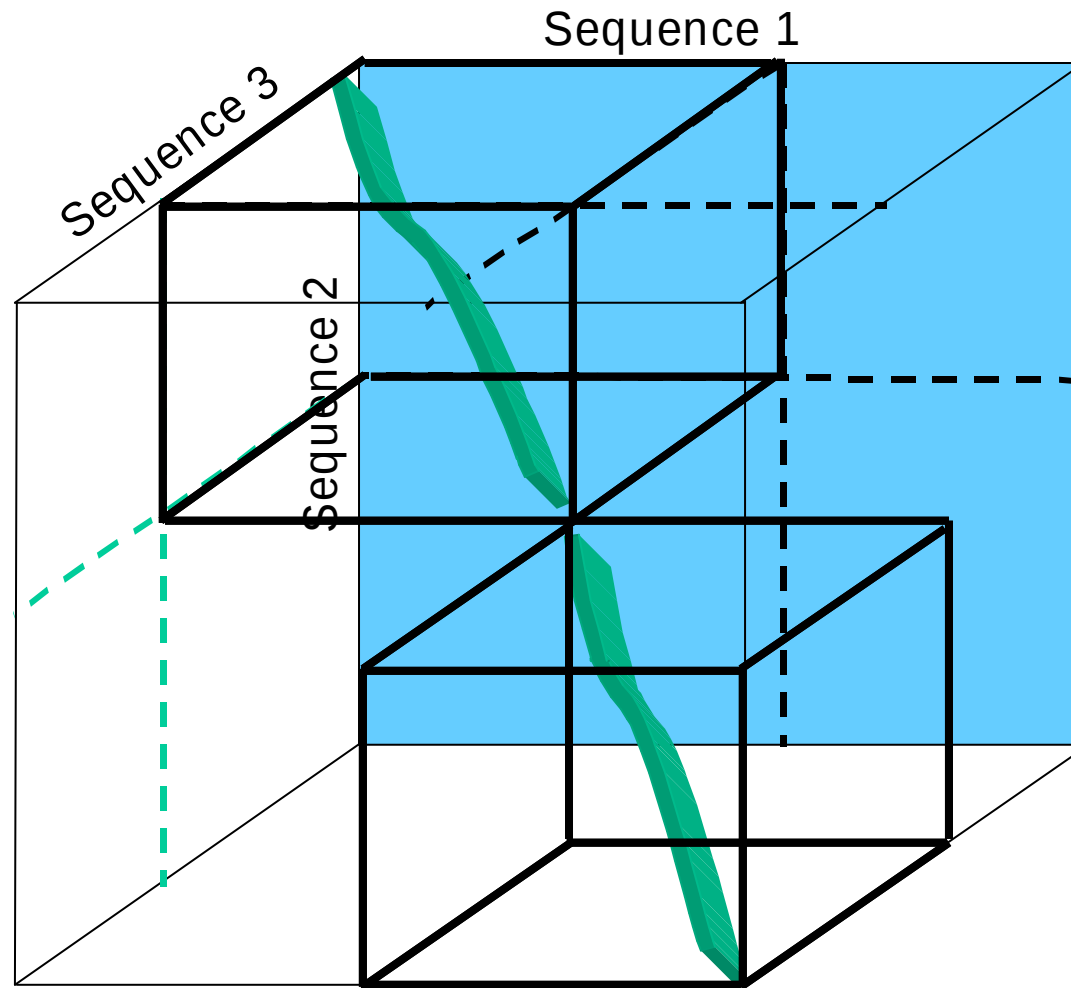
The DCA (Divide-and-Conquer) approach

Stoye et al. 1997

- n Each sequence is cut in two behind a suitable cut position somewhere close to its midpoint.
- n This way, the problem of aligning one family of (long) sequences is divided into the two problems of aligning two families of (shorter) sequences.
- n This procedure is re-iterated until the sequences are sufficiently short.
- n Optimal alignment by MSA.
- n Finally, the resulting short alignments are concatenated.



So in effect ...



Multiple alignment methods

q

q Progressive alignment

> incorporates phylogenetic information to guide the alignment process

q Iterative alignment

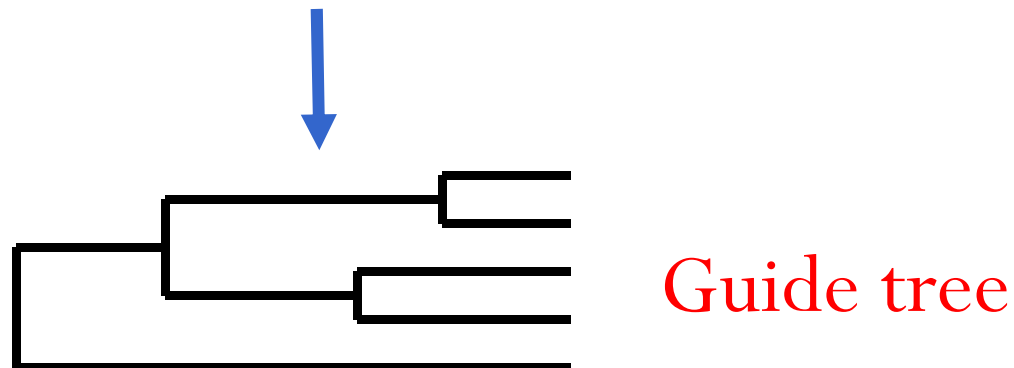
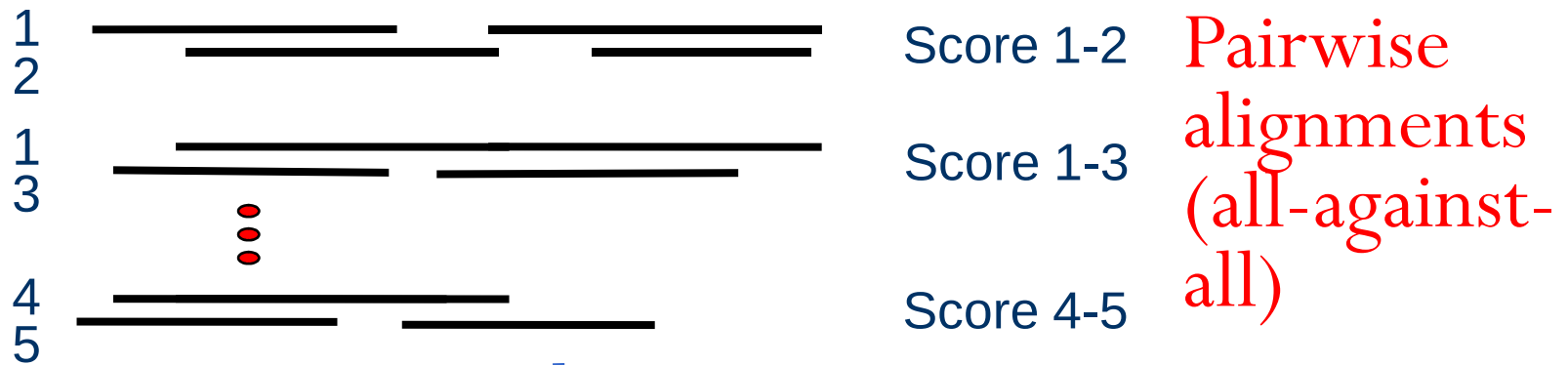
> correct for problems with progressive alignment by repeatedly realigning subgroups of sequence



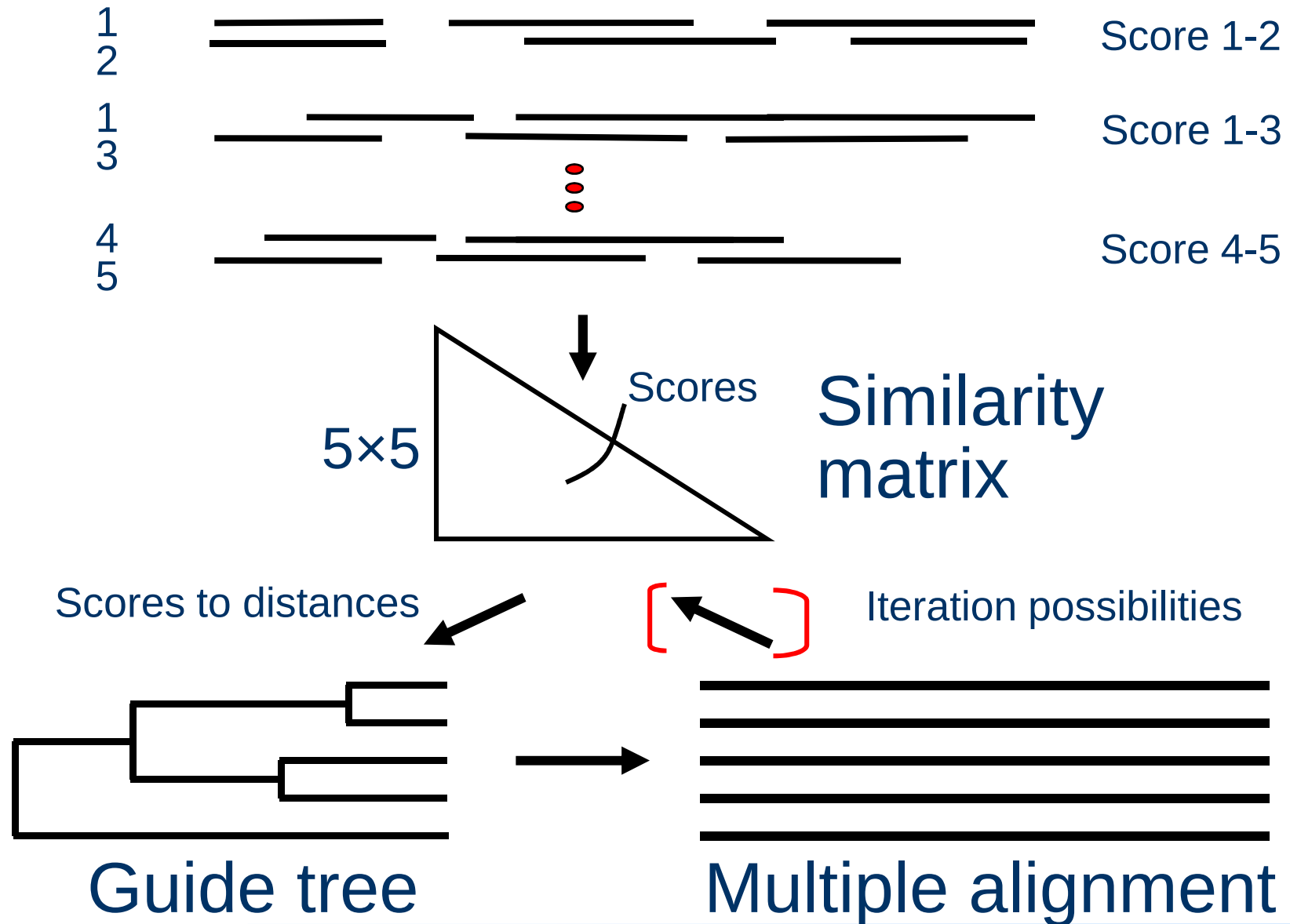
The progressive alignment method

- q Underlying idea: we are interested in aligning families of sequences that are evolutionary related
- q Principle: construct an approximate phylogenetic tree for the sequences to be aligned ('guide tree') and then build up the alignment by progressively adding sequences in the order specified by the tree.

