Nucleic Acids Research, 2005, Vol. 33, No. 2 ***511–518***
doi:10.1093/nar/gki198

MAFFT version 5: improvement in accuracy of multiple sequence alignment

Kazutaka Katoh[1,*], Kei-ichi Kuma[1], Hiroyuki Toh[1] and Takashi Miyata[2,3,4]

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto 611-0011, Japan, [2]Biohistory Research Hall, Takatsuki, Osaka 569-1125, Japan, [3]Department of Electrical Engineering and Bioscience, Science and Engineering, Waseda University, Tokyo 169-8555, Japan and [4]Department of Biophysics, Graduate School of Science, Kyoto University, Kyoto 606-8502, Japan

Received October 14, 2004; Revised November 16, 2004; Accepted December 29, 2004

ABSTRACT

The accuracy of multiple sequence alignment program MAFFT has been improved. The new version (5.3) of MAFFT offers new iterative refinement options, H-INS-i, F-INS-i and G-INS-i, in which pairwise alignment information are incorporated into objective function. These new options of MAFFT showed higher accuracy than currently available methods including TCoffee version 2 and CLUSTAL W in benchmark tests consisting of alignments of >50 sequences. Like the previously available options, the new options of MAFFT can handle hundreds of sequences on a standard desktop computer. We also examined the effect of the number of homologues included in an alignment. For a multiple alignment consisting of ~8 sequences with low similarity, the accuracy was improved (2–10 percentage points) when the sequences were aligned together with dozens of their close homologues (*E*-value < 10^{-5}–10^{-20}) collected from a database. Such improvement was generally observed for most methods, but remarkably large for the new options of MAFFT proposed here. Thus, we made a Ruby script, mafftE.rb, which aligns the input sequences together with their close homologues collected from SwissProt using NCBI-BLAST.

INTRODUCTION

Multiple alignment is an important tool for computational analysis of nucleotide or amino acid sequences. MAFFT (1) is one of the fastest methods among the currently available multiple alignment tools (2), and used in several projects, such as Pfam (3), ASTRAL (4) and MEROPS (5). In MAFFT, an initial alignment is constructed by the progressive method (6,7) and then refined by the iterative refinement method (8,9). The outline of procedure of the previous version of MAFFT is briefly explained below and in the lower part of Table 1. A user can select an appropriate strategy from the fastest one (FFT-NS-1) to the most accurate one (FFT-NS-i).

Progressive alignment (1). A rough distance between every pair of input sequences is rapidly calculated based on the number of 6-tuples shared by the two sequences (1,10,11). A guide tree is constructed from the distances with the UPGMA method (12) with modified linkage (see supplementary material on our web page, http://www.biophys.kyoto-u.ac.jp/~katoh/programs/align/mafft/suppl/). Input sequences are progressively aligned (6,7) following the branching order of the guide tree. This procedure is referred to as FFT-NS-1.

Progressive alignment (2). The initial distance matrix is less reliable than that based on all pairwise alignments. We can obtain more reliable distance matrix by using the FFT-NS-1 alignment (1,11,13). Progressive alignment is re-performed based on the new tree calculated from the new distance matrix. This method is referred to as FFT-NS-2.

Iterative refinement. The FFT-NS-2 alignment is further improved by the iterative refinement method (8,9) that optimizes the weighted sum-of-pairs (WSP) score proposed by Gotoh (14), using an approximate group-to-group alignment algorithm (1) and the tree-dependent restricted partitioning technique (15). This procedure is referred to as FFT-NS-i.

For the progressive alignment processes, a fast Fourier transform (FFT) approximation (1) is used in the FFT-NS-2, FFT-NS-1 and FFT-NS-i options (collectively denoted as FFT-NS-[12i] hereafter). When the sequences under consideration are highly conserved, these options require CPU times effectively proportional to average sequence length L for amino acid or nucleotide sequence alignments consisting homologues of a single gene. Note that it is not $L \log L$, although FFT takes $L \log L$ operations. This is because CPU time required by the FFT phase is much smaller than that by the

*To whom correspondence should be addressed. Tel: +81 774 38 3119; Fax: +81 774 38 3059; Email: kkatoh@kuicr.kyoto-u.ac.jp

Table 1. Options of version 5.3 (upper) and the previous version (lower) of MAFFT

	FFT	Initial distance matrix	Guide tree(s)	Iterative refinement	Alignment score	Command
G-INS-i	On[a]	Global[a]	UPG-m[e]	On	WSP+I[g]	`mafft --maxiterate 1000 --globalpair`
H-INS-i		FASTA-SW[b]	UPG-m[e]	On	WSP+I[g]	`mafft --maxiterate 1000 --fastswpair`
F-INS-i		FASTA[c]	UPG-m[e]	On	WSP+I[g]	`mafft --maxiterate 1000 --fastapair`
H-INS-1		FASTA-SW[b]	UPG-m[e]			`mafft --maxiterate 0 --fastswpair`
FFT-NS-i	On	6-tuple[d]	UPG-m×2[e,f]	On	WSP[h]	`mafft --maxiterate 1000`
FFT-NS-2	On	6-tuple[d]	UPG-m×2[e,f]			`mafft --maxiterate 0 --retree 2`
FFT-NS-1	On	6-tuple[d]	UPG-m[e]			`mafft --maxiterate 0 --retree 1`
NW-NS-i		6-tuple[d]	UPG-m×2[e,f]	On	WSP[h]	`mafft --maxiterate 1000 --nofft`
NW-NS-2		6-tuple[d]	UPG-m×2[e,f]			`mafft --maxiterate 0 --retree 2 --nofft`
NW-NS-1		6-tuple[d]	UPG-m[e]			`mafft --maxiterate 0 --retree 1 --nofft`

