BIOINFORMATICS

Vol. 14 no. 5 1998
Pages 407–422

COFFEE: an objective function for multiple sequence alignments

Cédric Notredame[1], Liisa Holm[1] and Desmond G. Higgins[2]

[1]EMBL Outstation–The European Bioinformatics Institute, Hinxton Hall, Hinxton, Cambridge CB10 1SD, UK and [2]Department of Biochemistry, University College, Cork, Ireland

Received on January 19, 1998; revised and accepted on February 24, 1998

Abstract

Motivation: *In order to increase the accuracy of multiple sequence alignments, we designed a new strategy for optimizing multiple sequence alignments by genetic algorithm. We named it COFFEE (Consistency based Objective Function For alignmEnt Evaluation). The COFFEE score reflects the level of consistency between a multiple sequence alignment and a library containing pairwise alignments of the same sequences.*

Results: *We show that multiple sequence alignments can be optimized for their COFFEE score with the genetic algorithm package SAGA. The COFFEE function is tested on 11 test cases made of structural alignments extracted from 3D_ali. These alignments are compared to those produced using five alternative methods. Results indicate that COFFEE outperforms the other methods when the level of identity between the sequences is low. Accuracy is evaluated by comparison with the structural alignments used as references. We also show that the COFFEE score can be used as a reliability index on multiple sequence alignments. Finally, we show that given a library of structure-based pairwise sequence alignments extracted from FSSP, SAGA can produce high-quality multiple sequence alignments. The main advantage of COFFEE is its flexibility. With COFFEE, any method suitable for making pairwise alignments can be extended to making multiple alignments.*

Availability: *The package is available along with the test cases through the WWW: http://www.ebi.ac.uk/~cedric*

Contact: *cedric.notredame@ebi.ac.uk*

Introduction

Multiple alignments are among the most important tools for analysing biological sequences. They can be useful for structure prediction, phylogenetic analysis, function prediction and polymerase chain reaction (PCR) primer design. Unfortunately, accurate multiple alignments may be difficult to build. There are two main reasons for this. First of all, it is difficult to evaluate the quality of a multiple alignment. Secondly, even when a function is available for the evaluation, it is algorithmically very hard to produce the alignment having the best possible score (optimal alignment).

Cost functions or scoring functions roughly fall into two categories. First of all, there are those that rely on a substitution matrix. These are the most widely used. They require a substitution matrix (Dayhoff, 1978; Henikoff and Henikoff, 1992) that gives a score to each possible amino acid substitution, a set of gap penalties that gives a cost to deletions/insertions (Altschul, 1989), and a set of sequence weights (Altschul *et al.*, 1989; Thompson *et al.*, 1994b). Under this scheme, an optimal multiple alignment is defined as the one having the lowest cost for substitutions and insertion/deletions. One of the most widely used scoring methods of this type is the 'weighted sums of pairs with affine (or semi affine) gap penalties' (Altschul and Erickson, 1986). The main limitation of these scoring schemes is that they rely on very general substitution matrices, usually established by statistical analysis of a large number of alignments. These may not necessarily be adapted to the set of sequences one is interested in. To compensate for this drawback, a second type of scoring scheme was designed: profiles (Gribskov *et al.*, 1987) and Hidden Markov Models (HMMs) (Krogh and Mitchison, 1995). Profiles allow the design of a sequence-specific scoring scheme that will take into account patterns of conservation and substitution characteristic of each position in the multiple alignment of a given family. To some extent, HMMs can be regarded as generalized profiles (Bucher and Hofmann, 1996). In HMMs, sequences are used to generate statistical models. The sequences of interest are then aligned to the model one after another to generate the multiple sequence alignment. The main drawback of HMMs is that to be general enough, the models require large numbers of sequences. However, this can be partially overcome by incorporating in the model some extra information such as Dirichlet mixtures (the equivalent of a substitution matrix in an HMM context) (Sjolander *et al.*, 1996).

Whatever scoring scheme one wishes to use, the optimization problem may be difficult. There are two types of optimization strategies: the greedy ones that rely on pairwise alignments and those that attempt to align all the sequences

© Oxford University Press

simultaneously. The main tool for making pairwise alignments is an algorithm known as dynamic programming (Needleman and Wunsch, 1970) and is often used for optimizing the sums of pairs. The complexity of the algorithm makes it hard to apply it to more than two sequences (or two alignments) at a time. Nevertheless, it allows greedy progressive alignments as described by Feng and Doolittle (1987) or Taylor (1988). In such a case, the sequences are aligned in an order imposed by some estimated phylogenetic tree. The alignment is called progressive because it starts by aligning together closely related sequences and continues by aligning these alignments two by two until the complete multiple alignment is built. Some of the most widely used multiple sequence alignment packages like ClustalW (Thompson *et al.*, 1994a), Multal (Taylor, 1988) and Pileup (Higgins and Sharp, 1988) are based on this algorithm. They have the advantage of being fast and simple, as well as reasonably sensitive. Their main drawback is that mistakes made at the beginning of the procedure are never corrected and can lead to misalignments due to the greediness of the strategy. It is to avoid this pitfall that the second type of methods have been designed. They mostly involve aligning all the sequences simultaneously. For the sums of pairs, this is a difficult problem that has been shown to be NP-complete (Wang and Jiang, 1994). However, using the Carrillo and Lipman (1988) algorithm implemented in the Multiple Sequence Alignment program MSA (Lipman *et al.*, 1989), one can simultaneously align up to 10 sequences. Other global alignment techniques using the sums of pairs cost function involve the use of stochastic heuristics such as simulated annealing (Ishikawa *et al.*, 1993a; Godzik and Skolnik, 1994; Kim *et al.*, 1994), genetic algorithms (Ishikawa *et al.*, 1993b; Notredame and Higgins, 1996) or iterative methods (Gotoh, 1996). Simulated annealing can also be used to optimize HMMs (Eddy, 1995).

The stochastic methods have two main advantages over the deterministic ones. First of all they have a lower complexity. This means that they do not have strong limitations on the number of sequences to align or on the length of these sequences. Secondly, these methods are more flexible regarding the objective function they can use. For instance, MSA is restricted to an approximation of the sums of pairs using semi-affine gap penalties (Lipman *et al.*, 1989) instead of the natural ones shown to be biologically more realistic (Altschul, 1989). This is not the case with simulated annealing (Kim *et al.*, 1994). The main drawback of stochastic methods is that they do not guarantee optimality. However, in some previous work, we showed that with the Sequence Alignment Genetic Algorithm (SAGA), results similar to MSA could be obtained (Notredame and Higgins, 1996). We also showed that the package was able to handle test cases with sizes much beyond the scope of MSA. The robustness of SAGA as an optimizer was confirmed by results obtained on a different objective function for RNA alignment (Notredame *et al.*, 1997) and motivated our choice to use SAGA for optimizing the new objective function described here.

The main argument for aligning all the sequences simultaneously instead of making a greedy progressive alignment is that using all the available information should improve the final result. However, one limitation of such methods is that regions of low similarity may induce some noise that will weaken the signal of the correct alignment (Morgenstern *et al.*, 1996). In order to avoid this, one would like a scheme that filters some of the initial information and allows its global use. The approach we propose here is an attempt to do so. The underlying principle is to generate a set of pairwise alignments and look for consistency among these alignments. In this case, we define the optimal multiple alignment as the most consistent one and produce it using the SAGA package.

The idea of using the consistency information in a multiple sequence alignment context is not new (Gotoh, 1990; Vingron and Argos, 1991; Kececioglu, 1993). In his scheme, Gotoh (1990) proposed the identification of regions that are fully consistent among all the pairwise alignments. These regions are used as anchor points in the multiple alignment, in order to decrease complexity. A similar strategy was described by Vingron and Argos (1991), allowing the computation of a multiple alignment from a set of dot matrices. Although very interesting, these methods had several pitfalls, including a sensitivity to noise (especially when some sequences are highly inconsistent with the rest of the set) and a high computational complexity. The work of Kececioglu (1993) bears a stronger similarity to the method we propose here. Kececioglu directly addressed the problem of finding a multiple alignment that has the highest level of similarity with a collection of pairwise alignments. Such an alignment is named 'maximum weight trace alignment' (MWT), and its computation was shown to be NP-complete. An optimization method was also described, based on dynamic programming and limited to a small number of sequences (six maximum).

More recently, a method was described that allows the construction of a multiple alignment using consistent motifs identified over the whole set of sequences by a variation of the dynamic programming algorithm (Morgenstern *et al.*, 1996). This algorithm should be less sensitive to noise than the one described by Vingron and Argos, but its main drawback is that it does rely on a greedy strategy for assembling the multiple alignment.

An important aspect of multiple sequence alignment often overlooked is estimation of the reliability. Since all the alignment scoring functions available are known to be intrinsically inaccurate, identifying the biologically relevant portions of a multiple alignment may be more important than increasing the overall accuracy of this alignment. A few tech-

niques have been proposed to identify accurately aligned positions in pairwise (Vingron and Argos, 1990; Mevissen and Vingron, 1996) and multiple sequence alignments (Gotoh, 1990; Rost, 1997). We show here that our method allows a reasonable estimation of a multiple alignment local reliability. The measure we use for reliability is in fact very simple and could easily be extended much further to incorporate other methods such as the one described by Mevissen and Vingron (1996).

