2682–2690 *Nucleic Acids Research, 1999, Vol. 27, No. 13*

A comprehensive comparison of multiple sequence alignment programs

Julie D. Thompson, Frédéric Plewniak and Olivier Poch*

Laboratoire de Biologie Structurale, Institut de Génétique et de Biologie Moléculaire et Cellulaire, (CNRS/INSERM/ULP), BP 163, 67404 Illkirch Cedex, France

Received March 18, 1999; Revised and Accepted May 14, 1999

ABSTRACT

In recent years improvements to existing programs and the introduction of new iterative algorithms have changed the state-of-the-art in protein sequence alignment. This paper presents the first systematic study of the most commonly used alignment programs using BAliBASE benchmark alignments as test cases. Even below the 'twilight zone' at 10–20% residue identity, the best programs were capable of correctly aligning on average 47% of the residues. We show that iterative algorithms often offer improved alignment accuracy though at the expense of computation time. A notable exception was the effect of introducing a single divergent sequence into a set of closely related sequences, causing the iteration to diverge away from the best alignment. Global alignment programs generally performed better than local methods, except in the presence of large N/C-terminal extensions and internal insertions. In these cases, a local algorithm was more successful in identifying the most conserved motifs. This study enables us to propose appropriate alignment strategies, depending on the nature of a particular set of sequences. The employment of more than one program based on different alignment techniques should significantly improve the quality of automatic protein sequence alignment methods. The results also indicate guidelines for improvement of alignment algorithms.

INTRODUCTION

The multiple alignment of protein sequences has become an essential tool in molecular biology. It has traditionally been used to find characteristic motifs and conserved regions in protein families, in the determination of evolutionary linkage and in the improved prediction of secondary and tertiary structure. With the rapid increase in the number of protein sequences, notably from the genome sequencing projects, automatic methods of searching protein databases for homologous sequences (1,2), followed by the multiple alignment of the top scoring hits (3–6) are becoming standard practice. These automatic systems frequently involve the alignment of large numbers of sequences, of very divergent sequences and of multi-domain proteins often with large N/C-terminal extensions or internal insertions. Moreover, with the available sequenced genomes, the alignment of single divergent sequences (typically of eukaryotic origin) with a large closely related group (typically of prokaryotic origin) is now commonplace. The development of accurate, reliable multiple alignment programs capable of handling these divergent sets of data is therefore of major importance. Although a dynamic programming algorithm (7) exists which guarantees a mathematically optimal alignment, the method is limited to a small number of short sequences since the computing power required for larger alignments becomes too prohibitive. To overcome this problem, various heuristic approaches have been developed leading to a huge quantity of programs using fundamentally different strategies (progressive, iterative, mixed, etc.) based on very different algorithms. Figure 1 shows some of the most commonly used programs today, together with examples of the main algorithms that have been developed recently. Traditionally the most popular approach has been the progressive alignment method (8). A multiple alignment is built up gradually by aligning the closest sequences first and successively adding in the more distant ones. A number of alignment programs based on this method exist, for example MULTALIGN (9), MULTAL (10), PILEUP (Wisconsin Package v.8; Genetics Computer Group, Madison, WI) and CLUSTALX (11), which provides a graphical interface for CLUSTALW (12). They use a global alignment algorithm (13) to construct an alignment of the entire length of the sequences. They differ mainly in the method used to determine the order of alignment of the sequences. MULTAL uses a sequential branching method to align the two closest sequences first and then subsequently align the next closest sequence to those already aligned. MULTALIGN and PILEUP construct a guide tree using the UPGMA method (14). A consensus method is then used to align larger and larger groups of sequences according to the branching order of the tree. CLUSTALX uses the alternative Neighbour-Joining algorithm (15) to construct a guide tree, incorporating in addition sequence weighting, position-specific gap penalties and a choice of residue comparison matrix depending on the degree of identity of the sequences. In contrast to the above global methods, PIMA (16) uses a local dynamic programming algorithm (17) to align only the most conserved motifs. PIMA offers two alignments by default using maximum linkage and sequential branching algorithms

*To whom correspondence should be addressed. Tel: +33 3 88 65 32 00; Fax: +33 3 88 65 32 01; Email: poch@igbmc.u-strasbg.fr

to decide the order of alignment, which we will refer to as MLPIMA and SBPIMA, respectively.

In addition, numerous new alignment algorithms have recently been developed which offer fresh approaches to the multiple alignment problem. A common point of interest has been the application of iterative strategies to refine and improve the initial alignment. A local alignment approach is implemented in the DIALIGN program (18) to construct multiple alignments based on segment-to-segment comparisons rather than the residue-to-residue comparisons used previously. The segments are incorporated into a multiple alignment using an iterative procedure. The PRRP program (19) optimises a progressive, global alignment by iteratively dividing the sequences into two groups, which are subsequently realigned using a global group-to-group alignment algorithm. SAGA (20) uses a genetic algorithm to select from an evolving population the alignment which optimises the COFFEE Objective Function (OF) (21). The OF is a measure of the consistency between the multiple alignment and a library of CLUSTALW pairwise alignments. Hidden Markov models (HMMs) have also been used as statistical models of the primary structure consensus of a sequence family (22,23). The program HMMT (24) uses a simulated annealing method to maximise the probability that an HMM represents the sequences to be aligned.

