Bioinformatique M1: Lecture 6
P. Derreumaux

ALIGNEMENT MULTIPLE DE SEQUENCES


An example of Multiple Alignment
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWWSNG--

Multiple sequence alignment: features

- some aligned residues, such as cysteines that form disulfide bridges, may be highly conserved
- there may be conserved motifs such as a transmembrane domains or signal sequences
- there may be conserved secondary structure features
- there may be regions with consistent patterns of insertions or deletions (indels)

. There may be functional and folding reasons

- One of the most essential tools in molecular biology

It is widely used in:

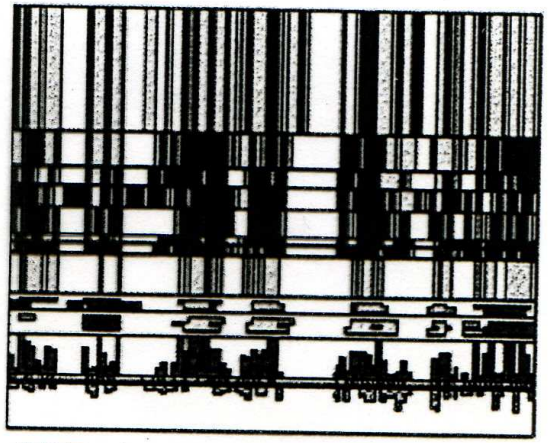
- Phylogenetic analysis
- Prediction of protein secondary/tertiary structure
- Finding diagnostic motifs patterns to characterize protein families
- Detecting new homologies between new genes and established sequence families

– Deriving Profiles

– homology modelling

Multiple Sequence Alignments

- Practically useful methods only since 1987
- Before 1987 they were constructed by hand
- The basic problem: no dynamic programming approach can be used
- First useful approach by D. Sankoff (1987) based on phylogenetics



(LEFT, adapted from Sonhammer et al. (1997). 'Pfam," Proteins 28:405-20. ABOVE, G Barton AMAS web page)

Ideal Multiple Sequence Alignment (MSA)

- **Fast**

  * for 2 sequences of L AA, Time/memory $\propto L^2$ using Dynamic Programming (NW).

  * for N sequences of L AA, Time $\propto L^N$

    e.g.

N	Time
3	2.5 min
4	6.25 hrs
5	39 days
7	2407 years.

- **Simple**

- **Accurate**

  good MSA $\longleftrightarrow$ good phylogenetic tree 'dendrogramme'

- **Informative**

  Two extreme cases { 4 Seq with 99% + 1 Seq 40%
  5 Seq with 5% identity sequence.

Various Methods

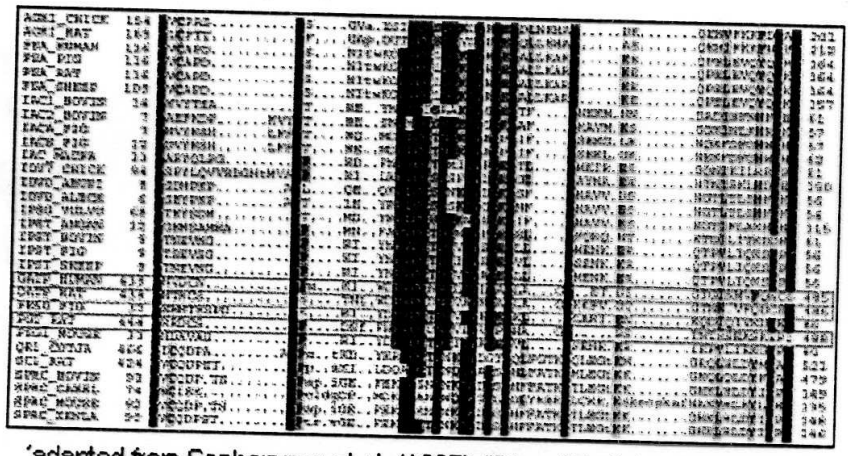
- Branch and Bound (guaranteed to find solution for 7 sequences)
- Divide and Conquer
- Genetic Algorithm
- HMM
- methods used to cluster the sequences into the most related groups (e.g PIMA, MAXHOM) or into a phylogenetic tree (e.g. ALIGN, GCG PILEUP and notably CLUSTAL W)

All these methods are progressive in character

Progressive approach pioneered by Feng, Doolittle (1987)

Progressive Multiple Alignments

- Most multiple alignments based on this approach
- Initial guess for a phylogenetic tree based on pairwise alignments
- Built progressively starting with most closely related sequences
- Follows branching order in phylogenetic tree
- Sufficiently fast
- Sensitive
- Algorithmically heuristic, no mathematical property associated with the alignment
- Biologically sound, it is common to derive alignments which are impossible to improve by eye



(adapted from Sonhammer et al. (1997). "Pfam," Proteins 28:405-20)

GLOBAL ALIGNMENT

CLUSTALW Program [Thompson, Higgins and Gibson, 1994]

CLUSTALW is one widely used implementation of profile-based progressive multiple alignment.

