doi:10.1006/jmbi.2000.4042 available online at http://www.idealibrary.com on IDEAL® J. Mol. Biol. (2000) **302**, 205–217

JMB

T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment

Cédric Notredame[1,2,3]*, Desmond G. Higgins[4] and Jaap Heringa[1]

[1]National Institute for Medical Research, The Ridgeway, Mill Hill, London NW7 1AA, UK

[2]ISREC, 155, Ch. des Boveresses, CH, 1066 Epalinges/s Lausanne Switzerland

[3]Information Genetique et Structurale, CNRS-UMR 1889 31 Ch. Joseph Aiguier 13402 Marseille, France

[4]Department of Biochemistry University College, Cork Ireland

*Corresponding author

We describe a new method (T-Coffee) for multiple sequence alignment that provides afice in speed as compared to the most commonly used alternatives. The method is broadly based on the popular progressive approach to multiple alignment but avoids the most serious pitfalls caused by the greedy nature of this algorithm. With T-Coffee we pre-process a data set of all pair-wise alignments between the sequences. This provides us with a library of alignment information that can be used to guide the progressive alignment. Intermediate alignments are then based not only on the sequences to be aligned next but also on how all of the sequences align with each other. This alignment information can be derived from heterogeneous sources such as a mixture of alignment programs and/or structure superposition. Here, we illustrate the power of the approach by using a combination of local and global pair-wise alignments to generate the library. The resulting alignments are significantly more reliable, as determined by comparison with a set of 141 test cases, than any of the popular alternatives that we tried. The improvement, especially clear with the more difficult test cases, is always visible, regardless of the phylogenetic spread of the sequences in the tests.

Keywords: pair-wise alignment; progressive alignment; local alignment; global alignment; multiple sequence alignment

Introduction

The simultaneous alignment of three or more nucleotide or amino acid sequences is one of the commonest tasks in bioinformatics. Multiple alignments are an essential pre-requisite to many further analyses of protein families such as homology modeling or phylogenetic reconstruction, or are simply used to illustrate conserved and variable sites within a family. These alignments may be further used to derive profiles (Gribskov et al., 1987) or hidden Markov models (Bucher et al., 1996; Haussler et al., 1993) that can be used to scour databases for distantly related members of the family.

The automatic generation of an accurate multiple alignment is potentially a daunting task. Ideally, one would make use of an in-depth knowledge of the evolutionary and structural relationships within the family, but this information is often lacking or difficult to use. General empirical models of protein evolution (Benner et al., 1992; Dayhoff, 1978; Henikoff & Henikoff, 1992) are widely used instead, but these can be difficult to apply when the sequences are less than 30 % identical (Sander & Schneider, 1991). Further, mathematically sound methods for carrying out alignments, using these models, can be extremely demanding in computer resources for more than a handful of sequences (Carrillo & Lipman, 1988; Wang & Jiang, 1994). In practice, heuristic methods are used for all but the smallest data sets.

The most commonly used heuristic methods are based on the progressive-alignment strategy (Feng & Doolittle, 1987; Hogeweg & Hesper, 1984; Taylor, 1988). with ClustalW (Thompson et al., 1994) being the most widely used implementation. The idea is to take an initial, approximate, phylogenetic tree between the sequences and to gradually build up the alignment, following the order in the tree. Although successful in a wide variety of cases, this method suffers from its greediness. Errors made in the first alignments cannot be rectified later as the rest of the sequences are added in. T-Coffee is an attempt to minimize that effect, and

Abbreviations used: 3D, three-dimensional; SP, sum-of-pairs; NBS(s), nucleotide-binding site(s).

E-mail address of the corresponding author: cedric.notredame@europe.com

0022-2836/00/010205–13 $35.00/0

although the strategy we propose here is also a greedy progressive method, it allows for much better use of information in the early stages, as we will see below.

The main alternative to progressive alignment is the simultaneous alignment of all the sequences. Two such packages exist (MSA (Lipman et al., 1989) and DCA (Stoye et al., 1997)), based on the Carrilo and Lipman (1988) algorithm, but they remain an extremely CPU and memory-intensive approach. Iterative strategies (Gotoh, 1996; Notredame & Higgins, 1996) are another interesting alternative. They do not provide any guarantees about finding optimal solutions but are reasonably robust and much less sensitive to the number of sequences than their deterministic counterparts.

All of these methods attempt to carry out global alignments, where one tries to align the full lengths of the sequences with each other. Alternatively, one might wish to consider local similarity, as occurs when two proteins share only a domain or motif. For two-sequence comparisons, there is the well-known Smith and Waterman (1981) algorithm. Here we use Lalign (Huang & Miller, 1991), from the FASTA package (Pearson & Lipman, 1988), which is a variant of the Smith and Waterman method. It produces sets of non-overlapping local alignments from the comparison of two sequences. For multiple sequences, the Gibbs sampler (Lawrence et al., 1993) and Dialign2 (Morgenstern, 1999) are the main automatic methods. These programs often perform well when there is a clear block of ungapped alignment shared by all of the sequences. They perform poorly, however, on general sets of test cases when compared with global methods (Thompson et al., 1999b; this work). In principle, a method able to combine the best properties of global and local multiple alignments might be very powerful. This is the second motivation for T-Coffee: the design of a method that provides a simple, flexible and, most importantly, accurate solution to the problem of how to combine information of this sort. Accuracy is tested as overall performance on 141 test case alignments from the BaliBase collection (Thompson et al., 1999a,b).

T-Coffee Algorithm

T-Coffee (Tree-based Consistency Objective Function for alignment Evaluation) has two main features. First, it provides a simple and flexible means of generating multiple alignments, using heterogeneous data sources. The data from these sources are provided to T-Coffee via a library of pair-wise alignments. Here we demonstrate the power of T-Coffee by computing multiple alignments using a library that was generated using a mixture of local and global pair-wise alignments (Figure 1).

The second main feature of T-Coffee is the optimization method, which is used to find the multiple alignment that best fits the pair-wise alignments in the input library. We use a so-called progressive strategy (Feng & Doolittle, 1987; Taylor, 1988; Thompson et al., 1994), which is similar to that used in ClustalW. This has the advantage of being fast and relatively robust. With T-Coffee, however, we make use of the information in the library to carry out progressive alignment in a manner that allows us to consider the alignments between all the pairs while we carry out each step of the progressive multiple alignment. This gives us progressive alignment, with all its advantages of speed and simplicity, but with a far lesser tendency to make errors like the one shown in Figure 2(a), i.e. misalignment of the word CAT. T-Coffee is a progressive alignment with an ability to consider information from all of the sequences during each alignment step, not just those being aligned at that stage.

