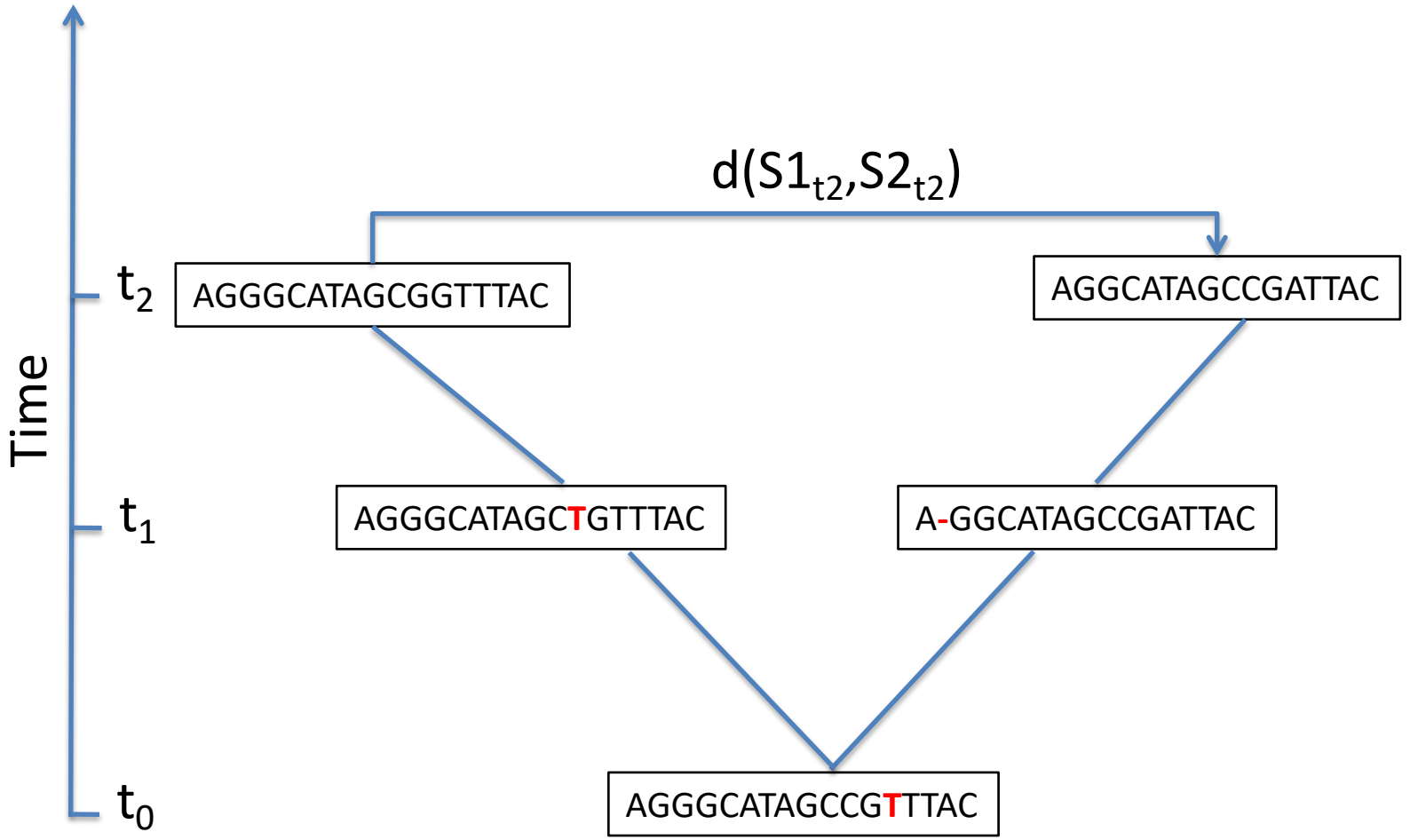
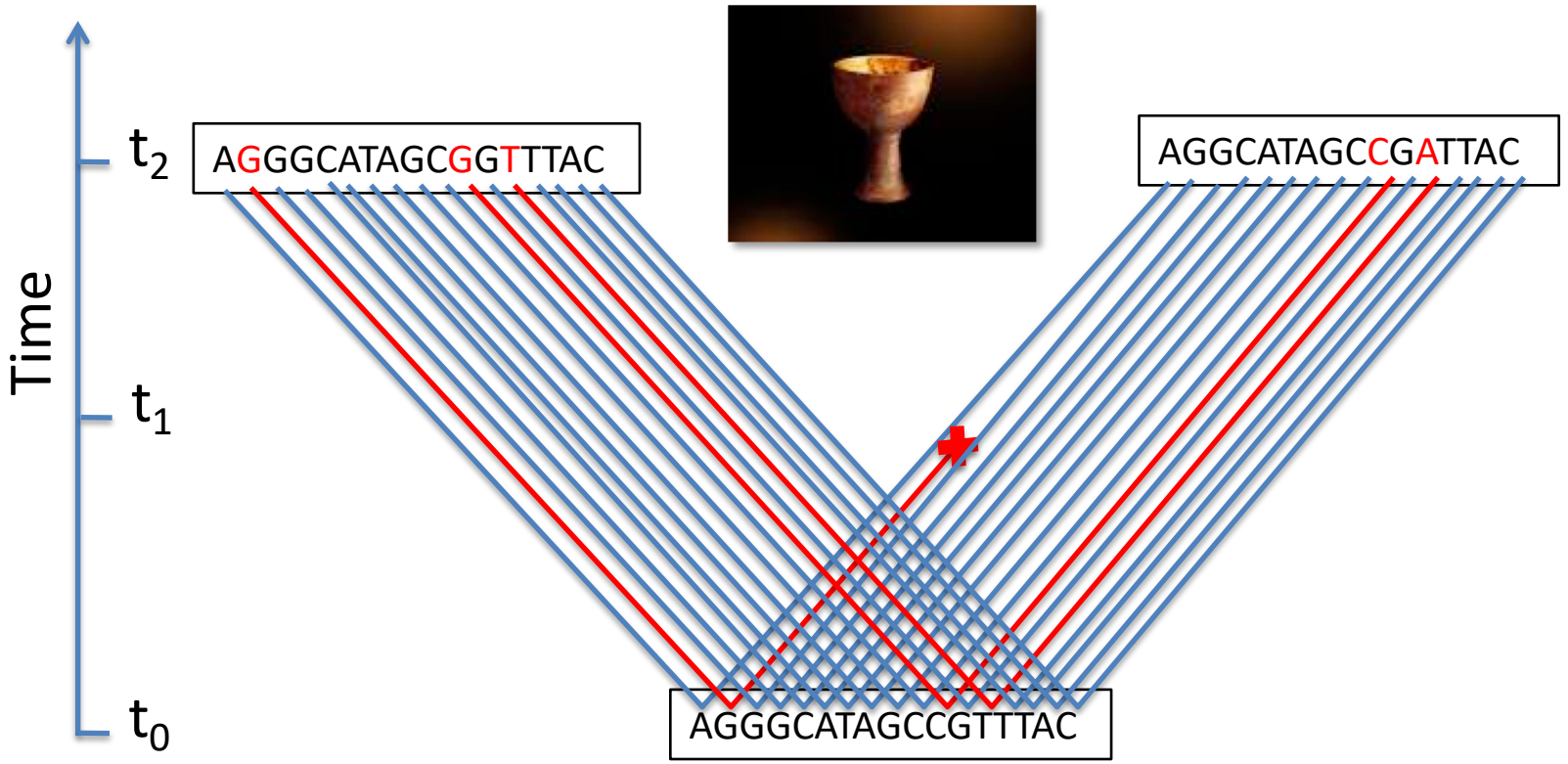
Algorithms in Sequence Analysis 7

Sequence alignment

The problem: We would like to know what has happened to two (or more) **homologous sequences** since they last shared a common ancestor!



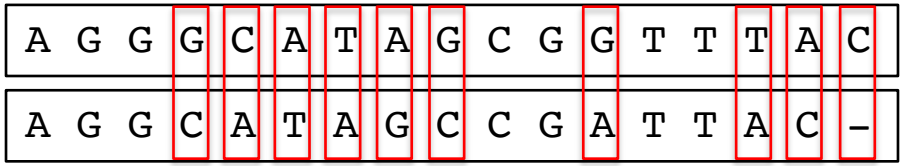
The quest to identify homologous positions in two sequences



Der ‘heilige Gral’ in der vergleichenden Sequenzanalyse

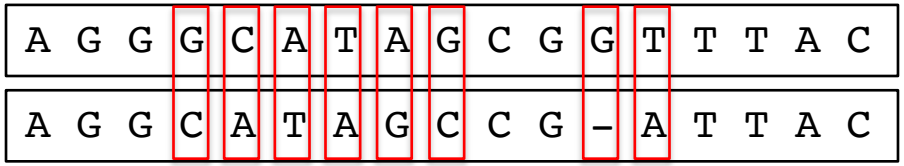
How to find this ‘true alignment’? We start with counting observed differences between the contemporary sequences, allowing for insertions, deletions and substitutions (Levenshtein Distance).

$d_{Levenshtein}(S1,S2)=10$



The problem: The Levenshtein distance changes with number and position of insertions/deletions

$d_{Levenshtein}(S1,S2)=8$



How to deal with this problem?

Finding the optimal alignment: Dynamic programming

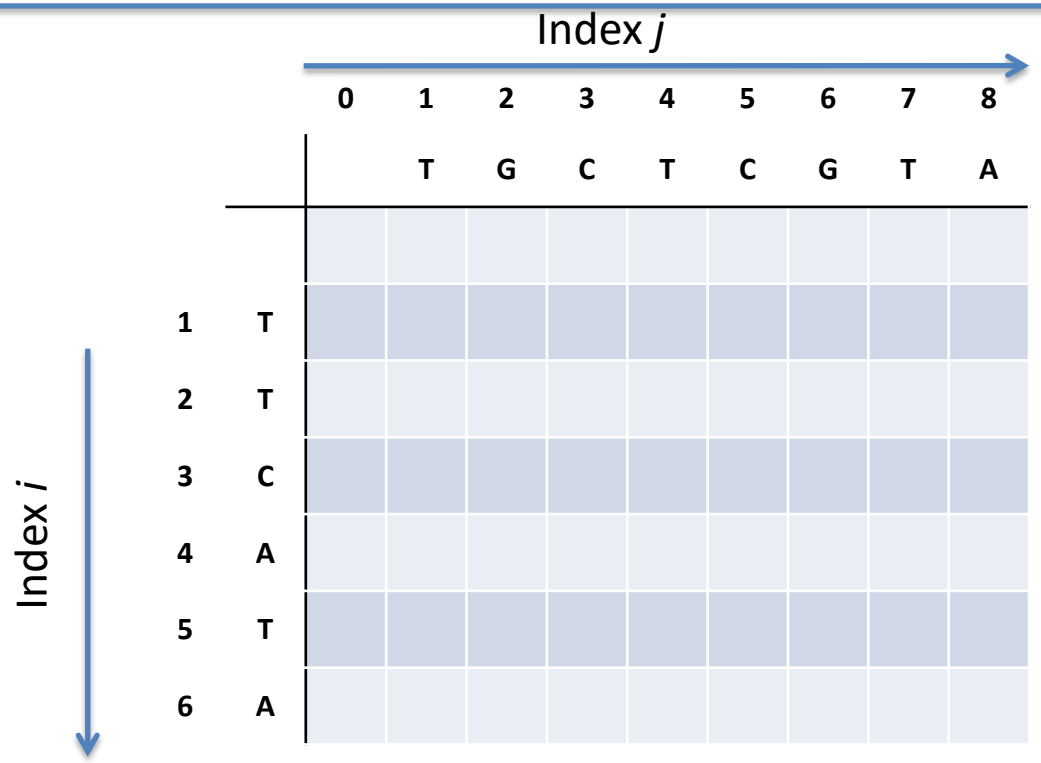
A **dynamic programming** approach usually includes:

- A mathematical description of the (biological) quality of a solution,

  i.e. a recursive objective function
- The computation of all intermediate values needed for obtaining the globally optimal solution, thereby avoiding double-computations
- The reconstruction of the globally optimal solution from the values obtained in the previous step (backtracking)

The Needleman-Wunsch Algorithm requires 3 things

1) The Matrix to take up (partial) alignment scores



2) A Scoring Function

$$S(a_i, b_j) = \begin{cases} +5, \text{if } a_i = b_j \\ -2, \text{if } a_i \neq b_j \\ -6, \text{for introduction of a gap} \end{cases}$$

3) An Objective Function

$$\sigma(i,j) = \max \begin{cases} \sigma(i-1, j-1) + S(a_i, b_j) \\ \sigma(i, j-1) + S(gap, b_j) \\ \sigma(i-1, j) + S(a_i, gap) \end{cases}$$

The Needleman-Wunsch Algorithm:
1) **Initialise** the matrix with cumulative gap scores

