BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 19 2010, pages 2383–2390
doi:10.1093/bioinformatics/btq439

Sequence analysis

Advance Access publication August 2, 2010

Multi-objective pairwise RNA sequence alignment

Akito Taneda

Graduate School of Science and Technology, Hirosaki University, Hirosaki, Aomori 036-8561, Japan

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: With an increase in the number of known biological functions of non-coding RNAs, the importance of RNA sequence alignment has risen. RNA sequence alignment problem has been investigated by many researchers as a mono-objective optimization problem where contributions from sequence similarity and secondary structure are taken into account through a single objective function. Since there is a trade-off between these two objective functions, usually we cannot obtain a single solution that has both the best sequence similarity score and the best structure score simultaneously. Multi-objective optimization is a widely used framework for the optimization problems with conflicting objective functions. So far, no one has examined how good alignments we can obtain by applying multi-objective optimization to structural RNA sequence alignment problem.

Results: We developed a pairwise RNA sequence alignment program, Cofolga2mo, based on multi-objective genetic algorithm (MOGA). We tested Cofolga2mo with a benchmark dataset which includes sequence pairs with a wide range of sequence identity, and we obtained at most 100 alignments for each inputted RNA sequence pair as an approximate set of weak Pareto optimal solutions. We found that the alignments in the approximate set give benchmark results comparable to those obtained by the state-of-the-art mono-objective RNA alignment algorithms. Moreover, we found that our algorithm is efficient in both time and memory usage compared to the other methods.

Availability: Our MOGA programs for structural RNA sequence alignment can be downloaded at http://rna.eit.hirosaki-u.ac.jp/cofolga2mo/

Contact: taneda@cc.hirosaki-u.ac.jp

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

Received on February 19, 2010; revised on July 17, 2010; accepted on July 25, 2010

1 INTRODUCTION

Since sequence alignment is a fundamental technology for various biological analyses such as gene prediction and phylogenetic inference, a number of sequence alignment algorithms has been proposed for protein, DNA and RNA. Compared to the cases of protein and DNA sequence alignment, RNA sequence alignment algorithms are usually computationally more demanding in both time and space because consensus secondary structure information has to be taken into account in addition to sequence similarity. This is true even when we align two RNA sequences, e.g. the pioneering dynamic programming algorithm (Sankoff, 1985) for structural RNA sequence alignment has computational complexities of $O(N^6)$ in time and $O(N^4)$ in space (where N indicates sequence length) for pairwise alignment in contrast to the lower complexities of non-structural (pure-sequence) pairwise sequence alignment algorithms such as Needleman–Wunsch algorithm (Needleman and Wunsch, 1970). To emphasize on the inclusion of secondary structure score, we refer to RNA sequence alignment as ‘structural RNA sequence alignment’ in the present article. To overcome the high computational complexities of structural RNA sequence alignment problem, various algorithms have been proposed and used so far: dynamic programming (Harmanci *et al.*, 2007; Havgaard *et al.*, 2007; Kiryu *et al.*, 2007b; Will *et al.*, 2007), formal grammar (Dowell and Eddy, 2006; Holmes, 2005), graph representation (Reeder and Giegerich, 2005), linear integer programming (Bauer *et al.*, 2007), Markov chain Monte Carlo sampling (Meyer and Miklos, 2007), stochastic sampling (Xu *et al.*, 2007), simulated annealing (Lindgreen *et al.*, 2007), max-margin model (Do *et al.*, 2008), maximum expected accuracy (Hamada *et al.*, 2009) and genetic algorithm (Taneda, 2008). Constraints (Dowell and Eddy, 2006; Harmanci *et al.*, 2007; Kiryu *et al.*, 2007b) and suboptimal solutions (Mathews, 2005) are also used combined with the approaches mentioned above.

In the previous structural RNA sequence alignment methods, structural RNA sequence alignment problem is formulated as a mono-objective optimization, where a sequence similarity contribution and a structural contribution are taken into account through a mono-objective function. However, structural RNA sequence alignment is essentially multi-objective optimization problem since competing two objective functions, sequence similarity score and secondary structure score, are simultaneously taken into account and there is a trade-off between the two scoring schemes: e.g. at a low sequence similarity, it is difficult to discriminate an accidental sequence similarity and evolutionary conserved nucleotides, hence non-structural alignment methods can give an inaccurate alignment due to accidental matches of nucleotides; on the other hand, if we construct an RNA alignment by maximizing a secondary structure score alone, the alignment can have a reduced sequence similarity score compared to the alignment obtained by a non-structural alignment method.

The previous structural RNA sequence alignment methods explicitly or implicitly assign relative weights to a sequence similarity score and structural score to scalarize the multiple objective functions, and the weights were usually determined based on a training data. In this sense, such a scalarized objective function of structural RNA sequence alignment methods can have a bias due to the training data. Multi-objective optimization is a useful framework for exploring a system with conflicting objective

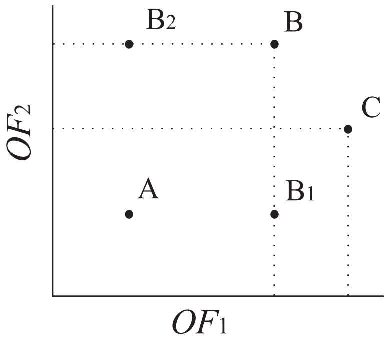

Fig. 1. A schematic illustration of the 'dominance' in a bi-objective optimization, where a higher value is better in each objective function (OF_1 and OF_2). Each circle indicates one solution. In this example, Solution B *strongly dominates* Solution A since the OF_1 and OF_2 of Solution B are strictly better than the OF_1 and OF_2 of Solution A, respectively. Solution C *strongly dominates* Solution A and B_1. Solution B_1 and B_2 are *dominated* (*weakly dominated*) by Solution B, where the value of at least one of the objective functions of the dominated solution is equal to the corresponding value of Solution B.

functions without introducing such a bias to the optimization procedure.

In multi-objective optimization, solutions are evaluated based on their 'dominance'; according to Deb (2001), Solution A *dominates* Solution B if 'all objective function values of Solution A are better than or equal to the corresponding values of Solution B' and 'at least one objective function of solution A is strictly better than that of solution B'. This type of 'dominance' is sometimes called 'weak dominance'. The solution which is not *dominated* by any other solution is referred to as a Pareto optimal solution. If all objective function values of Solution A are strictly better than the corresponding values of Solution B, Solution A *strongly dominates* Solution B. The solution which is not *strongly dominated* by any other solution is referred to as a weak Pareto optimal solution. The concept of dominance is illustrated in Figure 1. The definitions of *dominance*, *weak dominance*, *Pareto optimal solution*, *strong dominance* and *weak Pareto optimal solution* are taken from Deb (2001). The two solutions share strictly the same objective function values, which are neither *dominated* nor *strongly dominated* on each other.

