BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 13 2007, pages 1588–1598
doi:10.1093/bioinformatics/btm146

Structural bioinformatics

Murlet: a practical multiple alignment tool for structural RNA sequences

Hisanori Kiryu[1,2,*], Yasuo Tabei[3], Taishin Kin[1] and Kiyoshi Asai[1,3]

[1]Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology (AIST), 2-42 Aomi, Koto-ku, Tokyo 135-0064, [2]Graduate School of Information Sciences, Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192 and [3]Department of Computational Biology, Faculty of Frontier Science, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8561, Japan

Received on January 24, 2007; revised on March 19, 2007; accepted on April 10, 2007

Advance Access publication April 25, 2007

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Structural RNA genes exhibit unique evolutionary patterns that are designed to conserve their secondary structures; these patterns should be taken into account while constructing accurate multiple alignments of RNA genes. The Sankoff algorithm is a natural alignment algorithm that includes the effect of base-pair covariation in the alignment model. However, the extremely high computational cost of the Sankoff algorithm precludes its application to most RNA sequences.

Results: We propose an efficient algorithm for the multiple alignment of structural RNA sequences. Our algorithm is a variant of the Sankoff algorithm, and it uses an efficient scoring system that reduces the time and space requirements considerably without compromising on the alignment quality. First, our algorithm computes the match probability matrix that measures the alignability of each position pair between sequences as well as the base pairing probability matrix for each sequence. These probabilities are then combined to score the alignment using the Sankoff algorithm. By itself, our algorithm does not predict the consensus secondary structure of the alignment but uses external programs for the prediction. We demonstrate that both the alignment quality and the accuracy of the consensus secondary structure prediction from our alignment are the highest among the other programs examined. We also demonstrate that our algorithm can align relatively long RNA sequences such as the eukaryotic-type signal recognition particle RNA that is ~300 nt in length; multiple alignment of such sequences has not been possible by using other Sankoff-based algorithms. The algorithm is implemented in the software named 'Murlet'.

Availability: The C++ source code of the Murlet software and the test dataset used in this study are available at http://www.ncrna.org/papers/Murlet/

Contact: kiryu-h@aist.go.jp

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Recent studies have revealed that a substantial number of RNA transcripts do not code protein sequences in higher eukaryotic cells (Carninci *et al*., 2005; Dunham *et al*., 2004; Okazaki *et al*., 2002), and the question of whether such transcripts have any functional roles in cellular processes has attracted considerable interest. The existence of conserved secondary structures among phylogenetic relatives indicates the functional importance of such transcripts; therefore, it would be extremely interesting to detect conserved secondary structures from multiple alignments of genomic sequences. The evolutionary process of a structural RNA gene has a unique characteristic that the substitutions of distant bases are correlated in order to conserve their stem structures; hence, multiple alignment methods should account for such substitution patterns to enable accurate detection of the conserved structures. The Sankoff algorithm (Sankoff, 1985) is an alignment algorithm that naturally includes the effect of base-pair covariation in the alignment model. However, it is not practical to use the original version of the Sankoff algorithm due to its prohibitive computational cost. Hence, there have been intensive studies that have investigated practical variations of the Sankoff algorithm in recent years (Dowell and Eddy, 2006; Gorodkin *et al*., 1997; Havgaard *et al*., 2005; Hofacker *et al*., 2004; Holmes, 2005; Mathews and Turner, 2002; Uzilov *et al*., 2006). The algorithms proposed in these studies can be broadly categorized into two groups depending on how the secondary structures are scored in the algorithm.

In the first group, the algorithms score the structures using the free energy parameters collected by the Turner group (Mathews *et al*., 1999). These algorithms have the advantage of relatively accurate structure predictions. However, it is difficult for these algorithms to combine the structure energy with the homology information consistently. This group comprises the pairwise alignment programs Dynalign (Mathews and Turner, 2002; Uzilov *et al*., 2006) and Foldalign (Havgaard *et al*., 2005), and the multiple alignment program PMMulti (Hofacker *et al*., 2004).

In the second group, the algorithms score the structures as a part of the probabilistic model called the pair stochastic context-free grammar (PSCFG). These algorithms have the advantage that the parameters that score both the alignments and structures are determined in a unified manner. However, these algorithms have a potential disadvantage that the

*To whom correspondence should be addressed.

accuracies of the structure models might be only modest when compared with those in the first group; this is due to the limitations of PSCFG. The second group comprises the pairwise alignment program Consan (Dowell and Eddy, 2006) and the multiple alignment program Stemloc (Holmes, 2005).

These algorithms provide a variety of methods to reduce the enormous computation costs. Dynalign restricts the dynamic programming (DP) region to a narrow band such that only similar positions of sequences are compared. Foldalign and PMMulti limit the difference of subsequence lengths that are compared with each other. Stemloc implements a general method for combining the constraints in the structure space and those in the alignment space, which are computed using a Waterman–Eggert style suboptimal alignment algorithm (Waterman and Eggert, 1987). Consan constrains the DP region by anchoring the points in the DP matrix that have very high posterior probabilities of alignment according to the computations by the pair hidden Markov model (PHMM).

However, the computational costs remain relatively high despite these approximations, and it is impractical to use these programs for aligning sequences that are longer than 200 bases (as shown in Fig. 4). Therefore, several studies have sought for algorithms that can circumvent the Sankoff algorithm for fast computation of common secondary structures. For example, the SCARNA program (Tabei *et al.*, 2006) aligns a pair of stem candidate sets that are extracted from the base pairing probability matrices of sequences. The RNAcast program predicts secondary structures for unaligned sequences that have a common topology or consensus shape (Reeder and Giegerich, 2005), and the RNAmine algorithm (Hamada *et al.*, 2006) provides comprehensive list of the frequent stem motif patterns from unaligned sequences.

In this article, we propose a practical method based on the Sankoff algorithm for aligning multiple RNA sequences. We show that both the alignment quality and the accuracy of the consensus structure prediction from our alignment are the highest among the existing alignment softwares. Additionally, we show that our algorithm can align relatively long RNA sequences that have not been computable by other Sankoff-based multiple alignment algorithms. The algorithm is implemented in the software ‘Murlet’.

2 SYSTEMS AND METHODS

2.1 The model

First, we describe our algorithm for a pairwise sequence alignment. Our heuristic score system for the Sankoff algorithm is derived on the basis of two principles.

The first principle is the extensive preprocessing before applying the Sankoff algorithm. In general, the alignment of structural RNA sequences requires simultaneous consideration of complex information, such as base substitution score, gap insertion cost, stacking energy and various loop energies. If all these elements are included in the Sankoff model, the computation time would become unmanageably slow. Therefore, we used the match probability $p^{(a)}$ and the base-pairing probability $p^{(b)}$ to score the alignments and structures.

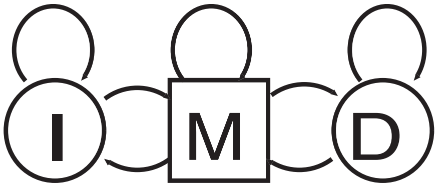

Fig. 1. The architecture of the PHMM used to calculate the match probabilities $p^{(a)}$. M indicates the match state, and I and D indicate the insertion and deletion states, respectively.

The match probability $p^{(a)}(i, j)$ is the posterior probability that sequence positions i and j will be matched in an alignment. The match probability is calculated by using the standard PHMM (Durbin *et al.*, 1998), as shown in Figure 1.

$$\begin{aligned} p^{(a)}(i,j) &= \sum_{\tau \in \Omega(i,j)} p^{(a)}(\tau | x, y) \\ p^{(a)}(\tau | x, y) &= \frac{1}{Z(x,y)} p^{(a)}(\tau, x, y) \\ Z(x,y) &= \sum_{\xi} p^{(a)}(\xi, x, y) \end{aligned}$$

where, $p^{(a)}(\tau|x,y)$ is the posterior probability of an alignment path τ given sequences x and y. $p^{(a)}(\tau, x, y)$ is the joint probability of generating the alignment path τ, and it is estimated by the product of the transition and emission probabilities of the PHMM model. $\Omega(i, j)$ is the set of alignment paths that pass through the point (i, j) in the DP matrix as the match state. The sum of the denominator in the second line is across all the possible alignment paths. $p^{(a)}(i, j)$ is calculated using the forward and backward algorithms. The computation of $p^{(a)}$ requires $\mathcal{O}(L^2)$ time and $\mathcal{O}(L^2)$ memory.

