BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 15 2006, pages 1823–1831
doi:10.1093/bioinformatics/btl194

Structural bioinformatics

INFO-RNA—a fast approach to inverse RNA folding

Anke Busch and Rolf Backofen*

Albert-Ludwigs-University Freiburg, Institute of Computer Science, Chair of Bioinformatics, Georges-Koehler-Allee 106, 79110 Freiburg, Germany

Received on March 14, 2006; revised on April 28, 2006; accepted on May 15, 2006
Advance Access publication May 18, 2006
Associate Editor: Martin Bishop

ABSTRACT

Motivation: The structure of RNA molecules is often crucial for their function. Therefore, secondary structure prediction has gained much interest. Here, we consider the inverse RNA folding problem, which means designing RNA sequences that fold into a given structure.

Results: We introduce a new algorithm for the inverse folding problem (INFO-RNA) that consists of two parts; a dynamic programming method for good initial sequences and a following improved stochastic local search that uses an effective neighbor selection method. During the initialization, we design a sequence that among all sequences adopts the given structure with the lowest possible energy. For the selection of neighbors during the search, we use a kind of look-ahead of one selection step applying an additional energy-based criterion. Afterwards, the pre-ordered neighbors are tested using the actual optimization criterion of minimizing the structure distance between the target structure and the mfe structure of the considered neighbor.

We compared our algorithm to RNAinverse and RNA-SSD for artificial and biological test sets. Using INFO-RNA, we performed better than RNAinverse and in most cases, we gained better results than RNA-SSD, the probably best inverse RNA folding tool on the market.

Availability: www.bioinf.uni-freiburg.de?Subpages/software.html

Contact: backofen@informatik.uni-freiburg.de

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

1 INTRODUCTION

RNAs are involved in translation (tRNA, rRNA), splicing (snRNA), processing of other RNAs (snoRNA, RNAseP) and regulatory processes (miRNA, siRNA) (Hüttenhofer *et al*., 2002). Furthermore, parts of mRNAs can adopt structures that regulate their own translation [SECIS (Hüttenhofer and Böck, 1998; Liu *et al*., 1998), IRE (Addess *et al*., 1997)]. The function of RNA molecules often depends on both the primary sequence and the secondary structure. Since prediction or experimental determination of three-dimensional RNA structures remain difficult, much work focuses on problems associated with its secondary structure, which can be described as a set of paired positions of the RNA sequence. These positions are assigned to complementary bases according to Watson and Crick (A and U, C and G). In some cases, other pairings (e.g. G and U) can be found. The problem of predicting the secondary structure of an RNA is called the RNA folding problem. Existing computational approaches are based on a thermodynamic model that gives a free energy value for each secondary structure (Zuker, 1994). The structure with the lowest energy [called the minimum free energy (mfe) structure] is expected to be the most stable one.

In this paper, we consider the inverse RNA folding problem, which is the design of RNA sequences that fold into a desired structure. This design is applicable to ribozymes and riboswitches (Knight, 2003; Winkler *et al*., 2004; Cech, 2004), which may be used as drugs in research and medicine. Furthermore, the inverse RNA folding can be applied to the design of noncoding RNAs, which are involved in a large variety of processes, e.g. gene regulation, chromosome replication and RNA modification (Storz, 2002).

Given an RNA secondary structure, we aim at finding an RNA sequence that is going to adopt this structure. Straight forwardly testing each sequence, whether its mfe structure is the searched one, is impossible since the number of sequences grows exponentially in the size of the structure (Hofacker, 1994). Thus, different heuristic local search strategies, which do not analyze the complete solution space, were used by existing programs dealing with inverse RNA folding (Hofacker *et al*., 1994; Andronescu *et al*., 2004; Dirks *et al*., 2004). One approach to inverse folding is implemented in RNAinverse, which is included in the Vienna RNA Package (Hofacker *et al*., 1994). There, the strategy of adaptive walk is used and local optima are found according to two different criteria, namely a structural distance between the mfe structure of the designed sequence and the target structure (*mfe-mode*) and the probability of folding into the target structure (*p-mode*). A second algorithm is called RNA-SSD (RNA Secondary Structure Designer) and was developed by Andronescu *et al*. (2004). It is based on a recursive stochastic local search that also tries to minimize a structure distance.

We present a new algorithm INFO-RNA for the INverse FOlding of RNA. It consists of two steps; a new design method for good initial sequences and a following improved stochastic local search that uses an effective neighbor selection method. Concerning the initialization step, we found out that a good choice is to use a sequence that among all sequences adopts the given structure with the lowest possible energy. We present a dynamic programming approach to solve this problem. Here, multi-branched loops (short: multiloops) are especially complicated to handle. For the selection of neighbors during the local search, we deviated from the arbitrary order used in RNAinverse and RNA-SSD. Using a kind of look-ahead of one selection step, we first order the set of neighbors using an energy-based criterion, which is much faster to calculate than the actual optimization criterion of minimizing the structure

*To whom correspondence should be addressed.

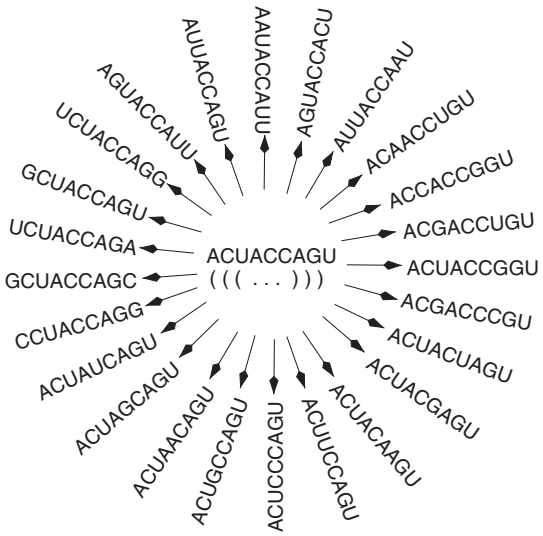

Fig. 1. Exemplary sequence (and structure) and all its sequence neighbors that can adopt this structure. The parenthesis notation is used for the representation of the RNA secondary structure. An opening '(' and a closing parenthesis ')' stand for a base pair, a dot '.' represents an unbound position.

distance between the target structure and the mfe structure of the considered neighbor. Afterwards, the neighbors are tested in the calculated order using the actual optimization criterion. We tested INFO-RNA on artificial as well as on real data and compared the results to the ones of RNA-SSD and RNAinverse.