Methods

The overall approach relies on the definition of an objective function (OF) describing the quality of multiple protein sequence alignments. Given a set of sequences and an 'all-against-all' collection of pairwise alignments of these sequences (library), the score of a multiple sequence alignment is defined as the measure of its consistency with the library. This objective function was optimized with the SAGA package. Sets of sequences with a known structure and for which a multiple structural alignment is available were extracted from the 3D_ali database (Pascarella and Argos, 1992) and used in order to validate the biological relevance of the new objective function. Two other test cases were designed using the DALI server (Holm and Sander, 1996a) and aligned using libraries made of structural pairwise alignments extracted from the FSSP database (Holm and Sander, 1993).

Objective function

The OF is a measure of quality for multiple sequence alignments. Ideally, the better its score, the more biologically relevant the multiple alignment. The method proposed here requires two components: (i) a set of pairwise reference alignments(library), (ii) the OF that evaluates the consistency between a multiple alignment and the pairwise alignments contained in the library. We named this objective function COFFEE (Consistency based Objective Function For alignmEnt Evaluation).

Creation of the library

A library is specific for a given set of sequences and is made of pairwise alignments. Taken together, these alignments should contain at least enough information to define a multiple alignment of the sequences in the set. In practice, given a set of N sequences, we included in the library a pairwise alignment for each of the $(N^2 - N)/2$ possible pairs of sequences. This choice is arbitrary since in theory there is no limit regarding the amount of redundancy one can incorporate into a library. For instance, instead of each pair of sequences being represented by a single pairwise alignment, one could use several alternative alignments of this pair, obtained by various methods. In fact, the library is mostly an interface between any method one can invent for generating pairwise alignments, and the COFFEE function optimized by SAGA. However, the method follows the rule 'garbage in/garbage out' and the overall properties of the COFFEE function will most likely reflect the properties of the method used to build the library.

The amount of time it takes to build the library depends on the alignment method used and increases quadratically with the number of sequences. Inside the evaluation algorithm, the library is stored in a look-up table. If each pair of sequences is represented only once, the amount of memory required for the storage increases quadratically with the number of alignments and linearly with their length.

For the analyses presented here, two types of libraries were built. The first one relies on ClustalW. Given a set of N sequences, each possible pair of sequences was aligned using ClustalW with default parameters. The collection of output files obtained that way constitutes the library (ClustalW library). The motivation for using ClustalW as a pairwise method stems from the fact that Clustal is using local gap penalties, even for two sequences.

In order to show that COFFEE is not dependent on the method used to construct the library, a second category of library was created using the FSSP database (Holm and Sander, 1996b). FSSP is a database containing all the PDB structures aligned with one another in a pairwise manner. For each test case, a set of sequences was chosen and the $(N^2 - N)/2$ pairwise structure alignments involving these sequences were extracted from the FSSP database to construct an FSSP library. We also used as references the multiple alignments contained in FSSP. An FSSP entry is always based around a guide structure to which all the other structures are aligned in a pairwise manner. This collection of pairwise alignments can be regarded as a pairwise-based multiple alignment. This means that if one is interested in a set of N protein structures, FSSP contains the N corresponding pairwise-based multiple alignments, each using one structure of the set as a guide. Generally speaking, these N multiple alignments do not have to be consistent with one another, but only consistent with the subset of the pairwise alignments that was used to produce them.

Evaluation procedure: the COFFEE function

Let us assume an alignment of N sequences and an appropriate library built for this set. Evaluation is made by comparing each pair of aligned residues (i.e. two residues aligned with each other or a residue aligned with a gap) observed in the multiple alignment to those present in the library (Figure 1). In such a comparison, residues are identified by their position in the sequence (gaps are not distinguished from one another). In the simplest scheme, the overall consistency score is equal to the number of pairs of residues present in the

multiple alignments that are also found in the library, divided by the total number of pairs observed in the multiple sequence alignment. This measure gives an overall score between 0 and 1. The maximum value a multiple alignment can have depends on the library. For the optimal score to be 1, all the alignments in the library need to be compatible with one another (e.g. when all the pairwise alignments have been extracted from the same multiple sequence alignment or when the sequences are almost identical).

In practice, this scheme needs extra readjustments to incorporate some important properties of the sequence set. For instance, the significance of the information content of each pairwise alignment is not identical. Several schemes have been described in the literature for weighting sequences according to the amount of information they bring to a multiple alignment (Altschul *et al.*, 1989; Sibbald and Argos, 1990; Vingron and Sibbald, 1993; Thompson *et al.*, 1994a). In COFFEE, our main concern was to decrease the amount of noise produced by inaccurate pairwise alignments in the library. To do so, each pairwise alignment in the library is given a weight that is a function of its quality. For this purpose, we used a very simple criterion: the weight of a pairwise alignment is equal to the per cent identity between the two aligned sequences in the library. This may seem counterintuitive since weighting schemes are normally used in order to decrease the amount of redundancy in a set of sequences (i.e. down-weighting sequences that have a lots of close relatives). Doing so makes sense in the context of profile searches (Gribskov *et al.*, 1987; Thompson *et al.*, 1994b) where it is important to prevent domination of the profile by a given subfamily. However, in the case of multiple sequence alignments made by global optimization, it is more important to make sure that closely related pairs of sequences are correctly aligned, regardless of the background noise introduced by other less related sequences. In such a context, a weight can be regarded as a constraint. The consequence is that the alignment of a given sequence will mostly be influenced by its closest relatives. On the other hand, if a sequence lacks any really close relative, its alignment will mostly be influenced by the consistency of its pairwise alignments with the rest of the library.

The COFFEE function can be formalized as follow. Given N aligned sequences $S_1 \ldots S_N$ in a multiple alignment, $A_{i,j}$ is the pairwise projection (obtained from the multiple alignment) of the sequences S_i and S_j, LEN($A_{i,j}$) is the length of this alignment, SCORE($A_{i,j}$) is the overall consistency (level of identity) between $A_{i,j}$ and the corresponding pairwise alignment in the library and $W_{i,j}$ is the weight associated with this pairwise alignment. Given these definitions, the COFFEE score is defined as follows:

$$\text{COFFEE score} = \left[\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} W_{i,j} \times \text{SCORE}(A_{i,j})\right] / \left[\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} W_{i,j} \times \text{LEN}(A_{i,j})\right] \quad (1)$$

with:

$$\text{SCORE}(A_{i,j}) = \text{number of aligned pairs of residues that are shared between } A_{i,j} \text{ and the library} \quad (2)$$

The COFFEE function presents some similarities with the 'weighted sums of pairs' (Altschul and Erickson, 1986). Here as well, we consider all the pairwise substitutions in the multiple alignment, and weight these in a way that reflects the relationships between the sequences. The library plays the role of the substitution matrix. The main differences between the COFFEE function and the 'weighted sums of pairs' are that (i) no extra gap penalties are applied in our scheme, since this information is already contained in the library, (ii) the COFFEE score is normalized by the value of the maximum score (i.e. its value is between 0 and 1) and (iii) the cost of the substitutions is made position dependent, thanks to the library (i.e. two similar pairs of residues will have potentially different scores if the indices of the residues are different). Under this formulation, an alignment having an optimal COFFEE score will be equivalent to an MWT alignment using a 'pairwise alignment graph' (Kececioglu, 1993).

The score defined above is a global measure for an entire alignment. It can also be adapted for local evaluation. We have defined two types of local scores: the residue score and the sequence score. The residue score is given below. S_i^x is the residue x in sequence S_i and $A_{i,j}^{x,y}$ is the pair of aligned residues S_i^x and S_j^y in the pairwise alignment $A_{i,j}$.

$$\text{residue score}(S_i^x) = \sum_{j=1,j!=1}^{N} W_{i,j} \times \text{OCCURRENCE}(A_{i,j}^{x,y}) / \sum_{j=1,j!=1}^{N} W_{i,j} \quad (3)$$

OCCURRENCE($A_{i,j}^{x,y}$) is equal to the number of occurrences of the pair $A_{i,j}^{x,y}$ in the reference library (0 or 1 when using the libraries described here). The sequence score is the natural extension of the residue score. It is defined as the sum of the score of each residue in a sequence divided by the number of residues in that sequence.

