# Methods for Multiple Alignment and Consensus Structure Prediction of RNAs implemented in MARNA

Sven Siebert and Rolf Backofen

**Abstract**

Multiple alignments of RNAs are an essential prerequisite to further analyzes such as homology modelling, motif description or illustration of conserved or variable binding sites. Beyond the comparison of RNAs on the sequence level, structural conformations determined by base-pairs have to be taken into account. Several pairwise sequence-structure alignment methods have been developed. They use extended alignment scores that evaluate secondary structure information in addition to sequence information. However, two problems for the multiple alignment step remain. First, how to combine pairwise sequence-structure alignments into a multiple alignment and second, how to generate secondary structure information for sequences whose structural information is missing. Here, we describe MARNA, its underlying methods and its usage. MARNA is an approach for multiple alignment of RNAs taking into considerations both the primary sequences and the secondary structures. It relies on the pairwise sequence-structure comparison strategy by generating a set of weighted alignment edges. This set is processed by a consistency-based multiple alignment method. Additionally, MARNA extracts a consensus-sequence and structure from this generated multiple alignment. MARNA can be accessed via the webpage
http://www.bioinf.uni-freiburg.de/Software/MARNA

**Key Words:** Multiple Alignment, RNA, sequence structure, consensus structure

## 1 Introduction

RNAs are nucleic acid polymers consisting of covalently bound nucleotides. RNA is primarily made up of four different bases: adenine, guanine, cytosine, and uracil. Single stranded RNA molecules tend to form hydrogen bonds resulting in spatial arrangements of these nucleotides. Many RNAs conserve a secondary structure of base-pairing interactions more than they conserve their sequence. Since the discovery of RNAs that act as enzymes ([1]) and the detection of huge classes of non-coding RNAs involved in regulation processes, RNAs

became more and more important. For the discovery of RNA classes, multiple sequence structure alignments are the best choice to detect RNAs with the same function. Furthermore, multiple alignments are an essential prerequisite to further analyzes such as homology modelling, motif description or illustration of conserved or variable binding sites.

Here, we want to focus on the concepts and the methods used in MARNA. MARNA is the abbreviation for *M*ultiple *A*lignment of *RNA*s. MARNA is an approach to align multiple RNAs taking into consideration both the primary sequences and the secondary structures. It is based on pairwise sequence-structure comparisons of RNAs as proposed by [5]. From these sequence-structure alignments, libraries of weighted alignment edges are generated. The weights reflect the sequential and structural conservation. For sequences whose secondary structures are missing, the libraries are generated by sampling low energy conformations. The libraries are then processed by a consistency-based multiple alignment method which is implemented in the T-Coffee system ([6]). In addition, MARNA is able to extract a consensus-sequence and -structure from a multiple alignment.

Suppose for the moment that one has a set of RNA sequences provided with secondary structures. In summary, the coarse grain of the MARNA method is as follows:

1. Generate and weight alignment edges between pairwise RNAs reflecting sequence and structure similarities.

2. Collect all weighted edges in a so-called library. This library is processed by a consistency-based multiple alignment method.

3. Find a consensus- sequence and structure from this multiple alignment.

MARNA is able to align RNAs without known conformations as well. For these sequences, several methods to assign structures to the sequences exist. MARNA is capable of integrating these methods and thus to align RNAs with initially unknown structures.

In this work, we focus on the methods used in MARNA and give some hints about parameter settings that determine the alignments, and structure choices, especially when structures are missing for some sequences. For detailed comparison studies of MARNA with related multiple alignment tools see e.g. [8, 2].

# 2 Materials

No materials given since we explain a theoretical concept.

# 3 Methods

## 3.1 Definitions

1. A *sequence* $S$ is a word over the alphabet $\{A, C, G, U\}$. $S[i]$ denotes the $i-th$ symbol in $S$.

2. An *arc* is a pair $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$ s.t. $i < j$. $i$ and $j$ are the ends of the arc. An arc represents a base-pair.

