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

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

[1] MRC
The Ridgeway,
Mill Hill,
London NW7 1AA, U.K.
United Kingdom

[2] ISREC
155 Chemin des Boveresses,
CH-1066 Epalinges/s Lausanne
Switzerland

[3] Department of Biochemistry
University College
Cork Ireland.

Email:
Cédric Notredame cnotred@nimr.mrc.ac.uk
Des Higgins des@chah.ucc.ie
Jaap Heringa jhering@nimr.mrc.ac.uk

Keywords: Pairwise alignment, Progressive alignment, Local alignment, Global alignment, Multiple Sequence alignment.

***To whom correspondence should be addressed.**

ABSTRACT

We describe a new method (T-Coffee) for multiple sequence alignment that provides a dramatic improvement in accuracy with a modest sacrifice 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 pairwise 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. In this paper we illustrate the power of the approach by using a combination of local and global pairwise 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.

INTRODUCTION

The simultaneous alignment of three or more nucleotide or amino acid sequences is one of the commonest tasks in bioinformatics. Virtually all proteins belong to multigene families and there are more and more examples with each protein appearing as the various genome sequencing projects deliver their data. 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 (1) or hidden Markov models (2, 3) 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 (4, 5, 6) are widely used instead but these can be difficult to apply when the sequences are less than 30% identical (7). 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 (8, 9). 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 (10, 11, 12) with ClustalW (13) 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. The sequences

are aligned singly or in groups, two by two, using dynamic programming (14, 15), until the alignment is finished. This method has proved successful in a wide variety of cases and often produces high quality alignments. It does, however, suffer from the algorithmic greediness inherent in this strategy. The first alignments that are made may contain errors and these cannot be rectified later as the rest of the sequences are considered. This directs the alignments into what we refer to as a local minimum (as distinct from a desired global minimum) using an energy landscape as an analogy. T-Coffee is an attempt to minimize this effect, and 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 later.

The main alternative to progressive alignment is the simultaneous alignment of all the sequences. The Carillo and Lipman algorithm (8) and MSA (16), its implementation, allow the simultaneous alignment of up to about 10 sequences. Even here, there are restrictions on the nature or the gap scoring functions that can be used and some data sets cannot be aligned if the sequences are too divergent. MSA was recently extended to larger data sets using a divide and conquer method (17) but remains an extremely CPU and memory intensive approach. Iterative strategies (18, 19) provide an interesting alternative, as these might be applicable to relatively large data sets. These 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 sequences, there is the well known Smith and Waterman algorithm (20). In this paper we use Lalign (21), from the FASTA package(22) which is a variant of Smith and Waterman. It produces sets of non-overlapping local alignments from the comparison of two sequences. For multiple sequences, MACAW (23) is a semi-automatic method that produces blocks of alignments shared by all or a subset of the sequences. The Gibbs sampler (24) and Dialign (25) 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 or where there are blocks of alignment separated by long insertions or deletions. They perform poorly, however, on general sets of test cases when compared with global methods ((26,44) and 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 the T-Coffee method. We generate multiple alignments using a combination of global and local alignment information from each pair of sequences. It provides a simple, flexible and, most importantly, accurate solution to the problem of how to combine information of this sort. In this paper, we only consider combining two sources of information. In principle, there is no limit to the number of sources that could be used. Accuracy is tested as overall performance on 141 test case alignments from the BaliBase collection (26, 27).

METHODS

T-Coffee (Tree based Consistency Objective Function for alignmEnt Evaluation) is a general strategy aimed at combining heterogeneous sources of alignment information into a single multiple sequence alignment. The protocol described here is an efficient way to combine sets of local and global pairwise alignments. This set of alignments is referred to as the primary library (Figure 1). A procedure named matrix extension is used to convert the primary library into a position specific scoring scheme called the extended library. This is used to generate the multiple alignment in a progressive manner, similar to ClustalW.