Optimizing an alignment for its COFFEE score: SAGA-COFFEE

The aim is to create an alignment having the best possible COFFEE score (optimal alignment). Doing so is a difficult

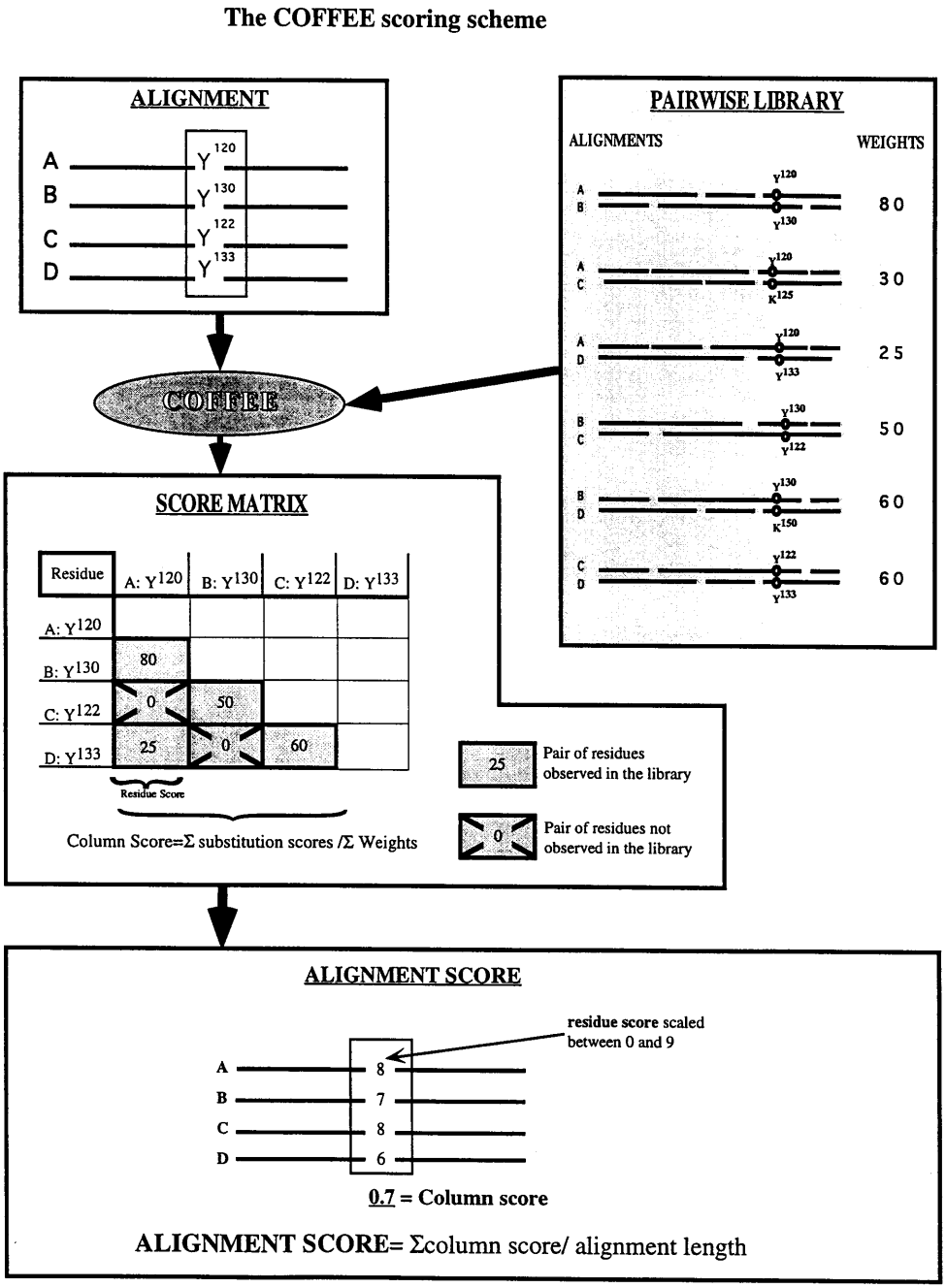

Fig. 1. COFFEE scoring scheme. This figure indicates how a column of an ALIGNMENT is evaluated by the COFFEE function using a REFERENCE LIBRARY. Each pair in the alignment is evaluated (SCORE MATRIX). In the score matrix, a pair receives a score of 0 if it does not appear in the library or a score equal to the WEIGHT of the pair of sequences in which it occurs in the PAIRWISE LIBRARY. Since the matrix is symmetrical, the column score is equal to the sum of half of the matrix entries, excluding the main diagonal. This value is divided by the maximum score of each entry (i.e. the sum of the weights contained in the library). The residue score is equal to the sum of the entries contained by one line of the matrix, divided by the sum of the maximum score of these entries.

task. The computational complexity of a dynamic programming solution is known to be NP-complete (Kececioglu, 1993). For reasons discussed in the Introduction, we used SAGA V0.93 (Notredame and Higgins, 1996).

Table 1. Accuracy of the prediction made on the category 5 of substitution

Test case	Length	Nseq.	Proportion (H+E) (%)	Avg Id. (%)	COFFEE score		Accuracy (H+E) %		Accuracy (ov.) %		CPU time (s)	N.G.
					Clustal	SAGA	Clustal	SAGA	Clustal	SAGA		
ac prot	248	14	57	21	0.48	0.56	39.2	50.2	35.2	45.9	21 009	535
binding	500	7	68	31	0.72	0.84	50.0	64.5	50.0	61.7	1003	166
cytc	146	6	43	42	0.84	0.87	89.1	90.7	88.3	86.1	699	259
fniii	136	9	48	17	0.49	0.62	42.0	47.0	35.7	43.6	936	480
gcr	52	8	57	36	0.86	0.89	80.8	83.1	76.7	80.2	91	55
globin	183	17	74	24	0.78	0.80	86.4	85.2	82.1	81.7	28 477	222
igb	194	37	53	24	0.63	0.67	74.8	78.1	65.6	69.4	110 453	132
lzm	213	6	53	39	0.87	0.87	72.2	72.3	72.3	72.4	256	105
phenyldiox	90	8	67	22	0.59	0.65	58.5	64.7	60.4	61.4	388	110
sbt	331	7	57	61	0.96	0.97	96.9	96.9	93.6	93.6	644	127
s prot	229	15	51	27	0.69	0.74	62.5	66.6	57.7	61.2	44 978	744
ceo	882	7	/	14	/	/	/	/	/	/	13 756	882
vjs	1207	8	/	12	/	/	/	/	/	/	43 568	1400

Test case: generic name of the test case, taken from 3D_ali for the first 11 (ac prot: acid proteases, binding: sugar/amino acid binding proteins, cytc: cytochrome c' ss, fniii: fibronectin type III, gcr: crystallins, globin: globins/phycocyanins/collicins, igb: immunoglobulin fold, lzm: lyzozymes/lactalbumin, phenyldiox: dihydroxybiphenyl dioxygenase, sbt: subtilisin, s_prot: serine protease fold) and from the DALI server for the last two. ceo includes: 1cbg, 1ceo, 1edg, 1byb, 1ghr, 1×yzA and vjs includes: 1cnv, 1vjs, 1smd, 2aaa, 1pamA, 2amg, 1ctn, 2ebn. Length: length of the reference alignment. Nseq: number of sequences in the alignment. Proportion (E+H): percentage of the substitutions involving E→E or H→H. Avg. Id.: average level of identity between the sequences. OF score: score measured with the COFFEE function using a ClustalW library on the ClustalW or the SAGA alignments. Accuracy (E+H): percentage of the (E+H) substitutions found identical between the SAGA (or ClustalW alignment) and the reference. Accuracy (ov.): percentage of substitutions similar in the SAGA (or ClustalW) alignment and in the reference. CPU time: cpu time in seconds using an alpha 8300 machine N.G.: number of generations needed by SAGA to find the solution. The results for the two last test cases analysis are presented in Table 6.

SAGA follows the general principles of genetic algorithms as described by Goldberg (1989) and Davis (1991). In SAGA, a population of alignments evolves through recombination, mutation and selection. This process goes on through series of cycles known as generations. Every generation, the alignments are evaluated for their score (COFFEE). This score is turned into some fitness measure. In an analogy with natural selection, the fitter an alignment, the more likely it is to survive and produce an offspring. From one generation to the next, some alignments will be kept unchanged (statistically the fittest), others will be randomly modified (mutations), combined with another alignment (cross-over) or will simply die (statistically the less fit). The new population generated that way will once again undergo the same chain of events, until the best scoring alignment cannot be improved for a specific number of generations (typically 100).

Operators play a central role in the GA strategy. They can be subdivided between two categories. First the cross-overs, which combine the content of two multiple alignments. Thanks to them, and to the pressure of selection, good blocks tend to be merged into the same alignments. On their own, cross-overs cannot create new blocks, this needs to be done by the second category of operators: the mutations. These are specific programs that input an alignment and modify it by inserting or moving patterns of gaps. Mutations can be slightly greedy (attempt to make some local optimization) or completely random.

A key concept in the genetic algorithm strategy is that the fitness-based selection is not absolute but statistical. To select an individual, a virtual wheel is created. On this wheel, each individual is given a number of slots proportional to its fitness. To make a selection, the wheel is spun. Therefore, the best individuals are simply more likely to survive, or to be selected for a cross-over or a mutation. This form of selection protects the GA search from excessive greediness, hence preventing it from converging onto the first local minimum encountered during the search.

