

# Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction

Jens Reeder and Robert Giegerich\*

Faculty of Technology, Bielefeld University, 33615 Bielefeld, Germany

Received on May 9, 2005; revised on June 16, 2005; accepted on July 6, 2005

Advance Access publication July 14, 2005

## ABSTRACT

**Motivation:** The well-known Sankoff algorithm for simultaneous RNA sequence alignment and folding is currently considered an ideal, but computationally over-expensive method. Available tools implement this algorithm under various pragmatic restrictions. They are still expensive to use, and it is difficult to judge if the moderate quality of results is because of the underlying model or to its imperfect implementation.

**Results:** We propose to redefine the consensus structure prediction problem in a way that does not imply a multiple sequence alignment step. For a family of RNA sequences, our method explicitly and independently enumerates the near-optimal abstract shape space, and predicts as the consensus an abstract shape common to all sequences. For each sequence, it delivers the thermodynamically best structure which has this common shape. Since the shape space is much smaller than the structure space, and identification of common shapes can be done in linear time (in the number of shapes considered), the method is essentially linear in the number of sequences. Our evaluation shows that the new method compares favorably with available alternatives.

**Availability:** The new method has been implemented in the program RNacast and is available on the Bielefeld Bioinformatics Server.

**Contact:** {jreeder, robert}@TechFak.Uni-Bielefeld.DE

**Supplementary information:** Available at <http://bibiserv.techfak.uni-bielefeld.de/rnacast/supplementary.html>

## INTRODUCTION

The role of RNA in all organisms is much broader and more fundamental than it was considered only recently (Lee *et al.*, 1993; Mattick and Gagen, 2001; Lee and Ambros, 2001; Skryabin *et al.*, 2003; Pfeffer *et al.*, 2004). With non-coding RNA, the structure of the molecule is often essential for its function. In analogy to coding RNA, where a conserved encoded protein hints at a similar metabolic function, structural conservation in RNA may give clues to RNA function and to finding of RNA genes. However, structure conservation is more complex to deal with computationally than sequence conservation.

## Comparative structure prediction and the Sankoff algorithm

The secondary structure of RNA—the level of base pairing—strongly determines the tertiary structure. Since the latter is computationally

intractable and experimentally expensive to obtain, secondary structure analysis has become an accepted substitute. Early computational approaches to secondary structure prediction were by Nussinov *et al.* (1978), and Waterman and Smith (1978). Today's methods use minimum free energy folding, pioneered by Zuker and Stiegler (1981). Such methods are widely used today, although it is known that their results are not completely reliable (Doshi *et al.*, 2004). Better results are generally achieved by comparative analysis of a family of homologous sequences, where sequence and structure conservation is exploited, using a resolved tertiary structure whenever available, sequence alignment, statistical methods and human expertise (Gutell *et al.*, 1992).

A first comparative approach based on thermodynamics was formulated by Sankoff as early as 1985 (Sankoff, 1985), which performs sequence alignment and minimal free energy folding simultaneously. Its time complexity is  $\mathcal{O}(n^6)$ , with space  $\mathcal{O}(n^4)$ , for two sequences of length  $n$ , and for more sequences, it becomes exponential in the number of sequences. Given these high computational costs, it seemed unlikely that this algorithm would ever be put into practice. For many years, it rested in oblivion.

Recently, however, interest in comparative methods for RNA structure prediction has been nurtured by findings on the functional versatility of RNA, and several related approaches have been suggested. Some emphasize the sequence conservation aspect, folding a predetermined sequence alignment under thermodynamic rules (RNAalifold, Hofacker *et al.*, 2002). The other extreme emphasizes thermodynamics and suggests to use multiple structure alignments of independently folded sequences (Höschmann *et al.*, 2004). Other approaches directly implement Sankoff's idea of simultaneous alignment and folding, but introduce various pragmatic restrictions, e.g. Dynalign (Mathews and Turner, 2002) and Foldalign (Gorodkin *et al.*, 1997). For a recent review of these and other tools, the reader is referred to the study of Gardner and Giegerich (2004).

Behind all these approaches, there is the original Sankoff approach as the ideal method—the one that every program tries to approximate in different ways. 'Making Sankoff practical' has been a recurring theme at the meetings of the computational RNA community. However, this road may require so many pragmatic restrictions that the ideal loses much of its attraction.

A way out of this dilemma may be to change the definition of a consensus structure. In Sankoff's approach, the consensus is a folded sequence alignment that optimizes a combined sequence similarity and energy score. What if we drop the implicit multiple sequence alignment step (as this problem is known to be NP-complete)? Let

\*To whom correspondence should be addressed.

us agree that a consensus structure for sequences  $s_1, \dots, s_k$  is a set of structures  $x_1, \dots, x_k$ , one for each  $s_i$ , that all have, in some mathematically precise sense, a common shape. Should a sequence alignment of  $s_1, \dots, s_k$ , compatible with the consensus, also be desired, it may be computed afterward from  $x_1, \dots, x_k$ , rather than from  $s_1, \dots, s_k$ , by multiple structure alignment (Höchsmann *et al.*, 2004). The latter phase will certainly need to resort to heuristics, but for the first phase, there may be a chance to achieve a complete and non-heuristic solution in acceptable time.