The base-pairing probability $p^{(b)}(i,k)$ is the probability that the pair positions i and k in the sequence forms a base pair, and it is calculated by using the McCaskill algorithm (McCaskill, 1990).

$$\begin{aligned} p^{(b)}(i,k) &= \sum_{\sigma \in \mathcal{S}(i,k)} p^{(b)}(\sigma | x) \\ p^{(b)}(\sigma | x) &= \frac{1}{Z(x)} \exp\left(-\frac{E(\sigma, x)}{RT}\right) \\ Z(x) &= \sum_{\zeta} \exp\left(-\frac{E(\zeta, x)}{RT}\right) \end{aligned}$$

where σ denotes a secondary structure candidate of sequence x; $E(\sigma, x)$, the secondary structure free energy that is computed using the energy parameters collected by the Turner group (Mathews *et al.*, 1999); R, the gas constant; T, the temperature; $Z(x)$, the partition function and $\mathcal{S}(i,k)$, the set of all the secondary structures that have a base pair between i and k. We let $q^{(b)}(i)$ denote the loop probability at position i.

$$q^{(b)}(i) = 1 - \sum_{k<i} p^{(b)}(k,i) - \sum_{i<k} p^{(b)}(i,k) \qquad (1)$$

The computation of $p^{(b)}$ requires $\mathcal{O}(L^3)$ time and $\mathcal{O}(L^2)$ memory.

Both $p^{(a)}$ and $p^{(b)}$ can be computed by much faster algorithms than the Sankoff algorithm and compactly represent complex information such as sequence homology and structure contexts. This enables us to keep the Sankoff model very simple. Since the quantities $p^{(a)}$ and $p^{(b)}$ do not include the effects of base-pair substitution, we also apply the base-pair substitution matrix $s(i, j, k, l)$ to score the events of the base-pair substitution.

The second principle is the maximal expected accuracy (MEA) principle. Recent studies have shown that the accuracy of the sequence alignment and the secondary structure predictions based on the principle of the maximization of expected accuracy (Holmes and Durbin, 1998; Miyazawa, 1995) perform better than those made by the conventional maximal likelihood algorithms (Do *et al.*, 2006; Knudsen and Hein, 2003; Pedersen *et al.*, 2006). A simple application of the MEA principle to the Sankoff algorithm that has a probabilistic scoring model such as PSCFG can be defined by the following expected accuracy z:

$$z = \sum_{(i,j)} p(x_i \sim y_j | x, y) + \sum_{((i,k),(j,l))} p((x_i, x_k) \sim (y_j, y_l) | x, y)$$

where $p(x_i \sim y_j | x, y)$ is the posterior probability that the bases x_i and y_j are aligned as unpaired bases, and $p((x_i, x_k) \sim (y_j, y_l) | x, y)$ is the posterior probability that the base pairs (x_i, x_k) and (y_j, y_l) aligned with each other as stem forming pairs. The sum of the first term is taken across the unpaired base matches in the candidate alignment and that of the second is taken across the base-pair matches. However, the computation of such function is demanding because the corresponding probabilistic model requires a large number of states to express the complex homology and structure information as described earlier.

Therefore, we have adopted an alternative heuristic score function (Equations 2–4) that is formally similar to the formula given earlier but can be computed more easily.

To provide a mathematical definition of our algorithm, we consider a consensus structure annotation $\mathcal{S}$ for each pairwise alignment $\mathcal{A}$ of length L, which consists of sequences x and y of lengths L_x and L_y, respectively.

$$\begin{aligned}
&\mathcal{S} = \mathcal{S}_{\mathcal{A}} = \{\mathcal{L}, \mathcal{P}\} \\
&\mathcal{L} = \{I \in \mathcal{C} | \text{column } I \text{ does not form any base pair}\} \\
&\mathcal{P} = \{(I, J) \in \mathcal{PC} | \text{columns } (I, J) \text{ form a base pair}\}
\end{aligned}$$

where the match column set $\mathcal{C}$ is the set of alignment columns without gap characters, and $\mathcal{PC} = \{(I, J) \in \mathcal{C} \times \mathcal{C} | 1 \le I < J \le L\}$ is the set of pairs of match columns. We consider only those cases where all the base pairs are formed between the match columns. We also ignore pseudoknotted structures. We assign a score e_L to each loop column $I \in \mathcal{L}$ and a score e_S to each column pair $(I, J) \in \mathcal{P}$.

$$e_L(i_I, j_I) = \gamma_L p^{(a)}(i_I, j_I) q^{(b)}(i_I) q^{(b)}(j_I) \tag{2}$$

$$\begin{aligned}
e_S(i_I, j_I, i_J, j_J) = {} & \gamma_S p^{(a)}(i_I, j_I) p^{(a)}(i_J, j_J) \\
& \times p^{(b)}(i_I, i_J) p^{(b)}(j_I, j_J) \\
& \times \exp(s(i_I, j_I, i_J, j_J))
\end{aligned} \tag{3}$$

where i_I and j_I represent the sequence positions of sequences x and y, respectively, aligned at column I. $s(i_I, j_I, i_J, j_J)$ denotes an element of the base pair substitution matrix. γ_L and γ_S are constant coefficients.

For each alignment $\mathcal{A}$ and its consensus structure candidate $\mathcal{S}$, our heuristic alignment score $z = z(\mathcal{A}, \mathcal{S})$ is defined as the sum of the loop match scores e_L and the base pair match scores e_S.

$$z = \sum_{I \in \mathcal{L}} e_L(i_I, j_I) + \sum_{(I,\ J) \in \mathcal{P}} e_S(i_I, j_I,\ i_J, j_J) \tag{4}$$

The alignment result $(\mathcal{A}_{\max}, \mathcal{S}_{\max})$ is obtained by taking the maximum $(z_{\max})$ of the score among all the alignments and structures.

To compute the maximum of $z(\mathcal{A}, \mathcal{S})$, we have adopted the following variant of the Sankoff algorithm.

$$M_i, j, k, l = \max \begin{cases} M_i + 1, j + 1, k - 1, l - 1 + e_S(i, j, k, l) \\ M_i + 1, j + 1, k, l + e_L(i, j) \\ M_i, j, k - 1, l - 1 + e_L(k, l) \\ M_i + 1, j, k, l \\ M_i, j + 1, k, l \\ M_i, j, k - 1, l \\ M_i, j, k, l - 1 \\ M_i, j, u, v + M_{u+1,\, v+1,\, k,\, l}\ for\ i < u < k,\ j < v < l \end{cases} \tag{5}$$

After the DP computation, the maximum of the score is obtained by $z_{\max} = M_{1,\,1,\,L_x,\,L_y}$. The computation of Equation (5) requires $\mathcal{O}(L^6)$ time and $\mathcal{O}(L^4)$ memory.

Note that the alignment result is defined in terms of the score function $z(\mathcal{A}, \mathcal{S})$ that depends only on the alignment $\mathcal{A}$ and the structure $\mathcal{S}$, and is independent of the grammar, or transition rules, of the parsing algorithm (Equation 5). We may use an arbitrary grammar to compute the alignment result provided that the grammar can parse all the alignments and structures and that it does not modify the score system. The latter condition implies that the model cannot have any transition scores and that the left and right emission scores have to be identical. The independence from a particular grammar also indicates that there are no problems with regard to the ambiguity of the grammar. For an ambiguous grammar, two or more parse trees may correspond to the same alignment and structure. Since the score is solely dependent on the alignment and structure, the choice of the parse trees depends upon the detailed order of computations. This indicates that the obtained parse tree has little relevance. However, the alignment and its associated structure are unique, and they are sufficient for our purpose.

In contrast, the computations of the match and pair probabilities are affected by the redundant enumeration of the alignment and structure. However, both the forward–backward algorithm of the model of Figure 1 and the McCaskill algorithm enumerate all the alignments and structures without redundancies. Thus, the whole algorithm is devoid of any redundancy problems.

2.2 Reduction of the DP region

Since the loop match score e_L and the base pair match score e_S are proportional to the match probability $p^{(a)}$, we consider restricting the $L_x \times L_y$ DP region to a smaller region that includes all the positions with $p^{(a)}(i, j) > \epsilon$, where ϵ is a prespecified threshold value.