[a]All pairwise alignments are computed by global alignment with an FFT approximation. The FFT approximation is disabled in the progressive alignment stage.
[b]All pairwise alignments are computed with FASTA (25) with the Smith–Waterman optimization.
[c]All pairwise alignments are computed with FASTA (25) without the Smith–Waterman optimization.
[d]Distance matrix is calculated based on the number of 6-tuples shared by two sequences (1,10).
[e]UPGMA tree with a modified linkage (for detail see Supplementary Material).
[f]Guide tree is recalculated based on the first alignment and progressive alignment is re-performed (1,13).
[g]'Importance' (*I*) value is considered as described in text.
[h]WSP score is optimized through the iterative refinement (14).

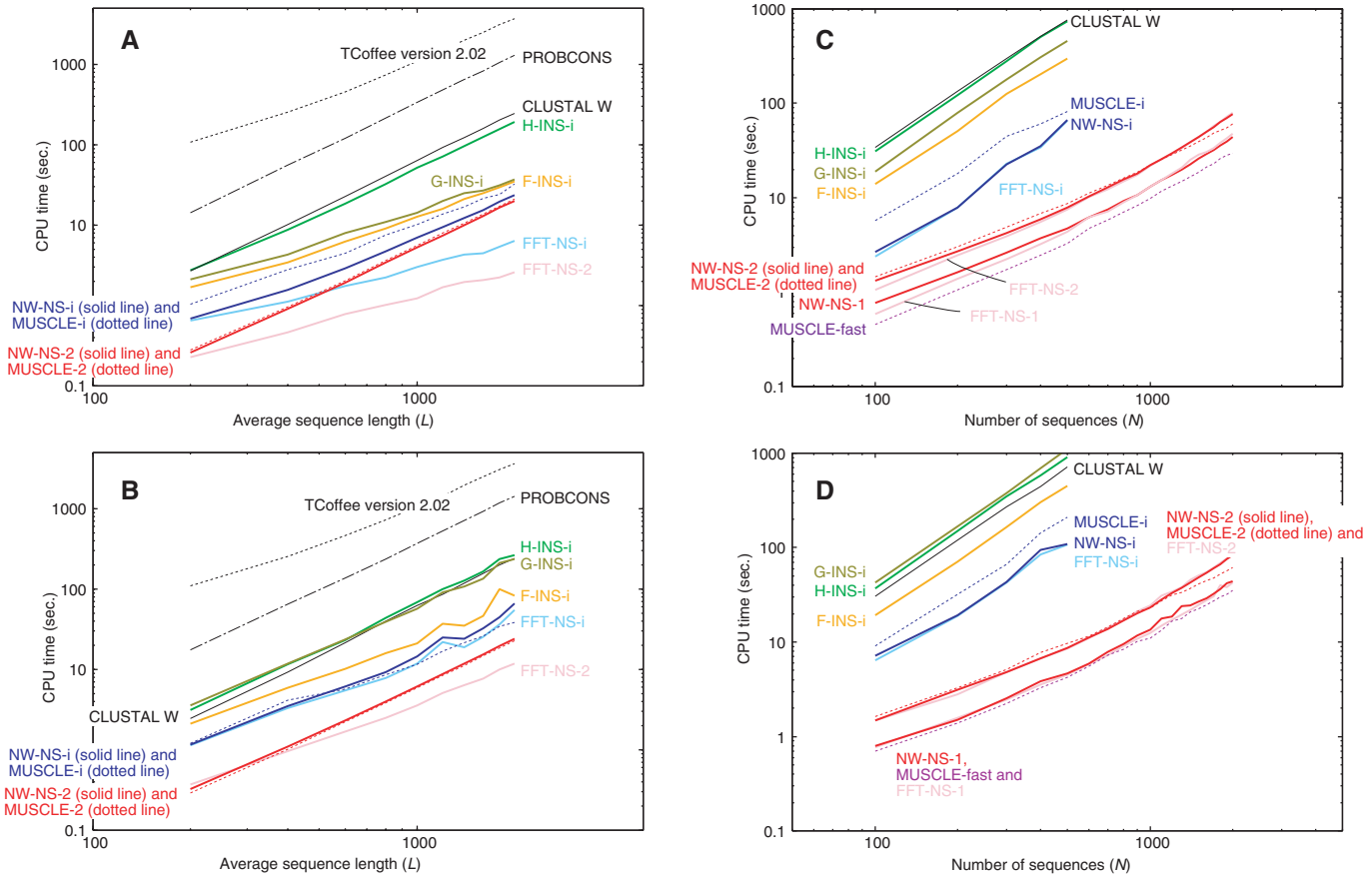

Figure 1. The CPU times required for various sizes of alignments. Sequences were generated using the ROSE program (29). (**A** and **B**) Average length (*L*) of input sequences versus CPU time. The number of sequence is 40. Average distance among input sequences is 100 PAM (A) (percentage identity ∼ 35–85) or 250 PAM (B) (percentage identity ∼ 15–65). (**C** and **D**) The number of input sequences (*N*) versus CPU time. Average sequence length is 300. Average distance among input sequences is 100 PAM (C) or 250 PAM (D). See Table 1 for command-line options for each strategy in MAFFT. Options of other programs are as follows:
TCoffee, default;
PROBCONS, default;
CLUSTAL W, default;
MUSCLE-i, `muscle -maxiters 16`;
MUSCLE-2, `muscle -maxiters 1`;
MUSCLE-fast, `muscle -sv -maxiters 1 -diags1 -distance1 kbit20_3`.

dynamic programming (DP) phase, unless extremely long sequences like genomic sequences are input. The slopes of the FFT-NS-[2i] lines are indeed near to 1 in Figure 1A.