### An alternative to the Sankoff method

**A hypothetical method** To explain our new approach, let us first consider a hypothetical, exhaustive method. Let  $s_1$  and  $s_2$  be two RNA sequences, both of length  $n$ . Let us enumerate their foldings in the order of increasing free energy, yielding  $x_1, x_2, \dots, x_{N_1}$  for  $s_1$  and  $y_1, y_2, \dots, y_{N_2}$  for  $s_2$ . The numbers  $N_1$  and  $N_2$  will be very large, even for small  $n$ , but let us ignore this for the moment.

If  $s_1$  and  $s_2$  have a common structure, there must be  $x_i = y_j$  for some  $i$  and  $j$ . In fact, there may be more such pairs. We rank them by  $(i + j)$ , and the pair  $(x_i, y_j)$  with minimal rank is our predicted consensus. Just as well, we may produce the  $k$  top-ranking consensus pairs.

Using known algorithmic techniques, we can implement the enumeration in  $\mathcal{O}[n^3 + n(N_1 + N_2)]$  time and  $\mathcal{O}(n^2)$  space, and the identification of common structures in  $\mathcal{O}[n(N_1 + N_2)]$  time and space, where we represent structures as strings and employ keyword or suffix trees for fast identity matching. Clearly, if we add a third sequence  $s_3$ , with structures  $z_1, z_2, \dots, z_{N_3}$ , the  $(N_1 + N_2)$  above is replaced by  $(N_1 + N_2 + N_3)$ , and hence, this method is additive in the number of sequences! However, it is not practical for the following two reasons:

- The numbers  $N_1, N_2, \dots$  are very large and  $N_i$  grows exponentially with  $n$ . Even if we restrict enumeration to an energy range of say 10% above the minimal free energy,  $N_i$  may be large as 100 000 or 1 000 000. This alone might not be a threat on today's computers, but there is one more problem.
- Sequences  $s_1$  and  $s_2$  need not have the same length, and hence their structures cannot be identical. We must allow for some flexibility in the relative position of helices. Therefore, we need to resort to some pairwise similarity computation, catapulting computation time of the identification phase to  $\mathcal{O}(n^2 \cdot N_1 \cdot N_2)$  or higher. The additive behavior is lost.

To make our hypothetical method practical, we need to restrict enumeration to a small, but representative sample of the folding space, and achieve identification of consensus pairs in linear time in spite of their not being identical.

**Outline of the consensus shapes prediction method** We build on the recent approach of abstract RNA shape analysis (Giegerich *et al.*, 2004) to solve both of the above problems. Deferring formal definitions, a shape is a family of structures, sharing a common pattern of helix nesting and adjacency. The near-optimal folding space contains only a (relatively) small number of shapes. Using abstract shape analysis, we enumerate representative structures—one per shape, and only those!—for both  $s_1$  and  $s_2$ . The highest ranking structure pair  $x_i$  and  $y_j$ , where both have the same shape, then forms our consensus pair. Although the structures  $x_i$  and  $y_j$  are only similar, their shapes can be easily computed, and identity matching on shapes can

be implemented in time  $\mathcal{O}[n \cdot (N_1 + N_2)]$  as sketched above, for significantly reduced  $N_1$  and  $N_2$ .

These ideas will be rigorously described below, and we shall report on their implementation and evaluation.

## RNA SHAPE ANALYSIS AND CONSENSUS SHAPES

### Abstract shapes

We recall the basic definitions of abstract shape analysis.

- An RNA sequence  $s$  has folding space  $\mathcal{F}(s)$ , the set of all admissible structures under the given base-pairing rules. For each structure  $x \in \mathcal{F}(s)$ , we can compute its free energy  $E(x)$ .
- The minimal free energy structure  $\text{mfe}(s)$  for a sequence  $s$  is the structure  $x \in \mathcal{F}(s)$  where  $E(x)$  is minimal.
- For efficient computation of shapes via dynamic programming, they must be represented as trees. Let  $\mathcal{S}$  be the tree-like domain of structures, and  $\mathcal{P}$  a tree-like domain of shapes. A shape abstraction is a mapping  $\pi$  from  $\mathcal{S}$  to  $\mathcal{P}$  that preserves juxtaposition and embedding.
- The abstract shape space of sequence  $s$  is  $\mathcal{P}(s) = \{\pi(x) | x \in \mathcal{F}(s)\}$ . The class of  $p$ -shaped structures in  $\mathcal{F}(s)$  is  $\mathcal{F}(s|p) = \{x | x \in \mathcal{F}(s), \pi(x) = p\}$ .
- The shape representative structure  $\hat{p} \in \mathcal{F}(s)$  for shape  $p$  is the structure whose free energy is minimal among all members of that shape class. We call it shrep for short.

Abstract shape analysis, as implemented by the program RNASHapes, is computed for an RNA sequence  $s$  and an energy range  $R$ . It delivers a list  $[(p_1, \hat{p}_1), \dots, (p_k, \hat{p}_k)]$  with the following properties:

- the  $p_i$  are different shapes, and the  $\hat{p}_i$  are their respective shreps,
- the list is ordered by increasing energy:  $\text{mfe}(s) = \hat{p}_1$  and  $E(\hat{p}_i) \leq E(\hat{p}_{i+1})$ ,
- the list is restricted to the energy range indicated by  $R$ :  $E(\hat{p}_i) \leq E(\text{mfe}(s)) + R$ , and
- the list completely covers this energy range in the abstract shape space: there is no shape  $p_{k+1}$  such that  $E(\hat{p}_{k+1}) \leq E(\hat{p}_1) + R$ .