SAGA V0.93 is mostly similar to the Version 0.91 described in Notredame and Higgins (1996). Most of the changes between Version 0.93 and 0.91 have to do with some improvement in the implementation and the user interface, but do not affect the algorithm itself. To optimize the COFFEE scores, SAGA was run using the default parameters described for SAGA 0.91 in Notredame and Higgins (1996). SAGA was also modified so that it could evaluate any alignment (including a ClustalW alignment) using the COFFEE function.

Test cases

To assess the biological accuracy of the COFFEE function and the efficiency of its optimization by SAGA, 13 test cases were designed. They are based on sequences with known three-dimensional structures, for which a structural alignment is available. This choice was guided by the fact that structure-based alignments are usually biologically more correct than any other alternative, especially when they involve proteins with low sequence similarity. For this reason, we used these structure-based alignments as a standard of truth in our analyses. Eleven test cases were extracted from the 3D_ali Release 2.0 (Pascarella and Argos, 1992). Alignments were selected according to the following criteria: alignments with more than five structures and with a consensus length larger than 50. In 3D_ali, 18 alignments meet this requirement. Among these, we removed those for which ClustalW produces a multiple alignment >95% identical to its structural counterpart (four alignments). We also removed three alignments which were impossible to align accurately using ClustalW or SAGA/COFFEE. These consist of sets of very distantly related sequences with unusually long insertions/deletions (barrel, nbd and virus in 3D_ali). These alignments are beyond the scope of conventional global sequence alignment algorithms. This leaves a total of 11 alignments used in our analyses. Their characteristics are shown in Table 1.

The two last test cases were extracted from the FSSP database. As opposed to the 11 other test cases, they have been specifically designed for making a multiple sequence alignment using a structure-based reference library. This explains their low level of average sequence identity, as can be seen in Table 1 (the two last entries, vjs and ceo).

Evaluation of the COFFEE function accuracy

When evaluating a new OF with SAGA, two main issues are involved: the quality of the optimization and the biological relevance of the optimal alignment. Another aspect involves the comparison of the new OF with already existing methods.

The evaluation of the biological relevance of the COFFEE function required the use of some references. The structural alignments described above were used for this purpose. Comparison between a sample alignment and its reference was made following the protocol described in Notredame and Higgins (1996), inspired by the method used by Vogt *et al.* (1995) for substitution matrix comparison and Gotoh (1996). All the pairs (excluding gaps) observed in the sample alignment were compared to those present in the reference. The level of similarity is defined as the ratio between the number of identical pairs in the two multiple alignments divided by the total number of pairs in the reference.

This procedure gives access to an overall comparison. It does not reflect the fact that in a global structural alignment, some positions are not correctly aligned because they cannot be aligned (this is true of any position where the two structures cannot be superimposed). In practice, structural alignment procedures may deal with these situations in different ways, producing sequence alignments that are sometimes locally arbitrary (especially in the loops). While in DALI these regions are explicitly excluded from the alignment, it is not so obvious to identify them in the multiple sequence alignments contained in 3D_ali. To overcome this type of noise, a procedure was designed that should be less affected by misalignments. For this alternative measure of biological relevance, we only take into account substitutions that involve a conservation of secondary structural state in the reference alignment (helix to helix and beta strand to beta strand). In the text and the tables, this category of substitution is referred to as (E+H). In most of the test cases, the (E+H) category makes up the majority of the observed pairs, as can be seen in Table 1.

For each of the first 11 test cases (3D_ali), the evaluation procedure involved making multiple alignments with five different methods (cf. the next section) and a ClustalW library (default parameters). The ClustalW library was used with SAGA for producing a multiple alignment having an optimal COFFEE score. The biological relevance of this alignment was then assessed by comparison with the structural reference, and compared to the accuracy obtained with the other methods on the same sequences. On the two last test cases (vjs and ceo), alignments were made using FSSP libraries. Alignment accuracy was assessed using the DALI scoring measure. Given a pairwise alignment, this is a measure of the quality of the structure superimposition implied by the alignment. The program used for this purpose (trimdali) returns the DALI score (Holm and Sander, 1993) and two other values: the length of the consensus (number of residues that could be superimposed) and the average RMS (the average deviation between equivalent Ca atoms). These values were computed for each possible pairwise projection of the multiple alignments and averaged. The scores obtained that way for the SAGA alignments were compared to similar scores measured on the FSSP pairwise-based multiple alignments.

Comparison of COFFEE with other methods

In total, six alignment methods where used to align the 3D ali test cases: ClustalW v1.6 (Thompson *et al.*, 1994a), SAGA with the MSA objective function (SAGA-MSA) (Notredame and Higgins, 1996), PRRP (V 2.5.1), the iterative alignment method recently described by Gotoh (1996), PILEUP (Higgins and Sharp, 1988) in GCG v9.1 and SAM (v2.0), a HMM package (Hughey and Krogh, 1996) and SAGA-COFFEE.

Apart from SAM, all these methods were used with the default parameters that came along with the package. In the case of SAM, since it is known that HMMs usually require large sets of sequences in order to evaluate a model, we used the Dirichlet mixture regularizer provided in the package, which is supposed to compensate for this type of problems. SAGA-MSA is the package previously described (Notredame and Higgins, 1996), Rationale 2 weights (Altschul *et al.*, 1989) were computed using the MSA package. (Lipman *et al.*, 1989). When possible, MSA was used on the same sequences as SAGA-MSA in order to control the quality of the optimization. Results were consistent with those previously reported.

Implementation

The COFFEE function and the procedure for building ClustalW pairwise libraries have been implemented in ANSI C. These programs have been integrated in Version 0.93 of the SAGA package also written in ANSI C. These are available upon request from the authors (http://www.ebi.ac.uk/~cedric).

Results

Accuracy and complexity of the optimization

Since our approach relies on the ability of SAGA to optimize the COFFEE function, we checked that this optimization was performed correctly. For each test case, a dummy library was created, containing sets of pairwise alignments identical to those observed in the reference multiple structure alignment. In such a case, the structural alignment has a score of 1 since it agrees completely with the library. Therefore, the maximum score that can be reached by SAGA also becomes 1. Since, under these artificial conditions, the score of the optimum is known, we could test the accuracy of SAGA's optimization. Several runs made on the same set reached the optimum value in an average of 5.4 runs out of 10. The lowest reproducibility was found with the largest test cases of Table 1 (igb or s prot with a score of 1 being reached, respectively, one and two times out of 10 runs). However, even if the optimal score is not reached, we found that it is always possible to produce an alignment with a score better than 0.94. Although they do not constitute a full proof, these results support the assumption that SAGA is a good choice for optimizing the COFFEE function.

An important aspect is the complexity of the program and the factors that influence it. As we previously reported when optimizing the sums of pairs with SAGA (Notredame and Higgins, 1996), establishing the complexity is not straightforward. The evaluation of a COFFEE score is quadratic with the number of sequences and linear with the consensus length. Using a given population size, the time required for one generation will vary accordingly. For instance, on a fast workstation, it takes ~4 s/generation for the gcr test case and ~7 min/generation for the igb test case. Unfortunately, establishing the complexity in terms of the number of generations needed to reach a global optimum is much harder. This depends on several factors: number of sequences, length of the consensus, relative similarity of the sequences, complexity of the pattern of gaps needed for optimality, operators used for mutations and cross-overs. Since the pattern of gaps is an unknown factor, it is impossible really to predict how many generations will be required for one specific test case. On the other hand, judging from the data in Table 1 (N generations column), it seems that the length of the alignment has a stronger effect than the number of sequences.

Comparison of the COFFEE function with other methods

Multiple alignments were produced with SAGA-COFFEE using ClustalW libraries (best scoring alignment out of 10 runs). These alignments were compared to the structural references. Multiple alignments of the same sets, generated with five other methods, were also compared to the references in order to evaluate relative performances. Since in the way it is used here, the COFFEE function depends heavily on ClustalW, special emphasis was given to the comparison of these two methods (Table 1).

The results are unambiguous. When considering the overall comparison, nine test cases showed that SAGA makes an improvement over ClustalW (in two of these, the improvement is >10%). The trend is similar when looking only at (E+H) substitutions, where 10 test cases out of 11 present an improvement. In the few cases where it occurs, the degradation made by SAGA is always <2%. The extent of the observed improvements usually correlates well with the differences in the scores measured with the COFFEE function. Degradation is only observed in the cases where the ClustalW alignment already has a high level of consistency with the reference library (>75%), as can be seen with the globin (COFFEE score of the ClustalW alignment = 0.78) and the cytochrome C (COFFEE score of the ClustalW alignment = 0.84).