In spite of this wide variety of alignment programs, there are few comparisons available of their relative performance and reliability. Twelve different global and local progressive alignment programs were compared (25) using alignments of four different protein domains as test cases. In general, the global methods performed better than local methods in the tests, but the performance of all the programs was affected by the number of sequences, the degree of identity of the sequences and the number of insertions/deletions in the alignment. Seven multiple alignment Web servers covering various global and local methods have been compared (26) to evaluate their ability to identify the reliable regions in an alignment. However, no comprehensive study and comparison of the numerous new alignment algorithms exists. The lack of a standard set of reference alignments has meant that existing programs could not be benchmarked and the increase in performance realised by the new iterative alignment methods could not be accurately measured. A benchmark alignment database called BAliBASE (27) has recently been developed specifically for this purpose. The 142 validated test alignments of real proteins based on three-dimensional superimpositions are organised into reference sets which represent some of the most common problems currently encountered when aligning real families of proteins. Core blocks in each alignment define those regions that can be reliably aligned. BAliBASE is available on the World Wide Web at http://www-igbmc.u-strasbg.fr/BioInfo/BAliBASE

In this paper, we present a systematic analysis and comparison of the main alignment programs currently in use (Fig. 1), using the BAliBASE reference alignments as test cases. A comparison of different scoring methods has highlighted the importance of the non-superimposable regions in the evaluation of a program. We show that the 'twilight zone' still exists as a real barrier for all the programs in this study, but that some alignment is possible below the twilight zone. The strong and weak points of the programs are highlighted, in particular the effect on alignment accuracy of different criteria such as the sequence length, the degree of identity of the sequences, their re-partition into subfamilies and the presence of large N/C-terminal extensions and internal insertions. This has enabled us to define possible strategies for improving the programs and guidelines for optimising alignments.

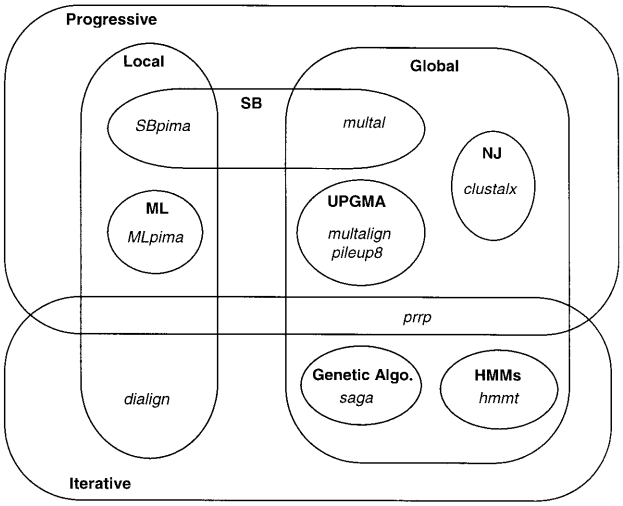

Figure 1. Schematic showing the relation between the different alignment programs and algorithms.

MATERIALS AND METHODS

All the programs were installed on a DEC Alpha 6100 computer running OSF Unix and each program was tested using default parameters (with the exception of the PRRP-b option, which indicates that the input sequences have not been pre-aligned). We assume that the parameters chosen by the authors have been selected to give a near optimal alignment under normal conditions and, therefore, for the purposes of this study no optimisation of parameters such as residue weight matrix and gap penalties was performed. The test alignments produced here provide a reference which will be used as a basis for further study of optimum parameters (work in progress).

Reference alignments

In order to evaluate and compare the 10 alignment programs selected for this study, we needed objective criteria to assess the quality of an alignment. The BAliBASE benchmark alignment database contains 142 reference alignments, divided into five hierarchical reference sets each containing at least 12 representative alignments (Table 1). The alignments of sequences sharing the same three-dimensional fold have been validated to ensure the alignment of functional and other conserved residues. Core blocks are defined for each alignment as being the regions that can be reliably aligned. The core blocks (representing 58% of the residues in the alignments) specifically exclude ambiguous or non-superimposable three-dimensional regions such as distinct secondary structures, unrelated secondary structure borders or structurally unreliable loop regions.

Table 1. BAliBASE reference sets, showing the number of alignments in each set

Reference	Short (<100 residues)	Medium (200–300 residues)	Long (>400 residues)
Reference 1: equidistant sequences of similar length			
V1 (<25% identity)	7	8	8
V2 (20–40% identity)	10	9	10
V3 (>35% identity)	10	10	8
Reference 2: family versus orphans	9	8	7
Reference 3: equidistant divergent families	5	3	5
Reference 4: N/C-terminal extensions	12		
Reference 5: insertions	12		

Reference 1 alignments consist of a small number of equidistant sequences of similar length, i.e. the per cent residue identity (% ID) between any two sequences is within a specified range and no large extensions or insertions have been introduced.

Reference 2 contains alignments of a family (closely related sequences with >25% ID), plus up to three 'orphan' sequences (distant members of the family with <20% ID, sharing a common fold). It is designed to evaluate program accuracy according to two criteria: (i) the stability of the family alignment when orphans are introduced into the sequence set; (ii) the quality of the alignment of the orphan sequences. The program MULTAL has been removed from this test since it frequently excludes the divergent orphans as unrelated or unalignable sequences.