3. A base is called *free*, if it is not involved in any arc.

4. A *secondary structure* is a set of arcs $P$, s.t. for any two arcs $(i_1, j_1), (i_2, j_2) \in P$ with $i_1 < i_2$, either $i_1 < j_1 < i_2 < j_2$ or $i_1 < i_2 < j_1 < j_2$.

5. An *RNA* is a tuple $(S, P)$, where $S$ is the sequence and $P$ is the set of arcs in a secondary structure.

6. An *alignment* $A$ of two RNAs $(S_1, P_1)$ and $(S_2, P_2)$ is a subset of $\{1, \ldots, |S_1|\} \cup \{-\} \times \{1, \ldots, |S_2|\} \cup \{-\}$, where for all pairs $(i, j), (i', j') \in A$ holds

   (a) $i \leq i' \Rightarrow j \leq j'$,

   (b) $i = i' \neq - \Rightarrow j = j'$ and

   (c) $j = j' \neq - \Rightarrow i = i'$.

   Requirement: for every $i \in \{1, \ldots, |S_1|\}$ there is some $j$ with $(i, j) \in A$ (and vice versa for $j \in \{1, \ldots, |S_2|\}$).

7. The pairs $(i, j) \in A$ are called *alignment edges*.

8. An alignment edge is called *realized* if neither $i = -$ nor $j = -$.

## 3.2 Pairwise Alignment

The scoring of an alignment $A$ of two RNAs $(S_1, P_1)$ and $(S_2, P_2)$ is based on the notion of edit operations on bases as well as on arcs. We recall the edit operations as given in [5] and present a slightly modified scoring scheme to finally compute an optimal alignment between two RNAs. Optimal means to find an alignment with minimum costs assuming that the costs of an alignment are composed of the costs of all executed edit operations.

### 3.2.1 Edit operations

1. Edit operations on *free* bases are :

   (a) *base match*: The base at position $i$ in the first RNA is matched with the base at position $j$ in the second RNA, i.e. $S_1[i] = S_2[j]$. The costs are 0.

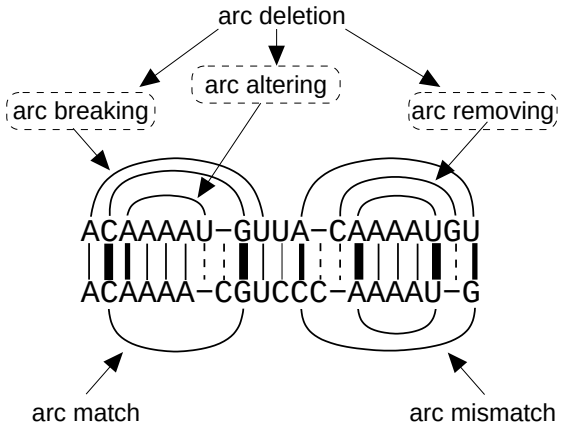



Figure 1: An alignment of two RNAs with corresponding edit operations on arcs. Alignment edges are drawn as solid lines (realized edges) and dashed lines (non-realized edges). The thickness of realized edges corresponds to similarity weights between bases. Non-realized edges are skipped for the multiple alignment step.

(b) *base mismatch*: The base at position $i$ in the first RNA is aligned with the base at position $j$ in the second RNA s.t. $S_1[i] \neq S_2[j]$. The costs are positive.

(c) *base deletion/insertion*: The base at position $i$ in the first RNA is aligned with a gap (deletion operation). The opposite case is the insertion operation. Both costs are positive.

2. Edit operations on *arcs* : Consider an arc $(i, j) \in P_1$ such that $i$ is aligned with $i'$ and $j$ is aligned with $j'$ for $i', j' \in S_2 \cup \{-\}$.

   (a) *arc match*: An arc match occurs if $i', j'$ form an arc $(i', j') \in P_2$ and $S_1[i] = S_2[i']$ and $S_1[j] = S_2[j']$.

   (b) *arc mismatch*: An arc mismatch occurs if $i', j'$ form an arc $(i', j') \in P_2$ and $S_1[i] \neq S_2[i']$ or $S_1[j] \neq S_2[j']$.