For each alignment $\mathcal{A}$, let $\mathcal{M}^{\epsilon}_{\mathcal{A}}$ denote the set of match positions in the alignment $\mathcal{A}$ that satisfy $p^{(a)}(i, j) > \epsilon$.

$$\mathcal{M}^{\epsilon}_{\mathcal{A}} = \{(i_I, j_I) | I \in \mathcal{C}_{\mathcal{A}}, p^{(a)}(i_I, j_I) > \epsilon\}$$

For a given initial alignment path and a threshold value $\epsilon > 0$, we then define the restricted DP region as the smallest region in the DP matrix that satisfies the following conditions:

(1) The region is simply connected, i.e. the region has no holes.

(2) The region includes the initial alignment path.

(3) For each alignment path $\mathcal{A}$ with $\mathcal{M}^{\epsilon}_{\mathcal{A}} \neq \emptyset$ in the full DP region, there exists an alignment $\mathcal{A}'$ in the restricted DP region that satisfies $\mathcal{M}^{\epsilon}_{\mathcal{A}} \subseteq \mathcal{M}^{\epsilon}_{\mathcal{A}'}$.

We have described the algorithm for computing the restricted DP region in the Supplementary Material. The third condition implies that if all the match probabilities $p^{(a)}(i, j)$ that are not greater than ϵ are set to zero, then there always exists an alignment in the restricted DP region that has the same score as the optimal score $z_{\max}$ in the full DP region. It implies that for a sufficiently low threshold value ϵ (we use $\epsilon = 0.0001$ throughout the article), restriction of the DP region rarely causes missing of the optimal alignment.

If two sequences are highly similar, the match probabilities concentrate along a specific diagonal in the DP matrix and the reduction of the DP region is quite significant. As shown in the later section, the elapsed time and memory are drastically reduced for similar sequences.

Previous studies (Dowell and Eddy, 2006; Holmes, 2005) have also proposed the algorithms that restrict the DP region using PHMM. In particular, our reduction method is a special case of a more general method proposed by Holmes (2005). However, our method is different from the above-mentioned algorithms in two aspects. First, our method removes only those positions with very small match probability from the DP region rather than selecting only the positions with large match probability as in their methods. This is because the positions with even the slightest match probability might have a large contribution to the DP score of the Sankoff algorithm as the match probability and the score function of the Sankoff algorithm are expected to be only roughly correlated. Second, in our algorithm, the score system of the Sankoff algorithm is more closely related to that of PHMM that is used to reduce the DP region. As defined in the Equations (2) and (3), both the loop match score e_L and the pair match score e_S are proportional to the match probability $p^{(a)}$. This ensures that the total DP score has no contribution from the positions with zero match probability $p^{(a)}(i, j) = 0$. On the other hand, the score systems of the Sankoff algorithm and the PHMM used to restrict the DP region are not directly related in the earlier algorithms. Hence, it is possible that the total alignment score has a large contribution from the positions with diminishing match probabilities. This implies that their restriction methods are at a higher risk of omitting the positions that significantly contribute to the total alignment score.

For these reasons, our restriction method is expected to have less possibility of missing the optimal alignment when compared with those of the previously mentioned algorithms.

2.3 Approximation methods

Most of the alignment softwares based on the Sankoff algorithm provide optional parameters to approximate the DP and to strike a balance between the computational cost and the alignment accuracy (Gorodkin *et al.*, 1997; Havgaard *et al.*, 2005; Hofacker *et al.*, 2004; Holmes, 2005; Mathews and Turner, 2002). Murlet provides two original approximations that constrain the DP region: the strip and skip approximations.

For a given initial alignment path, *the strip approximation* constrains the DP region to a strip region of fixed width δ around the alignment path. If the strip width δ is equal to one, then the resulting alignment after the DP computation is the same as the initial alignment, as in the QRNA software (Rivas and Eddy, 2001). If a diagonal path is specified as the initial alignment path, then the strip approximation corresponds to the band alignment that calculates only the region $|i - j| < \delta$ for row i and column j in the DP matrix.

The band approximation has been used in the previous version of Dynalign (Mathews and Turner, 2002). The limitation of the band approximation is that the band width cannot be smaller than the difference $|L_x - L_y|$ of two sequences. The approximation methods that are adopted by Foldalign and PMMulti also have a similar limitation. The recent version of Dynalign (Uzilov *et al.*, 2006) has adopted an alternative definition of the band region $|i(L_y/L_x) - j| < \delta$ that removes this limitation. The strip approximation is more general as compared to these approximations because the initial path can be arbitrarily far from the main diagonal of the DP matrix, and the strip width can be set to one irrespective of the difference of the sequence lengths.

If the restriction of the DP region by match probabilities is not applied, the strip approximation decreases the computational costs by $(\delta/L)^3$ times with respect to time and $(\delta/L)^2$ times with respect to memory.

The skip approximation constrains the points that are computed during the bifurcation transitions (the last line of Equation 5) to a restricted set of positions in the DP region.

$$\begin{aligned} & M_{i,j,u,v} + M_{u+1,v+1,k,l} \text{ for } i < u < k,\ j < v < l \\ & \Longrightarrow \text{if } (i,j),\ (k,l) \in \mathcal{K}, M_{i,j,u,v} + M_{u+1,v+1,k,l} \text{ for } (\mathrm{u},\mathrm{v}) \in \mathcal{K} \end{aligned} \tag{6}$$

That is, the bifurcation calculation is performed only when the end positions (i, j) and (k, l) are in the skip set $\mathcal{K}$, and the only case considered is the one where the mid position (u, v) is in the skip set $\mathcal{K}$. The skip set $\mathcal{K}$ is the set of grid positions in the DP region $\mathcal{R}$ that is defined as follows.

$$\mathcal{K} = \{(i, j) \in \mathcal{R} | i \in 1 + \kappa\mathbb{Z}, j \in \tau(i) + \kappa\mathbb{Z}\}$$

where $\mathbb{Z}$ is the set of integers, $\tau(\mathrm{i})$ is a point on the initial alignment path at row i, and $\kappa > 0$ is a given parameter. $\kappa = 1$ corresponds to the full bifurcation calculation in the DP region, and in the limit $\kappa \to \infty$, the algorithm can only parse non-bifurcating stem structures similar to the earlier version of Foldalign (Gorodkin *et al.*, 1997). The bifurcation part of computation, which requires $\mathcal{O}(L^6)$ time and $\mathcal{O}(L^4)$ memory, decreases by $1/\kappa^6$ times with respect to time and $1/\kappa^4$ times with respect to memory with the skip approximation.

If the skip size κ is three or more, the bifurcation part is not a dominant factor of computation for aligning sequences shorter than 500 bases. In such cases, the total memory consumption is dominated by the $\mathcal{O}(L^4)$ memory that stores the traceback pointers, for which Murlet requires only one byte per DP recursion. The total time consumption is dominated by the $\mathcal{O}(L^4)$ calculations of the first seven lines of Equation (5). The order of memory consumption is only $\mathcal{O}(L^3)$ for these calculations.

The skip approximation is considered because the occurrence frequency of bifurcations in the parse tree is small as compared to the lengths of the RNA sequences despite the fact that the bifurcation calculation is the most compute-intensive part of the Sankoff algorithm. However, the skip approximation may miss a few base pairs if two neighboring stems are close to each other and no skip points are placed between them.

For a given strip width δ and skip size κ, the DP region of the Sankoff algorithm is determined as follows (see Fig. 2): First, the initial alignment path is determined (Fig. 2a) by the following DP algorithm, which is an application of the MEA principle to the PHMM.

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + p^{(a)}(i, j) \\ M_{i-1,j} \\ M_{i,j-1} \end{cases}$$

We refer to the alignment obtained by this computation as the PHMM-MEA alignment. Next, the DP region is constrained to the strip region around the initial alignment path (Fig. 2b). The DP region is further constrained by removing the side regions with low match probabilities $p^{(a)}$ (Fig. 2c). Finally, the skip set $\mathcal{K}$ is determined within the DP region using the initial alignment path (Fig. 2d).

It is tedious to determine the appropriate strip width δ and skip size κ for each sequence pair being aligned. Murlet estimates the allocated memory and the computational time for each pairwise alignment and automatically determines the strip width and skip size so that the DP region is maximal under the given memory and time limits specified by the user.

