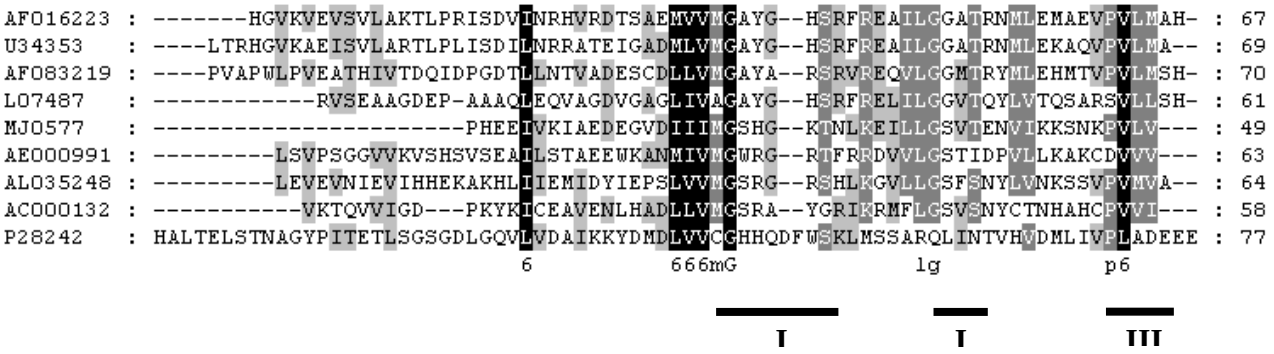
4 MULTIPLE SEQUENCE ALIGNMENTS

4.1. Introduction

Usually sequences either protein or DNA come in families. Sequences in a family have diverged from each other in their primary sequence during evolution, having separated either by a duplication in the genome or by speciation giving rise to corresponding sequences in related organisms. In either case they normally retain a similar function. If you have already a set of sequences belonging to the same family you can perform a database search for more members using pairwise alignments with one of the known family members as the query sequence (e.g. blast). However pairwise alignments with any one of the members may not find sequences distantly related to the ones you already have. An alternative approach is to use statistical features of the whole set of sequences in the search. Such features can be captured by a multiple sequence alignment. Example shows a multiple alignment of a family of ORF280. Some residues involved in protein structure or function are more conserved and are likely signatures for the family. In this section distinct algorithms to perform multiple sequence alignment will be described. Once a good multiple alignment is constructed, this can be used either for inference of the phylogenetic relationships between the organisms (see tree construction) or can be used as a seed to construct a profile. Such profile captures the signature of the protein (DNA) family in a probabilistic way and can be used to screen databases for additional members of the family.
Scoring a multiple alignment



4.1.1 Assumptions

Almost all alignment methods assume that the individual columns of an alignment are statistically independent.

The scoring function that usually is adopted is the following: $S(m)=G+\sum_i S(m_i)$ m is multiple alignment

A score is attributed to all the columns and the gaps. Most multiple alignment algorithms use an affine gap score that pay a higher cost for opening a gap than for extending it.

Usually the statistical relationship between the individual sequences is complex (a phylogenetic tree that reflects the relationship can have many intermediate ancestors). The scoring problem is greatly simplified by assuming that sequences have been generated independently (i.e. besides assuming an independence between the columns of an alignment we assume that the residues within the column are independent. This last assumptions can be reasonable if representative members of a sequence family are carefully chosen. It is often the case though that the sample of sequences is biased and

certain evolutionary subfamilies are over and underrepresented. A variety of tree based weighting schemes have been developed to partially compensate for the defects of the sequence independence assumption (see also construction of the BLOSUM matrices).

Problem: assumption of evolutionary independence

Tree-based weighting schemes

4.1.2 Sum of pair scores

Columns of an alignment are scored by a sum of pairs (SP). The SP score for a column is defined as

$$S(m_i) = \sum_{k \le l} s(m_i^k, m_i^l) \quad (1)$$

where s(a,b) come from substitution scoring matrices such as PAM and BLOSUM

4.2. Multidimensional dynamic programming

It is possible to generalise pairwise dynamic programming to the alignment of N sequences. But when assuming that the sequences are roughly of the same length L the memory complexity of the multidimensional dynamic programming is $O(L^N)$ and the time complexity $O(2^N L^N)$.). Such implementations become very impractical for large multiple alignments.