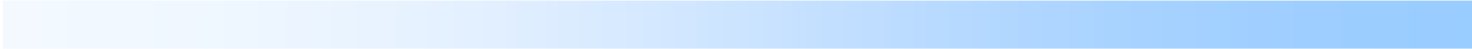
Making a guide tree



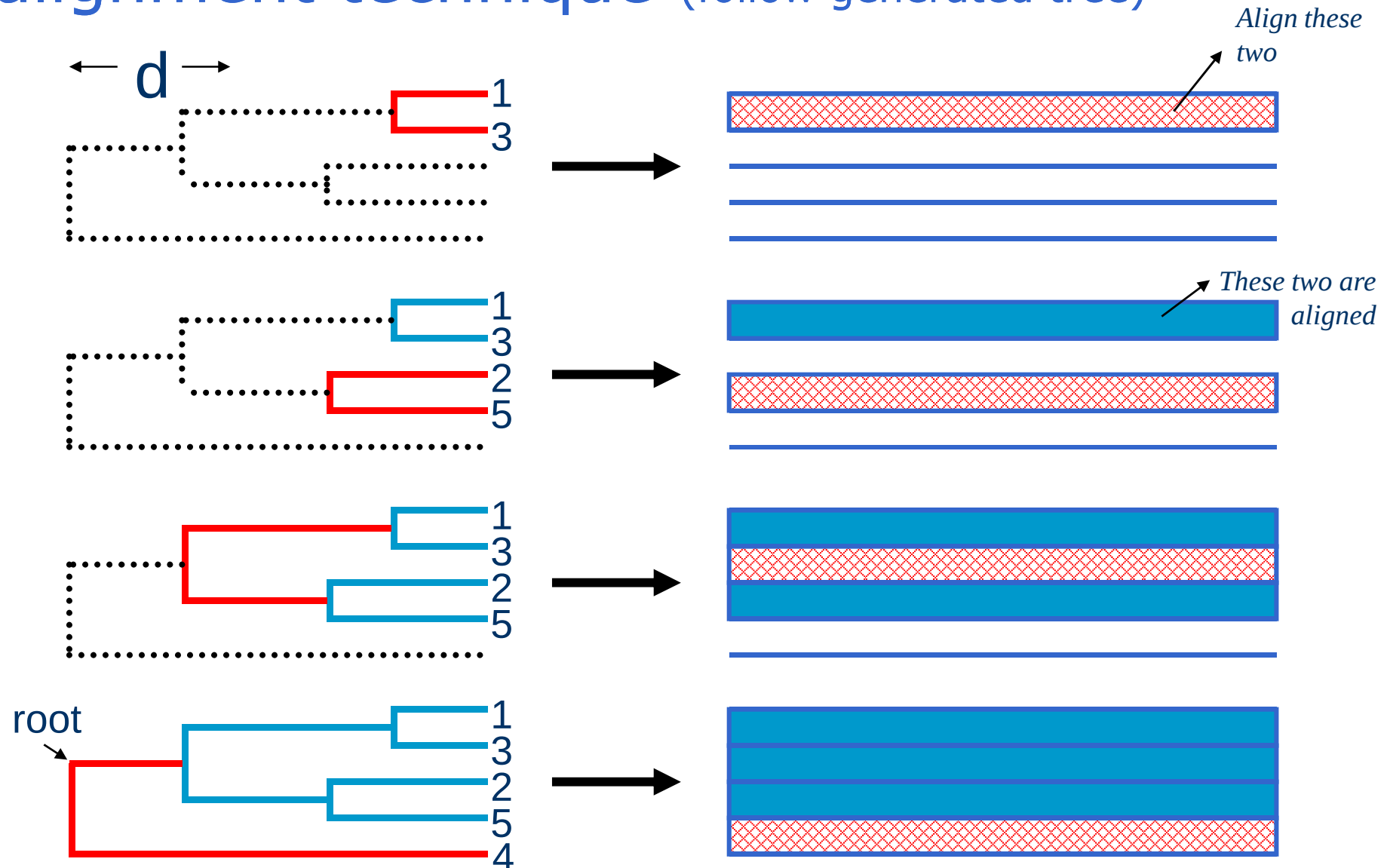
Progressive multiple alignment



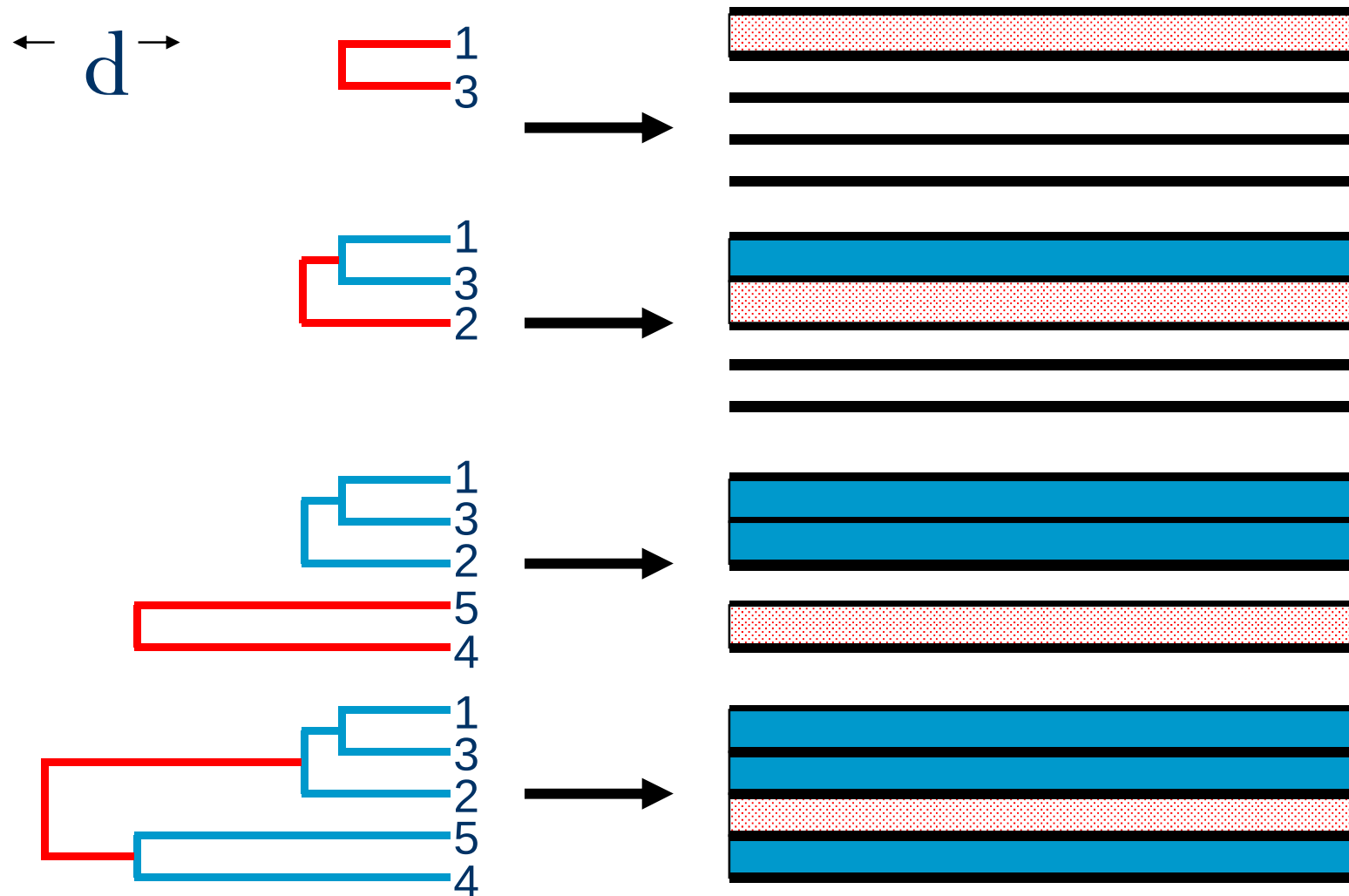
Progressive alignment strategy

1. Perform pair-wise alignments of all of the sequences (all against all; e.g. make $N(N-1)/2$ alignments)
 - Most methods use semi-global alignment
 2. Use the alignment scores to make a similarity (or distance) matrix
 3. Use that matrix to produce a guide tree
 4. Align the sequences successively, guided by the order and relationships indicated by the tree ($N-1$ alignment steps)
- 

General progressive multiple alignment technique (follow generated tree)

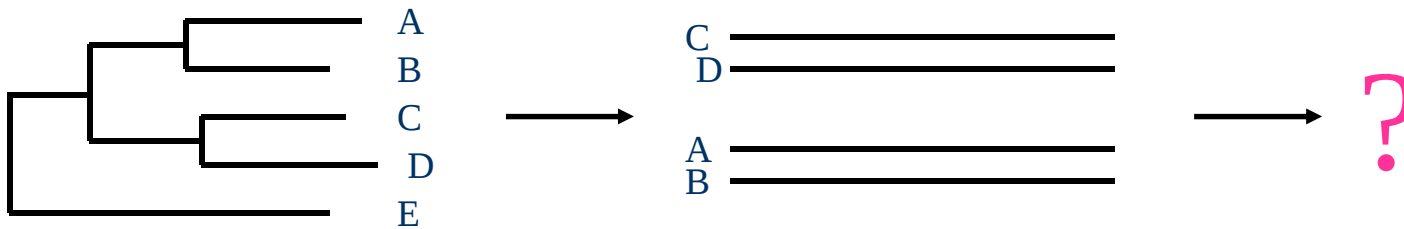


PRALINE progressive strategy



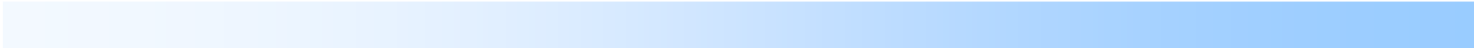
At each step, Praline checks which of the pair-wise alignments (sequence-sequence, sequence-profile, profile-profile) has the highest score – this one gets selected

But how can we align **blocks of sequences** ?



- n The dynamic programming algorithm performs well for pairwise alignment (two axes).
- n So we should try to treat the blocks as a “single” sequence ...

