IPSJ Transactions on Bioinformatics Vol. 1 2–12 (Nov. 2008)

Original Paper

Improvement in Speed and Accuracy of Multiple Sequence Alignment Program PRIME

Shinsuke Yamada,[†1,†2] Osamu Gotoh[†2,†3] and Hayato Yamana[†1]

Multiple sequence alignment (MSA) is a useful tool in bioinformatics. Although many MSA algorithms have been developed, there is still room for improvement in accuracy and speed. We have developed an MSA program PRIME, whose crucial feature is the use of a group-to-group sequence alignment algorithm with a piecewise linear gap cost. We have shown that PRIME is one of the most accurate MSA programs currently available. However, PRIME is slower than other leading MSA programs. To improve computational performance, we newly incorporate anchoring and grouping heuristics into PRIME. An anchoring method is to locate well-conserved regions in a given MSA as anchor points to reduce the region of DP matrix to be examined, while a grouping method detects conserved subfamily alignments specified by phylogenetic tree in a given MSA to reduce the number of iterative refinement steps. The results of BAliBASE 3.0 and PREFAB 4 benchmark tests indicated that these heuristics contributed to reduction in the computational time of PRIME by more than 60% while the average alignment accuracy measures decreased by at most 2%. Additionally, we evaluated the effectiveness of iterative refinement algorithm based on maximal expected accuracy (MEA). Our experiments revealed that when many sequences are aligned, the MEA-based algorithm significantly improves alignment accuracy compared with the standard version of PRIME at the expense of a considerable increase in computation time.

1. Introduction

Multiple sequence alignment (MSA) provides a useful information for elucidating the relationships among function, evolution, sequence, and structure of biological macromolecules such as genes and proteins [1)–5)]. Theoretically, we can calculate the optimal alignment of many sequences by n-dimensional dynamic programming (DP). However, a DP method is practically applicable only when a small number of sequences are aligned. In fact, even when a sum-of-pairs (SP) score with the simplest gap cost is used as an objective function, constructing optimal MSA is an NP-hard problem [6)]. Hence, many heuristic methods have been developed. Almost all practical methods currently available adopt either a progressive [7)–9)] or an iterative refinement [10)–14)] heuristic strategy.

To speed up iterative refinement, several programs, such as MAFFT [12)] and MUSCLE [15)], adopt an additional heuristic approach, *i.e.* given a pair of groups of sequences (MSAs), these methods first find candidate segment (consecutive columns) pairs that could contribute to the optimal alignment between the groups, determine the optimal combination of segment pairs from the candidate pairs, and then align the groups into a single MSA based on restricted DP space flanked by the selected segment pairs used as anchor points. Instead of finding segment pairs from separate groups, another approach extracts anchor points that specify well-conserved regions on the given MSA to be refined by the iterative refinement steps [11),15)]. Although some papers discussed methods for extracting well-conserved regions from an MSA, these methods were mainly concerned with analysis or correction of the MSA [16),17)]. In addition, the effects of anchoring on the quality of the resultant MSA have not been explicitly discussed until now. Acceleration of computation by yet another heuristics that reduces the number of iterative refinement steps by grouping of closely related members in an MSA have been tried only by Prrn [11)] without any quantitative evaluation of their effects.

In order to improve alignment accuracy, especially when some of the sequences to be aligned have long insertions or deletions, recent programs incorporate consistency information among pairwise sequence alignments [8),13),18),19)]. Other programs employ additional information such as pairwise structure alignment, sequence-structure alignment, or secondary structure prediction [20)–22)]. Moreover, some recent studies adopt probabilistic alignment algorithms based on maximal expected accuracy (MEA) in place of the standard Needleman-Wunsch type DP algorithms [18)]. MEA-based algorithms have also been successfully applied to some bioinformatics applications related to sequence alignment [23),24)].

Previously, we have devised an MSA algorithm using a piecewise linear gap

†1 Waseda University
†2 Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST)
†3 Kyoto University

cost [14], and developed a program named PRIME. Although PRIME can construct accurate MSAs comparable to the most accurate programs currently available, its computational speed is somewhat slower than those of most MSA programs. Compared with similar iterative refinement algorithms, PRIME employs a relatively computationally intensive group-to-group sequence alignment algorithm. Therefore, some heuristic methods for reducing the computation of PRIME without a large amount of accuracy loss are highly desirable.

In this work, we newly incorporate anchoring and grouping methods into PRIME. An anchoring method is to locate well-conserved regions in a given MSA that act as anchor points to reduce the region of DP matrix to be examined, while a grouping method detects conserved subfamily alignments in a given MSA to reduce the number of iterative refinement steps. The results of BAliBASE 3.0 and PREFAB 4 benchmark tests indicated that the computational speed of PRIME was reduced by more than 60% while average alignment accuracy measures decreased by at most 2%. Additionally, we evaluated the effectiveness of iterative refinement algorithm based on MEA. Our experiments indicated that the MEA-based algorithm significantly improves alignment accuracy compared with the standard version of PRIME, although considerably longer computation time is required especially when many sequences are aligned.

2. Algorithms

2.1 PRIME Overview

For a given set of sequences, PRIME constructs an MSA based on a doubly nested randomized iterative strategy similar to our previous MSA program Prrn [11]. The crucial feature of PRIME is a group-to-group sequence alignment algorithm with a piecewise linear gap cost [25], which is the key to a progressive or an iterative refinement method. In this subsection, we briefly describe the algorithms of PRIME.