4.3. Progressive alignment methods

This works by constructing a succession of pairwise alignments. Initially two sequences are chosen and aligned by standard pairwise alignment. The alignment is fixed. Then a third sequence is chosen and aligned to the first sequence and this process is iterated until all sequences have been aligned. The different progressive alignment strategies differ from each other in:

1) the way they order the sequences to do the alignment
2) in whether the progression involves only alignment of sequences to a single growing alignment or whether subfamilies are built up upon a tree structure and at certain points

alignments are aligned against alignments (e.g. when progressing a group of sequences has already been aligned. The question is how to add the next sequence to the alignment. In the first implementations the novel sequence is pairwise aligned to each of the existing set of aligned sequences and the highest scoring alignment is taken to continue. In the more advanced implementations such as clustalW, the groups of already aligned sequences are represented by a profile and the subsequent sequence is aligned to the profile.

3) in the procedure used to align and score sequences or alignments against existing alignments

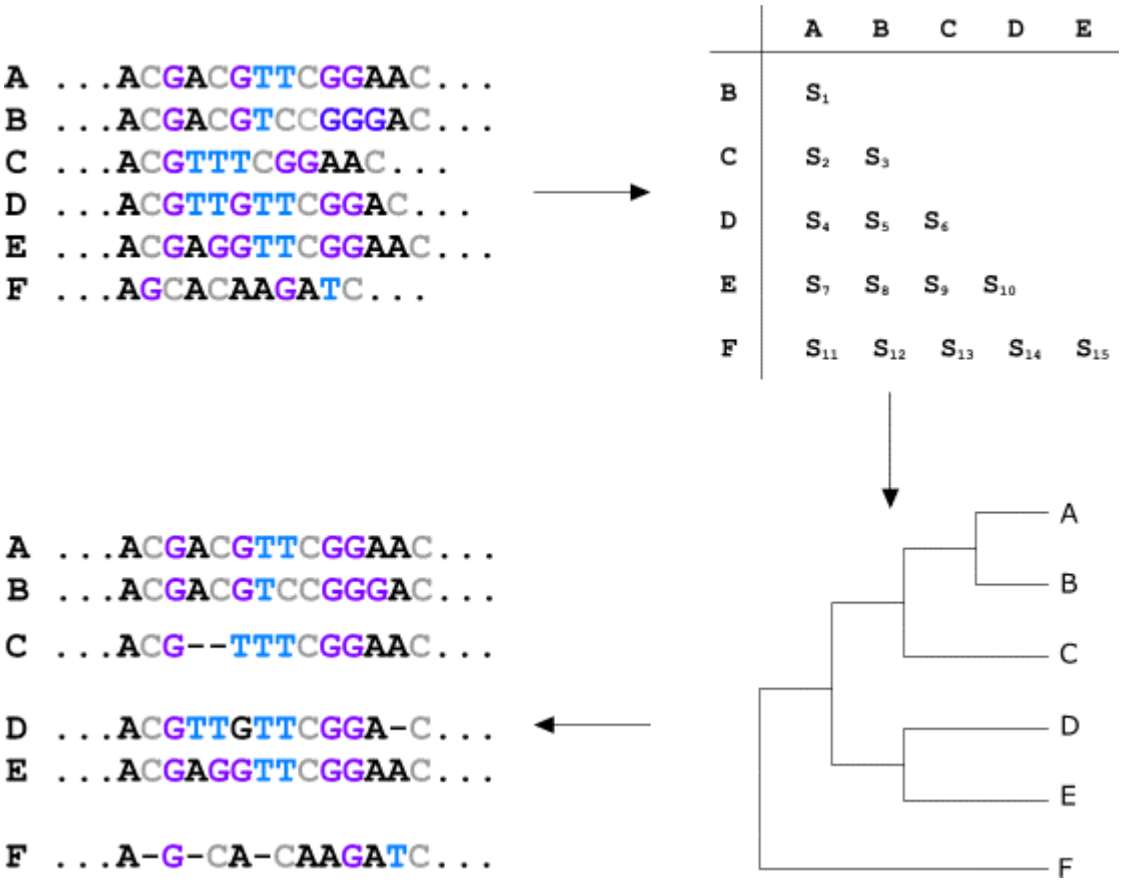
Progressive alignment is fast but heuristic: i.e. it does not guarantee to find the most optimal solution. The most important heuristic of progressive alignment is how to align the most similar pairs of sequences first. Most algorithms make use of a guide tree. This is a binary tree whose leaves represent sequences and whose interior nodes represent alignments. The root node represents a complete multiple alignment. The nodes furthest from the root represent the most similar pairs. The methods used to construct guide trees are similar to the methods to construct phylogenetic trees, but guide trees are typically "quick and dirty" trees unsuitable for serious phylogenetic inference.