How to represent a block of sequences

- n **Historically: consensus sequence**
single sequence that best represents the amino acids observed at each alignment position.
 - n **Modern methods: alignment profile**
representation that retains the information about frequencies of amino acids observed at each alignment position.
- 

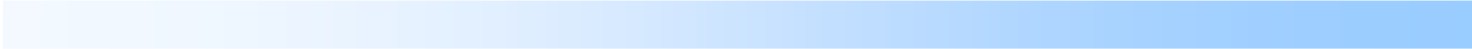
Consensus sequence

Sequence 1	F A T N M G T S D P P T H T R L R K L V S Q
Sequence 2	F V T N M N N S D G P T H T K L R K L V S T
Consensus	F * T N M * * S D * P T H T * L R K L V S *

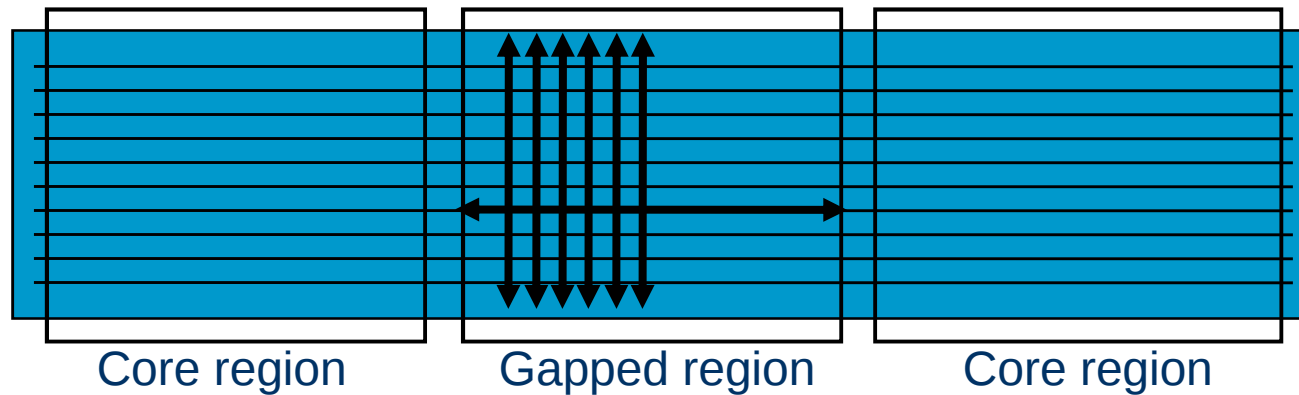
n **Problem:** loss of information

n **For larger blocks of sequences it
“punishes” more distant members**

Alignment profiles

- n **Advantage:** full representation of the sequence alignment (more information retained)
 - n Not only used in alignment methods, but also in sequence-database searching (to detect distant homologues)
 - n Also called PSSM in BLAST (*Position-specific scoring matrix*)
- 

Multiple alignment profiles



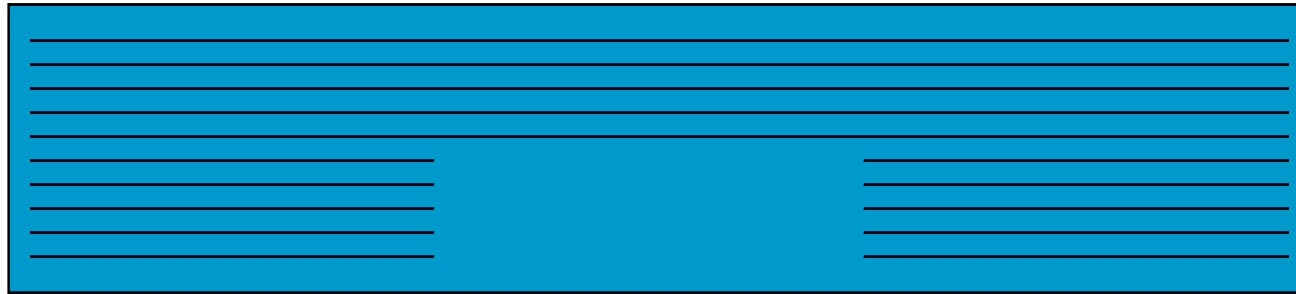
	i		
A	fA..	fA..	fA..
C	fC..	fC..	fC..
D	fD..	fD..	fD..
•	•	•	•
•	•	•	•
•	•	•	•
W	fW..	fW..	fW..
Y	fY..	fY..	fY..
-	Gapo, gapx	Gapo, gapx	Gapo, gapx

Position-dependent gap penalties

frequencies

Profile building

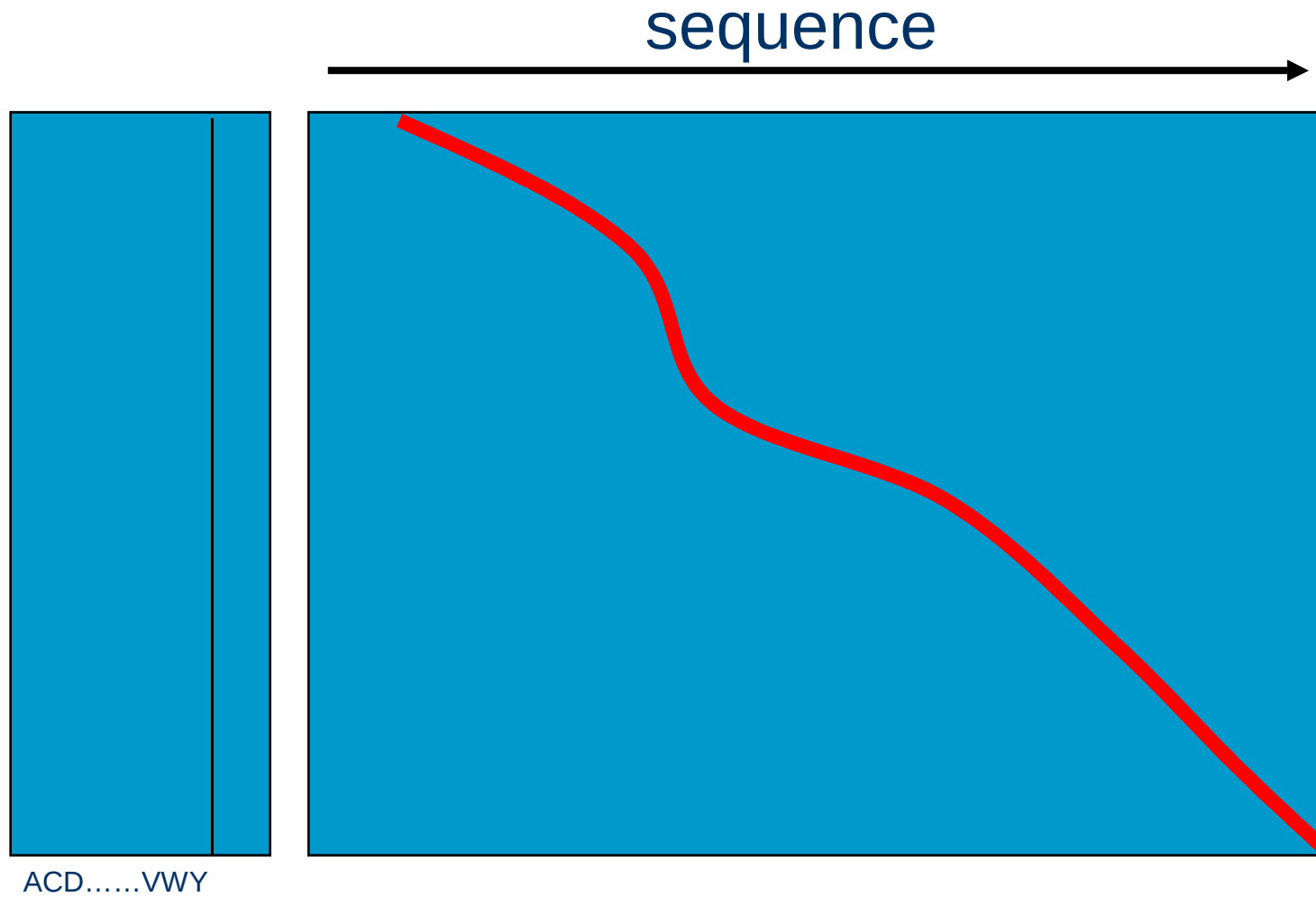
- q Example: each aa is represented as a frequency and gap penalties as weights.



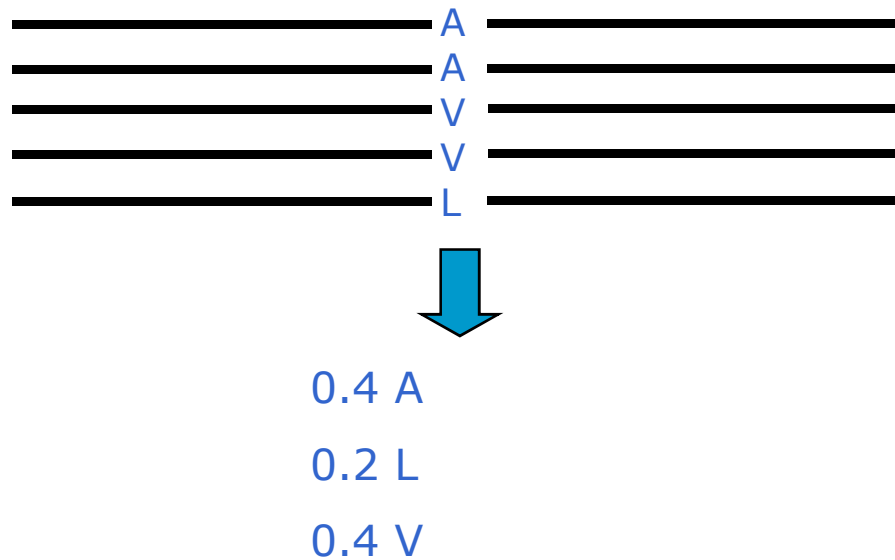
	<i>i</i>		
A	0.5	0.3	0
C	0	0.1	0.5
D	0	0	0.2
•	•	•	•
•	•	•	•
•	•	•	•
W	0	0.3	0.1
Y	0.5	0.3	0.2
Gap penalties	1.0	0.5	1.0