2 THE INFO-RNA APPROACH

The general problem of inverse RNA folding can be described as follows. Find an RNA sequence $S = S_1 \ldots S_n$ of length n that folds into a given secondary structure T, where $S_i \in B = \{A, C, G, U\}$ for $1 \leq i \leq n$. T can be described as a set of pairs (i_1, i_2), where $1 \leq i_1 < i_2 \leq n$ and positions i_1 and i_2 are paired. In the following, all regarded secondary structures are pseudoknot-free, where a structure T is called pseudoknot-free if for every 2 base pairs (bp) (i_1, i_2) and (j_1, j_2) in T holds $i_1 < i_2 < j_1 < j_2$ or $i_1 < j_1 < j_2 < i_2$. We have to analyze a search space of an exponentially high number of valid RNA sequences. These are sequences that can form the base pairs required for the target structure regardless of energy. Therefore, it is not possible to find a globally optimal solution by testing all candidate sequences and thus, local search methods are widely used to address the inverse folding problem. Consequently, the resulting local optima are not guaranteed to be globally optimal but are optimal among all their sequence neighbors. The sequence neighbors of a sequence S are all sequences S^i that differ from S in one unbound position or in two positions which have to pair concerning structure T (Fig. 1).

Except on the search strategy itself, the performance of the local search depends on the quality of the initializing sequence. Often, it is chosen at random. In the following, we are going to introduce a new method to create an excellent initializing sequence and describe the local search strategy we used.

2.1 The initializing step

The initializing step of INFO-RNA uses the technique of dynamic programming. This method was successfully applied to RNA

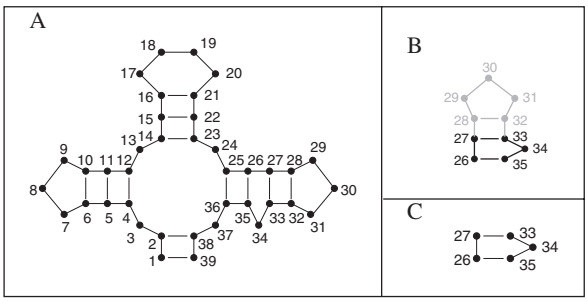

Fig. 2. (A) RNA secondary structure. (B) Substructure $T_{(26,35)}$ having a connected backbone. (C) Structural element $T^{(27,33)}_{(26,35)}$ without a connected backbone.

secondary structure prediction, e.g. by Zuker and Stiegler (1981), and related problems. Similar to Zuker and Stiegler, we use free energies of structural elements [stacks, bulge- (BL), interior- (IL), hairpin- (HL), multiloops (ML)]. They depend on the size of the loop, the closing base pairs, and on the free bases inside the loops and adjacent to the closing pairs. Since each pair belongs to two elements, neighbored elements in a structure are linked and base pairs cannot be handled independently. Additionally, their energy fraction depends on directly adjacent free bases. Free bases that are not adjacent to a base pair do not give any energy fraction. The free energy value of a pseudoknot-free structure is calculated by adding up all partial energies of its elements.

Given a target structure T, we find a sequence S that among all sequences adopts T with the lowest possible energy. Formally, this means that we find a sequence S resulting from $\text{argmin}_{S'}\, e(S', T)$ where $e(S', T)$ represents the free energy of sequence S' folded into structure T. For solving this problem, our dynamic programming algorithm needs linear time depending on the structure size. It starts with small substructures and enlarges them gradually by 1 bp. Thus, the algorithm starts at the closing pair of a hairpin loop, subsequently fixes it to pair assignments out of the set of valid pairs BP = {A-U,C-G,G-C,U-A,G-U,U-G}, and assigns the unbound positions of the loop such that they provide the lowest possible energy value for this small substructure under the condition that the closing pair is fixed. This is stored for all six possible assignments of the pair. Afterwards, the next pair to the HL-closing one is fixed. The energy can be calculated by the sum of the energy of the hairpin loop including the closing pair and the stacking energy of the current pair and the closing one of the HL. To find the best energy value, we have to minimize this sum over all possible assignments of the base pair closing the HL. This is demonstrated in equation 1 exemplarily, where $e(.)$ represents the mfe. It refers to Figure 2A.

$$e\left(\begin{array}{c}\text{18 19}\\ \text{17}\,\square\,\square\,\text{20}\\ \text{16}\,\square\,\text{21}\\ \text{A}_{15}\!-\!\text{U}_{22}\end{array}\right) = \min\left\{\begin{array}{l} e\left(\begin{array}{c}\text{18 19}\\ \text{17}\,\square\,\square\,\text{20}\\ \text{A}_{16}\!-\!\text{U}_{21}\end{array}\right) + e\left(\begin{array}{c}\text{A}_{16}\!-\!\text{U}_{21}\\ \text{A}_{15}\!-\!\text{U}_{22}\end{array}\right)\\ e\left(\begin{array}{c}\text{18 19}\\ \text{17}\,\square\,\square\,\text{20}\\ \text{U}_{16}\!-\!\text{A}_{21}\end{array}\right) + e\left(\begin{array}{c}\text{U}_{16}\!-\!\text{A}_{21}\\ \text{A}_{15}\!-\!\text{U}_{22}\end{array}\right)\\ \vdots\\ e\left(\begin{array}{c}\text{18 19}\\ \text{17}\,\square\,\square\,\text{20}\\ \text{U}_{16}\!-\!\text{G}_{21}\end{array}\right) + e\left(\begin{array}{c}\text{U}_{16}\!-\!\text{G}_{21}\\ \text{A}_{15}\!-\!\text{U}_{22}\end{array}\right)\end{array}\right\}. \qquad (1)$$

We define a substructure $T_{(i_1, i_2)}$ as structural part of T that is closed by pair (i_1, i_2) and has a connected backbone (Fig. 2B). $e(T_{(i_1, i_2)} \,|\, (S_{i_1}, S_{i_2}) \rightarrow (a_1, a_2))$ is defined to be its mfe under the

condition that the sequence positions of the closing pair of the substructure (S_{i_1}, S_{i_2}) are fixed to a base pair assignment (a_1, a_2). Furthermore, we define a structural element $T_{(i_1,i_2)}^{(i_1+k,\,i_2-l)}$ as part of structure T that consists of only two neighboring pairs (i_1, i_2) and $(i_1 + k, i_2 - l)$. These ones close the element. It does not have a connected backbone (Fig. 2C). In case of a ML, the structural element is closed by >2 bp and the definition is applied analogously. We define $e(T_{(i_1,i_2)}^{(i_1+k,\,i_2-l)} \mid (S_{i_1}, S_{i_2}) \rightarrow (a_1, a_2), (S_{i_1+k}, S_{i_2-l}) \rightarrow (b_1, b_2))$ as the mfe of the structural element that is closed by base pairs (i_1, i_2) and $(i_1 + k, i_2 - l)$ whose sequence positions are fixed to assignments (a_1, a_2) and (b_1, b_2), respectively. Equation (2) formalizes the example of Equation (1).