4.3.1 ClustalW

CLUSTAL (CLUSTALV, CLUSTALW, CLUSTALX; available at ftp://ftp-igbmc.u-strasbg.fr/pub/ClustalX/) is without doubt the most widely used progressive alignment program:

- construct a distance matrix of all N(N-1)/2 sequence pairs by pairwise dynamic programming alignment followed by approximate conversion of similarity scores to evolutionary distances using the model of Kimura (1983).
- Construct a guide tree by a neighbor-joining (Saitou and Nei, 1987) clustering algorithm (see Evolutionary Analysis).
- Progressively align at nodes in order of decreasing similarity, using sequence-sequence, sequence-profile, and profile-profile alignment.

Thus the most closely related sequences are aligned first, and then additional sequences and groups of sequences are added, guided by the initial alignments to produce a multiple sequence alignment.
The initial (pairwise) alignments used to produce the **guide tree** may be obtained by a fast k-tuple or pattern finding approach similar to FASTA (see Homology Search) that is useful for many sequences, or a slower, full dynamic programming method may be used. Sequence alignment is then again based on mutation probability matrices such as those discussed above.
The pairwise sequence alignments will thus produce a set of genetic distances that can be used to construct a phylogenetic tree by a distance method such as neighbor-joining (see here for an example of the neighbor-joining method). On the basis of the guide tree, sequences will be aligned (see figure).



An online version of CLUSTALW is available here.
A first problem with the progressive sequence alignment method is the choice of suitable scoring matrices (see above) and gap penalties (Gap Opening Penalty and Gap Extension Penalty can be changed) that apply to the set of sequences. ClustalW has implemented quite advanced heuristics for the gap score e.g. gap open and gap extend penalties are increased if there are no gaps in a column but gaps in a nearby column (see below).

The major problem with progressive sequence alignment programs is the dependence of the ultimate multiple sequence alignment on the initial pairwise sequence alignments. The very first sequences to be aligned are the most closely related on the sequence tree. If these sequences align well, there will be few errors in the initial alignments. However, the more distantly related these sequences, the more errors will be made, and these errors will be propagated to the multiple sequence alignment. This problem is the "once a gap always a gap problem".Once a group of sequences has been aligned their alignment to each other can not be changed anymore at a later stage as more data arrive. Iterative refinement methods circumvent this problem:
An initial alignment is generated. Then one sequence (or a set of sequences) is taken out and realigned to a profile of the remaining aligned sequences. If the overall score increases this alignment is retained. This process is repeated until the alignment does not change anymore (PRRP, ftp.genome.ad.jp/pub/genome/saitamacc).