In Cofolga2mo, we explore not Pareto optimal solutions but weak Pareto optimal solutions. This is because we have an interest in all solutions with the highest P or highest s. For example, when multiple solutions have the highest P in a solution set and they have values of s different from each other, only one solution (with the highest s) of the solutions with the highest P is included in Pareto optimal solutions, whereas weak Pareto optimal solutions contain all solutions with the highest P in the solution set. When we align RNA sequences with a very low sequence identity, the solutions with a low s and the highest P also can contain a good alignment since the sequence identity becomes less reliable in such a case. In addition, if we use an iterative-refinement approach to solve a multi-objective optimization problem, weak Pareto optimal solutions in a solution set can give a good guess for the better approximate set at the later steps (e.g. a weak Pareto optimal solution may evolve to a Pareto optimal solution by a slight increase in s); for this reason, we assign a dominance rank of one to the weak Pareto optimal solutions in a solution set (for the dominance rank, see the 'Section 2.3.2').

The goal of multi-objective optimization is to find all Pareto or weak Pareto optimal solutions in the set of all possible solutions. By using a multi-objective optimization method, we can obtain less-biased solutions for a system with conflicting multiple objective functions compared to when using its mono-objective counterparts. Since previous structural RNA sequence alignment methods utilize a biased single objective function, 'how good alignments can we obtain by using a multi-objective optimization?' is a natural question for structural RNA sequence alignment problem.

To explore weak Pareto optimal RNA sequence alignments, we propose a multi-objective genetic algorithm (MOGA; Deb, 2001) for pairwise structural RNA sequence alignment. MOGA is a powerful approach for solving multi-objective optimization problems, where Pareto or weak Pareto optimal solutions are explored with a dominance-based selection and genetic operators. So far, multi-objective optimization has been applied to various problems in bioinformatics (Handl *et al.*, 2007) such as sequence alignment (Paquete and Almeida, 2009; Roytberg *et al.*, 1999). To our knowledge, the present article is the first report on the application of multi-objective evolutionary algorithm to the structural RNA sequence alignment.

2 METHODS

We developed a structural RNA sequence alignment program, Cofolga2mo, on the basis of MOGA. Cofolga2mo is designed for global pairwise sequence alignment. The C language implementation of Cofolga2mo was derived from a genetic algorithm (GA) program, Cofolga2, previously developed for structural RNA sequence alignment (Taneda, 2008). Here, we briefly describe the algorithmic parts common to the previous version and focus on representing the newly introduced parts for multi-objective optimization. GA is a widely used technique for search and combinatorial optimization, which mimics evolution in nature (Goldberg, 1987). GA iteratively improves a population of solutions by applying genetic operators such as mutations and crossovers. In literature, various types of solution such as gray code (Goldberg, 1987) and random key (Bean, 1994) have been used in GA. In our GA, a pairwise sequence alignment is used as a solution.

2.1 Objective functions

The multiple objective functions taken into account in our MOGA are secondary structure score P and sequence similarity score s. For a given pairwise RNA sequence alignment of Sequences A and B, the P is defined using averaged base-pairing probabilities (BPPs), b_{ij}, as follows:

$$P=\sum_{i<j} b_{ij}, \tag{1}$$

$$b_{ij}=\begin{cases}(p^{\mathrm{A}}_{k_il_j}+p^{\mathrm{B}}_{m_in_j})/2 & p^{\mathrm{A}}_{k_il_j}\neq 0 \ \text{ and } \ p^{\mathrm{B}}_{m_in_j}\neq 0,\\ 0 & \text{otherwise,}\end{cases} \tag{2}$$

where i and j indicate the alignment column positions in a pairwise alignment. k_i and l_j (m_i and n_j) are the nucleotide positions in Sequence A (Sequence B) corresponding to alignment column position i and j, respectively. p^{A}_{kl} and p^{B}_{mn} are the BPPs of single sequence A and B, respectively. The p^{A}_{kl} and p^{B}_{mn} are computed by RNAfold (Hofacker *et al.*, 1994). It is noted that the P is applicable to pseudoknotted RNAs without a modification if we can obtain the BPPs for a pseudoknotted RNA by a BPP prediction method allowing pseudoknots, e.g. NUPACK (Dirks and Pierce, 2004).

The sequence similarity score s is calculated with the RIBOSUM-85-60 (Klein and Eddy, 2003). We use affine gap penalties of thirty for opening

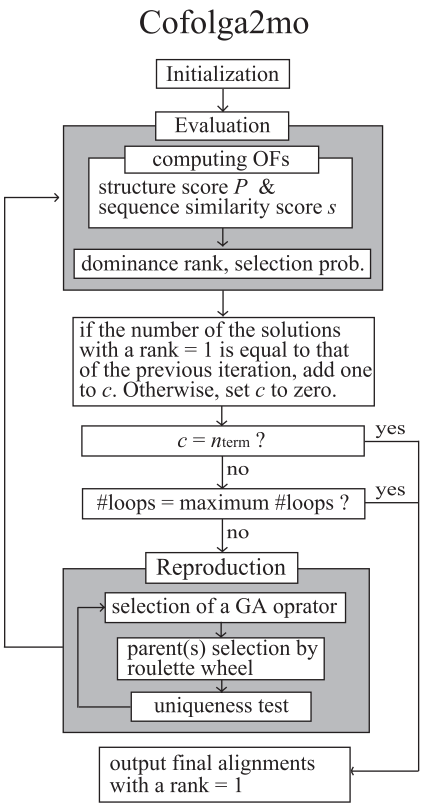
Cofolga2mo
Initialization
Evaluation
computing OFs
structure score P &
sequence similarity score s
dominance rank, selection prob.
if the number of the solutions with a rank = 1 is equal to that of the previous iteration, add one to c. Otherwise, set c to zero.
$c = n_{\mathrm{term}}$?
yes
no
#loops = maximum #loops ?
yes
no
Reproduction
selection of a GA oprator
parent(s) selection by roulette wheel
uniqueness test
output final alignments with a rank = 1

Fig. 2. Flowchart of Cofolga2mo algorithm. A stopping criterion $n_{\mathrm{term}} = 30$ was used in the present study.

and four for elongation, which are taken from our previous paper (Taneda, 2008).

