Published online September 15, 2004

Nucleic Acids Research, 2004, Vol. 32, No. 16 ***4843–4851***
doi:10.1093/nar/gkh779

Abstract shapes of RNA

Robert Giegerich*, Björn Voß[1] and Marc Rehmsmeier[1]

Institute for Bioinformatics and [1]International NRW Graduate School in Bioinformatics and Genome Research, Bielefeld University, P.O. Box 100 131, 33501 Bielefeld, Germany

Received June 9, 2004; Revised and Accepted August 1, 2004

ABSTRACT

The function of a non-protein-coding RNA is often determined by its structure. Since experimental determination of RNA structure is time-consuming and expensive, its computational prediction is of great interest, and efficient solutions based on thermodynamic parameters are known. Frequently, however, the predicted minimum free energy structures are not the native ones, leading to the necessity of generating suboptimal solutions. While this can be accomplished by a number of programs, the user is often confronted with large outputs of similar structures, although he or she is interested in structures with more fundamental differences, or, in other words, with different abstract shapes. Here, we formalize the concept of abstract shapes and introduce their efficient computation. Each shape of an RNA molecule comprises a class of similar structures and has a representative structure of minimal free energy within the class. Shape analysis is implemented in the program *RNAshapes*. We applied *RNAshapes* to the prediction of optimal and suboptimal abstract shapes of several RNAs. For a given energy range, the number of shapes is considerably smaller than the number of structures, and in all cases, the native structures were among the top shape representatives. This demonstrates that the researcher can quickly focus on the structures of interest, without processing up to thousands of near-optimal solutions. We complement this study with a large-scale analysis of the growth behaviour of structure and shape spaces. *RNAshapes* is available for download and as an online version on the Bielefeld Bioinformatics Server.

INTRODUCTION

The function of a non-protein-coding RNA is often determined by its structure. Since the experimental determination of RNA structure is time-consuming and expensive, its computational prediction is of great interest. Efficient solutions based on thermodynamic parameters have been known since (1), with improvements in the energy models (2) and extensions to related questions such as base pair probabilities (3). Ironically, for one of the most-studied classes of RNA, the transfer RNA (tRNA), predicted minimum free energy structures are frequently much different from the native cloverleaf structure, forming an elongated hairpin. This can be explained by the existence of modified bases in native tRNAs, which leads to the formation of a structure that is not the optimal under the energy model used. Similar problems are inaccuracies of the energy model, different chemical conditions in living cells and the fact that RNA molecules interact with other molecules that can alter their conformations. However, it seems reasonable that the energy of the native structure should not be too far away from the predicted minimum free energy, and thus the native structure should be present among suboptimal solutions in a small energy range above the minimum free energy. This is addressed by a number of programs that output suboptimal solutions (4,5). However, the number of suboptimal solutions grows exponentially with the size of the energy range, for moderately long sequences reaching several hundred thousand within an energy range of only a few kcal/mol. The problem has been tackled either by filtering the output to reduce the number of similar structures (6) or by directly reducing the number of solutions with the restriction to canonical structures (i.e. structures without lone base pairs) (7,8) or saturated structures (9,10). However, this step either leaves the necessity to first calculate a huge set of suboptimal solutions or reduces the output by too small an amount, with the additional danger of missing the native structure. Related approaches are the definition of macro-states (11) and statistical sampling of structures (12).

The user is usually only interested in structures that show fundamental differences. Small changes, such as additional base pairs or changing bulge loops are of minor significance. This can be addressed by modelling RNA structures with a more coarse-grained representation (13) that preserves more abstract differences such as the overall branching pattern. While a definition of an abstract representation is easy, a mathematical formulation is necessary, first, to ensure that certain properties hold, and, second, to allow for an efficient computation of suboptimal abstract representations.

Here, we formalize the concept of abstract shapes and introduce their efficient computation. Based on the notion that structures are formed by juxtaposition and embedding, we define abstract shapes in an analogous way, which makes them homomorphic images of structures. An abstract shape class has a representative structure with minimum free energy. Abstract shapes blend perfectly into the framework of Dynamic Programming, which is commonly applied to the calculation of minimum free energy structures and related questions, and can thus themselves be computed

*To whom correspondence should be addressed. Tel: +49 521 106 2913; Fax: +49 521 106 6411; Email: robert@techfak.uni-bielefeld.de

efficiently. The algorithm is implemented in the program *RNAshapes*.

We applied *RNAshapes* to the prediction of optimal and suboptimal abstract shapes of a tRNA, the HIV-1 leader RNA and the human small nuclear RNA (snRNA) U2. For a given energy range, the number of shapes was considerably smaller than the number of structures, and in all cases, the native structures were among the top shape representatives. This demonstrates that the user of *RNAshapes* can quickly focus on the structures of interest, without processing up to thousands of suboptimal solutions.

We complement the exemplary study with a large-scale analysis of the growth behaviour of structure and shape spaces on sequences from the Rfam database (14). Although both the structure and shape space grow exponentially with sequence length and suboptimal energy range, the shape space grows considerably slower. For growing sequence lengths or energy ranges, the ratio of shape space size and structure space size is decreasing.

RNAshapes is available for download and as an online version on the Bielefeld Bioinformatics Server (http://bibiserv.techfak.uni-bielefeld.de/rnashapes/).

DEFINING ABSTRACT SHAPES

Our aim is to obtain a holistic view of the near-optimal or even the complete folding space of a given RNA sequence. We shall partition the folding space into different classes of structures, by means of abstracting from structural detail. We call these classes abstract shapes, or shapes for short. To characterize an RNA molecule by studying its shapes, these classes must be disjoint. This allows us to collect meaningful statistics. We may be interested in the number of shapes, the size distribution of shapes or the distribution of free energy within and across shapes. Such analyses must be efficiently computable, in spite of the exponential size of the folding space.