$$e(T_{(15,22)} | (S_{15}, S_{(22)}) \rightarrow (A, U)) = \min_{(a_1,a_2)} \left\{ \begin{array}{l} e(T_{(16,21)} | (S_{16}, S_{21}) \rightarrow (a_1, a_2)) \\ + e\left(T_{(15,22)}^{(16,21)} \middle| \begin{array}{l} (S_{16}, S_{21}) \rightarrow (a_1, a_2) \\ (S_{15}, S_{22}) \rightarrow (A, U) \end{array} \right) \end{array} \right\}. \quad (2)$$

The energy value of any element depends on the assignment of the base pairs and, in case of a loop, on the unpaired bases adjacent to a stem and the loop size. The minimal energy of a substructure $T_{(i_1,\,i_2)}$ can be evaluated by adding the minimum energy of the one pair smaller substructure $T_{(i_{1+k},\,i_{2-1})}$ and the energy of the structural element $T_{(i_1,i_2)}^{(i_1+k,\,i_2-l)}$. Therefore, an already analyzed smaller substructure can be seen as black box, except for its closing pair. We calculate the lowest possible energies for substructures gradually by adding the next pair to a smaller substructure.

Of course, the question arises in which order the base pairs should be handled. For that purpose, we define an order $\prec$ in which base pairs are analyzed. $(i_1, i_2) \prec (j_1, j_2)$ means that base pair (i_1, i_2) is analyzed prior to base pair (j_1, j_2). The actual order in which the base pairs of the target structure are examined is defined as follows.

$$(i_1, i_2) \prec (j_1, j_2) \quad \text{if and only if} \quad i_1 > j_1 \quad (3)$$

Relating to the example of Figure 2A, the order of all pairs is the following: (28, 32) $\prec$ (27, 33) $\prec$ (26, 35) $\prec$ (25, 36) $\prec$ (16, 21) $\prec$ (15, 22) $\prec$ (14, 23) $\prec$ (6, 10) $\prec$ (5, 11) $\prec$ (4, 12) $\prec$ (2, 38) $\prec$ (1, 39).

All pairs that are part of the structural element that is closed by the current pair and that are smaller than the current one (concerning the order) are denoted as predecessors of the current pair. Since closing pairs of HLs do not depend on any other pair, they have no predecessor. The closing pair of a ML has as many predecessors as stems originate from the loop. All other pairs have exactly one predecessor. Table 1 shows the predecessors of Figure 2A.

Having set the order of all pairs, a dynamic programming matrix D is filled with minimal free energies. Each row in D represents a base pair of the target structure while each column stands for a possible assignment of the pairs. Thus, D has as many rows as pairs are in our desired structure and six columns, which represent the assignments A-U, U-A, C-G, G-C, G-U and U-G. In the following, pairs are no longer represented by their pairing positions, e.g. (i_1, i_2), but only by their numbers in D, e.g. i. The values in the matrix $D(i,a)$ give the mfe of a substructure ending at base pair i (represented by the row) that is assigned to $a \in BP$ (given by the column). Every substructure starts at one or more base pairs that do not have any predecessors.

Before giving a detailed description of the algorithm, we have to define some variables and notations that are used in the following equations. Now, T_i^j represents the structural element of T between base pairs i and j where i and j are row numbers in D. The free energy of the structural element between base pairs i and j assigned to a and b, respectively, is given by $e(T_i^j / i \rightarrow a, j \rightarrow b)$. Further definitions are shown in Table 2.

Table 1. Base pairs and their predecessors of the structure of Figure 2A

Base pair	Predecessor(s)
(28,32)	None
(27,33)	(28,32)
(26,35)	(27,33)
(25,36)	(26,35)
(16,21)	None
(15,22)	(16,21)
(14,23)	(15,22)
(6,10)	None
(5,11)	(6,10)
(4,12)	(5,11)
(2,38)	(4,12), (14,23), (25,36)
(1,39)	(2,38)

Table 2. Definition of variables

$p_k(i)$	k-th predecessor of pair i (sorted according to the order)
s	Number of stems originating from a ML (=number of predecessors of the closing base pair of the ML)
f	Number of free bases adjacent to stems in a ML
F	Total number of free bases in a ML
$e_{\mathrm{ML}}(s,F)$	Size-depending energy fraction of a ML with F free bases and s stems
$e^{sbs}(b)$	Single base stacking energy of a free base assigned to b and adjacent to one or two stems in a ML
H	Total number of free bases in a HL
$e_{\mathrm{HL}}(H)$	Size-depending energy fraction of a HL of size H
$e^{\mathrm{bonus}}_{a,b1,\ldots,bH}$	HL bonus energy depending on the assignment of the closing pair and the free bases. It is lower than 0 for some special tetra HLs. Otherwise it is set to 0.
$e_{\mathrm{TM}}(a,b_1,b_H)$	Terminal stacking and mismatch energy in HLs. It depends on the assignment a of the closing pair of the HL and the assignment of the directly adjacent free bases b_1 and b_H
$e_{\mathrm{AU}}(i,a)=$	Terminal AU penalty. It penalizes stems, whose last pair is assigned to A and U or G and U $e_{\mathrm{AU}}(i,a) = \begin{cases} 0.5, & \text{if } i \text{ is the last pair of a stem and a} \in \{\text{A–U, U–A, G–U, U–G}\} \\ 0, & \text{otherwise} \end{cases}$

During our dynamic programming approach, the fields in the matrix are filled row by row. Depending on the kind of the pair, i.e. on the number of predecessors, the values are calculated as follows.

(A) If base pair i has exactly one predecessor, i.e. it is a closing pair of a bulge loop, of an interior loop, or of a stack,

$$\forall a \in BP: \quad D(i,a) = e_{\mathrm{AU}}(i,a) + \min_{b \in BP} \left\{ D(i-1,b) + \min_{\substack{\text{assignment of free} \\ \text{bases in } T_i^{i-1} \text{ that are} \\ \text{adjacent to } i-1 \text{ or } i}} e\left(T_i^{i-1} \middle| \begin{array}{c} i \rightarrow a \\ i-1 \rightarrow b \end{array} \right) \right\},$$

where $e(T_i^{i-1} \mid i \rightarrow a, i-1 \rightarrow b)$ gives the energy of the structural

element between pairs $i-1$ and i assigned to a and b. Besides on a and b, this energy value depends on the assignment of the free bases directly adjacent to $i-1$ and i. Thus, both mentioned effects can be seen here: the dependency of the base pairs to each other and the dependency to the adjacent free bases.