It is noted that our algorithm does not output a consensus secondary structure of two RNAs. For the purpose, a post-processing by an alignment folding program (Bernhart *et al.*, 2008; Kiryu *et al.*, 2007a) will be needed.

2.2 Genetic algorithm

Our GA is designed on the basis of a standard GA framework, where initialization, evaluation and reproduction steps are used as GA modules. Flowchart of Cofolga2mo is shown in Figure 2. First, a population of solutions is initialized using a random generation of pairwise alignments. In the present study, a population size $N_{\mathrm{p}} = 50$ or 100 is used. Throughout a run, duplication of identical alignments in one population is not allowed. After the initialization, the solutions are evaluated and better solutions propagate to the next generations by reproduction procedures including mutations and crossovers. The iteration between evaluation and reproduction continues until a stopping criterion is satisfied or until the user-defined number of the maximum iteration is reached. The maximum iteration was set to 200 in the present study. As the stopping criterion, we use 'the number of the solutions which are not *strongly dominated* in the population is not changed for a continuous n_{term} times', where $n_{\mathrm{term}} = 30$ was used.

2.3 Cofolga2mo algorithm

2.3.1 Iniatilization In this step, we generate $N_{\mathrm{p}} - 1$ pairwise alignments by weighted stochastic backtracking (its detail is described in our previous paper; Taneda, 2008), where we randomly backtrack the dynamic programming matrix constructed based on StrAl algorithm (Dalli *et al.*, 2006). In addition to the random alignments, we add a non-random alignment obtained based on StrAl algorithm to the initial population in order to give a good initial guess for aligning the sequence pairs with a low sequence identity.

2.3.2 Evaluation The objective functions, P and s, for each solution are evaluated in this GA module. After evaluating N_{p} solutions in the current population, we assign a 'dominance rank' to each solution based on the values of the objective functions. The non-dominated sorting (Deb, 2001) for determining the dominance rank we use is as follows:

Step 1 Set $r = 1$ and create an empty set S. All solutions in the current population are copied to S.

Step 2 Find all solutions that are not *strongly dominated* in S. A rank of r is assigned to the solutions.

Step 3 Delete all solutions with a rank of r from S. Then, if S is empty, we stop; otherwise increment r by one and we move to Step 2.

In the worst case (e.g. when each dominance rank has only one solution), this algorithm requires $O(MN_{\mathrm{p}}^3)$ comparisons of an objective function, where M and N_{p} are the number of objective functions and the GA population size, respectively. It is noted that a more efficient $O(N_{\mathrm{p}} logN_{\mathrm{p}})$ algorithm (Jensen, 2003) for computing dominance rank exists.

The selection probability, which gives the size of virtual slots for the roulette wheel selection, is assigned to each solution in such a way that the selection probability is proportional to the inverse of the dominance rank.

2.3.3 Reproduction In this step, new child solutions are generated and they replaces the *strongly dominated* solutions in the current population, whose dominance rank $r > 1$ (elite-preserving strategy). To generate a child solution, we randomly invoke a genetic operator, and then select one parent or two parents (in accordance with a selected genetic operator) by roulette wheel selection. In Cofolga2mo algorithm, we use three random operators (two-point crossover, gap-block shuffling and local re-alignment with weighted stochastic backtracking) and two 'dominance-based' operators (dominance-based crossover and dominance-based gap-block shuffling). The idea of these operators is taken or derived from the genetic operators of RAGA (Notredame *et al.*, 1997), and the three random operators are same with those described in our previous paper (Taneda, 2008). The random two-point crossover randomly selects two parent solutions from the current population, then cuts them into at most three sub-alignments separated by an equivalent alignment block and splices the sub alignments to generate a new child solution. Random gap-block shuffling randomly shifts a continuous gap. Local re-alignment with weighted stochastic backtracking randomly selects a small region in a selected parent solution and re-aligns the small region in a random manner.

To give a greedy nature to the reproduction procedure, the two dominance-based operators are introduced. In the dominance-based two-point crossover, first we determine the equivalent alignment blocks between two randomly selected parent solutions to separate each parent alignment into at most three sub-alignments; then, by examining all combinations of the sub-alignment concatenation, we find the child solution that is not *strongly dominated* by the parent solutions and has the highest P among both the parent and generated child solutions. The dominance-based gap-block shuffling shifts a randomly selected continuous gap and explores the child solution that *strongly dominates* the parent solution and has the highest P among the generated child solutions. In the dominance-based operators, we greedily search not in a s direction but in a P direction; this is because from our experience, generating a solution with a high P is more difficult than generating that with a high s when we use random genetic operators.

The gap-block shuffling operators and local re-alignment with weighted stochastic backtracking have a parameter denoted as L, which controls the degree of modification (larger L corresponds to larger modification of alignment). In Cofolga2mo, we use $L=50$ which is same with that of our previous paper (Taneda, 2008).

The five genetic operators are invoked with an equal probability, i.e. 0.2. When each genetic operator fails to generate a new child solution (e.g. gap-block shuffling operator cannot generate a child alignment for a gap-less parent alignment), we again randomly select a genetic operator and parent solution(s), and try to generate a new solution.

2.4 Non-dominated sorting genetic algorithm 2

Non-dominated sorting genetic algorithm 2 (NSGA2; Deb, 2001; Deb *et al.*, 2000) is one of the standard evolutionary algorithms for solving multi-objective optimization problems. To compare Cofolga2mo algorithm described above and the standard MOGA algorithm, we implemented a structural RNA sequence alignment program, Cofolga2ns, based on NSGA2.

Flowchart of Cofolga2ns is shown in Supplementary Figure S1. In Cofolga2ns, the initialization exactly same with that of Cofolga2mo algorithm is performed (see 'Section 2.3.1'). Then Cofolga2ns proceeds in accordance with the framework mentioned in 'Section 2.2'; i.e., we repeat evaluation and reproduction steps to obtain an approximate set of weak Pareto optimal solutions. We use the stopping criterion same with that of Cofolga2mo described in 'Section 2.2'.