Classes of structures can be defined in many ways, but a few requirements seem appropriate to catch the intuition of a 'shape': when we feel (either intuitively or in some formal sense) that two structures are similar, they should either have the same shape or their shapes should be similar in the same sense. Within each abstract shape, we want to designate a concrete structure as its representative, such that looking at all the representatives gives a meaningful overview of what is there in the folding space. Furthermore, each abstract shape should also have an explicit representation, which is not a concrete structure and independent of primary sequence. Only then, we can study questions of comparative analysis, such as can two sequences (of possibly different lengths) fold into the same shape?

The domain of sequences is closed under juxtaposition-concatenating sequences s and t, we obtain the sequence st. The same holds for structures; if x and y are hairpins, and we paste the 3′ end of x to the 5′ end of y, then we obtain a structure that is simply an external loop with two adjacent hairpin structures. In addition to juxtaposition, structures are formed by recursive embedding. For example, implanting three adjacent hairpins into the loop of a fourth, we obtain a cloverleaf structure. Being formed by juxtaposition and embedding, structures are inherently tree-like. Although they can be represented in many ways—such as strings, circle graphs, squiggle plots or base pair lists—a tree representation is the one that can be used for all purposes without introducing artefacts or losing explicit information. In data type theory, this is called an initial data type: there is a simple mapping from the initial data type to any other representation, while an inverse mapping may be more complicated or may not even exist.

We want shapes to be homomorphic images of structures, which means that when structure x is embedded in structure y, then the shape of x is also embedded in the shape of y. For this reason, the principles of juxtaposition and embedding must apply to the shape domain as well. Before going into technical detail, we arrive at the following definitions:

Definition 1. *Let S be the tree-like domain of structures, and P a tree-like domain of shapes. A shape abstraction is a mapping π from S to P that preserves juxtaposition and embedding.*

Two structures x and y have the same shape when $\pi(x) = \pi(y)$. Two sequences s and t have a common shape if they have structures x_s and x_t such that $\pi(x_s) = \pi(x_t)$. To compare the shapes of two structures, they need not have the same primary sequence, or even sequence similarity or equal sequence lengths. Turning now to the folding space $F(s)$ of a given RNA sequence s, we define the desired classes as inverse images of π:

Definition 2. *For a given RNA sequence s, its (concrete) folding space $F(s)$ is the set of all legal structures according to the rules of base pairing. Its (abstract) shape space is $P(s) = \{\pi(x) \mid x \in F(s)\}$. The class of p-shaped structures in $F(s)$ is $\{x \mid x \in F(s), \pi(x) = p\}$.*

In other words, the shape class p is $\pi^{-1}(p) \cap F(s)$. As the inverse image of a function always induces an equivalence relation, we can define unique representatives:

Definition 3. *The representative structure $\hat{p}$ for shape class p is the element that has minimal free energy among all structures in the class.*

There is the rare case that two structures in a shape have the same energy, in which case we consider the smallest one under a lexicographic ordering on trees as the representative.

The shape representative structures will be called *shreps* for short, to distinguish them from an explicit representation of the shape as a whole, which we will introduce below.

The above definitions must be complemented by concrete data structures representing RNA structures and shapes. We now develop one such concretization that seems convenient and intuitive, but others might work just as well (see Appendix). For simplicity of presentation, we refrain from modelling dangling bases at the end of helices, but this could be added.

The different structural components in RNA are single-stranded regions, hairpin loops, stacking regions, bulges on the 5′ or on the 3′ side, internal loops and multiloops. Furthermore, we have lists of adjacent structures, such as the components of the external loop. Pseudoknots that can be represented as trees, such as the canonical simple recursive pseudoknots introduced by R. Giegerich and J. Reeder (submitted for publication), could also be included, but we choose to ignore them here. The above structural components are

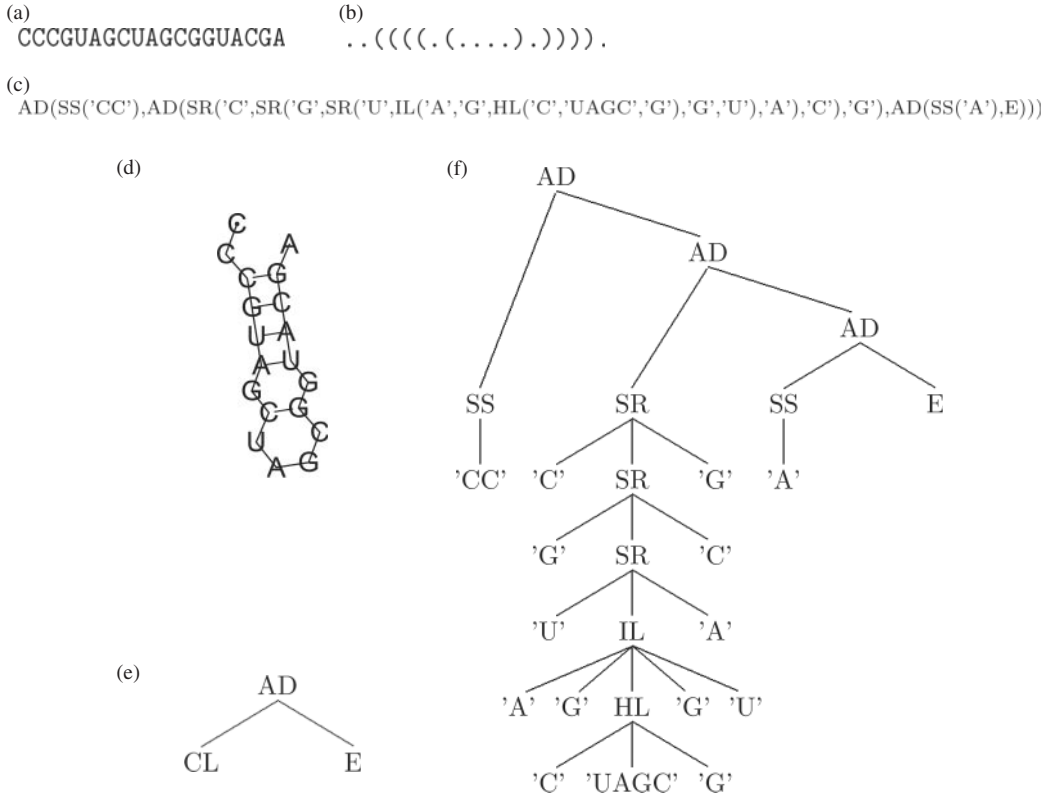

Figure 1. Representations of RNA secondary structure. (**a**) Primary sequence, (**b**) Vienna, (**c**) ASCII-tree, (**d**) squiggle, (**e**) shape and (**f**) tree.