(B) If base pair i has more than one predecessor, i.e. it is a closing pair of a multiloop, $\forall\ a \in BP$:

$$D(i,a) = e_{\mathrm{ML}}(s,F) + e_{\mathrm{AU}}(i,a) + \min_{\substack{a_1,\ldots,a_s \in BP \\ b_1,\ldots,b_f \in B}} \left\{ \sum_{k=1}^{s} D(p_k(i),a_k) + \sum_{j=1}^{f} e^{sbs}(b_j) \right\},$$

where the minimum is taken over all possible assignments of all predecessor base pairs $a_1,\ldots,a_s$ and of all free bases $b_1,\ldots,b_f$ adjacent to them. Straight forwardly implemented, this evaluation can be exponential in the number of stems originating from the ML and the number of adjacent free bases since the energy fraction of a free base adjacent to two stems depends on the assignments of both. But since usually only MLs with a low number of stems occur, even the naive solution is usable in practice. However, this complexity can be reduced to linear time for all MLs by introducing a further dynamic programming matrix M for each ML. It calculates the best free energy of the substructure closed by the closing pair of the ML dynamically. The evaluation of the ML starts with the first stem-ending pair according to the order of the pairs. The order is defined as given in Equation (3), but the definition of the predecessors is renewed. Now, pair i is predecessor in an ML to pair j iff $i \prec j$ and it does not exist any pair k in the ML such that $i \prec k \prec j$.

Matrix M includes a row for each stem-ending base pair except the one that is closing the ML. The matrix has to be re-calculated for each possible assignment of the closing pair of the loop since this pair is fixed and a kind of predecessor for the first stem-ending pair. In each step, the best energy of the part of the ML that includes the current base pair j and all pairs i with $i \prec j$ is evaluated. To this aim, all assignments of the previous base pair as well as of the stem-adjacent free base(s) between the current and the previous pair have to be taken into account. This has to be done, since the energy fraction of a free base depends on the assignment of all adjacent base pairs. Thus, we have to differentiate between the two cases given in Equations (4) and (5). There, i represents the base pair and the associated assignment is denoted as a.

The energy fraction of a free base assigned to t and adjacent left (l) or right (r) to base pair j assigned to b is given by the single base-stacking energy $e_l^{sbs}(t,j,b)$ and $e_r^{sbs}(t,j,b)$, respectively.

(I) If there is only one free base between the current base pair i and its predecessor i_{p},

$$M(i,a) = \min_{\substack{b \in BP \\ t \in B}} \left\{ \begin{array}{c} \min\{e_r^{sbs}(t,i,a), e_l^{sbs}(t,i_{\mathrm{p}},b)\} \\ + M(i_{\mathrm{p}},b) + D(i_{\mathrm{p}},b) \end{array} \right\}, \qquad (4)$$

where b denotes the assignment of base pair i_{p}, t represents the assignment of the free base between i_{p} and i.

(II) If there are more than one bases between the current base pair i and its predecessor i_{p},

$$M(i,a) = \min_{\substack{b \in BP \\ t_{\mathrm{p}} \in B}} \left\{ \begin{array}{c} e_l^{sbs}(t_{\mathrm{p}},i_{\mathrm{p}},b) + M(i_{\mathrm{p}},b) + D(i_{\mathrm{p}},b) \\ + \min_{t_c \in B} e_r^{sbs}(t_c,i,a) \end{array} \right\} \qquad (5)$$

where b denotes the assignment of base pair i_{p}, t_{p} and t_{c} represent the assignments of the free bases adjacent to i_{p} and to i, respectively.

The calculation of the first base pair in the ML (represented by the first row of M) works analogously. Here, the closing pair of the ML acts as its predecessor with a fixed assignment. Finally, the entry for the closing pair of the ML in D is evaluated analogously to Equations (4) and (5) depending on the entries of matrix M.

(C) If base pair i has no predecessor, i.e. it is a closing pair of a hairpin loop, $\forall\ a \in BP$:

$$D(i,a) = e_{\mathrm{HL}}(H) + \min_{b_1,\ldots,b_H \in B} \left\{ e^{\mathrm{bonus}}_{a,b_1,\ldots,b_H} + \left\{ \begin{array}{ll} e_{\mathrm{AU}}(i,a), & H=3 \\ e_{\mathrm{TM}}(a,b_1,b_H), & H>3 \end{array} \right\} \right\},$$

where the minimum is taken over all possible assignments of all free bases $b_1,\ldots,b_H$ in the HL.

Having filled the complete matrix D, we finally aim at finding the sequence that adopts the given structure with the lowest possible energy. For that purpose, we choose the smallest energy value of the last row of D. It gives the mfe a sequence can have, when folding into the target structure. To find the sequence that provides this energy, we are going back through matrix D along the path of the best predecessor assignments. For this reason, we store traceback pointers during the computation of D. Finally, all free bases that are not directly adjacent to a base pair and thus not giving any energy value are chosen arbitrarily.

Using this dynamic programming algorithm, we obtain a sequence that among all sequences adopts the target structure with the lowest possible energy. There is no other sequence that has lower energy when folding into this structure. Nevertheless, the sequence is not guaranteed to fold into it since actually this sequence can have even less energy when folding into another structure. Therefore, the resulting sequence is processed further in a second step.

Complexity. Filling matrices D and M, and generating the traceback are the things to do during the initialization. Since D has six entries per row and at most n/2 rows, it consists of at most $3n$ values. Hence, we have to check what time is needed per entry. For that purpose, we differentiate between the three kinds of entries in D depending on the corresponding base pair (A,B,C). During the calculation of values corresponding to pairs having exactly one predecessor (type A), the minima are taken over all assignments of the predecessor and the adjacent free bases. Thus, at most $6 * 4^4$ steps are needed. As already mentioned, straight forwardly, the complexity for entries representing a closing pair of a ML (type B) is exponentially high in the number of stems that form the loop and the number of free bases adjacent to them. Since usually only MLs with a low number of stems occur, even the naive solution is usable in practice. Nevertheless, this complexity can be reduced by using a separate dynamic programming matrix M for the evaluation of MLs. By doing this, all closing pairs of all MLs can be analyzed in at most linear time overall. Last but not least the complexity of type C entries has to be considered. Here, the entry corresponds to a closing pair of a HL. Thus, only in case of a tetra-loop, the assignment of all bases in the loop is important. For larger HLs, only the assignment of the free bases adjacent to the closing pair are taken into account. Therefore, the calculation of fields of type C needs at most 4^4 steps.