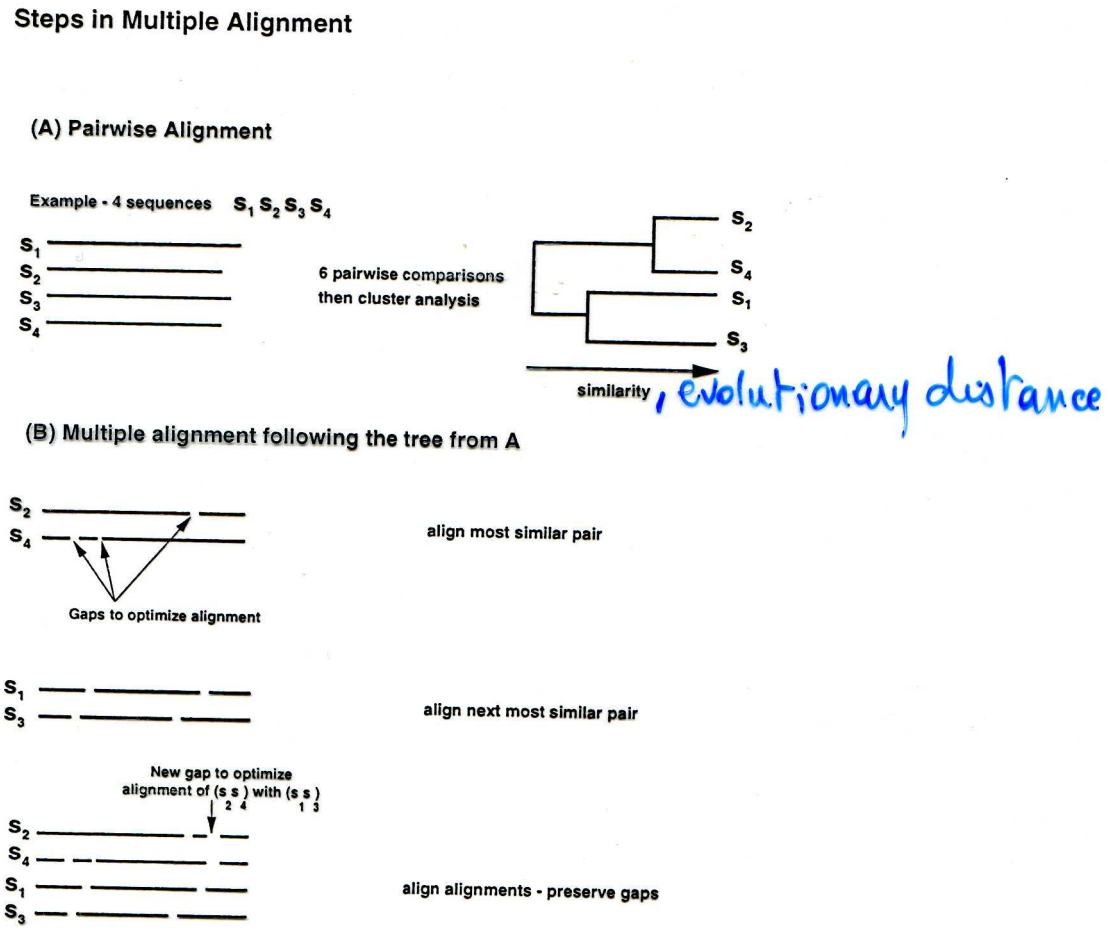
It is very similar to the Feng - Doolittle algorithm and it works as follows:

***1.** Construct a distance matrix of all N(N-1)/2 pairs of sequences by pairwise sequence alignment. Then convert the similarity scores to evolutionary distances using a specific model of evolution proposed by Kimura in 1983.*

***2.** Construct a guide-tree from this matrix using a clustering method called neighbor-joining proposed by Saitou and Nei in 1987.*

***3.** Progressively align nodes of the tree in order of decreasing similarity using sequences vs sequences, sequences vs profile and profile vs profile alignments.*

Figure 14: Progressive Alignment



Scoring along a tree is the main alternative to the simple "sum-of-pairs" cost model; only pairs of sequences that are adjacent (neighboring) in the tree are taken into consideration (or, at least they're weighted higher). Indeed, by weighting the pairs differently, we can score along a tree, yet employ Carrillo-Lipman and try out all possibly optimal alignment paths in the hyperlattice, see [AlL89] ! "Tree Alignment" subsumes methods that involve reconstructing ancestral sequences, too.