denoted by node labels SS, HL, SR, BL and BR, IL, ML and AD, respectively. For technical reasons, we need a label to denote an empty list of adjacent components (E). Individual nucleotides A, C, G and U, as well as strings thereof, represent themselves. Figure 1 shows a particular structure in several representations, including the one defined here. Note that in the tree representation, the primary sequence can be read from the leaves of the tree in left to right (5′ to 3′) order. In the text and for computer input, trees will be written as formulas, such as $AD(SS(ACGUU), E)$ or $AD(HL(C, UUU, G), E)$.

In the domain of shapes, we only care about open and closed structures, branching and adjacency. These situations are represented by node labels OP, CL, FK (from 'fork', as BR is already used above), AD and E. (Re-using AD and E in the domain of shapes should not give rise to confusion; they are just generic list constructors.)

The abstraction mapping π from structures to shapes is defined by the following equations. We use variables a and b for nucleotides l and l' for loop sequences, c for a list of adjacent components and x for arbitrary structures.

$$\pi(SS(l)) = OP \quad \textbf{1}$$

$$\pi(HL(a,l,b)) = CL \quad \textbf{2}$$

$$\pi(SR(a,x,b)) = \pi(x) \quad \textbf{3}$$

$$\pi(BL(a,l,x,b)) = \pi(x) \quad \textbf{4}$$

$$\pi(BR(a,x,l,b)) = \pi(x) \quad \textbf{5}$$

$$\pi(IL(a,l,x,l',b)) = \pi(x) \quad \textbf{6}$$

$$\pi(ML(a,c,b)) = FK(\pi(c)) \quad \textbf{7}$$

$$\pi(AD(SS(l),c)) = \pi(c) \quad \textbf{8}$$

$$\pi(AD(x,c)) = AD(\pi(x),\pi(c)) \quad \text{for } x \neq SS(l) \quad \textbf{9}$$

$$\pi(E) = E \quad \textbf{10}$$

It is easy to see that this abstraction function retains hairpins and multiloops, but abstracts from stack lengths, bulges, internal loops and single-stranded regions (except in the case of the completely unpaired structure). It completely abstracts from primary sequence. This abstraction might be too strong in some cases, especially with short sequences that do not have much chance to show shape variation on this level of abstraction. In such cases, weaker abstraction functions that retain more structural detail can be defined in a similar way. Our tool *RNAshapes* supports five different abstraction functions.

Although all the computational analysis of shapes is based on these tree representations, it is convenient for the human eye to introduce string representations for structures as well as for shapes. We define a notation for shapes, using

homomorphism ν_P as follows: $\ldots_k$ means k dots, $|l|$ is the length of string l and ε denotes the empty string.

$$\nu_P(OP) = \_ \quad \textbf{11}$$

$$\nu_P(CL) = [] \quad \textbf{12}$$

$$\nu_P(FK(c)) = [\nu_P(c)] \quad \textbf{13}$$

$$\nu_P(AD(x,c)) = \nu_P(x)\nu_P(c) \quad \textbf{14}$$

$$\nu_P(E) = \varepsilon \quad \textbf{15}$$

This is analogous to the familiar 'Vienna' notation for structures, here defined as ν_S:

$$\nu_S(SS(l)) = \cdots_{|l|} \quad \textbf{16}$$

$$\nu_S(HL(a,l,b)) = (\cdots_{|l|}) \quad \textbf{17}$$

$$\nu_S(SR(a,x,b)) = (\nu_S(x)) \quad \textbf{18}$$

$$\nu_S(BL(a,l,x,b)) = (\cdots_{|l|}\nu_S(x)) \quad \textbf{19}$$

$$\nu_S(BR(a,x,l,b)) = (\nu_S(x)\cdots_{|l|}) \quad \textbf{20}$$

$$\nu_S(IL(a,l,x,l',b)) = (\cdots_{|l|}\nu_S(x)\cdots_{|l|}) \quad \textbf{21}$$

$$\nu_S(ML(a,x,b)) = (\nu_S(x)) \quad \textbf{22}$$

$$\nu_S(AD(x,c)) = \nu_S(x)\nu_S(c) \quad \textbf{23}$$

$$\nu_S(E) = \varepsilon \quad \textbf{24}$$

Note the simple recursive definitions, whereas a direct definition of the mapping from the Vienna string $\nu_S(x)$ to the corresponding shape's notation $\nu_P(\pi(x))$ requires a parsing function. Such simplicity is the advantage of using a tree representation. Figure 2 shows some structures in Vienna notation, together with their shape notation under the abstraction function π.

Any sensible notation function must be injective, i.e. it must not map two distinct objects to the same notation. This appears to be violated, as $\nu_S(IL(a,l,x,l',b)) = \nu_S(ML(a,AD(x,E),b)) = (\nu_S(x))$. However, the recurrences that analyse the search space have been designed to be unambiguous (15), and hence a candidate of the form $ML(a,AD(x,E),b)$, a non-branching multiloop, is never considered. With this in mind, it is easy to show that both ν_S and ν_P are injective. Hence, our program *RNAshapes* can keep the tree representations for itself, and faithfully communicate with its users via the string representations of structures and shapes.