Position dependent gap penalties

Profile-sequence alignment



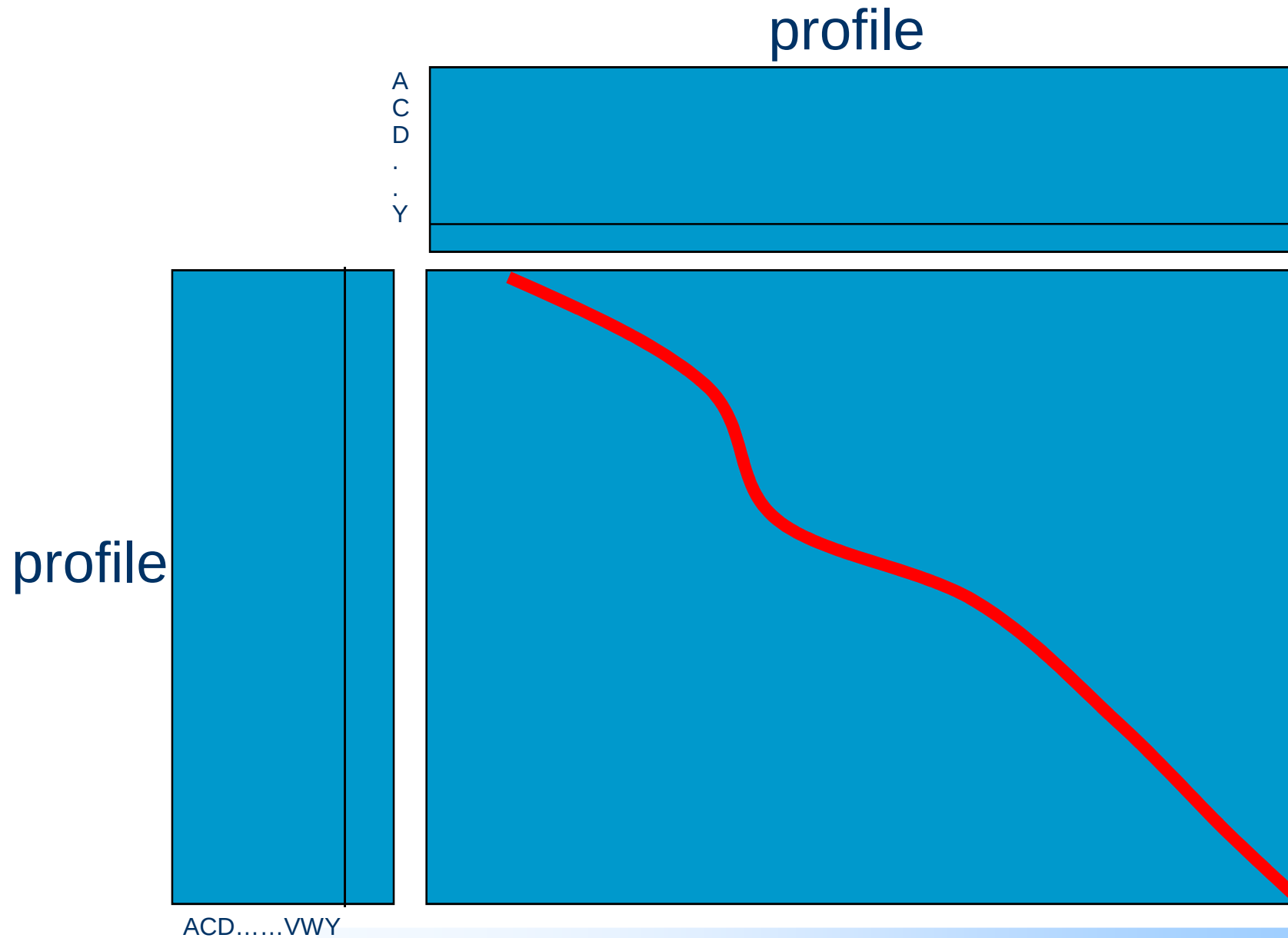
Sequence to profile alignment



Score of amino acid **L** in a sequence that is aligned against this profile position:

$$\text{Score} = 0.4 * s(\text{L}, \text{A}) + 0.2 * s(\text{L}, \text{L}) + 0.4 * s(\text{L}, \text{V})$$

Profile-profile alignment



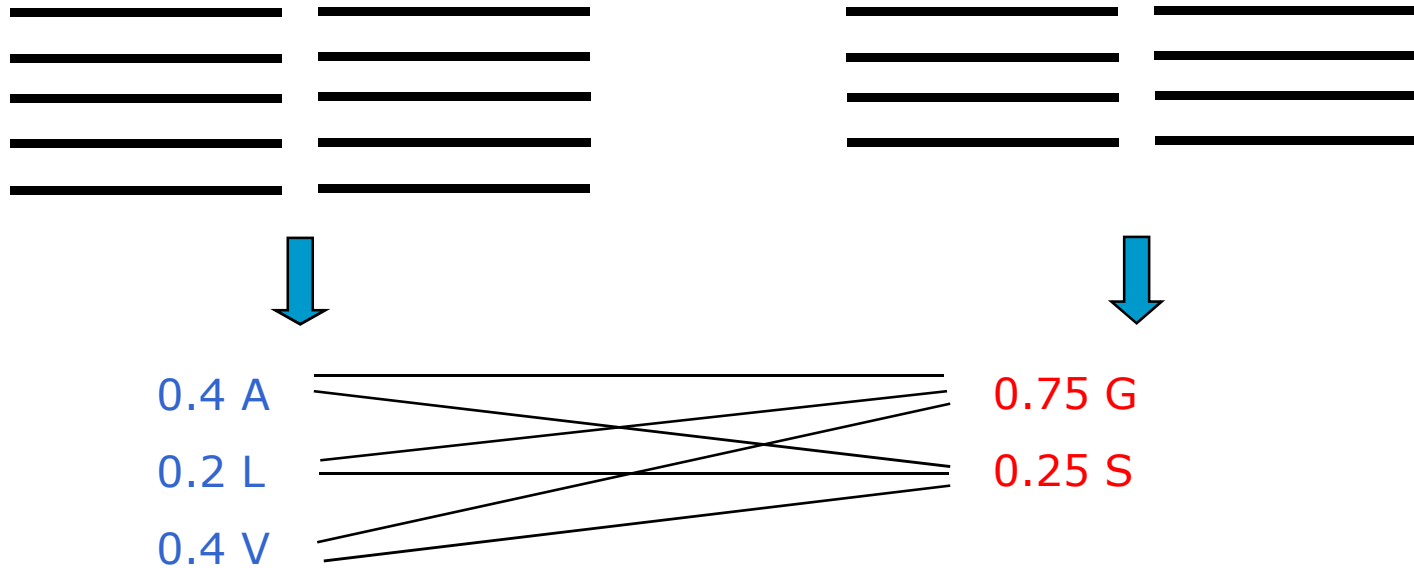
General function for profile-profile scoring



- q At each position (column) we have different residue frequencies for each amino acid (rows)
- q Instead of saying $S=s(aa_1, aa_2)$ for pairwise alignment
- q For comparing two profile positions we take:

$$S = \sum_i^{20} \sum_j^{20} faa_i \times faa_j \times s(aa_i, aa_j)$$

Profile to profile alignment



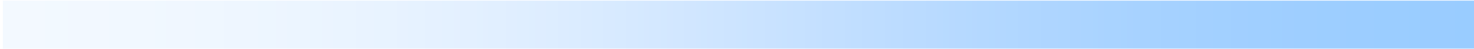
Match score of these two alignment columns using the a.a frequencies at the corresponding profile positions:

$$\begin{aligned} \text{Score} = & 0.4 * 0.75 * s(A, G) + 0.2 * 0.75 * s(L, G) + 0.4 * 0.75 * s(V, G) + \\ & + 0.4 * 0.25 * s(A, S) + 0.2 * 0.25 * s(L, S) + 0.4 * 0.25 * s(V, S) \end{aligned}$$

$s(x, y)$ is value in amino acid exchange matrix (e.g. PAM250, Blosum62) for amino acid pair (x, y)

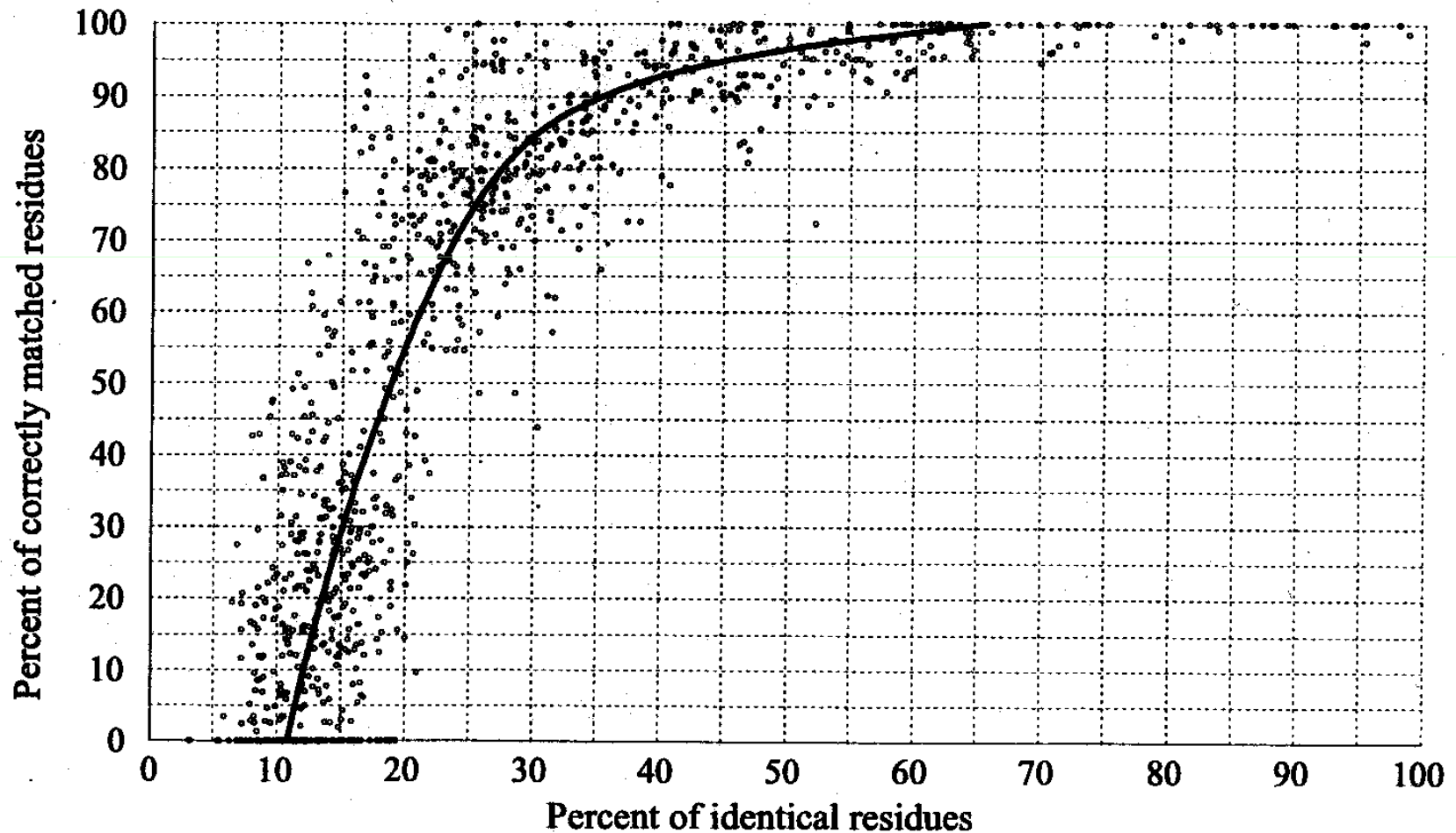
Progressive alignment strategy

Methods:

- n Biopat (Hogeweg and Hesper 1984 -- first integrated method ever)
 - n MULTAL (Taylor 1987)
 - n DIALIGN (1&2, Morgenstern 1996) – local MSA
 - n PRRP (Gotoh 1996)
 - n **ClustalW (Thompson et al 1994)**
 - n **PRALINE (Heringa 1999)**
 - n **T-Coffee (Notredame 2000)**
 - n **POA (Lee 2002)**
 - n **MUSCLE (Edgar 2004)**
 - n **PROBSCONS (Do, 2005)**
 - n **MAFFT**
- 