In order to put SAGA-COFFEE in a wider context, comparisons were made using five other different methods (Table 2). These results show that in most of the cases SAGA-COFFEE does reasonably well. When its alignment is not the best, it is usually within 3% of the best (except for the binding and gcr tests, for which the difference is greater). Apart from the HMM method (SAM) that has low performances, it is relatively hard to rank existing methods. PRRP is one of the newest methods available. It has been described as being one of the most accurate (Gotoh, 1996) and happens to be the only one that significantly outperforms SAGA-COFFEE on some test cases. It is also interesting to notice that SAGA-COFFEE is always among the best for test cases

having a low level of identity. This trend is confirmed by the results shown in Table 3, where the sequences are grouped according to their average similarity with the rest of their family (as measured on the reference structural alignment). In this table, we analysed the overall performance of each method and compared it with SAGA-COFFEE by counting (i) the overall per cent of (E+H) residues correctly aligned and (ii) the number of sequences for which SAGA-COFFEE makes a better (b)/worse (w) alignment than a given method. Overall, the results confirm that SAGA-COFFEE seems to do better than the other methods when the sequences have a low level of identity with the rest of their set. The poor performances of SAM can probably be explained by two reasons: the small number of sequences in each test case and perhaps some inadequate default settings in the program (in practice, SAM is often used as an alignment improver rather than on its own).

Sequence identity: minimum and maximum average identity of the sequences of each category with the rest of their alignment as measured on the reference structural alignment. Nseq.: number of sequences in a category. Nres.: number of residues. SAGA-COFFEE percentage of the (E+H) substitutions present in the reference structural alignment observed in the SAGA-COFFEE alignment. ClustalW: (%), similar but using ClustalW alignment; (b), number of sequences for which SAGA-COFFEE produces a better alignment than ClustalW; (w), number of sequences for which SAGA-COFFEE produces a worse alignment than ClustalW. PRRP: similar but with alignments produced with the Gotoh PRRP algorithm (see the text). PILEUP: pileup method from the GCG package. SAGA-MSA: SAGA using the MSA objective function. SAM: sequence alignment modelling by Hidden Markov Model. [Note that the (b) and (w) categories do not necessarily add up to the overall number of sequences because they do not include sequences having the same score with the two method compared.]

Test case: generic name of the test case, taken from 3D_ali (see 3D_ali for PDB identifiers), see Table 1 for more details. Nseq: number of sequences in the alignment. Avg. Id.: average level of identity between the sequences. SAGA-COFFEE accuracy of the alignments obtained with SAGA-COFFEE as judged by comparison with the structural alignment, only considering the (E+H) substitutions. ClustalW: similar but with ClustalW alignments. PRRP: similar but with alignments produced with the Gotoh PRRP algorithm (see the text). PILEUP: pileup method from the GCG package. SAGA-MSA: SAGA using the MSA objective function. SAM: sequence alignment modelling by Hidden Markov Model.

Table 2. Method comparison on the 3D_ali test cases

Test case	Avg. id. (%)	Nseq.	SAGA COFFEE (%)	PRRP (%)	ClustalW (%)	PILEUP (%)	SAGA MSA (%)	SAM (%)
ac prot	21	14	50.2	48.8	39.2	40.9	*51.2	27.9
binding	31	7	64.5	*76.2	50.0	66.6	64.2	36.9
cytc	42	6	90.7	89.4	89.1	*94.6	67.3	67.3
fniii	17	9	*47.0	36.3	42.0	37.8	45.2	16.2
gcr	36	8	83.1	*92.8	80.8	80.8	80.8	85.7
globin	24	17	85.2	*87.0	86.4	72.6	78.0	67.8
igb	24	37	*78.1	74.9	74.8	52.4	70.1	67.2
lzm	39	6	*72.3	71.1	72.2	*72.3	*72.3	55.3
phenyldiox	22	8	*64.7	49.9	58.5	37.4	55.6	45.7
sbt	61	7	96.9	96.7	96.9	*97.4	96.0	90.6
s prot	27	15	66.6	64.3	62.5	57.9	*68.5	61.7

*Indicates the method performing best on a test case.

Table 3. Method comparison on the 3D_ali test cases: global results

Sequence identity	Nseq.	Nres.	SAGA COFFEE (%)	PRRP (%)	b	w	ClustalW (%)	b	w	PILEUP (%)	b	w	SAGA MSA (%)	b	w	SAM (%)	b	w
[00.0–20.0]	28	3424	*63.3	62.2	18	8	49.7	11	6	42.4	23	3	56.9	20	7	36.4	18	3
[20.0–40.0]	88	12 010	*76.2	74.6	57	31	66.1	68	20	60.2	80	8	69.7	63	24	59.1	84	2
[40.0–100.0]	18	3808	89.3	*90.9	14	4	84.6	20	3	89.8	3	15	87.8	16	2	64.3	25	0

*Indicates the method performing best on a given range of identity.

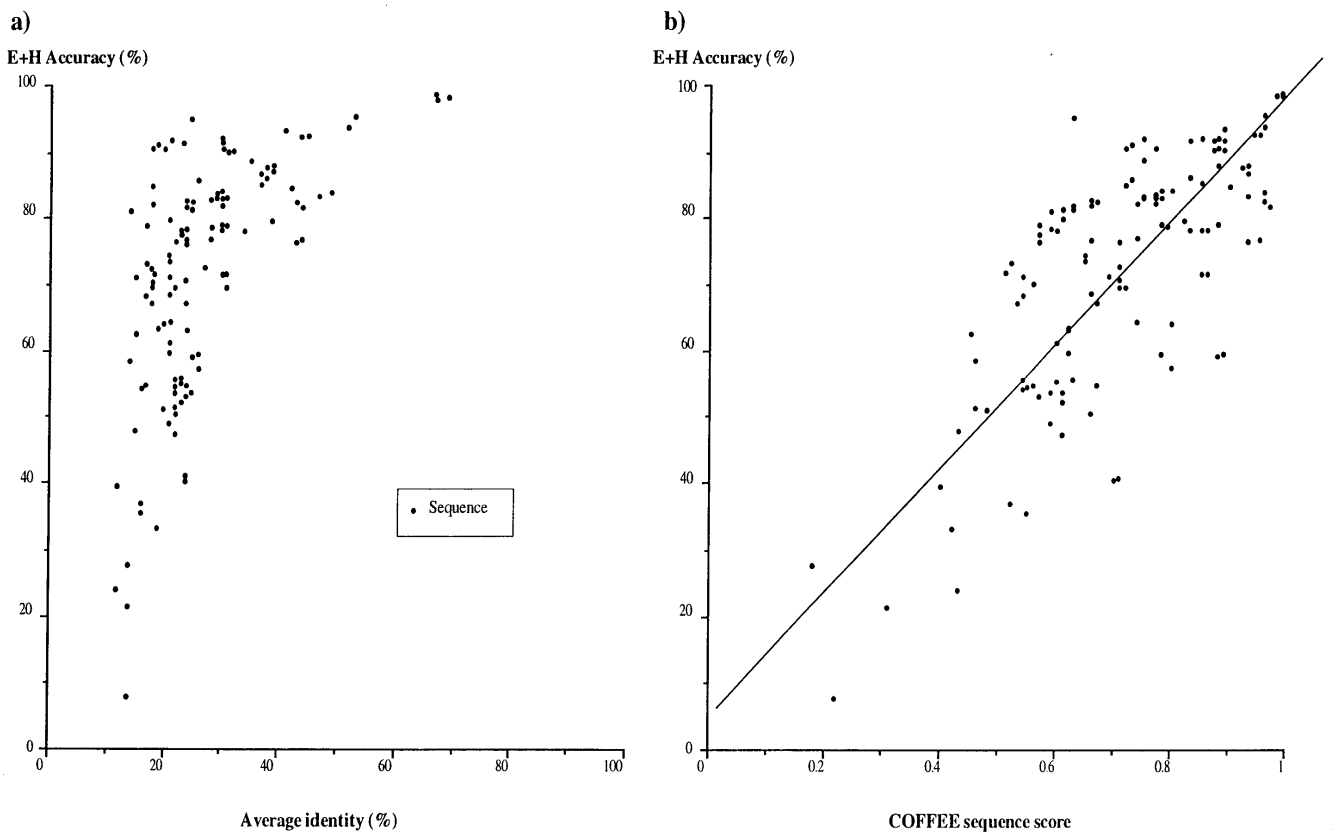

Fig. 2. Correlation between sequence score and alignment accuracy. (**a**) The average level of identity of each sequence with the rest of its alignment was measured on the reference structural alignment. The average level of accuracy of the SAGA-COFFEE alignment of each of these sequences was also estimated on the (E+H) category. The two values are plotted against one another. (**b**) For each sequence, the sequence score was measured on the SAGA-COFFEE alignment, this value was plotted against the accuracy of the sequence alignment. The coefficient of linear correlation was estimated on these points ($r = 0.65$).

These results also indicate that there is no such thing as an ideal method. Even if COFFEE seems to do better on average, one can see in Table 2 and III that the alignments it produces are not always the best. In fact, it seems that depending on the test case any method can do better than the others. Unfortunately, as discussed by Gotoh (1996), it is hard to discriminate the factors that should guide the choice of a method. For this reason, being able to identify the correct portions in a multiple sequence alignment may be even more important than being able to do a very accurate alignment.