   (c) *arc deletion*: Arc deletion means that $(i', j') \notin P_2$. Depending on how many gaps the two positions $i', j'$ occupy, we may have

      i. *arc breaking*: An arc breaking occurs if none of $j$ and $j'$ equals the symbol $-$.

      ii. *arc altering*: An arc altering occurs if exactly one of $j$ and $j'$ equals the symbol $-$.

      iii. *arc removing*: An arc removing occurs if both $j$ and $j'$ are equal to $-$.

Edit operation on arcs are depicted in Figure 1. Arc costs are as follows:

1. An arc match has costs 0.

2. An arc mismatch operation has costs $w_{am}(i,j,i',j')$ for two arcs $(i,j) \in P_1$ and $(i',j') \in P_2$.

3. An arc deletion operation has costs $w_{ad}(i,j,i',j')$. These costs are determined by the bases and by the number of gaps involved. We decompose the costs $w_{ad}(i,j,i',j')$ into a sum of two single functions for the left and right ends of the arcs:

$$w_{ad}(i,j,i',j') = w^l_{ad}(i,j) + w^r_{ad}(i',j')$$

In the following, we do not distinguish between left and right arc ends, and thus introduce the function $w^e_{ad}(i,j) = w^l_{ad}(i,j) + w^r_{ad}(i,j)$. We even simplify the scoring scheme further by defining $w^e_{ad}(i,j)$ to be composed of a base match, base mismatch or base deletion together with a fixed cost for deleting an arc. Hence, we set

$$w^e_{ad}(i,j) = w_{base}(i,j) + \frac{1}{2} w^{const}_{ad},$$

where $w^{const}_{ad}$ are the costs for deleting one arc.

### 3.2.2 Alignment Algorithm

In the following, we specify our algorithm similar to the one given in [5] that computes an optimal alignment between two RNAs with given secondary structures $(S_1, P_1)$ and $(S_2, P_2)$. We introduce two simple functions:

$$\psi_s(i) = \begin{cases} 1, & \text{if base at position i not free} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$\chi(i,j) = \begin{cases} 1, & \text{if } S_1[i] \neq S_2[j] \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Here, the costs for the edit operations on free bases base match, base mismatch and base deletion are combined into a single cost function $w_{base}(i,j)$, where $w_{base}(i,j) = 0$ only if $S_1[i] = S_2[j]$. Now, we can specify the alignment algorithm:

**Input:** Two RNAs $(S_1, P_1)$ and $(S_2, P_2)$.
**Output:** Sequence Structure Alignment.
**Method:**

```
ALIGN-RNAS()
1  for a_1 = (i_1, i_2) ∈ P_1 and a_2 = (j_1, j_2) ∈ P_2
2     do for i ← i_1 + 1 to i_2 − 1
3           do for j ← j_1 + 1 to j_2 − 1
4                 do
```

$$M(i,j) = min \begin{cases} M(i-1,j) + w_{base}(i,-) + \psi_1(i)\frac{1}{2}w_{ad}^{const}, \\ M(i,j-1) + w_{base}(-,j) + \psi_2(j)\frac{1}{2}w_{ad}^{const}, \\ M(i-1,j-1) + w_{base}(i,j) \\ \qquad +(\chi_1(i) + \chi_2(j))\frac{1}{2}w_{ad}^{const} \\ M(i'-1,j'-1) + B(a_k,a_l) \\ \qquad +(\chi(i',j') * \chi(i,j))w_{am}, \\ \qquad \text{if } a_k = (i',i) \in P_1 \text{ and } a_l = (j',j) \in P_2 \end{cases}$$

$$B(a_1,a_2) = M(i_2-1, j_2-1)$$

1. We need two two-dimensional matrices, both not exceeding the size of $nm$. The matrix $B$ contains the minimum costs of aligning the intervals $(i_1+1, i_2-1)$ and $(j_1+1, j_2-1)$ for arcs $a_k = (i_1, i_2) \in P_1$ and $a_l = (j_1, j_2) \in P_2$ provided that both arcs are aligned; i.e. we have an arc match or arc mismatch. The matrix $M$ is constructed when the two arcs $a_k$ and $a_l$ are considered. It is computed within the arc intervals in almost the same manner as a sequence alignment except that arc breaking costs are considered and computed at each single base. The algorithm proceeds from inside to outside, thereby taking arcs with minimal sequence lengths first.