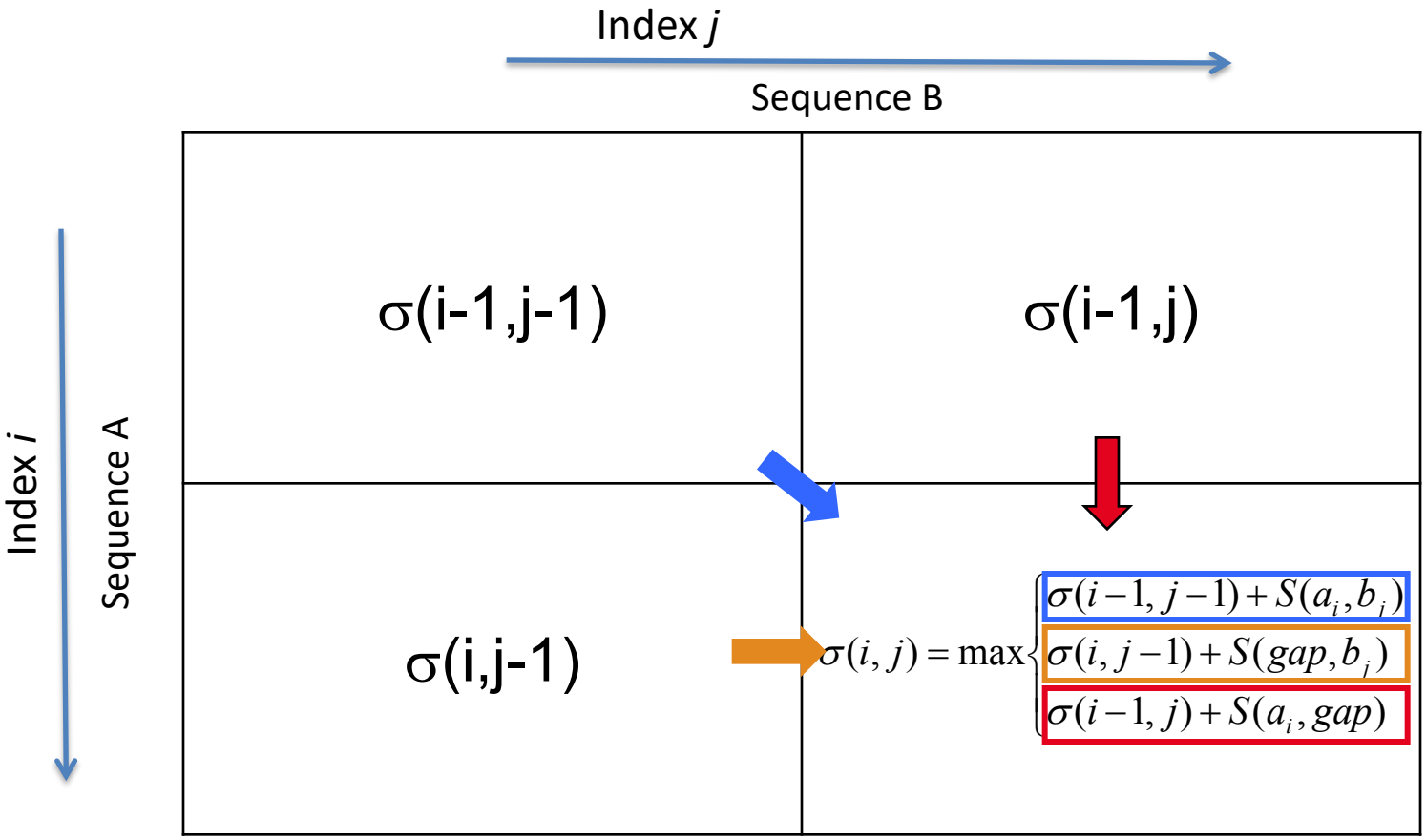
Index *j* →

		0	1	2	3	4	5	6	7	8
			T	G	C	T	C	G	T	A
		0	-6	-12	-18	-24	-30	-36	-42	-48
1	T	-6								
2	T	-12								
3	C	-18								
4	A	-24								
5	T	-30								
6	A	-36								

Index *i* ↓

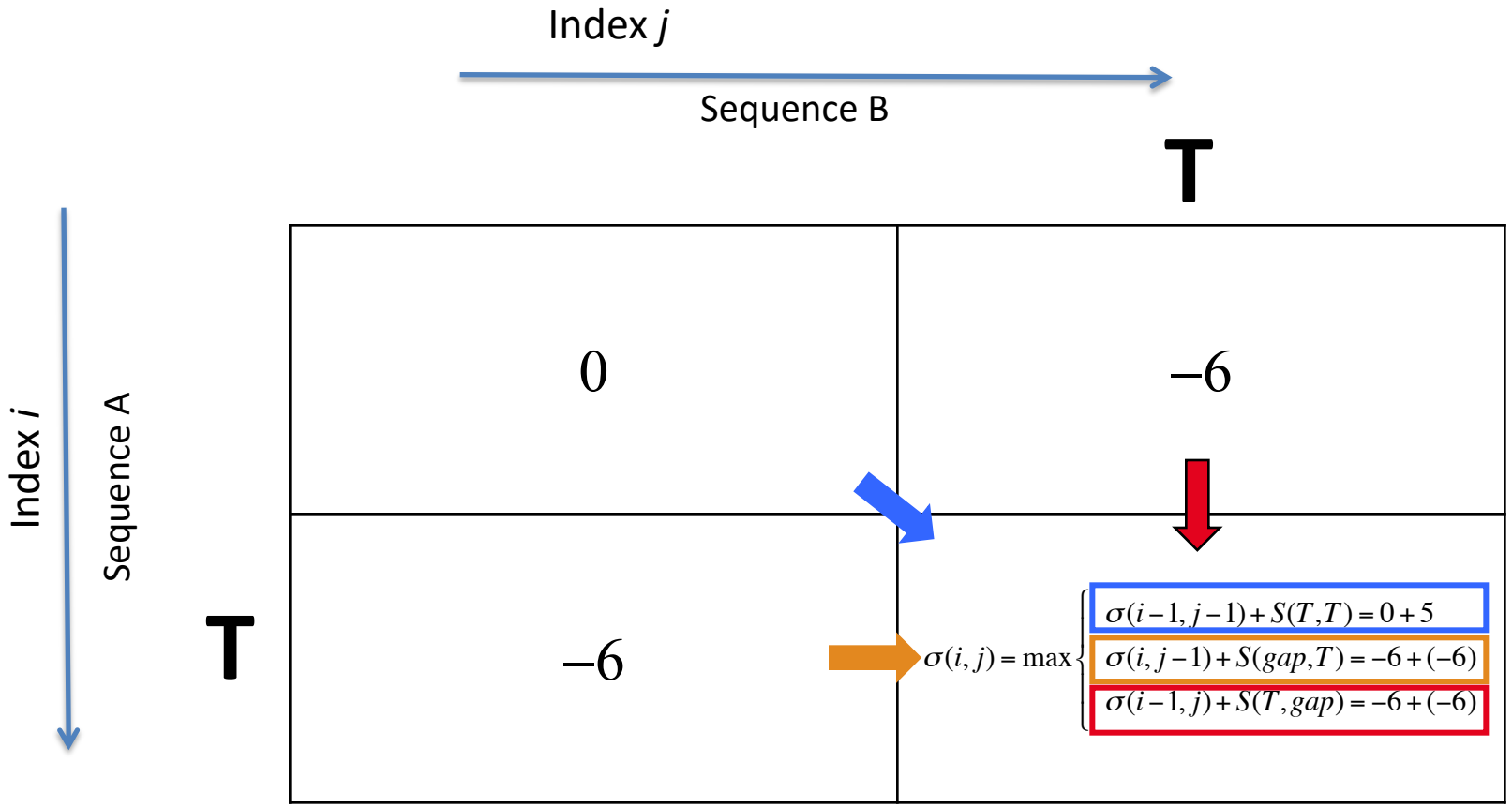
$$S(a_i, b_j) = \begin{cases} +5, if\ a_i = b_j \\ -2, if\ a_i \neq b_j \\ -6, \text{for introduction of a gap} \end{cases}$$

The Needleman-Wunsch Algorithm:
2) Recursive computation of intermediate alignment scores



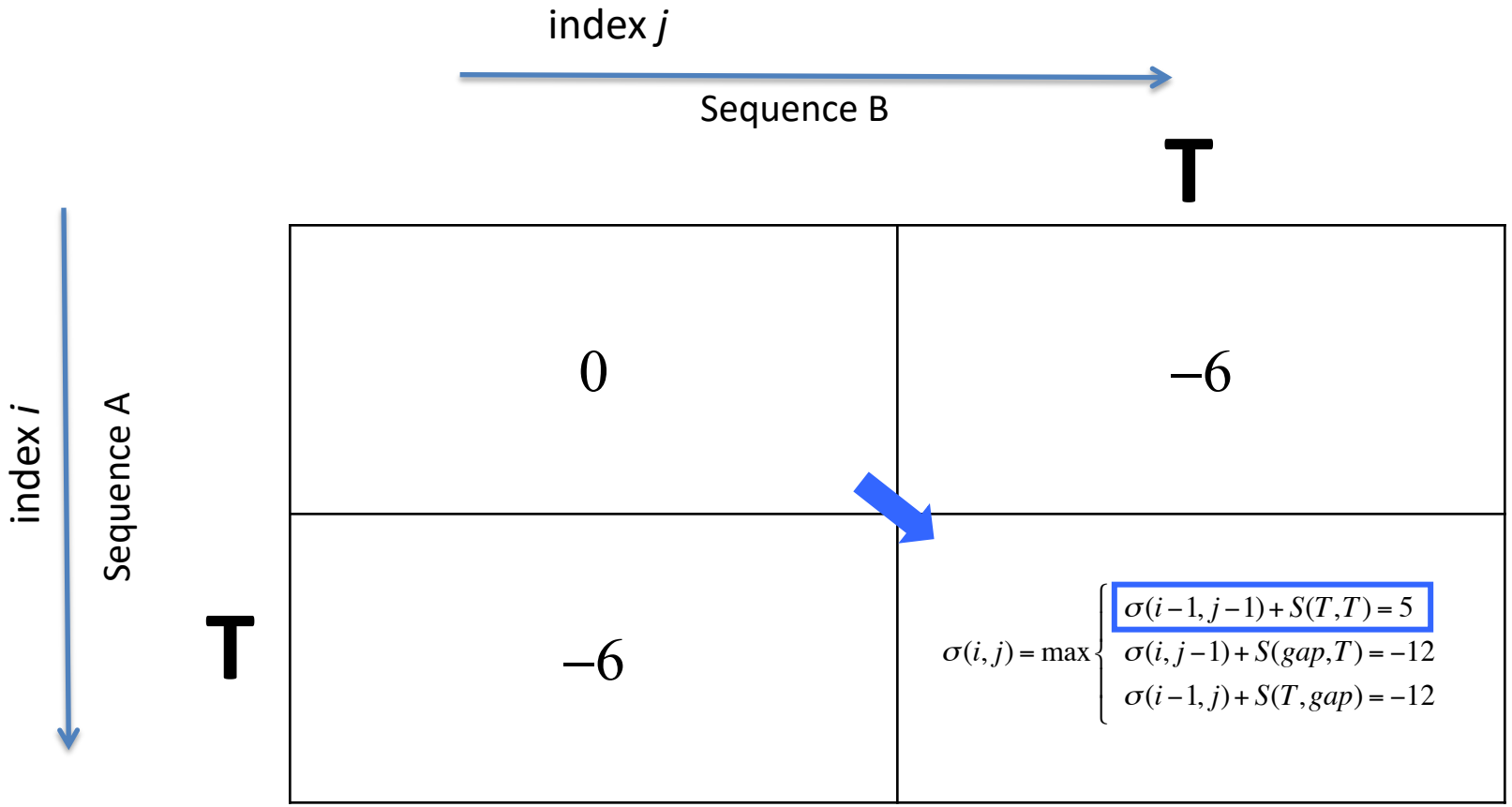
σ(i,j) is the optimal alignment score up to and including a_i and b_j

The Needleman-Wunsch Algorithm:
2) Recursive computation of intermediate alignment scores



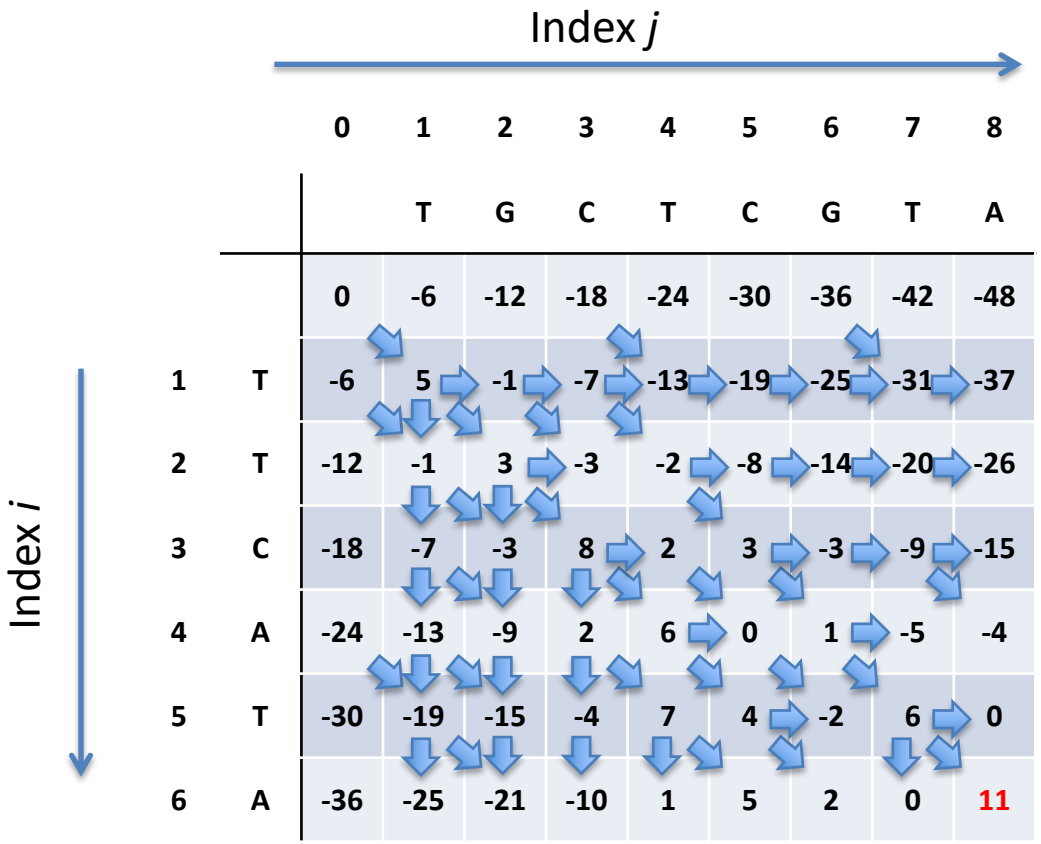
$\sigma(i,j)$ is the optimal alignment score up to and including a_i and b_j