Implementing shape analysis for a given RNA sequence s would be impractical if we had to compute first $F(s)$ and then map it into shapes via π. We would suffer from exactly the exponential explosion we want to circumvent. The theory of dynamic programming tells us that a given dynamic programming algorithm can efficiently compute any homomorphic image of its search space (16) as long as the associated objective function satisfies Bellman's principle of optimality, which means that the application of the objective function can be interleaved with the solution of subproblems. In RNA folding, the search space is $F(s)$, and homomorphic images are $F(s)$ itself, base pair maximization, free energy minimization, exact counts and E-values, shapes, vienna strings, shape strings, *shreps* as vienna strings and more. Moreover, we can use them in various combinations via the so-called product algebras (R. Giegerich and P. Steffen, manuscript in preparation). In the application section, we show analyses that compute all the shapes of near-optimal structures, together with their *shreps*, for a certain energy threshold. We also present analyses on the size of the shape space $P(s)$ in comparison with the size of the structure space $F(s)$. We shall not discuss implementation details here; readable (as well as executable) code can be obtained from the *RNAshapes* web site. Let us turn to the applications.

APPLICATIONS

The programs used in this section are *RNAshapes* for the prediction of shapes and *shreps*, and *RNAsubopt* from the Vienna RNA package (7) for complete suboptimal folding. They both rely on the thermodynamic energy parameters presented in (2).

Transfer RNA

tRNAs are one of the best analysed RNA families. Various experiments have revealed the biological active structure of tRNAs which is known as the cloverleaf structure. In contrast to this we found that out of 99 tRNA sequences from the Rfam database (14), only 30 have a cloverleaf as their predicted *mfe* structure (data not shown). The biological explanation for this is that tRNAs possess modified bases which may on the one hand be no longer capable of forming base pairs, or on the other hand are able to interact in a different way. This alters the free energy of the predicted conformation such that it rises above the free energy of the cloverleaf (or vice versa), letting the latter achieve the energetic optimum. For structure prediction, when the modifications are unknown, current practice is to calculate suboptimal structures for a certain energy range and to subsequently search (by eye or by a simple pattern matching algorithm) for the cloverleaf structure in the list of suboptimals. For tRNAs this means that about 50–300 structures have to be checked. To give an example we chose the *Natronobacterium pharaonis* tRNA for alanine (EMBL accession no. AB003409.1/96-167). The predicted *mfe* structure is one hairpin with three internal loops, as depicted in Figure 2a. The cloverleaf structure, shown in Figure 2c appears at position 104 in the energy sorted list of 199 suboptimals, produced by *RNAsubopt* in an energy range of 5 kcal/mol above the *mfe*. Using *RNAshapes*, we get three shapes of which the rank 3 *shrep* is the cloverleaf structure. The output of *RNAshapes* and the squiggle plots for the *shreps* are shown in Figure 2.

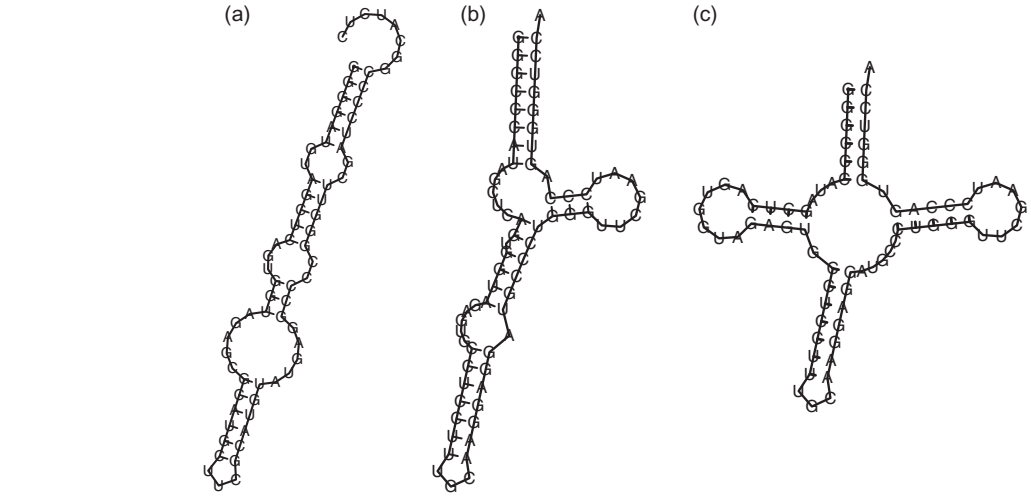

Figure 2. Predicted *shreps* for *N.pharaonis* tRNA-alanine in an energy range of 5 kcal/mol above the *mfe*. This energy range holds 199 structures. (**a**) 1st *shrep*: −35.9 kcal/mol; (**b**) 2nd *shrep*: −32.2 kcal/mol; and (**c**) 3rd *shrep*: −31.7 kcal/mol; (**d**) output of *RNAshapes*.