Reference 3 demonstrates the ability of the programs to correctly align equidistant divergent families into a single alignment. The reference alignments consist of up to four families, with <25% ID between any two sequences from different families. MULTAL is not included in reference 3 (see explanation in reference 2).

References 4 and 5 contain sequences with large N/C-terminal extensions or internal insertions, respectively. In order to evaluate a program's ability to identify the presence of the insertions, the core blocks in BAliBASE define only the most conserved motifs flanking the extension/insertion. These tests are not designed to judge the overall quality of an alignment. MULTAL is not included in these tests (see explanation in reference 2). HMMT is also excluded because many of the alignments contain only a small number of sequences.

Alignment scores

To assess the performance of the programs in this study, we calculate two different scores which estimate the quality of an alignment compared to the BAliBASE reference. The sum-of-pairs score (SPS) is calculated such that the score increases with the number of sequences correctly aligned. It is used to determine the extent to which the programs succeed in aligning some, if not all, of the sequences in an alignment. The column score (CS) is a binary score which tests the ability of the programs to align ALL of the sequences correctly.

Sum-of-pairs score. Suppose we have a test alignment of N sequences consisting of M columns. We can designate the ith column in the alignment by A_{i1}, A_{i2}, …, A_{iN}. For each pair of residues A_{ij} and A_{ik} we define p_{ijk} such that $p_{ijk} = 1$ if residues A_{ij} and A_{ik} are aligned with each other in the reference alignment, otherwise $p_{ijk} = 0$. The score S_i for the ith column is defined as:

$$S_i = \Sigma_{j=1, j\neq k}^{N} \Sigma_{k=1}^{N} p_{ijk}$$

The SPS for the alignment is then:

$$SPS = \Sigma_{i=1}^{M} S_i / \Sigma_{i=1}^{Mr} S_{ri}$$

where Mr is the number of columns in the reference alignment and S_{ri} is the score S_i for the ith column in the reference alignment.

Column score. For the ith column in the alignment described above, the score $C_i = 1$ if all the residues in the column are aligned in the reference alignment, otherwise, $C_i = 0$.
The CS for the alignment is then:

$$CS = \Sigma_{i=1}^{M} C_i / M$$

For each reference test we have selected the most suitable scoring function according to the nature of the test and the particular question posed. The two scoring systems have been implemented in the program BaliScore, which takes as input a reference alignment and a test alignment in MSF format, plus an optional BAliBASE annotation file describing the core blocks in the reference alignment. The output includes the two scores described above, plus an optional representation of the scores for each column in the test alignment. BAliScore is available by ftp from ftp-igbmc.u-strasbg.fr/pub/BAliBASE/BAliScore

Statistical methods

In each reference, BAliBASE provides a number of representative alignments that were used as a sample in statistical analyses. For each reference alignment we calculate a score estimating the accuracy of the alignment produced by every program. Since the distribution of scores is expected to be neither normal nor symmetric, we use the median as a measure of location and the interquartile range as a measure of dispersion. The first and third quartiles give an idea of the shape of the distribution.

Friedman tests (28) were used to assess whether or not there is a systematic pattern in the way programs are ranked by score for every alignment, i.e. whether or not some programs

significantly tend to perform better than others across reference alignments.

Wilcoxon signed rank tests (29) were used to determine whether a change in the conditions of an alignment, such as the addition of orphans (reference 2) or an increase in the number of family members (reference 3), leads to a significant difference between paired scores. The scores for HMMT are not included in the Wilcoxon signed rank tests, because different alignments may be obtained for the same input sequences each time the program is executed. Therefore, the difference in the scores obtained under different alignment conditions cannot be reliably compared.

RESULTS

Reference 1: a small number of approximately equidistant sequences

This test is designed to study the effect of sequence length and % ID on alignment program performance and provides a basis for the remaining tests. The importance of the ambiguous or non-superimposable regions in the evaluation of alignment program performance has been studied by comparing alignment scores based only on the core blocks defined in BAliBASE with scores obtained over the full-length sequences. The ambiguous regions represent 42% of the residues in BAliBASE and account for a mean 32, 22 and 11% of the full-length scores calculated in categories V1, V2 and V3, respectively. Obviously, some discrepencies in the program evaluation may arise using either of the scoring methods. Here we will present the results of this study using the core block scores, unless a comparison of the two scores sheds light on interesting results.

How do percent identity and sequence length affect program performance? Figure 2a shows the median core block scores obtained in the nine variability/length categories in reference 1. A decrease in accuracy of the alignments with decreasing identity is clearly demonstrated, with the greatest loss occurring in category V1 (<25% ID), which corresponds to the so-called 'twilight zone' of evolutionary relatedness. Nevertheless, some alignment is still possible below the twilight zone. The best alignment in V1 was achieved by PRRP, with 72% of the total residues correctly aligned. The highest scoring programs, PRRP, CLUSTALX and SAGA, correctly align on average 61% of the residues (or 42% of the columns) in the core blocks and 47% of the total residues (or 26% of the total columns) in V1. In contrast, between 20 and 40% identity, 92% of the residues (or 87% of the columns) in the core blocks and 82% (or 72% of the total columns) of the total residues are successfully aligned by these three programs. Figure 2b shows a plot of the median core block scores obtained by each of the 10 programs in identity ranges V1, V2 and V3. It can be seen that loss of accuracy with decreasing sequence identity is exhibited by all the programs in this study. The greatest difference in program scores is always observed in category V1. According to a Friedman test used to compare the performance of the alignment programs (Fig. 2c), PRRP ranks significantly higher ($\alpha = 0.05$) than the other programs, CLUSTALX and SAGA rank second highest, with the global alignment programs generally performing better than the local methods.