The Needleman-Wunsch Algorithm:
2) Recursive computation of intermediate alignment scores



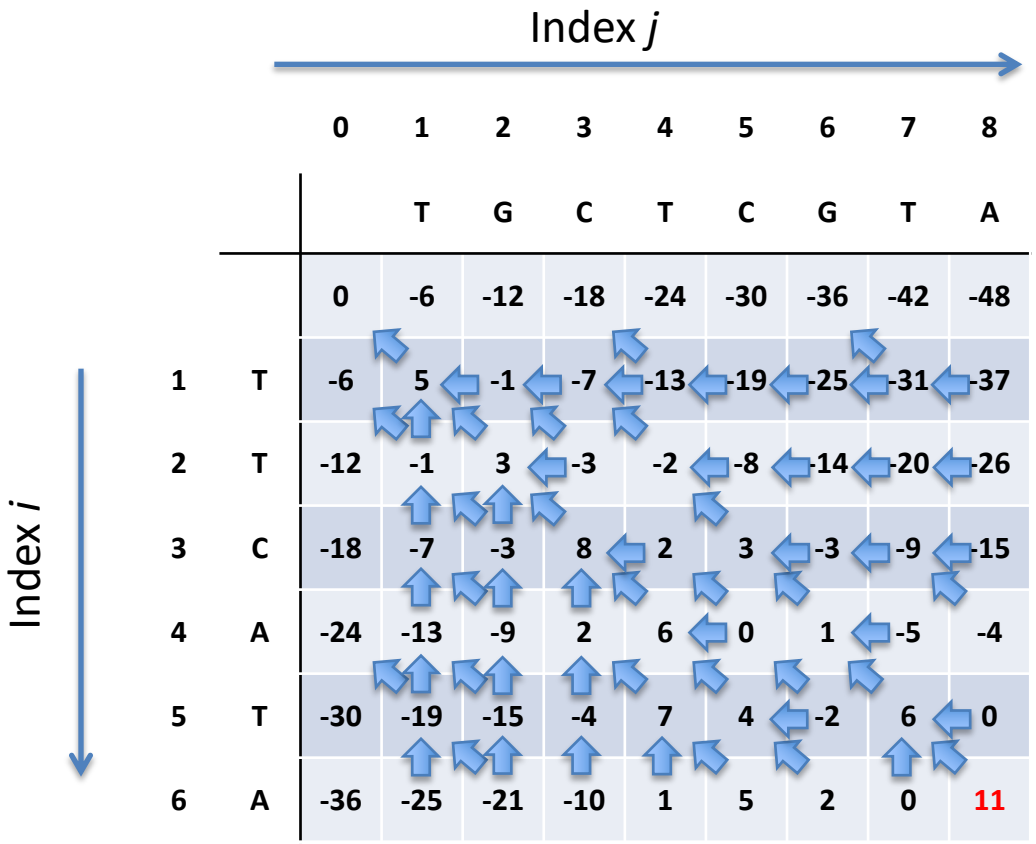
$\sigma(i,j)$ is the optimal alignment score up to and including a_i and b_j

The Needleman-Wunsch Algorithm:
3) Backtrace: Rekonstructing the optimal Alignment



Remember: The Backtrace starts in the case of **Needleman-Wunsch** always at the lower right cell

The Needleman-Wunsch algorithm (Backtracking): Reconstructing the optimal alignment

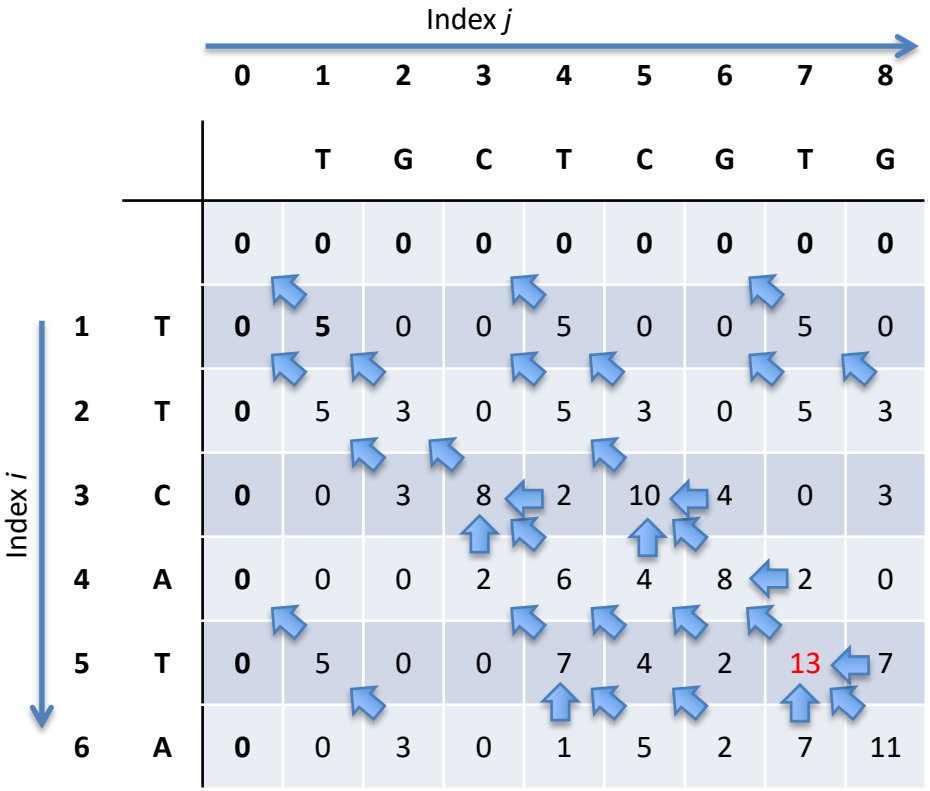


T G C T C G T A *
T – – T C A T A *

Just follow the pointers backwards to the origin to reconstruct the optimal alignment.

Smith-Waterman sequence alignment: An overview

Given Sequences A and B and the scoring function $S(a_i,b_j) = \begin{cases} +5, if\ a_i = b_j \\ -2, if\ a_i \neq b_j \\ -6, \text{for introduction of a gap} \end{cases}$



Optimal local alignment

T C G T *
T C A T *

- initialize a *n* x *m* matrix representing sequenzen *A* and *B* of length *m* and *n,* respectively. Set values of first row and column to 0.
- Compute recursively the $\sigma(i\ ,j) = \max = \begin{cases} \sigma(i-1,j-1) + s(a_i,b_j) & \text{match / mismatch} \\ \sigma(i-1,j) + s(a_i,-) & \text{gap in B} \\ \sigma(i,j-1) + s(-,b_j) & \text{gap in A} \\ 0 \end{cases}$
- The optimal local Alignment-Score is obtained by identifying the cell with the highest score σ(i ,j).
- The optimal local alignment is obtained by a backtrace from this cell to the first cell with a value of 0.

Scoring sequence similarity

What is a sensible way to judge sequence similarity?

1. fraction of identical sequence positions in two sequences

$$S(a_i,b_j) = \begin{cases} +5, if\ a_i = b_j \\ -2, if\ a_i \neq b_j \\ -6, \text{for introduction of a gap} \end{cases}$$

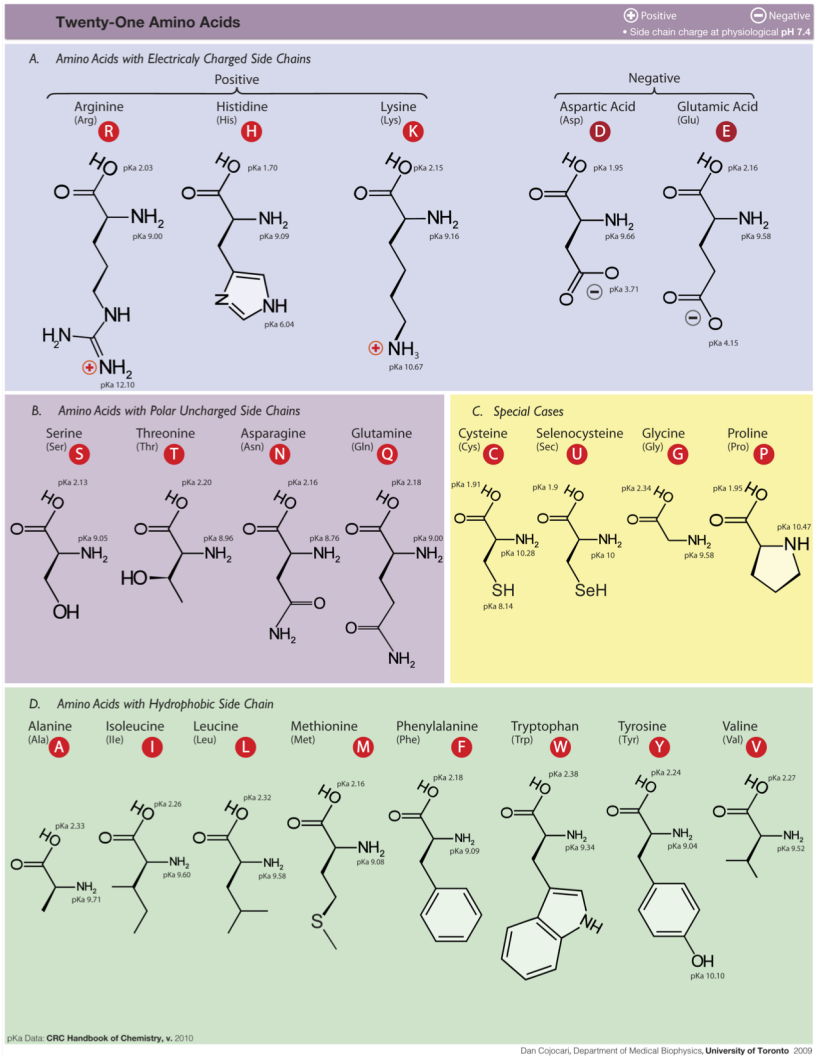
	A	G	C	T
A	5	-2	-2	-2
G	-2	5	-2	-2
C	-2	-2	5	-2
T	-2	-2	-2	5

1. fraction of similar sequence positions in two sequences

	A	G	C	T
A	?	?	?	?
G	?	?	?	?
C	?	?	?	?
T	?	?	?	?

This is not too relevant for DNA sequences but of great importance for protein sequences

Scoring amino acid sequence similarity



Some amino acids are more similar to each other than others. To understand why this is relevant during sequence alignment, recall the two main reasons for assessing sequence similarity:

1) Estimating evolutionary distance
2) Deciding on functional similarity

Scoring amino acid sequence similarity

Rationale

Different amino acids can vary in their similarity with respect to:

1) chemical properties (e.g., hydrophilic/lipophilic)
2) size
3) difference in the underlying codons (Glu-Asp: 1 substitution, Glu-Phe: 3 substitutions)
4) charge (positive/negative/neutral)

It is hard to invent de-novo a meaningful scoring scheme considering all these aspects. An empirical approach may be a more promising way to achieve this goal.

Scoring amino acid sequence similarity

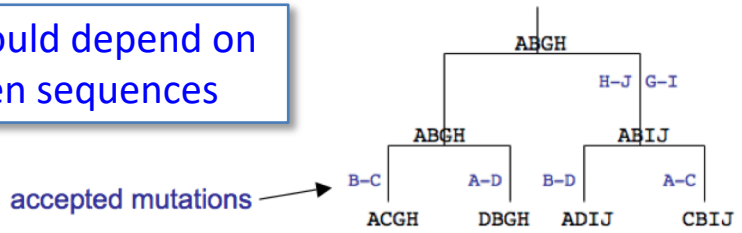
Approach 1:
invent a scoring schema based on observed aa changes in more **closely related** protein sequences (PAM matrix)

Approach 2:
invent a scoring schema based on observed aa changes in **conserved blocks of more distantly related** protein sequences (BLOSUM).

WE THINK WE KNOW THE TRUE ALIGNMENT

Scoring substitutions using the PAM matrix (**P**oint **A**ccepted **M**utations)

Key idea: The substitution score should depend on the evolutionary distance between sequences



The **PAM matrices** derived by Dayhoff (1978):

- are based on evolutionary distances.
- have been obtained from carefully aligned closely related protein sequences (71 gapless alignments of sequences having at least 85% similarity).

M. Dayhoff

Reference: Dayhoff *et al.* (1978). A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, vol. 5, suppl. 3, 345–352. National Biomedical Research Foundation, Silver Spring, MD, 1978.