2.1.1 Doubly Nested Randomized Iterative Strategy

PRIME uses a weighted sum-of-pairs (WSP) score as the objective function of MSA, M, to be optimized. WSP is defined as $\sum_{i<j} w_{i,j} \cdot S_{i,j}$, where $w_{i,j}$ is the weight for the pair of i-th and j-th sequences in M and $S_{i,j}$ is the score of pairwise alignment induced from i-th and j-th rows of M. In order to optimize WSP score, PRIME employs a doubly nested randomized iterative strategy [11], involving refinement of MSA, phylogenetic tree, and pair weights until these triples are mutually consistent. After preparation of an initial MSA M with a simple progressive method using a group-to-group sequence alignment algorithm, this strategy refines M as follows:

(1) calculate a distance matrix from M
(2) construct a phylogenetic tree from the distance matrix
(3) calculate pair weights from the phylogenetic tree
(4) (Optional) apply anchoring method to M
(5) (Optional) apply grouping method to M (and M_{bfr}) and the phylogenetic tree
(6) iteratively refine M using the phylogenetic tree and the pair weights
  (a) $M_{bfr} \leftarrow M$
  (b) compile a branch list
  (c) randomly choose a branch b from the branch list
  (d) divide M into two groups based on b
  (e) align these two groups into a single MSA M_{aft} using a group-to-group sequence alignment algorithm
  (f) if WSP score of M_{aft} is greater than that of M, then $M \leftarrow M_{aft}$
  (g) repeat steps 6c to 6f until no better WSP score of M is obtained after examining all divisions of M based on all branches in the branch list
(7) repeat steps 1 to 6 until WSP of M is equal to that of M_{bfr}

Note that the above procedure contains anchoring and grouping methods, which are introduced in this study. A branch list includes all branches of the phylogenetic tree, except those in the excluded branch list obtained at step 5.

2.1.2 Group-to-group Sequence Alignment Algorithm with Piecewise Linear Gap Cost

The core algorithm of PRIME is the group-to-group sequence alignment algorithm with a piecewise linear gap cost [14], which aligns two groups of sequences (MSAs) into a single MSA based on a two-dimensional DP. The piecewise linear gap cost is one of the concave functions, consisting of L linear functions [25]. Since the inclination of this gap cost, which corresponds to the gap extension penalty, becomes small as gap length increases, this gap cost could alleviate over-penalizing long insertions or deletions. The group-to-group sequence alignment algorithm employs essentially the same recurrent relations as the pairwise sequence alignment algorithm [26]. The major difference between group-to-group sequence alignment algorithm and pairwise one is exact calculation of gap opening and extension penalties [14),27),28)]. In order to calculate gap opening penalty, a gap state plays a crucial role. The gap state denotes the number of consecutive nulls (blank characters indicating absence of the corresponding residues in the sequence) up to the current position. By comparing gap states, we can easily detect opening of a gap and hence calculate the gap opening penalty. For calculation of a gap extension penalty, dynamic gap information is required in addition to the gap states. A dynamic gap is a gap inserted during the DP process, and dynamic gap information is held by a list of the positions and lengths of dynamic gaps. By combining the dynamic gap information and gap states, we can calculate gap extension penalty efficiently. For the detailed description of the algorithm, see the previous paper [14].

2.2 Anchoring and Grouping Methods

In order to reduce the computation, we have newly introduced two heuristics: anchoring and grouping methods. An anchoring method is to locate a run of consecutive conserved columns in a given MSA that acts as an anchor point. Fixing such anchor points can significantly reduce the amount of DP matrix to be examined, *i.e.* the computation at step 6e of the doubly nested randomized iterative strategy. A grouping method detects conserved subfamily alignments in a given MSA. A subfamily is specified by an internal node of a phylogenetic tree, and a subfamily alignment is one induced from an alignment consisting of all sequences included in the subtree that descend from the internal node. Fixing the subfamily alignments can reduce the number of iterative refinement steps. We employ two types of anchoring and grouping methods: one is based on conservation, and the other on comparison. The conservation-based anchoring and grouping methods are applied when we first execute steps 4 and 5 of the doubly nested randomized iterative strategy, while the comparison-based anchoring and grouping methods are applied to the second or later execution of these steps.

2.2.1 Conservation-based Methods

Conservation-based anchoring and grouping methods calculates sum-of-pairs (SP) score for columns or subfamily alignments in an MSA, respectively. Given an MSA, the conservation-based anchoring method detects a run of consecutive conserved columns based on the following algorithm:

(1) calculate SP score for i-th column, SP_i
(2) smooth SP score: $SP'_i \leftarrow 1/(2r+1) \cdot \sum_{-r \leq k \leq r} SP_{i+k}$
(3) detect anchor points based on Z-scores of SP'_i

Because we would like to detect consecutive well-conserved columns only, we omit those columns that contain any nulls by setting SP_i and SP'_i of such columns to zero at steps 1 and 2. The procedure of step 2 includes a parameter r, which is set to 1 in this study. At step 3, we regard a stretch of columns as conserved if the Z-score of SP'_i exceeds the threshold, 1.8, by default.

In the grouping method, conserved subfamily alignments induced from a given MSA are judged by their SP scores as follows. Given an MSA and a phylogenetic tree T, the procedure is executed in a bottom-up manner similarly to that used in a progressive alignment method:

(1) label ‘unknown’ for all internal nodes of T and ‘conserved’ for all leaves of T
(2) for each ‘unknown’-labeled internal node p of T
  (a) if either child nodes of p is labeled ‘non-conserved’, then label p ‘non-conserved’
  (b) else if both child nodes of p are labeled ‘conserved’; then label p ‘conserved’ if SPS/P is more than a threshold, or ‘non-conserved’ otherwise
(3) compile excluded branch list

SPS is the SP score of the subfamily alignment specified by p, and P is defined as $P = l \cdot n(n+1)/2$ where l is the alignment length, and n is the number of sequences included in the subfamily alignment. Note that l varies depending on p, because subfamily alignment can contain columns comprising nulls only, which must be ignored. The threshold in this study is set to 2.4. The excluded branch list consists of branches specified by the child nodes of the ‘conserved‘-labeled internal ones.