1-Generating a primary library of alignments

Two types of libraries are used: primary and extended. A primary library may contain local or global alignments. We use the structure described in (28), which does not require the alignments to be consistent. This means that it is possible to have several conflicting alignments in the same library (*e.g.* two different alignments of the same sequences). We include, in the library, information for each of the $N(N-1)/2$ sequence pairs, where N is the number of sequences. For each pair of sequences, global and local alignments are included. Global alignments are constructed using ClustalW with default parameters (version 1.75). Local alignments are the 10 top-scoring non-intersecting local alignments gathered using Lalign (default parameters) from the Fasta package (21, 22). It should be stressed that no multiple alignment is computed for the primary libraries.

In the library, each alignment is represented as a list of residue pairwise matches (e.g. Residue *x* of Sequence A matches 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 portions of alignments more likely to be correct, for example if there is a high level of identity between the sequences or a high level of local similarity. An appropriate strategy must take this into account when computing the multiple alignment and give priority to the most reliable residue pairs. This can be achieved by using a weighting scheme.

2-Derivation of the primary library weights

T-Coffee computes a weight for each pair of aligned residues declared in the library (Figure 2a). An ideal primary weight will reflect the correctness of a constraint. We use sequence identity, a criterion known to be a reasonable indicator of accuracy when aligning sequences with more than 30% identity (7). It is a weighting scheme that proved effective for a previous consistency based objective function (28). Libraries thus generated are lists of weighted pairwise constraints. Each constraint receives a weight equal to percent identity within the pairwise alignment it comes from. For each set of sequences, two primary libraries are computed along with their weights: one with ClustalW (global) and another with Lalign (local).

3-Combination of the libraries

Since our aim is the efficient combination of local and global information, the pooling of the ClustalW and Lalign primary libraries is an essential step. The process is straightforward. If any pair is duplicated between the two libraries, it is replaced with

a single entry that has a weight equal to the sum of the two weights. This 'stacking' of the signal is similar to previously described strategies (29, 30, 31). The size of the resulting library depends on the level of consistency between the two libraries. In the worst case (*i.e.* total inconsistency), it equals the sum of their individual sizes. This primary library can be used directly to compute a multiple sequence alignment. However, accuracy is improved when internal consistency is taken into account. To do this, we devised a reevaluation step named library extension.

4-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 (32). Recently, two heuristic optimization strategies were proposed (28, 33). 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 (28) is rather 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. We attempt to consider information from all sequences simultaneously by an approach which we call extension (Figure 2b). The overall idea is to combine information in such a manner that the final constraints for a given pair of sequences reflect some of the information contained in the whole library. To do so, a triplet approach is used as summarized in Figure 2b. The strategy bears some similarities with the concept of overlapping

weights developed in Dialign (25) or the intermediate sequence method proposed by Neuwald for searching databases (34). It can be explained as follows:

Let us consider 4 sequences A, B, C and D. A(x) is residue x within sequence A, B(y) the residue y of sequence B and $\mathcal{P}$(A(x), B(y),W) the pair associating these two residues with a weight W in the library (*e.g.* A(6) B(14) in Figure 2a). The extension strategy is shown in Figure 2b. If there exist two pairs $\mathcal{P}$(A(x),C(z),W_1) and $\mathcal{P}$(C(z),B(y),W_2), then a new pair $\mathcal{P}$(A(x), B(y),W_3) is added to the library. W_3 is set to the minimum of W_1 and W_2, so that it gets the value of its weakest component (*e.g.* a chain always has the strength of its weakest link). We call this new pair an alignment of A and B through sequence C (A(6) C(15), C(15) B(14) in Figure 2b). The same process is carried out through sequence D and generally for all triplets involving sequences 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). 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 pairwise alignments are totally inconsistent. In practice, with the data sets used here the complexity is closer to (O)N^3L.