We think we know (approximately) the 'true' alignment

For details see the Grundlagen lecture

Using PAM scoring matrices for evaluating alignments

PAM250 matrix (log-odds)

Cys	C	12																		
Ser	S	0	2																	
Thr	T	-2	1	3																
Pro	P	-1	1	0	6															
Ala	A	-2	1	1	1	2														
Gly	G	-3	1	0	-1	1	5													
Asn	N	-4	1	0	-1	0	0	2												
Asp	D	-5	0	0	-1	0	1	2	4											
Glu	E	-5	0	0	-1	0	0	1	3	4										
Gln	Q	-5	-1	-1	0	0	-1	1	2	2	4									
His	H	-3	-1	-1	0	-1	-2	2	1	1	3	6								
Arg	R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6							
Lys	K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5						
Met	M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6					
Ile	I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5				
Leu	L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6			
Val	V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4		
Phe	F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9	
Tyr	Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10
Trp	W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17
		C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
		Cys	Ser	Thr	Pro	Ala	Gly	Asn	Asp	Glu	Gln	His	Arg	Lys	Met	Ile	Leu	Val	Phe	Tyr	Trp
Hydrophobic		C			P	A	G								M	I	L	V			
Aromatic												H							F	Y	W
Polar			S	T				N			Q									Y	
Basic												H	R	K							
Acidic									D	E											

These log-odds scores can now be used for evaluating pairwise alignments

```
T A H G K
Y S D G D
```

$S_{alignment} = S_n(T,Y) + S_n(A,S) + S_n(H,D) + S_n(G,G) + S_n(K,D)$

$= -3 + 1 + 1 + 5 + 0 = 4$

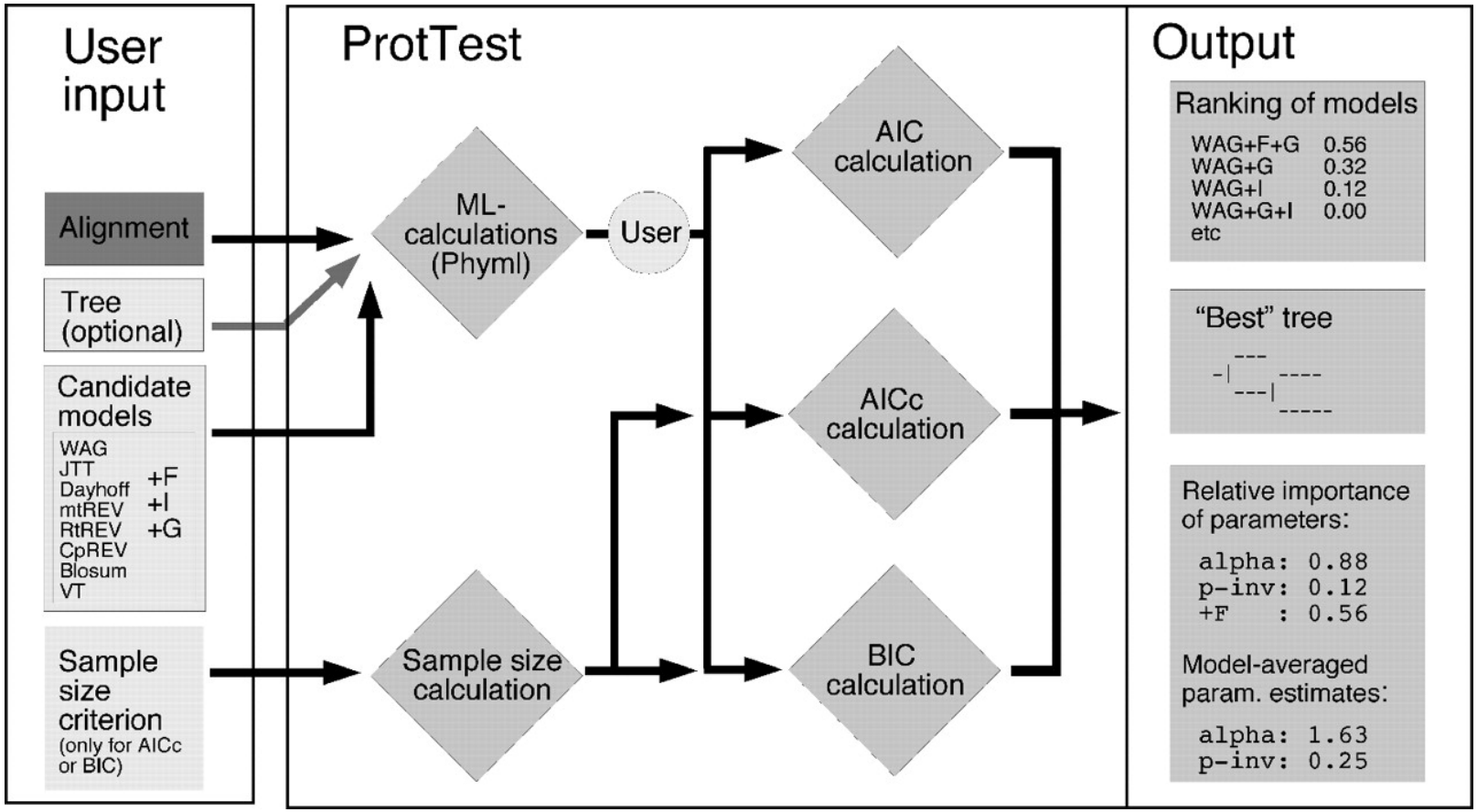
Source: J. van Helden

There is way more than just PAM[1], so which model should I use?

Name	Description	Publication
PAM	Count-based. Analysis of 71 closely related protein families. Different evolutionary distances are extrapolated.	Dayhoff et al. (1978) *Atlas of Protein Sequence and Structure* ***5*** *(3): 345–352*
BLOSUM	Count-based. Analysis of conserved, gap-free blocks within diverged proteins. Training data vary for different matrices	Henikoff et al. (1992) *PNAS* ***89*** *(22): 10915–10919*
JTT (Jones, Taylor, Thornton)	Count-based. Increased training data, single linkage clustering	Jones et al. (1992) *Computer Applications in the Biosciences* **8**: 275-282
WAG (Wheelan and Goldman)	Approximate likelihood method. Globular protein sequences comprising 3,905 amino acid sequences split into 182 protein families.	Wheelan et al, (2001) *Mol Biol Evol 18 (5): 691-699*
LG (Le and Gascuel)	Approximate likelihood method. Refines WAG by incorporating the variability of evolutionary rates across sites and by using a much larger and diverse database	Le et al. (2008) *Mol Biol Evol (2008) 25 (7): 1307-1320*
mtREV	Maximum likelihood (ML) method from the complete sequence data of mtDNA from 20 vertebrate species	Adachi et al (1996) J Mol Evol. 42(4):459-68.
cpREV	Transition matrix based on the best tree, called cpREV, takes into account distinct substitution patterns in plastid-encoded proteins	Adachi et al. (2000) J Mol Evol. 50(4):348-58.
CAT	Bayesian mixture model that allows the amino-acid replacement pattern at different sites of a protein alignment to be described by distinct substitution processes.	Lartillot et al. (2004) MBE 21(6):1095-109

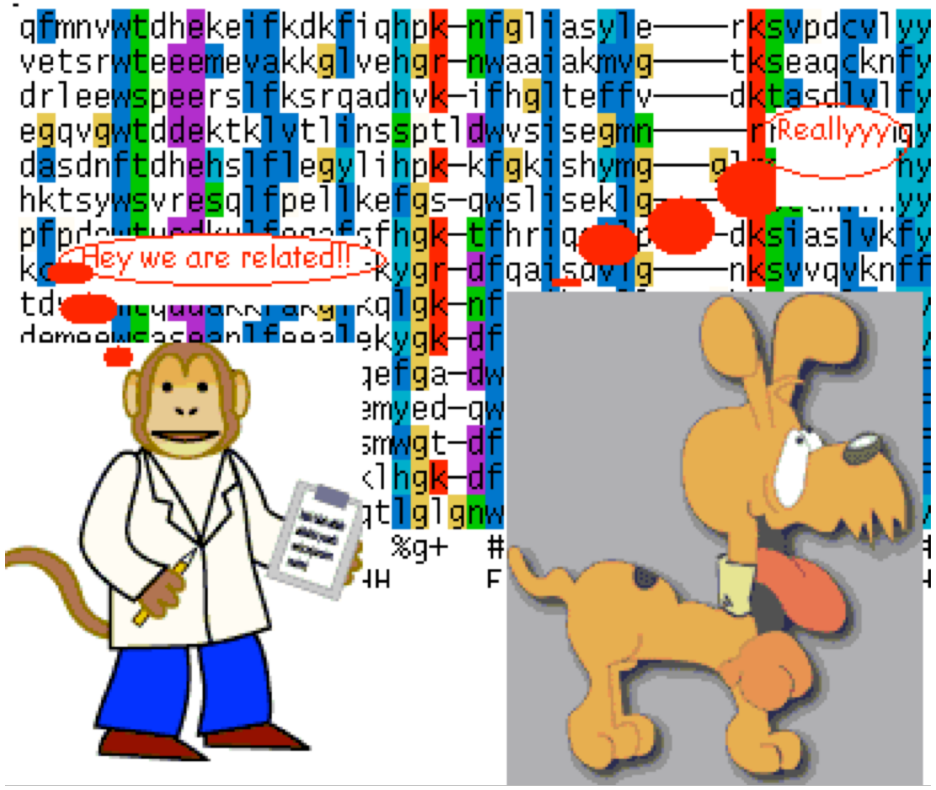
1 list of models is not exhaustive

The basic workflow of ProtTest Program for selecting the model giving the best fit to the data



Federico Abascal et al. Bioinformatics 2005;21:2104-2105

Multiple Sequence Alignment

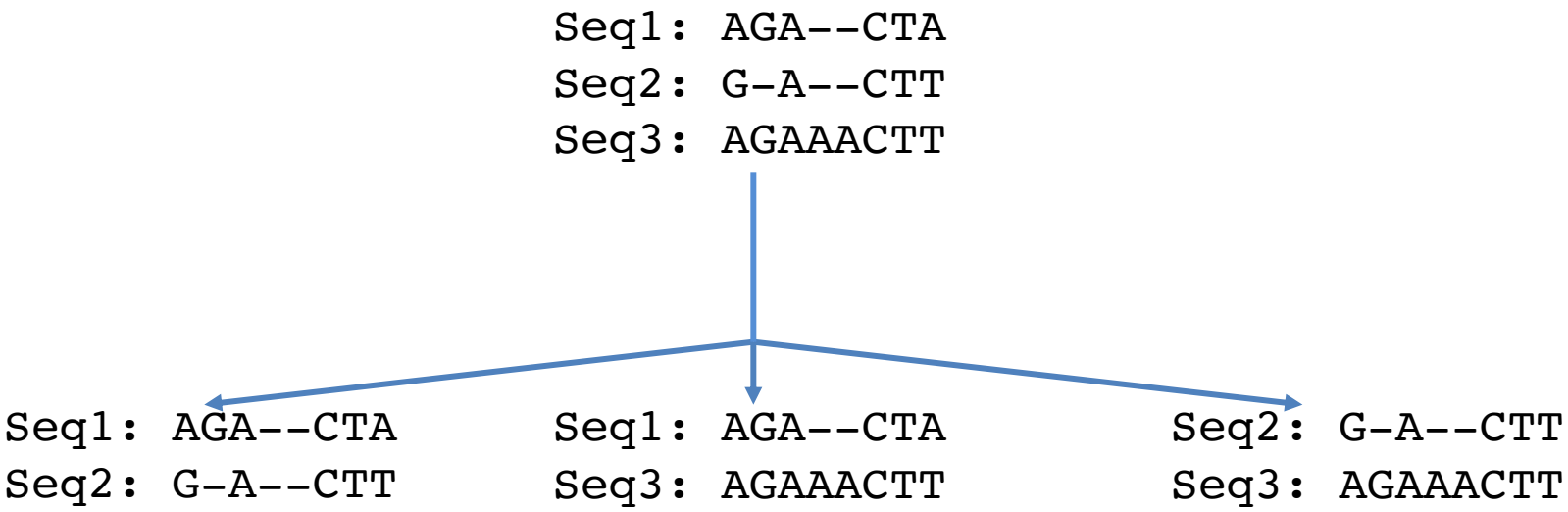


Picture from http://www.bioinformaticsworld.com

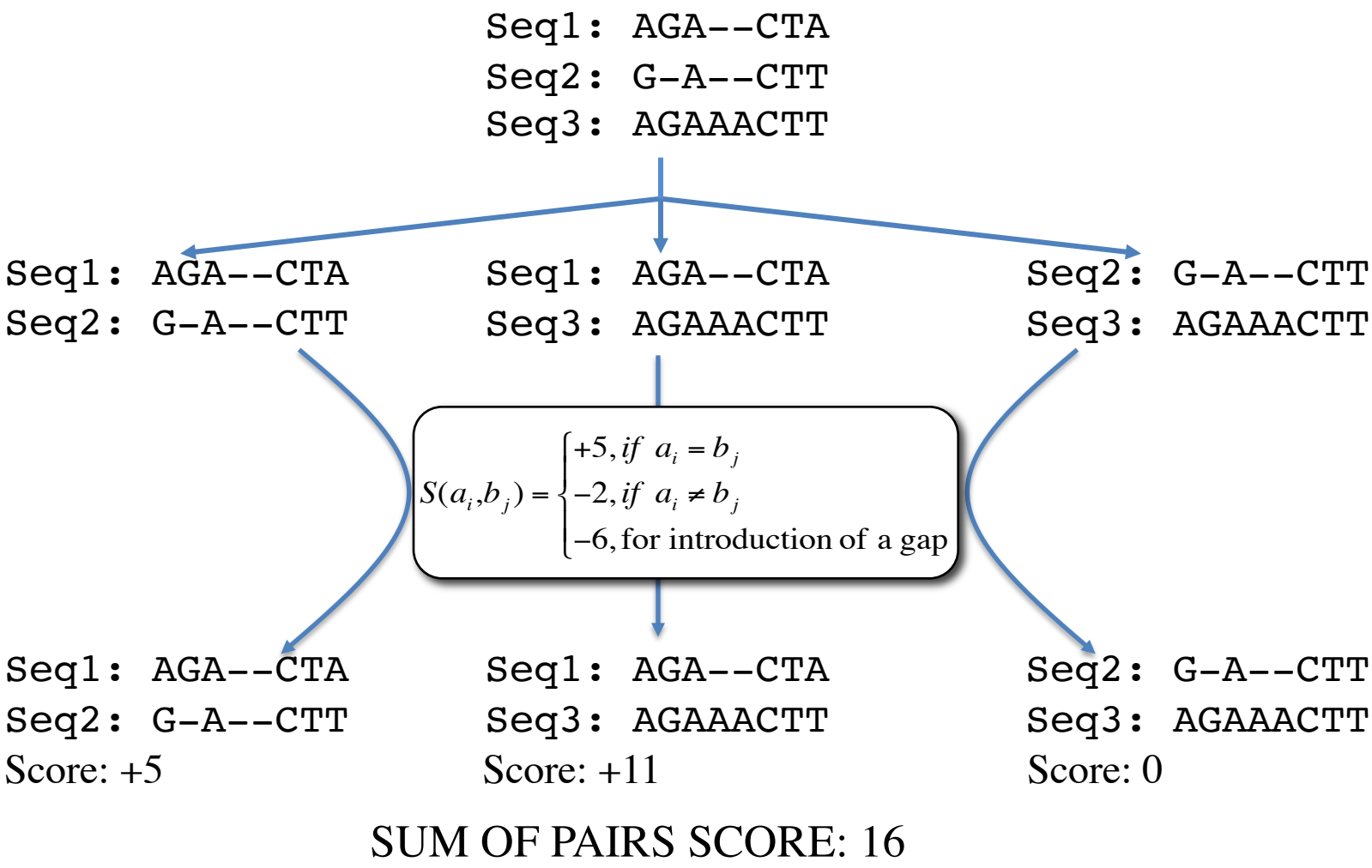
Multiple Sequence alignment
What is it good for?