2. From the above algorithm it is easy to see that the time complexity of $O(n^2m^2)$ results from running over the arcs in both sequences and computing the best alignment in between. The space complexity is determined by the sizes of the two matrices $B$ and $M$.

3. The resulting alignment can be obtained by a traceback step.

### 3.2.3 Alignment Weights

The alignment algorithm computes an alignment between two RNAs which is equivalent to an edit transcript composed of edit operations weighted with edit costs. For the multiple alignment step, these costs have to be transformed into similarity weights.

1. Note that the costs are a function $d$ with positive values fulfilling the *metric* conditions:

   (a) $d(S_1, S_2) = 0 \leftrightarrow S_1 = S_2$, i.e. the costs of two RNAs $S_1$ and $S_2$ is 0 if and only if the two RNAs are equal.

   (b) $d(S_1, S_2) = d(S_2, S_1)$, i.e. the edit transcript of transforming $S_1$ into $S_2$ has the same costs as the edit transcript of transforming $S_2$ into $S_1$.

   (c) $d(S_1, S_3) \leq d(S_1, S_2) + d(S_2, S_3)$, i.e. the costs of transforming $S_1$ into $S_2$ into $S_3$ are at least so high as the costs of transforming $S_1$ into $S_3$ directly.

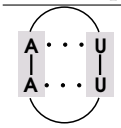
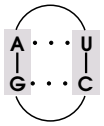
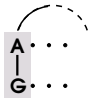
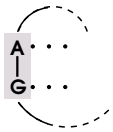
| Edit-Op | Name | Distance | Similarity |
|---|---|---|---|
|  | arc match | 0 | $4 \cdot M$ |
|  | arc mismatch | $w_{am}(A,U,G,C)$ | $4 \cdot M - w_{am}(A,U,G,C)$ |
|  | arc breaking<br>arc altering<br>(realized edge) | $w_{base}(A,G) + \frac{1}{2} w_{ad}^{const}$ | $M - w_{base}(A,G) - \frac{1}{2} w_{ad}^{const}$ |
|  | arc breaking<br>arc altering<br>(realized edge,<br>two arcs) | $w_{base}(A,G) + w_{ad}^{const}$ | $2 \cdot M - w_{base}(A,G) - w_{ad}^{const}$ |
|  | arc breaking<br>arc removing<br>(non-realized edge) | $w_{base}(A,-) + \frac{1}{2} w_{ad}^{const}$ | no realized edge |

Table 1: Edit operations on arcs together with the associated distances and their similarity values given to the T-Coffee system. Note that for arc-match and arc-mismatch, we assign half of the total similarity value to each alignment edge when building the library. Here, $w_{base}(A,C)$ are the costs for aligning $A$ with $C$ independent of whether the bases are free or not. $w_{ad}^{const}$ are the costs for deleting an arc.

2. *Transformation* from distances to similarities:

   (a) *Realized and non-realized edges*: Consider Figure 1 again. Alignment edges are constructed by means of edit operations. Non-realized edges, i.e. dashed lines in the figure, denote alignment edges that have exactly one gap at one of their ends. They are skipped for the multiple alignment step because they contain no information about aligning two nucleotides. Hence, we are left with realized edges. They are shown as thick or thin lines in the figure. The thickness corresponds to the similarity weights.

   (b) *Similarity weights*: Similarity weights are assigned to edit operations computed by the alignment algorithm. Here, we consider the number of nucleotides $r$ involved in an edit operation. We call this number the order of the edit operation. In our case, we have edit operations with

      i. $r = 4$ for an arc match or an arc mismatch,
      ii. $r = 2$ for a base match or a base mismatch and
      iii. $r = 1$ for a base deletion.