Pair-wise alignment quality *versus* sequence identity

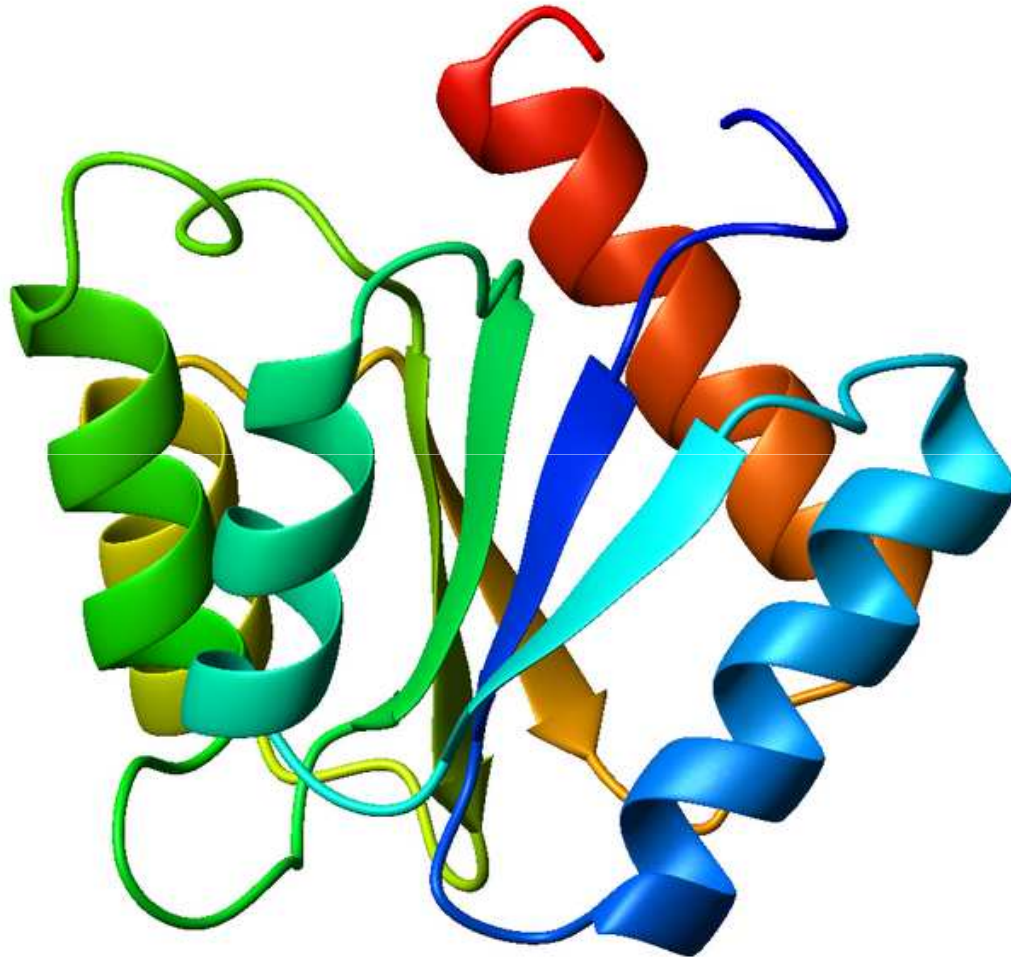
(Vogt et al., JMB 249, 816-831,1995)



Clustal, ClustalW, ClustalX

- n CLUSTAL W/X (Thompson *et al.*, 1994) uses Neighbour Joining (NJ) algorithm (Saitou and Nei, 1984), widely used in phylogenetic analysis, to construct a guide tree (see lecture on phylogenetic methods).
- n Sequence blocks are represented by profile, in which the individual sequences are additionally weighted according to the branch lengths in the NJ tree.
- n Further carefully crafted heuristics include:
 - (i) local gap penalties
 - (ii) automatic selection of the amino acid substitution matrix, (iii) automatic gap penalty adjustment
 - (iv) mechanism to delay alignment of sequences that appear to be distant at the time they are considered.
- n CLUSTAL (W/X) does not allow iteration (Hogeweg and Hesper, 1984; Corpet, 1988, Gotoh, 1996; Heringa, 1999, 2002)

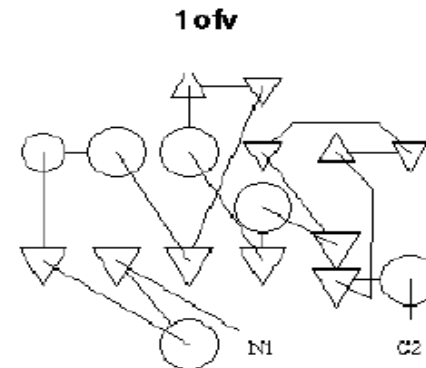
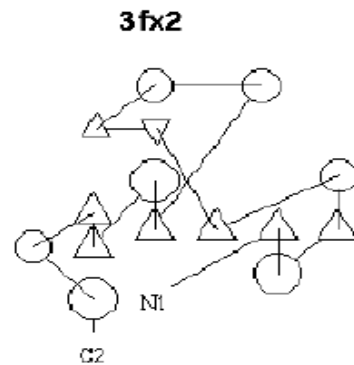
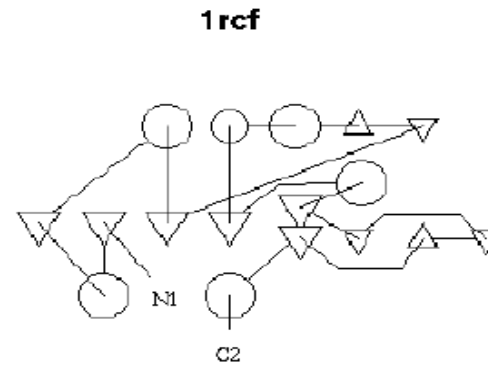
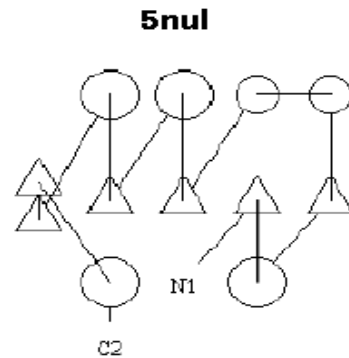
Aligning 13 Flavodoxins + cheY



Flavodoxin
fold: doubly
wound $\beta\alpha\beta$
structure

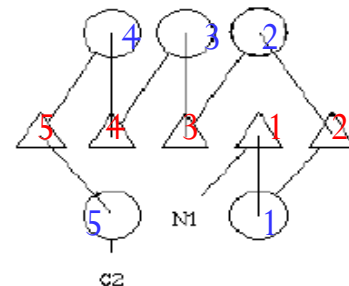
5($\beta\alpha$)

Flavodoxin family - TOPS diagrams



The basic topology of the flavodoxin fold is given below, the other four TOPS diagrams show flavodoxin folds with local insertions of secondary structure elements (David Gilbert)

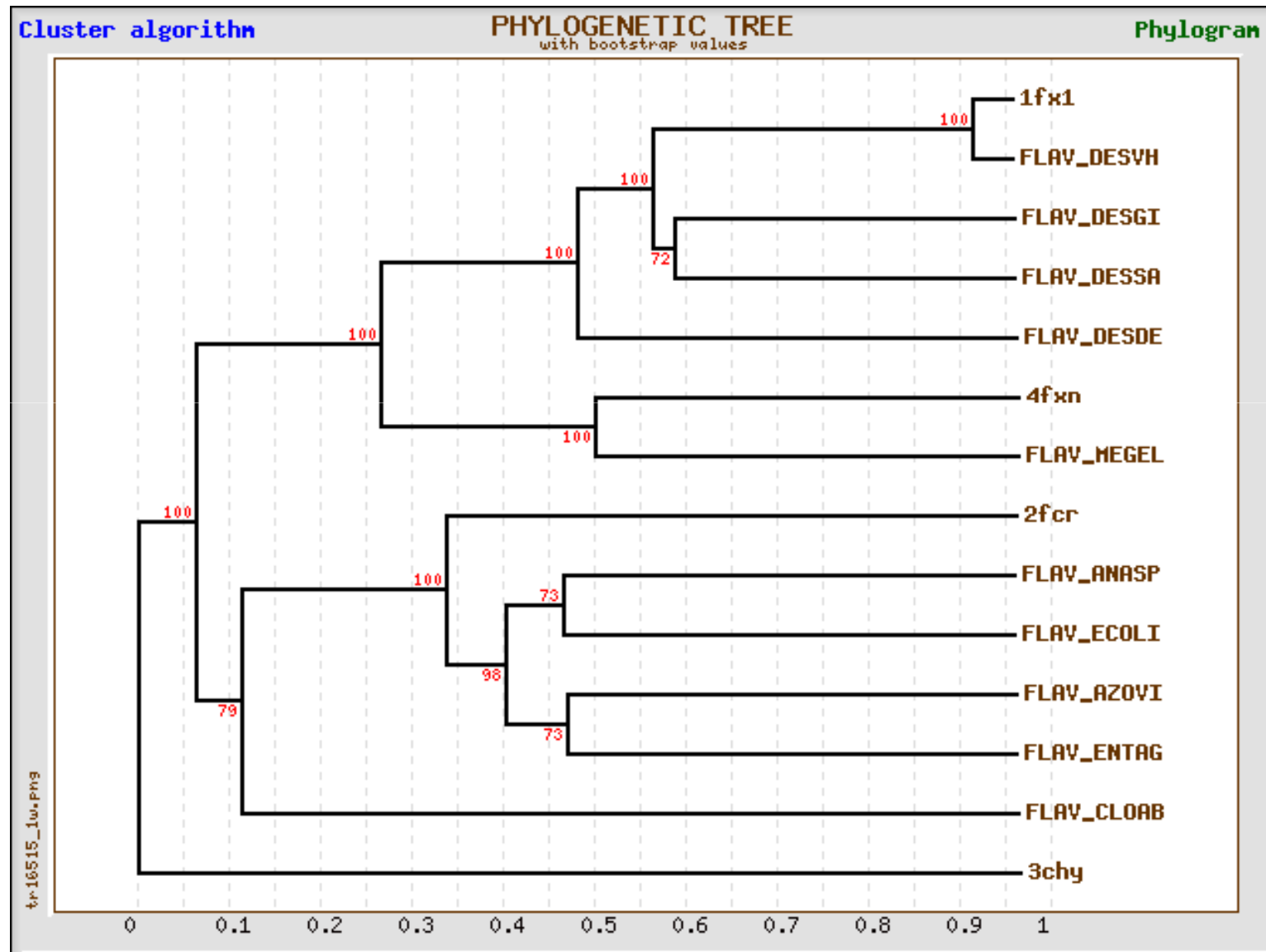
Basic Topology



○ α -helix

△ β -strand

Flavodoxin-cheY NJ tree



ClustalW web-interface

- Help Index
- General Help
- Formats
- Gaps
- Matrix
- References
- ClustalW Help
- ClustalW FAQ
- Jalview Help
- Scores Table
- Alignment
- Guide Tree
- Colours

- Similar Applications
 - Muscle
 - T-Coffee

ClustalW Submission Form

ClustalW is a general purpose multiple sequence alignment program for DNA or proteins. It produces biologically meaningful multiple sequence alignments of divergent sequences. It calculates the best match for the selected sequences, and lines them up so that the identities, similarities and differences can be seen. Evolutionary relationships can be seen via viewing Cladograms or Phylograms. [New users, please read the FAQ.](#)

>> [Download Software](#)   

YOUR EMAIL	ALIGNMENT TITLE	RESULTS	ALIGNMENT	CPU MODE
	Sequence	interactive	full	single
KTUP (WORD SIZE)	WINDOW LENGTH	SCORE TYPE	TOPDIAG	PAIRGAP
def	def	percent	def	def
MATRIX	GAP OPEN	END GAPS	GAP EXTENSION	GAP DISTANCES
def	def	def	def	def

OUTPUT		PHYLOGENETIC TREE		
OUTPUT FORMAT	OUTPUT ORDER	TREE TYPE	CORRECT DIST.	IGNORE GAPS
aln w/numbers	aligned	none	off	off

Enter or Paste a set of Sequences in any supported format:

Help

Upload a file:

Browse...