As pointed out earlier, the extended library can be regarded as a position specific scoring scheme. For each pair of sequences, it assigns its positive library weight to the amino acid matches corresponding to library pairs, while matches not expressed in the library receive a default value of 0. Several algorithms have been proposed for reconstructing a multiple alignment from lists of weighted constraints (33, 35).

Unfortunately, these algorithms usually have a high complexity and may have problems in dealing with noisy data (*i.e.* incorrect weights). As a more efficient alternative, we use progressive alignment with dynamic programming (10, 11, 12).

5-Progressive alignment strategy

In the progressive alignment strategy we use (13), pairwise comparisons are first made to produce a guide tree using the Neighbor Joining method (36). The progressive multiple alignment then follows the topology of the guide tree. 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. Further, 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 (15) with gap opening penalties and gap extension penalties set to 0 for aligning two sequences or two groups of pre-aligned sequences. The library replaces a standard substitution matrix.

6-Biological validation of the results

In order to test the accuracy of our method, we used the BaliBase database of multiple sequence alignments (26, 27). This collection contains 141 protein families. For most members within each family, a 3D structure is available. The BaliBase multiple alignments were constructed by manual structure comparison and validated using structure superposition algorithms such as SSAP (37) or DALI (38). The alignments

are thus unlikely to be biased toward any specific method. For analysis purposes the authors have annotated these alignments by marking the columns deemed to be correctly aligned. Such decisions were made in a conservative manner, only including blocks for which structural evidence is conclusive. Altogether, these trusted regions represent 58% of the aligned residues (27). There are five basic categories of alignments (families) in BaliBase, encompassing most of the situations that arise when making multiple sequence alignments. The first category is made of phylogenetically equidistant members. In the second category, each family 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 insertions and deletions. Overall, these 141 test cases constitute one of the most versatile and sensitive benchmark available today for assessing the accuracy of multiple sequence alignment methods (26).

Validation is done by comparing a calculated multiple alignment to its counterpart in BaliBase. The scoring scheme is the percentage of the trusted columns in the reference that have been correctly aligned. This columnwise comparison has been described as being more sensitive and discriminating (26) than the alternative pairwise comparison used by Gotoh (19). This is especially true in the case of categories 2 and 3 of BaliBase (26).

7-Comparison with other methods

To compare T-Coffee with other methods, multiple alignments of each BaliBase family were produced with other programs. Four such packages are included in this study. They cover most of the existing types of algorithms for multiple sequence

alignment. SAM is a hidden Markov model (HMM) algorithm (39, 40). It attempts to simultaneously align all the sequences by optimizing the parameters of a hidden Markov model. Prrp (19) also simultaneously aligns all the sequences, but in an iterative manner. ClustalW (13) is a progressive alignment method. Dialign2 (35) 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 (16) or DCA (17, 41) 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. An attempt was also made to use the Gibbs sampler, a local alignment procedure based on stochastic optimization methods (24). Unfortunately, it appeared that this method is not really suitable for most families in BaliBase (lack of well defined ungapped blocks and too few sequences).

8-Statistical validation

It is critical to establish whether differences observed between two methods are statistically meaningful. We used the same strategy as Gotoh (19). It involves applying the Wilcoxon signed matched-pair ranked test on the results obtained with two methods on the 141 BaliBase families. This non-parametric test allows the association of a P-value with the differences measured on these two series of results. This P-value is the probability that the observed differences may be due to chance. The lower the P-value, the more significant the result.

9-Implementation

T-Coffee is implemented in ANSI C. It is available on request from the authors and will be distributed with documentation and examples. For this work, the program was run on a LINUX platform with a Pentium II processor (330 MHz).