Generating a primary library of alignments

The primary library contains a set of pair-wise alignments between all of the sequences to be aligned. We use the structure described by Notredame et al. (1998). This does not require the alignments to be consistent (e.g. two or more different alignments of the same pair of sequences can be included). In the library, we include information on each of the $N(N-1)/2$ sequence pairs, where N is the number of sequences. Here, we use two alignment sources for each pair of sequences, one local and one global. The global alignments (Figures 1 and 2(b)) are constructed using ClustalW on the sequences, two at a time (default parameters; version 1.75). This is used to give one full-length alignment between each pair of sequences. The local alignments (Figure 1) are the ten top-scoring non-intersecting local alignments, between each pair of sequences, gathered using the Lalign program of the FASTA package with default parameters. Lalign is the FASTA implementation of the Sim program (Huang & Miller, 1991; Pearson & Lipman, 1988).

In the library, each alignment is represented as a list of pair-wise residue matches (e.g. residue x of sequence A is aligned with residue y of sequence B). In effect, each of these pairs is a constraint. All of these constraints are not equally important. Some may come from parts of alignments that are more likely to be correct. We take this into account when computing the multiple alignment and give priority to the most reliable residue pairs. This is achieved by using a weighting scheme.

Derivation of the primary library weights

T-Coffee assigns a weight to each pair of aligned residues in the library (Figure 2(b)). An ideal primary weight will reflect the correctness of a constraint. We use sequence identity, which is known

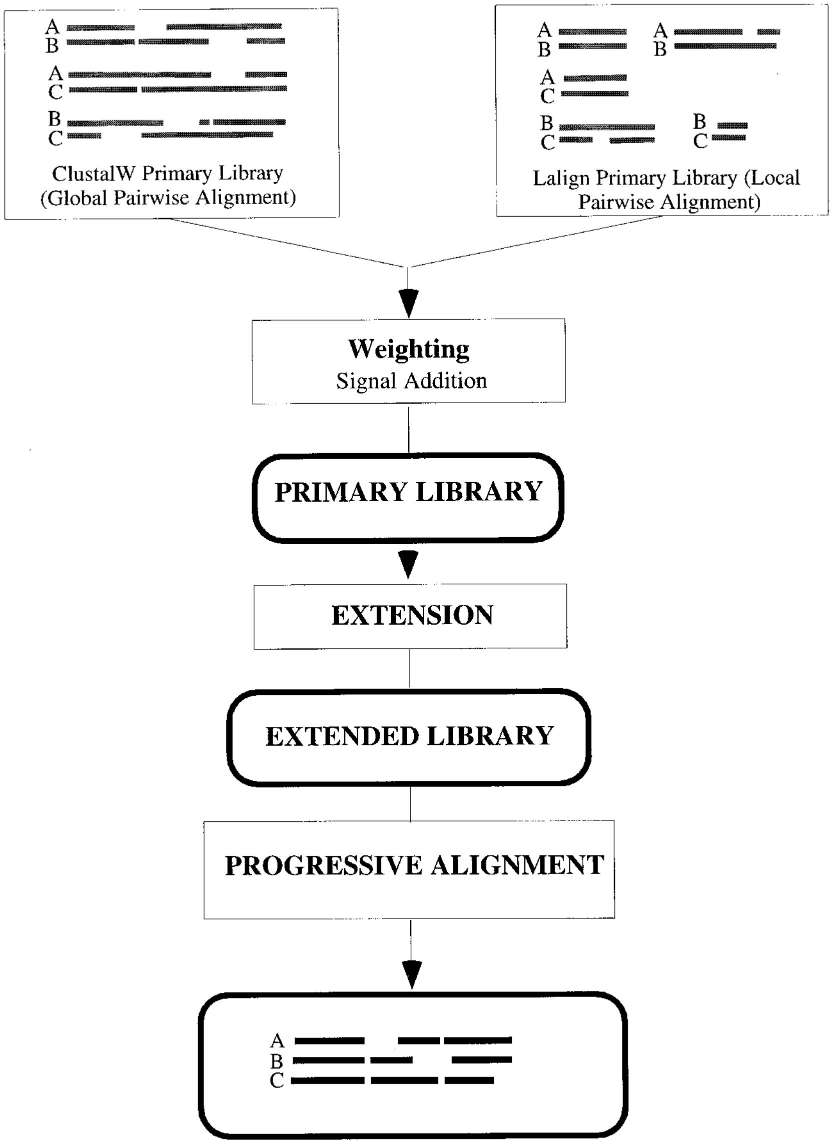

Figure 1. Layout of the T-Coffee strategy; the main steps required to compute a multiple sequence alignment using the T-Coffee method. Square blocks designate procedures while rounded blocks indicate data structures.

to be a reasonable indicator of accuracy when aligning sequences with more than 30% identity (Sander & Schneider, 1991). This weighting scheme proved to be highly effective for a previous consistency-based objective function (Notredame et al., 1998). It also has the advantage of great simplicity. Libraries are lists of weighted pair-wise constraints. Each constraint receives a weight equal to percent identity within the pair-wise alignment it comes from (Figure 2(b)). For each set of sequences, two primary libraries are computed along with their weights, one using ClustalW (global alignments; Figure 2(b)) and the second using Lalign (local).

Combination of the libraries

Our aim is the efficient combination of local and global alignment information. This is achieved by pooling the ClustalW and Lalign primary libraries in a simple process of addition. If any pair is duplicated between the two libraries, it is merged into a single entry that has a weight equal to the sum of the two weights. Otherwise, a new entry is created for the pair being considered. This "stacking" of the signal is similar to previously described strategies (Bucka-Lassen et al., 1999; Heringa, 1999; Taylor, 1999). Pairs of residues that did not occur are not represented (by default they will be considered to have a weight of zero).