MAFFT also offers three options NW-NS-[12i] with full DP (16) for the same part. In contrast to the case of FFT-NS-[12i], the time complexity of full DP options is proportional to L^2, independently to the similarity among input sequences. In most cases, an FFT-based strategy results in the same alignment as that by the corresponding option with full DP. The difference in accuracy was not statistically significant between the alignments generated with and without the FFT approximation in all cases we tested.

The parameters of MAFFT, such as gap penalties and scoring matrix, have not been minutely investigated because of limitation in the number of reference alignments available when it had been developed. Recently, large reference alignment databases, such as HOMSTRAD (17), SABmark (18) and PREFAB (11), have been independently established. They are valuable resources to select good parameters for a sequence alignment program as well as to evaluate the performance of a program.

CLUSTAL W is a widely utilized program of multiple sequence alignment. Other algorithms have tried to improve on the accuracy of CLUSTAL W. Gotoh (19) developed PRRP/N and found significant improvement by the iterative refinement method that uses the WSP score as objective function. TCoffee (20) employs progressive strategy but achieved the highest accuracy (1,21,22). This is because TCoffee constructs a multiple sequence alignment by combining information derived from heterogeneous sources, such as a global multiple alignment and local alignments. Although this ability is of great value, TCoffee requires a large CPU time proportional to N^3. Thus, it is hard to apply TCoffee to a large alignment consisting of dozens of sequences. The accuracy of TCoffee has recently been further improved in version 2 when compared with version 1. Recently, Edgar (11,23) implemented progressive and iterative refinement alignment strategies in MUSCLE. Although the algorithms of major options of MUSCLE seem similar to NW-NS-[2i] explained above, MUSCLE has an original option, MUSCLE-fast, which is faster and less accurate than other options of MUSCLE in most cases. Do *et al.* (manuscript submitted) proposed a new method, PROBCONS, whose accuracy is comparable or slightly higher than TCoffee version 2.

In MAFFT version 5.3, (i) parameters were optimized based on a number of reference alignments and (ii) three new strategies (G-INS-i, H-INS-i and F-INS-i; collectively denoted as [GHF]-INS-i hereafter) were introduced. In an attempt to improve alignment accuracy, [GHF]-INS-i uses a TCoffee-like approach (20) in incorporating all pairwise alignment information into an objective function. Iterative refinement for the objective function can be performed in reasonable time, as [GHF] INS-i was designed to have low computational complexity.

It was suggested that the accuracy of a multiple alignment of distantly related sequences is improved if they are aligned together with a number of their homologues (24), because the information from many sequences is expected to reduce the 'noise' (19). However, this possibility has not been quantitatively examined for recently developed methods. We also evaluated the effect of the number of homologues involved in an alignment.

MATERIALS AND METHODS

Introducing pairwise alignment information

MAFFT version 5.3 has three new iterative refinement options, G-INS-i, H-INS-i and F-INS-i. In these three strategies, all pairwise alignment information are included when constructing a multiple alignment. Three different algorithms for all pairwise alignment were tested; G-INS-i uses global alignment with an FFT approximation (1), whereas the other two options ([HF]-INS-i) incorporate local alignment information. To obtain local alignment information, H-INS-i uses the fasta34 program of the FASTA version 3.4t24 (25). F-INS-i uses a modified fasta34 program, in which the Smith–Waterman optimization is disabled. The [HF]-INS-i options do not use FFT.

The outlines of the algorithms of [GHF]-INS-i are as follows:

(i) An initial distance matrix is constructed from the pairwise scores, instead of shared 6-tuples, using the equation shown in (1) and a guide tree is build with the UPGMA method with modified linkage. Unlike FFT-NS-[2i] explained above, the re-construction of guide tree is not performed, because it did not provide significant improvement in accuracy in our tests.

(ii) Each pairwise alignment is divided into gap-free segments and number n is assigned to each segment. The information of these segments is stored in a set of arrays, score [$S(s, t, n)$], which represents the alignment score of the nth gap-free segment between sequences s and t, length [$L(s, t, n)$] of the aligned segment (s, t, n), position [$\mathbf{P}(s, t, n)$] in each of sequences s and t, and the importance value [$E(s, t, n)$] that is calculated, as described below, from the score of the segment and how frequently the residues are involved in gap-free segments. We denote $(s, t, p, q) \in \mathbf{P}(s, t, n)$ if the pth site of sequence s is aligned to the qth site of sequence t in aligned segment (s, t, n).