Scoring multiple sequence alignments: Sum Of Pairs Score (simple)

Approach: break an unsolved problem down to problems for which there already exists a solution.



Computing the Sum Of Pairs Score



Aligning multiple sequences

Task: Align 4 sequences following a pairwise approach.

Pair 1

```
Sequence 1: NYLS
Sequence 2: NFS
```

```
NYLS
N-FS
```

?

Pair 2

```
Sequence 3: NKYLS
Sequence 4: NFLS
```

```
NKYLS
NF-LS
```

Aligning multiple sequences
Scoring the alignment of two alignments

Task: Align 4 sequences following a pairwise approach.

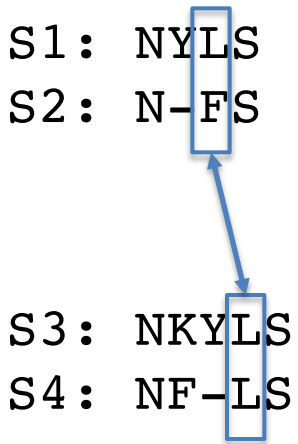
Pair 1

```
Sequence 1: NYLS      NYLS
Sequence 2: NFS       N-FS
```

Pair 2

```
Sequence 3: NKYLS     NKYLS
Sequence 4: NFLS      NF-LS
```

```
N-YLS
N--FS
NKYLS
NF-LS
```



$$Score(LFLL) = (S(L_1,L_3)+ S(L_1,L_4)+ S(F_2,L_3)+ S(F_2,L_4))$$

Progressive alignment strategy
Scoring the alignment of two alignments

Task: Align 4 sequences following a pairwise approach but use different pairings.

Pair 1

```
Sequence 1: NYLS      N-YLS
Sequence 3: NKYLS     NKYLS
```

Pair 2

```
Sequence 2: NFS       NF-S
Sequence 4: NFLS      NFLS
```

```
N-YLS
NKYLS
N-F-S
N-FLS
```

Alignment 1:

```
S1: N-YLS
S2: N--FS
S3: NKYLS
S4: NF-LS
```

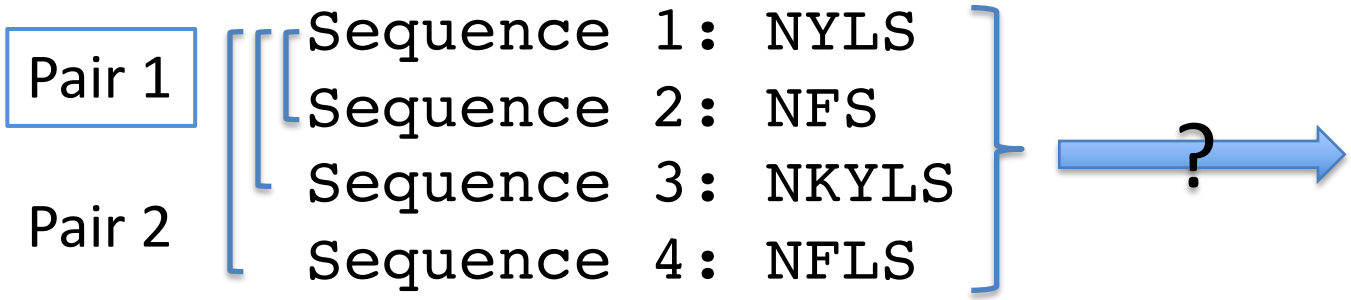
Alignment 2:

```
S1: N-YLS
S3: NKYLS
S2: N-F-S
S4: N-FLS
```

Thus, the alignment can change with the order of the sequences!

Progressive alignment strategy

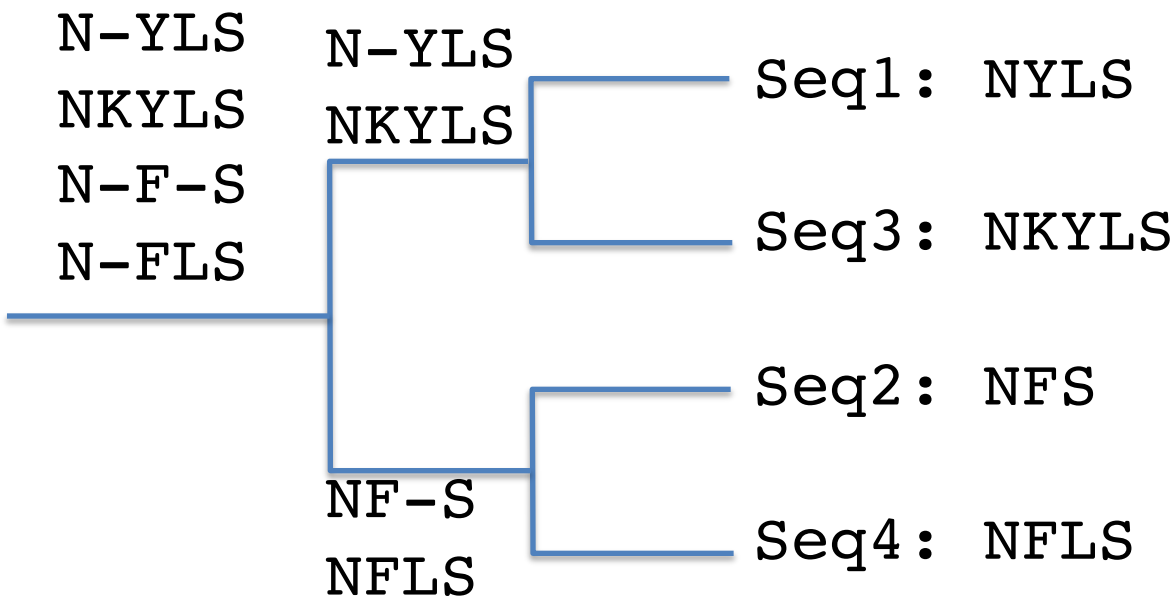
Task: Cope with the problem that the alignment changes with the sequence order



Remember the assumption: The sequences evolved along a tree
Thus, it may be a good idea to align them along exactly this tree.

Progressive alignment strategy

1. Reconstruct a tree
2. Align the sequences progressing from the leafs to the root
3. Align sub-alignments (profiles) at the nodes where internal branches meet



Problem: Where do we get the tree from when we require an MSA for reconstructing such a phylogeny? Typical hen-and-egg problem…

Getting the tree for a set of sequences without performing an MSA

Re-formulation of the problem: Look for the tree that groups sequences according to their similarity rather than for the tree that groups sequences according to their phylogenetic relationships.

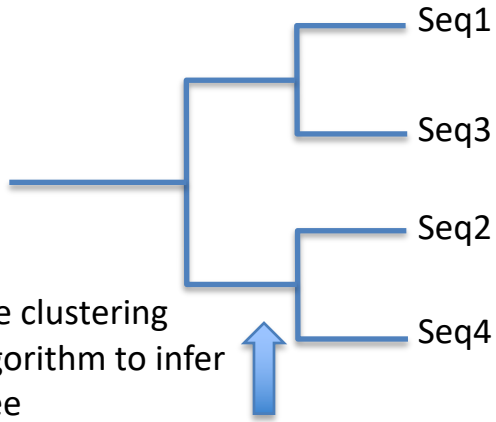
```
Seq1: NYLS
Seq2: NFS
Seq3: NKYLS
Seq4: NFLS
```

compute all optimal pairwise alignments

compute pairwise distance matrix*

	S1	S2	S3	S4
S1	0			
S2	4	0		
S3	2	4	0	
S4	4	2	4	0

use clustering algorithm to infer tree



*values in this matrix are for illustrative purpose only and are not computed from the example sequences

ClustalW (Higgins et al. 1994)
One of the most well-known MSA algorithms

Journal List › Nucleic Acids Res › v.22(22); Nov 11, 1994 › PMC308517

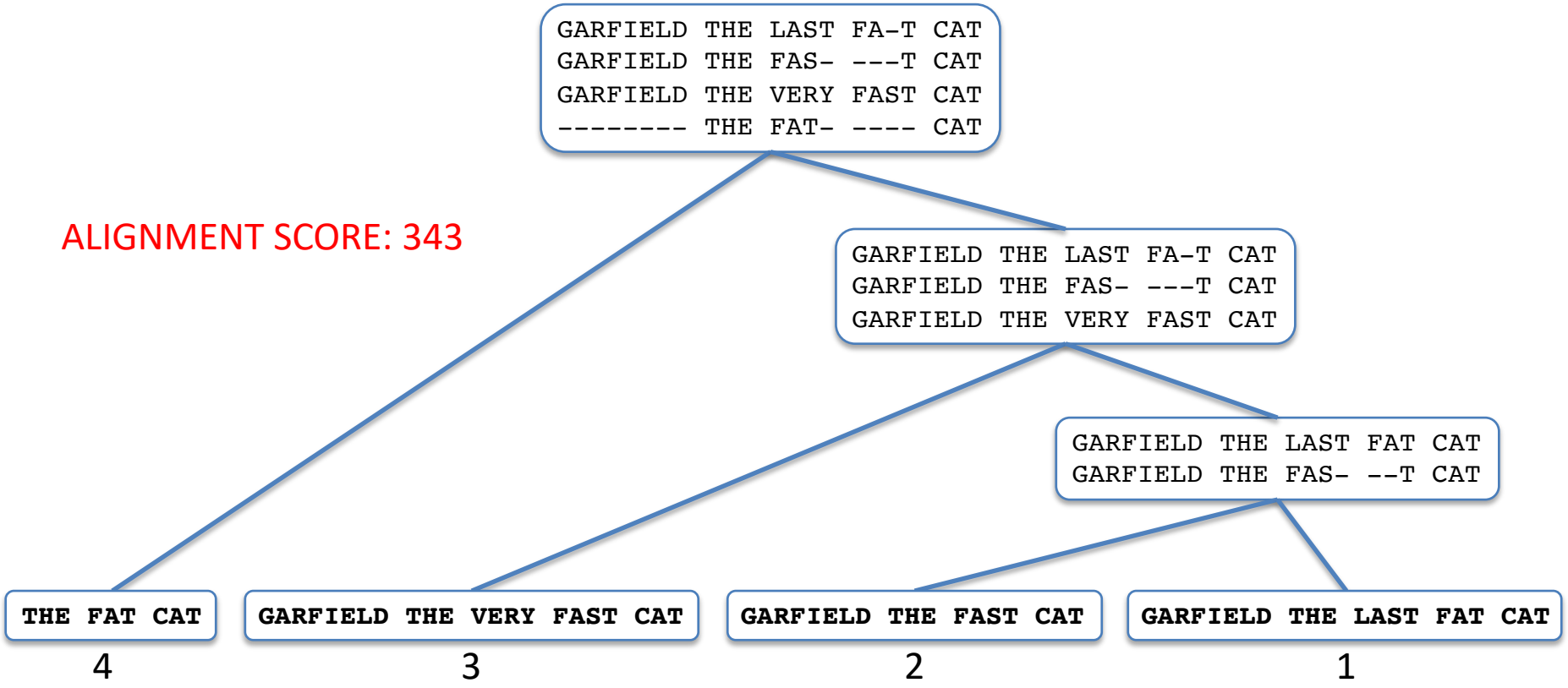
Nucleic Acids Research

Nucleic Acids Res. Nov 11, 1994; 22(22): 4673–4680. PMCID: PMC308517

CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.