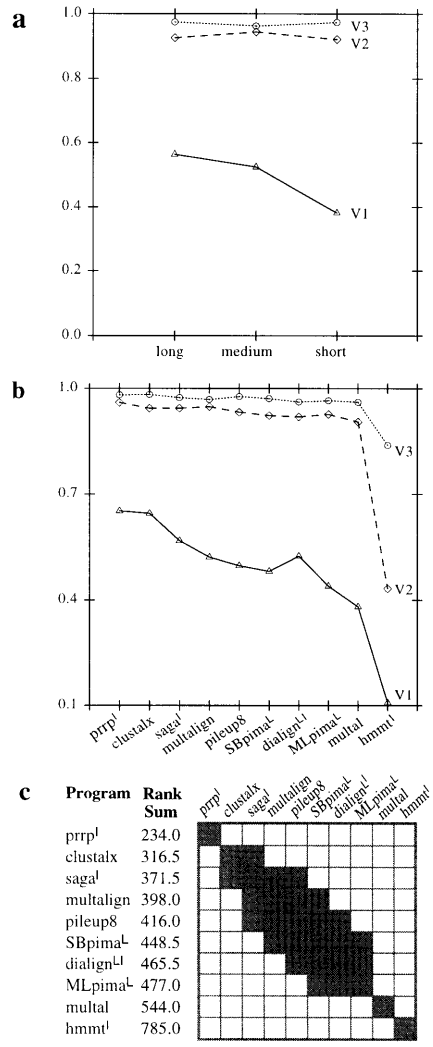

Figure 2. (**a**) SPS for reference 1, showing the median score in each category. (**b**) Median SPS for the programs in reference 1, categories V1, V2 and V3. Programs are displayed in the order of the Friedman test, with the highest scoring program on the left. (**c**) Results of the Friedman rank test to compare the performance of the programs in reference 1 (S = 9, N = 81, test statistic = 106.9). For each test alignment, the programs are assigned a rank between 1 and 10 (with 1 indicating the highest scoring program). The ranks are then summed over all alignments. Thus, a lower rank sum indicates that a program tends to achieve higher scores. The programs are listed in rank sum order. The grey boxes indicate that the two corresponding programs cannot be differentiated using the Friedman test ($\alpha = 5\%$). ^{L}, local alignment program; ^{I}, iterative alignment program.

Figure 3a shows the median core block scores for the length categories short, medium and long in V1. It is important to note that for long sequences the local programs achieve a similar quality of alignment to the global ones, with the exception only of PRRP, which ranks significantly higher ($\alpha = 0.05$) than the other programs in a Friedman test. In general, the core blocks in medium and long sequences are aligned better than in short ones by all the programs except CLUSTALX. However, an analysis of the full-length scores (Fig. 3b) reveals: (i) a general decrease in full-length scores compared to core block scores;

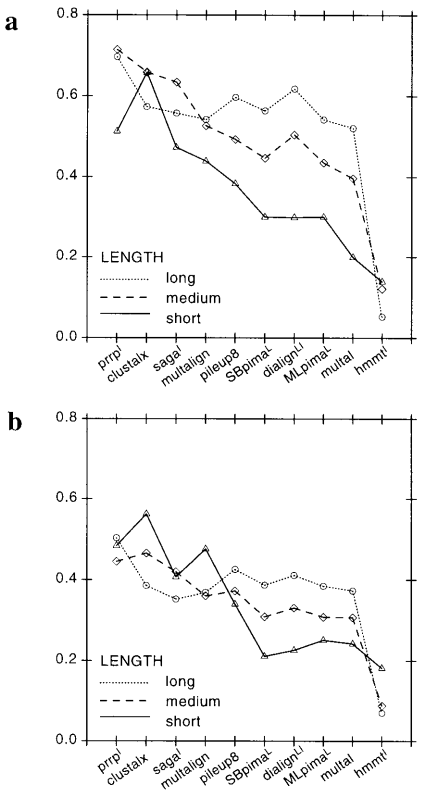

Figure 3. Median SPS for the programs in reference 1 for length categories short, medium and long. Only scores in category V1 are shown. (**a**) Scores based on the core blocks only; (**b**) scores based on the full-length alignment. [L], local alignment program; [I], iterative alignment program.

(ii) an inversion of order observed above for global programs with the scores now decreasing with increasing sequence length; (iii) in contrast, the scores for the local programs maintain the same order as before, with the scores increasing with increasing sequence length.