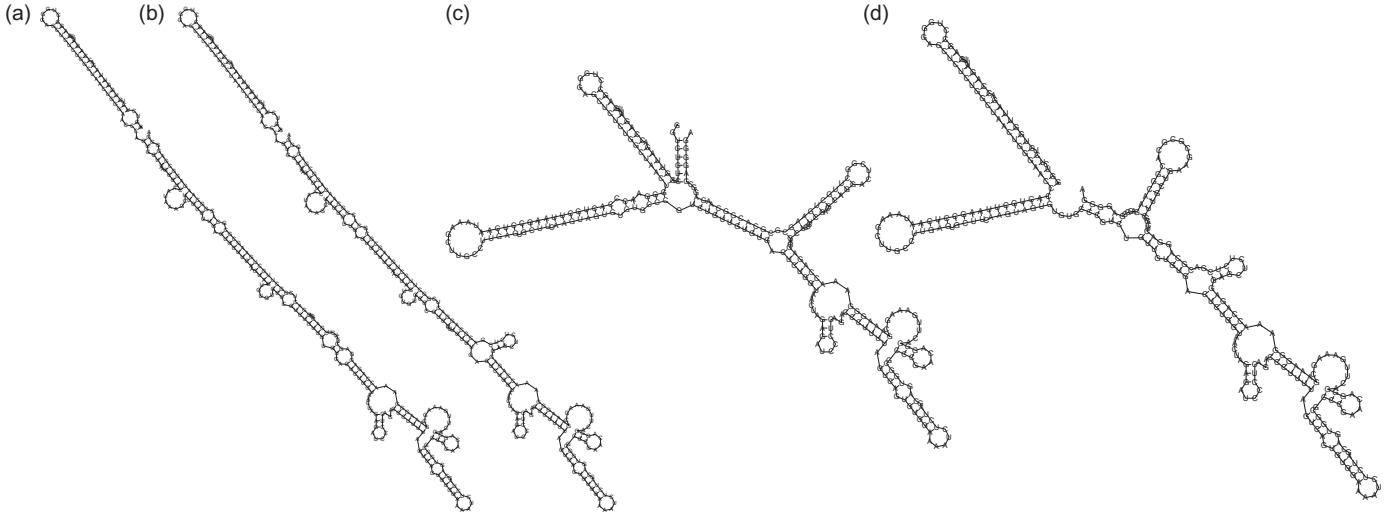

Figure 3. Subset of 19 predicted *shreps* for HIV1-leader in an energy range of 6 kcal/mol above the *mfe*. (**a**) 1st *shrep*: −108.3 kcal/mol, [][[][[][]]], S_1; (**b**) 2nd *shrep*: −107.9 kcal/mol, [][[[][[][]]]]; (**c**) 3rd *shrep*: −106.8 kcal/mol [[][[[[][[][]]]]]]; (**d**) 12th *shrep*: −102.8 kcal/mol, [][][[[[][[][]]]]][], S_2.

Leader of HIV-1 genome

The full-length HIV-1 RNA serves both as messenger RNA (mRNA) and as the viral genome. The untranslated leader of this RNA carries several regulatory elements. Their regulatory functions can be roughly divided into two groups: regulation of gene expression (transcription, translation, etc.) and virion-associated functions (dimerization, reverse transcription, etc.). Laborious experiments by Huthoff and Berkhout (17) showed that this dual nature goes parallel with two alternating conformations, a branched structure (S_2) and a more stable structure (S_1), which mainly consists of two adjacent helices. Structure prediction in an energy range of 3 kcal/mol based on the approach of abstract shapes revealed three *shreps* which are shown in Figure 3a–c. Figure 3a shows the *mfe* structure, which corresponds to the aforementioned structure S_1. The third *shrep* (Figure 3c) shows good correspondence to the conformation S_2. Further analysis, with a relaxed energy threshold of 6 kcal/mol produced 19 *shreps* and revealed that *shrep* 12 (Figure 3d) is equal to S_2. Summarizing this means that 19 *shreps* had to be checked until both correct conformations could be identified. Performing the same

approach based on complete suboptimal folding and not considering lonely base pairs would have meant checking approximately 200 000 structures.

Human snRNA U2

Human snRNA U2 is an essential part of the spliceosome and forms five stem–loops, of which four are present in the predicted energy-optimal structure (Figure 4a). The second *shrep* has all five stems and an additional central helix (Figure 4b). As shown in Figure 4, a third shape is present in the near-optimal structure space which implies structural variability. The same three conformations have been predicted based on the *paRNAss* approach presented in Voss *et al*. (18). Conversely, the structure of U2 snRNA is supposed to be important for its correct function and therefore it should have evolved to exclude equally stable but dissimilar conformations in which it could get trapped and thereby inactivated. The spliceosome is a dynamic assembly of snRNAs (U1, U2, U4, U5 and U6) and numerous associated proteins (19). Hence, a solution to the above contradiction could be that the active conformation of U2 snRNA gets stabilized by these RNAs and proteins. Kitagawa *et al*. (20) analysed human snRNA U2, and in contrast proposed only one prominent structure in the folding space. Their findings are probably due to the use of MFOLD (21), which produces a heuristic subset of all feasible structures, and their coarse grained 'tree representation distance'.

Size of the shape space

For any RNA sequence s, the number of suboptimal structures grows exponential with the sequence length N (22) as well as with the considered energy range above the *mfe*. For example, for the leader of HIV-1, with a sequence length of 281 nt, and in an energy range of 6 kcal/mol the number of suboptimal structures exceeds 200 000 even when restricting to structures without isolated base pairs. In contrast to this, the number of *shreps* is 19, and thus stays significantly smaller. In order to reveal more general properties about the growth behaviour of the folding space $F(s)$ and the shape space $P(s)$, we analysed sequences from the Rfam data base (14) with lengths ranging from 20 to 300 nt in an energy range of 5 kcal/mol. Additionally, sequences of length ∼100 nt for energy ranges from 0 to 10 kcal/mol were examined to reveal the influence of the energy range. As a last experiment, we estimated the base of the exponential expression relating the number of structures (without isolated base pairs) and shapes, respectively, to the sequence length N [$\text{size}(F(s)) = c_F * a^N$, $\text{size}(P(s)) = c_P * b^N$]. For this purpose, we computed the number of all possible structures and shapes for random sequences of various lengths. We chose 30 sequences for each length; for the shape analysis at length 120 only one data point was calculated due to computational constraints. Figure 5a and b illustrates the slower (but still exponential) growth of $P(s)$ compared to $F(s)$ with growing sequence length in an energy range of 5 kcal/mol above the *mfe*. For a growing energy range but fixed sequence length, $P(s)$ grows slower (but still exponential) than $F(s)$, too (data not shown). The ratio of shapes to structures is decreasing (asymptotically) with growing sequence length as well as with growing energy range (see Figure 5c and d). This also expresses the differences in growth rates between $P(s)$ and $F(s)$ for either sequence length or energy range. Figure 5e shows the overall number of structures and shapes for random sequences of increasing length. Their approximation by functions exponential in sequence length N gives estimates for $\text{size}(F(s))$ and $\text{size}(P(s))$. Our analyses lead to $\text{size}(F(s)) \approx 0.04 * 1.4^N$ and $\text{size}(P(s)) \approx 0.21 * 1.1^N$.