The computation time t is estimated by the following formula.

$$t = a\mu_{\text{traceback}} + b\mu^{\frac{6}{4}}_{\text{bifurcation}} \tag{7}$$

where $\mu_{\text{traceback}}$ is the size of the $\mathcal{O}(L^4)$ memory that is required to store traceback information of the Sankoff algorithm, $\mu_{\text{bifurcation}}$ is the $\mathcal{O}(L^4)$ memory that is required to store the scores of the child states of the bifurcation transitions, a and b are fitting parameters, and $\mu^{\frac{6}{4}}_{\text{bifurcation}}$ is the estimated number of bifurcation calculations (see Equation 6).

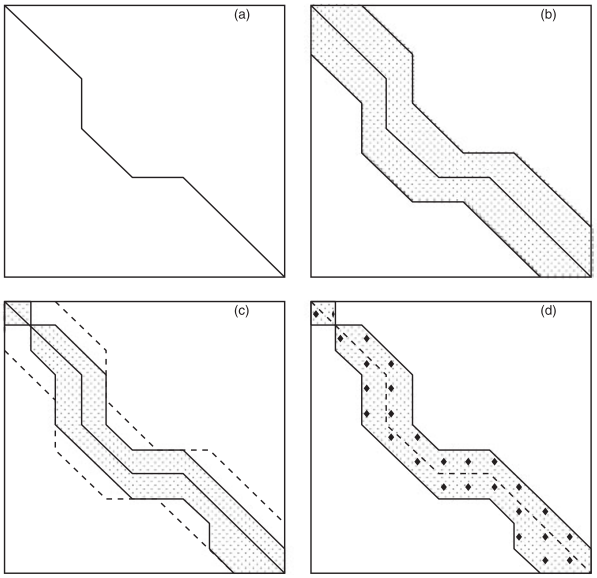

Fig. 2. Procedure to constrain the DP region of the Sankoff algorithm. (**a**) The initial DP alignment is calculated by the PHMM-MEA method. (**b**) The DP region is constrained to a strip region around the initial DP path. (**c**) The DP region is reduced further by removing the regions with low match probabilities. (**d**) The skip set is fixed within the DP region.

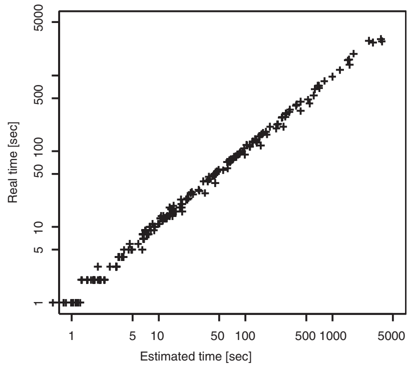

Fig. 3. A scatter plot showing the accuracy of the estimation of computation time. The x-axis is the estimated time in seconds, as computed by Equation (7). The y-axis is the elapsed time in seconds for the pairwise alignment. There are 246 data points.

Figure 3 shows a scatter plot of the estimated time (x-axis) and the real time (y-axis). We used the pairwise alignments derived from the dataset of Table 2. We varied the strip width δ from 0.1 to 0.5 and skip size κ from 1 to 5 and measured the elapsed time for the computation of the pairwise alignments. As observed in the figure, the computation time can be estimated with a reasonable accuracy.

2.4 Probabilistic consistency transformations

For three or more sequences in the same sequence family, Do *et al.* introduced the probabilistic consistency transformation (PCT) of match probability matrices (Do *et al.*, 2005), which is defined by the formula,

$$p^{(a)\mathrm{PCT}}_{x,y}(i,j) \leftarrow \frac{1}{N}\sum_{w\in X,m} p^{(a)}_{x,w}(i,m)p^{(a)}_{w,y}(m,j)$$

where x, y and w represent sequences in X, and i, j and m are the sequence positions in sequences x, y and w, respectively. $p^{(a)\mathrm{PCT}}$ are the match probabilities after the transformation. This computation requires $\mathcal{O}(N^3L^3)$ time for N sequences of length L. By this transformation, the match probability $p^{(a)}_{x,y}(i,j)$ is increased if there are positions in other sequences that are likely to match with both i and j, and it is decreased if there are no such positions. Thus, the transformation introduces the family specific homology information into the match probabilities.

Here, we propose the PCT of the base pairing probability matrices defined by the formula,

$$p^{(b)\mathrm{PCT}}_{x}(i,k) \leftarrow \frac{1}{N}\sum_{w\in X,m,n} p^{(a)}_{x,w}(i,m)p^{(a)}_{x,w}(k,n)p^{(b)}_{w}(m,n)$$

The computation requires $\mathcal{O}(N^2L^4)$ time. The corresponding loop probabilities $q^{(b)\mathrm{PCT}}_{x}(i)$ are computed by applying Equation (1) to $p^{(b)\mathrm{PCT}}_{x}(i,k)$. Then, $q^{(b)\mathrm{PCT}}_{x}(i)$ assumes a value between 0 and 1.

$$0 \leq q^{(b)\mathrm{PCT}}_{x}(i) \leq 1 \tag{8}$$

This justifies the consideration of the transformed matrices $p^{(b)\mathrm{PCT}}_{x}(i,k)$ as the pair probability matrices. The proof of the formula 8 is presented in the Supplementary Material. As in the case of match probabilities, the transformation introduces the family specific structure information into the base-pairing probabilities. We show in the later section that the PCT of the match probabilities considerably improves the alignment accuracy.

The PCTs of $p^{(a)}$ and $p^{(b)}$ are performed for the sparse matrix representations of the probability matrices to reduce the computation time.

2.5 Multiple alignment procedure

We now describe the multiple alignment procedure. First, the base pairing probability matrices and the match probability matrices are computed for each sequence and each pair of sequences, respectively. Next, PCT is performed for the match probabilities; subsequently, PCT of the base-pairing probabilities using the transformed match probabilities. The similarity between a pair of sequences is defined by the score of the Sankoff algorithm along the PHMM-MEA alignment path. Using this similarity measure, a guide tree is constructed by using the unweighted pair group method (UPGMA) clustering algorithm. The progressive alignment is then performed using the guide tree. To align the two groups of aligned sequences, the base-pairing probabilities are averaged across all the sequences of each group. Further, the match probabilities are averaged across all the pairs of sequences between the two groups. The base-pair substitution score $s(i_I, j_I, i_J, j_J)$ in Equation (3) is computed as the sum of the corresponding values for all the pairs of sequences between the groups. We set the proportionality constants γ_L and γ_S (Equations 2 and 3) as dependent on the number of sequences N_1 and N_2 in the two groups as follows:

$$\gamma_L = 0.005$$
$$\gamma_S = 4.0N_1N_2$$

As shown in the Supplementary Material, all the examined multiple alignment programs that make the structure prediction are inferior to Pfold with regard to the accuracy of the predicted structures. It suggests that, at present, it is practical to distinguish between the issue of multiple alignment and that of consensus structure prediction and to use the specialized programs to resolve the latter. Therefore, Murlet does not predict the consensus structure and returns only the aligned sequences.

2.6 The dataset

We collected the test dataset from the Rfam7.0 database (Griffiths-Jones *et al.*, 2003). We used only the hand-curated seed alignments with the consensus structures published in literatures. For each sequence family, we generated up to 1000 random combinations of 10 sequences. We then removed the alignments with mean pairwise sequence identity higher than 95%. Because we are considering the global multiple alignment problem, we removed the alignments that contained more than 30% of the total alignment characters as gap characters. We also removed the alignments that contained < 5% of the total alignment characters as gap characters because the algorithms that merely penalize or forbid the gap insertions show high accuracies for such alignments. We found it difficult to collect completely exclusive alignment set for several sequence families. Therefore, we removed only those alignments sharing more than 30% of sequences with another alignment. Inspecting the number of families and the number of sub-alignments available for each family, we chose the dataset shown in Table 2.

The dataset consists of 85 multiple alignments of 10 sequences. There are 17 sequence families, and there are five alignments for each family. The dataset is reasonably diverse; its mean length varies from 54 bases to 291 bases, and the mean pairwise sequence identities varies from 40 to 94%.

We also used the multiple alignments of BRAlibaseII benchmark dataset for the evaluation (Gardner *et al.*, 2005). The dataset consists of 481 multiple alignments of 5 sequences that are composed of tRNA, Intron_gpII, 5S_rRNA, U5 families in the Rfam5.0 database, and the signal recognition particle RNA family (SRP) in the SRPDB database (Larsen and Zwieb, 1993). As shown in Table 1, approximately half of the alignments have more than 70% sequence identities and few alignments have sequence identities < 50%. Since their dataset does not contain consensus structure annotations to the alignments, we have extracted the consensus structures from the original databases. Since the secondary structures are annotated to all the sequences in SRPDB, we have defined the base pairs that are supported by four or more sequences in the alignment as the consensus base pairs.