This primary library can be used directly to compute a multiple sequence alignment. We could find an alignment that best matched the weighted pairs of residues. However, we enormously increase the value of the information in the library by examining the consistency of each pair of residues with residue pairs from all of the other alignments. For each pair of aligned residues in the library, we can assign a weight that reflects the degree to which those residues align consistently with residues

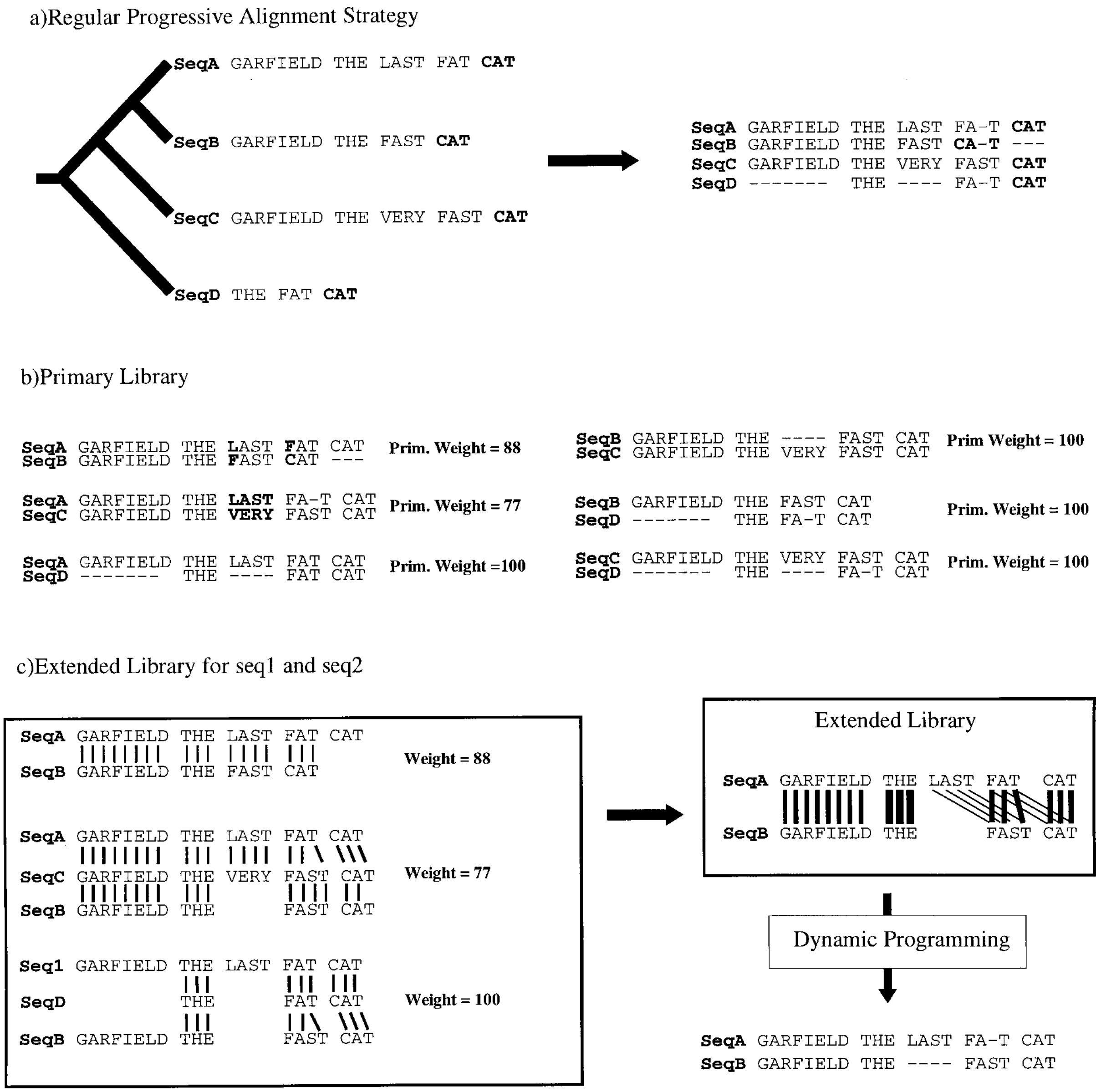
a)Regular Progressive Alignment Strategy
SeqA GARFIELD THE LAST FAT CAT
SeqB GARFIELD THE FAST CAT
SeqC GARFIELD THE VERY FAST CAT
SeqD THE FAT CAT
SeqA GARFIELD THE LAST FA-T CAT
SeqB GARFIELD THE FAST CA-T ---
SeqC GARFIELD THE VERY FAST CAT
SeqD ------- THE ---- FA-T CAT
b)Primary Library
SeqA GARFIELD THE LAST FAT CAT
SeqB GARFIELD THE FAST CAT ---
Prim. Weight = 88
SeqB GARFIELD THE ---- FAST CAT
SeqC GARFIELD THE VERY FAST CAT
Prim Weight = 100
SeqA GARFIELD THE LAST FA-T CAT
SeqC GARFIELD THE VERY FAST CAT
Prim. Weight = 77
SeqB GARFIELD THE FAST CAT
SeqD ------- THE FA-T CAT
Prim. Weight = 100
SeqA GARFIELD THE LAST FAT CAT
SeqD ------- THE ---- FAT CAT
Prim. Weight =100
SeqC GARFIELD THE VERY FAST CAT
SeqD ------- THE ---- FA-T CAT
Prim. Weight = 100
c)Extended Library for seq1 and seq2
SeqA GARFIELD THE LAST FAT CAT
SeqB GARFIELD THE FAST CAT
Weight = 88
SeqA GARFIELD THE LAST FAT CAT
SeqC GARFIELD THE VERY FAST CAT
SeqB GARFIELD THE FAST CAT
Weight = 77
Seq1 GARFIELD THE LAST FAT CAT
SeqD THE FAT CAT
SeqB GARFIELD THE FAST CAT
Weight = 100
Extended Library
SeqA GARFIELD THE LAST FAT CAT
SeqB GARFIELD THE FAST CAT
Dynamic Programming
SeqA GARFIELD THE LAST FA-T CAT
SeqB GARFIELD THE ---- FAST CAT

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.

from all the other sequences. This process is called library extension.

Extending the library

Fitting a set of weighted constraints into a multiple alignment is a well-known problem, formulated by Kececioglu as an instance of the "maximum weight trace", an NP-complete problem (Kececioglu, 1993). Recently, two optimization strategies were proposed (Notredame et al., 1998; Reinert et al., 1997). The first one relies on a genetic algorithm while the second is based on a graph-theoretical method using a branch and bound algorithm. Neither of these methods is entirely satisfactory. The genetic algorithm (Notredame & Higgins, 1996) is robust but may require prohibitive computation time. The graph-theory-based algorithm has a complexity only partially characterized and may fail in some cases for reasons that are difficult to predict.