(iii) The frequency value $f(s, p)$, which represents how frequently the pth site of sequence s is involved in gap-free segments, is calculated as

$$f(s,p) = \sum_{n,t,q}^{(s,t,p,q)\in\mathbf{P}(s,t,n)} w_t,$$

where w_t is the weighting factor [for definition see (7)] for sequence t. The importance value $E(s, t, n)$ for aligned segment is calculated as

$$E(s,t,n) = \sum_{i,j}^{(s,t,i,j)\in\mathbf{P}(s,t,n)} \frac{f(s,i)+f(t,j)}{2L(s,t,n)} \times S(s,t,n).$$

We define the importance matrix $I(s, t, p, q)$ between the pth site of sequence s and the qth site of sequence t as

$$I(s,t,p,q) = \begin{cases} \sum_n E(s,t,n) & \text{if } (s,t,p,q) \in \mathbf{P}(s,t,n) \\ 0 & \text{otherwise.} \end{cases}$$

(iv) An alignment of a subset of given sequences, which is generated during the procedures of progressive and iterative refinement methods, is referred to as 'group'.

To align groups i and j, matrix $H(\text{group}i, \text{group}j, p, q)$ is constructed as

$$H(\text{group}i, \text{group}j, p, q) = \sum_{s,t}^{s\in\text{group}i,\ t\in\text{group}j} w_{st}\{\hat{M}_{A(s,p)A(t,q)} + W^I I(s,t,p,q)\},$$

where $A(s, p)$ is the pth amino acid residue on sequence s. $\hat{M}_{ab}$ is a score between a pair of amino acids a and b. The score matrices examined in this study are described in the 'parameter optimization' section. w_{st} is a weighting factor between sequences s and t. A weighting scheme proposed by Thompson *et al.* (7) is used in progressive alignment stage, and a weighting scheme proposed by Gotoh (14) is used in iterative refinement stage. W^I is a weighting factor, which was set at 2.7 in the current version, for the importance value. The alignment between two groups is computed by applying the DP algorithm (16) to matrix $H(,,,)$ at each step of progressive alignment. The alignment produced by this procedure is referred to as [GHF]-INS-1. Of these three progressive strategies, we evaluated the H-INS-1 option only.

(v) The [GHF]-INS-1 alignment is improved by the iterative refinement method ([GHF]-INS-i), which optimizes an objective score defined as the summation of the WSP score (14) and the importance values defined above. This score is referred to as WSP+I in this paper.

To reduce the CPU time consumed by this step, highly conserved regions are anchored and excluded from re-alignment if they are found (19,23). Conserved regions are identified only from sequence similarity, without considering the importance matrix $I(,,,)$, in the current version.

Performance evaluation

An up-to-date version of HOM39 (26) was extracted from the July 2004 release of HOMSTRAD (17) (http://www-cryst.bioc.cam.ac.uk/~homstrad/) based on two criteria used in (26). HOMSTRAD is a curated database of structural alignments of homologous proteins whose coordinates are available. Each entry of HOMSTRAD, a structural alignment, is extended by introducing homologous sequences with CLUSTAL W. Only the alignments based on structural superposition were used in this study. Out of 1033 entries of the HOMSTRAD, 55 entries (19.7% pairwise identity, 7.69 sequences and 159 aligned residues on average) were extracted for the evaluation of alignment accuracy. This dataset is referred to as 'HOM+0' in this paper.

We made the 'HOM+20,' 'HOM+50' and 'HOM+100' datasets by extending each entry of HOM+0 in a way similar to PREFAB (11). Amino acid sequences similar (E-value < 10^{-10}) to each member of an entry were collected from the SwissProt database (rel. 43) using BLAST (27) and added to the entry. If more than n (=20, 50 or 100) sequences were collected, we randomly selected n sequences to be added. Only amino acid positions of the sequences that were reported to show significant similarity by BLAST were added. The accuracy of an alignment was measured by the fraction of columns aligned identically to the reference alignment. When we evaluated the accuracy, the n sequences added to the HOM+n were removed.

SABmark (18) version 1.65 was downloaded from http://bioinformatics.vub.ac.be/databases/databases.html. SABmark is designed to assess the performance of protein sequence alignment algorithms and consists of two parts, the Twilight Zone set (with 'very low' similarity; referred to as the TWI set in this paper) and the Superfamily set (with 'low' similarity; referred to as SUP). The TWI set was mainly used in the present study to examine the abilities of algorithms for aligning distantly related sequences. The TWI set was also extended in the same manner as described above. These are hereafter referred to as 'TWI+n' (n = 0, 20 and 50). The accuracy value f_D, the ratio of the number of correctly aligned residues divided by the length of reference alignment, was calculated using the score.pl script provided by the authors of SABmark. The accuracies were separately considered for two subsets. One subset (denoted as TWIf+n) includes only the sequence pairs classified to the same family by Van Walle *et al.* (18), and the other subset (denoted as TWIs+n) consists of the sequence pairs classified not to the same family but to the same superfamily.

The PREFAB (11) version 3 dataset was downloaded from http://www.drive5.com/muscle/prefab.htm. The accuracy was measured using Q, the number of correctly aligned residue pairs divided by the number of residue pairs in the reference alignment (11).

Parameter optimization

Gap-opening penalty S^{op} and the offset value S^a [for definitions see (1)] were determined to provide the highest accuracy of the FFT-NS-2 strategy for the TWIf+0 set using golden section search (28). We examined five scoring matrices (BLOSUM45, 62, 80, JTT100 and JTT200) and selected the matrix providing the highest accuracy.