Pairwise Scores: S

= the number of identities in the best alignment divided by the number of residues compared (gap positions are excluded).

$$D = 1 - \frac{S}{100}$$

D: distance without correction for multiple substitutions

Scores can be calculated using dynamic programming (slow but accurate) or by the method of Wilbur and Lipman (extremely fast but approximate)

Models of evolution: The key is
Correcting for Multiple Substitution
at single sites.

Why?

This is because, as sequences
diverge, more than one substitution
will happen at many site.

However, you only see one difference
when you look at the present day
sequences.

Models for the Probability of Substitution Among Base Types

simplest

1. Base frequencies are equal and all substitutions are equally likely (Jukes-Cantor)

↓

2. Base frequencies are equal but transitions and transversions occur at different rates (Kimura 2 parameter)

↓

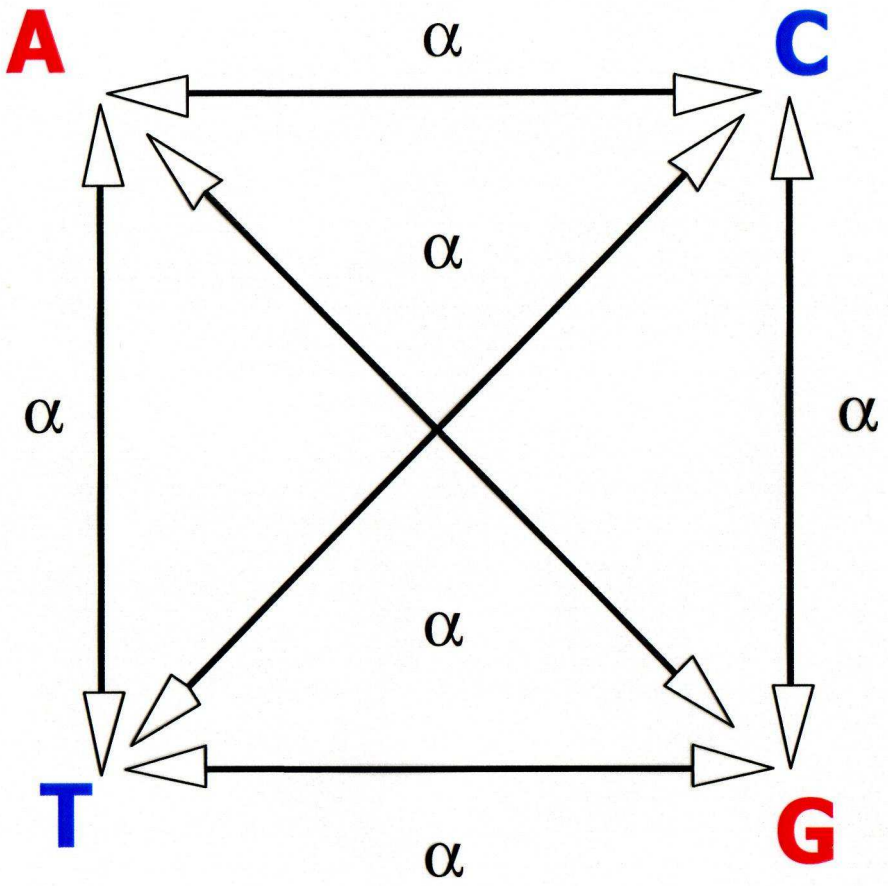
3. Unequal base frequencies and transitions and transversions occur at different rates (Hasegawa-Kishino-Yano)

↓

4. Unequal base frequencies and all substitution types occur at different rates (General Reversible Model)

Most complex

Jukes – Cantor Model



All substitutions occur at the same rate (α)

Models of DNA Substitution: (Jukes-Cantor, 1969)

- Assumptions:
  - i. All bases evolve independently
  - ii. All bases are at equal frequency
  - iii. Each base can change with equal probability (α)
  - iv. Mutations arise according to a Poisson distribution (rare and independent events)

- From this the number of substitutions per site (d) can be estimated by;