Thus, each entry in D of type A or C can be calculated in constant time and hence, evaluating all of them needs $O(n)$ time.

Furthermore, the calculation of all entries for pairs closing a ML can also be done in $O(n)$ time. Consequently, the whole matrix D can be evaluated in linear time as well as the generation of the traceback.

2.2 The local search step

After generating the start sequence, local optima are found by mutating iteratively. For that purpose, neighbored sequences are tested to check whether they provide a better value according to an objective function. In INFO-RNA, we use the objective function of minimizing the structure distance between the mfe structure of the designed sequence and the target structure (mfe-mode) as used in RNAinverse. This structure distance is defined as the number of differentially paired or unpaired bases. Furthermore, we also optimize small substructures first and proceed them to larger ones as done by Hofacker *et al*. (1994), since running the optimization directly on the full-length sequence would take too much time. The idea is that a substructure, which is optimal for a subsequence, will appear in the full sequence with enhanced probability even if this is not assured.

In INFO-RNA, the local search is a stochastic local search (SLS) (Hoss, 1998). This strategy has a lot in common with the widely used search strategy of adaptive walk (AW), which moves to the first found neighbor of the sequence that has an mfe structure with a lower structure distance to the target structure than the current one. But whereas the AW often gets stuck in local optima (sequences which are better than all their neighbors but not necessarily the globally best solution), the stochastic local search is allowed to move to worse neighbors with a fixed probability p_w to overcome local optima. A tested neighbor is retained if its mfe structure has a smaller distance to the target structure than the current one. Otherwise, it is kept with probability p_w. We set p_w to 0.1 since this has turned out to be the best value during our experiments. The search terminates after a fixed number of steps. We set this number to 10 times the length of the structure. As moves to worse neighbors are allowed, the last sequence is not necessarily the best one found. Thus, the best found sequence is stored during the search and finally given.

During the search, not all sequence neighbors are candidates for mutation. Only positions that do not pair correctly and positions adjacent to those are tested. While in RNAinverse these neighbors are tested in arbitrary order, during INFO-RNA, the order can be chosen depending on a look-ahead of one selection step. Thereto, the energy of each candidate sequence folded into the target structure is calculated. Then, the resulting energy difference to the current one is evaluated. Let $e(S, T)$ be the energy of sequence S folded into the wanted structure T. Let S' be a neighbor of S and $e(S', T)$ the energy of S' when folding into T. Then, the energy difference is given by $e(S, T) - e(S', T)$. The higher the difference is, the earlier the neighbor is examined according to the actual optimization criterion. Using INFO-RNA, the order of testing the neighbors can be chosen depending on energy as described above (NE-mode) or arbitrarily (NA-mode) as done in RNAinverse. This pre-selection step can be done easily, since all structural elements contribute additively to the energy of the whole structure and thus, only structural elements that are closed by the mutated pair or include a mutated free base have to be re-evaluated. After fixing the test order of the neighbors, they are evaluated concerning the actual optimization criterion of minimizing the structure distance of the mfe structure of the sequence and the target structure (mfe-mode). To evaluate the folding energy, we use functions from the Vienna RNA Package.

3 RESULTS

We evaluated the performance of INFO-RNA and compared it with two other inverse RNA folding algorithms: RNAinverse from the Vienna RNA Package (Hofacker *et al*., 1994) and RNA-SSD (Andronescu *et al*., 2004). For that purpose, we chose several artificially generated RNA structures as well as some test sets containing real biological data. To make our results comparable, we chose some biological test sets similar to that of Andronescu *et al*. (2004). We used an executable of RNA-SSD, kindly provided by the authors, and repeated the tests they had done, since first, RNA-SSD was improved since its publication and second, we used faster PCs.

When using RNAinverse, maximizing the probability of folding into the target structure (p-mode) often gives better results than minimizing the structure distance between the mfe structure of the designed sequence and the target structure (mfe-mode) but the former works only for short structures up to a size of ~200 since it operates on the whole structure. Thus, we always chose the mode of RNAinverse that gives a result at all and if both modes gained a solution, the better one was chosen. We call a run successful, if the mfe structure of the final sequence is the target one. Otherwise, it is denoted as unsuccessful.

All computations were done on PCs with 3 GHz Intel Pentium 4 processors and 2 GB RAM. Since all tested algorithms are nondeterministic, we performed multiple runs on each problem instance. In the following, all given runtimes E_T denote expected times required for finding a solution. They are given in CPU seconds and calculated in the same way as done by Andronescu *et al*. (2004) by $E_S + (1/f_s - 1)\, E_U$, where E_S and E_U represent the average time needed for successful and unsuccessful runs, respectively. The fraction of successful runs is given by f_S. A problem arises if f_S is 0. Then the expected time E_T is set to infinity, which means that a solution will never be found. In the following tables, we indicate these cases by a –. During our tests, INFO-RNA will be used in mfe-mode in combination with the NE-mode, if nothing further is mentioned.

3.1 Artificial test sets

Our first three test sets consist of artifically generated structures. For that purpose, we generated RNA structures with some user-given structural features, e.g. the overall size of the structure, loop sizes and the length of the stems. For all sizes, minimal and maximal values are fixed. A structure generator chooses values among valid sizes as well as structural elements at random. The values we used are summarized in the Supplementary Data.

We generated two test sets of 300 structures each. Test set Ia consists of short structures up to a length of 200, while test set Ib includes larger structures of sizes between 300 and 700. Even if test sets Ia and Ib also include multiloop structures, we additionally analyzed two small but complex ones in test set Ic. Structure Ic-1 turned out to be hard to design because it includes a stem just consisting of only 1 bp. None of the tested programs managed it to design a sequence folding into this structure. Structure Ic-2 differs from Ic-1 just in the challenging stem, which consists of 3 bp here (Fig. 3). Our results show that this slight difference made the structure much easier to design.