RESULTS

1-Combining local and global alignments without extension

The effect of combining local and global alignments is shown in Table 1a and b. Three alternative primary libraries (*i.e.* without extension) were used to make the alignments: the ClustalW pairwise library (C), the Lalign pairwise library (L) and pooling of the ClustalW and Lalign pairwise libraries (CL). The results very clearly indicate that CL is an improvement over C and L. In each of the five BaliBase categories, the combination of local and global information induced an improvement over the two single method-based protocols. On average (Tot in Table 1a), CL is at least 7.6 percentage points better than C or L. The statistical significance of this result is confirmed by the Wilcoxon test (Table 1b) as the observed differences between CL and C or L are associated with P values lower than 0.001. This shows the efficiency of combining local and global information to make an alignment. Next we illustrate the effect of library extension.

2-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 improved performance when compared to their non-extended counterparts (C, L, CL) (Table 1a). These differences in performance, all highly significant (Table 1b), show that library extension always results in an improvement, regardless of the BaliBase category (Table 1a). Most importantly, CLE significantly outperforms all the alternative protocols in all cases. Table 1a also shows that the performance of CLE is highly sustained. It is consistently

the best protocol and enhances the alignment accuracy with values between 3.2 and 5.7 percentage points over the next best. In contrast, the second best protocol varies over the BaliBase categories (CE in Cat. 1 and 2, CL for Cat. 3, LE in Cat. 4 and Cat. 5). This clearly indicates that the library extension greatly improves the ability of the algorithm to handle a wide range of situations.

These results show that the combination of local (Lalign) and global (ClustalW) information boosts the quality of multiple alignment. Table 1b indicates that the CLE protocol is outperformed by the second best protocol (CL) in less than 12% of the cases, as assessed over 141 BaliBase families. It should be stressed that to obtain these results, no training or fine-tuning of the parameters was performed. The parameters used for Lalign and ClustalW were the defaults, published by the authors in versions that predate BaliBase by more than a year.

3-Comparing T-Coffee with other multiple sequence alignment methods

The protocol used to assess the five methods (SAM, Dialign, ClustalW, Prrp and T-Coffee (CLE)) is identical to that described in the previous section and the results are organized in a similar layout (Table 2a and 2b).

Table 2a and b indicate that Prrp slightly outperforms ClustalW (P value: 0.02). The difference confirms the results reported by Gotoh (42) with Prrp being the second best method in four out of five categories. Categories 4 and 5 involve families of proteins with long N/C terminal extensions, where one would expect local alignment methods to perform well. Indeed, Dialign2 outperforms all the global methods (apart from T-Coffee) in category 4 and is a close competitor to Prrp for category 5. The results for

category 4 also indicate that although ClustalW is never the best method, its performance is well sustained over various BaliBase categories.

T-Coffee (CLE protocol) shows the highest average accuracy in each BaliBase category. In these five categories, all the observed differences between T-Coffee and the other methods are associated with P values lower than 0.01. For example, the average difference in accuracy between T-Coffee and its closest competitor (Prrp) is nearly 6 percentage points, associated with a highly significant P-value of 0.003 (Table 2a, Total1). The unweighted average over the five categories (Table 2a, Total2) is even more dramatic with a 9.7 percentage points better performance for T-Coffee.

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 (19) and shows that the alignments for families with less than 30 % average sequence improve the most. At this low identity level, there is a more than 2/3 chance to obtain the best alignment when using T-Coffee rather than Prrp.

4-Application to Serine/Threonine kinases

A major application of any alignment algorithm will be the delineation of motifs or domains. These elements are crucial for an in-depth understanding of sequence function. Their correct identification can be crucial for homology modeling or drug design. 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. These