2.2.2 Comparison-based Methods

Comparison-based methods detect unchanged columns or subfamily alignments

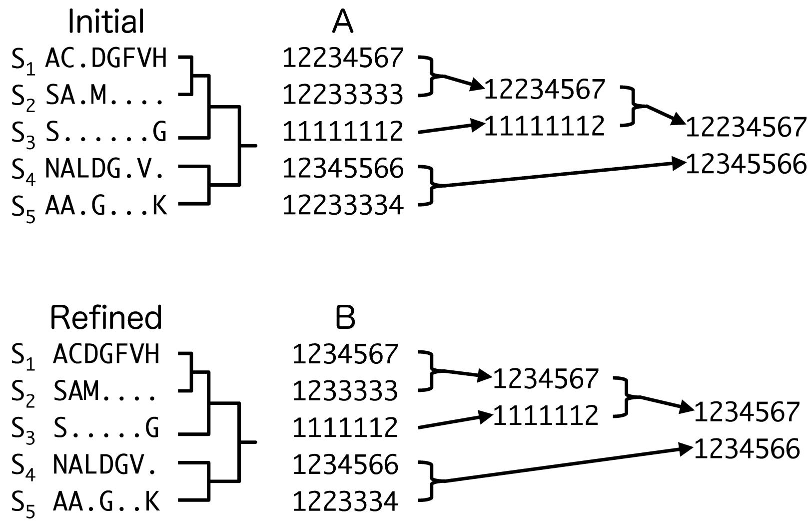

Fig. 1 Example of comparison-based grouping method. This figure exemplifies the procedure for detecting identically aligned subfamilies. The example contains two identically aligned subfamilies, one consisting of S_1, S_2, and S_3; and the other consisting of S_4 and S_5. '.' denotes a null. Note that the first columns of A and B are omitted for brevity.

between two MSAs before and after step 6 of the doubly nested randomized strategy. In these methods, unchanged columns or subfamily alignments during the iterative refinement are considered to be conserved. Therefore, the comparison-based methods are parameter-free unlike the conservation-based counterparts. Both comparison-based methods first convert MSAs into index matrices whose element represents the number of non-null residues on the row up to the relevant column (**Fig. 1**). To describe these methods explicitly, we introduce several symbols. Let A and B be index matrices which are constructed from the two MSAs to be compared. Here, the MSAs corresponding to A and B are denoted by M^A and M^B, respectively. $\mathbf{m}_i^A$ represents the i-th column of M^A, and $\mathbf{m}_j^B$ is similarly defined. Without loss of generality, we assume that both A and B have the same number of rows, n. l_A and l_B denote the respective lengths of M^A and M^B. A_k, $\mathbf{a}_i$, and $a_{k,i}$ mean k-th row of A, i-th column of A, and i-th element of A_k, respectively. B_k, $\mathbf{b}_j$, and $b_{k,j}$ are defined similarly. We set all the elements of the first column (column number 0) of A and B to zeros, that is, $a_{k,0} = b_{k,0} = 0$ for all $k \in \{1, \cdots, n\}$. Hence, matrices A and B have $l_A + 1$ and $l_B + 1$ columns, respectively. Column vectors $\mathbf{a}_i$ and $\mathbf{b}_j$ are partially ordered on the relation '$\leq$' as follows; if $a_{k,i} \leq b_{k,j}$ for all k, then $\mathbf{a}_i \leq \mathbf{b}_j$. Clearly, $\mathbf{a}_i = \mathbf{b}_j$ only if $a_{k,i} = b_{k,j}$ for all k. In practice, we can use slightly less stringent conditions for the inequality: if $R_i^A \leq R_j^B$, then $\mathbf{a}_i \leq \mathbf{b}_j$, where $R_i^A \equiv \max_{1 \leq k \leq n} a_{k,i}$. R_j^B is defined analogously. These relaxed conditions for the inequality play a key role for the grouping method.

The anchoring method detects alignment columns that are identical between M^A and M^B. Since identical (unchanged) columns can contain some nulls, the comparison-based anchoring method can detect null-containing columns as well as those consisting of residues only. $\mathbf{m}_i^A$ is regarded as identical to $\mathbf{m}_j^B$, if the following four conditions are simultaneously satisfied: (1) $\mathbf{m}_i^A \neq \phi$, (2) $\mathbf{m}_j^B \neq \phi$, (3) $\mathbf{a}_{i-1} = \mathbf{b}_{j-1}$, and (4) $\mathbf{a}_i = \mathbf{b}_j$, where ϕ denotes the empty column whose elements are all null characters. The conditions (1) and (2) can be replaced by $(1)'$ $\mathbf{a}_{i-1} \neq \mathbf{a}_i$ and $(2)'$ $\mathbf{b}_{j-1} \neq \mathbf{b}_j$ as easily verified by the definitions of A and B. Under the relaxed conditions, the column vectors $\mathbf{a}_i$ and $\mathbf{b}_j$ are replaced by the corresponding elements of R_i^A and R_j^B. The index pairs (i, j) that satisfy the condition (4) are efficiently found by the following simple algorithm:

(1) convert M^A and M^B into A and B
(2) $i \leftarrow 0$ and $j \leftarrow 0$
(3) do the following procedure until either $i = l_A$ or $j = l_B$
  (a) if $\mathbf{a}_i = \mathbf{b}_j$, then record the index pair (i, j)
  (b) if $\mathbf{a}_i \leq \mathbf{b}_j$, then $i \leftarrow i + 1$
  (c) if $\mathbf{b}_j \leq \mathbf{a}_i$, then $j \leftarrow j + 1$
  (d) if neither $\mathbf{a}_i \leq \mathbf{b}_j$ nor $\mathbf{b}_j \leq \mathbf{a}_i$, then $i \leftarrow i + 1$ and $j \leftarrow j + 1$
(4) detect unchanged anchor points based on the recorded index pairs

In this procedure, both $\mathbf{a}_i \leq \mathbf{b}_j$ and $\mathbf{b}_j \leq \mathbf{a}_i$ can hold simultaneously. The columns sandwiched between two records with consecutive column numbers are regarded as identical columns. For example, if the index pairs $(i-1, j-1), (i, j), \cdots, (i+k, j+k)$ are recorded, the consecutive columns $\mathbf{m}_i^A \cdots \mathbf{m}_{i+k}^A$ are identical to $\mathbf{m}_j^B \cdots \mathbf{m}_{j+k}^B$. In a special case, we regard M^A and M^B are identical to each other, and denote as $M^A = M^B$, if all non-empty columns in M^A are identical to some columns in M^B and *vice versa*. Note that this method could miss some identical columns when, for example, two columns are permutated. Specifically, two identical column pairs ($\mathbf{m}_i^A$, $\mathbf{m}_j^B$) and ($\mathbf{m}_p^A$, $\mathbf{m}_q^B$) can not be detected, if $i < p$ and $q < j$. Since such columns could be artifacts, this method suffices for our purpose.