Correlation between COFFEE score and alignment accuracy

As mentioned in Methods, the score can be assessed at a local level (sequence score or residue score). One of the benefits of such evaluation is that local score and accuracy can be correlated, thus allowing the identification of potentially correct portions of an alignment with a known risk of error. The 3D_ali structure-based alignments were used once more to validate this approach. Generally speaking, a high residue score will indicate that the pairs in which a given residue is involved are also found in the pairwise library. On the other hand, if none of the pairings in which a given residue is involved are found in the library, this residue will be considered unaligned.

We evaluated the COFFEE score of each sequence in each alignment. In each of these sequences the (E+H) average accuracy was also measured. The graph in Figure 2b shows the relationship between sequence score and (E+H) average accuracy. The correlation between these two quantities is reasonable ($r = 0.65$). When considering the values used for this graph, we found that for >85% of the sequences it is possible to predict the actual accuracy of the alignment with a ±10% error rate. In terms of prediction, this is a substantial improvement over what can be obtained when measuring the average level of identity between one sequence and its multiple alignment, as shown in Figure 2a.

Table 4. Average accuracy of the alignment of each sequence as a function of its sequence score (3D_ali test cases)

Sequence score	N. residues (%)		N. sequences (%)		Accuracy (E+H) (%)	
	ClustalW	SAGA	ClustalW	SAGA	ClustalW	SAGA
[0.00–0.33]	5.8	2.6	6.7	3.0	14.3	9.9
[0.33–0.66]	36.8	33.7	40.3	38.1	63.2	67.2
[0.66–1.00]	57.4	63.7	53.0	59.0	82.0	82.5
TOTAL	19 242 residues		134 sequences			

Sequence score: minimum and maximum score of the sequences in each category. N. residues: percentage of residues belonging to each category estimated on SAGA or ClustalW alignments. N. sequences: percentage of the total sequences belonging to each category of score as measured on the SAGA and the ClustalW alignments. Accuracy (E+H): accuracy associated with each category of score in the SAGA and ClustalW alignments. TOTAL: total number of residues and sequences in the comparison.

Table 5. Accuracy of the prediction made on the category 5 of substitution

Test case	Accuracy (%)		Correct substitution (%)	
	ClustalW	SAGA	ClustalW	SAGA
ac prot	56.8	68.2	9.6	15.7
binding	64.3	61.4	31.5	40.0
cytc	96.2	93.9	72.1	73.5
fniii	81.5	77.7	13.8	15.6
gcr	75.5	77.4	63.4	74.5
globin	97.2	95.0	63.1	66.5
igb	88.8	85.5	39.0	42.3
lzm	91.5	91.8	61.3	61.5
phenyl	78.0	72.5	34.3	40.0
s prot	82.2	82.4	45.2	50.1
sbt	89.7	89.7	85.2	87.0

Test case: generic name of the test case taken from 3D_ali. Accuracy: ratio between the total number of substitutions in category 5 (in SAGA and ClustalW alignments) and the number of these substitutions present in the reference alignment. % Correct substitutions: percentage of the correct substitutions (over the total number, all substitution categories included) identified in the category 5 of substitution.

The correlation between score and accuracy becomes slightly more apparent when looking at the data in a more global way (Table 4). In this case, the sequences have been grouped according to their score, and the accuracy of their alignment was measured. One can see that the higher the score of a sequence, the higher its average alignment accuracy. We also found that the distribution of the sequences among the three categories was modified when using ClustalW instead of SAGA. SAGA produces more sequences with a high score than does ClustalW. This means that not only are SAGA alignments more accurate than ClustalW, it is also possible to identify them for being so. In practice, the sequence score, as imperfect as it may seem, provides a fast and simple way to identify sequences that do not really belong to a set or that are so remotely related to the rest that their alignment should be considered with care.

The sequence score is a global measure. It does not reflect the local variations that occur at the residue level. To analyse these kinds of data and assess their utility for predicting correct portions of an alignment, the score of each residue in each multiple sequence alignment was evaluated using equation (3). These scores were scaled in a range varying from 0 to 9. A residue has a score of 9 if >90% of the pairs in which it is involved are also present in the reference library, and so forth for 8 ([80–90%]), etc. Once residue scores have been evaluated, substitution classes can be defined. For instance, the class 5 of substitutions includes all the residues of a multiple alignment having a residue score superior to or equal to 5 (Figure 3a), the class 0 of substitution includes all the residues in the alignment.

Figure 3a gives an example of such an evaluation. In this alignment, each residue is replaced by its score, and the residues that belong to the category 5 of substitution are boxed. Figure 3b shows the correctly aligned residues in this category. It is possible to see that using our measure, one can identify some of the correct portions in the SAGA fniii alignment. As can be gathered from Table 1, fniii is a very demanding test case. Except for the two first sequences, which are almost identical, all the other members of this set have a very low level of identity with one another. This is especially true for the sequence 2hft_1 which illustrates well the advantages and limits of our method. This sequence is not the most remotely related to the set. It has an average identity of 14%, whereas two other members (3hhr_c and 2hft) are more distantly related with 12% average identity. Despite this fact, 2hft_1 gets the lowest sequence score in the multiple alignment (0.29). This correlates well with the fact that it also has the lowest alignment accuracy of the set [18% overall, 20% for the (E+H) category]. Similarly, the only non-terminal stretch of this sequence that belongs to the category 5 is one of the only portions to be correctly aligned (Figure 3a and b).

The same type of analyses were carried out on the 10 other test cases (Table 5). Our measures indicate that using the category 5 of substitution, a substantial portion of correctly aligned residues can be identified. When comparing Clus-

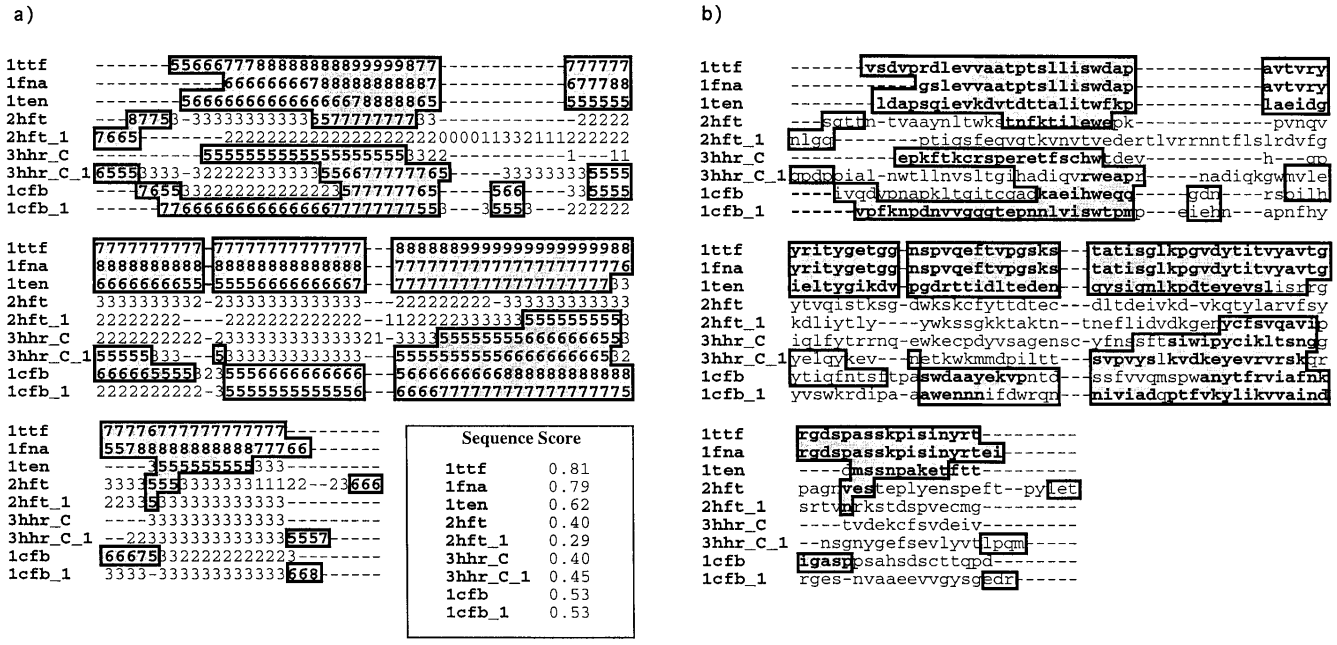

Fig. 3. Evaluation of the accuracy of the fniii test case (fibronectin type III family). (**a**) Sequences in the fniii test case were aligned by SAGA-COFFEE using a ClustalW library. The alignment obtained that way was evaluated locally. The sequences names are the PDB identifiers. The suffix _1, _2.. indicates that several portions of the same sequence have been used (cf. 3D_ali for further details). In this alignment, each residue has been replaced by its score. The gray boxes indicate all the residues that belong to category 5 of substitution (i.e. having a score ≥5). The sequence score box lists the values measured on each sequence. (**b**) The accuracy in the category 5 of substitution (boxes) was evaluated by comparison with the reference alignment. Residues shadowed in gray in the boxes are correctly aligned to one another. Boxed residues not shadowed are not correctly aligned with each other or with the rest of the category 5 residues. Residues not contained in the boxes are not taken into account for this evaluation.

talW and SAGA, we found that more correct residues can be identified with SAGA. This improvement is sometimes achieved at the cost of a slightly lower accuracy (more false positives) in the SAGA alignments.