2.4.1 Evaluation In NSGA2, after evaluating the objective functions of each solution, dominance ranks are assigned to the combined population, R, of the N_p parent solutions + N_p child solutions by using the $O(MN_\mathrm{p}^2)$ non-dominated sorting (Deb, 2001), where M is the number of objective functions (at the first GA iteration, the N_p parent solutions alone are used as R). Similar to the case of Cofolga2mo, a more efficient $O(N_\mathrm{p} logN_\mathrm{p})$ algorithm (Jensen, 2003) can accelerate the non-dominated sorting. Then the solutions in R are sorted in ascending order of dominance rank. Moreover, sorting in descending order of crowding distance (Deb, 2001) is also performed for a subset of R for niching, i.e. for maintaining the solutions in one population as diverged as possible within the elite-preserving strategy. The top N_p solutions in the sorted R are used as the parent solutions of the next generation.

2.4.2 Reproduction In reproduction step of NSGA2, N_p child solutions are generated from the parent N_p solutions by using crossover and mutation operators. Cofolga2ns uses the five genetic operators same with those used in Cofolga2mo. Parent selection is performed with crowded tournament selection (Deb, 2001).

2.5 Alignment quality benchmark and comparison

To assess the quality of pairwise RNA sequence alignments, we used the Matthews correlation coefficient (CC), sensitivity (SEN), positive predictive value (PPV) and sum-of-pairs score (SPS), where SPS was calculated with bali_score.c (Thompson *et al.*, 1999). CC, SEN and PPV are defined based on the number of correctly predicted base pairs (TP), the number of negative pairs (= the base pairs not included in the reference base pairs) predicted as negative (TN), the number of the negative pairs incorrectly predicted as a base pair (FP) and the number of the reference base pairs which is not included in the predicted base pairs (FN)

$$\mathrm{CC}=\frac{\mathrm{TP}\cdot\mathrm{TN}-\mathrm{FP}\cdot\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}, \tag{3}$$

$$\mathrm{SEN}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\quad \mathrm{PPV}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{4}$$

Due to the large *TN* in RNA secondary structure prediction, CC can be well approximated by the geometric mean of SEN and PPV, i.e. usually CC increases with SEN and PPV (Gorodkin *et al.*, 2001). For the sequences in all alignments obtained by Cofolga2mo, Cofolga2ns and the other alignment programs, we assigned the base pairs predicted by RNAalifold (Bernhart *et al.*, 2008) to evaluate CC, SEN and PPV. Our alignment quality benchmark was performed by using the reference pairwise RNA sequence alignments taken from the BRAliBase 2.1 k2-dataset (Wilm *et al.*, 2006), whose sequence identity ranges from 16% to 75%. This dataset is composed of 5010 pairwise alignments taken from 32 RNA families. Details for each RNA family in the BRAliBase 2.1 k2-dataset we used can be found in Supplementary Table S1. Since the BRAliBase 2.1 does not provide reference base pairs, we extracted the corresponding base pair information from Rfam 7.0 (Gardner *et al.*, 2009) on which the BRAliBase 2.1 was constructed and we added reference secondary structures to the BRAliBase 2.1 k2-dataset for alignment quality evaluation.

Table 1. Comparison results between Cofolga2mo and Cofolga2ns for 5010 RNA sequence pairs taken from the BRAliBase 2.1 k2-dataset

Method	#solutions	CC	SEN	PPV	SPS
Cofolga2mo(50)	34 (±14)	0.80	0.78	0.83	0.79
Cofolga2mo(100)	56 (±30)	**0.81**	**0.79**	**0.84**	0.79
Cofolga2ns(50)	37 (±14)	**0.81**	0.78	**0.84**	0.79
Cofolga2ns(100)	62 (±32)	**0.81**	**0.79**	**0.84**	**0.80**

Mean values for CC, SEN, PPV and SPS are shown. The alignments with the best CC in each approximate set of weak Pareto optimal solutions are used to calculate the mean values. A '#solutions' column is the average number of solutions (±SD) for a sequence pair. Population sizes of 50 and 100 were used (a population size is indicated in the parenthesis of a 'method' column). The highest value in each measure is indicated by boldface.

In addition to the BRAliBase 2.1 k2-dataset, we performed benchmarks with the test dataset used to benchmark CONSAN in Dowell and Eddy (2006) and a dataset which was constructed based on the internal transcribed spacer 2 (ITS2) database (Schultz *et al.*, 2005; Selig *et al.*, 2008). As the CONSAN dataset, we used R100-pairs.stk, percid.stk and stemloc.stk (Dowell and Eddy, 2006); these datasets include 200, 324 and 22 RNA sequence pairs with an annotated structure and a reference alignment, respectively. The sequences and annotated structures in the ITS2 dataset were taken from 'the original 5000 sequences and structures' of the ITS2 database (ITS2.html; Schultz *et al.*, 2005). We extracted the sequence pairs that have sequence identities ranging from 37% to 75% and lengths of 100 to 150 nt (mean length is 141.3 nt), where sequence identities were calculated based on the pairwise alignments constructed by MAFFT. As a result, 289 ITS2 sequence pairs were obtained. The ITS2 dataset can be browsed at the Cofolga2mo website.

By using the datasets, we compared the benchmark results by Cofolga2mo with those by the other alignment methods including both non-structural and structural alignment methods. The alignment programs used in the comparison are Foldalign 2.1.0 (Havgaard *et al.*, 2007), LocARNA 0.99 (Will *et al.*, 2007), Dynalign 4.5 (Harmanci *et al.*, 2007), LARA 1.3.1 (Bauer *et al.*, 2007), StrAl 0.5.2 (Dalli *et al.*, 2006), MAFFT 6.240 (Katoh *et al.*, 2002) and ClustalW 1.83 (Thompson *et al.*, 1994). The commands for invoking them are same with those summarized in Table 1 of our previous paper (Taneda, 2008). In addition to the lowest energy alignments of Dynalign, suboptimal alignments by Dynalign were also obtained using a 'maxtrace = 100' option (the other parameters were set to the values same with those of the lowest energy calculation) for a performance comparison.