2.7 Accuracy measures

The accuracy of the alignments is measured by the standard sum-of-pairs score (SPS) (Carillo and Lipman, 1988). To measure the efficiency of the structural alignment, the consensus structures are predicted from the alignment results using the Pfold program (Knudsen and Hein, 2003). The Matthews correlation coefficients (MCC) are then calculated for the predictions (Matthews, 1975). MCC is defined by the formula

$$\mathrm{MCC} = \frac{\mathrm{tp} \cdot \mathrm{tn} - \mathrm{fp} \cdot \mathrm{fn}}{\sqrt{(\mathrm{tp}+\mathrm{fp})(\mathrm{tp}+\mathrm{fn})(\mathrm{tn}+\mathrm{fp})(\mathrm{tn}+\mathrm{fn})}}$$

where tp indicates the number of correctly predicted base pairs; tn, the number of base pairs that are correctly predicted as unpaired; fp, the number of incorrectly predicted base pairs and fn, the number of true base pairs that are not predicted. Note that tn is computed in units of base pairs and is very large in most cases. The numbers are computed by assigning both reference and predicted consensus structures to each sequence using the alignment and then counting the matches and mismatches of base pairs for all the sequences.

Table 1. Distribution of sequence identity in the BRAlibaseII multiple alignment dataset

Family	Number	Length	% identity	0–50%	50–70%	70–100%
tRNA	98	76	69	21	10	67
Intron_ gpII	92	80	64	1	61	30
5S_rRNA	89	117	70	0	41	48
U5	109	118	72	0	52	57
SRP	93	300	67	6	55	32
Total	481			28	219	234

The first four columns show the family name, the number of alignments, the mean length of sequences and the average value of the mean pairwise sequence identity, respectively. The last three columns show the number distribution of the mean pairwise sequence identities of alignments.

We did not use the consensus structures predicted by Stemloc, PMMulti and RNAforester since the accuracies of their predictions are lower than those of Pfold (see the Supplementary Material).

Since we used the external program Pfold for the computation of MCC, the upper limit of the MCC values is bound by the efficiency of the Pfold program. Furthermore, the results may be skewed by the compatibility of the programs with the Pfold software. To compensate for these inconveniences in our MCC measurement, we also measured the efficiency of structural alignment using the novel indicators sum-of-stem-pairs score (SSS), sum-of-quadruples score (SQS) and pair-column score (PCS) that quantify how well the true stems are aligned to each other. These indicators do not depend on the structure predictions to the alignment results and only use the reference alignments with annotated structure and the subject alignments. They are regarded as analogous to SPS and the column score (or TC score) (Carillo and Lipman, 1988; Thompson *et al.*, 1994), which are frequently used for the evaluation of sequence alignments.

The SQS is defined as the fraction of the count of the pairs of *base pairs* that are correctly aligned as observed in the reference alignment. The counts are computed for all the pairs of sequences. The base-pairing positions of each sequence are derived from the annotated consensus structure in the obvious manner. The SSS is defined similarly; however, the criterion of a count is less stringent and allows the match of base pairs at different alignment columns in the reference alignment. In other words, it counts one if a base pair is aligned to another base pair irrespective of their alignment columns in the reference alignment. SSS measures how well each stem is aligned to another stem in the multiple alignment solely on the basis of the structural annotation and its values are of practical importance for the consensus structure prediction. The PCS is the fraction of base-pairing columns that are correctly reproduced in the subject alignment. PCS is more strongly dependent on the number of aligned sequences as compared to SQS and SSS and indicates the reliability of the alignment at the level of whole columns.

SQS and PCS take values between 0 and 1, and they are equal to 1 if the subject alignment is identical to the reference alignment. SSS is also a non-negative number, and it is equal to 1 if the alignment is identical to the reference alignment. Additionally, it is ≤ 1 if all the stem regions in the reference alignment do not contain gap characters. However, it might be > 1 when two or more sequences have gap characters in the stem regions of the reference alignment. The mathematical definitions and examples of computations for these measures are presented in the Supplementary Material.

3 IMPLEMENTATION

Murlet was implemented using the C++ language. For the computation of the match probabilities, we used the ProbCons software (version 1.10) (Do *et al.*, 2005). For the computation of the base-pairing probabilities, we used the RNAAlifold

Table 2. Comparison of the SPS and MCC values for several multiple alignment programs

Family name	Length	% Identity	Murlet SPS/MCC	ProbCons SPS/MCC	ClustalW SPS/MCC	Stemloc SPS/MCC	PMMulti SPS/MCC	RNAcast SPS/MCC
UnaL2	54	73	**0.97**/0.44	**0.97**/**0.46**	0.85/0.36	0.92/0.41	0.70/0.24	0.37/0.40(2/5)
SECIS	64	41	**0.73**/**0.74**	0.68/0.58	0.35/0.00	0.64/0.69	0.35/0.53	0.39/0.61(4/5)
tRNA	73	45	0.91/0.95	0.88/0.91	0.63/0.66	**0.92**/**0.97**	0.70/0.82	0.37/0.50(1/5)
sno_14q_I_II	75	64	**0.93**/**0.79**	**0.93**/0.75	0.83/0.52	0.83/0.74	0.47/0.61	– (0/5)
SRP_bact	93	47	**0.62**/0.64	**0.62**/0.59	**0.62**/0.60	0.53/0.70 (3/5)	0.46/0.56(4/5)	0.48/**0.73**(3/5)
THI	105	55	**0.84**/**0.76**	**0.84**/0.73	0.59/0.46	0.79/0.75	0.59/0.54	0.19/0.32(1/5)
S_box	107	66	**0.90**/**0.84**	**0.90**/0.83	0.83/0.79	0.85/**0.84**	0.51/0.62	0.51/0.69
5S_rRNA	116	57	0.87/**0.61**	0.87/0.59	0.83/0.54	**0.88**/**0.61**(4/5)	0.58/0.50	0.41/0.51
Retroviral_psi	117	92	**0.98**/0.88	**0.98**/0.88	0.97/0.88	0.96/**0.89**	0.87/0.76	0.76/0.76(4/5)
RFN	140	66	**0.92**/0.67	0.91/**0.68**	0.84/0.65	0.88/0.67	0.62/0.56(3/5)	0.49/0.53(2/5)
5_8S_rRNA	154	61	**0.90**/**0.36**	0.89/0.29	0.80/0.14	0.75/0.24 (1/5)	0.69/0.23(3/5)	0.11/0.18(2/5)
U1	157	59	**0.79**/**0.71**	0.78/0.65	0.73/0.60	– (0/5)	0.55/0.53(2/5)	0.33/0.46
Lysine	181	49	**0.78**/**0.85**	0.76/0.77	0.61/0.51	– (0/5)	0.41/0.58(3/5)	0.46/0.75
U2	182	62	**0.76**/**0.78**	**0.76**/0.63	0.67/0.47	0.60/0.49 (1/5)	– (0/5)	0.32/0.52(2/5)
T-box	244	45	**0.51**/**0.65**	**0.51**/0.59	0.35/0.36	– (0/5)	– (0/5)	0.06/0.07(2/5)
IRES_HCV	261	94	**0.98**/**0.74**	**0.98**/**0.74**	**0.98**/**0.74**	– (0/5)	– (0/5)	0.33/0.40
SRP_euk_arch	291	40	**0.44**/0.60	0.41/0.35	0.35/0.25	– (0/5)	– (0/5)	0.32/**0.62**
Average (all)(85/85)			**0.81**/**0.71**	0.80/0.65	0.70/0.50	–	–	–
Average (Stemloc)(49/85)			**0.86**/**0.71**	0.85/0.66	0.73/0.50	0.79/0.67	–	–
Average (PMMulti)(50/85)			**0.86**/**0.71**	0.85/0.67	0.73/0.51	–	0.58/0.54	–
Average (RNAcast)(53/85)			**0.80**/**0.71**	0.79/0.65	0.70/0.53	–	–	0.40/0.55
Average (common) (24/85)			**0.86**/**0.74**	0.85/0.71	0.74/0.58	0.83/**0.74**	0.59/0.59	0.49/0.62