The grouping method tries to extract identical subfamily alignments in the two MSAs. Given M^A, M^B, and a phylogenetic tree T, the comparison-based grouping method is nearly the same as the conservation-based counterpart:

(1) convert M^A and M^B into A and B
(2) label 'unknown' for all internal nodes of T and 'identical' for all leaves of T
(3) for each 'unknown'-labeled internal node p of T
  (a) if either child nodes of p is labeled 'non-identical', then label p 'non-identical'
  (b) else if both child nodes of p are labeled 'identical'; then label p 'identical' if $M_p^A = M_p^B$, or 'non-identical' otherwise
(4) compile excluded branch list

The M_p^A denotes the alignment induced from M^A consisting of all sequences included in the subtree of T that descend from p. M_p^B is defined analogously. For the examination of $M_p^A = M_p^B$, the comparison-based anchoring method under the relaxed conditions is used. At step 3a of the comparison-based anchoring method, the equality condition $\mathbf{a}_i = \mathbf{b}_j$ (with $O(n)$ computation) can be substituted with the equality conditions $R_i^{A_s} = R_j^{B_s}$ and $R_i^{A_t} = R_j^{B_t}$ (with $O(1)$ computation) where s and t are the child nodes of p, since subfamily alignments specified by s and t have already been found to be identical. The excluded branch list consists of branches specified by the child nodes of the 'identical'-labeled internal ones. Fig. 1 shows an example of the grouping method.

2.3 Group-to-group Sequence Alignment Algorithm Based on Maximal Expected Accuracy

In order to evaluate the effectiveness of a group-to-group sequence alignment algorithm based on MEA, we incorporate this algorithm into PRIME. The basic idea of MEA is to maximize the expected number of 'correctly' aligned residue pairs [29]. We adopt an approach similar to that used in ProbCons [18]. Using a simple three-state (match, insertion, and deletion) pair hidden Markov model, we first compute a posterior probability matrix for a pair of sequences from $\mathbf{m}^A$ and $\mathbf{m}^B$. Each element of the matrix is a posterior probability where i-th residue of a sequence in $\mathbf{m}^A$ is matched with j-th residue of a sequence in $\mathbf{m}^B$. Posterior probabilities are calculated using the standard forward and backward algorithms [30]. Then, a simple DP algorithm is employed to align the groups:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + S(\mathbf{m}_i^A, \mathbf{m}_j^B) \\ H_{i-1,j} \\ H_{i,j-1} \end{cases} \tag{1}$$

where $S(\mathbf{m}_i^A, \mathbf{m}_j^B) = \sum_{p \in M^A, q \in M^B} s(m_{p,i}^A, m_{q,j}^B)$. If both $m_{p,i}^A$ and $m_{q,j}^B$ are residues, $s(m_{p,i}^A, m_{q,j}^B)$ is a corresponding posterior probability, or zero otherwise.

3. Results

3.1 Benchmarks

We examined several variants of PRIME and other MSA programs shown in **Table 1**. The variants of PRIME differ from one another in the group-to-group sequence alignment algorithms, the use of the anchoring and the grouping methods, and methods in construction of the initial MSAs. *pcw* and *afn* mean the group-to-group sequence alignment algorithms with the piecewise linear and the affine gap costs, respectively. *ag* refers to the use of the anchoring and grouping methods; the PRIME variants with *ag* apply the steps 4 and 5 of the doubly nested randomized iterative strategy, while those without *ag* do not involve these steps. *mea* denotes the group-to-group sequence alignment algorithm based on MEA. Note that $\text{PRIME}_{pcw,mea}$ and $\text{PRIME}_{afn,mea}$ first calculate an initial MSA by the MEA-based algorithm, and then iteratively refine the MSA by the group-to-group sequence alignment with the piecewise linear gap cost and the affine gap cost, respectively. In the case of $\text{PRIME}_{mea,mea}$, both initial MSA calculation and iterative refinement are done by the MEA-based algorithm.

For evaluation, two benchmark tests were executed: BAliBASE version 3.0 [32)–34)] and PREFAB version 4 [10]. BAliBASE consists of alignments constructed by human expertise, categorized into five references according to the nature of sequences to be aligned. Reference 1 is further divided into two sub-

Table 1 List of evaluated programs.

Program	Version	Note
PRIME$_{pcw}$		BLOSUM62, $g(x) = \max\{-(x+9), -(0.5x+21.5)\}$
PRIME$_{pcw,ag}$		PRIME$_{pcw}$ with anchoring and grouping methods
PRIME$_{afn}$		BLOSUM62, $g(x) = -(x+9)$
PRIME$_{afn,ag}$		PRIME$_{afn}$ with anchoring and grouping methods
PRIME$_{pcw,mea}$		MEA-based initial MSA, refined with PRIME$_{pcw}$
PRIME$_{afn,mea}$		MEA-based initial MSA, refined with PRIME$_{pcw}$
PRIME$_{mea,mea}$		MEA-based initial MSA, refined with MEA-based
Prrn [11)]	3.4	`-b2 -mblosum62 -u1 -v9`
MAFFT [13)]	6.240	`--maxiterate 1000 --localpair` (L-INS-i)
ProbCons [18)]	1.12	default
T-Coffee [8)]	5.05	default
MUSCLE [10)]	3.6	default
DIALIGN-T [31)]	0.2.2	default
POA [7)]	2	`-do_global -do_progressive blosum80_trunc.mat`
ClustalW [9)]	1.83	default

references based on sequence identities. The contents of each reference are as follows. Reference 1 alignment consists of phylogenetically equidistant sequences of similar length. The average sequence identities of reference 1.1 are less than 20%, while those of reference 1.2, 20-40%. Alignments in reference 2 include a few distantly related sequences, in addition to closely related ones. In reference 3, each alignment comprises equidistant subfamilies. Sequences in alignments of references 4 and 5 contain long N/C terminal extensions, or long internal insertions, respectively. Except for reference 4, each reference consists of two test sets: full-length and trimmed sets. In this study, we used only the full-length sets.