Run

Reset

If you plan to use these services during a course please contact us using the email below.
[Please read the FAQ](#) before seeking help from our support staff.

CLUSTAL X (1.64b) multiple sequence alignment Flavodoxin-cheY

```

1fx1      -PKALIVYGSTTGNTTEYTAETIARQLANAG-Y-EVDSRDAASVEAGGLFEGFDLVLLGCSTWGDDSI-----LQDDFIPLFD-SLEETGAQGRK
FLAV_DESVH MPKALIVYGSTTGNTTEYTAETIARELADAG-Y-EVDSRDAASVEAGGLFEGFDLVLLGCSTWGDDSI-----LQDDFIPLFD-SLEETGAQGRK
FLAV_DESGI MPKALIVYGSTTGNTEGVAEIAKTLNSEG-M-ETTVNVADVTPAGLAEGYDVVLLGCSTWGDDIE-----LQEDFVPLYE-DLDRAGLKDKK
FLAV_DESSA MSKSLIVYGSTTGNTETAEEYVAEAFENKE-I-DVELKNVTDVSVADLNGYDIVLFGCSTWGEIE-----LQDDFIPLYD-SLENADLKGGK
FLAV_DESDE MSKVLIVFGSSTGNTESIAQKLEELIAAGG-H-EVTLNNAADASAENLADGYDAVLFGCSAWGMEDLE-----MQDDFLSLFE-EFNRFGLAGRK
FLAV_CLOAB -MKISILYSSKTGKTERVAKLIEEGVKRSGNI-EVKTMNLDAVDKKFLQE-SEGIIFGTPTYAN-----ISWEMKKWID-ESSEFNLEGKL
FLAV_MEGEL --MVEIVYWSGTGNTEAMANEIEAAVKAAG-A-DVESRFEDTNVDDVAS-KDVILLGCPAMGSE--E-----LEDSVVEPFF-TDLAPKLKGGK
4fxn      --MKIVYWSGTGNTEAMANEIEAAVKAAG-A-DVNTINVSDVNIDELLN-EDILILGCSAMGDE--V-----LEESEFEPFI-EEISTKISGK
FLAV_ANASP SKKIGLFYGTQTGKTESVAEIIRDEFGNDVVT---LHDVSQAEVTDLND-YQYLIIGCPTWNIGELQ---SD-----WEGLYS-ELDDVDFNGKL
FLAV_AZOVI -AKIGLFFGSNTGKTRKVAKSIKKRFDDETMSD---ALNVNRVSAEDFAQ-YQFLILGTPTLGEGPLGLSSDCENESWEEFLP-KIEGLDFSGKT
2fcr      --KIGIFFSTSTGNTTEVADFIGKTLGAKADAP---IDVDDVTDPQALKD-YDLLFLGAPTWNTGADTERSGT---SWDEFLYDKLPEVDMKDL
FLAV_ENTAG MATIGIFFGSDTGQTRKVAKLIHQKLDGIADAP---LDVRRATREQFLS--YPVLLLGTPTLGDGELPGVEAGSQYDSWQEFTN-TLSEADLTGKT
FLAV_ECOLI -AITGIFFGSDTGNTENIAKMIQKQLGKDVAD---VHDIAKSSKEDLEA-YDILLGIPTWYYGEAQ-CD-----WDDFFP-TLEEIDFNGKL
3chy      --ADKELKFLVVDDFSTMRRIVRNLKELG---FNNVEEAEDGVDALN-----KLQAGGYGFV--I-----SDWNMPNMDG-LELLKTIR--
          .      . . .      :      .      .      :

1fx1      VACFGCGDSSYEYF--CGAVDAIEEKLKNLGAEVQDG-----LRIDGDPRAARDDIVGWAHDVRGAI-----
FLAV_DESVH VACFGCGDSSYEYF--CGAVDAIEEKLKNLGAEVQDG-----LRIDGDPRAARDDIVGWAHDVRGAI-----
FLAV_DESGI VGVFGCGDSSYTYF--CGAVDVIEKKAELGATLVASS-----LKIDGEPDSAE--VLDWAREVLARV-----
FLAV_DESSA VSVFGCGDSDYTYF--CGAVDAIEEKLKMGAVVIGDS-----LKIDGDPERDE--IVSWGSGIADKI-----
FLAV_DESDE VAAFASGDQEYEHF--CGAVPAIEERAKELGATIIAEG-----LKMEGDASNDPEAVASFAEDVLKQL-----
FLAV_CLOAB GAAFSTANSIAGGS--DIALLTILNHLMVKGMLVYSGGVA---FGKPKTHLGYVHINEIQENEDENARIFGERIANKVQKIF-----
FLAV_MEGEL VGLFGSYGWGSGE---WMDAWKQRTEDTGATVIGTA-----IVN-EMPDNAPECKE-LGEAAKA-----
4fxn      VALFGSYGWGDGK---WMRDFEERMNGYGCVVETP-----LIVQNEPDEAEQDCIEFGKKIANI-----
FLAV_ANASP VAYFGTGDQIGYADNFQDAIGILEEKISQRGGKTGVYWSTDGYDFNDSKALR-NGKFVGLALDEDNQSDLTDDRIKSWVAQLKSEFGL-----
FLAV_AZOVI VALFGLGDQVGYPENYLDALGELYSFFKDRGAKIVGSWSTDGYEFESSEAVV-DGKFVGLALDLNQSGKTDERVAAWLAQIAPEFGLSL----
2fcr      VAIFGLGDAEGYPDNFCDAIEEIHDCFAKQGAKPVGFNSPDDYDYEESKSVR-DGKFLGLPLDMVNDQIPMEKRVAGWVEAVVSETGV-----
FLAV_ENTAG VALFGLGDQLNYSKNFVSAMRILYDLVIARGACVVGWNPREGYKFSFSAALLENNEFVGLPLDQENQYDLTEERIDSWLEKLKPAVL-----
FLAV_ECOLI VALFGCGDQEDYAEYFCDALGTIRDIIEPRGATIVGHWPTAGYHFEASKGLADDDHFVGLAIDEDRQPELTAERVEKWKQISEELHLDEILNA
3chy      AD--GAMSALPVL---MVTAEAKKENIIAAQAGAS-----GYV-VKPFTAATLEEKLNKIFEKLG-----
          .      .      :      .      .

```

The secondary structures of 4 sequences are known and can be used to assess the alignment (red is β -strand, blue is α -helix)

There are problems ...

Accuracy is very important !!!!

- q Progressive multiple alignment is a greedy strategy: Alignment errors during the construction of the MSA **cannot be repaired anymore and** these errors are propagated into later progressive steps.
- q Comparisons of sequences at early steps during progressive alignment cannot make use of information from other sequences.
- q It is only later during the alignment progression that more information from other sequences (e.g. through profile representation) becomes employed in the alignment steps.



Progressive multiple alignment

“Once a gap, always a gap”

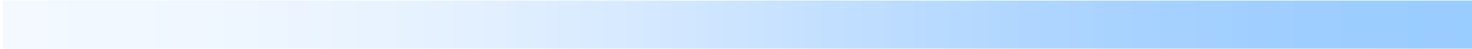
Feng & Doolittle, 1987



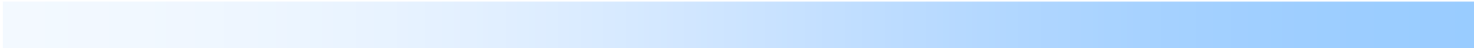
Additional strategies for multiple sequence alignment

- **Matrix extension (T-coffee)**
- Profile pre-processing (Praline)
- Secondary structure-induced alignment

Objective: try to avoid (early) errors



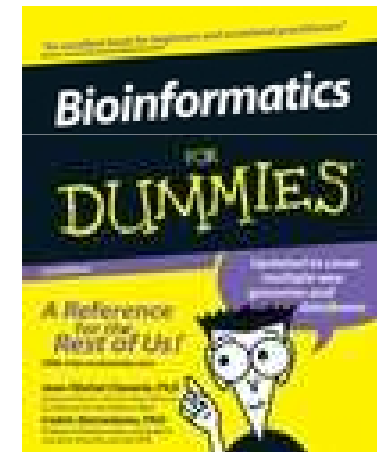
Integrating alignment methods and alignment information with **T-Coffee**

- Integrating different pair-wise alignment techniques (NW, SW, ..)
 - Combining different multiple alignment methods (consensus multiple alignment)
 - Combining sequence alignment methods with structural alignment techniques
 - Plug in user knowledge
- 

Matrix extension

T-Coffee

*Tree-based Consistency Objective Function
For alignment Evaluation*

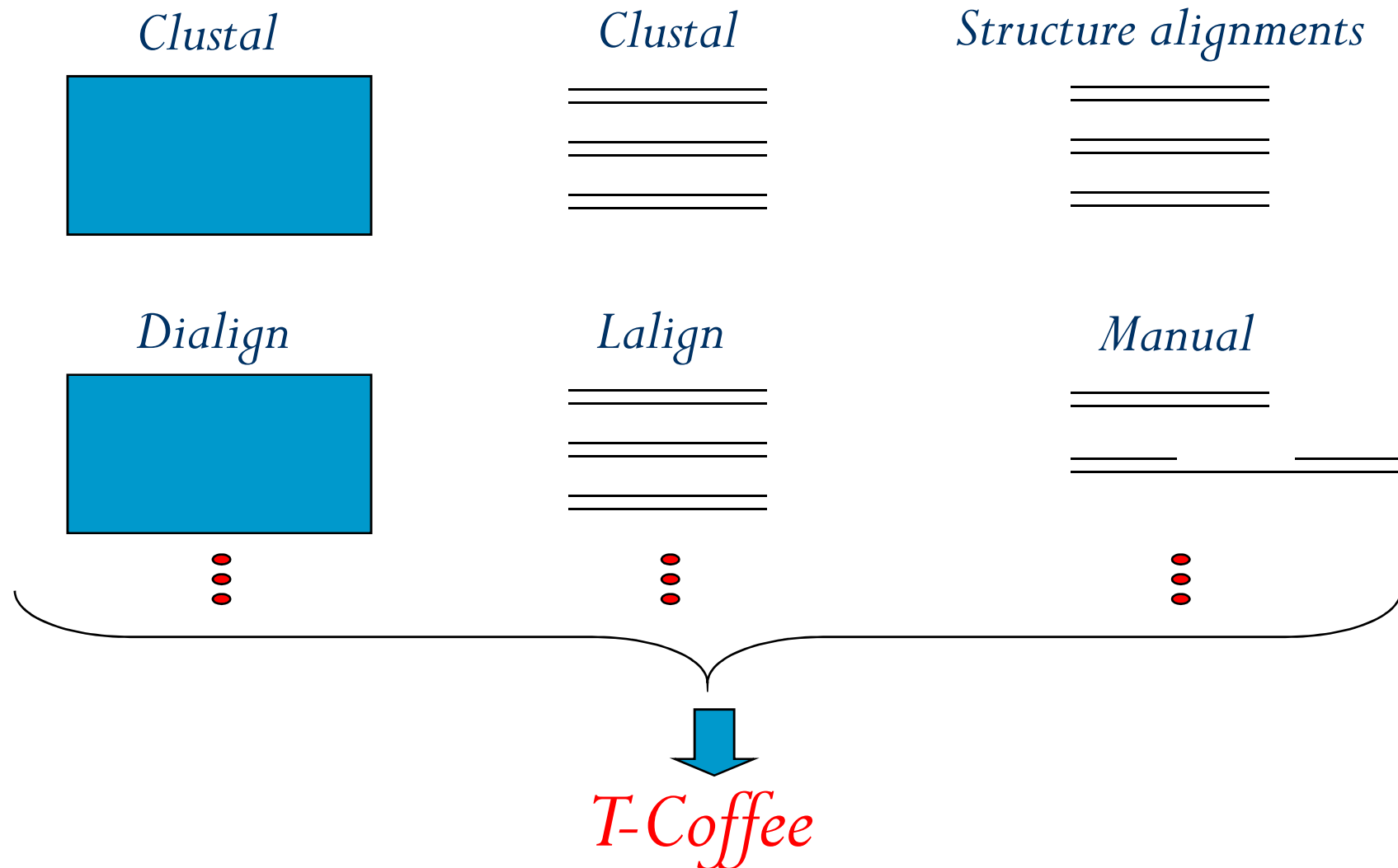


Cedric Notredame (“*Bioinformatics for dummies*”)