proteins belong to a subfamily of protein Serine/Threonine kinases. Each sequence (apart from gcn2) is identified by its SwissProt identifier. A 3D structure is also available for each sequence. Each of the 19 sequences in the family contains two nucleotide-binding sites (NBS), marked by red letters in Figure 4. T-Coffee was the only method able to align the two motifs across all 19 sequences. The large insertion in kin1_yeast prevents the other methods from correctly aligning the second Nucleotide Binding Site (NBS): Prrp aligned 15 of the 19 motifs, ClustalW 16, Dialign 11 and Sam 15. The Gibbs sampler (24) was also run several times on the full set of sequences but could never align more than 10 of the motifs (and only when provided with an estimate of the total number of blocks in the alignment). Thanks to its use of local information, T-Coffee managed to get all the motifs aligned as in the reference alignment. Moreover, T-Coffee was the only program that correctly aligned the second NBS of the kp68_human. Sequence kp68_human is an interferon induced kinase. It is an essential component of the viral response, activated by interacting with double stranded RNA (43) and inducing an inhibition of protein synthesis.

5-Efficiency

The CPU time consumption of T-Coffee was measured. Although some steps are cubic with the number of sequences (see Methods), these do not appear to be computational bottlenecks in the context of this work. The graphs in Figure 5 indicate clearly that the apparent complexity of the program is quadratic, both relative to the average sequence lengths (Fig 5a) as to the number of sequences (Fig 5b). This complexity is the same as that of ClustalW, even if in absolute time, the overhead is slightly higher, which makes our program on average 8 times slower than ClustalW. For example, T-Coffee was tested with a data-set of 155 sequences having an even

phylogenetic spread, an average identity of 15% and an average length of 160 residues. This computation required 60 Megabytes of Memory and 4 hours of CPU-time on a PentiumII processor (330 MHz).

DISCUSSION

T-Coffee is a new progressive method for sequence alignment. It can combine signals from heterogeneous sources (sequence alignment programs, structure alignments, threading, manual alignment, motifs, specific constraints) into a unique consensus multiple sequence alignment. We showed 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 proved successful in a wide variety of cases.

While making the consensus alignment, T-Coffee does not only select the best combination of Lalign and ClustalW pairwise alignments, but it can generate improved alternative alignments. The main difference with 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" (11) 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 (44), to date no satisfying 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 short coming 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, that incorporate segments in a multiple alignment following a strict order based on their weight (25), such fortuitous segments can be a major pitfall.

Although the protocol proposed here (Lalign + ClustalW pairwise 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 and a scheme could be designed to allow combination of the main multiple sequence alignment methods using the T-Coffee protocol.

Acknowledgements

The authors wish to thank the following people: Philipp Bucher for useful discussions and advise at an early stage of the project, Willie Taylor for useful discussions, Nigel Douglas for providing us with an efficient LINUX environment, Julie Thompson for access to BaliBase, Webb Miller and Bill Pearson for allowing us to use and modify Lalign from the FASTA package.

REFERENCES

1. M. Gribskov, M. McLachlan, D. Eisenberg, *Proceedings of the National Academy of Sciences* **84**, 4355-5358 (1987).

2. D. Haussler, A. Krogh, I. S. Mian, K. Sjölander, Protein Modeling using Hidden Markov Models: Analysis of Globins, L. Hunter, Ed., Proceedings for the 26th Hawaii International Conference on Systems Sciences, Wailea, HI, U.S.A. (Los Alamitos, CA: IEEE Computer Society Press, 1993).

3. P. Bucher, K. Karplus, N. Moeri, K. Hofmann, *Comput Chem* **20**, 3-23 (1996).

4. M. O. Dayhoff, R. M. Schwarz, B. C. Orcutt, in *Atlas of Protein Sequence and Structure* M. O. Dayhoff, Ed. (National Biomedical Research Foundation, Washington, D.C., 1979), vol. 5, Suppl. 3, pp. 353-358.