Further investigation of the effect of sequence length. The fact that all the programs tend to align the core blocks in longer sequences better than in short sequences is surprising. This result may be due to the sequence length itself or to a difference in the nature of the core blocks in each length category. In order to investigate the cause, we decided to compare the medium length alignments in category V1 with a new population of short sequences artificially created by dividing the medium length sequences into two half-length alignments, ensuring that the division was not made within a core block. In this case, a Wilcoxon signed rank test showed that the core blocks in the short sequences are aligned better than in the medium length sequences ($P < 10^{-3}$). We deduce that the observed reduction in accuracy for short sequences in reference 1 is not due simply to the length of the sequences.

We conclude that this effect is due to the nature and re-partition of the core blocks and the length of the gaps found in the BAliBASE alignments. Although the overall % ID are the same for the short, medium and long sequences in V1, the core blocks in the short sequences are less well conserved than those in the longer sequences (Table 2). We verified that the same re-partition of conserved motifs is observed in categories V2 and V3 (data available on the BAliBASE WWW server). In fact, both the conserved motifs and the insertions/deletions are shorter in the short sequences.

Reference 2: a related family with divergent, orphan sequences

Here we test not only the ability of the programs to align divergent 'orphan' sequences (10–20% ID with the family and between orphans) with a family of highly related (>25% ID) sequences, but also the degree to which the alignment of the family produced by the program is disrupted by the introduction of the orphans. As both of these questions may depend effectively on the size of the family, the tests were repeated with small families of four sequences and larger families of 14–22 sequences.

Effect of the orphan on the family alignment. The families were first aligned with no orphans present to provide a reference for comparison. Alignments were then constructed with one, two and three orphans (all with 10–20% ID) to investigate whether the original family alignment is disorganised by the introduction of the orphans. In each case, the SPS was calculated for the program alignment of the family compared with the BAliBASE reference alignment. Surprisingly, a Wilcoxon signed rank test indicates no significant reduction in the scores for the family alignments ($P = 0.686$ and 0.713 for small and large families, respectively). Nevertheless, a small number of cases were observed in which the presence of the orphans resulted in a loss of alignment quality of up to 6.9% for the large families and 23.6% for the small ones.

Alignment of the orphans. Figure 4 shows the SPS for the alignment of a single orphan against a closely related family. The global alignment programs again perform better than the local ones in this test. However, CLUSTALX and SAGA now rank above PRRP. A Wilcoxon signed rank test (for all programs except HMMT) to compare the alignment of the orphan against small and large families (four and 14–22 sequences,

Table 2. Statistics of core blocks in reference 1, category V1

	Mean residue % ID	Mean residues in core blocks (%)	Mean residue % ID in core blocks	Mean longest pairwise motif length	Mean longest pairwise insertion length
Short	16	40	18	3.4	11
Medium	16	31	25	5.5	19
Long	18	37	26	6.0	31

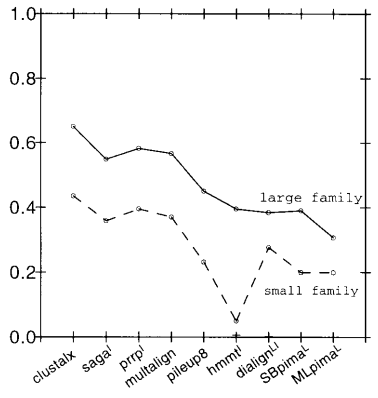

Figure 4. Median SPS for aligning one orphan sequence with a family of closely related sequences in reference 2. Large families consist of 14–22 sequences, small ones contain four sequences. Programs are shown in the order of the Friedman test for large families. ^{L}, local alignment program; ^{I}, iterative alignment program.

respectively) indicates a significant improvement (P = 0.057) in the alignment when the family consists of more sequences.

The ability of all the programs to correctly align an orphan sequence is also affected by the presence of other orphans in the sequence set (data not shown). The exact correlation is not clear, but depending on the relatedness between the orphans and of the orphans to the family, the alignment can either improve or deteriorate.

Reference 3: families of related sequences

This test is designed to assess the ability of the programs to correctly align approximately equidistant divergent families (<20% ID) composed of highly related sequences (>25% ID) into a single multiple alignment. We can compare the results of this test with the corresponding alignments in reference 1. In the latter, we aligned small numbers of equidistant divergent sequences, here we align small numbers of equidistant divergent families of sequences. The CS is used in these tests, as it is a better estimator of the quality of the alignment between the families. The SPS used previously are more influenced by the quality of the alignment within the families.

Alignment of families of sequences. Figure 5a shows the CS for the programs in the order obtained from the Friedman test. It can be seen that the iterative strategies of PRRP and SAGA perform better in this test than the traditional progressive alignment methods. CLUSTALX performs better than the other progressive methods, with the global methods generally ranking higher than the local methods.

Comparison with alignment of individual sequences. In order to compare the alignment of equidistant families of sequences with the alignment of individual equidistant sequences, we constructed a new set of reference 1 type alignments by selecting one sequence from each family in the reference 3 alignments. A comparison of the scores for the alignment of the families with the scores for the individual sequences shows that the families are aligned more successfully by all the programs except MLPIMA and SBPIMA (Fig. 5b).