3 RESULTS AND DISCUSSION

3.1 Approximate set of weak Pareto optimal alignments

Figure 3 shows typical distributions of the solutions obtained by Cofolga2mo, where solutions are plotted on a *s–P* plane. In the figure, an initial distribution, distributions obtained at the third and tenth GA iterations, and the final distribution are plotted. In Figure 3, we can see that a randomly generated initial distribution gradually

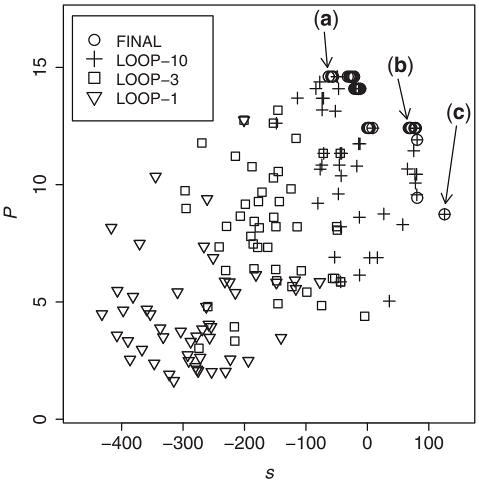

Fig. 3. An example of the solution distributions in an objective function space. This figure was plotted based on the results for a tRNA pair (the tRNA sequences were taken from the BRAliBase 2.1 k2-dataset (Wilm *et al.*, 2006); a sequence identity = 38% and SCI = 1.13). This calculation was performed by Cofolg2mo with a population size of 50. In the figure, not only the results of the final solutions (denoted by '◯'), but those at the first ('▽'), third ('□') and tenth ('+') GA iteration are plotted, where each symbol corresponds to one alignment. Alignments corresponding to solutions (a), (b) and (c) are shown in Figure 4.

```
(a) s = -62.88 P = 14.61 SPS = 0.63 SCI = 0.79 BRAli = 0.50

 1 GGCGGCAUGGCCAA-GC------GGUAAGGCAGGGGACUGCAAAUCCUUUAUCCCCAGUUCAAAUCUGGGUGCCGCCU
 2 GCCUUGAAAGCUCAACAACUAGAGCUUUGGUCUUGUAAACC------AGGAGAGAGGGUAAACU-CCCUCUCAAGGCU
   * *   *  **  *         * *  **    * *   *         *      **  *   *    *   * **

(b)  s = 69.90 P = 12.42 SPS = 1.00 SCI = 1.02 BRAli = 1.02

 1 GGCGGCAUGGCCAAGCGGU-AAGGCAGGGGACUGCAAAUCCUUUAUCCCCAGUUCAAAUCUGGGUGCCGCCU
 2 GCCUUGAAAGCUCAACAACUAGAGCUUUGGUCUUGUAAACCAGGAGAGAGGGUAAACU-CCCUCUCAAGGCU
   * *   *  ** * *    *  **   ** **   ** **   *      **  *   *    *   * **

(c)  s = 125.59 P = 8.74 SPS = 0.46 SCI = 0.71 BRAli = 0.33

 1 GGCGGCAUGGCCAAGCGGUAAGGCAGGGGACUGCAAAUCCUUUAUCCCCAGUUCAAAUCUGGGUGCCGCCU
 2 GCCUUGAAAGCUCAACAACUAGAGCUUUGGUCUUGUAAACCAGGAGAGAGGGUAAACUCCCUCUCAAGGCU
   * *   *  ** * *    **      *       *  *          * * ** **    *   * **
```

Fig. 4. Example alignments taken from the approximate set of weak Pareto optimal solutions corresponding to solutions (a), (b), and (c) in Figure 3.

evolves to a better approximate set of weak Pareto optimal solutions. In Figure 4, three alignments (a), (b) and (c) corresponding to the solutions indicated by arrows in Figure 3 are shown. Alignment (b) is exactly the same with its reference alignment, i.e. Alignment (b) has a SPS = 1.0. Alignments (a) and (c) are the highest P alignment and the highest s alignment, respectively. Structure score P decreases in order of (a), (b) and (c) with an increase in sequence similarity score s. In these examples, a trade-off between the penalty for gap insertion and a structure score is clearly seen.

3.2 Comparison of two MOGAs

Benchmark results for Cofolga2mo and Cofolga2ns are shown in Table 1, where the results obtained by using both methods with population sizes of 50 and 100 are shown. The average numbers (±SD) of the approximate weak Pareto optimal alignments for each RNA family are summarized in Supplementary Table S2. As can be seen from Table 1, the results obtained with a population size of 100 are slightly better than or equal to those with a population size of 50 for all evaluation measures. When we use a population size of 100, Cofolga2mo and Cofolga2ns output the alignments with a similar quality except for SPS, where Cofolga2ns is slightly better than Cofolga2mo. In the case of using a population size of 50, although the difference between the alignment qualities of Cofogla2mo and Cofolga2ns is larger than that of a population size of 100, the performances of Cofogla2mo and Cofolga2ns are still comparable. Figure 5 is the comparison results for the computational times of Cofolga2mo and Cofolga2ns when we use a population size of 50. The comparison was performed for five RNA sequences randomly extracted from each of the 32 RNA families in the BRAliBase 2.1 k2-dataset we used. As a result, we found that Cofolga2ns is much slower than Cofolga2mo for the almost overall sequence length range we used. This inefficiency of Cofolga2ns is mainly due to the algorithm of the reproduction step in Cofolga2ns, where N_p child solutions are generated at every GA iteration, while only N_p- (the number of the solutions with rank 1 in the parent population) child solutions are generated in the reproduction step of Cofolga2mo. Evaluation of newly generated child solutions is the most time-consuming part of Cofolga2mo and Cofolga2ns, hence this difference can make Cofolga2ns slower than Cofolga2mo. Indeed, the dominance sorting algorithm of Cofolga2mo has a higher complexity compared to the $O(MN_\mathrm{p}^2)$ non-dominated sorting of Cofolga2ns, the practical computational times of the two dominance sorting algorithms were much smaller compared to those of solution evaluation in the present study (data not shown). When we used a larger maximum GA iteration = 400, we obtained a figure almost similar to Figure 5. Since Cofolga2mo is faster than Cofolga2ns and the alignment performances of the two algorithms are comparable, we use Cofolga2mo to obtain an approximate set of weak Pareto optimal RNA sequence alignments at the rest of the present article. It is noted that a more efficient $O(N_\mathrm{p} logN_\mathrm{p})$ algorithm (Jensen, 2003)

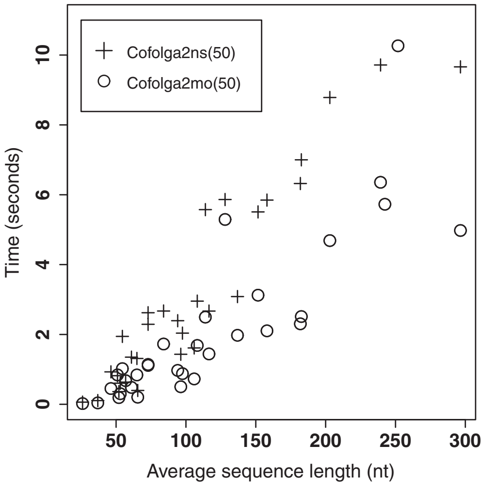

Fig. 5. Sequence length dependence of the computational times measured for Cofolga2mo and Cofolga2ns. The results obtained with a population size of 50 are plotted. Each symbol corresponds to an averaged computational time for randomly selected five RNA sequence pairs taken from an RNA family. Times were measured using a Core2Duo PC (2.80 GHz; 2 GBytes RAM; CentOS 5.3).