5. S. Henikoff, J. G. Henikoff, *Proc. Natl. Acad. Sci.* **89**, 10915-10919 (1992).

6. S. A. Benner, M. A. Cohen, G. H. Gonnet, *J. Mol. Biol.* **229**, 1065-1082 (1993).

7. C. Sander, R. Schneider, *Proteins: Structure, Function, and Genetics* **9**, 56-68 (1991).

8. H. Carrillo, D. J. Lipman, *SIAM J. Appl. Math.* **48**, 1073-1082 (1988).

9. L. Wang, T. Jiang, *Journal of computational biology* **1**, 337-348 (1994).

10. P. Hogeweg, B. Hesper, *J. Mol. Evol.* **20**, 175-186 (1984).

11. D.-F. Feng, R. F. Doolittle, *Journal of Molecular Evolution* **25**, 351-360 (1987).

12. W. R. Taylor, *Journal of Molecular Evolution* **28**, 161-169 (1988).

13. J. Thompson, D. Higgins, T. Gibson, *Nucleic Acids Res.* **22**, 4673-4690 (1994).

14. S. B. Needleman, C. D. Wunsch, *J. Mol. Biol.* **48**, 443-53 (1970).

15. O. Gotoh, *J. Mol. Biol.* **162**, 705-708 (1982).

16. D. J. Lipman, S. F. Altschul, J. D. Kececioglu, *Proc. Natl. Acad. Sci. USA* **86**, 4412-4415 (1989).

17. J. Stoye, V. Moulton, A. W. Dress, *Comput Appl Biosci* **13**, 625-6 (1997).

18. C. Notredame, D. G. Higgins, *Nucleic Acids Res.* **24**, 1515-1524 (1996).

19. O. Gotoh, *J. Mol. Biol.* **264**, 823-838 (1996).

20. T. F. Smith, M. S. Waterman, *J. Mol. Biol.* **147**, 195-197 (1981).

21. X. Huang, W. Miller, *Adv. Appl. Math.* **12**, 337-357 (1991).

22. W. R. Pearson, D. J. Lipman, *Proceedings of the National Academy of Sciences* **85**, 2444-2448 (1988).

23. G. D. Schuler, S. F. Altschul, D. J. Lipman, *Proteins* **9**, 180-191 (1991).

24. C. E. Lawrence, et al., *Science* **262**, 208-214 (1993).

25. B. Morgenstern, A. Dress, T. Wener, *Proc. Natl. Acad. Sci. USA* **93**, 12098-12103 (1996).

26. J. D. Thompson, F. Plewniak, O. Poch, *Nucleic Acids Res.* **27**, 2682-2690 (1999).

27. J. Thompson, F. Plewniak, O. Poch, *Bioinformatics* **15**, In Press (1999).

28. C. Notredame, L. Holm, D. G. Higgins, *Bioinformatics* **14**, 407-22 (1998).

29. K. Bucka-Lassen, O. Caprani, J. Hein, *Bioinformatics* **15**, 122-30 (1999).

30. W. R. Taylor, *Protein Sci* **8**, 654-65 (1999).

31. J. Heringa, *Computers and Chemistry* **23**, 341-364 (1999).

32. J. D. Kececioglu, *Lecture Notes in Computer Science* **684**, 106-119 (1983).

33. K. Reinert, H. P. Lenhof, P. Mutzel, K. Melhorn, J. D. Kececioglu, *Recomb97* , 241-249 (1997).

34. A. F. Neuwald, J. S. Liu, D. J. Lipman, C. E. Lawrence, *Nucleic Acids Res* **25**, 1665-77 (1997).

35. B. Morgenstern, *Bioinformatics* **15**, 211-8 (1999).

36. N. Saitou, M. Nei, *Mol. Biol. Evol.* **4**, 406-425 (1987).

37. C. A. Orengo, W. R. Taylor, *Methods in Enzymology* **266**, 617-635 (1996).

38. L. Holm, C. Sander, *Trends in Biochemical Sciences* **20**, 478-480 (1995).

39. S. R. Eddy, Multiple alignment using hidden Markov models, C. Rawlings, et al., Eds., Third International converence on Intelligent Systems for Molecular Biology (ISMB), Cambridge, England (Menlo Park, CA: AAAI Press, 1995).