PREFAB is composed of automatically generated alignments in contrast to BAliBASE. PREFAB contains three data sets: main, weighting, and long gap sets. The main set corresponds to the previous PREFAB version 3, which is not categorized. The weighting set involves alignments each of which consists of subfamilies with unbalanced numbers of members. Each alignment of the long gap set, a subset of the main set, contains one or more gaps whose lengths are more than 10. Note that each reference alignment of PREFAB is provided as a pairwise alignment of a pair of PDB sequences of known structures.

To evaluate alignment accuracy based on BAliBASE, we use sum-of-pairs and column scores [35)]. The sum-of-pairs score is defined as the proportion of correctly aligned residue pairs, while the column score represents the proportion of correctly aligned columns. For alignment evaluation of PREFAB, the quality score is used, which measures only two PDB sequences within each alignment. The quality score is the ratio of correctly aligned residue pairs of the reference pairwise alignment. The definition of these scores implies that quality, sum-of-pairs, and column scores have the same value if the reference alignment is pairwise.

3.2 Results of BAliBASE Benchmark Test

The average sum-of-pairs and column scores of BAliBASE are shown in **Table 2** and **Table 3**, respectively. The last columns of both tables represent the rank sums of the Friedman test. The program with the smallest rank sum means that the program consistently constructs the most accurate MSAs even if it does not achieve the largest average score. The Friedman test based on sum-of-pairs score indicates that the tested programs are classified into four groups according to the significance (P-value $< 5.0 \times 10^{-2}$) in their performances. The most accurate group consists of PRIME$_{pcw}$, PRIME$_{pcw,mea}$, PRIME$_{afn,mea}$, PRIME$_{mea,mea}$, MAFFT, and ProbCons. The second most accurate one consists of PRIME$_{pcw,ag}$, PRIME$_{afn}$, PRIME$_{afn,ag}$, Prrn, and T-Coffee. MUSCLE is classified into the third group. The accuracies of DIALIGN-T, POA, and ClustalW are comparable to each other and are significantly lower than that of MUSCLE.

The tendency of the Friedman test based on column score is slightly different; PRIME$_{pcw,ag}$ and PRIME$_{afn}$, in addition to MAFFT and ProbCons, are classified into the most accurate group, and the accuracy of PRIME$_{afn,ag}$ is comparable to that of Prrn, T-Coffee and MUSCLE. The Wilcoxon signed rank test based on sum-of-pairs score indicated that the accuracy difference between PRIME$_{pcw}$ and PRIME$_{pcw,mea}$ is significant (P-value: 1.3×10^{-6}), while the difference is insignificant in terms of column score (P-value: 0.10). The same tendency is also reproduced in comparison between PRIME$_{afn}$ and PRIME$_{afn,mea}$. The Wilcoxon signed rank test indicated that the accuracy difference between PRIME$_{pcw,mea}$ and PRIME$_{mea,mea}$ is statistically insignificant (respective P-values: 0.63 and 0.89), while PRIME$_{mea,mea}$ is significantly more accurate than PRIME$_{afn,mea}$ in terms of both sum-of-pairs and column scores (respective P-values: 1.5×10^{-2} and 3.6×10^{-3}).

Table 2 Average sum-of-pairs scores of BAliBASE. Each column shows average sum-of-pairs scores using all alignments of each reference of BAliBASE. Overall and Ranksum columns show the average sum-of-pairs scores and the rank sum of the Friedman test using sum-of-pairs scores on all alignment of each reference, respectively. A smaller rank sum means better accuracy.

	Ref. 1.1	Ref. 1.2	Ref. 2	Ref. 3	Ref. 4	Ref. 5	Overall	Ranksum
PRIME_{pcw}	0.638	0.932	0.917	0.858	0.906	0.885	0.858	1354
$\mathrm{PRIME}_{pcw,ag}$	0.633	0.929	0.917	0.844	0.913	0.876	0.855	1501
PRIME_{afn}	0.627	0.930	0.899	0.845	0.883	0.864	0.844	1518
$\mathrm{PRIME}_{afn,ag}$	0.620	0.929	0.898	0.823	0.869	0.859	0.836	1682
$\mathrm{PRIME}_{pcw,mea}$	0.641	0.937	0.925	0.856	0.923	0.890	0.865	1066
$\mathrm{PRIME}_{afn,mea}$	0.631	0.934	0.902	0.851	0.882	0.875	0.847	1266
$\mathrm{PRIME}_{mea,mea}$	0.646	0.941	0.880	0.829	0.855	0.888	0.839	1061
Prrn	0.572	0.923	0.902	0.822	0.860	0.822	0.821	1688
MAFFT	0.671	0.938	0.923	0.852	0.918	0.891	0.868	1052
ProbCons	0.669	0.943	0.914	0.847	0.898	0.882	0.861	1072
T-Coffee	0.578	0.924	0.910	0.789	0.860	0.847	0.821	1865
MUSCLE	0.590	0.918	0.886	0.803	0.866	0.843	0.821	1976
DIALIGN-T	0.484	0.883	0.855	0.737	0.795	0.781	0.760	2637
POA	0.474	0.857	0.857	0.733	0.805	0.754	0.753	2748
ClustalW	0.497	0.864	0.848	0.722	0.786	0.713	0.748	2592

Table 3 Average column scores of BAliBASE. Each column shows average column scores using all alignments of each reference of BAliBASE. Overall and Ranksum columns show the average column scores and the rank sum of the Friedman test using column scores on all alignment of each reference, respectively. A smaller rank sum means better accuracy.