The first three columns list the Rfam family name, mean sequence length of each family and the mean pairwise percentage identity. The remaining columns show the SPS and MCC values of the alignment results. The MCC values are computed for the structures predicted by the Pfold software. The sequence families are sorted in the ascending order of the mean sequence lengths. Since Stemloc, PMMulti and RNAcast did not align the entire dataset within the time and memory limits, we indicated the fraction of the number of data that was returned in parentheses. The last five rows show the average values of SPS and MCC for each software. The values in 'Average (all)' indicate the average values across all the families. 'Average (Stemloc)', 'Average (PMMulti)', 'Average (RNAcast)' and 'Average (common)' indicate the average values across the partial alignment set for which Stemloc, PMMulti, RNAcast, and all the programs returned results, respectively. The ratios of the number of alignments to the whole dataset are indicated in the round brackets. For each row, the highest values of SPS and MCC are shown in bold type face.

program of the Vienna RNA package (version 1.5) (Hofacker *et al*., 2002; Hofacker, 2003). The base pair substitution matrix was extracted from the Stemloc software in the DART package (Holmes, 2005). Experiments were performed on a cluster of Linux machines equipped with dual AMD Opteron 850 2.4 GHz processors and 6 GB RAM. Due to the formidable time and memory consumption of Stemloc and PMMulti for longer sequence families, we limited the time and the maximal resident physical memory of the process to 500 min and 3.5 GB, respectively. We terminated the computation if the process exceeded the time or memory limit. A stand-alone program of Pfold was obtained for the consensus structure prediction to the alignment results (courtesy of Dr B. Knudsen).

4 DISCUSSION

4.1 Comparison of programs

Table 2 shows a comparison of the accuracy of the alignment for various alignment algorithms. The first three columns indicate the Rfam family name, mean sequence length and mean pairwise percent identity. The remaining columns show the SPS and MCC values for various algorithms: ClustalW (Thompson *et al*., 1994) is based on the ordinary DP algorithm of sequence alignment that does not account for the secondary structure. ProbCons (Do *et al*., 2005) is based on the PHMM-MEA algorithm. Murlet, Stemloc (Holmes, 2005) and PMMulti (Hofacker *et al*., 2004) are based on the Sankoff algorithm.

In Reference (Reeder and Giegerich, 2005), a multiple structural alignment method was proposed as an alternative to the Sankoff algorithm. First, this method predicts the secondary structures that have the same topology or consensus shape for all the unaligned sequences; subsequently, it performs the progressive alignment for the sequences with structure annotation. The secondary structures are predicted by the RNAcast program and the alignments are computed by the RNAforester program, (Hochsmann *et al*., 2004). For the sake of brevity, we have indicated this method as 'RNAcast' in the following tables, though the efficiency depends on both the RNAforester program as well as the RNAcast program.

For Murlet, we set the time and memory limits for each pairwise alignment to 10 min and 2 GB, respectively. The other softwares were used with the default option. If some of the five alignments in the family did not return within the limits of 3.5 GB and 500 min, the fraction of the alignments returned is indicated within parentheses in Table 2.

The last five rows indicate the average values of SPS and MCC for each program. 'Average (all)' indicates the average values taken over all the families. 'Average (Stemloc)', 'Average (PMMulti)', 'Average (RNAcast)' and 'Average (common)'

Table 3. Comparison of the SPS and MCC values for the BRAlibaseII multiple alignment dataset

	Murlet SPS/MCC	ProbCons SPS/MCC	ClustalW SPS/MCC	Stemloc SPS/MCC	PMMulti SPS/MCC	RNAcast SPS/MCC
Average (all)(481/481)	**0.88**/**0.77**	**0.88**/0.75	0.84/0.72	–	–	–
Average (Stemloc)(386/481)	**0.88**/**0.78**	**0.88**/0.75	0.83/0.72	0.86/**0.78**	–	–
Average (PMMulti)(374/481)	**0.89**/**0.77**	**0.89**/0.75	0.84/0.72	–	0.80/0.74	–
Average (RNAcast)(421/481)	**0.89**/**0.77**	0.88/0.74	0.85/0.72	–	–	0.62/0.66
Average (common) (310/481)	**0.90**/**0.77**	0.89/0.75	0.85/0.73	0.88/**0.77**	0.81/0.74	0.64/0.67

The MCC values are computed for the structures predicted by the Pfold software. ‘Average’ implies the same as that indicated in the last rows of Table 2. For each row, the highest values of SPS and MCC are shown in bold type face.

represent the average values across the partial alignment set for which Stemloc, PMMulti, RNAcast and all the programs returned results, respectively. The ratios of the number of alignments to the whole dataset are indicated in parentheses.

Table 2 shows that among the softwares examined, the performance of Murlet is the best in terms of both the alignment accuracy SPS and the accuracy of the structure prediction MCC. Although the SPS values of ProbCons and the MCC values of Stemloc are relatively close to those of Murlet, the MCC values of ProbCons and the SPS values of Stemloc are much lower than the corresponding values of Murlet. The table also shows that the accuracies of ClustalW, PMMulti and RNAcast are lower than those of the other programs. Within the time and memory limit, Stemloc and PMMulti could not align most of the RNA sequences that were longer than 150 bases. In almost all the cases, the failures of Stemloc and PMMulti are caused by excessive memory requirements.

For the present dataset, RNAcast frequently failed to identify any consensus structures from the sequences. We changed the optional parameter ‘−c’ from 10 (default) to 50, which corresponds to the inclusion of the suboptimal structures that have free energy up to 50% higher than the minimal free energy but the number of correctly returned data remained unchanged.

Table 3 shows the SPS and MCC values for the BRAlibaseII multiple alignment dataset. Although the SPS and MCC values are relatively high and the differences of scores among the programs are smaller than the dataset of Table 2, Murlet still shows the highest accuracies with regard to both the SPS and MCC values.

Table 4 shows a comparison of the SSS, SQS and PCS for different softwares. The test sets are the same as those in the last five rows of Table 2. The superiority of Murlet when compared with the other programs is more obvious with respect to these measures. Moreover, Murlet is the only Sankoff-based program that performs better than the PHMM-based ProbCons software in all the accuracy measures. The table indicates that Murlet is the best among the examined programs for the structural alignment of RNA sequences.

4.2 Reduction of time and memory

Figure 4 shows the memory and time consumption of the programs. Each data point corresponds to a sequence family shown in Table 2. The x-axis represents the mean sequence length of the sequence family, and the y-axes represent the maximal resident physical memory in MB (left) and the elapsed time in minutes (right). The memory and time consumptions of ClustalW, ProbCons and RNAcast are very small when compared with those of the Sankoff-based programs, and several points for these programs coincide in the figure. The memory consumption of Stemloc and PMMulti drastically increases for sequences that are longer than 100 bases, and these programs cannot align sequences above 200 nts within the limits. In contrast, Murlet can align 10 sequences of the SRP_euk_arch family of mean length 291, within a realistic memory (570 MB) and time (32 min).

Table 4. Comparison of the accuracy of structural alignments using the proposed accuracy measures

	Program	SSS	SQS	PCS
Average (all)	Murlet	**0.81**	**0.79**	**0.55**
	ProbCons	0.75	0.75	0.52
	ClustalW	0.60	0.60	0.34
Average (Stemloc)	Murlet	**0.84**	**0.83**	**0.59**
	ProbCons	0.79	0.79	0.56
	ClustalW	0.63	0.63	0.39
	Stemloc	0.78	0.76	0.50
Average (PMMulti)	Murlet	**0.84**	**0.83**	**0.59**
	ProbCons	0.80	0.80	0.56
	ClustalW	0.63	0.63	0.36
	PMMulti	0.58	0.51	0.22
Average (RNAcast)	Murlet	**0.79**	**0.77**	**0.52**
	ProbCons	0.73	0.73	0.51
	ClustalW	0.60	0.60	0.38
	RNAcast	0.46	0.36	0.08
Average (common)	Murlet	**0.85**	**0.84**	**0.64**
	ProbCons	0.81	0.81	0.63
	ClustalW	0.67	0.66	0.48
	Stemloc	0.83	0.80	0.58
	PMMulti	0.60	0.51	0.25
	RNAcast	0.55	0.45	0.17

The test sets are the same as those shown in the last five rows of Table 2. For each alignment set and accuracy measure, the highest value of each measure is shown in bold type face for each dataset.