Availability

MAFFT was written in C, and runs on Linux, Mac OS X and the Cygwin environment on Windows. The MAFFT package is available at http://www.biophys.kyoto-u.ac.jp/~katoh/programs/align/mafft/. The fasta34 program of the FASTA package (25) must be installed to run the H-INS-i option. The F-INS-i option requires a one-line modification of the source code of the fasta34 program (see supplementary material on our web page, http://www.biophys.kyoto-u.ac.jp/~katoh/programs/align/mafft/suppl/). The G-INS-i option requires no additional package. The performances were measured on a 2.8 GHz Xeon processor with 1 GB of RAM running SuSE Linux 9.0. The gcc version 3.3.1 compiler was used with the '-O3' optimization option.

We also made a Ruby script mafftE.rb that aligns input sequences together with their homologues automatically collected from local database or SwissProt using NCBI-BLAST.

RESULTS AND DISCUSSION

We evaluated the performance of MAFFT version 5.3 using the HOM, TWIf, TWIs and PREFAB datasets, and compared it with those of several methods developed by other groups, including TCoffee version 2.02 (20), CLUSTAL W version 1.83 (7), PROBCONS version 1.06 and MUSCLE version 3.41 (11,23). As for MUSCLE, the most accurate option

was used, which probably corresponds to NW-NS-i of MAFFT (see Table 1).

Improvement in accuracy attained by new parameter set

The golden section search provided the optimal parameters of 1.53 for S^{op} and 0.123 for S^{a} when the FFT-NS-2 option was applied to the TWIf+0. Of the five matrices examined, BLOSUM62 showed the highest accuracy. The improvement in accuracy from the previous parameter set (JTT200, $S^{op} = 2.40$, $S^{a} = 0.06$) was ~5 percentage points. Although the new parameter set was not optimal for other options or for other datasets (HOM, TWIs and PREFAB), the new parameter set provided generally higher accuracy than the previous parameter set by ~5 percentage points, as shown in Supplementary Material. Accuracy values for various combinations of S^{op} and S^{a} are also shown in Supplementary Material. Note that comparisons between MAFFT and other methods based on other than TWIf+0 are important, because FFT-NS-2 option of MAFFT was tuned for TWIf+0. The results of the comparisons are described below.

Improvement in accuracy by introducing pairwise alignment information

Tables 2, 3 and 4 show the results of benchmark tests using the HOM, TWIf, TWIs and PREFAB datasets, respectively. The difference in accuracy between [GHF]-INS-i and FFT-NS-i was at most 6%, which corresponded to the improvement by introducing the strategy presented in this paper. As noted in Materials and Methods, the W^{I} value was set at 2.7 in the calculations shown in Tables 2, 3 and 4. However, the optimal W^{I} value that provided the highest accuracy differed depending on the number of homologues, sequence similarity and other conditions. The accuracy values on various W^{I} value are shown in Supplementary Material. For new strategies presented here, further investigation is necessary to determine the optimal parameters including the W^{I} values and the parameters for pairwise alignments.

The difference in accuracy among newly proposed strategies, G-INS-i, H-INS-i and F-INS-i was small. A slight tendency was observed that G-INS-i is suitable for alignments consisting of large number of sequences, whereas F-INS-i and H-INS-i are suitable for alignments consisting of small number of sequences. In addition, G-INS-i is expected to be not suitable for alignments with large gaps, as it uses all pairwise global alignments. Although G-INS-i uses an FFT approximation for the all pairwise alignment process, its accuracy was virtually identical to that of a strategy in which full DP was performed for this process (data not shown).

Reducing CPU time

In the [GHF]-INS-i strategies, all pairwise alignment and iterative refinement processes are time consuming. Figure 1 A–D shows the CPU time as the function of the sizes of sequence data generated by the ROSE program (29). The CPU time consumed by all pairwise alignment process can be reduced by approximations, such as banded alignment implemented in FASTA (F-INS-i uses it) or the FFT approximation (G-INS-i uses it), although the latter is effective only for highly conserved sequences.

Table 2. Comparison of performances of several methods based on 55 alignments in HOM tests