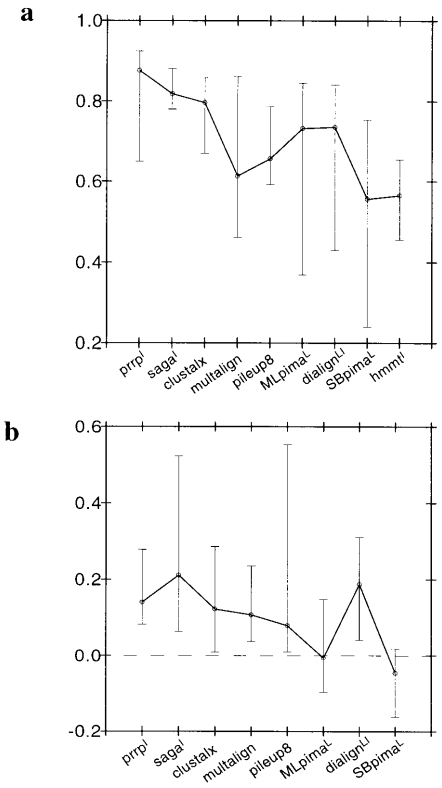

Figure 5. (**a**) Median CS for aligning subgroups of sequences in reference 3. The error bars indicate the interquartile range. Programs are shown in the order of the Friedman test. (**b**) Comparison of reference 3 with reference 1. The difference in the CS ($Score_{ref3} - Score_{ref1}$) where $Score_{refi}$ is the score for an alignment in reference i. ^{L}, local alignment program; ^{I}, iterative alignment program.

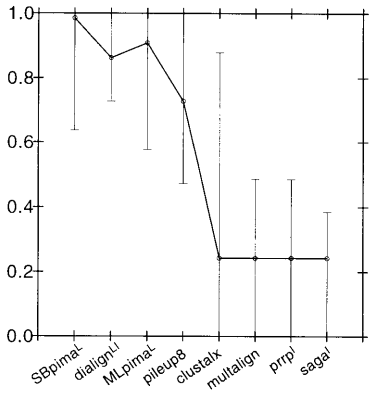

Figure 6. Median CS for N/C-terminal extensions. The error bars indicate the interquartile range. Programs are shown in the order of the Friedman test. ^{L}, local alignment program; ^{I}, iterative alignment program.

Reference 4: N/C-terminal extensions

All the previous tests have involved sequences of similar lengths. We now introduce sequences with large N/C-terminal extensions to investigate whether the programs are capable of aligning the core blocks flanking the extensions. No large

internal insertions are introduced at this stage. Figure 6 shows the median CS for the programs. An inversion of the previous ranking of the alignment programs is observed, with the three programs which implement a local alignment strategy now out-performing the global methods. PILEUP8 is the only program based on a global alignment method which ranks with the local methods in the Friedman test ($\alpha = 0.05$). The iterative strategies of PRRP and SAGA are not successful in this test. In fact, the global methods often fail to locate the flanking core blocks, resulting in a total misalignment of those sequences with large extensions.

Reference 5: internal insertions

This test also contains sequences of unequal length, but in contrast to reference 4, the insertions are internal to the homologous domains and not at the N/C-terminus. We use only the most conserved core blocks flanking the insertions which are defined in BAliBASE. The median CS for the programs are shown in Figure 7. Although the local program DIALIGN remains one of the top ranking programs in this test as in reference 4, MLPIMA and SBPIMA are less successful and in fact rank lower than the global alignment programs.

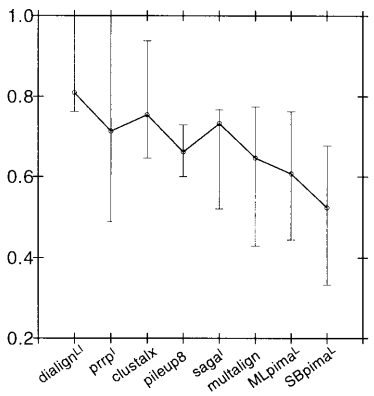

Figure 7. Median CS for internal insertions. The error bars indicate the interquartile range. Programs are shown in the order of the Friedman test. ^{L}, local alignment program; ^{I}, iterative alignment program.

DISCUSSION

One of the objectives of this study was to establish an objective benchmarking system that can be used to compare, evaluate and improve multiple alignment programs. The BAliBASE alignments provided real test cases containing proteins or modules whose three-dimensional structures have been determined. The alignments are validated to ensure the correct alignment of catalytic and other conserved residues and core blocks are annotated which include only the regions that can be reliably aligned. In the light of the results, it seems clear that it is indispensable to be able to compare alignment scores based on the entire sequences and scores based only on the validated core blocks. Indeed, the ambiguous regions excluded from the BAliBASE core blocks represent on average 32% of the full-length SPS for alignments of very divergent sequences and 11% of the score even for highly related sequences. Even though full-length scores may be informative, the identification and use of the core blocks in scoring is indispensable for reliable evaluation of the programs, notably when the difference between the scores is weak.