J D Thompson, D G Higgins, and T J Gibson

Progressive alignment strategy
Problems: Once a gap always a gap

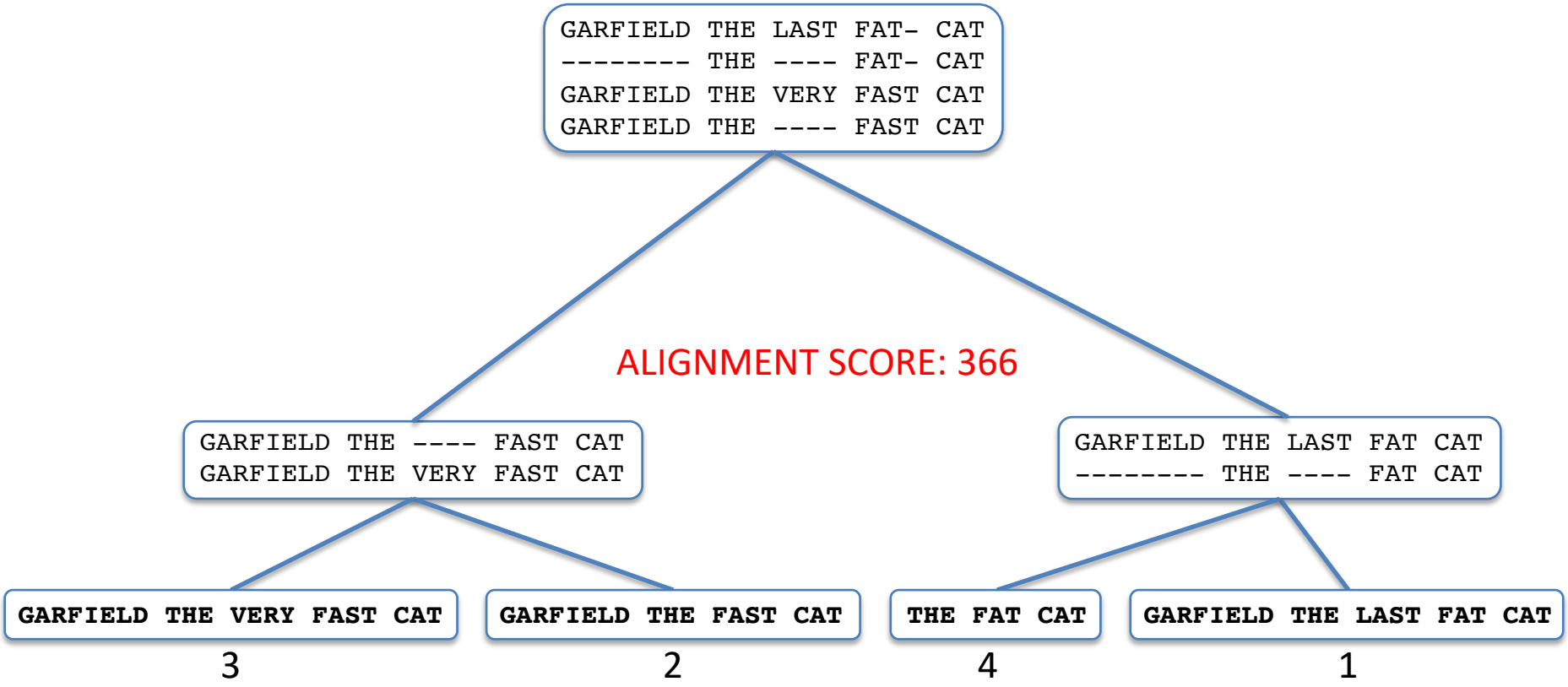


It is easy to see that the greedy strategy of a progressive alignment is not guaranteed to arrive at the globally optimal alignment.

Alignment computed with Clustalw2 using a custom guide tree as shown without sequence weights

Progressive alignment strategy

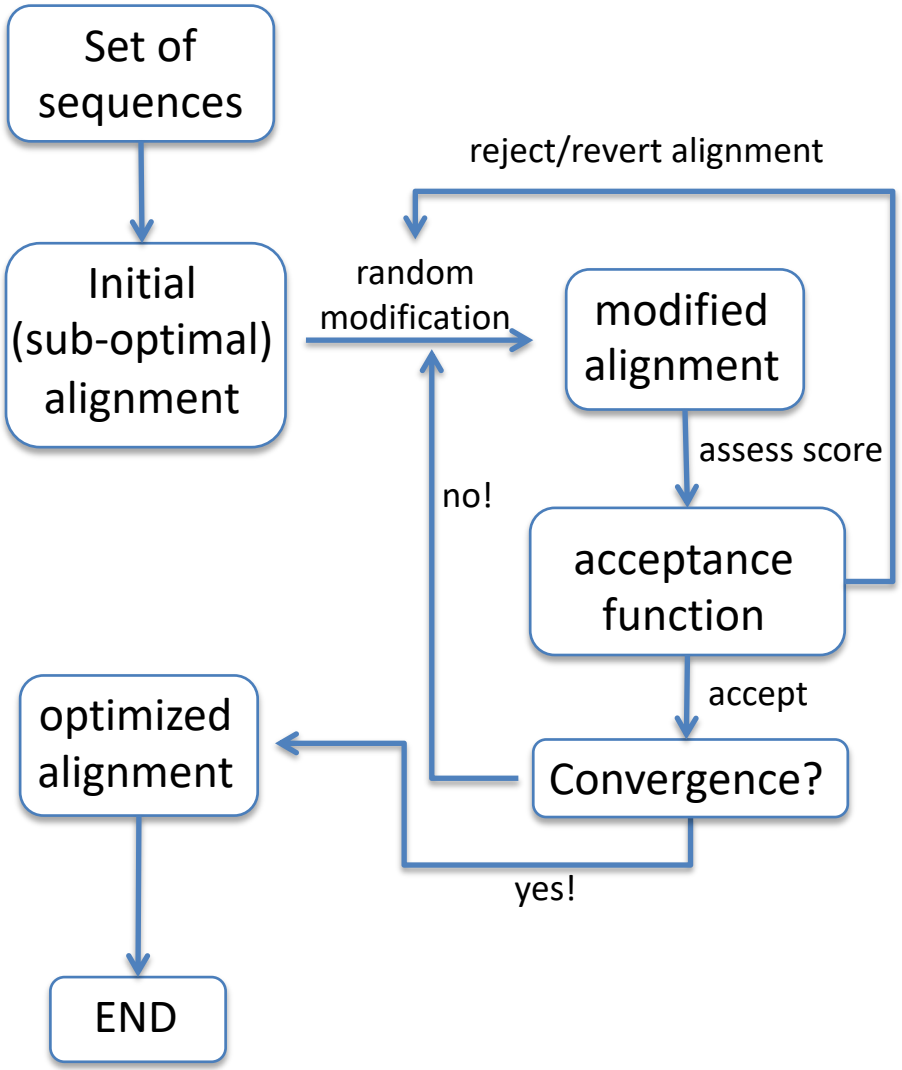
Problems: Once a gap always a gap



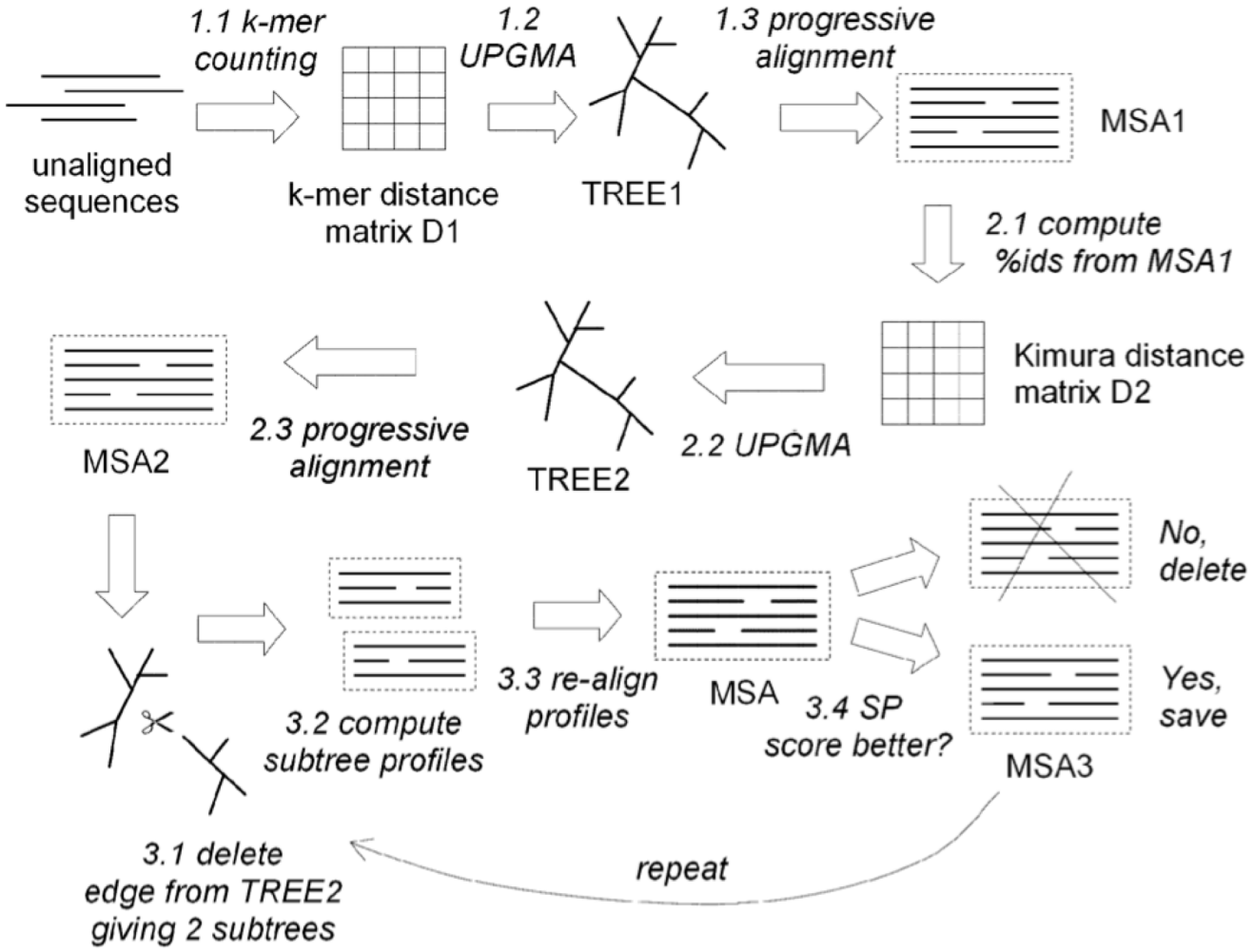
It is now easy to see that the appropriate choice of the guide tree has a substantial impact on the outcome of a multiple sequence alignment.

Alignment computed with Clustalw2 using standard settings

How to overcome previous (and limiting) decisions? Iterative alignment strategies aim at optimizing an initial and potentially sub-optimal alignment (outline)

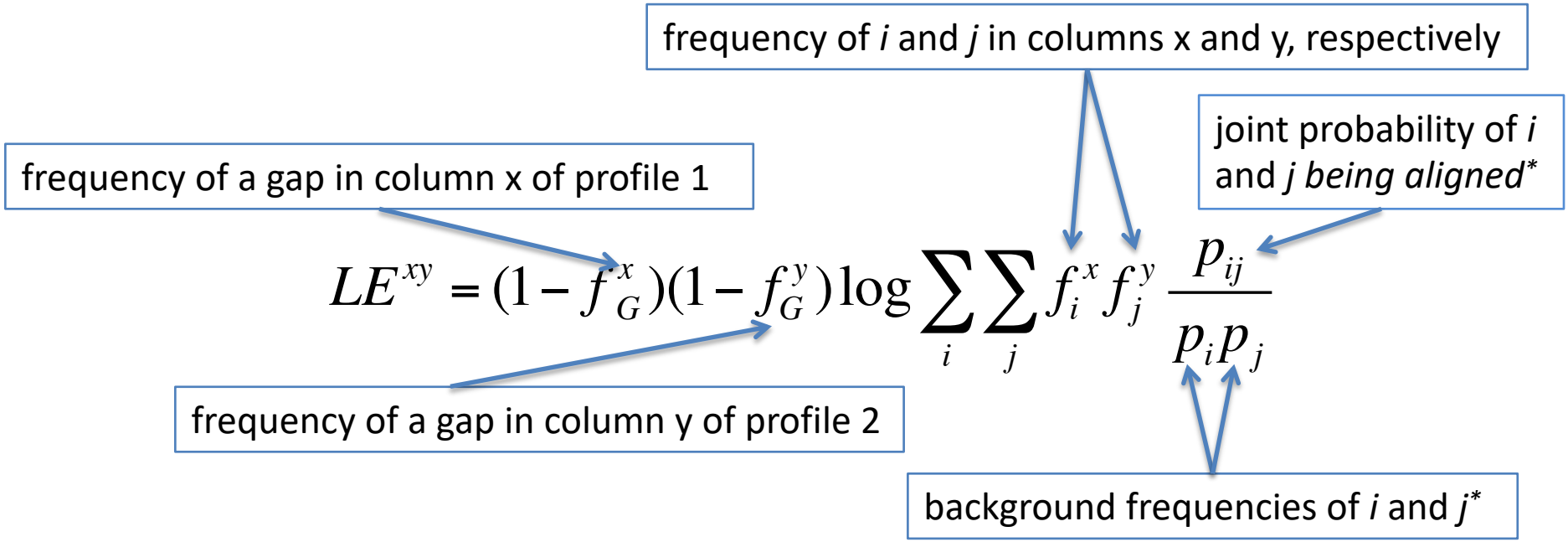


One example of a stochastic iterative alignment MUSCLE



Robert Edgar (2004) Nucleic Acids Res 32:1792--1797

MSA with Muscle: Scoring the alignment of column x from profile 1 and column y from profile 2 (Log Expectation score)

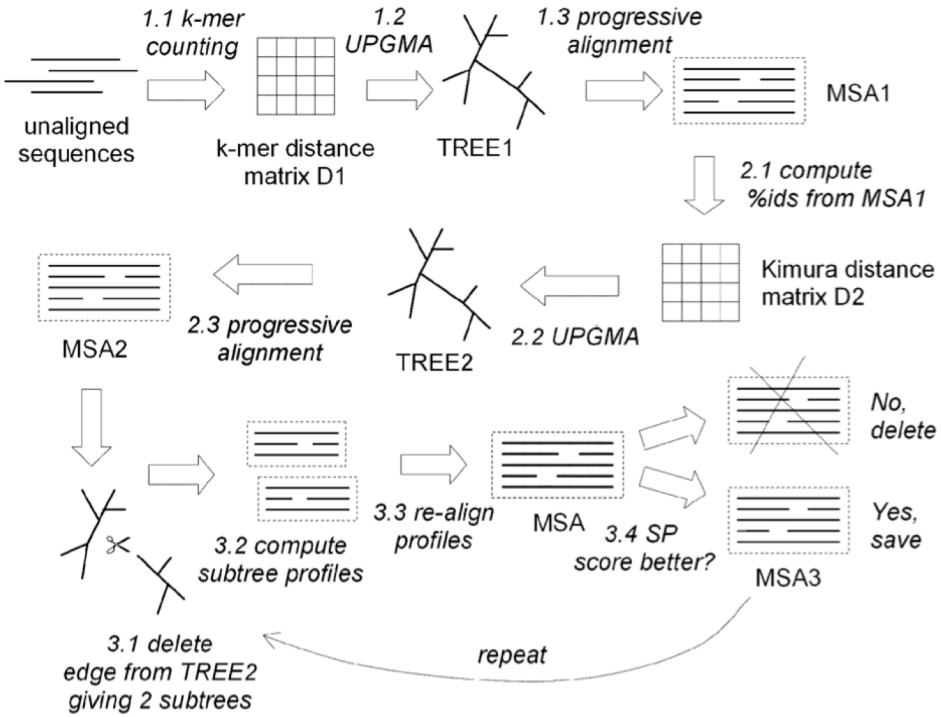


i,j represent letters from the sequence alphabet

Robert Edgar (2004) Nucleic Acids Res 32:1792--1797

* 240 PAM VTML matrix. Muller et al. (2002) Mol. Biol. Evol., 19, 8-13.