We circumvent the problem by using a heuristic algorithm that we call library extension (Figure 2(c)). The overall idea is to combine information in such a manner that the final weight, for any pair of residues, reflects some of the information contained in the whole library. To do so, a triplet approach is used, as summarized in Figure 2(c). The strategy bears some similarities to the concept of overlapping weights developed in Dialign2 (Morgenstern, 1999) or the intermediate-sequence method proposed by Neuwald *et al.* (1997) for searching databases. It is based on taking each aligned residue pair from the library and checking the alignment of the two residues with residues from the remaining sequences. For instance, let us consider the four sequences A, B, C and D of Figure 2. Let us call A(G) the G of GARFIELD in sequence A, B(G) the equivalent G in sequence B and $W(A(G), B(G))$ the weight associated with this pair of symbols in the primary library. In the direct alignment of A and B, A(G) and B(G) are matched (Figure 2(b) and (c)). Therefore, the initial weight for that pair of residues can be set to 88 (primary weight of the alignment of sequence A and B, which is the percent of identity of this pair).

If we now look at the alignment of sequence A and sequence B through sequence C (Figure 2(c)), we can see that the A(G) and C(G) are aligned, as well as C(G) and A(G). We conclude that there is an alignment of A(G) with B(G) through sequence C. We associate that alignment with a weight equal to the minimum of $W_1 = W(A(G), C(G))$ and $W_2 = W(C(G), B(G))$. Since $W_1 = 77$ and $W_2 = 100$, the resulting weight is set to 77. In the extended library, this new value is added to the previous one to give a total weight of 165 (i.e. $77 + 88$) for the pair A(G), B(G).

The complete extension will require an examination of all the remaining triplets. Not all of them bring information. For instance, the alignment of A and B through sequence D does not contain any information relative to A(G) or B(G), and, therefore, it has no influence on the weight associated with A(G) and B(G). In summary, the weight associated with a pair of residues will be the sum of all the weights gathered through the examination of all the triplets involving that pair. The more intermediate sequences supporting the alignment of that pair, the higher its weight. Extension will be carried out on each pair of residues of A and B. Once the operation is complete, sequence pair A and B will have gathered information from all the other sequences in the set. This scenario is repeated for each remaining pair (AC, AD, BC, BD, CD) of sequences. The complete set of pairs constitutes the extended library. The worst-case complexity of this computation is $O(N^3L^2)$ with L being the average sequence length. However, this will only occur when all the included pair-wise alignments are totally inconsistent. In practice, with the data sets used here, the complexity is closer to $(O)N^3L$.

Weights will be zero for any residue pairs that never occur (this will be true of the majority of residue pairs). Otherwise, the weight will reflect a combination of the similarity of the pair of sequences or sequence segments that the residue pair comes from and the consistency of that residue pair with all other residue pairs in the primary library. These scores can then be used to align any two sequences from our data set using conventional dynamic programming (Gotoh, 1982). When one normally aligns a pair of sequences, one uses a set of scores derived from some general table of amino acid weights such as a Blosum matrix (Henikoff & Henikoff, 1992). In our case, we can replace that matrix with a set of scores that are specific to every possible pair of residues in our two sequences. This will allow an alignment to be carried out that will take account of the particular residues in the two sequences but will also be guided towards consistency with all of the other sequences in the data set (Figure 2(c)). This is a powerful ability and can be used to carry out progressive alignment while avoiding many of the local-minimum problems normally associated with that technique.

Progressive alignment strategy

In the progressive alignment (Thompson *et al.*, 1994), pair-wise alignments are first made to produce a distance matrix between all the sequences, which in turn is used to produce a guide tree using the neighbor-joining method (Saitou & Nei, 1987). This is a phylogenetic tree, which is used to direct the grouping of sequences during the multiple alignment process (Figure 2(a)). The closest two sequences on the tree are aligned first using normal dynamic programming. This alignment uses the weights in the extended library above to align the residues in the two sequences. This pair of sequences is then fixed and any gaps that have been introduced cannot be shifted later. Then the next closest two sequences are aligned or a sequence is added to the existing alignment of the first two sequences, depending which is suggested by the guide tree. The next two closest sequences or pre-aligned group of sequences are always joined. This continues until all the sequences have been aligned. To align two groups of pre-aligned sequences the scores from the extended library are used, as before, but the average library scores in each column of existing alignment are taken.

As used here, the procedure does not require any additional parameters such as gap penalties. This stems, in part, from the fact that the substitution values (the library weights) were computed on alignments where such penalties had already been applied. Furthermore, high scoring segments that show consistency within the data set see their score enhanced by the extension to such a point that they become insensitive to gap penalties. In practice, this means that during the progressive phase, we use a dynamic-programming algorithm (Gotoh, 1982) with gap-opening penalties and gap-extension penalties set to zero for aligning two sequences or two groups of pre-aligned sequences.

Biological validation of the results

In order to test the accuracy of our method, we used the BaliBase database of multiple sequence alignments (Thompson et al., 1999a,b). This collection contains 141 protein alignments that we use as test cases. For most members within each test case, a three-dimensional (3D) structure is available. The BaliBase multiple alignments were constructed by manual structure comparison and validated using structure-superposition algorithms such as SSAP (Orengo & Taylor, 1996) or DALI (Holm & Sander, 1995). The alignments are thus unlikely to be biased toward any specific multiple-alignment method. For analysis purposes the authors have annotated these alignments by marking blocks of columns deemed to be correctly aligned. Such decisions were made in a conservative manner, only including blocks for which structural evidence is conclusive. This removes most scope for human error but also removes many sections where there are no meaningful alignment between the structures. Altogether, these trusted regions represent 58 % of the aligned residues and have a level of identity on average 5 percentage points higher than that of the complete alignment. There are five basic categories of alignments (families) in BaliBase. They encompass most of the situations that arise when making multiple sequence alignments. The level of average identity within each BaliBase alignment can be seen in Figure 3; it ranges from 10 to 70 %. The coverage is similar for each of the five categories. The first category is made up of phylogenetically equidistant members. In the second category, each alignment contains one orphan sequence with a group of close relatives. The third category contains two distant groups, while the fourth and fifth categories, respectively, involve long terminal and internal insertions. Overall, these 141 test cases constitute the most versatile and sensitive benchmark available today for assessing the accuracy of multiple sequence alignment methods (Thompson et al., 1999b). The version of BaliBase used here is the one that was publicly available in January 1999, and is a more recent version, with different alignment files, than that used in the analysis by Thompson et al. (1999b). The differences between the two BaliBase releases