      Since we have split the arc deletion operation into two separate edit operations for the arc ends, we have an edit operation with $r = 2$ if the arc end is aligned with a nucleotide, and an edit operation with $r = 1$ if the arc end is aligned with $-$.

      The similarity weights can be achieved by choosing a maximal similarity value M, such that every value can be subtracted from the value $rM$ for each edge value. The value $M$ is multiplied by $r$ because this value is dependent on the order of the edit operations and we therefore ensure that all similarity values are positive.

### 3.3 Multiple Alignment

Now, we are ready for the multiple alignment step. Suppose we have set of $n$ RNAs together with their secondary structures. The main idea is to use the same strategy as proposed by the multiple alignment tool T-Coffee ([6]):

1. Recall that a single alignment between two RNAs provides a set of weighted, realized alignment edges.

2. The pairwise comparison strategy in a set of $n$ RNAs yields $n(n-1)/2$ alignments. All these alignments produce an amount of weighted alignment edges each reflecting the sequence structure similarity between two bases. These edges are collected in a so-called library.

3. Now, the T-Coffee strategy is performed on this data set:

(a) *Library Extension*: The library containing all pairwise alignments with their weighted alignment edges is turned into an extended library to improve all pairwise alignments by taking into considerations how all other sequences align with the current two. For instance, if we consider two RNAs specified by their alignment and their weighted alignment edges then a third sequence is considered how this sequence is aligned with the first and the second sequence. For any alignment of two RNAs $R_1$ and $R_2$, any other RNA $R_3$ is considered for improving the initial alignment. For this purpose, T-Coffee considers the alignment of $R_1$ and $R_2$ via $R_3$ by considering alignment edges from the alignment of $R_1$ and $R_3$ with edges of from the alignment of $R_2$ and $R_3$. These additional weighted edges together with the edges of the direct comparison of the first two RNAs are considered to improve this alignment by a dynamic programming approach. This procedure is executed $n(n-1)$ times, i.e. for each pairwise set of RNAs. The result is the extended library containing all improved pairwise alignments.

(b) *Progressive Alignment*: Pairwise distances of the sequence set were computed due to the alignment algorithm. They form the distance matrix which is used to produce a neighbour-joining tree ([7]) that guides the alignment process. Residue weights that are stored in the extended library are now used for this task. The two closest sequences are aligned first. This alignment is fixed and the next closest sequence is aligned to this existing alignment or two new sequences are aligned or two existing alignments are aligned. In case of aligning an already existing alignment the average score in each column is taken. Gap penalties need not be set because they are already included in the alignment as sequence identity and residue weights, i.e. residues which are aligned with gaps get a weight of zero.

### 3.4 Combining several Structures

The multiple alignment of these RNAs assumes the existence of a known structure for each RNA like e.g. an experimentally confirmed structure.

1. Whenever the structures are not known in advance, secondary structure prediction programs like Mfold([9]) and RNAfold([4]) may help to assign the minimum free structure to an RNA. The drawback here is that these structures are not necessarily the real existent structures which might be responsible for their functions.

2. In order to overcome this difficulty, we assign multiple structures to each sequence covering different folds. We call this set the ensemble of structures. We mainly use two different programs for generating these structures:

(a) RNAsubopt([4]): This program generates suboptimal structures by stochastic backtracking. The number of desired structures can be set individually.

(b) RNAshapes([3]): This program avoids the large output of similar suboptimal structures; it rather outputs structures of more fundamental differences.

3. The generated structures for each sequence $S_l$ form an ensemble, denoted $E_S^l = \{E_1^l, \ldots, E_n^l\}$. Since each structure $E_i^l$ has its own energy, it occurs with probability, say $Pr(E_i^l)$. Here, we consider rather a small set of important structures, in contrast to the explosive number of all suboptimal structures.

4. *Probability*: Due to their different energies of these structures assigned to sequence $S_l$, the probability of seeing a certain structure $E_k^l$ in a set of structures $E_{S_l}$ with restricted size $n$ is:

$$Pr(E_k^l|E_{S_l}) = \frac{Pr(E_k^l)}{\sum_{1\leq i\leq n} Pr(E_i^l)} \tag{3}$$

where $Pr(E_k^l)$ is the probability of forming structure $E_k^l$ in sequence $S_l$. In MARNA, the simplification of the uniform distribution is made, i.e. each structure has the same probability.