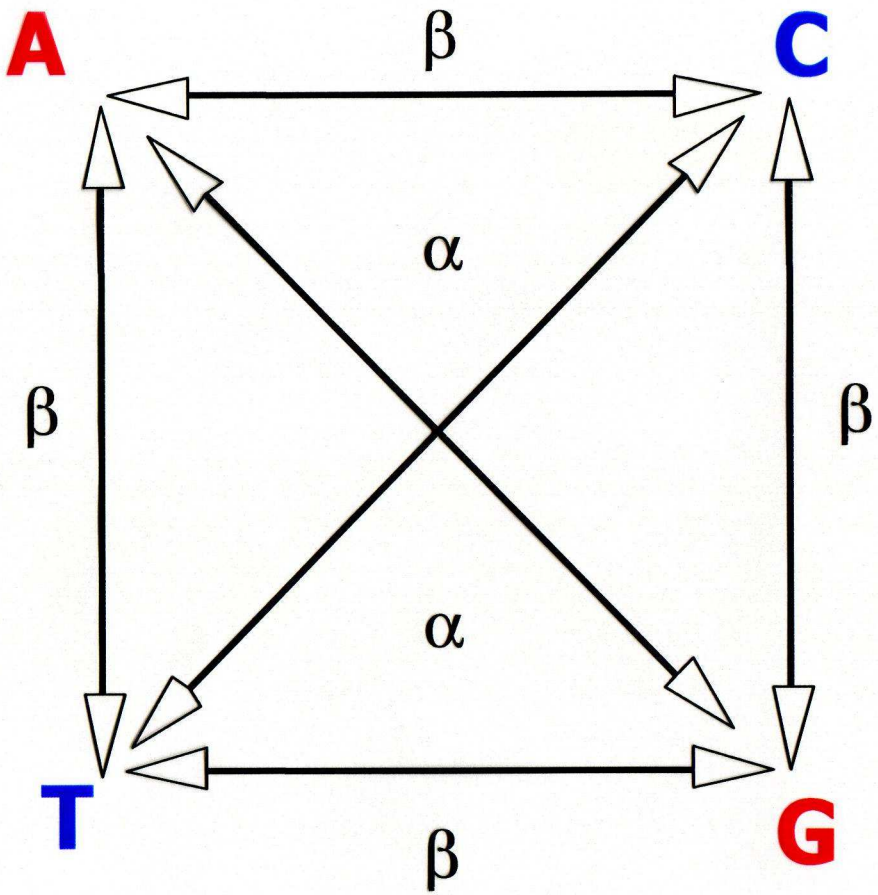
$$d = -3/4 \ln (1-4/3P)$$

where P is the proportion of observed nucleotide differences between 2 sequences.

The analogous model for amino acid sequences is

$$d = -19/20 \ln \left(1 - \frac{20}{19} \cdot P\right)$$

Kimura model (2 parameters)



Transitions (α) and transversions (β) occur at a different rate

Models of DNA Substitution: (Kimura, 1980)

- Assumptions:
  - i. All bases evolve independently
  - ii. All bases are at equal frequency
  - iii. Transitions and transversions occur with different probabilities (α and β)
  - iv. The Jukes-Cantor model is applied to transitions and transversions independently

- From this the expected number of substitutions per site (d) can be estimated by;

$$d = -1/2 \ln (1-2P-Q) - 1/4 \ln (1-2Q)$$

where P is the proportion of observed transitions and Q the proportion of observed transversions between 2 sequences

Construction Arbres

- méthodes phénétiques (fondées sur les distances)
  = méthodes de classification par regroupement successifs

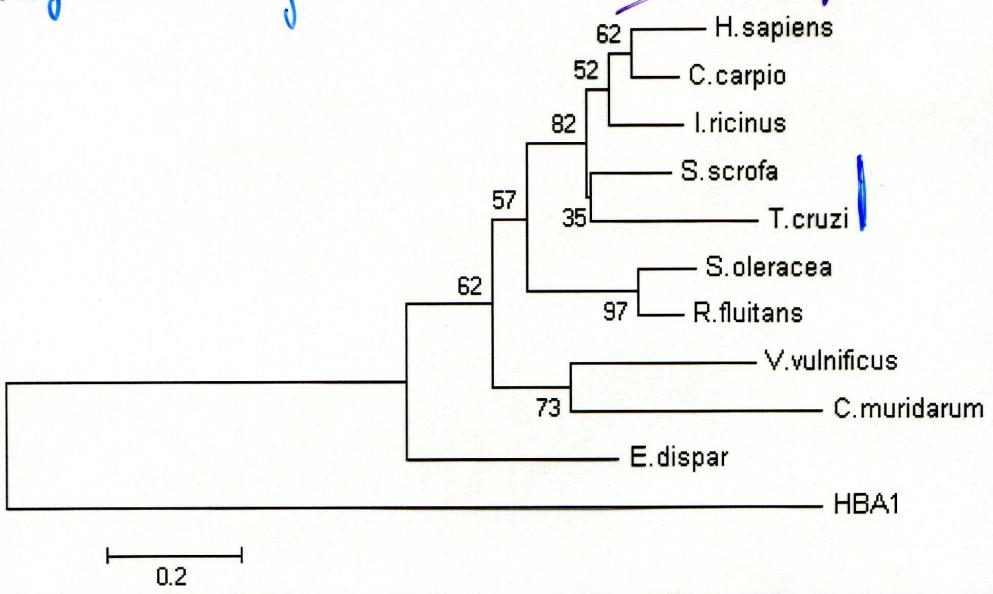
  ex : – UPGMA (Unweight Pair Group with Arithmetic mean)
  – NEIGHBOR-JOINING. (used by CLUSTALW)
  – FITCH