The strength of shape analysis lies in four aspects (for details see Giegerich *et al.*, 2004):

- It produces a non-heuristic, mathematically well defined synoptic view of the near-optimal folding space, allowing us to concentrate on a small number of shreps.
- Shape analysis uses the full energy model (Mathews *et al.*, 1999) and runs in the same asymptotic space and time complexity as suboptimal RNA folding.
- Shapes are meaningful across sequences, hence lend themselves to a comparative approach. This aspect is exploited here for the first time.
- The approach is generic with respect to the shape abstraction ( $\pi$ ) that is actually used. Shapes can be more or less abstract, depending on the level of detail considered relevant.

We illustrate the latter point by defining two shape abstractions used in this study. In general, shape abstractions retain nesting and

**Table 1.** Definition of Level 5 and Level 3 shape abstractions

$\pi_5(.) = \varepsilon$	$\varrho_5(.) = \varepsilon$
$\pi_5(s) = \pi_5(s)$	$\varrho_5(s) = \varrho_5(s)$
$\pi_5(s) = \pi_5(s)$	$\varrho_5(s) = \varrho_5(s)$
$\pi_5((s)) = [\varrho_5(s)]$	$\varrho_5((s)) = \varrho_5(s)$
$\pi_5((s)s') = [\varrho_5(s)]\pi_5(s')$	$\varrho_5((s)s') = \pi_5((s)s')$
$\pi_3(.) = \varepsilon$	$\varrho_3((s)) = \varrho_3(s)$
$\pi_3(s) = \pi_3(s)$	$\varrho_3(s) = \pi_3(s)$ In all other cases
$\pi_3(s) = \pi_3(s)$	
$\pi_3((s)) = [\varrho_3(s)]$	
$\pi_3((s)s') = [\varrho_3(s)]\pi_3(s')$	

$s$  and  $s'$  denote a non-empty, well-balanced dot-bracket string, and  $\varepsilon$  denotes the empty string. Brackets in the input/output string are written in bold face. Note that difference lies with  $\varrho_5$  versus  $\varrho_3$ , where the former reads across bulges and internal loops, whereas the latter decides to record a new helix part with every interruption.

adjacency of helices, but disregard their size and concrete position in the primary sequence. They may choose to retain or to discard bulges and internal loops, which leads to different levels of abstraction. ‘Level 5’ is the strongest abstraction and does not account for bulges etc. at all. ‘Level 3’ retains helix interruptions, but does not specify whether they result from 5'-bulges, 3'-bulges or internal loops.

As we are not concerned with the algorithmics of shape analysis here, we can forget about tree-like representations of structures and shapes, and define shape abstractions as mappings from the more familiar string representations of structures to string representations of shapes. Structures are represented as dot-bracket strings, e.g. ‘(((...((...((...))))).((...))...))’). The Level 5 shape of this structure is represented as ‘[[[[]]]’], its Level 3 shape as ‘[[[[]]][]]’). In Table 1, we provide equations defining shape abstractions  $\pi_5$  and  $\pi_3$ .

The rule about the choice of abstraction level is that we generally prefer to work with the less abstract level 3, except for long molecules where a stronger abstraction speeds up the program because the shape space is reduced further.

Incidentally, Level 3 shapes correspond to the two-interval patterns described in Viallette (2004).

## Rankings of true shapes

In order to evaluate whether shape analysis bears promise toward consensus prediction, we performed two preliminary studies, using several sequence families from Rfam (Griffiths-Jones *et al.*, 2003) and other databases (Supplementary material) where the ‘true’ structure  $s$  is known. From this true structure, we can compute the ‘true’ shape  $p^* = \pi(s)$ . Question 1 asks for the rank  $i$  such that  $p_i = p^*$  in the list of shapes returned by shape analysis. Table 2 shows the outcome. The average rank of the true shape is 5.06, where in 32 of 69 cases (46%) the true shape has rank 1.

The advantage of shape analysis over complete suboptimal folding (Wuchty *et al.*, 1998) is witnessed by two detailed observations: for one of the tRNA sequences, the true shape has rank 3, whereas the true structure has rank 104 in the complete enumeration. In the worst case observed, a U12 RNA sequence, the true shape has rank 28, whereas its associated true structure has rank 3 695 033. This confirms our hope that the shape space is small enough to completely enumerate its interesting part. But it also confirms that the reliability

**Table 2.** Ranks of true shape in the list of near-optimal shapes using RNAsHapes

Rank of true shape	1	2	3	4	5	5–9	10–19	20+	Total
lin4	9	0	0	0	0	0	0	0	9
IRES	5	2	0	0	0	0	0	0	7
tRNA	3	1	5	2	0	0	0	0	11
srp RNA	2	2	0	0	0	0	0	0	4
Riboswitch	7	0	0	0	0	0	0	0	7
S box	4	5	2	0	0	0	0	0	11
5S rRNA	1	2	0	1	0	1	0	0	5
U12 RNA	0	0	0	0	1	0	1	4	6
U1 RNA	1	1	0	1	0	0	1	0	4
U2 RNA	0	0	0	0	0	3	0	2	5

**Table 3.** This histogram shows the rank of the reference shape in all pairwise predictions

Rank of true shape	1	2	3	4	5	≥6	Total
lin4	36	0	0	0	0	0	36
IRES	11	10	0	0	0	0	21
tRNA	22	22	11	0	0	0	55
srp RNA	5	1	0	0	0	0	6
Riboswitch	21	0	0	0	0	0	21
S box	21	31	3	0	0	0	55
5S rRNA	8	1	1	0	0	0	10
U12 RNA	0	3	0	0	0	12	15
U1 RNA	4	0	1	0	1	0	6
U2 RNA	0	0	0	0	0	10	10

(in terms of correctly predicted shapes) of single sequence folding lies ~46%—not useless, but not dependable either.