	Ref. 1.1	Ref. 1.2	Ref. 2	Ref. 3	Ref. 4	Ref. 5	Overall	Ranksum
PRIME_{pcw}	0.412	0.834	0.441	0.557	0.579	0.526	0.568	1386
$\mathrm{PRIME}_{pcw,ag}$	0.406	0.829	0.435	0.511	0.588	0.509	0.560	1524
PRIME_{afn}	0.367	0.832	0.388	0.529	0.514	0.477	0.528	1534
$\mathrm{PRIME}_{afn,ag}$	0.366	0.830	0.381	0.493	0.447	0.476	0.506	1666
$\mathrm{PRIME}_{pcw,mea}$	0.416	0.841	0.439	0.547	0.603	0.521	0.574	1266
$\mathrm{PRIME}_{afn,mea}$	0.378	0.841	0.368	0.547	0.502	0.491	0.529	1440
$\mathrm{PRIME}_{mea,mea}$	0.404	0.850	0.374	0.478	0.511	0.544	0.532	1146
Prrn	0.335	0.791	0.405	0.483	0.487	0.421	0.501	1669
MAFFT	0.449	0.839	0.442	0.561	0.609	0.518	0.583	1178
ProbCons	0.414	0.858	0.393	0.549	0.540	0.521	0.554	1169
T-Coffee	0.307	0.813	0.369	0.361	0.492	0.457	0.480	1879
MUSCLE	0.353	0.804	0.337	0.382	0.481	0.439	0.480	1958
DIALIGN-T	0.251	0.703	0.278	0.346	0.426	0.397	0.410	2402
POA	0.224	0.678	0.265	0.343	0.412	0.323	0.389	2568
ClustalW	0.221	0.707	0.219	0.271	0.404	0.237	0.368	2297

3.3 Results of PREFAB Benchmark Test

The average quality scores of the three sets of PREFAB are shown in **Table 4**. Compared with those of BAliBASE, the results of the Friedman test of the main set are somewhat different. The most accurate group consists of $\mathrm{PRIME}_{pcw,mea}$, $\mathrm{PRIME}_{afn,mea}$, $\mathrm{PRIME}_{mea,mea}$, and MAFFT. In the second accurate group, PRIME_{pcw}, PRIME_{afn}, Prrn, and ProbCons are included. $\mathrm{PRIME}_{pcw,ag}$ and

Table 4 Average quality scores of PREFAB. Each QS and Ranksum columns show the average quality scores and the rank sum of the Friedman test using quality scores on all alignments of each reference of PREFAB, respectively. A smaller rank sum means better accuracy.

	Main		Weighting		Long gap	
	QS	Ranksum	QS	Ranksum	QS	Ranksum
PRIME$_{pcw}$	0.719	11879	0.652	762	0.657	2114
PRIME$_{pcw,ag}$	0.712	12893	0.650	771	0.642	2359
PRIME$_{afn}$	0.718	11816	0.637	808	0.651	2187
PRIME$_{afn,ag}$	0.711	12882	0.634	806	0.645	2297
PRIME$_{pcw,mea}$	0.724	10830	0.655	723	0.658	1976
PRIME$_{afn,mea}$	0.722	10864	0.641	768	0.655	1988
PRIME$_{mea,mea}$	0.694	9972	0.622	606	0.606	1755
Prrn	0.721	11662	0.624	806	0.652	2134
MAFFT	0.723	11020	0.637	780	0.662	1978
ProbCons	0.716	11770	0.658	646	0.648	1978
T-Coffee	0.673	14922	0.620	790	0.605	2713
MUSCLE	0.679	14489	0.613	830	0.598	2803
DIALIGN-T	0.609	18856	0.586	980	0.520	3706
POA	0.603	19838	0.554	1099	0.513	3806
ClustalW	0.617	17548	0.603	824	0.519	3526

PRIME$_{afn,ag}$ are classified into the third group. The fourth one is comprised of T-Coffee and MUSCLE. The fifth one consists of ClustalW only. DIALIGN-T and POA are included in the rest one. The Wilcoxon signed rank test of the main set showed that PRIME$_{pcw,mea}$ is significantly more accurate than that of PRIME$_{pcw}$ (P-value: 3.7×10^{-6}). Similarly, the accuracies of PRIME$_{afn,mea}$ are statistically better than those of PRIME$_{afn}$ (P-value: 1.3×10^{-6}).

In the case of the weighting set, all programs except DIALIGN-T and POA are comparable to each other. The Friedman test of the long gap set divides the tested programs into four groups. The most accurate group is composed of PRIME$_{pcw}$, PRIME$_{afn}$, PRIME$_{pcw,mea}$, PRIME$_{afn,mea}$, PRIME$_{mea,mea}$, Prrn, MAFFT, and ProbCons. The second most accurate group includes PRIME$_{pcw,ag}$, PRIME$_{afn,ag}$, and T-Coffee. The third one consists only of MUSCLE. The rest of programs, DIALIGN-T, POA, and ClustalW showed comparable performances to each other, consisting of the fourth group.

3.4 Computation Time

The computation time of each program for executing the benchmarks is compiled in **Table 5**. The computer we used is Pentium3 933 MHz with 1 GB memory, running on RedHat Linux 7.3. As expected, the computation of PRIME$_{pcw,ag}$ and PRIME$_{afn,ag}$ are reduced more than 60%, compared with that of PRIME$_{pcw}$ and PRIME$_{afn}$. Note that variants of PRIME with MEA-based algorithm are rather slow, partly because we do not currently use lookup table and interpolation techniques for calculating posterior probabilities; implementing these techniques would improve the speed to some extent.

Table 5 Computation time. BAliBASE column shows the total times (sec.) spent for construction of all alignments of each reference by each program, whereas PREFAB column shows those used for whole alignments of the main and weighting sets only.