A global estimation was made to evaluate the accuracy that can be expected when using any of the 10 substitution categories on a SAGA alignment. The proportion of correct substitutions predicted that way was also measured. These results are presented in Figure 4a and b, respectively. Residues are grouped in three classes, depending on the score of the sequences in which they occur. Figure 4a confirms that high-scoring residues are usually correctly aligned (high accuracy). However, the higher the substitution category, the smaller the number of residues on which a prediction can be made, as shown in Figure 4b. These graphs confirm that the residue score can be used as a measure for predicting accuracy; they also indicate that the sequence score is informative when making a prediction on a residue.

Making a multiple structural alignment

The analysis carried out with the ClustalW libraries represents only one possible application for the COFFEE function. Generally speaking, the COFFEE scheme allows the combination of the information contained by any reference library, regardless of the method used for its construction. To illustrate this fact, we show that it is possible to build a structure-based multiple sequence alignment when a library of high-quality pairwise structural alignments is available. We used COFFEE on two sets of proteins (vjs and ceo) using appropriate FSSP libraries.

It was impossible to improve significantly over FSSP for the ceo test case, made of endoglucanases and other related carbohydrate degradation enzymes. This can be explained by the fact that the FSSP alignment with the best DALI score (the one using 1ceo as a guide) already has a high level of consistency with the library (COFFEE score = 0.82). This shows quite clearly in the fact that this alignment is 88% similar to the SAGA-COFFEE one.

The second set is made of amylases and other carbohydrate degradation enzymes. Table 6 is used to compare the SAGA-COFFEE alignment of these sequences with the corresponding FSSP pairwise-based multiple alignments. These results clearly indicate that the alignment produced by SAGA is better than any of the FSSP multiple alignments, regardless of the criterion used to evaluate this improvement (DALI score, consensus length or RMS). This result was to be expected since SAGA makes use of much more information

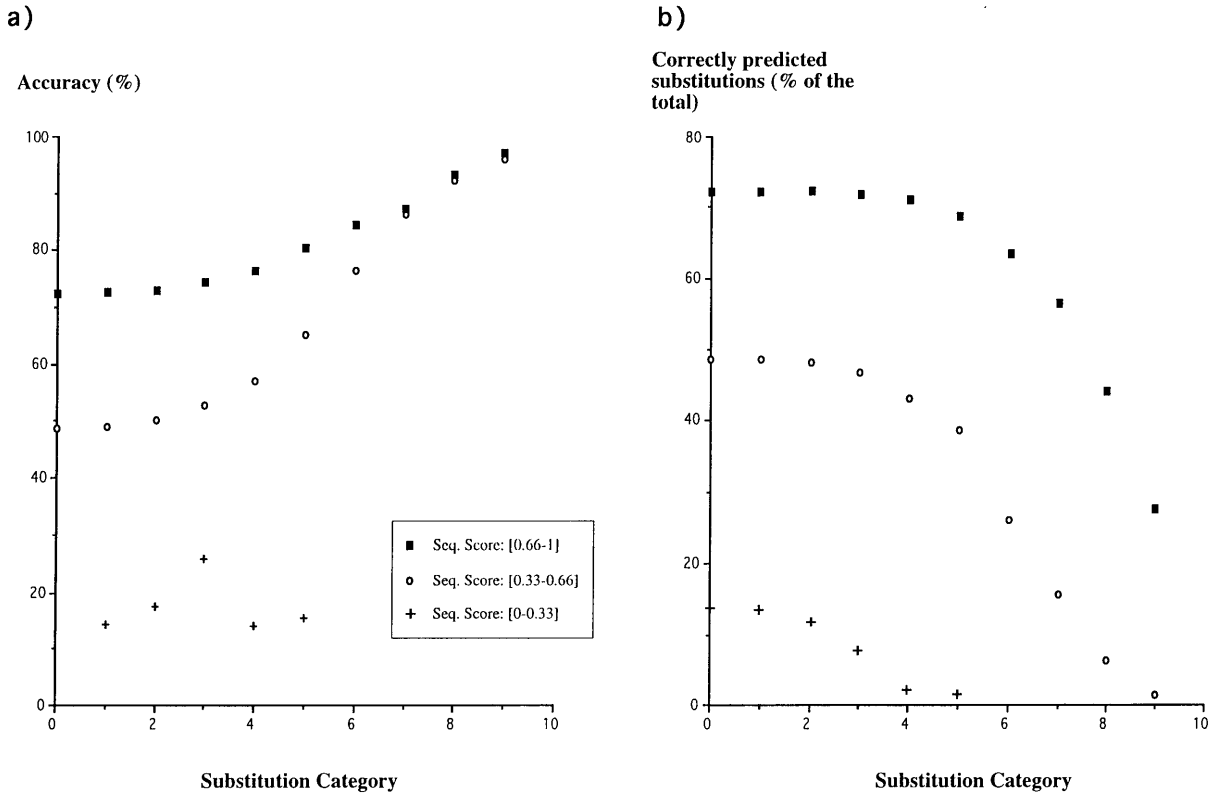

Fig. 4. Prediction of correctly aligned residues using the residue COFFEE score. (**a**) The accuracy of the alignments (number of correct substitutions in one of the categories divided by the total number of substitutions in this category) of each sequence was measured. To do so, sequences were divided into three groups, depending on their sequence score. The graph was made for each of the three groups. (**b**) For each sequence, the number of correct substitutions contained in each category was evaluated and divided by the overall number of substitutions involving that sequence. This value was plotted against the category of substitution.

than any of the FSSP alignments. In Table 6, entries are sorted according to the DALI score. This allows one to see that the DALI and COFFEE scores correlate well for the FSSP alignments, and supports the idea that the COFFEE score is also a good indicator of the alignment quality when the library is based on structural alignments.

Table 6. Comparison of FSSP and SAGA multiple alignments

Guide sequence	Average DALI score	Average consensus length	Average RMS (Å)	COFFEE score
2ebn	1152.6	186.5	3.73	0.53
1cnv	1205.2	196.4	3.63	0.59
1vjs	1258.4	198.8	3.62	0.50
1ctn	1331.2	196.9	3.53	0.60
1smd	1667.1	219.4	3.40	0.65
2amg	1672.9	217.7	3.42	0.67
2aaa	1766.8	224.9	3.45	0.69
1pamA	1786.3	225.8	3.30	0.70
SAGA-COFFEE	1860.0	230.9	3.20	0.79

Guide sequence: sequence used as a guide in the FSSP multiple alignment (SAGA indicates the alignment obtained with SAGA-COFFEE). Average DALI score: average DALI score of each pair of sequences in the alignment. The table is sorted according to the values of these entries. Average consensus length: average of the number of residues superimposable by DALI in each pair of sequence. Average RMS: the average of the RMS values measured by DALI on each pair of the alignment in Ångströms. COFFEE score: score given by SAGA to the multiple alignments using the same library.

In theory, we could have used the DALI score as an objective function, and optimized it with SAGA. In such a context, DALI would be used to evaluate all the pairwise projections in order to give a score similar to the one shown in the 'DALI score' column of Table 6. However, this is not possible in practice because the computation of a DALI score is much more expensive than the evaluation of a COFFEE score. DALI score used on a multiple alignment is quadratic with the number of sequences and quadratic with the length of the alignment. The COFFEE score is also quadratic with the number of sequences, but only linear with the length of the alignment. In consequence, even if we were to assume a global DALI score to be biologically more realistic than the FSSP library-based COFFEE score, COFFEE still appears as a good trade-off between approximating DALI and saving on computational cost.

Discussion

In this work, we show that alignments can be evaluated for their MWT score using the COFFEE function and subsequently optimized with the genetic algorithm package SAGA. Given a heterogeneous, non-consistent collection of pairwise alignments, one can extract the corresponding multiple alignment with COFFEE and SAGA.

We have shown here that the SAGA-COFFEE scheme outperforms most of the commonly used alternative packages when applied to sequences having low levels of identity. The comparison made with other global optimization techniques such as SAGA-MSA and PRRP indicates that the method is not only better because it does a global optimization, but also probably because of the way it uses information, filtering some of the noise through the library of pairwise alignments. The weighting scheme also plays a role in this improvement. It helps turning the relationship between the sequences into some of the constraints that define the optimal alignment. It is because all these constraints (library and weights) are unlikely to be consistent that the genetic algorithm strategy proves to be a very appropriate mean of optimization. There is little doubt that the performances of our method will also depend on the relationship between the sequences. Sets with a lot of intermediate sequences (i.e. a dense phylogenetic tree) are likely to lead to more accurate alignments. However, the fact that COFFEE proves able to deal with sequences having a very low level of identity is quite encouraging regarding the robustness of the method.