Method	Dataset	Accuracy (%)	Improvement	CPU time (s)
G-INS-i				
	HOM+0	42.58[b]	—	44.31
	HOM+20	**52.06**	+9.48[c]	182.3
	HOM+50	**53.85**	+11.2[c]	514.2
	HOM+100	**54.61**	+12.0[c]	1405
H-INS-i				
	HOM+0	43.20[b]	—	38.68
	HOM+20	49.56	+6.36[c]	151.2
	HOM+50	53.37	+10.2[c]	426.8
	HOM+100	53.29	+10.1[c]	1110
F-INS-i				
	HOM+0	43.14[b]	—	32.06
	HOM+20	51.26	+8.12[c]	122.0
	HOM+50	53.72	+10.6[c]	342.0
	HOM+100	53.57	+10.4[c]	758.4
H-INS-1				
	HOM+0	38.55[a]	—	14.30
	HOM+20	46.00[a]	+7.45[c]	73.81
	HOM+50	48.80[a]	+10.3[c]	237.9
	HOM+100	48.35[a]	+9.80[c]	636.6
FFT-NS-i				
	HOM+0	43.57[b]	—	32.57
	HOM+20	49.57[b]	+6.00[c]	73.84
	HOM+50	50.68[b]	+7.11[c]	155.87
	HOM+100	50.73[b]	+7.16[c]	365.8
FFT-NS-2				
	HOM+0	35.94[a]	—	6.22
	HOM+20	45.06[a]	+9.12[c]	15.23
	HOM+50	44.42[a]	+8.48[c]	26.46
	HOM+100	43.61[a]	+7.67[c]	43.46
PROBCONS 1.06				
	HOM+0	**47.95**	—	91.13
	HOM+20	51.78	+3.83[d]	590.1
	HOM+50	51.59	+3.64[d]	2237
	HOM+100	51.81	+3.86[d]	7634
MUSCLE-i 3.41				
	HOM+0	43.44[b]	—	37.20
	HOM+20	45.94[b]	+2.50	113.6
	HOM+50	46.90[a]	+3.46	403.7
	HOM+100	48.07[a]	+4.63[c]	719.4
TCoffee 2.02				
	HOM+0	43.49[b]	—	486.4
	HOM+20	48.26	+4.77[c]	5007
	HOM+50	49.71	+6.22[c]	28 250
	HOM+100	49.94	+6.45[c]	71 390
CLUSTAL W 1.83				
	HOM+0	36.77[a]	—	16.29
	HOM+20	36.57[a]	–0.20	87.98
	HOM+50	37.33[a]	+0.56	242.5
	HOM+100	36.77[a]	+0.00	620.6

The highest accuracy value within each dataset is in boldface.
[a]The difference from the highest accuracy was shown to be significant ($P < 0.01$) by both the Wilcoxon test and the Friedman test.
[b]The Wilcoxon test showed a significant difference but the Friedman test did not.
[c]The improvement of score from HOM+0 was shown to be significant by both the Wilcoxon test and the Friedman test.
[d]The Wilcoxon test showed a significant improvement but the Friedman test did not. See Table 1 for command-line options for each method in MAFFT. Command-line option for MUSCLE-i is `muscle -maxiters 1000`.

The CPU time for the iterative refinement process can be slightly reduced by always accepting the new alignment produced by group-to-group re-alignment without calculation of WSP+I score. Note that the WSP+I score may become worse

Table 3. Comparison of performances of several methods based on 209 alignments in TWI tests

Method	Dataset	TWIs		TWIf		CPU
		f_D (%)	Improvement	f_D (%)	Improvement	time (s)
G-INS-i						
	TWI+0	20.73[a]	—	41.68[b]	—	232.1
	TWI+20	**27.38**	+6.65[c]	47.00[b]	+5.32[c]	747.4
	TWI+50	**29.58**	+8.85[c]	**51.11**	+9.43[c]	1724
H-INS-i						
	TWI+0	**23.36**	—	42.78	—	154.1
	TWI+20	26.30[a]	+2.94[c]	**47.40**	+4.62[c]	467.0
	TWI+50	27.87[a]	+4.51[c]	50.29[b]	+7.51[c]	1102
F-INS-i						
	TWI+0	22.03[a]	—	43.21	—	155.8
	TWI+20	25.80[a]	+3.77[c]	47.12	+3.91[c]	405.1
	TWI+50	27.25[a]	+5.22[c]	47.59[a]	+4.38[c]	882.8
H-INS-1						
	TWI+0	18.28[a]	—	38.20[a]	—	30.29
	TWI+20	22.48[a]	+4.20[c]	43.81[a]	+5.61[c]	144.6
	TWI+50	24.77[a]	+6.49[c]	45.76[a]	+7.56[c]	460.6
FFT-NS-i						
	TWI+0	18.16[a]	—	37.46[a]	—	124.1
	TWI+20	21.64[a]	+3.48[c]	40.88[a]	+2.29[c]	303.8
	TWI+50	22.76[a]	+4.60[c]	44.85[a]	+7.49[c]	565.6
FFT-NS-2						
	TWI+0	12.89[a]	—	30.27[a]	—	19.41
	TWI+20	16.14[a]	+3.25[c]	33.59[a]	+3.32[c]	44.54
	TWI+50	17.49[a]	+4.60[c]	37.08[a]	+6.87[c]	77.36
PROBCONS 1.06						
	TWI+0	22.06	—	**44.48**	—	234.0
	TWI+20	22.79[a]	+0.73[d]	43.81[a]	–0.67	1747
	TWI+50	22.53[a]	+0.47	44.86[a]	+0.38	6889
MUSCLE-i 3.41						
	TWI+0	15.67[a]	—	36.38[a]	—	382.3
	TWI+20	17.98[a]	+2.31[c]	36.68[a]	+0.30	999.9
	TWI+50	19.61[a]	+3.94[c]	38.17[a]	+1.79[c]	2152
TCoffee 2.02						
	TWI+0	21.80[b]	—	44.20	—	1378
	TWI+20	22.81[a]	+1.01[c]	44.56[b]	+0.36[c]	13 900
	TWI+50	21.85[a]	+0.05[d]	45.18[a]	+0.98[c]	82 200
CLUSTAL W 1.83						
	TWI+0	12.76[a]	—	34.28[a]	—	31.52
	TWI+20	11.72[a]	–1.04	33.59[a]	–0.69	152.7
	TWI+50	12.91[a]	+0.15	34.95[a]	+0.67	458.8

See the footnote of Table 2.

by re-alignment, as MAFFT employs an approximate DP algorithm for group-to-group alignment described previously (1). According to our test, this rough iterative strategy gave less accurate results for alignments of few sequences, such as HOM+0, TWIf+0 and TWIs+0. However, this strategy performed well for alignments composed of a large number of sequences, such as HOM+50, +100, TWIf+50 and TWIs+50. Its accuracy was sometimes rather higher than that of G-INS-i (data not shown).