Question 2 asks whether this improves when we move toward a comparative approach by using pairs of sequences. In Table 3 we consider all pairs of predictions (within each family), and report on the rank of the true shape in the list of all common shapes. In the pairwise approach, the average rank of the true shape improves to 3.13, and the true shape now has rank 1 in 128 of 235 cases (53%). We conclude that the power of comparative analysis is well captured by our approach, and expect even better performance by using three or more sequences.

## Consensus shape prediction

We now summarize the proposed method of consensus shape prediction.

For a set of sequences  $\{s_1, \dots, s_k\}$ , intentionally a family of related RNA sequences, we enumerate their shape spaces  $\mathcal{P}(s_1), \dots, \mathcal{P}(s_k)$ . Upon those, we define:

**DEFINITION 1.** A shape  $p$  is a common shape of  $\{s_1, \dots, s_k\}$  if  $p \in \bigcap_{i=1}^k \mathcal{P}(s_i)$ .

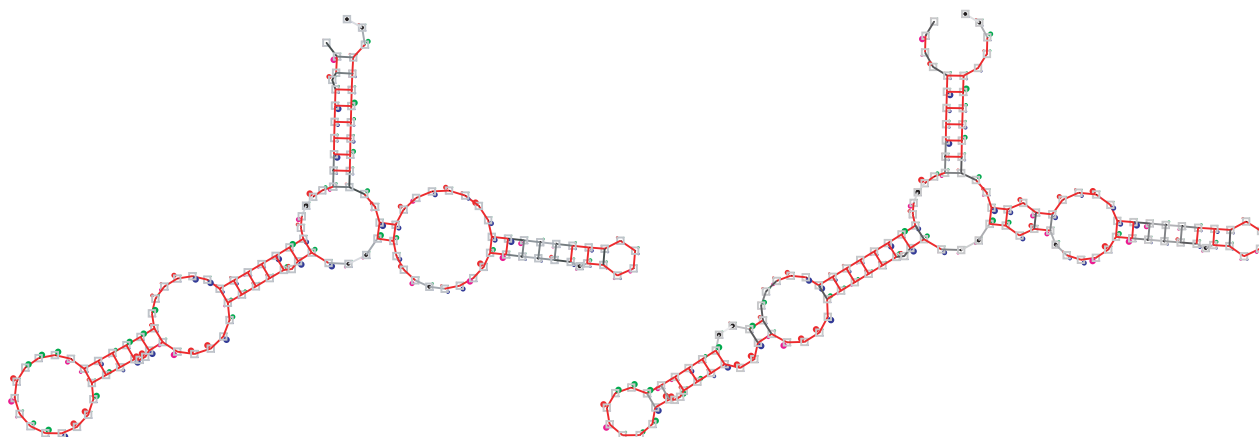
**DEFINITION 2.** The consensus shape for sequences  $\{s_1, \dots, s_k\}$  is the common shape  $p$  that minimizes  $\text{rank}(\hat{p}_1, \dots, \hat{p}_k)$ .

```

Shape: [[[]]] Score: -223.50
CCUUUGCAGGCAGCGGAAUCCCCACCUGGUAACAGGUGCCUCUGCGGCCAAAAGCCACGUGUAUAAGAUACACCUGCAAAGG
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
(-34.10) R = 2
CCUUUGCAGGCAGCGGAAUCCCCACCUGGUGACAGGUGCCUCUGCGGCCGAAAGCCACGUGUGUAAGACACACCUGCAAAGG
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
(-39.10) R = 2
GCACGCAAGCCGCGGGAACUCCCCUUGGUAACAAGGACCCGCGGGGCCGAAAGCCAGUUCUCUGAACCUUGCGUGU
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
(-34.10) R = 2
GCAUGAUGGUGUGGGAACUCCCCUUGGUAACAAGGACCCACGGGGCCAAAAGCCAGUCCUACGACCCAUCAUGC
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
(-34.70) R = 3
GCAUGACGCGCGUGGGAACUCCUCCUUGGUAACAAGGACCCACGGGGCCAAAAGCCACGCCACACGGGCCCGUCAUGU
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
(-41.90) R = 1
GCAUGUUGGCCGUGGGAACUCCUUGGUAACAAGGACCCACGGGGCCGAAAGCCAUUCCUACGACCCCAACAUGU
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
(-39.60) R = 1

```

**Fig. 1.** Example output for a family of IRES elements of Picornaviridae viruses. It first shows the common shape and the achieved score. Thereafter, for each input RNA and not aligned, there is the sequence, the predicted shrep, its energy and its individual rank in the shape space.



**Fig. 2.** Two multiple RNA structure alignments of five 5S rRNAs computed by RNAforester. On the left-hand side is the alignment of the structures as found in the database (Szymanski *et al.*, 2000). On the right-hand side, the output shreps of RNAcst served as input for the alignment. Obviously, the structures are similar, with RNAcst predicting a few additional compatible base pairs. The alignment visualization should be interpreted as follows: the frequencies of the bases A, C, G and U are proportional to the radius of circles that are arranged for each residue clockwise on the corners of a square, starting at the upper left corner. Additionally, these circles are colored red, green, blue and magenta for the bases A, C, G and U, respectively. The frequency of a gap is proportional to a black circle growing at the center of the square.