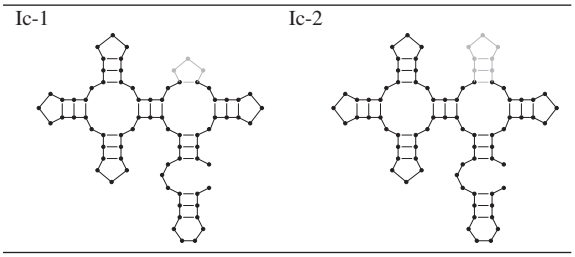

Fig. 3. Special ML structures of Ic differing in the part given in gray.

Table 3. Results for test sets Ia and Ib

Name	INFO-RNA	RNA-SSD	RNAinverse
Ia (CSR)	300/300	298/300	294/300
Ia ($\bar{E}_T$)	0.1	0.2	41.9
Ib (CSR)	300/300	294/300	1/300
Ib ($\bar{E}_T$)	9.1	46.8	–

Results for INFO-RNA, RNA-SSD and RNAinverse concerning success and speed on artificial test sets Ia and Ib. The complete succession rate (CSR) gives the fraction of structures for which the respective algorithm found a solution in all runs done for each structure. $\bar{E}_T$ represents the average expected time needed to compute a solution (given in CPU seconds).

Using the artificial test sets Ia–c, we analyzed success and speed of INFO-RNA, RNA-SSD and RNAinverse. Please note that due to the large sizes of structures included in test set Ib the p-mode of RNAinverse was not applied to this test set. For test set Ia, we examined each structure 100 times with each algorithm and tested for how many structures the respective algorithm succeeded in all 100 cases. The same was done for test set Ib, but here, each structure was examined only 10 times. Table 3 summarizes the results. INFO-RNA was always successful for all 300 structures of Ia as well as Ib. For small structures (Ia), RNA-SSD and RNAinverse performed only a little worse. RNA-SSD needed twice as long and RNAinverse 400 times as long as INFO-RNA did. For test set Ib, which includes larger structures, RNA-SSD also performed only a little worse than INFO-RNA but was much slower.

The structures of Ic were examined 100 times each. The resulting succession rates and expected times are given in Table 4. Since for structure Ic-1 all algorithms failed in all cases, no times are given. But all three algorithms designed sequences whose mfe structures are in a small structure distance to the target one. In Table 4, the succession rates in parentheses give the fraction of sequences whose mfe structures have a distance of two to the target one. For structure Ic-2, the fraction of successful runs differs among all three algorithms. While INFO-RNA failed in only one run (out of 100), RNA-SSD did not succeed in 38 cases. Furthermore, the latter is >3000 times slower than INFO-RNA. It can be summarized that, for test set Ic, INFO-RNA has a better succession rate than the other two algorithms and is much faster.

3.2 Biological test sets

Computationally predicted structures for known RNA sequences. In order to test the performance for real biological data, we used two further test sets. These ones consist of structures that are computationally predicted for known RNA sequences. All structures were predicted by RNAfold from the Vienna RNA Package (Hofacker *et al*., 1994), the same procedure that is used to evaluate the foldings during INFO-RNA, RNA-SSD and RNAinverse. Thus, it is guaranteed that at least one sequence exists, whose mfe structure is the analyzed one.

Table 4. Results for test set Ic

Name	INFO-RNA	RNA-SSD	RNAinverse
Ic-1(SR)	0/100	0/100	0/100
Ic-1(E_T)	—	—	—
Ic-1(2)(SR)	(100/100)	(87/100)	(79/100)
Ic-1(2)(E_T)	(6.1)	(2484)	(9.4)
Ic-2(SR)	99/100	62/100	44/100
Ic-2(E_T)	0.6	1996.8	21.34

Results for INFO-RNA, RNA-SSD and RNAinverse concerning performance and speed on the artificial structures of test set Ic. The succession rate (SR) gives the fraction of structures for which the respective algorithm found a correct solution. E_T represents the expected time needed to compute a solution (given in CPU seconds). The values in parentheses in lines Ic-1(2) give the succession rates and the expected times to compute a solution within a distance of two from Ic-1.

Table 5. Results for test set IIb.

Subset of IIb	INFO-RNA	RNA-SSD	RNAinv.
220–400 (ASR)	100%	93%	2.0%
220–400 ($\bar{E}_T$)	2.4	226.8	—
400–900 (ASR)	100%	93%	0.3%
400–900 ($\bar{E}_T$)	93.3	285.3	—
900–1975(ASR)	100%	81%	0.0%
900–1975($\bar{E}_T$)	1447.4	3043.9	—

Results for subsets (depending on the structure size) of test set IIb obtained by running INFO-RNA and RNA-SSD 25 times each and RNAinverse ten times for each structure. The time $\bar{E}_T$ is the average expected time needed to compute a solution for a structure of the respective subset. The average succession rate (ASR) gives the average fraction of successful runs.

The first test set of computationally predicted structures consists of 24 structures of 260–1475 bases also analyzed by Andronescu *et al*. (2004). They created a set of ribosomal RNA sequences obtained from the Ribosomal Database Project (Cole *et al*., 2003) and predicted their mfe structures. We refer to this as test set IIa. Since Andronescu *et al*. have already shown that RNA-SSD performs better than RNAinverse when analyzing structures of IIa, we restricted our tests to a comparison of INFO-RNA and RNA-SSD. For each structure, we performed between 10 and 50 runs per algorithm similar to Andronescu *et al*. (2004). As both algorithms are successful here, we turned our attention to the comparison of speed. For that purpose, we applied INFO-RNA in a slightly different way. If it did not succeed within 300 CPU seconds, INFO-RNA was aborted and, thus, terminated unsuccessfully. Afterwards, the neighbor-testing-mode is changed for the next runs (from the energy-dependent NE-mode to the arbitrary NA-mode or back), as it seems that the current strategy is not successful for the given structure. The new mode is retained till

Table 6. Results for test set III.