5. *Alignment weights*: Consider two sequences $S_1$ and $S_2$ with $n_1$ structures for the first sequence and $n_2$ structures for the second sequence. If both $n_1 = 1$ and $n_2 = 1$, then the alignment algorithm outputs weighted alignment edges as we have seen before. These alignment edges are all multiplied by 1 because the number of structures in the ensemble equals 1. Suppose we consider an ensemble of structure greater than 1, then we have to make $n_1 \times n_2$ comparisons, i.e. each combination of $(S_1, E_k^1)$ and $(S_2, E_l^2)$, $1 \leq k \leq n_1$, $1 \leq l \leq n_2$, has to be considered. The number of realized alignment edges is quadratic, i.e. proportional to $n_1 \times n_2$. The alignment weights are now influenced by the structural diversity. Each alignment edge is reweighted by the factor $Pr(E_k^1|E_{S_1})Pr(E_l^2|E_{S_2})$. If both $|E_{S_1}| = 1$ and $|E_{S_2}| = 1$, then the alignment edges are weighted by the factor 1.

## 3.5 Consensus Structure

Once we have computed the final alignment, we are ready to calculate a consensus structure from this alignment. Here, we explicitly use structure information for the calculation of the alignment. Hence, the calculation of the consensus structure should be based on these ensemble structures.

1. To exemplify the basic idea, suppose that exactly one structure per sequence is given. Each structure must then be interpreted as the "real"

known structure. A conserved base pair between two columns in the alignment is found if the majority of sequences have a base pair at the corresponding sequence positions. The remaining problem is that the resulting set of conserved base pairs alone does not form a secondary structure and is thus not a valid consensus structure. This is a problem common to all approaches for calculating a consensus structure.

2. We find a remedy by calculating a consensus secondary structure that maximizes base pair conservation. So let $c, c'$ be two columns with $1 \leq c < c' \leq m$, where $m$ is the number of columns of the multiple alignment. Furthermore, let $bp\_cons(c, c')$ be the number of sequences that have a base pair between the corresponding sequence positions. The consensus structure is then defined to be a secondary structure $P \subseteq [1..m] \times [1..m]$ such that

$$\sum_{(c,c') \in P} bp\_cons(c, c')$$

is maximized.

3. This can be calculated using dynamic programming. Let $N_{i,j}$ with $1 \leq i, j \leq m$ be the maximal base pair conservation for all columns between $i$ and $j$:

$$N_{i,j} = \max_{P} \sum_{\substack{(c,c') \in P \\ i \leq c < c' \leq j}} bp\_cons(c, c')$$

The corresponding recursion equation for $N_{i,j}$ is

$$N_{i,j} = \max \begin{cases} N_{i+1,j}, \\ N_{i,j-1}, \\ N_{i+1,j-1} + bp\_cons(i, j), \\ \max\limits_{i<k<j} \{N_{i,i+k} + N_{i+k+1,j}\} \end{cases}$$

It is a dynamic programming approach, where the traceback reports the consensus structure of the alignment.

4. Finally, we have to consider again the case where we are given structure ensembles for some (or all) sequences. Consider a multiple alignment of $K$ sequences. For each sequence $S_k$, let $E_{S_k}$ be the ensemble of structures calculated for $S_k$. For each column $c$, let $i_c^k$ be either the position that corresponds to column $c$ in sequence $S_k$ (if aligned), or $-$ otherwise. Furthermore, let $\delta_P(c, c')$ be the index function of $P$, i.e. $\delta_P(c, c')$ is 1 if $(c, c') \in P$, and 0 otherwise. Then

| edit operations | default | sequential | structural |
|---|---|---|---|
| base deletion | 2.0 | 2.0 | 0.1 |
| base mismatch | 1.0 | 1.0 | 0.1 |
| arc breaking | 1.5 | 0.1 | 1.5 |
| arc mismatch | 1.8 | 0.1 | 1.8 |

Table 2: Data sets found out for weighting sequential or structural properties or on a mixture of both (default values). The values correspond to costs which can be set in the MARNA system.

$$bp\_cons(c, c') = \sum_{k=1}^{K} \sum_{E_i^K \in E_k} \delta_P(i_c^k, i_{c'}^k) \cdot Pr[P|E_{S_k}],$$

where $Pr[P|E_{S_k}]$ is defined as given in equation 3.