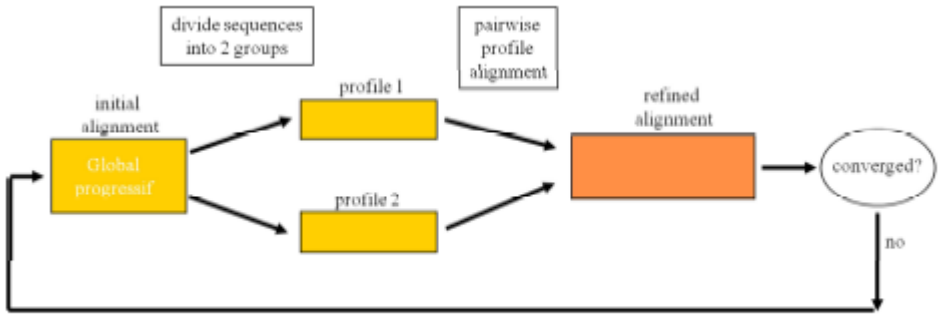


Fig. 6.7 Overview of iterative alignment methods.

ClustalW and other similar progressive alignment programs are useful to align related sequences. If more distantly related sequences need to be aligned, HMM might be more useful.

ClustalW2 (more detailed)

Probably the most well-known progressive alignment program is Clustal (www.ebi.ac.uk/tools/clustalw2). It has two variants: Clustal W, which provides a simple textbased interface, with the W standing for "weighting" to represent the ability of the program to Multiple sequence alignment provide weights to the sequence and program parameters; and Clustal X, which provides a more user-friendly graphical interface. It is already at version 2, Clustal W2, where the multiple sequence alignment programs have been completely rewritten in C++ in order to facilitate the further development of the alignment algorithms in the future and to allow proper porting of the programs to the latest versions of Linux, Macintosh and Windows operating systems. Clustal W2 allows faster alignment of large data sets and has increased alignment accuracy.

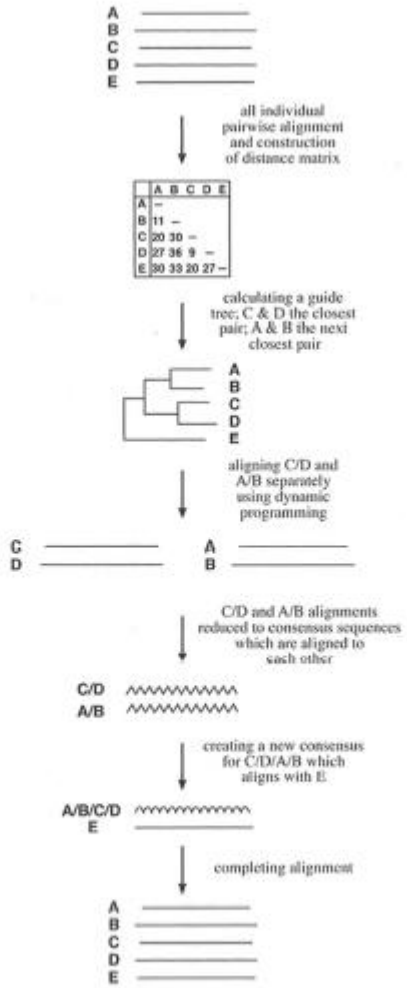


Fig. 6.4 Schematic of a typical progressive alignment procedure. Angled wavy lines represent consensus sequences for sequence pairs A/B and C/D. Curved wavy lines represent a consensus for A/B/C/D.

The different steps used by ClustalW2:

1. Perform pairwise alignments of all the sequences. This may be done by an extremely fast, but approximate k-tuple or pattern-finding approach similar to FASTA, or by a slow, but accurate full dynamic programming method for global alignment. Pairwise scores are calculated as the number of identities in the best alignment divided by the number of residues compared (gap positions are excluded). These are converted to distances by dividing by 100 and subtracting from 1.0 to give number of differences per site. As the pairwise score is calculated independently of the matrix and gaps chosen, it will always be the same value for a particular pair of sequences.