	BAliBASE	PREFAB
PRIME$_{pcw}$	1.1×10^6	5.4×10^5
PRIME$_{pcw,ag}$	3.6×10^5	2.1×10^5
PRIME$_{afn}$	4.1×10^5	3.6×10^5
PRIME$_{afn,ag}$	1.4×10^5	1.3×10^5
PRIME$_{pcw,mea}$	1.1×10^6	1.2×10^6
PRIME$_{afn,mea}$	5.0×10^5	1.0×10^6
PRIME$_{mea,mea}$	7.9×10^5	1.9×10^6
Prrn	6.7×10^5	1.9×10^5
MAFFT	1.7×10^4	2.3×10^4
ProbCons	1.4×10^5	5.7×10^5
T-Coffee	1.7×10^5	8.7×10^5
MUSCLE	5.8×10^3	1.4×10^4
DIALIGN-T	2.0×10^4	1.2×10^5
POA	7.8×10^3	2.7×10^4
ClustalW	6.0×10^3	2.8×10^4

4. Discussion

Compared with other leading MSA programs, PRIME adopts a computationally intensive group-to-group sequence alignment algorithm. Therefore, some heuristics for reducing the computation with a minimal loss in accuracy is highly desired. Accordingly, we newly introduced anchoring and grouping methods into PRIME. As a result of BAliBASE and PREFAB benchmark tests, PRIME$_{pcw,ag}$ and PRIME$_{afn,ag}$ are proven to be much faster than PRIME$_{pcw}$ and PRIME$_{afn}$, while average alignment accuracy measures decrease by at most 2%. However, the choice of appropriate parameters is a difficult problem, because there is a tradeoff between speed and accuracy. In this study, we selected the parameters of the anchoring and grouping methods based on the observation that the average sum-of-pairs and column scores of the previous BAliBASE version 2.0 decreased by less than 1%. Similarly, there are several choices of how to use anchor points. Although the entire span of anchor points is currently fixed, it is possible to fix only an internal region of these points, or each region is used only for dividing the DP matrix into pieces. This choice could also provide another tradeoff.

In this study, we also evaluated the effectiveness of the MEA-based algorithm. The alignment accuracy of MEA-based algorithm is robust; although the average sum-of-pairs and column scores of PRIME$_{mea,mea}$ is relatively smaller than those of the score-based variants like PRIME$_{pcw}$, the rank sum of PRIME$_{mea,mea}$ often exceeds those of the other variants. However, the computation with PRIME$_{mea,mea}$ is expensive especially when many sequences are aligned, because the computational complexity of calculating substitution cost is proportional to the product of the numbers of sequences in the groups. In addition, the accuracies of PRIME$_{pcw,mea}$ is comparable to those of PRIME$_{mea,mea}$. PRIME$_{pcw}$ and PRIME$_{pcw,mea}$ differ from each other only by the way of construction of initial MSAs; the former relies on a score-based algorithm, while the latter uses an MEA-based algorithm. In fact, we have observed that pairwise alignments constructed by an MEA-based algorithm are generally more accurate than those obtained by the corresponding score-based algorithm (data not shown). MEA-based initial MSA probably contributes to improvement in accuracy even after similarly performed iterative refinement. Therefore, when not so many sequences are aligned, PRIME$_{pcw,mea}$ may replace PRIME$_{pcw}$ to construct the most accurate MSA.

Acknowledgments The authors thank Dr Kiyoshi Asai for helpful discussions. The authors also thank Mr Akinlade Damilola Abiodun for proofreading this manuscript. SY was supported by Waseda University Grant for Special Research Projects (2005A-953 and 2006B-294). OG was partially supported by KAKENHI (Grant-in-Aid for Scientific Research) on Priority Areas "Comparative Genomics" from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1) Edgar, R.C. and Batzoglou, S.: Multiple sequence alignment, *Curr Opin Struct Biol*, Vol.16, No.3, pp.368–373 (2006).
2) Gotoh, O.: Multiple sequence alignment: algorithms and applications, *Adv Biophys*, Vol.36, pp.159–206 (1999).
3) Notredame, C.: Recent progress in multiple sequence alignment: A survey, *Pharmacogenomics*, Vol.3, No.1, pp.131–144 (2002).
4) Notredame, C.: Recent evolutions of multiple sequence alignment algorithms, *PLoS Comput Biol*, Vol.3, No.8, p.e123 (2007).
5) Gotoh, O., Yamada, S. and Yada, T.: Multiple sequence alignment, *Handbook of computational molecular biology*, Aluru, S. (Ed.), Chapman & Hall/CRC, pp.3–1–3–36 (2005).
6) Jiang, T. and Wang, L.: Algorithmic methods for multiple sequence alignment, *Current topics in computational molecular biology*, Jiang, T., Xu, Y. and Zhang, M.Q. (Eds.), Tsinghua University Press (2002).
7) Grasso, C. and Lee, C.: Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems, *Bioinformatics*, Vol.20, No.10, pp.1546–1556 (2004).
8) Notredame, C., Higgins, D.G. and Heringa, J.: T-Coffee: A novel method for fast and accurate multiple sequence alignment, *J Mol Biol*, Vol.302, No.1, pp.205–217 (2000).
9) Thompson, J.D., Higgins, D.G. and Gibson, T.J.: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic Acids Res*, Vol.22, No.22, pp.4673–4680 (1994).
10) Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Res*, Vol.32, No.5, pp.1792–1797 (2004).
11) Gotoh, O.: Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments, *J Mol Biol*, Vol.264, No.4, pp.823–838 (1996).
12) Katoh, K., Misawa, K., Kuma, K. and Miyata, T.: MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform, *Nucleic Acids Res*, Vol.30, No.14, pp.3059–3066 (2002).
13) Katoh, K., Kuma, K., Toh, H. and Miyata, T.: MAFFT version 5: improvement in accuracy of multiple sequence alignment, *Nucleic Acids Res*, Vol.33, No.2, pp.511–518 (2005).
14) Yamada, S., Gotoh, O. and Yamana, Y.: Improvement in accuracy of multiple sequence alignment using novel group-to-group sequence alignment algorithm with piecewise linear gap cost, *BMC Bioinformatics*, Vol.7, p.524 (2006).
15) Edgar, R.C.: MUSCLE: A multiple sequence alignment method with reduced time