- méthodes cladistiques (fondées sur les séquences)
  s'intéressent au nombre de mutations (Indels, substitutions) qui affectent chacun des sites (positions) de la séquence
  → méthode de parcimonie
  → méthode de compatibilité
  → méthode de vraisemblance maximum.

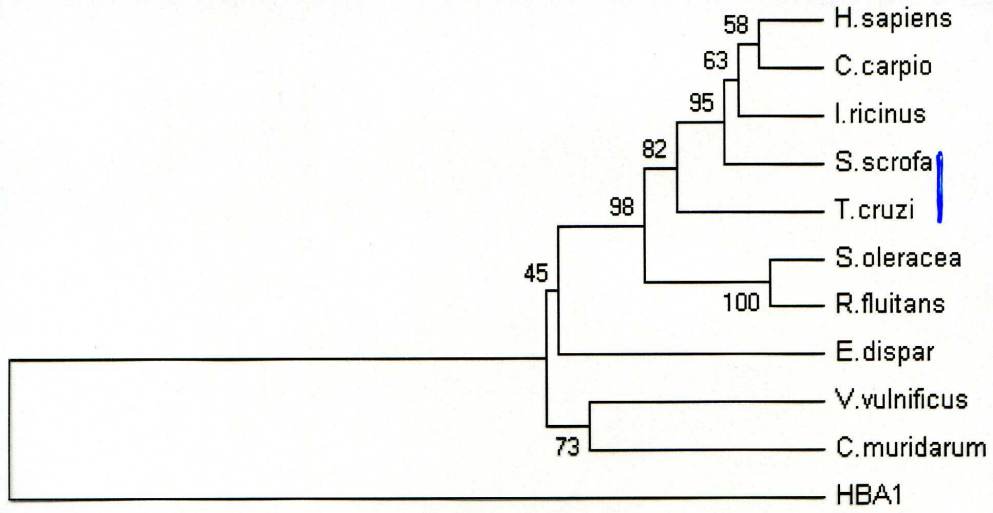
≠ arbres selon méthodes utilisées.

Neighbor Joining

boostrap values



Maximum Likelihood



Test on 10 homologues of PRDX1
(HBA1 outlier group)

Problems with Progressive Alignments

1. Local Minimum Problem

- It stems from greedy nature of alignment (mistakes made early in alignment cannot be corrected later) "Once a gap, always a gap"
- A better tree gives a better alignment (UPGMA<neighbour-joining tree method)

↳ Statistical methods for evaluating calculated trees.

Multiple Alignment- First pair

- Align the two most closely-related sequences first.
- This alignment is then 'fixed' and will never change. If a gap is to be introduced subsequently, then it will be introduced in the same place in both sequences, but their relative alignment remains unchanged.

Problems with Progressive Alignments.

2. Parameter Choice Problem

- It stems from using just one set of parameters (G, PAM or BLOSUM) (and hoping that they will do for all)

These parameters are not critical, as long as the sequences are rather similar.

If the sequences are however dissimilar, the results are heavily parameter-dependent.

The authors of Clustal W (W means Weight) try to circumvent the problem of setting the parameters adequate.

ClustalW weighs the sequences according to the distance of each sequence from the root.

- **Initial gap penalties**

Initially, two gap penalties are used: a gap opening penalty (GOP) which gives the cost of opening a new gap of any length and a gap extension penalty (GEP) which gives the cost of every item in a gap. Initial values can be set by the user from a menu. The software then automatically attempts to choose appropriate gap penalties for each sequence alignment, depending on the following factors.