DISCUSSION

We have introduced the concept of abstract shapes and their efficient computation. We showed that the number of near-optimal structures that a researcher has to process either by eye or by automatic post-filtering approaches can be reduced from several hundred thousands to only a few. This dramatic reduction of output is a major relief for the researcher who is now very quickly directed to alternative solutions with interesting differences. The computation of shapes and *shreps* is not a heuristics. It is based on a complete evaluation of the folding space. Looking at, say, the top 10 shapes of some RNA gives

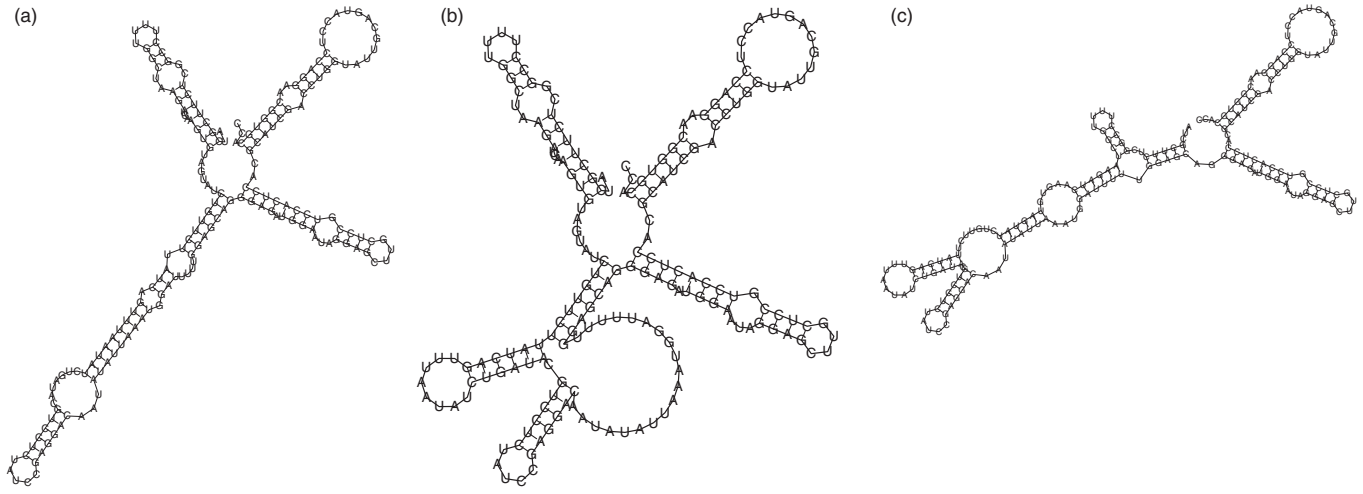

Figure 4. Predicted *shreps* for human U2 snRNA in an energy range of 3 kcal/mol above the *mfe*. (**a**) 1st *shrep*: −69.12 kcal/mol, [][][][]; (**b**) 2nd *shrep*: −68.02 kcal/mol, [][][[]][]; and (**c**) 3rd *shrep*: −67.32 kcal/mol, [][[[[]]]].