Here, rank is a scoring function that combines the individual shrep scores. We will discuss several meaningful scoring functions in the Algorithms section.

Computing the intersection of  $\mathcal{P}(s_1), \dots, \mathcal{P}(s_k)$ , once we have those, is a trivial task. Usually, there are much less common shapes, than there are shapes in  $\mathcal{P}(s_1), \dots, \mathcal{P}(s_k)$ . We then sort all common shapes by their rank and get the consensus shape as the first one in the list.

Note, that the above definitions not only yield the consensus shape, but from shape analysis we also get the set of shreps—the resulting output is a  $(k + 1)$ -tuple  $(p, [\hat{p}_1, \dots, \hat{p}_k])$ . These shreps constitute an (unaligned) multiple RNA structure prediction for the input sequences.

## ALGORITHM IMPLEMENTATION

### The program RNAcst

The above method has been implemented by the program RNAcst, which stems from ‘RNA consensus abstract shapes technique’. Although most of the method is clear from our definition of the consensus shape, a few details remain to be fixed.

**Step 1.** Our algorithm starts with sequences  $s_1, \dots, s_k$  as input, and an energy threshold  $R$ . Let  $n$  be their average length. We run RNASHapes on each individual sequence with the provided energy range  $R$ . Theoretically every sequence could have its own  $R$ , but in practice we use only one.

**Step 2.** Within the  $k$  resulting lists (the shape spaces), we identify all shapes that occur in all the lists. We use hashing techniques for fast identity matching of shapes. Thus this phase runs in a time proportional to  $k \cdot |\mathcal{P}(s_1)|$ . After this step, we have a list of all  $l$  common shapes, together with their shreps:  $[(p_1, [\hat{p}_1^1, \dots, \hat{p}_k^1]), \dots, (p_l, [\hat{p}_1^l, \dots, \hat{p}_k^l])]$ .

**Step 3.** We evaluate each common shape with the scoring function and obtain a sorted list of all common shapes. The first shape of this list is returned as the consensus shape, along with its shreps. If desired, the  $r \leq l$  best common shapes can be reported as well.

We propose to use the output of RNAcst as input for RNAforester (Höchsmann *et al.*, 2004), a multiple RNA structure alignment program. The unaligned RNAcst output is shown in Figure 1. The resulting alignment is shown in Figure 2.

Left to be defined is the scoring function rank. We propose three different possibilities:

- (1) *Rank sum score*: Each shrep contributes with its individual rank in the sorted shape space of its sequence:  $\text{rank}_1(p_i, \hat{p}_1^i, \dots, \hat{p}_k^i) = \text{rank}(\hat{p}_1^i) + \dots + \text{rank}(\hat{p}_k^i)$ .
- (2) *Sum of energies*:  $\text{rank}_2(p_i, \hat{p}_1^i, \dots, \hat{p}_k^i) = E(\hat{p}_1^i) + \dots + E(\hat{p}_k^i)$ .
- (3) *Sum of probabilities*:  $\text{rank}_3(p_i, \hat{p}_1^i, \dots, \hat{p}_k^i) = \text{Prob}(\hat{p}_1^i) + \dots + \text{Prob}(\hat{p}_k^i)$ , where  $\text{Prob}(\dots)$  are the probabilities coming from the partition function (McCaskill, 1990), requiring extra  $O(k \cdot n^3)$  steps to compute. In this case, we have to maximize overall scores.

Overall, it turned out that  $\text{rank}_2$ , the simple sum of energies, performs best, followed by  $\text{rank}_3$  and at last the rank sum score. But, prediction accuracy for all three scoring function does not differ much. The method seems to be relatively robust, concerning the choice of scoring function. We decided to use  $\text{rank}_2$  for all computations discussed in this paper.

### Technical limitations

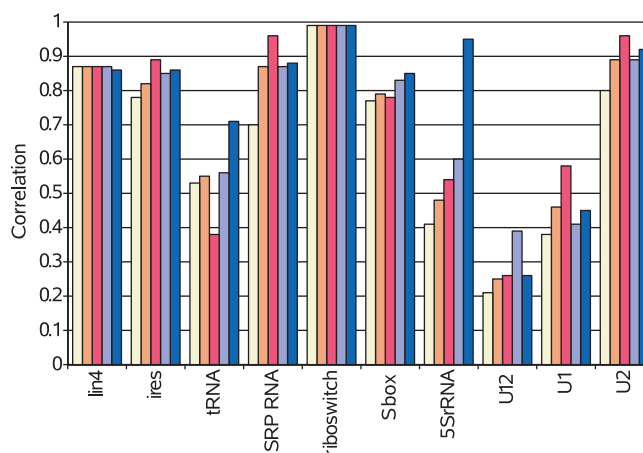
Mathematically, at least one common shape always exists for any set of sequences. This does not hold when we limit the search space, and hence the choice of the energy range  $R$  is critical. If the true shape is missing in the enumerated shape space of one of our sequences (because the energy of its shrep is too high), this shape cannot become a common shape, nor the consensus. The predicted consensus must be wrong in this case. This has consequences: theoretically, there is no upper bound for the number  $k$  of sequences to be considered, and from the efficiency point of view,  $k$  can be quite large. But with larger datasets, chances grow that one of the sequences is an outlier where the chosen energy range just is not large enough, and the true consensus shape is missed owing to the effect explained above. Increasing  $k$  implies a tendency to increase  $R$ , unless we know in advance that the dataset is very homogeneous. When more sequences require to look deeper into the shape space, these two sources of increased efforts multiply. In practice, we suggest to use not more than 10 sequences.