FFT-NS-i is the most accurate option in the previous version. The accuracy has been improved by introducing the new options. However, when the highest accuracy is not required for the alignment, the FFT-NS-i option may be still useful, considering the balance between computational time and accuracy.

Comparison with other methods

For alignments involving dozens of sequences (HOM+*n*, TWIs+*n* and PREFAB; $n \neq 0$), new strategies presented here ([GHF]-INS-i) outperformed other methods, including TCoffee and PROBCONS, in accuracy as shown in Tables 2, 3 and 4. According to the PREFAB test (Table 4), the difference was large when the input sequences were distantly related.

Results of the HOM+0 and TWI+0 tests are shown in Tables 2 and 3, respectively. In these cases, the number of input sequences is small (~8). We carried out two more tests based on BAliBASE (30) and the SUP set of SABmark, for which close homologues have not been added. PROBCONS outperformed [GHF]-INS-i by 1–3 percentage points for BAliBASE, whereas [GHF]-INS-i options outperformed PROBCONS by 1–3 percentage points for the SUP set of SABmark. In most of cases of such small alignments, TCoffee, PROBCONS and three new strategies ([GHF]-INS-i) of MAFFT were small in accuracy. However, for the HOM+0 test, the accuracy of PROBCONS was remarkably high as shown in Table 2; the difference from H-INS-i was 5% and statistically significant according to the Wilcoxon test.

Table 4. Accuracy values of several methods for the PREFAB tests

Method	Identity (%) 0–20	20–40	40–70	70–100	All	CPU time (s)
G-INS-i	46.75	82.77	**96.30**	98.60	68.85	16 030
H-INS-i	**48.22**	83.32	95.83	**98.64**	69.70	15 060
F-INS-i	48.00	**83.35**	95.83	98.62	69.61	9007
H-INS-1	46.13[a]	82.16[a]	95.42[b]	98.60	68.27	9910
FFT-NS-i	45.72[a]	80.95[a]	93.83[b]	98.55	67.48	4176
FFT-NS-2	43.19[a]	79.52[a]	93.23[b]	98.47	65.74	930.4
FFT-NS-1	40.90[a]	77.50[a]	93.46[a]	98.59	63.92	666.2
TCoffee 2.02	45.30[a]	82.36[b]	95.20	98.62	67.96	973 600
PROBCONS 1.06	45.63[b]	82.10	95.01[b]	98.18	67.95	142 200
MUSCLE-i	42.77[a]	80.43[a]	95.43	98.28	66.05	13 260
MUSCLE-fast	38.44[a]	76.65[a]	93.42[a]	97.91	62.44	544.1
CLUSTAL W (default)	33.96[a]	74.14[a]	93.54[b]	97.85	59.45	12 970

The highest accuracy value within each percent identity range is in bold letters.
[a]The difference from the highest accuracy was found to be significant ($P < 0.01$) by both the Wilcoxon test and the Friedman test.
[b]The Wilcoxon test showed a significant difference but the Friedman test did not. See Table 1 for command-line options for each strategy in MAFFT. Options of other programs are as follows:
TCoffee, default;
PROBCONS, default;
MUSCLE-i, `muscle -maxiters 1000;`
MUSCLE-fast, `muscle -sv -maxiters 1 -diags1 -distance1 kbit20_3;`
CLUSTAL W (default), default.

The CPU times of [GHF]-INS-i were several times smaller than those of TCoffee and PROBCONS.

Effect of the number of homologues involved in an alignment

As expected, the accuracy of a multiple alignment tended to increase with increasing number of homologues involved in the alignment. Although observed more or less for most methods, this tendency was remarkable for MAFFT. The improvement by adding close homologues became small, when the position-specific gap penalty (1) was disabled (data not shown). Thus, this technique probably contributed to the improvement. The position-specific gap penalty (1,7,9,23) was motivated by a consideration as follows. In a group-to-group alignment process, each group of sequences may contain gaps. If the gap is newly introduced at the same position as one of such existing gaps, the new gap should be less penalized, because the new and existing gaps are probably resulting from a single insertion or deletion event.

According to the HOM tests (Table 2), the improvements by adding 50–100 homologues were at most ~10 percentage points and statistically significant according to both the Wilcoxon test and the Friedman test. The improvement was comparable with that by introducing the structural information of one or two proteins (26), and rather larger than that by modification of algorithm (from FFT-NS-i to [GHF]-INS-i) presented in this paper. Similar results were obtained for the TWIf+*n* and TWIs+*n* datasets shown in Table 3. The accuracy values under various conditions (*n* and *E*-values) are shown in Supplementary Material. The maximum accuracy was obtained in the case of $n > 50$ or $n > 100$ and threshold of *E*-value = 10^{-5}–10^{-20}.

These results suggest the importance of including a number of homologues for obtaining an accurate sequence alignment. An ability to handle a large number of sequences is therefore important for a multiple sequence alignment program.