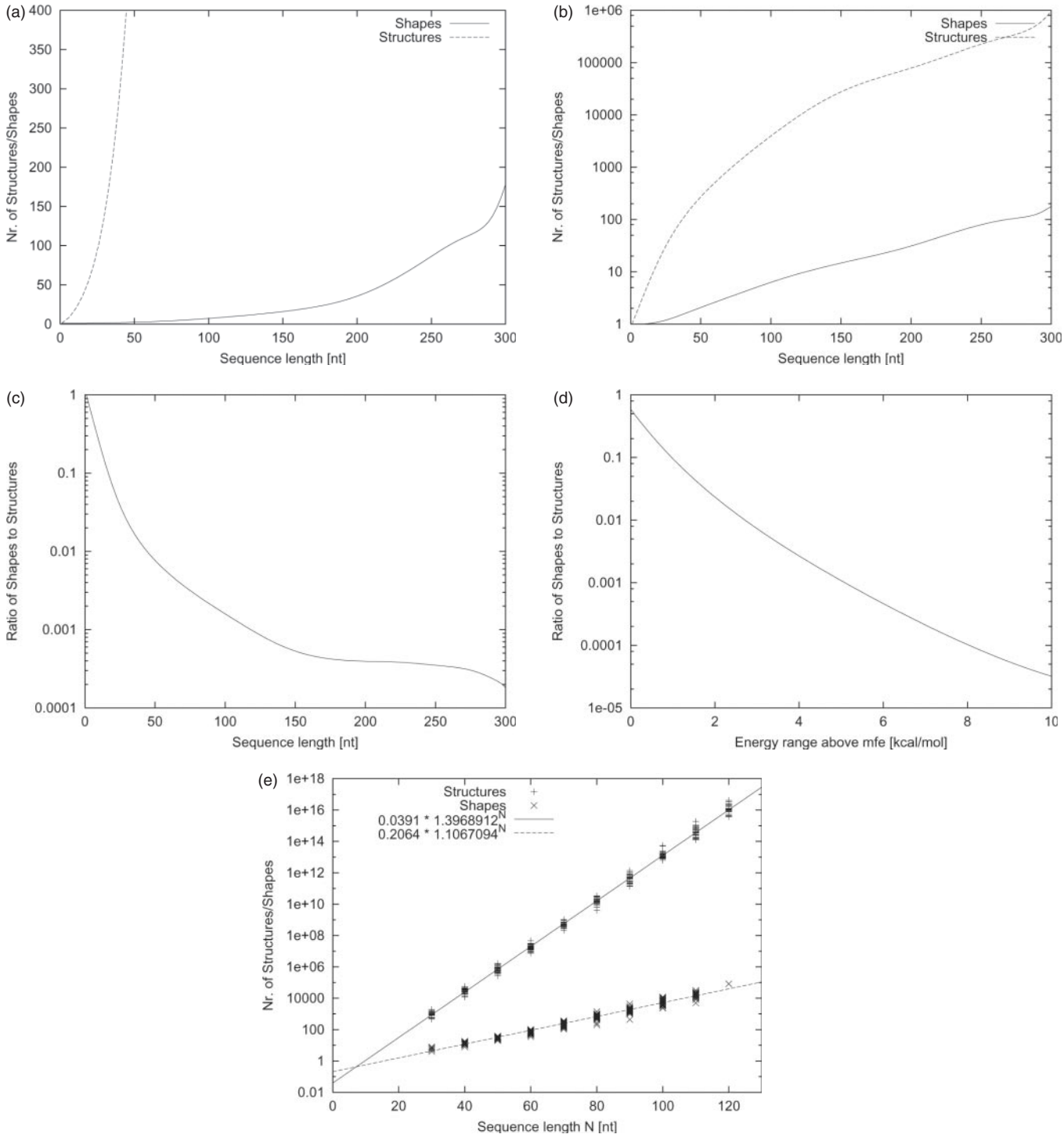

Figure 5. Comparison of folding space and shape space. (**a**) Growth of structure space, respective shape space with sequence length (energy range 5 kcal/mol). (**b**) Growth of structure space, respective shape space with sequence length (energy range 5 kcal/mol, log-scale). (**c**) Shape/structure ratio for growing sequence length (energy range 5 kcal/mol). (**d**) Shape/structure ratio for growing energy range ($n = 100$). (**e**) Overall number of structures, respective shapes (log-scale) with growing sequence length.

an unbiased view of its potentially relevant structures. Hence, it might be useful to record shape information as a very compact index in the databases that classify non-coding RNA, such as Rfam (14) or SCOR (23). Using exact string matching on shape representations, a researcher could quickly obtain a pre-classification of a novel RNA before resorting to more expensive methods.

The advantage of using shapes gets even larger in comparative studies of secondary structures. For example, structural motifs, such as the Iron Responsive Element, can be identified

by a multiple local structure alignment of suboptimal solutions from various co-regulated RNA sequences. If only two such sequences are analysed, the number of necessary pairwise comparisons grows already quadratic with the number of suboptimal solutions. Thus, the computational burden can be reduced with the computation of abstract shapes in a quadratic way, too.

The concept of abstract shapes is very flexible and allows alternative definitions with only little extra implementation effort. The definition of choice depends on the actual analysis. In this paper, we demonstrated the usefulness of our approach for RNAs that show major differences in their alternative conformations, e.g. the hairpin and cloverleaf structures of tRNAs. For other types of RNA, a finer shape definition might be of interest. For example, in miRNA precursors, one might be interested in the number and positions of individual bulges instead of abstracting to the level of complete hairpins. A shape abstraction function π', implementing this idea, is shown in the Appendix.

We see two further problems that may be adressed with the ideas presented here. One problem is the fact that *mfe* structure prediction can go wrong as it does not consider the folding kinetics. Having reduced the near-optimal folding space to a handful of *shreps*, they may go under closer (and more expensive) scrutiny as to the viability of their folding path. This would require a systematic study, using cases where there is a known effect of folding kinetics, of the question whether the folding path affects the shape of the native fold, the structure within the shape (as the *shrep* of the native shape need not be the native structure), or possibly both.

The other open problem is the *de novo* prediction of non-coding RNA genes. It seems plausible that an RNA driven by evolution to attain a specific, functional structure should carry some signal that can be detected. Previous attempts focusing on lower-than-average *mfe* have not been successful, as shown in (24). We have tested the hypothesis that a functional RNA, compared to mRNA, should have a smaller number of shapes in a certain range above *mfe*, but the results were not conclusive (data not shown). However, taking into account the folding path, as discussed above, this could be a promising road to follow.

ACKNOWLEDGEMENTS

The ideas presented here owe a lot to intense discussions with the participants at the Benasque workshop on 'Regulatory and Functional RNAs' organized by E. Rivas and E. Westhof in 2003.