Another point of advice can be given as a rule of thumb. All predicted consensus shreps have the same number of helices. If their native structures are suspected to have different numbers of helices (as with a mixture of four and five stem tRNA) the energy threshold should be large enough to accommodate loss of the extra stem.

### EVALUATION

In our preliminary tests, we evaluated that our method is capable to identify the correct shape in 53% of all pairwise predictions. When we are using RNAcast in the multiple way, the correct shape is predicted for 6 of 10 families and for 3 more families the true shape is on rank 2 or 3. But predicting the correct shape alone is not good enough. Within a shape class, there is considerable structural variation possible. Since shapes abstract from concrete helix positions and sizes, it is theoretically possible that the shrep of a correct shape does not share a single base pair with the true structure.

In this section we evaluate the accuracy achieved by RNAcast on the base pair level and compare it with other tools.



**Fig. 3.** Accuracy (MCC) of RNAcast on a set of RNA families, sorted by size. The bars correspond from left to right to RNAfold, RNAcast pairwise, RNAcast multiple (both  $\pi_5$ ), RNAcast pairwise, RNAcast multiple (both  $\pi_3$ ).

**Table 4.** Prediction accuracy for a set of 5S rRNA

Program	Sensitivity	Selectivity	Correlation
RNAfold	43.08	41.2	0.41
Dynalign	84.20	—	—
RNAcast (pairwise) $\pi_3$	59.10	62.40	0.60
RNAcast (multiple) $\pi_3$	91.98	97.82	0.95

Note: The evaluation of Dynalign, taken from Mathews and Turner (2002), allows for slipping helices, which we do not allow in our evaluation.

In particular we will answer the following questions:

- (1) How accurate are the shreps, given the correct shape?
- (2) What is the improvement over single sequence folding algorithms?
- (3) How does RNAcast perform compared with other pairwise and multiple folding algorithms?
- (4) What are the reasons for wrong predictions?

We evaluate the structure predictions in terms of sensitivity, selectivity and the Matthews correlation coefficient (MCC):

$$\text{Sens.} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{Sel.} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}.$$

True positive base pairs (TP), true negatives (TN) and false negatives (FN) are counted as usual. No slipping of helices is allowed. For false positives (FP) we use the counting method proposed by Gardner and Giegerich (2004): Predicted base pairs, that do not occur in the reference structure, but are compatible with it, are not counted for FP. A base pair  $i \cdot j$  is compatible if neither  $i$  nor  $j$  is paired to another base in the reference and there is no other base pair  $k \cdot l$  that violates the nesting convention (i.e.  $k < i < l < j$ ). This assumption is meaningful, since the reference structures used in this study often



**Table 5.** Comparison with the Gardner study

	RNAcast pairwise		RNAcast multiple		Carnac		Dyalign	
	Sensitivity	Correlation	Sensitivity	Correlation	Sensitivity	Correlation	Sensitivity	Correlation
11 tRNA-PHE	45.2	0.49	71.4	0.75	71.4	0.81	54.78	0.54
5 RNase P	61.3	0.58	65.6	0.63	64.9	0.79	31.95	0.32

RNAcast uses shape abstraction  $\pi_5$ . The Dyalign RNase P results may improve for a larger window size. For detailed Carnac and Dyalign parameter sets see Gardner and Giegerich (2004).

**Table 6.** Recurring situations, for which RNAcast predicts the wrong shape

Error classification	
Additional hairpin predicted	3
Multiloop enclosure	42
One hairpin missed	22
Otherwise	41
Total	107

come as a consensus. The members of a specific RNA family share all base pairs in the consensus, but may have additional ones.

### Accuracy of the true shrep

Let us first assume, that we already know the correct shape of the family under evaluation. We are then asking for the corresponding shreps. We either look them up in the RNashapes output, or we generate an RNA folding program restricted to that specific shape and compute the optimal structure directly. We did this for the same set of RNA families as used in the preliminary study and evaluated the accuracy of the shreps.

On average, the sensitivity is 78.2% and selectivity is 78.6%, compared with 65.4 and 65% for the mfe-prediction of the single sequence. This shows, that knowing the correct shape improves secondary structure prediction of single sequences significantly. However, in most realistic cases we do not know the true shape in advance. The best we can do then is to rely on the consensus shape computed by RNAcast. Note that, even when the predicted consensus shape is incorrect, it still may be close to the correct shape, in which case the predicted structures may also come close to the truth.

### Improvement over single sequence prediction

We now evaluate the accuracy of the structures predicted by RNAcast, whether or not the predicted shape is correct.

We folded each RNA family in five different ways:

- (1) Single sequence prediction using RNAfold (Hofacker *et al.*, 1994),
- (2 and 4) RNAcast on all pairwise combinations using shape abstraction  $\pi_5$  and  $\pi_3$ , and
- (3 and 5) RNAcast in a multiple way on all family members, again in each case with  $\pi_5$  and  $\pi_3$ .