2. Use the alignment scores to produce an approximate phylogenetic tree. The guide tree can be produced by neighbor-joining method or UPGMA (Unweighted Pair Group Method with Arithmetic Mean), which is a simple agglomerative or hierarchical clustering method. The latter is marginally less accurate, but on large alignments (e.g. 10 000 globin sequences) this is offset by the savings in processing time (2 h versus 12 h).

3. Align the sequences sequentially using the dynamic programming algorithm for global alignment, guided by the phylogenetic relationships indicated by the guide tree. Four series of

substitution matrices can be chosen (BLOSUM, PAM, Gonnet and an identity matrix, which gives a score of 10 to two identical residues and a score of zero otherwise). Within a chosen serie of substitution matrices, Clustal applies different substitution matrices when aligning sequences, depending on the evolutionary distances measured from the guide tree. For example in the BLOSUM serie, for closely related sequences that are aligned in the first steps, Clustal will use the BLOSUM62 matrix; while for more divergent sequences that are aligned in later steps of progressive alignment, the BLOSUM45 matrix may be used instead. Another feature is the use of **adjustable gap penalties** that allow more insertions and deletions in regions that are outside conserved domains, but fewer in conserved regions. In addition, gaps that are too close to one another can be penalized more than gaps occurring in isolated loci. Position-specific gap penalties bias alignment algorithms toward placing Multiple sequence alignment gaps where previous gaps were opened during each pairwise merge step. Here, the rationale is that gap opening events that occur simultaneously in a group of sequences likely represent a single evolutionary event and hence should not be overpenalized. In addition, for globular protein sequences, hydrophobic residues are abundant in core regions where sequence indels are likely to affect proper folding, whereas hydrophilic residues are abundant on the protein surface, where extra loops are more likely to be tolerated. Contributions of sequences are weighted according to their relationships on the guided tree. **Redundant and closely related groups of sequences are down-weighted** in order to prevent them from dominating the alignment. The weight factor for each sequence is determined by its branch length on the guide tree. The branch lengths are normalized by how many times sequences share a basal branch from the root of the tree (Fig. 6.5). The alignment scores between two positions in the multiple sequence alignment are then calculated using the resulting weights as multiplication factors.

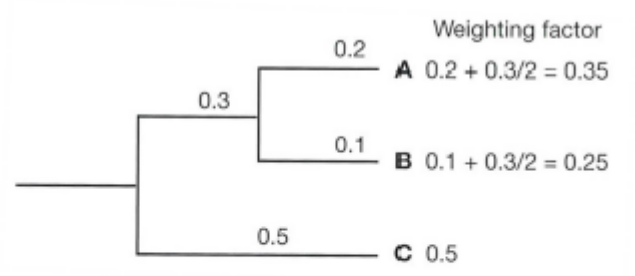


Fig. 6.5 Weighting scheme used by Clustal W2. Sequences that arise from a unique branch deep in the tree receive a weighting factor equal to the distance from the root. Other sequences that arise from branches shared with other sequences receive a weighting factor that is less than the sum of the branch lengths from the root. The weighting factors are normalized so that the largest weight is 1.

4.4. Recent developments

ClustalW has for a long time been the only frequently used multiple alignment program. ClustalW showed very appropriate for aligning related sequences (related protein sequences mainly). However, as more genomes become sequenced it becomes also interesting to align noncoding parts of a sequence (long DNA stretches) between the sequences of distinct organisms (e.g. phylogenetic footprinting (see comparative genomics). To this end distinct novel algorithms have been developed.

MLAGAN — an extension to LAGAN — and MAVID an extension to AVID enable the multiple alignment of large genomic sequences. It involves a progressive alignment phase, based on LAGAN (AVID), which first aligns the genomes of the most closely related organisms, then

incorporates the others in order of phylogenetic distance (this information is provided by the user). For example, when aligning human, mouse and rat, mouse and rat will be aligned first, followed by human. The application of multiple alignment to several whole-genomes has not previously been published. However, as the genome sequences of organisms such as rat, mouse and human are now available, all possible combinations of comparisons between them can be performed, and graphical interfaces such as mVISTA or mPIPmaker can be used to display the results, taking one species as a reference. Whole-genome multiple alignment seems to be the next challenge.