1) Dependence on the weight matrix

It has been shown (16,28) that varying the gap penalties used with different weight matrices can improve the accuracy of sequence alignments. Here, we use the average score for two mismatched residues (ie. off-diagonal values in the matrix) as a scaling factor for the GOP.

2) Dependence on the similarity of the sequences

The percent identity of the two (groups of) sequences to be aligned is used to increase the GOP for closely related sequences and decrease it for more divergent sequences on a linear scale.

3) Dependence on the lengths of the sequences

The scores for both true and false sequence alignments grow with the length of the sequences. We use the logarithm of the length of the shorter sequence to increase the GOP with sequence length.

Using these three modifications, the initial GOP calculated by the program is:

GOP->(GOP+log(MIN(N,M))) * (average residue mismatch score) * (percent identity scaling factor) where N, M are the lengths of the two sequences.

4) Dependence on the difference in the lengths of the sequences

The GEP is modified depending on the difference between the lengths of the two sequences to be aligned. If one sequence is much shorter than the other, the GEP is increased to inhibit too many long gaps in the shorter sequence. The initial GEP calculated by the program is:

GEP -> GEP*(1.0+|log(N/M)|) where N, M are the lengths of the two sequences.

- **Position-specific gap penalties**

In most dynamic programming applications, the initial gap opening and extension penalties are applied equally at every position in the sequence, regardless of the location of a gap, except for terminal gaps which are usually allowed at no cost. In CLUSTAL W, before any pair of sequences or prealigned groups of sequences are aligned, we generate a table of gap opening penalties for every position in the two (sets of) sequences. An example is shown in figure 3. We manipulate the initial gap opening penalty in a position specific manner, in order to make gaps more or less likely at different positions.

The local gap penalty modification rules are applied in a hierarchical manner. The exact details of each rule are given below. Firstly, if there is a gap at a position, the gap opening and gap extension penalties are lowered; the other rules do not apply. This makes gaps more likely at positions where there are already gaps. If there is no gap at a position, then the gap opening penalty is increased if the position is within 8 residues of an existing gap. This discourages gaps that are too close together. Finally, at any position within a run of hydrophilic residues, the penalty is decreased. These runs usually indicate loop regions in protein structures. If there is no run of hydrophilic residues, the penalty is modified using a table of residue specific gap propensities (12). These propensities were derived by counting the frequency of each residue at either end of gaps in alignments of proteins of known structure. An illustration of the application of these rules from one part of the globin example, in figure 1, is given in figure 3.

details of gap penalties

1) Lowered gap penalties at existing gaps

If there are already gaps at a position, then the GOP is reduced in proportion to the number of sequences with a gap at this position and the GEP is lowered by a half. The new gap opening penalty is calculated as:

GOP -> GOP*0.3*(no. of sequences without a gap/no. of sequences).

2) Increased gap penalties near existing gaps

If a position does not have any gaps but is within 8 residues of an existing gap, the GOP is increased by:

GOP -> GOP*(2+((8-distance from gap)*2)/8)

3) Reduced gap penalties in hydrophilic stretches

Any run of 5 hydrophilic residues is considered to be a hydrophilic stretch. The residues that are to be considered hydrophilic may be set by the user but are conservatively set to D, E, G, K, N, Q, P, R or S by default. If, at any position, there are no gaps and any of the sequences has such a stretch, the GOP is reduced by one third.

4) Residue specific penalties

If there is no hydrophilic stretch and the position does not contain any gaps, then the GOP is multiplied by one of the 20 numbers in table 1, depending on the residue. If there is a mixture of residues at a position, the multiplication factor is the average of all the contributions from each sequence.

Méthodes d'evaluation des arbre

↳ Confiance en regard de la configuration de l'arbre.

Elles partent du postulat que les sites évoluent de manière indépendante les uns des autres.

- **Bootstrap**

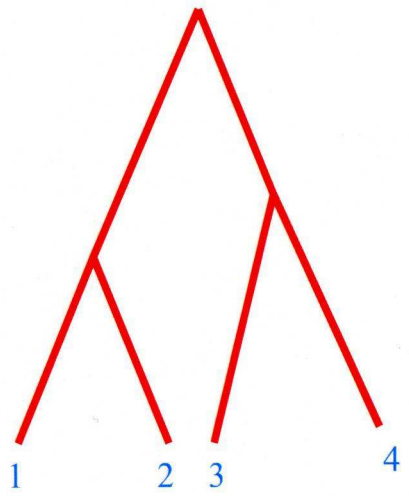
It involves making N random samples of sites from the alignment ($500 < N \leq 1000$); drawing N trees (1 from each sample and counting how many times each grouping from the original tree occurs in the sample trees.