Des Higgins

Jaap Heringa *J. Mol. Biol.*, 302, 205-217;2000

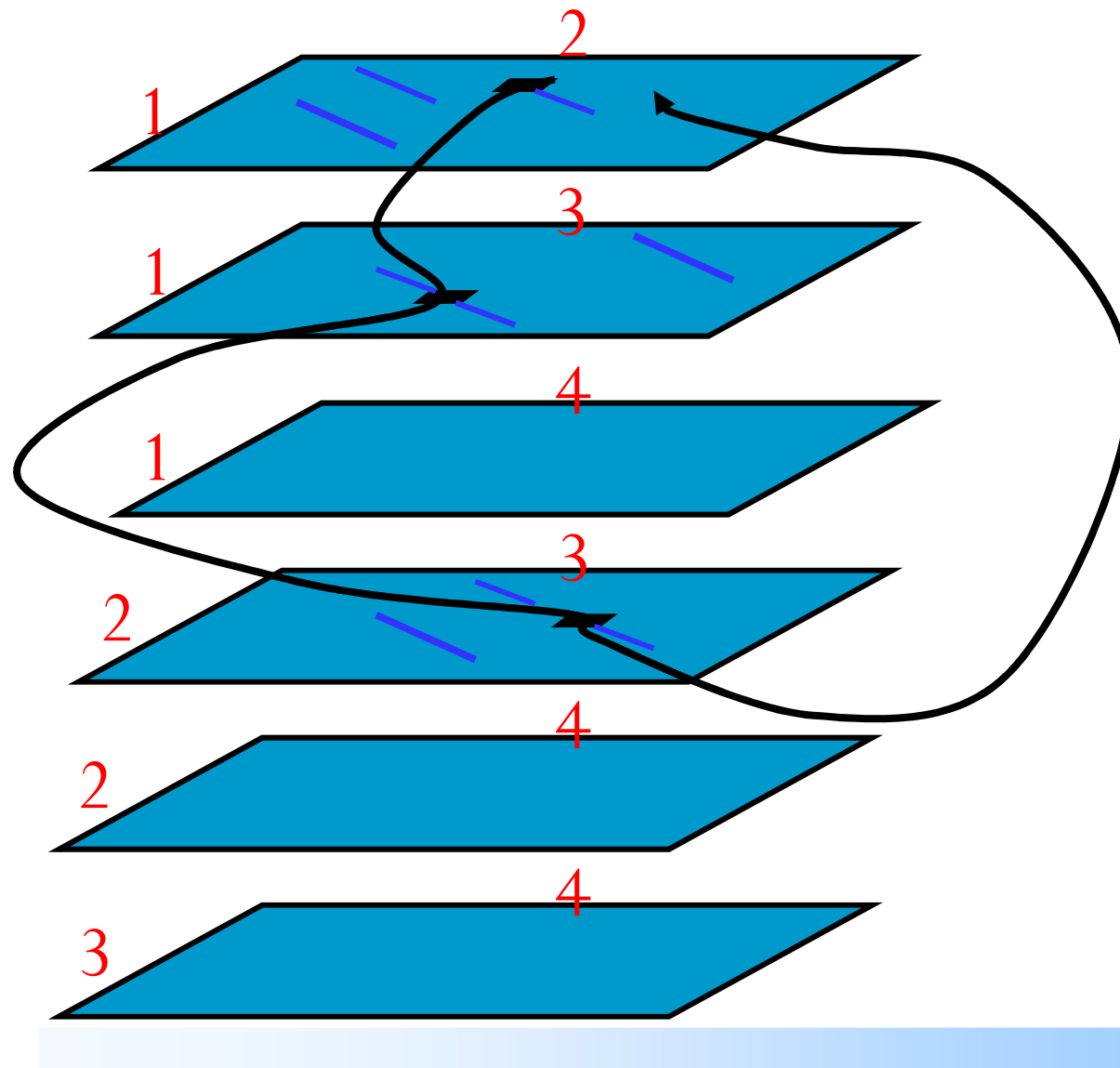
Using different sources of alignment information



T-Coffee library system

<i>Seq1</i>	<i>AA1</i>	<i>Seq2</i>	<i>AA2</i>	<i>Weight</i>
		⋮		
3	V31	5	L33	10
3	V31	6	L34	14
		⋮		
5	L33	6	R35	21
5	I33	6	I36	35
		⋮		

Matrix extension



Search matrix extension – alignment transitivity

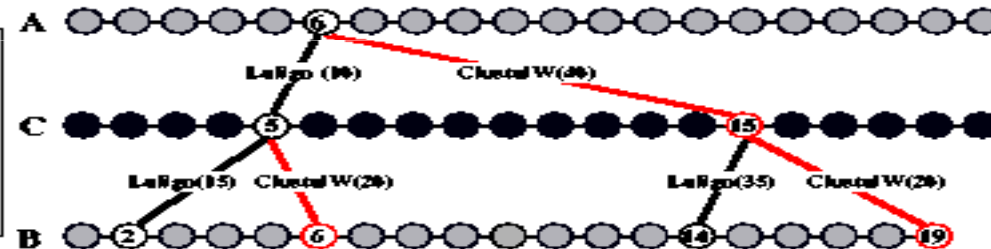
a) Signal Addition: Computation of the primary weights.

Primary Library		
Res. 1	Res. 2	Weight
A(6)	B(3)	30
A(6)	B(14)	25



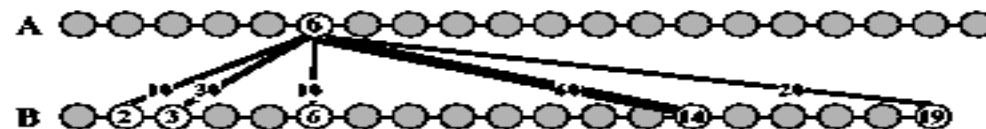
b) Extension: Alignment of A and B through sequence C.

Res. 1	Res. 2	Weight
A(6)	B(2)	$10 = \min(10, 15)$
A(6)	B(6)	$10 = \min(10, 20)$
A(6)	B(14)	$35 = \min(40, 35)$
A(6)	B(19)	$20 = \min(40, 20)$

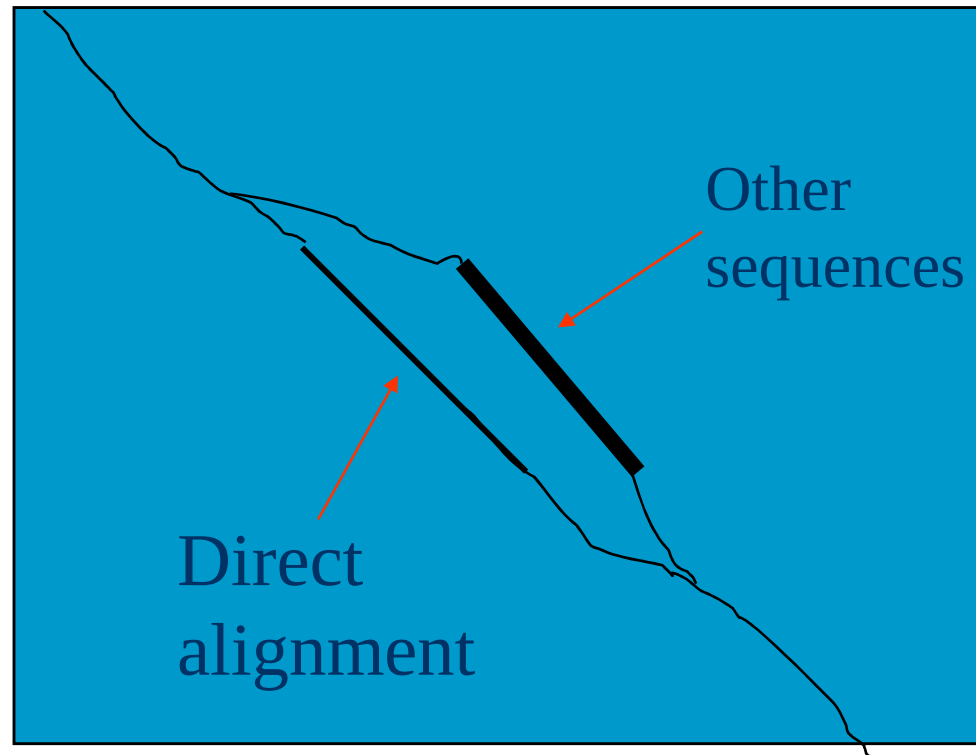


c) Combination: Position Specific Scoring Scheme for A and B using information from sequence A, B and C.

Extended Library		
Res. 1	Res. 2	Weight
A(6)	B(2)	10
A(6)	B(3)	30
A(6)	B(6)	10
A(6)	B(14)	$60 = 25 + 35$
A(6)	B(19)	20