One of the main advantages of the COFFEE strategy is the flexibility given to the user for defining the library. Here, by using two completely different pairwise alignment methods, we managed to produce high-quality multiple alignments in both cases. This is interesting, but constitutes only a first step. The structure of the libraries we have been using is very simple. They only rely on an 'all-against-all' comparison using one type of pairwise alignment algorithm per library. In practice, this scheme can easily be extended to much more complex library structures.

It is common sense to have a higher confidence in results that can be reproduced using independent methods. Some prediction methods rely on this type of assumption, such as the block definition strategy described by Henikoff *et al.* (1995).These methods usually limit themselves to identifying highly conserved patterns among a set of solutions. With the COFFEE strategy, we go much further and make it possible to find a consensus solution whatever the number of constraints and whatever their relative compatibility. Of course, it is not enough for a solution to exist, one also needs to know how accurate this solution is. In this work, we have shown that the level of consistency of a solution is a good indicator of such accuracy.

This accuracy prediction constitutes the other main aspect of the COFFEE function. Several methods have been proposed that attempt to predict correct portions of a pairwise alignment given a set of sub-optimal alignments (Gotoh, 1990; Vingron and Argos, 1990; Mevissen and Vingron, 1996). Using these methods, libraries could be designed with large numbers of sub-optimal alignments. Here again, the difference between the COFFEE method and other previously proposed approaches is that not only is it possible to predict the correct portions in an alignment, but it is also possible to optimize a multiple alignment for having as many reliable regions as possible.

SAGA-COFFEE still needs to be improved on several accounts. For instance, further approaches will involve the definition of more complex libraries that will hopefully lead to more meaningful consistency indices. The main source of inspiration when doing so will be the work done on pairwise alignment stability (Mevissen and Vingron, 1996). The other direction we plan to take has to do with the combination of scoring schemes. We have seen here that there is no uniform solution to the multiple sequence alignment problem. For this reason, it would make sense to generate libraries containing alternative alignments made by all the available methods (PRRP, ClustalW, HMM, etc.). COFFEE could then be used to merge this information and hopefully extract the best of each alignment. This will require some improvement of the COFFEE function and its adaptation to highly redundant library. Another crucial aspect will be increasing the efficiency of the algorithm. At present, SAGA-COFFEE is an extremely slow method; however, we hope to improve on this by using a more appropriate type of seeding.

Finally, another important aspect of our approach will involve the refinement of the method used here for building multiple structural alignments. The project will be based on a procedure similar to the one described above: the design of more efficient weights and an attempt to use the alternative

structural alignments that can be produced by the DALI method, using a wider range of DALI test cases.

Acknowledgements

The authors wish to thank Miguel Andrade and Thure Etzold for very useful comments and corrections. They also wish to thank the referees for their useful remarks and interesting suggestions.

References

Altschul,S.F. (1989) Gap costs for multiple sequence alignment. *J. Theor. Biol.*, **138**, 297–309.

Altschul,S.F. and Erickson,B.W. (1986) Optimal sequence alignment using affine gap costs. *Bull. Math. Biol.*, **48**, 603–616.

Altschul,S.F., Carroll,R.J. and Lipman,D.J. (1989) Weights for data related by a tree. *J. Mol. Biol.*, **207**, 647–653.

Bucher,P. and Hofmann,K. (1996) A sequence similarity search algorithm based on a probabilistic interpretation of an alignment scoring system. In *Fourth International Conference on Intelligent Systems for Molecular Biology.* AAAI Press, St Louis, MO.

Carrillo,H. and Lipman,D.J. (1988) The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.*, **48**, 1073–1082.

Davis,L. (1991) *The Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.

Dayhoff,M.O. (1978) *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Washington, DC.

Eddy,S.R. (1995) Multiple alignment using hidden Markov models. In *Third International Conference on Intelligent Systems for Molecular Biology (ISMB)*, Cambridge, England. AAAI Press, Menlo Park, CA.

Feng,D.-F. and Doolittle,R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, **25**, 351–360.

Godzik,A. and Skolnik,J. (1994) Flexible algorithm for direct multiple alignment of protein structures and sequences. *Comput. Applic. Biosci.*, **10**, 587–596.

Goldberg,D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York.

Gotoh,O. (1990) Consistency of optimal sequence alignments. *Bull. Math. Biol.*, **52**, 509–525.

Gotoh,O. (1996) Significant improvement in accuracy of multiple protein sequence alignments by iterative refinements as assessed by reference to structural alignments. *J. Mol. Biol.*, **264**, 823–838.

Gribskov,M., McLachlan,M. and Eisenberg,D. (1987) Profile analysis: Detection of distantly related proteins. *Proc. Natl Acad. Sci. USA*, **84**, 4355–4358.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Henikoff,S., Henikoff,J., Alford,W. and Pietrokovsky,S. (1995) Automated construction and graphical representation of blocks from unaligned sequences. *Gene*, **163**, GC17–26.

Higgins,D.G. and Sharp,P.M. (1988) CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, **73**, 237–244.

Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, **233**, 123–138.

Holm,L. and Sander,C. (1996a) Alignment of three-dimensional protein structures: network server for database searching. *Methods Enzymol.*, **266**, 653–662.

Holm,L. and Sander,C. (1996b) The FSSP database: fold classification based on structure-structure alignment of proteins. *Nucleic Acids Res.*, **24**, 206–210.

Hughey,R. and Krogh,A. (1996) Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Comput. Applic. Biosci.*, **12**, 95–107.

Ishikawa,M., Toya,T., Hoshida,M., Nitta,K., Ogiwara,A. and Kanehisa,M. (1993a) Multiple sequence alignment by parallel simulated annealing. *Comput. Applic. Biosci.*, **9**, 267–273.

Ishikawa,M., Toya,T. and Tokoti,Y. (1993b) Parallel iterative aligner with genetic algorithm. In *Artificial Intelligence and Genome Workshop, 13th International Conference on Artificial Intelligence*, Chambery, France.

Kececioglu,J.D. (1993) The maximum weight trace problem in multiple sequence alignmnet. *Lecture Notes Comput. Sci.*, **684**, 106–119.

Kim,J., Pramanik,S. and Chung,M.J. (1994) Multiple sequence alignment using simulated annealing. *Comput. Applic. Biosci.*, **10**, 419–426.

Krogh,A. and Mitchison,G. (1995) Maximum entropy weighting of aligned sequences of proteins or DNA. In *Third International Conference on Intelligent Systems for Molecular Biology (ISMB)*, Cambridge, England. AAAI Press, Menlo Park, CA.

Lipman,D.J., Altschul,S.F. and Kececioglu,J.D. (1989) A tool for multiple sequence alignment. *Proc. Natl Acad. Sci. USA*, **86**, 4412–4415.

Mevissen,H.T. and Vingron,M. (1996) Quantifying the local reliability of a sequence alignment. *Protein Eng.*, **9**, 127–132.

Morgenstern,B., Dress,A. and Wener,T. (1996) Multiple DNA and protein sequence based on segment-to-segment comparison. *Proc. Natl Acad. Sci. USA*, **93**, 12098–12103.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Notredame,C. and Higgins,D.G. (1996) SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res.*, **24**, 1515–1524.

Notredame,C., O'Brien,E.A. and Higgins,D.G. (1997) RAGA: RNA sequence alignment by genetic algorithm. *Nucleic Acids Res.*, **25**, 4570–4580.

Pascarella,S. and Argos,P. (1992) A data bank merging related protein structures and sequences. *Protein Eng.*, **5**, 121–137.

Rost,B. (1997) AQUA Server. http://www.ebi.ac.uk/~rost/Aqua/aqua.html

Sibbald,P.R. and Argos,P. (1990) Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *J. Mol. Biol.*, **216**, 813–818.

Sjolander,K., Karplus,K., Brown,M., Huguey,R., Krogh,A., Saira,M. and Haussler,D. (1996) Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Applic. Biosci.*, **12**, 327–345.

Taylor,W.R. (1988) A flexible method to align large numbers of biological sequences. *J. Mol. Evol.*, **28**, 161–169.

Thompson,J., Higgins,D. and Gibson,T. (1994a) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4690.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994b) Improved sensitivity of profile searches through the use of sequence weights and gab excision. *Comput. Applic. Biolsci.*, **10**, 19–29.

Vingron,M. and Argos,P. (1990) Determination of reliable regions in protein sequence alignment. *Protein Eng.*, **3**, 565–569.

Vingron,M. and Argos,P. (1991) Motif recognition and alignment for many sequences by comparison of dot-matrices. *J. Mol. Biol.*, **218**, 33–43.

Vingron,M. and Sibbald,P. (1993) Weighting in sequence space: a comparison of methods in terms of generalized sequences. *Proc. Natl Acad. Sci.*, **90**, 8777–8781.

Vogt,G., Etzold,T. and Argos,P. (1995) An assessement of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited. *J. Mol. Biol.*, **299**, 816–831.

Wang,L. and Jiang,T. (1994) On the complexity of multiple sequence alignment. *J. Comput. Biol.*, **1**, 337–348.