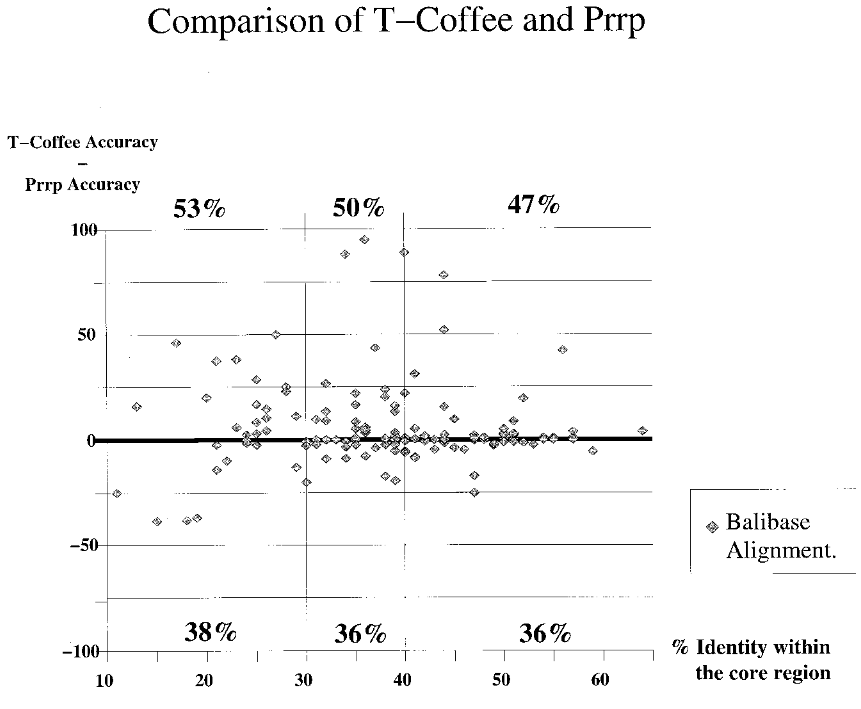

Figure 3. Comparison between T-Coffee and Prrp. For each alignment in BaliBase, the average level of pair-wise identity was measured on the core regions of the reference alignment. Alignment accuracy was assessed for T-Coffee and Prrp on these core regions. The latter two values were subtracted (%T-Coffee accuracy −%Prrp accuracy) and plotted versus the average identity for the alignment. Points in the top area indicate alignments where T-Coffee is outperforming Prrp and inversely for the bottom area. Alignments have been divided into three sets: below 30 % identity (34 alignments), between 30 and 40 % identity (52 alignments) and above 40 % identity (55 alignments). The percentages given in the corners of the plot indicate the fraction of alignments for which T-Coffee outperforms Prrp (top) and vice-versa (bottom). These percentages do not add up to one hundred as for some alignments the same accuracy was obtained with each method (e.g. for alignments having less than 30 % identity, T-Coffee outperforms Prrp in 53 % of the cases, Prrp outperforms T-Coffee in 38 % of the cases, and the two methods draw in 9 % of the cases).

mostly affect category 4, where four test-cases out of 13 are different between the two releases.

Validation is carried out by comparing each calculated multiple alignment with its counterpart in BaliBase. The scoring scheme is the percentage of the trusted columns in the reference that have been correctly aligned. This column-wise comparison has been described as being more sensitive and discriminating (Thompson et al., 1999b) than the alternative pair-wise comparison used by Gotoh (1996), especially in the case of categories 2 and 3 of BaliBase. In the context of this work, the column measure, applied to the trusted regions, is the default that was used to generate our results.

For comparison purposes, we also implemented the so-called sum-of-pairs (SP) measure, where a calculated alignment is assessed on proper alignment of all possible pairs of residues in each of the alignment columns. This measure generally leads to a more gradual loss of score in case of misalignment than the above column-count measure. The comparison routine we used was devised following Baliscore, the program made available and used by Thompson et al. (1999b), although some updates were effected to ensure accurate alignment comparison.

Comparison with other methods

To compare T-Coffee with other methods, we produced multiple alignments of each BaliBase family with other programs. We evaluated three such packages here. They include the methods described as performing best by the authors of BaliBase. Together, they cover a large portion of the existing algorithms for multiple sequence alignments. Prrp (Gotoh, 1996) attempts to simultaneously align all the sequences in an iterative manner. It uses a scoring function very similar to the MSA program (weighted sums of pairs; Lipman et al., 1989). ClustalW (Thompson et al., 1994) is a progressive-alignment method. Dialign2 (Morgenstern, 1999) is a segment-based method that constructs the multiple alignment by assembling a collection of high-scoring segments in a sequence-independent progressive manner. Methods based on multidimensional dynamic programming like MSA (Lipman et al., 1989) or DCA (Stoye, 1998) could not be used in the evaluation as they aborted the construction of alignments in about 10 % of the BaliBase sets. For the alignments that MSA and DCA could construct, the accuracy was comparable to Prrp.

We constructed the alignments by extracting the sequences from the BaliBase reference alignments and realigning them with a given program. In each case, the parameters used were the default settings provided by the authors. We made no attempt of tuning, either on T-Coffee and its associated methods, or on the methods used for comparison. The packages used here are those that were available in January 1999 when they were downloaded from the sites indicated by the authors in their respective publications and installed on our machines.

Statistical validation

It is critical to establish whether differences observed between two methods are statistically meaningful. We used the same strategy as Gotoh (1996), which involves applying the Wilcoxon signed matched-pair ranked test on the results obtained with two methods on the 141 BaliBase alignments. This non-parametric test allows the association of a P-value with the differences measured on these two series of results. In that case, the P-value is the probability that the observed differences may be due to chance. The lower the P-value, the more significant the result.

Implementation

T-Coffee is implemented in ANSI C. Its tree-parsing and tree-calculating facilities were taken from the ClustalW package, and it uses a modified version of the Lalign program. T-Coffee is available free of charge on request from the authors and is distributed with documentation and examples (send a request to cedric.notredame@europe.com). Here, the program was run on a LINUX platform with Pentium II processors (330 MHz).

Results

Combining local and global alignments without extension

The effect of combining local and global alignments is shown in Table 1. Three alternative primary libraries (i.e. without extension) were used to make the alignments: the ClustalW pair-wise library (C), the Lalign pair-wise library (L), and pooling of the ClustalW and Lalign pair-wise libraries (CL). In each of the five BaliBase categories, the combination of local and global information (CL) induced a statistically meaningful improvement over the two single method-based protocols (Table 1). On average (Total in Table 1), CL is at least 7.6 percentage points better than C or L. The Wilcoxon test shows that these differences are associated with P-values lower than 0.001.