Table 2. BRAliBase 2.1 benchmark results for Cofolga2mo and comparison with the results obtained by the other alignment programs

Method	#solutions	CC	SEN	PPV	SPS
Cofolga2mo(best CC)	34 (±14)	**0.80**	**0.78**	**0.83**	0.79
Cofolga2mo(best SPS)		0.75	0.73	0.78	**0.85**
Cofolga2mo(mean±SD)		0.66±0.22	0.64±0.23	0.68±0.23	0.68±0.24
Foldalign	1	0.76	0.73	0.80	0.80
Dynalign	1	0.74	0.70	0.78	0.63
LocARNA	1	0.72	0.69	0.75	0.73
LARA	1	0.69	0.67	0.71	0.69
MAFFT	1	0.64	0.64	0.64	0.75
StrAl	1	0.63	0.63	0.63	0.72
ClustalW	1	0.57	0.58	0.58	0.67

Mean values for CC, SEN, PPV and SPS are tabulated. Results obtained with a population size of 50 are used for the Cofolga2mo results. In a ‘Cofolga2mo(best CC)’ row, mean values of CC, SEN, PPV and SPS calculated for the alignments with the best CC in each approximate set of weak Pareto optimal solutions are shown. The best SPS results are shown in a ‘Cofolga2mo(best SPS)’ row. Means and SDs not for some best performance measure solutions but for all weak Pareto optimal solutions are shown in a ‘Cofolga2mo(mean±SD)’ row. A ‘#solutions’ column is the average number of solutions (±SD) for a sequence pair. The highest value in each column is indicated by boldface. It is noted that the CC in the ‘Cofolga2mo(best CC)’ row and the SPS in the ‘Cofolga2mo(best SPS)’ row are the upper bound values of the Cofolga2mo results in the benchmark. Details for each RNA family can be found in Supplementary Table S3–6 for the best CC results and in Supplementary Table S7–10 for the best SPS results.

for computing dominance rank can accelerate the non-dominated sorting in Cofolga2mo and Cofolga2ns.

3.3 Comparison in alignment quality benchmark

Table 2 shows the alignment quality benchmark results for Cofolga2mo and other sequence alignment methods. Since Cofolga2mo outputs multiple solutions (= an obtained approximate set of weak Pareto optimal solutions) for an inputted sequence pair, the alignment with the best CC in the approximate set is used to obtain the ‘Cofolga2mo(best CC)’ row of Table 2 (detailed results for each RNA family are shown in Supplementary Table S3–6). In this benchmark, Cofolga2mo outputted 34(±14) approximate weak Pareto optimal solutions in average (±SD) when we used a population size of 50. As can be seen in Table 2 and Supplementary Table S3, the alignments obtained by Cofolga2mo contain accurate RNA sequence alignments comparable to those obtained by the other methods for almost all RNA families of the test dataset. It is noted that the CC of the best CC results is the upper bound of the Cofolga2mo results in the benchmark. Sequence identity dependence of the CC is shown in Supplementary Figure S2. Supplementary Figure S2 shows that the approximate set of weak Pareto optimal solutions can give accurate alignments for a wide range of sequence identity.

In addition to the results of the best CC, the best SPS results are also shown in the ‘Cofolga2mo(best SPS)’ row of Table 2 and Supplementary Table S7–10. In the best SPS results, while the result corresponding to the maximized measure (SPS) is improved compared to its counterpart of the best CC results, such maximization simultaneously causes worsening in the other measures as a trade-off; Cofolga2mo’s SPS (0.85) in the ‘best SPS’ results is better than that (0.79) of ‘best CC’, whereas the CC, SEN and PPV decreased to 0.75, 0.73 and 0.78 in the ‘best SPS’ results, respectively (Table 2). Similar to the best CC results, the SPS in the best SPS results corresponds to the upper bound of the Cofolga2mo results in the benchmark.

In addition to the benchmark mentioned above, we compared the approximate set of weak Pareto optimal solutions obtained by Cofolga2mo with the suboptimal solutions computed by Dynalign to examine the difference between the approximate set and the suboptimal solutions computed based on the mono-objective function including a free energy and sequence similarity. For each sequence pair, at most 100 suboptimal solutions were computed by Dynalign. The comparison was performed with the 2030 tRNA and 964 5S rRNA sequence pairs taken from the BRAliBase 2.1 k2-dataset. As a result, Dynalign outputted 28(±24) and 85(±21) suboptimal solutions in average (±SD) for the tRNA and 5S rRNA dataset, respectively. The results for the approximate weak Pareto optimal alignment and the suboptimal alignment with the best CC for each sequence pair are summarized in Table 3. Table 3 shows that the approximate weak Pareto optimal solutions include accurate alignments comparable to the suboptimal alignments based on the mono-objective approach for both the tRNA and 5S rRNA dataset; Cofolga2mo showed better CC and SEN in the tRNA dataset, whereas Dynalign suboptimal solutions include better CC, SEN and PPV for the 5S rRNA dataset; Cofolga2mo gave better SPS for both dataset.

The results for the benchmark performed with the CONSAN dataset are shown in Table 4. In this benchmark, we found that the Dynalign’s suboptimal solutions and the approximate weak Pareto optimal solutions by Cofolga2mo contain accurate alignments. Sequence identity dependence of the benchmark results is shown in Supplementary Figure S3. In Supplementary Figure S3, Cofolga2mo has a slightly wider SDs compared to those of Dynalign(subopt) at a sequence identity range 0–60%; this implies that more diverse alignments are included in the approximate weak Pareto optimal solutions.