Evidence of distinct patterns of residue conservation

Surprisingly, all the programs align the core blocks in short sequences less well than in longer ones. The comparison of the full-length and core block scores has enabled us to suggest that this result may be due to different patterns of conservation in short versus longer sequences. Although the short, medium and long sequences in BAliBASE share similar mean residue % ID, in the short sequences the conserved residues are scattered into shorter motifs and often single residue motifs are observed (Table 2). Although the reasons for these differences are not clear, the fact that we have used sequences corresponding to fully folded entities (proteins or modules) suggests that the differences observed here reflect real structural differences. This is further supported by the fact that, in a population of short sequences artificially created by dividing real proteins of medium length into two fragments, the short sequences are actually aligned more successfully. These results may suggest that the number and re-partition of the residues required to maintain a common fold for small proteins or modules is not the same as that required to maintain a larger fold (work in progress).

The ‘twilight zone’

We have shown that all the programs in this study are capable of correctly aligning on average 80% of the residues in an alignment for sequences with >20% ID. Comparisons between the programs at this level of sequence identity are indecisive. At 10–20% ID (reference 1, V1), an important loss of accuracy occurs, with the best programs correctly aligning on average only 47% of the residues. The ‘twilight zone’ (30) clearly constitutes a real barrier for all the programs in this study. Below the twilight zone, the alignments produced by the programs are often unreliable with very large dispersions of the scores. In fact, for long sequences of more than 400 residues in V1, the global and local programs can no longer be distinguished. It is clear that efforts to improve the quality of alignment programs should now be concentrated on the alignment of sequences below 20–25% residue identity. Nevertheless, it should be said that even below the twilight zone some progress has been made, notably by PRRP, which is capable of aligning 27–72% of the total residues correctly.

Non-iterative versus iterative methods

An evaluation of the improvement introduced by the new iterative methods is inconclusive. Four programs are distinguished as being the most successful under the distinct alignment conditions tested, PRRP, SAGA, CLUSTALX and DIALIGN, and it should be noted that three of these programs use iterative strategies to refine the alignment. The remaining program, CLUSTALX, has clearly improved on the traditional progressive alignment programs, although for long sequences the default parameters may not be optimal. We have shown that the new iterative algorithms often offer improved alignment accuracy, successfully ‘learning’ and improving an alignment if enough information is included in the sequence dataset, as highlighted in the test cases of equidistant families

of sequences. However, the iteration process may sometimes be unstable in the presence of a bias in the sequence set, such as a single orphan sequence, the iteration may diverge away from the correct alignment. Of the local programs, DIALIGN, which iteratively uses a local segment alignment algorithm, is the most successful. In contrast, the iteration implemented in HMMT does not perform as well as the other global alignment methods in the tests which include up to 25 sequences. Even in tests with 100 sequences, HMMT does not rank above the global programs (data not shown). The application of an iterative strategy clearly improves the accuracy of alignment under certain conditions. Nevertheless, it is obvious that the choice of fundamental algorithm implemented at each iteration is equally, if not more, important.

A big disadvantage of the current iterative techniques is the heavy time penalty incurred. As an example, for 89 histone sequences consisting of 66–92 residues, the CPU time required for the alignment is 161 s for CLUSTALX, 13 649 s for DIALIGN and 13 209 s for PRRP. The question is, is the time penalty justified by the increase in alignment quality achieved by the iterative strategies? It should be possible to develop a more efficient strategy for the refinement of the progressive alignments which can obtain a similar quality but more rapidly.

Global versus local methods

In general, two basic classes of alignment program have been developed. Global alignment programs attempt to align the sequences over their whole length, whereas local programs search only for the most conserved motifs. The most effective alignment algorithm depends on the nature of the sequences to be aligned. Global algorithms produce the most accurate and reliable alignments in the tests involving equidistant sequences, divergent families of sequences and the alignment of orphan sequences with a family. This result confirms the findings of McClure *et al*. (25). However, we have shown that in the presence of large N/C-terminal extensions and internal insertions, DIALIGN, which implements a local, gap-free segment alignment, is the most successful program at locating the highly conserved flanking core blocks. However, the total alignment outside the most conserved motifs remains unreliable, analogous to the results in reference 1. Global programs which tend to favour a collinear alignment of the entire lengths of the sequences are less successful, often producing a total misalignment of the sequences.

An improved alignment strategy

The results of these tests suggest possible ways to improve program accuracy for families and divergent sequences. The alignment of orphan sequences with a family is more successful if the family contains more sequences. However, the effect of aligning several orphan sequences simultaneously is unpredictable, depending on the relatedness between the sequences. The alignment of the orphans may also be improved if a small subfamily can be created for the orphan, as illustrated by the tests of equidistant families. This has practical applications, notably in the context of the genome sequencing projects, which lead to the problem of the alignment of a small number of orphan, eukaryotic sequences with large families of prokaryotic origin. It is clear that one should include the maximum number of sequences possible to achieve the best results. In fact, even highly related sequences can provide additional useful information. If no sequences are available to form subfamilies with the orphans, then the orphans should be aligned individually with the family. In reality, this is precisely what the progressive programs try to do in following the branching order of a guide tree. However, in difficult cases, such as the introduction of several highly divergent orphans, the guide tree based only on pairwise alignments of all the sequences may not be correct. This suggests that the progressive programs may be improved by a reconstruction of the guide tree during the multiple alignment process.

We have shown that the choice of an alignment program depends on the sequence set to be aligned and that no single ‘best’ program exists. In particular, the re-partition of the sequences, the sequence length and the presence of N/C-terminal extensions affect the accuracy and reliability of the programs. None of the alignment programs included in this study are capable of producing good, reliable alignments in all of these instances.