Effect of the library extension

The three previously used libraries (C, L, CL) were extended. In all three cases, extended libraries (CE, LE, CLE) induced significantly improved performances when compared to their non-extended counterparts (C, L, CL), regardless of the BaliBase category (Table 1). Most importantly, CLE significantly outperforms all of the alternative protocols in all categories. Table 1 also shows that the performance of CLE is highly sustained, while in contrast, the second-best protocol varies over the BaliBase categories (CE in Cat1 and 2, CL for Cat3,

Table 1. The effect of combining local and global alignments

Name	Protocol			Cat1 (81)	Cat2 (23)	Cat3 (4)	Cat4 (12)	Cat5 (11)	Total (141)	Significance
C	ClustalW pw	. . .	. . .	70.6	26.7	43.0	56.0	60.0	58.9	7.8[a]
CE	ClustalW pw	. . .	extend	77.1	33.6	47.6	64.8	75.9	66.3	17.7[a]
L	. . .	Lalign pw	. . .	65.4	12.1	22.8	53.9	66.0	52.0	7.8[a]
LE	. . .	Lalign pw	extend	72.6	25.6	47.2	77.5	85.5	64.2	16.3[a]
CL	ClustalW pw	Lalign pw	. . .	76.2	32.0	48.3	76.2	74.6	66.5	12.1[a]
CLE	ClustalW pw	Lalign pw	extend	**80.7**	**37.3**	**52.9**	**83.2**	**88.7**	**72.1**	. . .

Protocol shows the way the library was created. ClustalW pw and Lalign pw show the pair-wise alignments computed with one of these programs, using default parameters. Extend indicates that the library was extended before progressive alignment. CLE uses a combination of ClustalW and Lalign alignments and library extension. Cat1 to Cat5 are the five reference categories of BaliBase; number sin parentheses indicate the number of alignments in a category. The average accuracy is then given for each protocol. The best accuracies in each column are shown in bold and underlined. Total gives the average accuracy across all 141 test alignments. The last column shows the percentage of times that CLE is outperformed by each other protocol. The statistical significance of the improvement of CLE over each protocol is shown by

[a] ($P < 0.001$).

LE in Cat4 and Cat5). These results show that the combination of local (Lalign) and global (ClustalW) information boosts the quality of multiple alignment. Table 1 indicates that the CLE protocol is outperformed by the second-best protocol (CL) in only about 12 % of the cases, as assessed over 141 BaliBase alignments.

Comparing T-Coffee with other multiple sequence alignment methods

The protocol used to assess the four methods (Dialign2, ClustalW, Prrp and T-Coffee (CLE)) is identical with that described in the previous section, and the results are organized in a similar layout (Table 2). Each program was executed using its default parameters (see, Using T-Coffee).

T-Coffee (CLE protocol) shows the highest average accuracy in each BaliBase category (Table 2), even in category 4 where long internal deletions require a method able to deal with local similarity such as Dialign2. These differences are all statistically significant (Table 2). When considering the unweighted average accuracy over the five categories (Table 2, Total2) T-Coffee is 9.7 % more accurate than the next-best method, Prrp. In most BaliBase categories, Prrp is the second-best method, slightly outperforming ClustalW, as reported by Gotoh (1996). Repeating the evaluation on the complete alignments (as opposed to the core regions only) shows that the trend is persistent: T-Coffee still outperforms all the other methods in the five categories. However, when measured over the complete alignments, ClustalW becomes the second-best protocol, with an unweighted average accuracy of 43.7 % as opposed to 48.7 % for T-Coffee. The difference of 5 % in performance is statistically significant, as the Wilcoxon test results in a P-value lower than 0.01.

These alignments were also evaluated using the sum-of-pairs measure, where each pair of residues is compared between the two alignments. This measure is less drastic than the column measure as it allows one to score columns that are partially correct. For instance, it tolerates the complete misalignment of one sequence without making all the columns count as being wrong. This measurement, carried out on the annotated blocks of BaliBase, gave similar results: T-Coffee outperforms the other methods in the five categories. The differences between the methods are slightly less pronounced: T-coffee achieves 89.7 % of the pairs correctly aligned, while Prrp, the second-best, aligns 86.2 % of the pairs correctly. ClustalW comes third, with 85.6 % of the pairs correctly aligned. We conclude that the increase in alignment accuracy observed with T-Coffee is significant and consistent over the two generally applied accuracy measures used here.

Most of the improvement with T-Coffee tends to concentrate in the BaliBase alignments having a low level of average identity. Figure 3 follows the representation proposed by Gotoh for comparing two methods (Gotoh, 1996) and shows that the alignments for test-cases with less than 30 % average sequence identity improve the most. The Figure shows that at this low identity level, there is an almost two-thirds chance of obtaining the best alignment when using T-Coffee rather than Prrp.

Application to serine/threonine kinases

A major application of any alignment algorithm will be the delineation of motifs or domains. In Figure 4 we show an example that illustrates the usefulness of T-Coffee for identifying functional features of a series of kinases taken from BaliBase (kinase3 in ref5). These proteins belong to a subfamily of protein serine/threonine kinases. Each sequence is identified by its SwissProt identifier except for gcn2, which is from PDB. A 3D structure is also available for 11 of these sequences. Each of the 19 sequences in the alignment contains a nucleotide-binding site (NBS), marked by bold-type capital letters in Figure 4. In all these sequences, the NBS is followed by a second conserved motif toward the C terminus (also marked in capital letters). T-Coffee was able to accurately align 18 of the 19 NBSs, as were Dialign2 and Prrp. ClustalW was only able to correctly align 16 of these NBSs. The second motif is more difficult because of the long indel in st11_yeast. Here as well, T-Coffee can properly align 18 of the motifs, while Prrp and ClustalW get 15 correct, and Dialign2 only 13. This trend is confirmed with the measure of the accuracy on that portion of the alignment. The column measure indicates a score of 0 % for ClustalW and Prrp, 30.9 % for Dialign2 and 39.8 % for T-Coffee. The SP measure gives a score of 65.4 % for ClustalW, 73.1 % for Prrp, 83.0 % for Dialign2, and 92.7 % for T-Coffee. The Gibbs sampler (Lawrence *et al.*, 1993) was also attempted on the set of kinase sequences, but could never align more than ten of the motifs (and only when provided with an estimate of the total number of blocks in the alignment). As a result of combining local and global alignment information, T-Coffee managed to align almost all of the motifs as in the BaliBase reference alignment. Moreover, T-Coffee was the only program that correctly aligned the second motif of kp68_human, which is an interferon-induced kinase and an essential component of the viral response. It is activated by interacting with double-stranded RNA (Meurs *et al.*, 1990), whereupon it induces inhibition of protein synthesis.