REFERENCES

1. Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, **9**, 133–148.
2. Mathews,D.H., Sabina,J., Zuker,M. and Turner,D.H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.
3. McCaskill,J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.
4. Zuker,M. (1989) On finding all suboptimal foldings of an RNA molecule. *Science*, **244**, 48–52.
5. Wuchty,S., Fontana,W., Hofacker,I.L. and Schuster,P. (1999) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**, 145–169.
6. Zuker,M. (1989) *Mathematical Methods for DNA Sequences*. CRC Press, Boca Raton, FL, Chapter 7, pp. 159–194.
7. Hofacker,I.L., Fontana,W., Stadler,P.F., Bonhoeffer,L.S., Tacker,M. and Schuster,P. (1994) Fast folding and comparison of RNA secondary structures. *Chem. Monthly*, **125**, 167–188.
8. Hofacker,I.L., Fontana,W., Bonhoeffer,S. and Stadler,P.F. (2000) *RNAfold Manual v1.4*.
9. Zuker,M. and Sankoff,D. (1984) RNA secondary structures and their prediction. *Bull. Math. Biol.*, **46**, 591–621.
10. Evers,D.J. and Giegerich,R. (2001) Reducing the conformation space in RNA structure prediction. In *Proceedings of the German Conference on Bioinformatics*, pp. 118–124.
11. Flamm,C., Hofacker,I.L., Stadler,P.F. and Wolnger,M.T. (2002) Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, **216**, 155–173.
12. Ding,Y. and Lawrence,C.E. (2003) A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res.*, **31**, 7280–7301.
13. Shapiro,B.A. (1988) An algorithm for comparing multiple RNA secondary structures. *Comput. Appl. Biosci.*, **4**, 381–393.
14. Griffiths-Jones,S., Bateman,A., Marshall,M., Khanna,A. and Eddy,S.R. (2003) Rfam, an RNA family database. *Nucleic Acids Res.*, **31**, 439–441.
15. Giegerich,R. (2000) Explaining and controlling ambiguity in dynamic programming. *Proceedings of Combinatorial Pattern Matching*, pp. 46–59.
16. Giegerich,R., Meyer,C. and Steffen,P. (2004) A discipline of dynamic programming over sequence data. *Sci. Comput. Programm.*, **51**, 215–263.
17. Huthoff,H. and Berkhout,B. (2001) Two alternating structures of the HIV-1 leader RNA. *RNA*, **7**, 143–157.
18. Voss,B., Meyer,C. and Giegerich,R. (2004) Evaluating the predictability of conformational switching in RNA. *Bioinformatics*, **20**, 1573–1582.
19. Burge,C.B., Tuschl,T. and Sharp,P.A. (1999) *The RNA World, 2nd edn.* Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.
20. Kitagawa,J., Futamura,Y. and Yamamoto,K. (2003) Analysis of the conformational energy landscape of human snRNA with a metric based on tree representation of RNA structures. *Nucleic Acids Res.*, **31**, 2006–2013.
21. Zuker,M., Mathews,D.H. and Turner,D.H. (1999) Algorithms and Thermodynamics for RNA secondary structure prediction: a practical guide. In Barciszewski,J. and Clark,B.F.C. (eds), *RNA Biochemistry and Biotechnology*. NATO ASI Series, Kluwer Academic Publishers, Dordrecht, The Netherlands.
22. Waterman,M.S. (1995) *Introduction to Computational Biology*. Chapman & Hall/CRC, London, UK.
23. Klosterman,P.S., Tamura,M., Holbrook,S.R. and Brenner,S.E. (2002) SCOR: a structural classification of RNA database. *Nucleic Acids Res.*, **30**, 392–394.
24. Workman,C. and Krogh,A. (1999) No evidence that mRNAs have lower folding free energies than random sequences with the same dinucleotide distribution. *Nucleic Acids Res.*, **27**, 4816–4822.

APPENDIX

Appendix A: an alternative shape abstraction

To obtain a less abstract version of shapes, we would consider *CL* as a unary operator and define the shape abstraction mapping π' as follows:

$$\pi'(SS(l)) = OP$$

$$\pi'(HL(a,l,b)) = CL(OP)$$

$$\pi'(SR(a,x,b)) = \pi'(x)$$

$$\pi'(BL(a,l,x,b)) = CL(AD(OP, AD(\pi'(x), E)))$$

$$\pi'(BR(a,x,l,b)) = CL(AD(\pi'(x), AD(OP, E)))$$

$$\pi'(IL(a,l,x,l',b)) = CL(AD(OP, AD(\pi'(x), AD(OP, E))))$$

$$\pi'(ML(a,c,b)) = FK(\pi'(c))$$

$$\pi'(AD(SS(l),c)) = \pi'(c)$$

$$\pi'(AD(x,c)) = AD(\pi'(x), \pi'(c))$$

$$\pi'(E) = E$$

The representation mapping ν_P must be adjusted to *CL* now having an argument by replacing Equation 12 with

$$\nu_P(CL(x)) = [\nu_P(x)]$$

Under abstraction π', two structures like $((\ldots((\ldots))))$ and $((((\ldots))\ldots))$ now belong to different shapes, [_[_]] and [[_]_], where under π, they both belong to shape [].

It should be clear that intermediate levels of abstraction are also possible, e.g. by retaining bulges only when they are longer than a single nucleotide.

Appendix B: implementation details

RNAshapes makes use of a grammar describing the folding space of RNA including dangling bases and disallowing isolated base pairs. The evaluation is based on algebras like the ones shown in Equations 11–15 and 16–24 for shapes and 'Vienna' notation, respectively. Analogous to these examples, an algebra for free energy calculation scores the structural elements with their energy contribution obtained from the thermodynamic energy parameters. Our implementation makes use of a combination of these three algebras in a triple-algebra of the form: (energy, shape, 'Vienna' notation). The essential part of the implementation is the objective function h which filters the list of (intermediate) solutions and keeps entries with lowest free energy for each distinct shape. It is defined as follows:

$$h([s_1, \ldots, s_n]) = \hat{h}([\,], \text{filter}(\text{e\_range}, [s_1, \ldots, s_n])), \text{where}$$

$$\hat{h}([sh_1, \ldots, sh_m], [s_1, \ldots, s_n])$$

$$= \hat{h}(\text{insert}(s_1, [sh_1, \ldots, sh_m]), [s_2, \ldots, s_n]), \text{where}$$

$$\text{insert}((x_e, x_s, x_v), [\,]) = [(x_e, x_s, x_v)]$$

$$\text{insert}\big((x_e, x_s, x_v), [(y_e, y_s, y_v)_1, \ldots, (y_e, y_s, y_v)_m]\big)$$

$$= \begin{cases} [(y_e, y_s, y_v)_1, \ldots], & x_s = y_s \;\&\&\; x_e \geqslant y_e \\ [(x_e, x_s, x_v), (y_e, y_s, y_v)_2, \ldots], & x_s = y_s \;\&\&\; x_e < y_e \\ [(y_e, y_s, y_v)_1, \text{insert}\big((x_e, x_s, x_v), [(y_e, y_s, y_v)_2, \ldots]\big)], & x_s \neq y_s \end{cases}$$

s_k refers to an (intermediate) solution and sh_k to an (intermediate) 'shape-optimal' solution, where 'shape-optimal' means that it attains the (so far) lowest free energy for its shape. s_k and sh_k are both of the triple-form (energy, shape, 'Vienna' notation). The function *filter* removes solutions that have higher free energy than the current minimal solution plus the chosen energy range (e_range).