and space complexity, *BMC Bioinformatics*, Vol.5, p.113 (2004).
16) Lassmann, T. and Sonnhammer, E.: Automatic extraction of reliable regions from multiple sequence alignments, *BMC Bioinformatics*, Vol.8, p.S9 (2007).
17) Thompson, J.D., Thierry, J.C. and Poch, O.: RASCAL: rapid scanning and correction of multiple sequence alignments, *Bioinformatics*, Vol.19, No.9, pp.1155–1161 (2003).
18) Do, C.B., Mahabhashyam, M.S.P., Brudno, M. and Batzoglou, S.: ProbCons: Probabilistic consistency-based multiple sequence alignment, *Genome Res*, Vol.15, No.2, pp.330–340 (2005).
19) Wallace, I., O'Sullivan, O., Higgins, D.G. and Notredame, C.: M-Coffee: combining multiple sequence alignment methods with T-Coffee, *Nucleic Acids Res*, Vol.34, No.6, pp.1692–1699 (2006).
20) O'Sullivan, O., Suhre, K., Abergel, C., Higgins, D.G. and Notredame, C.: 3DCoffee: combining protein sequences and structures within multiple sequence alignments, *J Mol Biol*, Vol.340, No.2, pp.385–395 (2004).
21) Pei, J. and Grishin, N.: MUMMALS: multiple sequence alignment improved by using hidden Markov models with local structural information, *Nucleic Acids Res*, Vol.34, No.16, pp.4364–4374 (2006).
22) Zhou, H. and Zhou, Y.: SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures, *Bioinformatics*, Vol.21, No.18, pp.3615–3621 (2005).
23) DeCaprio, D., Vinson, J., Pearson, M., Montgomery, P., Doherty, M. and Galagan, J.: Conrad: gene prediction using conditional random fields, *Genome Res*, Vol.17, No.9, pp.1389–1398 (2007).
24) Kiryu, H., Kin, T. and Asai, K.: Robust prediction of consensus secondary structures using averaged base pairing probability matrices, *Bioinformatics*, Vol.23, No.4, pp.434–441 (2007).
25) Gotoh, O.: Optimal sequence alignment allowing for long gaps, *Bull Math Biol*, Vol.52, No.3, pp.359–373 (1990).
26) Gotoh, O.: An improved algorithm for matching biological sequences, *J Mol Biol*, Vol.162, No.3, pp.705–708 (1982).
27) Gotoh, O.: Optimal alignment between groups of sequences and its application to multiple sequence alignment, *Comput Appl Biosci*, Vol.9, No.3, pp.361–370 (1993).
28) Gotoh, O.: Further improvement in methods of group-to-group sequence alignment with generalized profile operations, *Comput Appl Biosci*, Vol.10, No.4, pp.379–387 (1994).
29) Miyazawa, S.: A reliable sequence alignment method based on probabilities of residue correspondences, *Protein Eng*, Vol.8, No.10, pp.999–1009 (1995).
30) Durin, R., Eddy, S.R., Krogh, A. and Mitchison, G.: *Biological sequence analysis*, Cambridge University Press (1998).
31) Subramanian, A.R., Weyer-Menkhoff, J., Kaufmann, M. and Morgenstern, B.: DIALIGN-T: An improved algorithm for segment-based multiple sequence alignment, *BMC Bioinformatics*, Vol.6, p.66 (2005).
32) Thompson, J.D., Plewniak, F. and Poch, O.: BAliBASE: A benchmark alignment database for the evaluation of multiple alignment programs, *Bioinformatics*, Vol.15, No.1, pp.87–88 (1999).
33) Bahr, A., Thompson, J., Thierry, J. and Poch, O.: BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations, *Nucleic Acids Res*, Vol.29, No.1, pp.323–326 (2001).
34) Thompson, J.D., Koehl, P., Ripp, R. and Poch, O.: BAliBASE 3.0: Latest developments of the multiple sequence alignment benchmark, *Proteins*, Vol.61, No.1, pp.127–136 (2005).
35) Thompson, J.D., Plewniak, F. and Poch, O.: A comprehensive comparison of multiple sequence alignment programs, *Nucleic Acids Res*, Vol.27, No.13, pp.2682–2690 (1999).

(Received November 19, 2007)
(Revised January 22, 2008)
(Accepted January 23, 2008)
(Released November 28, 2008)

(Communicated by *Tetsuo Shibuya*)

Shinsuke Yamada received the B. of Information and Computer Science (2002), M. of Information and Computer Science (2004), and Dr.Eng. (2008) degrees in Computer Science from Waseda University. Since 2008, he has been an AIST research staff in Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology. His current research interests include bioinformatics, and high performance computing. He is a member of ACM, ISCB (The International Society for Computational Biology) and JSBi (Japanese Society for Bioinformatics).

Osamu Gotoh received the B.S. (1970), M.S. (1972), and Ph.D. (1979) degrees in Physics from Faculty of Science, University of Tokyo. From 1976 to 2001, he was a researcher in Department of Biochemistry, Saitama Cancer Center Research Institute. From 2001 to 2003, he was a Team Leader at Computational Biology Research Center, AIST. Since 2003, he has been a professor of Department of Informatics, Kyoto University. His current research interests are focused on bioinformatics, particularly on comparative genome sequence analyses. He is a member of JSBi (Japanese Society for Bioinformatics), ISCB (The International Society for Computational Biology), BPSJ (The Biophysical Society of Japan), and JBS (The Japanese Biochemical Society).

Hayato Yamana received the B.S. (1987), M.S. (1989), and Dr.Eng. (1993) degrees in Computer Science from Waseda University. From 1993 to 2000, he was a researcher in Electorotechnical Laboratory, AIST, MITI. From 2000 to 2005, he was an associate professor of Waseda University. Since 2005, he has been a professor of Waseda University. His current research interests include, bioinformatics, data mining, distributed computing, and information retrieval. He is a member of IPSJ, IEICE, ACM, and IEEE.