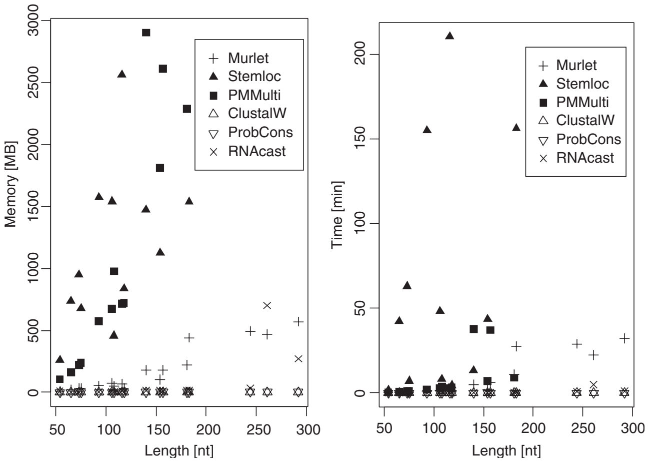

Fig. 4. Elapsed time and the maximal resident memory for computing alignments of Table 2. In both figures, *x*-axis represents the mean length of the sequence families. *Y*-axes represent the maximal resident physical memory of the process in megabytes (MB) (left) and the elapsed time in minutes (right). Each data point represents a specific sequence family of Table 2. Only the alignments returned correctly are plotted. The memory and time consumptions of ClustalW, ProbCons and RNAcast are very small when compared with those of the Sankoff-based programs, and several points for these programs coincide in the figure.

Figure 5 shows the dependence of the reduction of time and memory requirements on the sequence identities. We used 188 multiple alignments of four sequences collected from the Hammerhead_3 ribozyme family in the Rfam database. We compared the estimated time and the allocated memory between the full DP region and those of the region reduced by the match probabilities. For all 188 alignments, the two cases returned exactly the same alignment results. The mean SPS and MCC values were 0.87 and 0.85, respectively. The ratios of time and memory were binned for each 5% segment of the sequence identity, and the mean value for each bin was plotted. The figure shows the general trend that the time and memory usage decreases with the sequence identity. In particular, for sequence identities larger than 60%, the time and memory requirements are several hundred times smaller than those in the full DP case.

4.3 Effects of probabilistic consistency transformations

Figure 6 shows the density plots of the match probability distribution. The probabilities in the left figure are computed using the forward–backward algorithm of PHMM. The sequences are taken from the tRNA family shown in Table 2. The figure on the right represents the probabilities after PCT. Although the dense regions are broadened by the transformation, they are still concentrated around the main diagonal of the DP matrix.

Figure 7 shows an example of the true secondary structure of tRNA (left) and the corresponding base pairing probability matrices (right). The base pairing probability matrix as computed by the McCaskill algorithm is shown in the lower-left part of the figure on the right and that obtained after the transformation is shown in the upper-right part of the matrix. As indicated by the arrow in the figure, the McCaskill algorithm fails to identify one of the four stems of tRNA. PCT corrects this failure by adding small probabilities to this region.

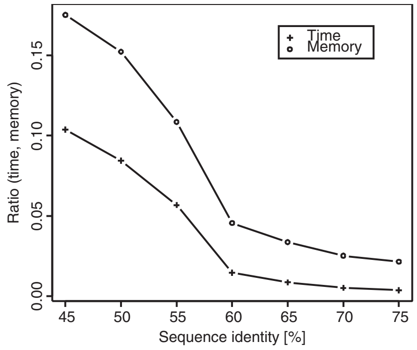

Fig. 5. Dependence of the reduction of time and memory on the sequence identity. The dataset contains 188 multiple alignments of four sequences collected from the Hammerhead_3 ribozyme family in the Rfam database. Their mean length is 55 bases. The *x*-axis represents the mean pairwise sequence identity and the *y*-axis represents the ratio of the estimated time and allocated memory for the DP calculation between the full DP and the DP in the reduced DP region. The data points are categorized into bins of width 5%, and the mean values of the bins are plotted.

Table 5 shows the effects of the PCTs on the alignment accuracies. For all the measures, the accuracies are the highest when the transformation is performed on both the match and pair probabilities. Further, the PCT of the pair probabilities are more significant than that of the match probabilities, and the latter is only effective when the former is also performed. This indicates that McCaskill algorithm often predicts incorrect base pairs and this results in considerable degradation of the alignment quality.

It is known that alignment errors that occur in the earlier pairwise alignments during the progressive alignment method have a considerable impact on the final alignment result. Therefore, it is important to investigate whether the PCTs improve the alignment quality at the level of pairwise alignment.

Table 6 shows the improvement of the pairwise alignment accuracy with the use of PCTs. We have used the test set that consists of 85 pairs of sequences that are randomly selected from each of the multiple alignments of Table 2. The PCTs have been applied by using the other eight sequences that belong to the same multiple alignment. In order to show the levels of accuracy by comparison, we measured the accuracies of pairwise alignments for several pairwise alignment programs as well as the multiple alignment programs.

Foldalign was used with the option that restricts the maximal difference of the segment lengths that are compared to each other to 50 bases. Dynalign was used with the band width of 20 and the gap penalty 0.4 kcal/mol. Murlet was used with the same option as indicated in the multiple case. The other programs were used with their default option. As in the case of multiple alignment, we terminated the program if the computation time or memory exceeded the limits of 500 min and 3.5 GB,

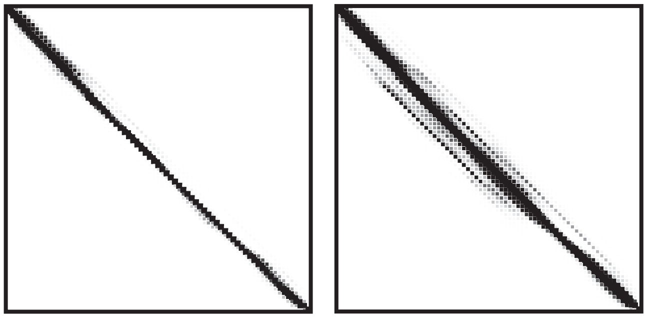

Fig. 6. PCT for match probabilities. The figures on the left and right indicate the match probabilities before and after the transformation, respectively.

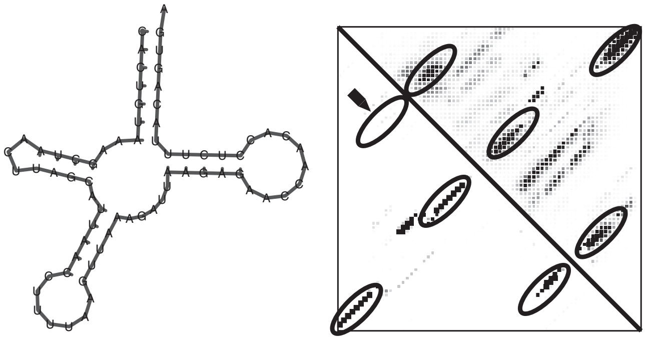

Fig. 7. PCT for the base-pairing probabilities. The left figure is the secondary structure of tRNA, which was plotted using the RNAplot program of the Vienna RNA package (Hofacker, 2003). The right figure illustrates the base-pairing probabilities of a tRNA sequence. The lower left part of the matrix is computed by the McCaskill algorithm. The upper right part is after PCT. In both triangles, the regions of the true stems of tRNA are indicated by ovals. The stem region that was missed by the McCaskill algorithm is indicated by the arrow.

respectively. Only Murlet, ProbCons, and ClustalW returned all the data within these limits. The MCC values were calculated for both the Pfold predictions and the original consensus structure predictions (if available) and the better score is listed in Table 6. Only Dynalign demonstrated the better structure predictions than Pfold (original: 0.61, Pfold: 0.60). The first column of Table 6 shows the programs that are compared with Murlet. The second column shows the fraction of the number of the data returned correctly. The third column shows the mean SPS and MCC values for the programs of the first column. The total computation time in minutes is also shown in parentheses. The fourth column shows the pairwise alignment accuracy and the total computation time of Murlet for the dataset that is used to compute the values of the third column. The fifth column is similar except that PCT is applied to the match and base-pair probabilities of each pairwise dataset by using the other eight sequences that belong to the same multiple alignment of Table 2. Since the datasets are different for each row, the comparisons are meaningful only within the row.