# 4 Notes

1. MARNA can be tested online via the webpage
   http://www.bioinf.uni-freiburg.de/Software/MARNA/index.html.
   MARNA is also available as a downloadable file (see webpage).

2. MARNA offers mainly two choices to adjust your alignments:

   (a) *Parameter settings*: MARNA relies on the comparison of pairwise RNAs. These comparisons are accomplished by alignments with costs assigned to edit operations on bases and arcs. These costs can be set individually.

   (b) *Structure computation*: The alignment of RNAs take into account both the primary sequences and the secondary structures. The easiest case is when the secondary structures are known in advance, and the computation is reduced to find common sequential and structural properties. Otherwise, the structures have to be found. MARNA provides in addition to user-defined structures the assignment of different kinds of structures. These include the assignment of minimum free energy structures, shaped structures or an ensemble of low energy structures.

3. *Parameter settings*: Parameters can be set individually depending on weighting some edit operations more or less. A series of tests has brought three data sets to obtain alignments based on sequential or structural properties or on a mixture on both. These data sets are shown in Table 2.

4. Parameters settings influence the resulting alignments. Choose the default parameter settings first. It has been confirmed that this data set recognizes conserved sequential and structural properties very well.

5. Beyond the parameter settings, the assignment of different structures to the sequences are quite important as well. The easiest case is when user-defined structures are given as input.

6. Structure Choice: Here are some hints to choose the right structure assignments if no structures are given to the sequences.

   (a) If the RNAs are sequentially related and have nearly the same length then choose the minimum free energy structures.

   (b) The shaped structures are suited to cover a lot of diverse structural conformations for each single sequence. Choose shape structures, if no clear consensus structure is observable at first glance.

   (c) The ensemble set of low energy conformations is best chosen if you guess that these RNA sequences resemble structurally in some way. An ensemble consists of multiple structures. This ensemble contains similar structures if almost all suboptimal structures are similar.

7. The running time of MARNA crucially depends on the structure choices. Suppose n RNAs of nearly the same length without structure specifications are given. If the mfe structures are chosen that are assigned to the sequences then the multiple alignment and the consensus structure computation can be done in reasonable time. Suppose you choose en ensemble of three suboptimal structures to each RNA, then the computation time is ninefold because for each pair of RNAs nine pairwise sequence structure comparisons have to be made.

# References


[1] Jennifer A. Doudna and Thomas R. Cech. The chemical repertoire of natural ribozymes. *Nature*, 418(6894):222–8, 2002.

[2] Paul P. Gardner and Robert Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5:140, 2004.

[3] Robert Giegerich, Bjrn Voss, and Marc Rehmsmeier. Abstract shapes of RNA. *Nucleic Acids Research*, 32(16):4843–51, 2004.

[4] Ivo L. Hofacker, Walter Fontana, Peter F. Stadler, Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte Chemie*, 125:167–188, 1994.

[5] Tao Jiang, Guohui Lin, Bin Ma, and Kaizhong Zhang. A general edit distance between RNA structures. *Journal of Computational Biology*, 9(2):371–88, 2002.

[6] C. Notredame, D. G. Higgins, and J. Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205–17, 2000.

[7] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–25, 1987.

[8] Sven Siebert and Rolf Backofen. MARNA: multiple alignment and consensus structure prediction of RNAs based on sequence structure comparisons. *Bioinformatics*, 21(16):3352–9, 2005.

[9] M. Zuker. Prediction of RNA secondary structure by energy minimization. *Methods in Molecular Biology*, 25:267–94, 1994.