40. R. Hughey, A. Krogh, *Computer Applications in Biological Science* **12**, 95-107 (1996).

41. J. Stoye, *Gene* **211**, GC45-56 (1998).

42. O. Gotoh, *Comput Appl Biosci* **10**, 379-87 (1994).

43. E. Meurs, et al., *Cell* **62**, 379-390 (1990).

44. M. A. McClure, T. K. Vasi, W. M. Fitch, *Molecular Biology Evolution* **11**, 571-592 (1994).

Table 1

a)Evaluation of different protocols with BaliBase.

Method				BaliBase Reference (Number of families)					
Name	Protocol			Cat 1 (82)	Cat 2 (23)	Cat 3 (12)	Cat 4 (13)	Cat 5 (11)	Tot (141)
C	ClustalW pw			70.6	26.7	43.0	56.0	60.0	58.9
CE	ClustalW pw		**extend**	77.1	33.6	47.6	64.8	75.9	66.3
L		Lalign pw		65.4	12.1	22.8	53.9	66.0	52.0
LE		Lalign pw	**extend**	72.6	25.6	47.2	77.5	85.5	64.2
CL	ClustalW pw	Lalign pw		76.2	32.0	48.3	76.2	74.6	66.5
CLE	ClustalW pw	Lalign pw	**extend**	**80.7**	**37.3**	**52.9**	**83.2**	**88.7**	**72.1**

b) Relative performances of the different protocols.

	REFERENCE METHODS											
Versus	**C (%)**		**CE(%)**		**L(%)**		**LE(%)**		**CL(%)**		**CLE(%)**	
	B	*W*	*B*	*W*	*B*	*W*	*B*	*W*	*B*	*W*	*B*	*W*
C	/		68.8	**12.8	36.2	**57.4	60.3	**28.4	63.8	**15.6	**79.4**	** 7.8
CE	12.8	**68.8			22.7	**73.0	38.3	+53.2	37.6	48.9	**62.4**	**17.7
L	57.4	**36.2	22.7	**73.0			66.7	**12.1	77.3	**12.8	**87.2**	** 7.8
LE	28.4	**60.3	38.3	+53.2	12.1	**66.7			41.8	41.1	**68.1**	**16.3
CL	15.6	**63.8	48.9	37.6	12.8	**77.3	41.1	41.8			**67.4**	**12.1
CLE	7.8	**79.4	62.4	**17.7	7.8	**87.2	16.3	**68.0	**12.1	67.4		

a) *Protocol* indicates the way the library was created. *ClustalW pw* indicates computation of all the ClustalW pairwise alignments with default parameters, likewise with *Lalign pw. Extend* indicates that the library was extended before the progressive alignment. *Cat 1* is the first category of families in BaliBase. The average accuracy for each protocol is given in %. For each category, the best protocol is indicated in Bold and underlined. Tot indicates the average value obtained on the 141 families.
b) C, CE… refers to the protocols in the previous table. B (better) indicates the proportion of families (141 in total) for which the reference method outperformed the horizontal one. W (worse) indicates the proportion of methods for which the column method was outperformed by the one in the row. W and B do not add up to hundred because for some families the compared methods yielded the same score. The '+' and '*' annotations indicate statistical significance of the observed differences between the two methods according to the Wilcoxon test. No annotation: P>0.1, +: P<0.1, *P<0.01, **P<0.001. Bold figures indicate the best performances.