Efficiency

The complexity of the whole procedure is given by:

$$O(N^2L^2) + O(N^3L) + O(N^3) + O(NL^2)$$

where $O(N^2L^2)$ is associated with the computation of the pair-wise library, $O(N^3L)$ the extension,

Table 2. T-Coffee compared with other multiple sequence alignment methods

Method	Cat1 (81)	Cat2 (23)	Cat3 (4)	Cat4 (12)	Cat5 (11)	Total1 (141)	Total2 (141)	Significance
Dialign	71.0	25.2	35.1	74.7	80.4	61.5	57.3	11.3[a]
ClustalW	78.5	32.2	42.5	65.7	74.3	66.4	58.6	26.2[a]
Prrp	78.6	32.5	50.2	51.1	82.7	66.4	59.0	36.9[a]
T-Coffee	**80.7**	**37.3**	**52.9**	**83.2**	**88.7**	**72.1**	**68.7**	

Method indicates the name of the method evaluated. T-Coffee is the protocol CLE in Table 1. Total1 gives the average accuracy across all the 141 alignments. Total2 is the average accuracy across the five BaliBase categories (unweighted). The last column shows the percentage of times that T-Coffee is outperformed by each other protocol. The statistical significance of the improvement of T-Coffee over each method is shown by

[a] (P < 0.001). The Table layout is otherwise similar to that of Table 1.

```
                                               NBS
g11a_orysa lshfkllkklgcgdigsvylsels---gtesyfamKVMDKas-------
kp68_human gmdfkeieligsggfgqvfkakhr---idgktyviKRVKYnn-------
gcn2       --tlkrlnfsgqgafgqvvkarna---ldsryyaiKKIRNte-------
st11_yeast pknwlkgacigsgsfgsvylgmna---htgelmavKQVEIknnnigvpt
kin3_yeast rseyqvleeigrgsfgsvrkvihi---ptkkllvrKDIKYgh-------
nima_emeni adkyevlekigcgsfgiirkvkrk---sdgfilcrKEINYik-------
kin1_yeast lgdwefvetvgagsmgkvklakhr---ytnevcavKIVNRat----kaf
kcc1_yeast kkkyvfgktlgagtfgvvrqaknt---etgedvavKILIKka-------
ks62_human psqfellkvlgqgsfgkvflvkkisgsdarqlyamKVLKKat-------
kpc1_yeast ldnfvllkvlgkgnfgkvilsksk---ntdrlcaiKVLKKdn-------
ypk2_yeast iddfdllkvigkgsfgkvmqvrkk---dtqkiyalKALRKay-------
krac_dicdi vadfellnlvgkgsfgkviqvrkk---dtgevyamKVLSKkh-------
kgp2_drome ltdlrviatlgvggfgrvelvqtn--gdssrsfalkqmkksq-------
kapa_mouse ldqfdriktlgtgsfgrvmlvkhk---esgnhyamKILDKqk-------
kdca_drome lenyitravlgngsfgtvmlvrek---sgknyyaaKMMSKed-------
ark1_human mndfsvhriigrggfgevygcrkr---dtgkmyamKCLDKkr-------
dmk_human  rddfeilkvigrgafsevavvkmk---qtgqvyamKIMNKwd-------
dbf2_yeast nrdfemitqvgqggygqvylarkk---dtkevcalKILNKkl-------
pim1_human esqyqvgpllgsggfgsvysgirv---sdnlpvaiKHVEKdr-------

g11a_orysa ---------------------------------lasrk-kl---lraqt
kp68_human ---------------------------------ekaer-----------
gcn2       -------------------------------------e-kl---stmis
st11_yeast nnkqansdenneqeeqqekiedvgavshpktnqnihrk-mv---dalqh
kin3_yeast ---------------------------------mns-k-er---qqlia
nima_emeni ---------------------------------mst-k-er---eqlta
kin1_yeast ---------------------------------lhkeq-ml---pppkn
kcc1_yeast ---------------------------------lkgnkvql---ealyd
ks62_human ---------------------------------lkvr--dr---vrtkm
kpc1_yeast ---------------------------------iiqnh-di---esara
ypk2_yeast ---------------------------------ivskc-ev---thtla
krac_dicdi ---------------------------------ivehn-ev---ehtls
kgp2_drome ---------------------------------ivetr-qq---qhims
kapa_mouse ---------------------------------vvklk-qi---ehtln
kdca_drome ---------------------------------lvrlk-qv---ahvhn
ark1_human ---------------------------------ikmkq-ge---tlaln
dmk_human  ---------------------------------mlkrg-ev---scfre
dbf2_yeast ---------------------------------lfkln-et---khvlt
[illegible]
```

Figure 4. Example of a T-Coffee alignment. This N-terminal alignment of 19 kinases shows two boxes containing the nucleotide-binding site and a conserved motif. The residues in capital letters are annotated as core regions in BaliBase. The core residues in red are correctly aligned with respect to the BaliBase reference. This alignment belongs to BaliBase category 5 (long insertion).

$O(N^3)$ the computation of the NJ tree and $O(NL^2)$ the computation of the progressive alignment (assuming N sequences of length L that can be aligned in a multiple alignment of length L).

The CPU time consumption of T-Coffee was analyzed empirically. Our measurements (data not shown) indicate that with alignments of similar size as those considered here, the apparent complexity of the program is quadratic, both relative to the average sequence lengths and to the number of sequences. This result can be explained by the fact that in the cases analyzed here, $L \gg N$. Therefore, the time required for the library and the alignment computation is much larger than the time required for the library extension: $O(N^2L^2) + O(NL^2) \gg O(N^3L)$. The complexity of the latter is the same as that of ClustalW, even if in absolute time, the overhead is higher. For instance, given the Lalign and ClustalW primary libraries, T-Coffee is about two times slower than ClustalW.

Discussion

T-Coffee is a new progressive method for sequence alignment. It can combine signals from heterogeneous sources (e.g. sequence-alignment programs, structure alignments, threading, manual alignment, motifs and specific constraints) into a unique consensus multiple sequence alignment. We show here that a combination of local and global alignments leads to a significant increase in alignment accuracy. The method is more accurate than its counterparts and has proved successful in a wide variety of cases.

The main difference from traditional progressive alignment methods is that, instead of using a substitution matrix for aligning the sequences, a position-specific scoring scheme is used (the extended library). Thanks to the extension process, the values contained in the library for a given pair of sequences also depend on information from the other sequences in the set. In this way, errors are less likely to occur during early stages of the progressive alignment. As a consequence, even though the paradigm ''once a gap always a gap'' (Feng & Doolittle, 1987) remains true, misplacing gaps becomes much less likely.