Perspectives

There are several issues for further improvement in accuracy and speed. (i) TCoffee has a merit that it can combine alignments based on different principles. O'Sullivan *et al*. (26) reported that the accuracy of a multiple alignment is improved when structural information of many proteins is included. We are planning to enable MAFFT to include structural information. (ii) There might be a more efficient way to collect and select the homologues from databases for improving the accuracy of an alignment. For example, we should exclude very close homologues of a sequence already involved in the alignment, because such close homologues are expected to bring little information.

SUPPLEMENTARY MATERIAL

Supplementary Material is available at NAR Online.

ACKNOWLEDGEMENTS

This work was supported in part by a Grant-in-Aid for Creative Scientific Research and a Grant for the Biodiversity Research of the 21st Century COE (A14) from the Ministry of Education, Culture, Sports, Science and Technology of Japan. One of the authors (H.T.) is financially supported by PDBj (BIRD). Funding to pay the Open Access publication charges for this article was provided by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

1. Katoh,K., Misawa,K., Kuma,K. and Miyata,T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res*., **30**, 3059–3066.
2. Grasso,C. and Lee,C. (2004) Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems. *Bioinformatics*, **20**, 1546–1556.
3. Bateman,A., Coin,L., Durbin,R., Finn,R.D., Hollich,V., Griffiths-Jones,S., Khanna,A., Marshall,M., Moxon,S., Sonnhammer,E.L., Studholme,D.J., Yeats,C. and Eddy,S.R. (2004) The pfam protein families database. *Nucleic Acids Res*., **32**, D138–D141.
4. Chandonia,J.M., Hon,G., Walker,N.S., Lo Conte,L., Koehl,P., Levitt,M. and Brenner,S.E. (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res*., **32**, D189–D192.
5. Rawlings,N.D., Tolle,D.P. and Barrett,A.J. (2004) MEROPS: the peptidase database. *Nucleic Acids Res*., **32**, D160–D164.
6. Feng,D.F. and Doolittle,R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol*., **25**, 351–360.
7. Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4680.
8. Berger,M.P. and Munson,P.J. (1991) A novel randomized iterative strategy for aligning multiple protein sequences. *Comput. Appl. Biosci*., **7**, 479–484.
9. Gotoh,O. (1993) Optimal alignment between groups of sequences and its application to multiple sequence alignment. *Comput. Appl. Biosci*., **9**, 361–370.

10. Jones,D.T., Taylor,W.R. and Thornton,J.M. (1992) The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.*, **8**, 275–282.
11. Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, **32**, 1792–1797.
12. Sokal,R.R. and Michener,C.D. (1958) A statistical method for evaluating systematic relationships. *Univ. Kans. Scientif. Bull.*, **28**, 1409–1438.
13. Tateno,Y., Ikeo,K., Imanishi,T., Watanabe,H., Endo,T., Yamaguchi,Y., Suzuki,Y., Takahashi,K., Tsunoyama,K., Kawai,M., Kawanishi,Y., Naitou,K. and Gojobori,T. (1997) Evolutionary motif and its biological and structural significance. *J. Mol. Evol.*, **44**, S38–S43.
14. Gotoh,O. (1995) A weighting system and algorithm for aligning many phylogenetically related sequences. *Comput. Appl. Biosci.*, **11**, 543–551.
15. Hirosawa,M., Totoki,Y., Hoshida,M. and Ishikawa,M. (1995) Comprehensive study on iterative algorithms of multiple sequence alignment. *Comput. Appl. Biosci.*, **11**, 13–18.
16. Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
17. Stebbings,L.A. and Mizuguchi,K. (2004) HOMSTRAD: recent developments of the homologous protein structure alignment database. *Nucleic Acids Res.*, **32**, D203–D207.
18. Van Walle,I., Lasters,I. and Wyns,L. (2004) Align-m—a new algorithm for multiple alignment of highly divergent sequences. *Bioinformatics*, **20**, 1428–1435.
19. Gotoh,O. (1996) Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J. Mol. Biol.*, **264**, 823–838.
20. Notredame,C., Higgins,D.G. and Heringa,J. (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.
21. Lassmann,T. and Sonnhammer,E.L. (2002) Quality assessment of multiple alignment programs. *FEBS Lett.*, **529**, 126–130.
22. Raghava,G.P., Searle,S.M., Audley,P.C., Barber,J.D. and Barton,G.J. (2003) OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics*, **4**, 47.
23. Edgar,R.C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 113.
24. Thompson,J.D., Plewniak,F., Thierry,J. and Poch,O. (2000) DbClustal: rapid and reliable global multiple alignments of protein sequences detected by database searches. *Nucleic Acids Res.*, **28**, 2919–2926.
25. Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.
26. O'Sullivan,O., Suhre,K., Abergel,C., Higgins,D.G. and Notredame,C. (2004) 3DCoffee: combining protein sequences and structures within multiple sequence alignments. *J. Mol. Biol.*, **340**, 385–395.
27. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
28. Press,W.H., Teukolsky,S.A., Vetterling,W.T. and Flannery,B.P. (1995) *Numerical Recipes in C: The Art of Scientific Computing, 2nd edn*. Cambridge University Press, Cambridge.
29. Stoye,J., Evers,D. and Meyer,F. (1997) Generating benchmarks for multiple sequence alignments and phylogenetic reconstructions. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **5**, 303–306.
30. Thompson,J.D., Plewniak,F. and Poch,O. (1999) BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, **15**, 87–88.