Table 2

a) Evaluation of five packages with BaliBase

Method Name	BaliBase Reference (Number of families)						
	Cat 1 (82)	Cat 2 (23)	Cat 3 (12)	Cat 4 (13)	Cat 5 (11)	Total 1 (141)	Total 2 (141)
Sam	46.8	20.0	13.9	43.9	42.7	39.8	33.4
Dialign	71.0	25.2	35.1	74.7	80.4	61.5	57.3
ClustalW	78.5	32.2	42.5	65.7	74.3	66.4	58.6
Prrp	78.6	32.5	50.2	51.1	82.7	66.4	59.0
T-Coffee (CLE)	**80.7**	**37.3**	**52.9**	**83.2**	**88.7**	**72.1**	**68.7**

b) Relative performances of the five packaged

Versus	REFERENCE METHODS									
	Sam		Dialign		ClustalW		Prrp		T-Coffee (CLE)	
	B	*W*	*B*	*W*	*B*	*W*	*B*	*W*	*B*	*W*
Sam	/		72.3	**14.2	83.7	**3.5	85.1	** 7.8	**92.2**	** 2.1
Dialign	14.2	**72.3	/		57.4	**31.9	62.4	**24.8	**76.6**	**11.3
ClustalW	3.5	**83.7	31.9	**57.4	/		46.8	+34.0	**58.9**	**26.2
Prrp	7.8	**85.1	24.8	**62.4	34.0	+46.8	/		**49.6**	*36.9
T-Coffee(CLE)	2.1	**92.2	11.3	**76.6	26.2	**58.9	36.9	**49.6	/	

a) *Method Name* indicates the name of the method evaluated. T-Coffee is the protocol CLE in Table 1a and b. *Total 1* is the average accuracy as measured on the 141 families. *Total 2* is the average accuracy of the five BaliBase categories (unweighted). The table layout is similar to Table 1a.
b) The table layout and the annotation for statistical significance are described in the table 1b legend.

FIGURE LEGENDS

Figure 1: Layout of the T-Coffee strategy.

This figures outlines the steps required to compute a multiple sequence alignment using the T-Coffee method. Square Blocks designate procedures while rounded blocks indicate data structures (See text for details).

Figure 2 The Library Extension

a)*Primary Library*: Direct alignment of sequence A and B using ClustalW (red lines) and Lalign (black lines). The only constraints reported are those involving residue 6 within sequence A. The red lines indicate constraints found in a ClustalW pairwise alignment, the black line indicate constraints found with Lalign. The numbers in parenthesis indicate the value of the weight associated with each constraint.

b)*Library Extension*: The alignment of A and B through sequence C is carried out and associated constraints are added to the library.

c)*Extended Library*: The extended library is made of all the weighted constraints found between A and B. The thickness of the lines indicate the value of the weight.

Figure 3 Comparison between T-Coffee and Prrp.

For each family within BaliBase, the average level of pairwise identity was measured on the reference alignment. Alignment accuracy was assessed for T-Coffee and Prrp. The latter two values were subtracted (%T-Coffee accuracy -%Prrp accuracy) and plotted versus the average BaliBase identity for the family. Dots in the white area indicate families where T-Coffee is outperforming Prrp and inversely for the grey zone. Families have been divided in two sets: below 30 % identity (85 families) and

above (56 families). The percentages given in the corners of the plot indicate the fraction of families for which T-Coffee outperforms Prrp (top) and vice-versa (bottom). These percentages do not add up to hundred as for some families, the same accuracy was obtained with each method (*e.g.* for alignment having less than 30% id, T-Coffee outperforms Prrp in 56% of the cases, Prrp outperforms T-Coffee in 32 % while both methods draw in 12 % of the cases).

Figure 4 Example of a T-Coffee alignment.

This N terminal alignment of 19 kinases shows two boxes containing the Nucleotide Binding Site. The residues in capital are annotated as core regions in BaliBase. The core residues in red are correctly aligned as respect to the BaliBase reference. This family belongs to BaliBase category 5 (long insertion).

Figure 5 Measure of the T-Coffee complexity

a) *Effect of the alignment length.* CPU time required for computing four-sequence multiple alignments is measured (32 families) and plotted versus the length of these multiple alignments.

b) *Effect of the number of sequences.* Measured CPU time versus the number of sequences contained in each alignment. All the alignments have an approximate length of 350 residues (between 295 and 382).