- **Half Jacknife**

This technique resamples half of the sequence sites considered and eliminates the rest. it gives results very similar to those obtained by bootstrap.

Assessing Reliability: Bootstrap

Say we've inferred the following tree



Would like to get confidence levels that 1 & 2 belong together, and 3&4 belong together

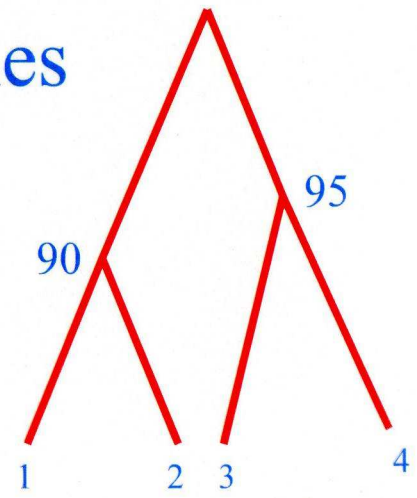
Assessing Reliability: Bootstrap

Say chose 6th, 1st, 6th, 8th, …

	12345678		6168 …
1	GCAGTACT		AGAT …
2	GTAGTACT	→	AGAT …
3	ACAATACC		AAAC …
4	ACAACACT		AAAT …

Assessing Reliability: Bootstrap

- Use pseudosample to construct tree
- Repeat many times
- Confidence of (1) and (2) together is fraction of times they appear together in trees generated from pseudosamples

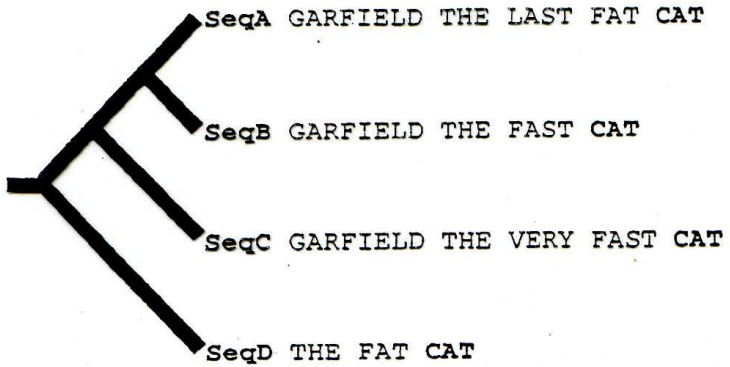


There are some specific cases where ClustalW is know to **have problems**.

- If the sequences are **similar only in some smaller regions**, while the larger parts are not recognisably similar, then ClustalW may have problems aligning all sequences properly. This is because ClustalW tries to find global alignments, not local. In such a case, it may be wise to cut out the similar parts with some other tool (text editor).
- If one sequence contains a **large insertion** compared to the rest, then there may be problems, for much the same reason as the previous point.
- If one sequence contains a **repetitive element** (such as a domain), while another sequence only contains one copy of the element, then ClustalW may split the single domain into two half-domains to try to align the first half with the first the domain in the first sequence, and the other half to the second domain in the first sequence. There are many proteins that contain multiple, very similar copies of a domain, so one swhould watch out for this.

T-coffee : global vs. local alignment.

a)Regular Progressive Alignment Strategy




```
SeqA GARFIELD THE LAST FA-T CAT
SeqB GARFIELD THE FAST CA-T ---
SeqC GARFIELD THE VERY FAST CAT
SeqD -------  THE ----  FA-T CAT
```

Progressive

b)Primary Library

```
SeqA GARFIELD THE LAST FAT CAT   Prim. Weight = 88
SeqB GARFIELD THE FAST CAT ---

SeqA GARFIELD THE LAST FA-T CAT  Prim. Weight = 77
SeqC GARFIELD THE VERY FAST CAT

SeqA GARFIELD THE LAST FAT CAT   Prim. Weight =100
SeqD -------  THE ---- FAT CAT

SeqB GARFIELD THE ---- FAST CAT  Prim Weight = 100
SeqC GARFIELD THE VERY FAST CAT

SeqB GARFIELD THE FAST CAT       Prim. Weight = 100
SeqD -------  THE FA-T CAT

SeqC GARFIELD THE VERY FAST CAT  Prim. Weight = 100
SeqD -------  THE ---- FA-T CAT
```