	Name	Size	INFO-RNA SR	E_T	RNA-SSD SR	E_T
1	Minimal catalytic domains of the hairpin ribozyme satellite RNA of the tobacco ringspot virus (Figure 1a) (Fedor, 2000)	65	100/100	0.03	100/100	0.04
2	U3 snoRNA 5′-domain from *Chlamydomonas reinhardtii*, *in vivo* probing (Figure 6B) (Antal *et al*., 2000)	79	100/100	0.01	100/100	0.02
3	*H. marismortui* 5S rRNA (Figure 2) (Szymanski *et al*., 2002)	122	(100/100)(2)	(45.2)	(100/100)(2)	(2163.9)
4	VS Ribozyme from *Neurospora* mitochondria (Figure 1A) (Lafontaine *et al*., 2001)	167	100/100	0.1	100/100	0.3
5	R180 ribozyme (Figure 2B) (Sun *et al*., 2002)	180	37/100 (63/100)(2)	194.0	58/100 (20/100)(2)	2267.8
6	XS1 ribozyme, *Bacillus subtilis* P RNA-based ribozyme (Figure 2A) (Mobley and Pan, 1999)*	314	100/100	19.0	100/100	22.4
7	Homo Sapiens RNase P RNA (Figure 4) (Pitulle *et al*., 1998) *	340	100/100	66.8	94/100	491.1
8	S20 mRNA from *E.coli* (Figure 2) (Mackie, 1992)	372	100/100	110.8	87/100	728.2
9	*Halobacterium cutirubrum* RNAse P RNA (Figure 2) (Haas *et al*., 1990)*	376	(100/100)(4)	(5026.8)	(1/100)(6)	(220530.0)
10	Group II intron ribozyme D135 from ai5γ (Figure 5) (Swisher *et al*., 2001)	583	100/100	7.9	100/100	3.9

Originally pseudoknotted structures are marked with an asterisk (*). Here, pseudoknots are removed by disregarding 8 bp in each case. All other are pseudoknot-free. The succession rate (SR) gives the fraction of runs in which the respective algorithm found a correct solution. E_T represents the expected time needed to compute a solution. For structures were no correct solution was found, SR and E_T are given in parentheses. They reflect the fraction in which the best approximate solution was found and the time needed for it, respectively. The distance to the target structure is given additionally.

it fails. Thus, we always applied the mode with less unsuccessful terminations. If both modes led to the same number of failures, NE-mode was chosen. This strategy of testing is obvious, since users of the program will change the parameters as well, if the algorithm has failed with their current parameter values. Since RNA-SSD does not include these modes, the strategy was only applied in case of INFO-RNA. Generally, it can be said that INFO-RNA performs much faster than RNA-SSD did in (Andronescu *et al*., 2004). But we repeated all tests with a newer version of RNA-SSD on our PCs. For test set IIa, the results are comparable for INFO-RNA and RNA-SSD. However, INFO-RNA failed for only one structure, while RNA-SSD did for three. Detailed results can in be seen in Supplementary Data.

The second test set of computationally predicted structures consists of 308 structures of 220–1975 bases. They are the mfe structures of all annotated eukaryotic rRNA gene sequences from release 9.27 of the Ribosomal Database Project (RDP-II) (Cole *et al*., 2005). We refer to this as test set IIb. The whole set of eukaryotic sequences from RDP-II was chosen since the performance of INFO-RNA is to be tested on more than some exemplary sequences chosen in (Andronescu *et al*., 2004). For INFO-RNA and RNA-SSD, we performed 25 runs for each structure, and because of the longer runtime only 10 times for RNAinverse. Furthermore, runs of RNAinverse were terminated unsuccessfully if no solution was found after 3600 CPU seconds. To determine the success of the algorithms for classes of structures in a certain size range, we divided test set IIb into three subsets according to the size of the structures. The results are shown in Table 5. INFO-RNA performs best and fastest for all three subsets of IIb and all runs for each structure were successful.

Structures from the biological literature. Finally, we analyzed the performance of INFO-RNA and RNA-SSD on a test set containing structures published in the literature. This set is chosen identical to test set C of Andronescu *et al*. (2004). We refer to it as test set III. Pseudoknots were removed by disregarding pairs in pseudoknots. Results are given in Table 6. We did not examine the performance of RNAinverse since Andronescu *et al*. have already done this. To analyze success and speed of INFO-RNA and RNA-SSD, we examined 100 runs per structure for each algorithm. The succession rates and expected computing times demonstrate the excellent performance of INFO-RNA. In all but one case, it was faster than RNA-SSD. Furthermore, it succeeded for more structures than RNA-SSD and unsuccessful runs terminated with better approximate solutions.

3.3 Stability of the designed sequences

Another important item for the validation of INFO-RNA is the question of the stability of the designed sequences. We analyzed the stability of some arbitrarily chosen structures of test sets IIa+b. The selected biological sequences underlying test set IIa were chosen according to Andronescu *et al*. to assure comparability. For each, we compared the stability of its mfe fold to that of the designed sequences when folding into the predicted structure. For that purpose, we used the partion function option of RNAfold of the Vienna RNA Package (Hofacker *et al*., 1994). For each designed sequence S as well as for the biological sequences S^b, we computed the probability $P(T|S)$ of the final sequence folding into the target structure T. The designed sequences were sorted according to their stability. The best, the median and the worst ones as well as the results for the biological sequences are given in Table 7. Sequences designed by INFO-RNA are much more stable than the biological ones and the sequences obtained by Andronescu *et al*. (2004).

In a second step, we analyzed arbitrarily chosen sequences underlying test set IIb (a small, a medium and a long one) and evaluated the stability of their mfe folds to that of the designed sequences when folding into the predicted structure. Results are also given in Table 7. Again, sequences designed by INFO-RNA have a much

Table 7. The stability of test set IIa and IIb.

No. of structure	IIa			IIb			IIb (all)	
	1	12	17	113	258	130	$>P(T \mid S^b)$	$>10^3 \times P(T \mid S_b)$
Size	260	506	856	277	716	1225		
Best $P(T \mid S)$	0.16865	0.01470	1.46×10^{-5}	0.13196	0.00112	2.46×10^{-10}	100%	99%
Median $P(T \mid S)$	0.00901	0.00052	4.65×10^{-8}	0.02637	2.33×10^{-6}	2.83×10^{-14}	100%	78.2%
Worst $P(T \mid S)$	4.99×10^{-6}	9.17×10^{-7}	2.71×10^{-13}	0.00059	6.75×10^{-10}	3.09×10^{-18}	76.3%	34.7%
Biol. $P(T \mid S^b)$	0.00023	4.03×10^{-8}	1.00×10^{-16}	3.60×10^{-7}	1.67×10^{-14}	1.86×10^{-23}		

Stability of some exemplary results of test sets IIa (chosen according to (Andronescu *et al*. (2004) and IIb (chosen arbitrarily). Best $P(T \mid S)$ gives the highest probability reached by one of our designed sequences for structure T. Median $P(T \mid S)$ and worst $P(T \mid S)$ are defined analogously. The overall stability results for test set IIb are given in the right part.

higher stability than the biological ones. Furthermore, Table 7 shows the excellent quality of all results of test set IIb. In all cases, the best and even the median designed sequences have a higher stability than the biological ones. For most structures, the best designed sequence was >1000 times more stable than the biological one.