The benchmark results for the ITS2 dataset is tabulated in Supplementary Table S11. We used the alignments with the best CC to evaluate Cofolga2mo and the suboptimal solutions by Dynalign. In this benchmark test, Cofolga2mo with a population size of 50 and 100 showed high performances (CC = 0.79 and 0.81, respectively). Although the optimal and suboptimal alignments by Dynalign are more accurate (CC = 0.82 and 0.89, respectively) compared to the solutions of Cofolga2mo, Dynalign needs much longer computational times (longer than 7 min per alignment in average)

Table 3. Performance comparison between approximate weak Pareto optimal solutions by Cofolga2mo and the suboptimal solutions by Dynalign

	tRNA					5S_rRNA				
Rfam ID	#solutions	CC	SEN	PPV	SPS	#solutions	CC	SEN	PPV	SPS
Cofolga2mo	39 (±21)	**0.93**	**0.94**	0.92	**0.86**	60 (±29)	0.78	0.73	0.83	**0.85**
Dynalign(subopt)	28 (±24)	0.92	0.92	**0.93**	0.73	85 (±21)	**0.8**	**0.74**	**0.86**	0.66

Mean values for CC, SEN, PPV, and SPS are shown. The mean values were calculated for the alignments with the best CC in each sequence pair. The results for the tRNA and 5S rRNA sub-datasets taken from the BRAliBase 2.1 k2-dataset are shown. Results obtained with a population size of 100 are shown for Cofolga2mo. The maximum number of suboptimal solutions of Dynalign was set to 100. A '#solutions' column is the average number of solutions (±SD) for a sequence pair. The highest value in each performance evaluation measure is indicated by boldface.

Table 4. The benchmark results for the CONSAN dataset

Dataset	Cofolga2mo(50)	Cofolga2mo(100)	Dynalign(subopt)	Foldalign	Dynalign	LocARNA	LARA	MAFFT
R100.pairs.stk	0.79 (0.65±0.22)	0.80 (0.64±0.21)	**0.84** (0.64±0.16)	0.77	0.75	0.72	0.70	0.69
percid.stk	0.78 (0.64±0.21)	0.79 (0.64±0.21)	**0.85** (0.64±0.18)	0.76	0.72	0.70	0.71	0.61
stemloc.stk	0.74 (0.63±0.30)	0.75 (0.64±0.28)	**0.79** (0.61±0.27)	0.70	0.73	0.67	0.65	0.54
All	0.78 (0.65±0.22)	0.79 (0.64±0.21)	**0.84** (0.64±0.18)	0.76	0.73	0.71	0.71	0.63
#solutions	30 (±15)	49 (±31)	61 (±37)	1	1	1	1	1

Mean CC for each method is shown. The dataset file names are written in a 'dataset' column. For the alignment methods [Cofolga2mo and Dynalign(subopt)], which output multiple solutions for a given sequence pair, the results for the best CC alignments are shown (the mean and SD for all solutions obtained by each method are indicated in parenthesis). The highest value in each dataset is indicated by boldface. It is noted that the values of CC in the Cofolga2mo and Dynalign(subopt) results are the upper bound values of each result in the benchmark. The average number of solutions (±SD) for a sequence pair in the all CONSAN datasets are given in a '#solutions' row.

compared to those of Cofolga2mo, which needs only several seconds to align the ITS2 sequence pairs.

3.4 Computational time and memory usage

Sequence length dependence of the computational times measured for the 32 RNA families in the BRAliBase 2.1 benchmark dataset is shown in Figure 6. The averaged computational time for an RNA family was measured for five RNA sequence pairs randomly extracted from the RNA family. In the figure, the measured computational times for four other structural RNA sequence alignment methods are also plotted. In this comparison, we found that the computational speed of Cofolga2mo with a population size of 50 (denoted by Cofolga2mo(50) hereafter) is comparable with that of LARA for the whole sequence length range we used. LocARNA is faster than Cofolga2mo(50) except for the case of the longest RNA family (SRP_euk_arch). Cofolga2mo(50) is consistently faster than Dynalign and Foldalign for the RNA families longer than ~100 and 140 nt, respectively. Cofolga2mo with a population size of 100 needed computational times approximately twice as long as those of Cofolga2mo(50). In the computational time of Cofolga2mo, the time for the BPP computation by RNAfold is included.

Similar to the genetic algorithm previously we developed (Taneda, 2008), Cofolga2mo needs a small memory to structurally align two RNA sequences. We measured the memory usage of Cofolga2mo when aligning two SRP RNA sequences (we used the SRP_euk_arch with 62% average sequence identity and a SCI = 0.75 in the BRAliBase 2.1 k2-dataset). This sequence pair contains the sequences of 317 and 305 nt; this is one of the longest sequence pairs used in the present study. As a result, we found that Cofolga2mo with population sizes of 50 and 100 can align the two SRP RNA sequences with only 5 and 6 MB, respectively, where memory usages for a perl script and RNAfold are included. Since much smaller calculations by the other structural RNA sequence alignment methods need larger memories according to a literature (Harmanci *et al.*, 2008) (e.g. Foldalign, LocARNA and Dynalign needs 13.3, 7.6 and 21.1 MB to align a 5S rRNA sequence pair), Cofolga2mo's memory usage is small compared to the other methods.

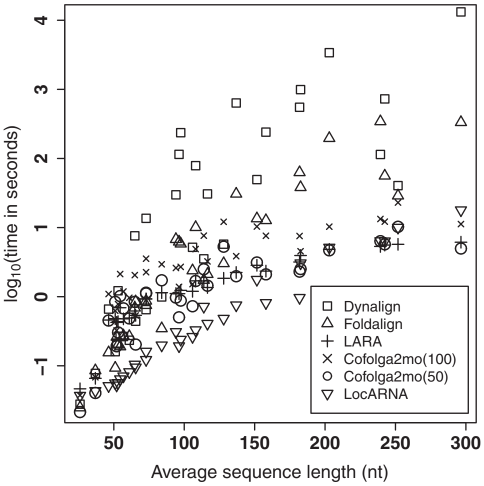

Fig. 6. Comparison in the computational times of structural RNA sequence alignment methods. The results obtained by running Cofolga2mo with a population size of 50 and 100 are plotted. Each symbol corresponds to the averaged computational time for randomly selected five RNA sequence pairs taken from an RNA family. Computational times were measured using a Core2Duo PC (2.80 GHz; 2 GB RAM; CentOS 5.3).