4.5. Editing multiple sequence alignments

Once an alignment has been generated, visualization tools allow manual identification of regions with reliably predicted homology; many of these tools also allow for interactive alignment editing. For alignments of sequences with low similarity, post-processing is extremely important as most regions in a low-identity alignment will not be reliably aligned. It is imperative that the user checks the alignment carefully for biological relevance and edit the alignment if necessary. This involved introducing or removing gaps to maximize biologically meaningful matches. Sometimes, portions that are ambiguously aligned and deemed incorrect have to be deleted. In manual editing, empirical evidence and mere experience is needed to make corrections on alignment. Typically, high confidence aligned regions can be identified by looking for groups of residues with strongly conserved physicochemical properties (e.g., hydropathy, polarity, and volume), using alternative alignment objective functions for identifying reliable columns, using the consensus of several alignment methods, or even better, cross-referencing aligned positions with amino acid residues in three-dimensional protein structures. **BioEdit** (www.mbio.ncsu.edu/bioedit/bioedit.html) is a multifunctional sequence alignment editor for Windows. It has a coloring scheme for nucleotide or amino acid residues that facilitates manual editing. In addition, it is able to do BLAST searches, plasmid drawing, restriction mapping, …(Fig. 6.15). **Jalview** (www.jalview.org) is a multiple alignment editor written in Java. It is used widely in a variety of web pages (e.g. the EBI Clustalw server and the Pfam protein domain database) but is available as a general purpose alignment editor.

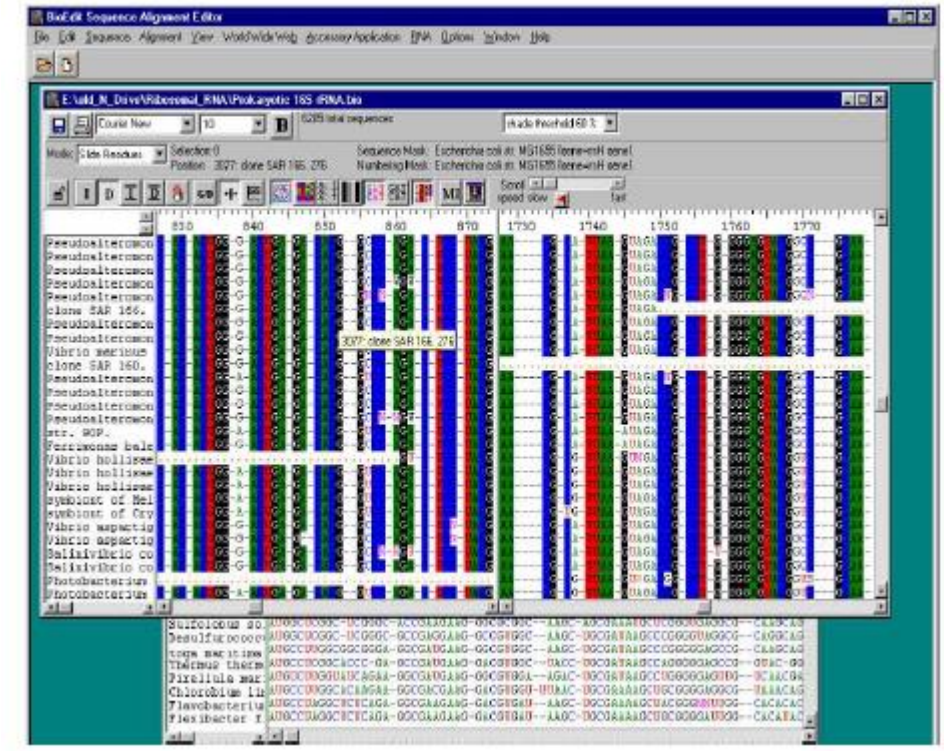


Fig. 6.15 *Screenshot from the BioEdit multiple sequence alignment editor.*