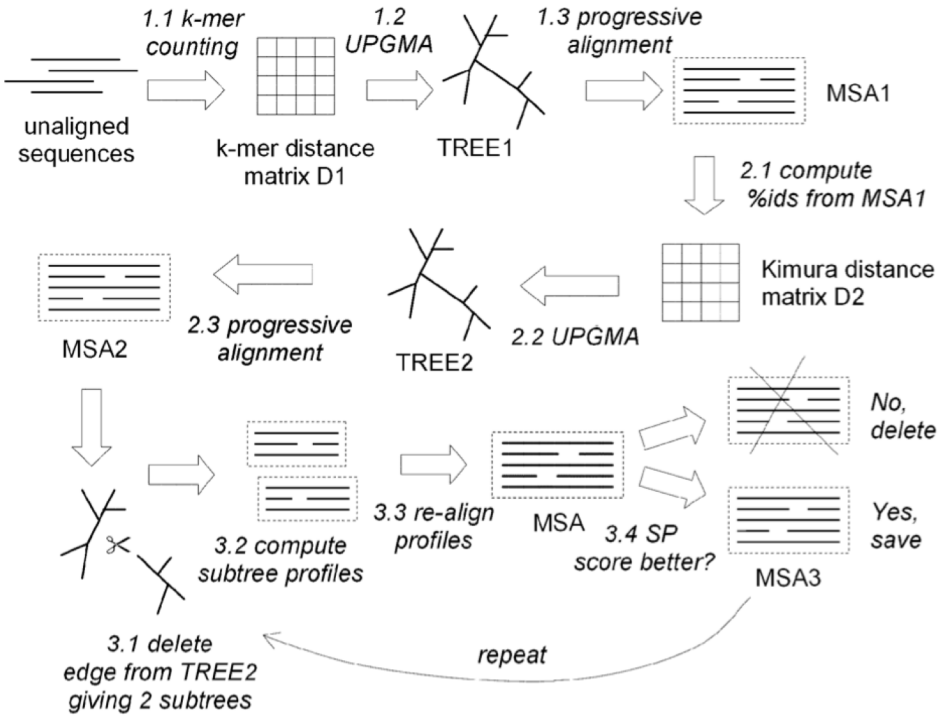
Stochastic iterative alignment
MUSCLE: Steps 1 - 2



1) generate initial alignment
    1) compute pairwise kmer distance to produce distance matrix D1
    2) use UPGMA* clustering to produce guide tree1
    3) perform progressive alignment along guide tree 1 producing MSA1
2) generate refined alignment
    1) compute pairwise corrected distances from MSA1 resulting in distance matrix D2
    2) use UPGMA* clustering to produce refined guide tree D2
    3) perform progressive alignment along guide tree 2 producing MSA2

*groups sequences according to similarity rather than according to evolutionary relationships

Stochastic iterative alignment
MUSCLE: Step3 – Iterative optimization



3) Optimization of alignment
    1) bisect guide tree by removing internal edge (edge chosen in order of decreasing distance from root)
    2) compute the profile (sub-alignment) for the sequences of each sub-tree
    3) align the two profiles and determine alignment score
    4) compare resulting score to previous score.
        1) If alignment score has increased, store optimized MSA together with score
        2) else discard
    5) Goto 1 unless convergence or maximum number of iterations reached.
4) Output optimized alignment

Consistency based alignment strategies (T-COFFEE)

The COFFEE strategy

Point: The optimal MSA is defined as the one that agrees the most with all optimal pair-wise alignments

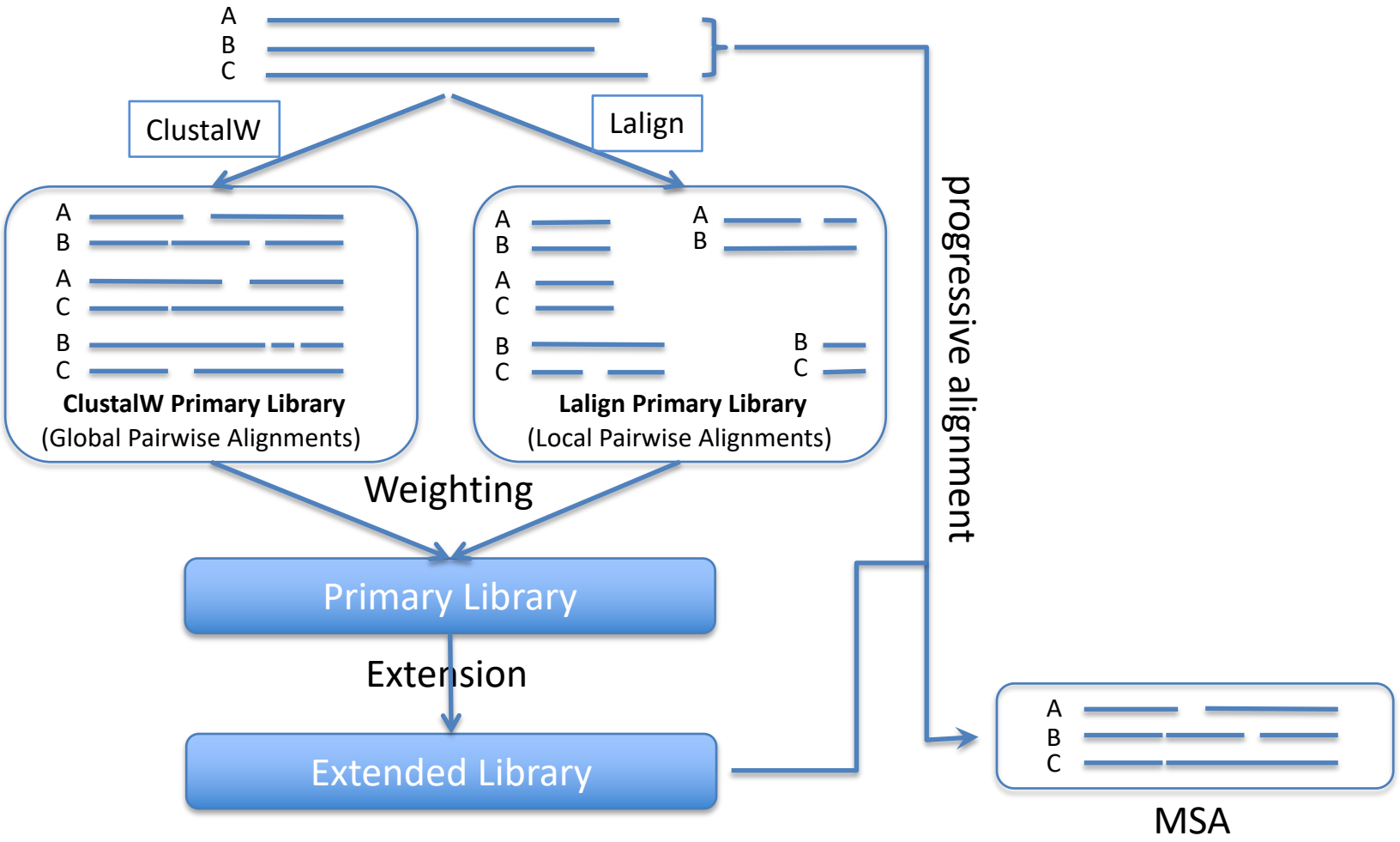
Features:

- does not depend on a specific scoring system
- can apply any method capable to align two sequences
- position dependent, i.e. the score associated with the alignment of two residues depends on their position within the sequence rather that their individual nature

Rationale: given a set of independent observations, the constellation most often observed is typically closer to the truth

Consistency based Objective Function For alignEment Evaluation (COFFEE)

Strategy of T-Coffee for aligning multiple sequences



Notredame et al. (2000) J Mol Biol **302:**205-217

T-Coffee: Primary Weighting

```
SeqA GARFIELD THE LAST FAT CAT
SeqB GARFIELD THE FAST CAT ---
```
88

```
SeqA GARFIELD THE LAST FA-T CAT
SeqC GARFIELD THE VERY FAST CAT
```
77[1]

```
SeqA GARFIELD THE LAST FAT CAT
SeqD -------- THE ---- FAT CAT
```
100

```
SeqB GARFIELD THE ---- FAST CAT
SeqC GARFIELD THE VERY FAST CAT
```
100

```
SeqB GARFIELD THE FAST CAT
SeqD -------- THE FA-T CAT
```
100

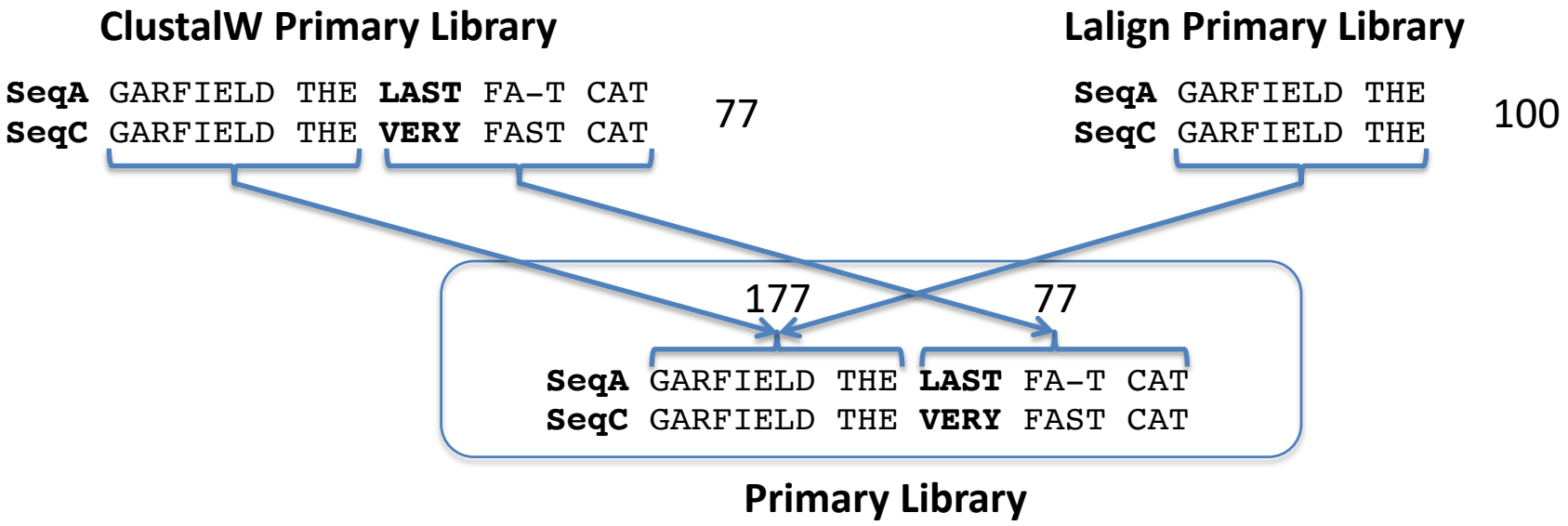
```
SeqC GARFIELD THE VERY FAST CAT
SeqD -------- THE ---- FA-T CAT
```
100

Compute primary weight for each pairing as the %identity from the alignment it comes from (matches/aligned positions * 100)

[1] This is the original weight from the publication that I cannot reproduce. I'm getting a weight of 80!

Pooling the two Libraries

Rule: If any residue pair is present in both libraries, it is merged into a single entry with a combined weight equal to the sum of the individual pairs.