4 CONCLUSION

In the present article, we have proposed MOGAs, Cofolga2mo and Cofolga2ns, which compute the approximate set of weak Pareto optimal solutions for structural pairwise RNA sequence alignment. By performing a BRAliBase 2.1 benchmark test, we found that the approximate set for RNA sequence alignments contain high-quality alignments comparable to the alignments obtained by the other recent mono-objective structural RNA alignment programs. Moreover, we found that Cofolga2mo can give accurate tRNA and 5S rRNA alignments comparable to the suboptimal alignments by Dynalign with much smaller computational resources. Although the algorithm proposed in the present study is designed for pairwise alignment construction, its extension to multiple alignment will be possible in various ways: e.g. we can generate approximate Pareto optimal multiple alignments by straightforwardly extending the present MOGA, i.e. by using crossovers and mutation operators for modifying a multiple alignment like SAGA (Notredame and Higgins, 1996); constructing a multiple alignment based on the approximate set of the pairwise alignments obtained for all pairwise combinations of inputted RNA sequences is also an interesting approach. Since MOGA usually outputs multiple solutions, development of a methodology for selecting the solution which is favorable for the decision-maker is also an important direction of the future research. A graphical user interface might be helpful for the purpose.

Funding: Grant for Priority Research Designated by the President of Hirosaki University (in part).

Conflict of Interest: none declared.

REFERENCES

Bauer,M. *et al.* (2007) Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. *BMC Bioinformatics,* **8**, 271.

Bean,J.C. (1994) Genetic algorithms and random keys for sequencing and optimization. *ORSA J. Comput.,* **6**, 154–160.

Bernhart,S. *et al.* (2008) RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics,* **9**, 474.

Dalli,D. *et al.* (2006) STRAL: progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time. *Bioinformatics,* **22**, 1593–1599.

Deb,K. (2001) Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester.

Deb,K. *et al.* (2000) A fast elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.,* **6**, 182–197.

Dirks,R.M. and Pierce,N.A. (2004) An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *J. Comput. Chem.,* **25**, 1295–1304.

Do,C.B. *et al.* (2008) A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics,* **24**, 68–76.

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

Gardner,P.P. *et al.* (2009) Rfam: updates to the RNA families database. *Nucleic Acids Res.,* **37**, D136–D140.

Goldberg,D.E. (1987) *Genetic Algorithms in Search, Optimization and Machine learning*. Addison-Wesley, New York.

Gorodkin,J. *et al.* (2001) Discovering common stem-loop motifs in unaligned RNA sequences. *Nucleic Acids Res.,* **29**, 2135–2144.

Hamada,M. *et al.* (2009) CentroidAlign: fast and accurate aligner for structured RNAs by maximizing expected sum-of-pairs score. *Bioinformatics,* **25**, 3236–3243.

Handl,J. *et al.* (2007) Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Trans. Comput. Biol. Bioinform.,* **4**, 279–292.

Harmanci,A. *et al.* (2007) Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign. *BMC Bioinformatics*, **8**, 130.

Harmanci,A. *et al.* (2008) PARTS: probabilistic alignment for RNA joint secondary structure prediction. *Nucleic Acids Res.,* **36**, 2406–2417.

Havgaard,J. *et al.* (2007) Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. *PLoS Comput. Biol.,* **3**, 1896–1908.

Hofacker,I. *et al.* (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.,* **125**, 167–188.

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

Jensen,M.T. (2003) Reducing the run-time complexity of multiobjective EAs: the NSGA-II and other algorithms. *IEEE Trans. Evol. Comput.,* **7**, 503–515.

Katoh,K. *et al.* (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.,* **30**, 3059–3066.

Kiryu,H. *et al.* (2007a) Robust prediction of consensus secondary structures using averaged base pairing probability matrices. *Bioinformatics,* **23**, 434–441.

Kiryu,H. *et al.* (2007b) Murlet: a practical multiple alignment tool for structural RNA sequences. *Bioinformatics,* **23**, 1588–1598.

Klein,R. and Eddy,S. (2003) RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics,* **4**, 44.

Lindgreen,S. *et al.* (2007) MASTR: multiple alignment and structure prediction of non-coding RNAs using simulated annealing. *Bioinformatics,* **23**, 3304–3311.

Mathews,D. (2005) Predicting a set of minimal free energy RNA secondary structures common to two sequences. *Bioinformatics,* **21**, 2246–2253.

Meyer,I.M. and Miklos,I. (2007) SimulFold: simultaneously inferring RNA structures including pseudoknots, alignments, and trees using a Bayesian MCMC framework. *PLoS Comput. Biol.,* **3**, e149.

Needleman,S. and Wunsch,C. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.,* **48**, 443–453.

Notredame,C. and Higgins,D. (1996) SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res.,* **24**, 1515–1524.

Notredame,C. *et al.* (1997) RAGA: RNA sequence alignment by genetic algorithm. *Nucleic Acids Res.,* **25**, 4570–4580.

Paquete,L. and Almeida,J.P.O. (2009) Experiments with Bicriteria Sequence Alignment. In Shi,Y. *et al.* (eds) *Cutting-Edge Research Topics on Multiple Criteria Decision Making*, volume 35 of Communications in Computer and Information Science. Springer, Berlin Heidelberg, pp. 45–51 .

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

Roytberg,M.A. *et al.* (1999) Pareto-optimal alignment of biological sequences. *Biophysics,* **44**, 565–577.

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

Schultz,J. *et al.* (2005) A common core of secondary structure of the internal transcribed spacer 2 (its2) throughout the eukaryota. *RNA,* **11**, 361–364.

Selig,C. *et al.* (2008) The ITS2 Database II: homology modelling RNA structure for molecular systematics. *Nucleic Acids Res.,* **36**, D377–D380.

Taneda,A. (2008) An efficient genetic algorithm for structural RNA pairwise alignment and its application to non-coding RNA discovery in yeast. *BMC Bioinformatics,* **9**, 521.

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.

Thompson,J. *et al.* (1999) BAliBASE: a benchmark alignments database for the evaluation of multiple sequence alignment programs. *Bioinformatics,* **15**, 87–88.

Will,S. *et al.* (2007) Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comp. Biol.,* **3**, e65.

Wilm,A. *et al.* (2006) An enhanced RNA alignment benchmark for sequence alignment programs. *Algorithms Mol. Biol.,* **1**, 19.

Xu,X. *et al.* (2007) RNA Sampler: a new sampling based algorithm for common RNA secondary structure prediction and structural alignment. *Bioinformatics,* **23**, 1883–1891.