Table 5. Effects of PCTs on the alignment accuracy

	SPS	MCC	SSS	SQS	PCS
$p^{(a)}$ and $p^{(b)}$	**0.81**	**0.71**	**0.81**	**0.79**	**0.55**
$p^{(b)}$	0.80	0.68	0.79	0.77	0.51
$p^{(a)}$	0.74	0.67	0.76	0.72	0.44
None	0.74	0.68	0.78	0.73	0.45

The first column of each row indicates to which of the probabilities ($p^{(a)}$ and $p^{(b)}$) that underwent transformation. The test set is identical to that of Table 2. For each accuracy measure, the highest value is shown in bold type face. The MCC values are computed for the structures predicted by the Pfold software.

Table 6. Improvement of the accuracy of pairwise alignment by PCT

Program	Fraction	SPS/MCC (Time)	Murlet SPS/MCC (Time)	Murlet with PCT SPS/MCC (Time)
ProbCons	85/85	0.75/0.54 (0.3)	0.76/0.56 (3)	**0.79**/**0.60** (84)
ClustalW	85/85	0.69/0.49 (0.7)	0.76/0.56 (3)	**0.79**/**0.60** (84)
Stemloc	78/85	0.72/0.55 (223)	0.78/0.57 (3)	**0.81**/**0.60** (61)
PMMulti	67/85	0.62/**0.61** (70)	0.78/0.58 (2)	**0.82**/**0.61** (44)
RNAcast	84/85	0.45/0.53 (2)	0.75/0.56 (3)	**0.79**/**0.60** (84)
Consan	74/85	0.82/**0.62** (2982)	0.80/0.58 (2)	**0.84**/0.61 (48)
Foldalign	60/85	0.75/0.59 (551)	0.81/0.58 (1)	**0.84**/**0.60** (15)
Dynalign	73/85	0.51/**0.61** (6515)	0.79/0.58 (2)	**0.82**/**0.61** (52)

The PCTs of $p^{(a)}$ and $p^{(b)}$ are applied by using the other 8 sequences that belong to the same multiple alignment of Table 2. The total computation time (in minutes) for each program and dataset is enclosed within parentheses. For each row, the highest SPS and MCC values are shown in bold type face. Except for Dynalign, the MCC values are computed for the structures predicted by the Pfold software. For Dynalign, the MCC value is calculated for the original predicted structures.

Table 6 indicates that Consan is currently the best pairwise alignment program because only Consan shows better scores with regard to SPS and MCC when compared with those of Murlet without PCT. Although the computation by Murlet is very fast if the PCTs are not applied, the alignment accuracies are only modest due to the inaccurate estimation of the match and base-pair probabilities. In the presence of multiple sequences, the inference of probability matrices by Murlet is greatly enhanced by the PCTs, which makes the accuracy of pairwise alignment of Murlet comparable to the best pairwise alignment programs while keeping the computation time ~60 times smaller than that of Consan.

Thus, the PCTs efficiently improve the quality of pairwise alignment by using the information of the other sequences, which results in the enhancement of the final multiple alignment.

5 CONCLUSION

We have developed an efficient method to align multiple sequences of structural RNAs. First, the method computes the base-pairing probabilities and match probabilities. A simple Sankoff algorithm is then applied to obtain the final alignment by using these probabilities.

We have developed several novel ideas and methods in this article:

- The heuristic score system (Equation 3) that is inspired by the MEA algorithm and is represented by the product of the match probabilities and base-pair probabilities.
- Efficient reduction of the DP region using the match probabilities.
- The strip approximation that restricts the DP region to a strip region around the initial alignment path.
- The skip approximation that limits the compute-intensive bifurcation calculations.
- Dynamical determination of the strip width and the skip size based on the estimation of the memory and time consumption. This method has greatly improved the utility of the program.
- The PCT for the base-pairing probabilities, which has been proved essential for ensuring high alignment quality.
- Novel accuracy measures SSS, SQS and PCS for the structural RNA alignment that measure the fraction of the base pairs aligned in the same manner as observed in the reference alignment.

We have shown that our method has the highest accuracy among the examined programs with regard to both the alignment quality and the structure prediction from our alignment. We have also shown that our program can align relatively long (∼300 bases) RNA sequences within a realistic memory and time.

We have only optimized the proportionality constants of the loop match score and the stem match score; however, we have not optimized the pair substitution matrix and the parameters of the models that are used to calculate the match and pair probabilities. It will be interesting to investigate the application of machine-learning methods in order to optimize these parameters in an integrated manner.

ACKNOWLEDGEMENTS

This work was partially supported by the ‘Functional RNA Project’ funded by the New Energy and Industrial Technology Development Organization (NEDO) of Japan. The authors thank their colleagues who worked on the project for their discussions and comments. H.K. thanks Michiaki Hamada for inspiring discussions.

Conflict of Interest: none declared.

REFERENCES

Carillo,H. and Lipman,D. (1988) The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.*, **48**, 1073–1082.

Carninci,P. *et al.* (2005) The transcriptional landscape of the mammalian genome. *Science*, **309**, 1559–1563.

Do,C. *et al.* (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **15**, 330–340.

Do,C. *et al.* (2006) CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, **22**, e90–e98.

Dowell,R. and Eddy,S. (2006) Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinformatics*, **7**, 400.

Dunham,A. *et al.* (2004) The DNA sequence and analysis of human chromosome 13. *Nature*, **428**, 522–528.

Durbin,R. *et al.* (1998) *Biological sequence analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK.

Gardner,P. *et al.* (2005) A benchmark of multiple sequence alignment programs upon structural RNAs (Evaluation Studies). *Nucleic Acids Res.*, **33**, 2433–2439.

Gorodkin,J. *et al.* (1997) Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, **25**, 3724–3732.

Griffiths-Jones,S. *et al.* (2003) Rfam: an RNA family database. *Nucleic Acids Res.*, **31**, 439–441.

Hamada,M. *et al.* (2006) Mining frequent stem patterns from unaligned RNA sequences. *Bioinformatics*, in press.

Havgaard,J. *et al.* (2005) The FOLDALIGN web server for pairwise structural RNA alignment and mutual motif search. *Nucleic Acids Res.*, **33**, W650–W653.

Hochsmann,M. *et al.* (2004) Pure multiple RNA secondary structure alignments: a progressive profile approach. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **1**, 53–62.

Hofacker,I. (2003) Vienna RNA secondary structure server. *Nucleic Acids Res.*, **31**, 3429–3431.

Hofacker,I. *et al.* (2002) Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, **319**, 1059–1066.

Hofacker,I. *et al.* (2004) Alignment of RNA base pairing probability matrices. *Bioinformatics*, **20**, 2222–2227.

Holmes,I. (2005) Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*, **6**, 73.

Holmes,I. and Durbin,R. (1998) Dynamic programming alignment accuracy. *J. Comput. Biol.*, **5**, 493–504.

Knudsen,B. and Hein,J. (2003) Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.*, **31**, 3423–3428.

Larsen,N. and Zwieb,C. (1993) The signal recognition particle database (SRPDB). *Nucleic Acids Res.*, **21**, 3019–3020.

Matthews,B. (1975) Comparison of predicted and observed secondary structure of t4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Mathews,D. and Turner,D. (2002) Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.*, **317**, 191–203.

Mathews,D. *et al.* (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.

McCaskill,J. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

Miyazawa,S. (1995) A reliable sequence alignment method based on probabilities of residue correspondences. *Protein Eng.*, **8**, 999–1009.

Okazaki,Y. *et al.* (2002) Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cDNAs. *Nature*, **420**, 563–573.

Pedersen,J. *et al.* (2006) Identification and classification of conserved RNA secondary structures in the human genome. *PLoS Comput. Biol.*, **2**, e33.

Reeder,J. and Giegerich,R. (2005) Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction (Evaluation Studies). *Bioinformatics*, **21**, 3516–3523.

Rivas,E. and Eddy,S. (2001) Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, **2**, 8.

Sankoff,D. (1985) Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, **45**, 810–825.

Tabei,Y. *et al.* (2006) SCARNA: fast and accurate structural alignment of RNA sequences by matching fixed-length stem fragments. *Bioinformatics*, **22**, 1723–1729.

Thompson,J. *et al.* (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.

Uzilov,A. *et al.* (2006) Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change. *BMC Bioinformatics*, **7**, 173.

Waterman,M. and Eggert,M. (1987) A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *J. Mol. Biol.*, **197**, 723–728.