Other alignment problems, such as multi-domain proteins, runs of residues, repeats and transmembrane proteins, that have not been addressed here will be included in future updates of BAliBASE. We have preferred to concentrate on the alignment performance of the programs and other factors which may also affect the choice of program, such as ease of use, program portability and computer time and space requirements, have not been addressed here. All the programs in this study have been tested using default parameters. Work is now in progress to investigate the effect of changing alignment parameters such as residue comparison matrices and gap opening and extension penalties. It may be possible that a more suitable choice of parameters will significantly improve the performance of the alignment programs for certain tests.

It is clear that future work to improve alignment programs should concentrate on the problems of large insertions, extensions and sequence fragments. The alignment of sequences of similar length is relatively successful, even if there is only weak identity between the sequences. Another important area of interest which is becoming more and more frequent is the alignment of families of sequences. Increasing the number of sequences in an alignment set often significantly improves the alignment quality of divergent sequences.

The results of this program comparison may be used to indicate the most suitable program for a particular alignment problem. It should now be possible to predetermine the nature of a set of sequences, in particular the re-partition of sequence identities and the presence of unequal length sequences. An alignment server would then be able to automatically select the program which is likely to give the most accurate results.

ACKNOWLEDGEMENTS

We would like to thank D. Moras and J.-C. Thierry for their continuous support during this work. We are grateful to L. Moulinier, J.-M. Wurtz, F. Jeanmougin and T. Gibson for helpful discussions and critical reading of the manuscript. The work was supported by institute funds from INSERM, CNRS, HUS and Bristol-Myers Squibb.

REFERENCES

1. Pearson,W.R. and Lipman,D.J. (1988) *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.
2. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) *J. Mol. Biol.*, **215**, 403–410.
3. Sonnhammer,E.L. and Kahn,D. (1994) *Protein Sci.*, **3**, 482–492.
4. Gracy,J. and Argos,P. (1998) *Bioinformatics*, **14**, 164–173.
5. Jiang,J. and Jacob,H.J. (1998) *Genome Res.*, **8**, 268–275.
6. Taylor,W.R. (1998) *J. Mol. Biol.*, **280**, 375–406.
7. Gupta,S.K., Kececioglu,J.D. and Schaffer,AA. (1995) *J. Comput. Biol.*, **2**, 459–472.
8. Feng,D.F. and Doolittle,R.F. (1987) *J. Mol. Evol.*, **25**, 351–360.
9. Barton,G.J. and Sternberg,J.E. (1987) *J. Mol. Biol.*, **198**, 327–337.
10. Taylor,W.R. (1988) *J. Mol. Evol.*, **28**, 161–169.
11. Thompson,J.D., Gibson,T.J., Plewniak,F., Jeanmougin,F. and Higgins,D.G. (1997) *Nucleic Acids Res.*, **25**, 4876–4882.
12. Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) *Nucleic Acids Res.*, **22**, 4673–4680.
13. Needleman,S.B. and Wunsch,C.D. (1970) *J. Mol. Biol.*, **48**, 443–453.
14. Sneath,P.H. and Sokal,R.R. (1973) *Numerical Taxonomy*. Freeman, San Francisco, CA.
15. Saitou,N. and Nei,M. (1987) *Mol. Biol. Evol.*, **4**, 406–425.
16. Smith,R.F and Smith,T.F. (1992) *Protein Eng.*, **5**, 35–41.
17. Smith,R.F., Waterman,M.S. and Fitch,W.M. (1981) *J. Mol. Evol.*, **18**, 38–46.
18. Morgenstein,B., Dress,A. and Werner,T. (1996) *Proc. Natl Acad. Sci. USA*, **93**, 12098–12103.
19. Gotoh,O. (1996) *J. Mol. Biol.*, **264**, 823–838.
20. Notredame,C. and Higgins,D.G. (1996) *Nucleic Acids Res.*, **24**, 1515–1524.
21. Notredame,C., Holm,L. and Higgins,D.G. (1998) *Bioinformatics*, **14**, 407–422.
22. Baldi,P., Chauvin,Y., Hunkapiller,T. and McClure,M.A. (1994) *Proc. Natl Acad. Sci. USA*, **91**, 1059–1063.
23. Krogh,A., Mian,I.S. and Haussler,D. (1994) *Nucleic Acids Res.*, **22**, 4768–4778.
24. Eddy,S.R. (1995) *Ismb*, **3**, 114–120.
25. McClure,M.A, Vasi,T.K. and Fitch W.M. (1994) *Mol. Biol. Evol.*, **11**, 571–592.
26. Briffeuil,P., Baudoux,G., Lambert,C., De Bolle,X., Vinals,C., Feytmans,E. and Depiereux,E. (1998) *Bioinformatics*, **4**, 357–366.
27. Thompson,J.D., Plewniak,F. and Poch,O. (1999) *Bioinformatics*, **1**, 87–88.
28. Friedman,M. (1937) *J. Am. Stat. Assoc.*, **32**, 675–701.
29. Wilcoxon,F. (1947) *Biometrics*, **3**, 119–122.
30. Doolittle,R.F. (1981) *Science*, **214**, 149–159.