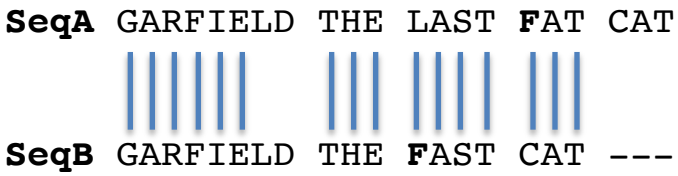


Note, non-observed residue pairings get a weight of 0

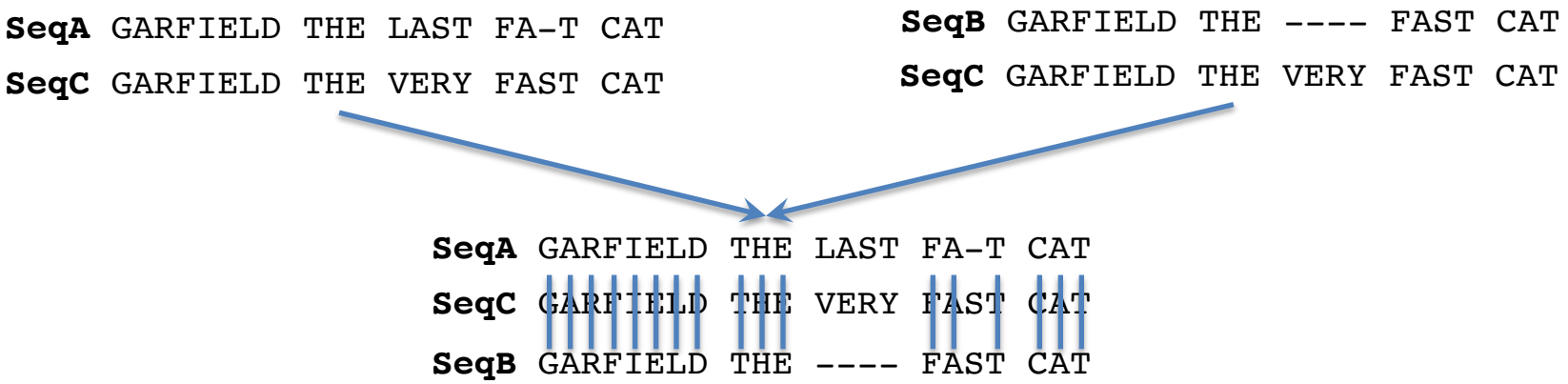
Extending the primary library

Follow a triplet approach: ie, look at the induced alignment A-B via C

We have one pair-wise alignment of sequences **A** and **B**.



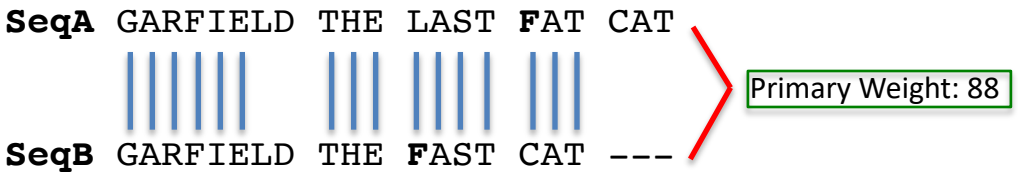
We have one indirect pair-wise alignment of sequences **A** and **B** via sequence **C**.



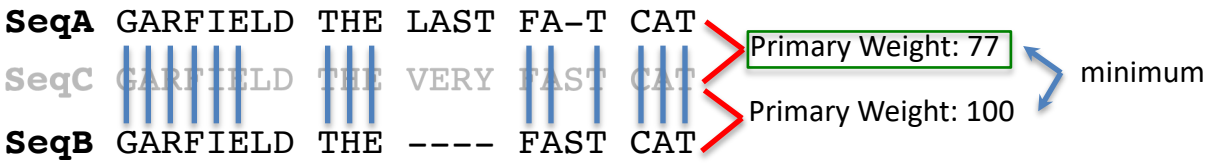
Extending the primary library

Follow a triplet approach: i.e., look at the induced alignment A-B via C

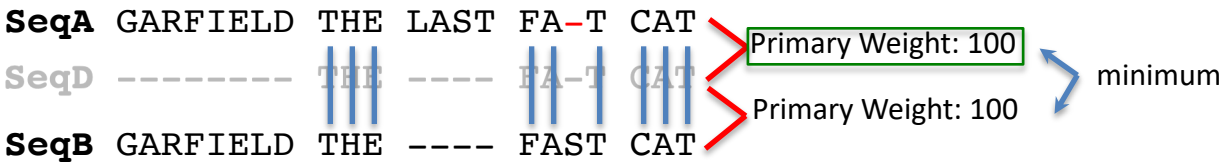
We have one pair-wise alignment of sequences **A** and **B**.



We have one indirect pair-wise alignment of sequences **A** and **B** via sequence **C**.



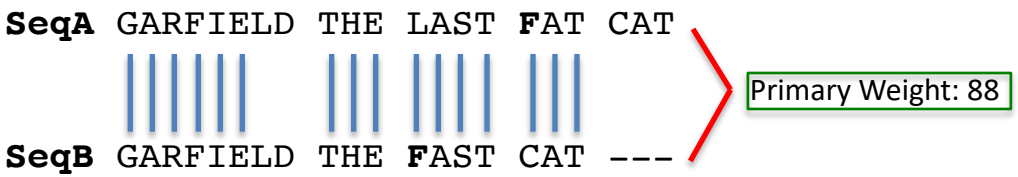
And we have one indirect pair-wise alignment of sequences **A** and **B** via sequence **D**.



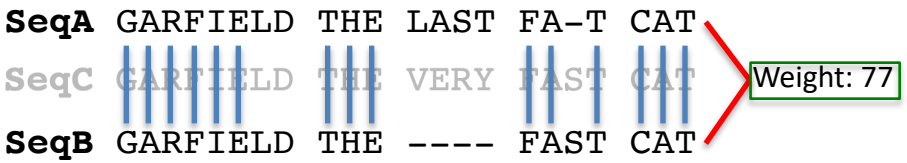
Extending the primary library

Follow a triplet approach: i.e., look at the induced alignment A-B via C

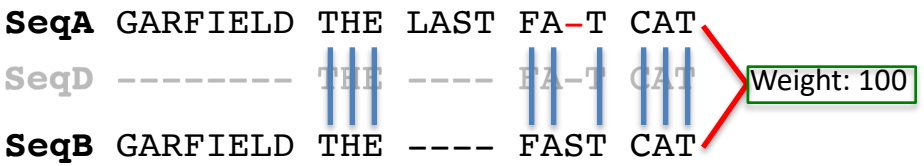
Pair-wise alignment of sequences **A** and **B**.



Indirect pair-wise alignment of sequences **A** and **B** via **C**.



Indirect pair-wise alignment of sequences **A** and **B** via **D**.



compute

Extended Library	
Pairing	Weight
$G_{A1} – G_{B1}$	165
$G_{A2} – G_{B2}$	165
.	.
.	.
$T_{A9} – T_{B9}$	265
H_{A10}- H_{B10}	265
.	.
$L_{A12} – F_{B12}$	88
.	.
.	.
$F_{A17} – C_{B17}$	88
$F_{A17} – F_{B13}$	177
…	…

Extending the primary library

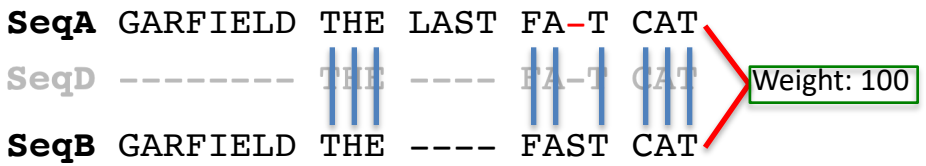
Follow a triplet approach: i.e., look at the induced alignment A-B via C

Pair-wise alignment of sequences **A** and **B**.

Use the extended library for the final scoring of the MSA.
Note, these are now position-specific scores[1]!

Indirect pair-

Indirect pair-wise alignment of sequences **A** and **B** via **D**.

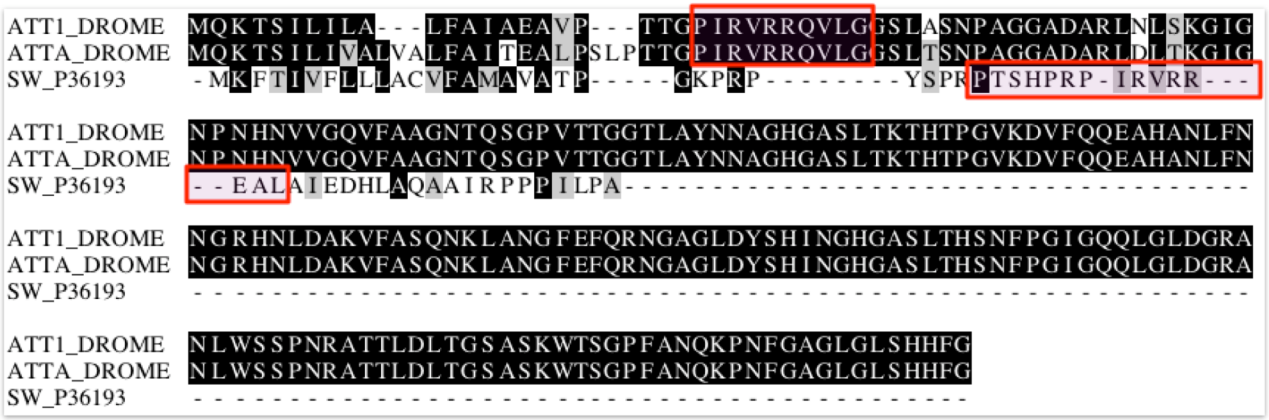


Extended Library	
Pairing	Weight
$G_{A1} – G_{B1}$	165
	165
	.
	.
	265
H_{A10}- H_{B10}	265
.	.
$L_{A12} – F_{B12}$	88
.	.
.	.
$F_{A17} – C_{B17}$	88
$F_{A17} – F_{B13}$	177
…	…

[1] we never had this before!

Different programs, different alignments, different biological conclusions

ClustalW



T-Coffee

Open questions

- Is the alignment *correct* ?
- Can I make it *better* ?
- Which programs are *best* ?
- How do you *know* if its correct ?

Open questions

- Is the alignment *correct* ?

  Define correct! But at least there is software available to assess the 'stability' of an alignment, i.e. is the alignment the same when I reverse the sequences.

- Can I make it *better* ?

  Define better!

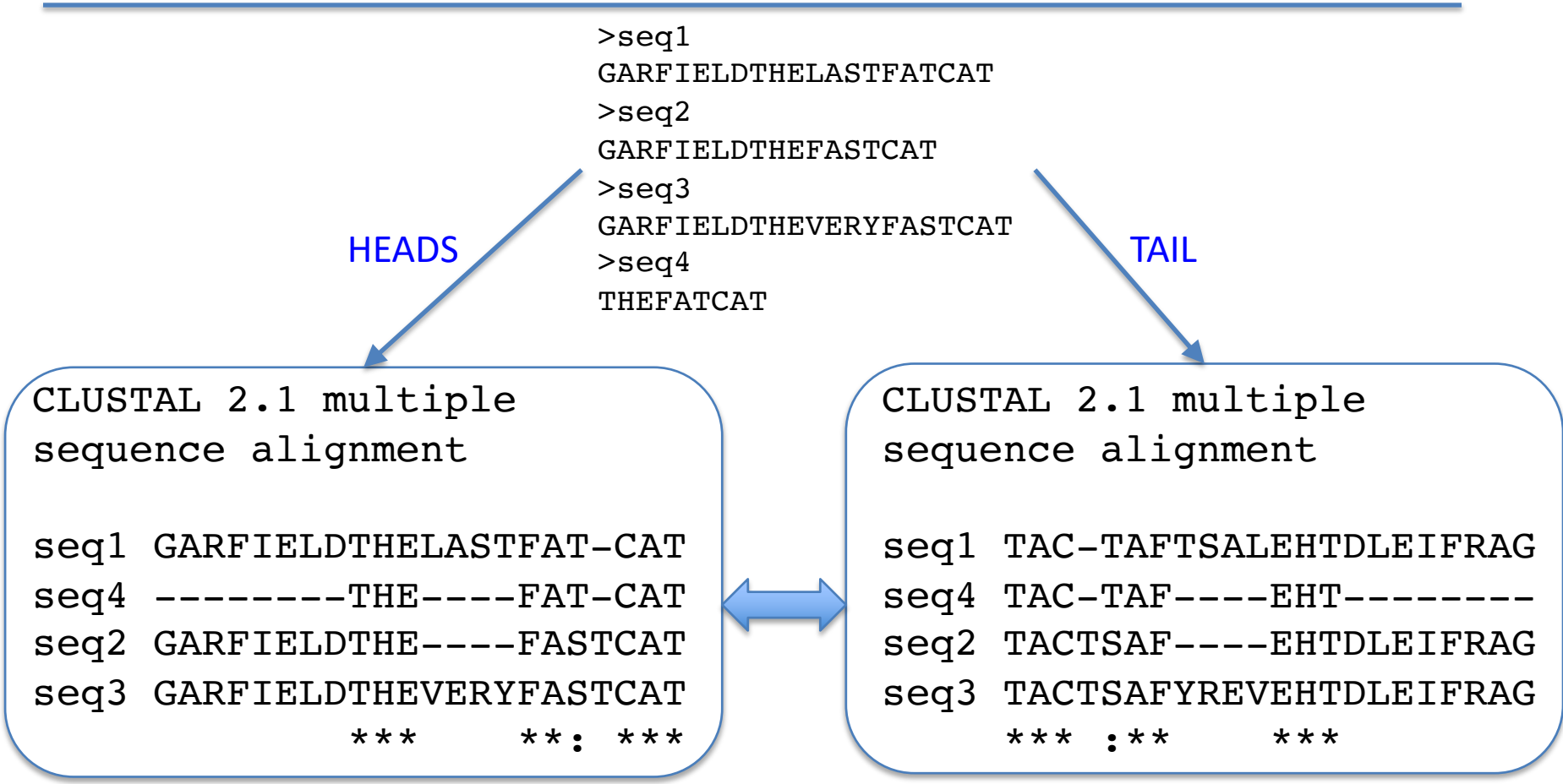
- Which programs are *best* ?

  It depends…

- How do you *know* if its correct ?

  Structural information, Biology

Heads or tails: a simple reliability check for multiple sequence alignments.



In essence: Consider pairings of amino acids in alignment columns more reliable, if they are observed both in the Heads and the Tails alignment.

Landan and Graur (2007) Heads or tails: a simple reliability check for multiple sequence alignments.