In Figure 3 we plot the (average) MCC of each prediction method. We can see, that 'going comparative' pays off. In all cases but one (multiple tRNA folding with  $\pi_5$ ) RNAcast performs better than single

sequence prediction. The clover leaf prediction for tRNA failed—one arm of the clover leaf was missed. However, using the less abstract shape mapping yields the correct shape and a higher accuracy. We can further see, that using multiple sequences increases the reliability of prediction. Overall,  $\pi_3$  gives the highest accuracy, especially for shorter sequences ( $\leq 150$  bases), where additional bulges or internal loops may be more important than in longer sequences. The averaged MCC for RNAcast multiple with shape abstraction  $\pi_3$  is 0.77. This is a definite increase, compared with 0.64 for single sequence prediction.

Next we relate our method to existing comparative tools.

### Comparison with the Sankoff approach

**Comparison with Dyalign** Dyalign was chosen as a state-of-the-art representative of the Sankoff approach. In Mathews and Turner (2002), the sensitivity of Dyalign is measured on a set of 5S rRNA. We found a secondary structure for five sequences of that set in the database (Szymanski *et al.*, 2000) and evaluated RNAcast on them. Single sequence prediction performs relatively bad on this dataset (RNAfold in Table 4). Using RNAcast in a pairwise fashion improves the accuracy clearly, but still is not satisfying. For Dyalign (pairwise only) we compute an average sensitivity of 84.2%. This is only topped by running RNAcast multiple on five sequences simultaneously. The sensitivity is 92% and selectivity is almost perfect at 97.8%.

**Comparison with the Gardner study** Recently, several multiple RNA folding algorithms were evaluated by Gardner and Giegerich (2004). The study included three different approaches, where 'Plan B' referred to tools that approximate the Sankoff approach of simultaneous alignment and folding. We choose the *Saccharomyces cerevisiae* tRNA-PHE (11 sequences, high sequence similarity) and the *Escherichia coli* RNase P (5 sequences, medium similarity) datasets from that study and compare the prediction accuracies. Since Dyalign permits only pairwise folding, Gardner *et al.* folded the reference sequence with each of the other sequences at a time. We did the same with our program. The corresponding results are in column 'pairwise' and 'Dyalign' in Table 5. Carnac (Touzet and Perriquet, 2004) can fold multiple sequences and performed quite well in the study. Naturally, our method yields much better results for a multiple sequence input than for only two sequences (column 'multiple', Table 5). The sensitivity is comparable with Carnac, which in turn is almost perfectly selective, and thus has a better correlation.

### Detailed analysis of mispredictions

Overall, we observed 107 cases where RNAcast was not able to find the true shape. But how bad are the wrong shapes? By visual

**Table 7.** Performance measured on a 2.8 GHz Dual Xeon system with 2GB RAM

	Length	RNAcast ( $\pi_5$ ) Time (s)	Memory (MB)	RNAcast ( $\pi_3$ ) Time (s)	Memory (MB)	Dynalign Time (s)	Memory (MB)
tRNA	72–75	0.6	19	0.8	20	488	20
U2 RNA	188	4.5	29	5	31	7631	92
RNAse P	237–245	22.5	80	58	257	12718	141

For RNAcast  $R$  was set to 10, for Dynalign  $M$  was set to 15.

inspection we could classify a few recurring situations, listed in Table 6.

In three cases, all tRNA, RNAcast predicts an additional hairpin, not mentioned in the database. This hairpin is the variable arm and, therefore, predicted correctly. On other sequences, we found that parts of the reference structure were enclosed by an additional helix, thus forming a multiloop. From our point of view, this situation could not be counted as false, either. It further confirms our choice not to count compatible base pairs as FP. Another point of error was the loss of one hairpin in 22 cases. Instead of the hairpin, we either see a single-stranded region or the region is consumed by the prolongation of a neighboring helix. Nevertheless, the remaining structure is accurate. All remaining cases (41) differ substantially from the reference and have to be counted as wrong predictions without an excuse.

In general, the accuracy for the first three situations is still rather high on the base pair level, in fact higher than single sequence predictions. Usually, sequences for which RNAcast predicts a wrong shape have a low accuracy and are poorly predicted by RNAfold, too.

## Efficiency

As is to be expected from the asymptotic analysis, the efficiency of RNAcast is quite good. It strongly depends on RNASHAPES' efficiency, which has been optimized recently. On a typical sequence pair of tRNAs (72 and 75 nt) running time remains <1 s, whereas, Dynalign takes 488 s (Table 7). Adding more sequences of similar size increases the runtime only linearly: 10 tRNAs are processed within 5 s. Since the calls to RNASHAPES are independent of each other, they can execute in parallel. The RNAse P example (5 sequences of ~240 bases each) with  $\pi_5$  can actually be done in 12 s.

## DISCUSSION

### Differences from the Sankoff notion of consensus

Let us once more relate RNAcast to Dynalign, which is the best available approximation to the Sankoff algorithm. It is important to keep in mind that although the Sankoff algorithm can, in principle, maximize sequence similarity alongside with free energy minimization, its Dynalign implementation minimizes gap penalties, but otherwise ignores sequence content.

The quantitative results in the previous section show that the new alternative method is comparable or better in the quality of predictions, and much faster computationally. In that section, results from both tools were compared with a 'gold standard', which is much easier than comparing them with each other, because they pursue different objectives.