All these results show that in INFO-RNA, it is not necessary to optimize the stability additionally. It suffices to minimize the structure distance of the mfe structures to the target one to design highly stable structures.

4 DISCUSSION

We have introduced a very fast and successful new approach of the inverse RNA folding problem, called INFO-RNA. In general, it outperforms existing tools. It consists of two major steps: a new initialization method and a following advanced stochastic local search that uses an effective neighbor selection method. The former is implemented by a dynamic programming approach, which finds a sequence that among all sequences adopts the target structure with the lowest possible energy. It is done in linear time depending on the structure size. We have shown that this initial sequence is an excellent starting point for the subsequent local search, which is short but powerful. Only few local search steps and less time are needed to generate a good sequence that folds into the target structure. This is due to an energy-based pre-ordering of the set of neighbors which can be calculated much faster than the actual optimization criterion of minimizing the structure distance. Since all computational approaches for (inverse) RNA folding are based on a thermodynamic model, the sequences designed by INFO-RNA are not guaranteed to fold into the target structure in a cell.

To test the performance of INFO-RNA, we analyzed several test sets of artifically generated as well as biological RNA structures and compared the results with RNA-SSD and RNAinverse. In general, INFO-RNA outperforms RNA-SSD and it performs substantially better than RNAinverse. However, it should be noted that RNAinverse was also designed to produce random samples from the sequence space. Obviously, INFO-RNA cannot be used to produce samples since the initializing sequence is rather fixed apart from a random variation in unbound bases located in loop regions. We performed some initial experiments to investigate the performance of INFO-RNA using a random initializing sequence. The improved stochastic local search alone is still able to produce comparable results, although INFO-RNA looses much of its speed. Thus, for the sampling task INFO-RNA should be used without the initialization method.

Since G–C base pairs are energetically most favorable, the initializiation sequences of INFO-RNA have a high GC content. This GC content is subsequently reduced by the local search but the final sequences are still enriched in G's and C's which might explain the high stability of the designed sequences. In future, it is desirable to introduce sequence constraints in INFO-RNA to reduce the GC content.

ACKNOWLEDGEMENTS

The authors would like to thank Michael Hiller and Sebastian Will for reading the manuscript and for their helpful comments. Funding to pay the Open Access publication charges was provided by the Albert-Ludwigs-University Freiburg.

Conflict of Interest: none declared.

REFERENCES

Addess,K.J. *et al*. (1997) Structure and dynamics of the iron responsive element RNA: implications for binding of the RNA by iron regulatory binding proteins. *J. Mol. Biol*., **274**, 72–83.

Andronescu,M. *et al*. (2004) A new algorithm for RNA secondary structure design. *J. Mol. Biol*., **336**, 607–624.

Antal,M. *et al*. (2000) molecular characterization at the RNA and gene levels of U3 snoRNA from a unicellular green alga, *Chlamydomonas reinhardtii*. *Nucleic Acids Res*., **28**, 2959–2968.

Cech,T.R. (2004) RNA finds a simpler way. *Nature*, **428**, 263–264.

Cole,J.R. *et al*. (2005) The Ribosomal Database Project (RDP-II): sequences and tools for high-throughput rRNA analysis. *Nucleic Acids Res*., **33**, D294–D296.

Cole,J.R. *et al*. (2003) The Ribosomal Database Project (RDP-II): previewing a new autoaligner that allows regular updates and the new prokaryotic taxonomy. *Nucleic Acids Res*., **31**, 442–443.

Dirks,R.M. *et al*. (2004) Paradigms for computational nucleic acid design. *Nucleic Acids Res*., **32**, 1392–403.

Fedor,M.J. (2000) Structure and function of the hairpin ribozyme. *J. Mol. Biol*., **297**, 269–291.

Haas,E.S. *et al*. (1996) Comparative analysis of ribonuclease P RNA structure in Archaea. *Nucleic Acids Res*., **24**, 1252–259.

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

Hofacker,I.L. (1994) The rules of the evolutionary game for RNA: a statistical characterization of the sequence to structure mapping in RNA.

Hoos,H.H. (1998) Stochastic Local Search—Methods,Models,Applications. PhD thesis, Darmstadt University of Technology, Darmstadt, Germany.

Hüttenhofer,A. and Böck,A. (1998) RNA structures involved in selenoprotein synthesis. In Simons,R. W. and Grunberg-Manago,M. (eds), *RNA Structure and Function*. Cold Spring Habor Laboratory Press, Cold Spring Habor, pp. 603–639.

Hüttenhofer,A. *et al*. (2002) RNomics: identification and function of small, non-messenger RNAs. *Curr. Opin. Chem. Biol*., **6**, 835–843.

Knight,J. (2003) Gene regulation: switched on to RNA. *Nature*, **425**, 232–233.

Lafontaine,D.A. *et al*. (2001) Structure, folding and activity of the VS ribozyme: importance of the 2-3-6 helical junction. *EMBO J*., **20**, 1415–1424.

Liu,Z. *et al*. (1998) The nature of the minimal 'selenocysteine insertion sequence' (SECIS) in *Escherichia coli*. *Nucleic Acids Res*., **26**, 896–902.

Mackie,G.A. (1992) Secondary structure of the mRNA for ribosomal protein S20. Implications for cleavage by ribonuclease E. *J. Biol. Chem*., **267**, 1054–1061.

Mobley,E.M. and Pan,T. (1999) Design and isolation of ribozyme-substrate pairs using RNase P-based ribozymes containing altered substrate binding sites. *Nucleic Acids Res*., **27**, 4298–4304.

Pitulle,C. *et al*. (1998) Comparative structure analysis of vertebrate ribonuclease P RNA. *Nucleic Acids Res*., **26**, 3333–3339.

Storz,G. (2002) An expanding universe of noncoding RNAs. *Science*, **296**, 1260–1263.

Sun,L. *et al*. (2002) A selected ribozyme catalyzing diverse dipeptide synthesis. *Chem. Biol*., **9**, 619–628.

Swisher,J. *et al*. (2001) Visualizing the solventinaccessible core of a group II intron ribozyme. *EMBO J*., **20**, 2051–2061.

Szymanski,M. *et al*. (2002) 5s Ribosomal RNA Database. *Nucleic Acids Res*., **30**, 176–178.

Winkler,W.C. *et al*. (2004) Control of gene expression by a natural metabolite-responsive ribozyme. *Nature*, **428**, 281–286.

Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res*., **9**, 133–148.

Zuker,M. (1994) Prediction of RNA secondary structure by energy minimization. *Methods Mol. Biol*., **25**, 267–294.