b)Primary Library

```
SeqA GARFIELD THE LAST FAT CAT
SeqB GARFIELD THE FAST CAT ---        Prim. Weight = 88

SeqA GARFIELD THE LAST FA-T CAT
SeqC GARFIELD THE VERY FAST CAT       Prim. Weight = 77

SeqA GARFIELD THE LAST FAT CAT
SeqD -------  THE ---- FAT CAT        Prim. Weight =100

SeqB GARFIELD THE ---- FAST CAT
SeqC GARFIELD THE VERY FAST CAT       Prim Weight = 100

SeqB GARFIELD THE FAST CAT
SeqD -------  THE FA-T CAT           Prim. Weight = 100

SeqC GARFIELD THE VERY FAST CAT
SeqD -------  THE ---- FA-T CAT      Prim. Weight = 100
```

c)Extended Library for seqA and seqB

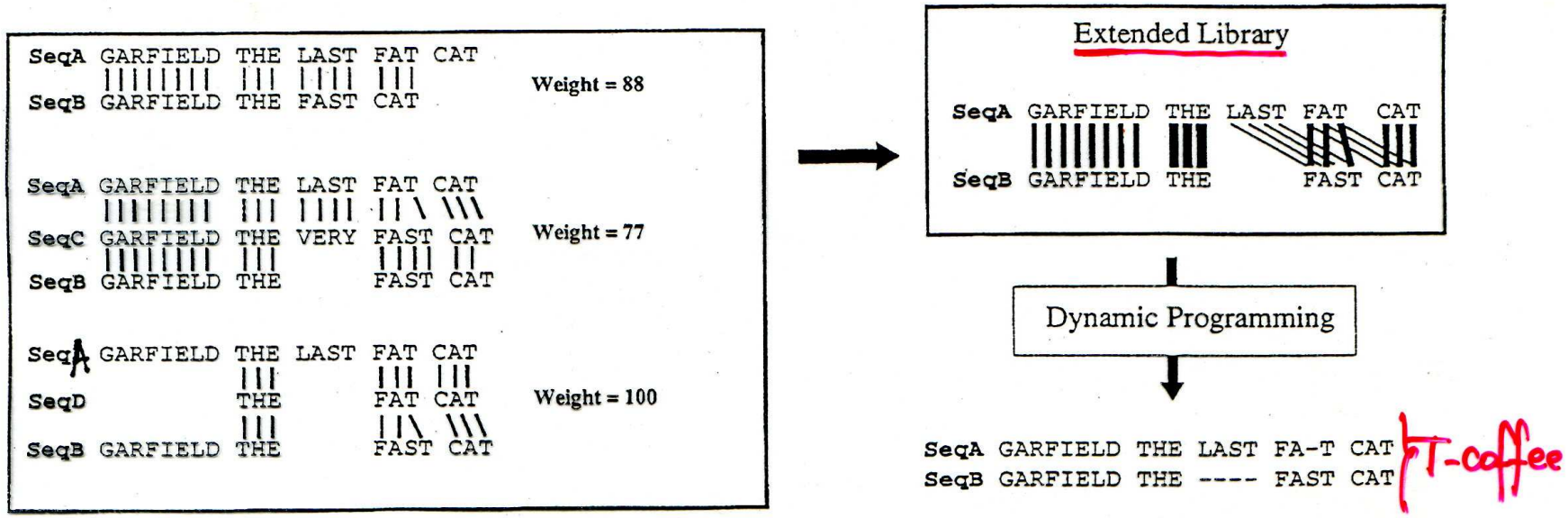


Figure 2. The library extension. (a) Progressive alignment. Four sequences have been designed. The tree indicates the order in which the sequences are aligned when using a progressive method such as ClustalW. The resulting alignment is shown, with the word CAT misaligned. (b) Primary library. Each pair of sequences is aligned using ClustalW. In these alignments, each pair of aligned residues is associated with a weight equal to the average identity among matched residues within the complete alignment (mismatches are indicated in bold type). (c) Library extension for a pair of sequences. The three possible alignments of sequence A and B are shown (A and B, A and B through C, A and B through D). These alignments are combined, as explained in the text, to produce the position-specific library. This library is resolved by dynamic programming to give the correct alignment. The thickness of the lines indicates the strength of the weight.

Available at EBI
(www.ebi.ac.uk)

- Clustal W 2
- T-coffee
- MAFFT (Multiple Alignment using Fast Fourier Transform)
- MUSCLE (Multiple (Alignment) Sequence Comparison by Log-Expectation)

Multiple Genome Alignment

- MGA, MAUVE, etc...

Common Mistakes in MSA