Remember that we have not presented another approach to implement the Sankoff algorithm, but we have significantly changed the problem definition. Although the Sankoff approach determines a sequence alignment reflecting a common set of base pairs, consensus shape prediction produces a consensus abstract shape together with its shrep for each sequence, but no alignment. Since this deviates from the traditional and accepted notion, let us discuss common aspects as well as differences from a conceptual point of view.

The Sankoff approach produces sequences aligned according to the predicted common base pairs, hence with the same Level 5 shape. However, their Level 3 shapes may be different, as some sequences may have gaps where others have bulges. In either case, the structures reported are not necessarily the shreps of their respective shapes. One may refold the structures individually, with the consensus base pairs fixed, but then the refolded structures may be 'out of shape' because they exhibit additional hairpins.

RNAcast predictions are unaligned. Using the predicted shreps, a multiple structure alignment may be obtained via RNAforester or a similar structure alignment tools. From the structure alignment, a sequence alignment consistent with the consensus shape may be easily derived. The structure alignment also minimizes the number of gaps, but in contrast to the Sankoff approach, it does so after structure prediction, and not simultaneously. Hence, one may expect cases where the Sankoff approach produces results that fix more strongly the relative positions of helices, whereas with RNAcast, conserved helices may move more flexibly. However, we have not observed this effect to a significant amount in our studies.

### Potential improvements

Reality differs from our evaluation scenario. Database families can be considered reliable homologs, but when a new (putative) family is investigated, we cannot be sure whether structure is preserved. With consensus shape prediction, we would like to implement a safeguard against members in the sequence set that really do not share the common shape of the rest. Such a situation will result most probably in a consensus garbage. We expect that leave-one-out tests can be designed to recognize this situation. Such tests can be implemented efficiently, because only Steps 2 and 3 of the RNAcast algorithm, but not the most expensive Step 1 must be iterated.

We have performed an overall evaluation of our new method, but have not yet tried to optimally adjust it to particular datasets. For example, when studying short molecules like microRNA precursors, Level 2 abstraction, which distinguishes 5'-, 3'-bulges and internal loops, might be more conclusive than Level 3. More systematic study and experience is needed to provide guidance about the most conclusive level of shape abstraction to be used in a particular context.

*Conflict of Interest:* none declared.

## REFERENCES

- Doshi,K. *et al.* (2004) Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. *BMC Bioinformatics*, **5**, 105.
- Gardner,P. and Giegerich,R. (2004) A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, **5**, 140.
- Giegerich,R. *et al.* (2004) Abstract shapes of RNA. *Nucleic Acids Res.*, **32**, 4843–4851.
- Gorodkin,J. *et al.* (1997) Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, **25**, 3724–3732.
- Griffiths-Jones,S. *et al.* (2003) Rfam: an RNA family database. *Nucleic Acids Res.*, **31**, 439–441.
- Gutell,R.R. *et al.* (1992) Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucleic Acids Res.*, **20**, 5785–5795.
- Höchsmann,M. *et al.* (2004) Pure multiple RNA secondary structure alignments: a progressive profile approach. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **1**, 53–62.
- Hofacker,I.L. *et al.* (2002) Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, **319**, 1059–1066.
- Hofacker,I.L. *et al.* (1994) Fast folding and comparison of RNA secondary structures. *Monatshfte f. Chemie*, **125**, 167–188.
- Lee,R. and Ambros,V. (2001) An extensive class of small RNAs in *Caenorhabditis elegans*. *Science*, **294**, 862–864.
- Lee,R. *et al.* (1993) The *C. elegans* heterochronic gene *lin-4* encodes small RNAs with antisense complementarity to *lin-14*. *Cell*, **75**, 843–854.
- Mathews,D.H. *et al.* (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.
- Mathews,D.H. and Turner,D.H. (2002) Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.*, **317**, 191–203.
- Mattick,J. and Gagen,M. (2001) The evolution of controlled multitasked gene networks: the role of introns and other noncoding RNAs in the development of complex organisms. *Mol. Biol. Evol.*, **18**, 1611–1630.
- McCaskill,J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.
- Nussinov,R. *et al.* (1978) Algorithms for loop matchings. *SIAM J. Appl. Math.*, **35**, 68–82.
- Pfeffer,S. *et al.* (2004) Identification of virus-encoded microRNAs. *Science*, **304**, 734–736.
- Sankoff,D. (1985) Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, **45**, 810–825.
- Skryabin,B. *et al.* (2003) Neuronal untranslated BC1 RNA: targeted gene elimination in mice. *Mol. Cell Biol.*, **23**, 6435–6441.
- Szymanski,M. *et al.* (2000) 5S ribosomal RNA database Y2K. *Nucleic Acids Res.*, **28**, 166–167.
- Touzet,H. and Perriquet,O. (2004) CARNAC: folding families of related RNAs. *Nucleic Acids Res.*, **32**, 142–145.
- Vialeto,S. (2004) On the computational complexity of 2-interval pattern matching problems. *Theor. Comput. Sci.*, **312**, 223–249.
- Waterman,M.S. and Smith,T.F. (1978) RNA secondary structure: a complete mathematical analysis. *Math. Biosciences*, **42**, 257–266.
- Witwer,C. *et al.* (2001) Conserved RNA secondary structures in Picornaviridae genomes. *Nucleic Acids Res.*, **29**, 5079–5089.
- Wuchty,S. *et al.* (1998) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**, 145–165.
- Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary informations. *Nucleic Acids Res.*, **9**, 133–148.