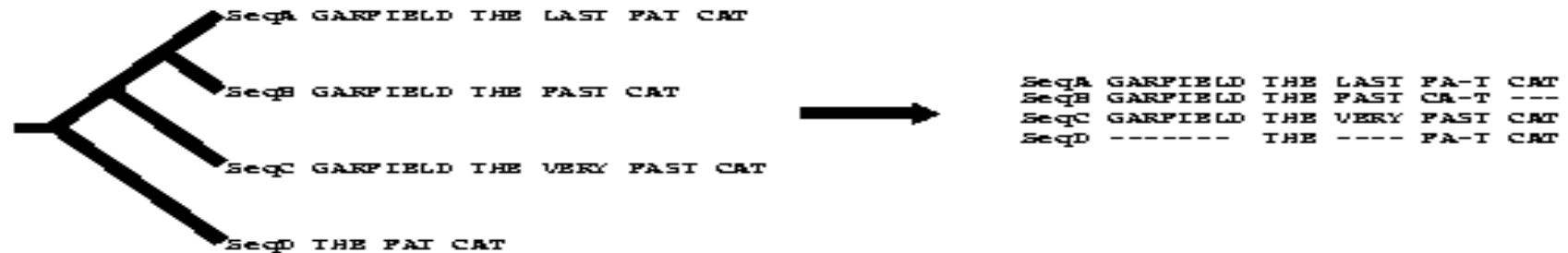


T-Coffee



Search matrix extension

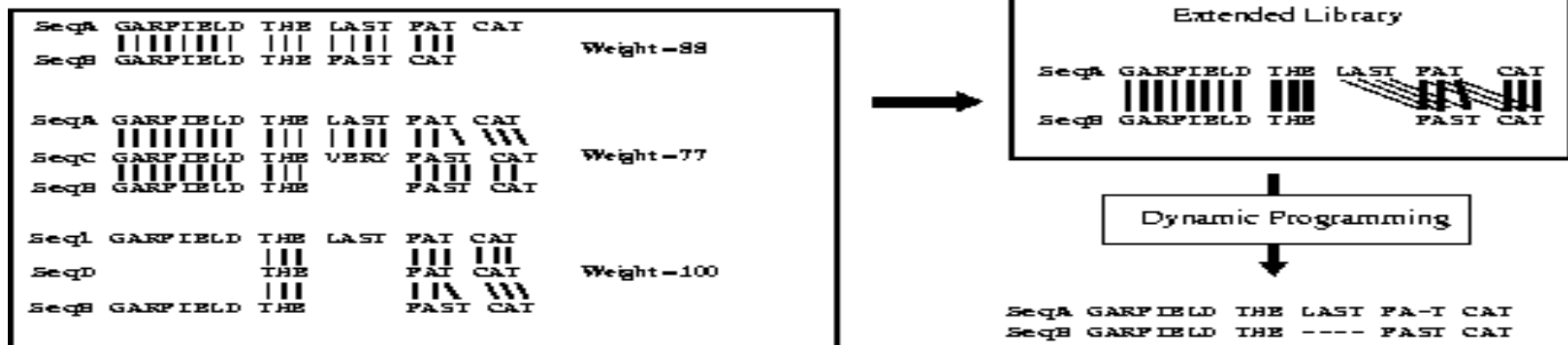
a) Regular Progressive Alignment Strategy



b) Primary Library

SeqA GARFIELD THE LAST PAI CAT	Prim. Weight = 88	SeqB GARFIELD THE ---- FAST CAT	Prim. Weight = 100
SeqB GARFIELD THE FAST CAT ---		SeqC GARFIELD THE VERY FAST CAT	
SeqA GARFIELD THE LAST PA-T CAT	Prim. Weight = 77	SeqB GARFIELD THE FAST CAT	Prim. Weight = 100
SeqC GARFIELD THE VERY FAST CAT		SeqD ----- THE PA-T CAT	
SeqA GARFIELD THE LAST PAI CAT	Prim. Weight = 100	SeqC GARFIELD THE VERY FAST CAT	Prim. Weight = 100
SeqD ----- THE ---- PAI CAT		SeqD ----- THE ---- PA-T CAT	

c) Extended Library for seq1 and seq2



T-COFFEE web-interface

[HOME](#) [references](#) [help](#) 

TCOFFEE::Advanced

Computes a **multiple sequence alignment** and the associated **phylogenetic tree**.

Limitations: Maximum number of sequences is **50**
Maximum length of sequences is **2000**

Sequence Input

Paste or upload your set of sequences in [fasta](#) format.

[Upload File](#)

or paste data

Alignment Computation

Tooffee computes its alignments by combining a collection of smaller alignments named a Library. In this section you can select the methods you want to combine into the Tcoffee library. You can choose pairwise and multiple sequence alignment methods.

[Pairwise Methods](#)

☒ Mlaligr_id_pair ☐ Mclustaln_pair
☐ Mfast_pair
☒ Mslow_pair

Output

Use this section to control the Output Format

[Alignmen: Format](#)

☒ clustaln_aln ☐ pir_aln ☐ pir_seq ☒ score_pdf
☐ gog ☒ fasta_aln ☐ fasta_seq ☐ score_ascii
☐ msf_aln ☒ phytip ☒ score_html

[Case](#)

[Residue Number](#)

[Order](#)

You may paste your e-mail address:

3D-COFFEE

- n Computes structural alignments
- n Structures associated with the sequences are retrieved and the information is used to optimise the MSA
- n More accurate ... but for many proteins we do not have a structure

The screenshot shows the 'EXPRESSO(3DCoffee)::Advanced' web interface. At the top, there are links for 'HOME', 'references', and 'help'. Below the title, a text box explains that the server computes 'Structure' based Multiple Sequence Alignments and that it will automatically fetch related structures. A note states: 'Structure based sequence alignments are potentially more accurate than simple sequence alignments.' Limitations are listed: 'Maximum number of sequences is 50' and 'Maximum length of sequences is 2000'.

The interface is divided into three main sections:

- Sequence Input:** Contains a section for '1-Prepare your set of sequences in a fasta file.' with a 'Browse...' button and a text area for 'or paste data'.
- Structure Input:** Contains a section for '2-Upload your PDB structure if needed (Do NOT upload Confidential Data!)'. It has three 'Browse...' buttons for 'PDB File'.
- Library Computation:** Contains a section for '3-Select the methods you want to use to compute the library'. It lists various methods with checkboxes: 'Mishra_id_pair' (checked), 'Mishra_var' (unchecked), 'Mishra_pair' (unchecked), 'Mishra_var' (unchecked), 'Mishra_pair' (checked), and 'Mishra_var' (checked).

but....

T-COFFEE (V1.23) multiple sequence alignment Flavodoxin-cheY

```
1fx1      ----PKALIVYGSTTGNTTEYTAETIARQLANAG-YEVDSDRAASVE-AGGLFEGFDLVLLGCSTWGDDSE-----LQDDFIPL-FDSLEETGAQGRK-----
FLAV_DESVH ---MPKALIVYGSTTGNTTEYTAETIARELADAG-YEVDSDRAASVE-AGGLFEGFDLVLLGCSTWGDDSE-----LQDDFIPL-FDSLEETGAQGRK-----
FLAV_DESGI ---MPKALIVYGSTTGNTTEGVAEIAKTLNSEG-METTVVNVADVT-APGLAEGYDVLLGCSTWGDDSE-----LQEDFVPL-YEDLDRAGLKDKK-----
FLAV_DESSA ---MSKSLIVYGSTTGNTTETAEEYVAEAFENKE-IDVELKNVTDVS-VADLNGYDIVLFGCSTWGDEEIE-----LQDDFIPL-YDSLENADLKGGK-----
FLAV_DESDE ---MSKVLIVFGSSTGNTESIAQKLEELIAAGG-HEVTLNNAADAS-AENLADGYDAVLFGCSAWGMEDLE-----MQDDFSLF-FEEFNRFGLAGRK-----
4fxn      -----MKIVYWSGTGNTKMAELIAKGIIESG-KDVNTINVSDVN-IDELL-NEDILILGCSAMGDEVLE-----ESEFEPF-IEEIS-TKISGKK-----
FLAV_MEGEL -----MVEIVYWSGTGNTTEAMANEIEAAVKAAG-ADVSVRFEDTN-VDDVA-SKDVIILLGCPAMGSEEE-----DSVVEPF-FTDLA-PKLGKGG-----
FLAV_CLOAB -----MKISILYSSKTGKTERVAKLIEEGVKRSGNIEVKTMNLDAVD-KKFLQ-ESEGIIFGTPTYAN-----ISWEMKKW-IDESSEFNLEGL-----
2fcr      -----KIGIFFSTSTGNTTEVADFIGTLGAKA---DAPIDVDDVTPQAL-KDYDLLFLGAPTWNATGA---DTERSGTSDWDEFLYDKLPEVDMKDL-----
FLAV_ENTAG ---MATIGIFFGSDTGQTRKVAKLHQLDGLA---DAPLDVRRAT-REQF-LSYPVLLLTGPTLGDGELPGVEAGSQYDSWQEF-TNTLSEADLTGKT-----
FLAV_ANASP ---SKKIGLFYGTQGTGKTESVAEIRDEFNDV---VTLHDVSQAE-VTDL-NDYQYLIIGCPTWNIGEL-----QSDWEGL-YSELDDVDVFNGKL-----
FLAV_AZOVI ---AKIGLFFGSNTGKTRKVAKSIKKRFDET-M-SDALNVNRVS-AEDF-AQYQFLILGTPTLGEGLPGLSSDCENESWEEF-LPKIEGLDFSGKT-----
FLAV_ECOLI ---AITGIFFGSDTGNTENIAKMIQKQLGKDV---ADVHDIKSS-KEDL-EAYDILLGIPTWYYGEA-----QCDWDDF-FPTLEEIDFNGKL-----
3chy      ADKELKFLVVD--DFSTMRRIVRNLLKELGFN-NVE-EAEDGVDALNKLQ-AGGYGFVISDWNMPNMDGLE-----LLKTIRADGAMSALPVLNV
          ::      .      .      :      .      ::
```

```
1fx1      -----VACFGCGDSS--YEYFCGA-VDAIEEKLKNLGAEIVQDG-----LRIDGDPRAA--RDDIVGWAHDVIRGAI-----
FLAV_DESVH -----VACFGCGDSS--YEYFCGA-VDAIEEKLKNLGAEIVQDG-----LRIDGDPRAA--RDDIVGWAHDVIRGAI-----
FLAV_DESGI -----VGVFGCGDSS--YTYFCGA-VDVIEKKAELGATLVASS-----LKIDGEPDSA---EVLWAREVLARV-----
FLAV_DESSA -----VSVFGCGDSD--YTYFCGA-VDAIEEKLKMGAVVIGDS-----LKIDGDPE---RDEIVSWGSGIADKI-----
FLAV_DESDE -----VAAFASGDQE--YEHFCGA-VPAIEERAKELGATIIAEG-----LKMEGDASND--PEAVASFAEDVLKQL-----
4fxn      -----VALFGS-----YGWGDGKWMRDFEERMNGYGCVVVETP-----LIVQNEPD--EAQDCIEFGKKIANI-----
FLAV_MEGEL -----VGLFGS-----YGWGSGEWMDAWKQRTEDTGATVIGTA-----IV--NEMP--DNAPECKELGEAAKA-----
FLAV_CLOAB -----GAAFSTANSI--AGGS DIA-LLTILNHLMVKGMLVY---SGGVAFGPKPKTHLGYVHINEIQENEDENARIFGERIANKVQIF-----
2fcr      -----VAIFGLGDAEGYPDNFCDA-IEEIHDCFAKQGAQKPVGFSNPDDYDYEESKSVRDG-KFLGLPLDMVNDQIPMEKRVAGVWEAVVSETGV-----
FLAV_ENTAG -----VALFGLGDQLNYSKNFVSA-MRILYDLVIARGACVVGNWPREGYKFSFSAALLENNEFVGLPLDQENQYDLTEERIDSWLEKLKPAVL-----
FLAV_ANASP -----VAYFGTGDQIGYADNFQDA-IGILEEKISQRGGKTGVYWSTDGYDFNDSKALRNG-KFVGLALDEDNQSDLTDDRISWVAQLKSEFGL-----
FLAV_AZOVI -----VALFGLGDQVGYPENYLDA-LGELYSFFKDRGAKIVGSWSTDGYEFESSEAVVDG-KFVGLALDLNQSGKTDERVAAWLAQIAPEFGLSL----
FLAV_ECOLI -----VALFGCGDQEDYAEYFCDA-LGTIRDIIEPRGATIVGHWPTAGYHFEASKGLADDDHVFVGLAIDEDRQPELTAERVEKWKQISEELHLDEILNA
3chy      TAEAKKENIAAAQAGASGYVVKPFT--AATLEEKLNKIFEKLG-----
```