The second important feature of T-Coffee is the combination of local and global information. Although it has long been suspected that such a combination was probably necessary for computing high-quality alignments (McClure *et al*., 1994), to date no satisfactory formula had been found to address this problem efficiently. Through combining local and global alignments from widely used programs with a new formalism, T-Coffee appears to provide a convincing solution. The end-user benefits from the simplicity of the method and does not need to provide any extra parameter values.

A key ingredient of the method is the primary weighting scheme. A shortcoming of the current use of average sequence identity is that this tends to overweight small segments where high similarity is more likely to occur by chance. This is particularly significant when weighting shorter segments obtained from a local alignment program such as Lalign. The main reason why T-Coffee can tolerate such noise is because short high-scoring segments are rarely consistent enough to have a strong effect on the position-specific scoring scheme after extension. Moreover, final alignments are processed using dynamic programming (progressive alignment). This makes it less likely for misplaced high-scoring segments to affect the alignment. For other protocols, which incorporate segments in a multiple alignment following a strict order based on their weight (Morgenstern, 1999), such fortuitous segments can be a major pitfall.

Although the protocol proposed here (Lalign + - ClustalW pair-wise alignments + extension) employs a minimal combination of local and global information, there is no theoretical limit to the number of methods that can be used. For instance, alignments from structural comparisons could be combined with sequence alignments. It is also possible to incorporate, in the library, information extracted from multiple alignments.

Acknowledgments

The authors thank the following people: Philipp Bucher for useful discussions and advice at an early stage of the project, Willie Taylor for useful discussions, Nigel Douglas for providing us with an efficient LINUX environment, and Webb Miller and Bill Pearson for allowing us to use and modify Lalign from the FASTA package. This work was funded in part by grant 3100-49669.96 from the Swiss National Science Foundation.

References

Benner, S. A., Cohen, M. A. & Gonnet, G. H. (1992). Response to Barton's letter: computer speed and sequence comparison. *Science*, **257**, 609-1610.

Bucher, P., Karplus, K., Moeri, N. & Hofmann, K. (1996). A flexible motif search technique based on generalized profiles. *Comput. Chem.* **20**, 3-23.

Bucka-Lassen, K., Caprani, O. & Hein, J. (1999). Combining many multiple alignments in one improved alignment. *Bioinformatics*, **15**, 122-130.

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

Dayhoff, M. O. (1978). *Atlas of Protein Sequence and Structure*, vol. 4, Suppl. 3, National Biomedical Research Foundation, Washington, USA, DC.

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

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.* **162**, 705-708.

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. & Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proc. Natl Acad. Sci. USA*, **84**, 4355-5358.

Haussler, D., Krogh, A., Mian, I. S. & Sjölander, K. (1993). *Proceedings for the 26th Hawaii International Conference on Systems Sciences*, Wailea, HI, USA.

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

Heringa, J. (1999). Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple alignment. *Comput. Chem.* **23**, 341-364.

Hogeweg, P. & Hesper, B. (1984). The alignment of sets of sequences and the construction of phylogenetic trees. An integrated method. *J. Mol. Evol.* **20**, 175-186.

Holm, L. & Sander, C. (1995). *Third International conference on Intelligent Systems for Molecular Biology* (*ISMB*), Cambridge, England.

Huang, X. & Miller, W. (1991). A time-efficient, linear-space local similarity algorithm. *Advan. Appl. Math.* **12**, 337-357.

Kececioglu, J. D. (1993). The maximum weight trace problem in multiple sequence alignment. *Lect. Notes Comput. Sci.* **684**, 106-119.

Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F. & Wootton, J. C. (1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, **62**, 208-214.

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

McClure, M. A., Vasi, T. K. & Fitch, W. M. (1994). Comparative analysis of multiple protein-sequence alignment methods. Mol. Biol. Evol. **11**, 571-592.

Meurs, E., Chong, K., Galabru, J., Thomas, N. S. B., William, B. R. & Hovanessian, A. G. (1990). Molecular cloning and characterization of the human double stranded RNA-activated kinase induced by interferon. Cell, **62**, 379-390.

Morgenstern, B. (1999). Dialign2: improvement of the segment-to-segment approach to multiple sequence alignment. Bioinformatics, **15**, 211-218.

Neuwald, A. F., Liu, J. S., Lipman, D. J. & Lawrence, C. B. (1997). Extracting protein alignment models from the sequence database. Nucl. Acids Res. **25**, 1665-1677.

Notredame, C. & Higgins, D. G. (1996). SAGA: sequence alignment by genetic algorithm. Nucl. Acids Res. **24**, 1515-1524.

Notredame, C., Holm, L. & Higgins, D. G. (1998). COFFEE: an objective function for multiple sequence alignments. Bioinformatics, **14**, 407-422.

Orengo, C. A. & Taylor, W. R. (1996). SSAP: sequential structure alignment program for protein structure comparison. Methods Enzymol. **266**, 617-635.

Pearson, W. R. & Lipman, D. J. (1988). Improved tools for biological sequence comparison. Proc. Natl Acad. Sci. USA, **85**, 2444-2448.

Reinert, K., Lenhof, H. P., Mutzel, P., Meihorn, K. & Kececioglu, J. D. (1997). A branch-and-cut algorithm for multiple sequence alignment. Recomb97, 241-249.

Saitou, N. & Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**, 406-125.

Sander, C. & Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins: Struct. Funct. Genet. **9**, 56-68.

Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. J. Mol. Biol. **147**, 195-197.

Stoye, J. (1998). Multiple sequence alignment with the divide-and-conquer method. Gene, **211**, GC45-56.

Stoye, J., Moulton, V. & Dress, A. W. (1997). DCA: an efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. Comput. Appl. Biosci. **13**, 625-626.

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

Taylor, W. R. (1999). Protein structure comparison using iterated double dynamic programming. Protein Sci. **8**, 654-665.

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

Thompson, J., Plewniak, F. & Poch, O. (1999a). BaliBase: a benchmark alignment database for the evaluation of multiple sequence alignment programs. Bioinformatics, **15**, 87-88.

Thompson, J. D., Plewniak, F. & Poch, O. (1999b). A comprehensive comparison of multiple sequence alignment programs. Nucl. Acids Res. **27**, 2682-2690.

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

Edited by J. Thornton

(Received 2 May 2000; received in revised form 15 June 2000; accepted 10 July 2000)