Molecular Biology, Vol. 37, No. 5, 2003, pp. 723–732. Translated from Molekulyarnaya Biologiya, Vol. 37, No. 5, 2003, pp. 850–860.
Original Russian Text Copyright © 2003 by Gorbunov, Mironov, Lyubetsky.

BIOINFORMATICS

UDC 577.1

Search for Conserved Secondary Structures of RNA

K. Yu. Gorbunov[1], A. A. Mironov[2], and V. A. Lyubetsky[1]

[1] *Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, 101447 Russia;*
E-mail: gorbunov@iitp.ru
[2] *State Research Center GosNIIGenetika, Moscow, 113545 Russia*

Received April 03, 2003

Abstract—We suggest a new algorithm to search a given set of the RNA sequences for conserved secondary structures. The algorithm is based on alignment of the sequences for potential helical strands. This procedure can be used to search for new structured RNAs and new regulatory elements. It is efficient for the genome-scale analysis. The results of various tests run with this algorithm are shown.

Key words: RNA secondary structure, algorithm of dynamic programming, tRNA structure, RFN structure, regulation of transcription, regulation of translation.

INTRODUCTION

The role of the secondary RNA structure is well known. It is essential for structural RNAs (rRNAs, snRNAs), viral RNAs, ribozymes, and for regulation of gene expression. The known bacterial mRNA-level regulation systems use the attenuation principle, i.e., realize one of the two alternative structures. The “restricting” structure works as a terminator at the level of transcription, and blocks the ribosome-binding site at the level of translation. The alternative “permitting” structure works in the opposite direction, preventing formation of transcription terminator and preventing block of the ribosome-binding site. The “permitting” structure is often stabilized by various proteins and other molecules: a protein for regulation of ribosomal protein expression, low-molecular-weight ligands for regulation of riboflavin and thiamin synthesis [1, 2], other RNAs for the T-box [3]. The biological importance of RNA secondary structures made the problem of its prediction from one sequence or from the set of given sequences one of the classical problems of bioinformatics.

Various approaches have been developed to analyze and to predict RNA secondary structures. Most popular is an approach based on optimizing some function of the secondary structure quality. The dynamic programming method was first suggested by Tumanyan *et al.* in 1966 [5]. In 1980 Nussinov and Jackobson developed a dynamic programming procedure to search for a secondary structure with the maximal number of paired bases for the given sequence [6]. Later, Zuker suggested to use dynamic programming to search for structures with minimal free energy (also for a given sequence); this procedure admits additional restrictions obtained from the experimental data [7–9]. This Zuker’s algorithm (http://www.bio-info.rpi.edu/~zukerm/rna/) at present remains the most common procedure to predict secondary RNA structures. Unfortunately, all the above-mentioned approaches have some drawbacks. For example, they predict the cloverleaf structure for only 80% of tRNA sequences. An essential drawback of the approach basing on free energy minimization is its obvious dependence on the original energy parameters. Moreover, in some cases, e.g., in attenuator analysis, the “permitting” structure is of main interest, though far from being optimal as regards the attenuator mechanism. For these reasons, we think that generally the algorithms of optimization (with the functionals available to researchers) can hardly be successful.

Another direction in the study and prediction of the RNA secondary structure is dynamic analysis of secondary structure formation in the process of RNA synthesis. A kinetic model of tRNA folding [10–12] has been proposed and realized as a Monte Carlo procedure. This approach has improved our understanding of the processes involved in RNA structure formation; however, its predictive ability is also limited.

The most promising approach is based on searching for conserved structures in the sequences of isofunctional RNAs. It was successfully applied to predict the structures of tRNA, rRNA, snRNA. However, this procedure required additional information [13] obtained from experiments, and most of the work was manual; again, the studied sequences had original natural alignment, and this considerably alleviated the problem. Later works used earlier-made alignments and considered correlated substitutions preserving potential helices (e.g., see [14, 15]). Some works rely on statistical analysis of complementary segments of the sequences [16, 17]. The methods basing on genetic algorithms to search for conserved secondary RNA structures started to develop recently [18, 19].

0026-8933/03/3705-0723$25.00 © 2003 MAIK “Nauka/Interperiodica”

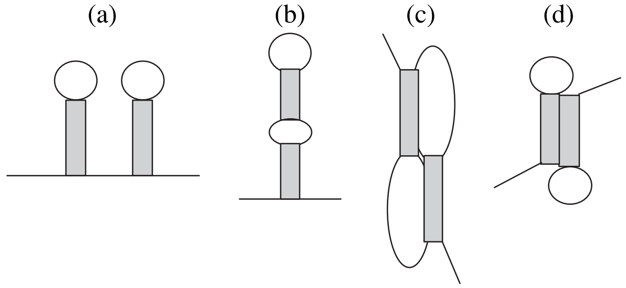

Fig. 1. Possible mutual positioning of two helices: (a) two hairpins, (b) embedded helices with an internal loop; (c) pseudoknots; (d) mutually exclusive helices.

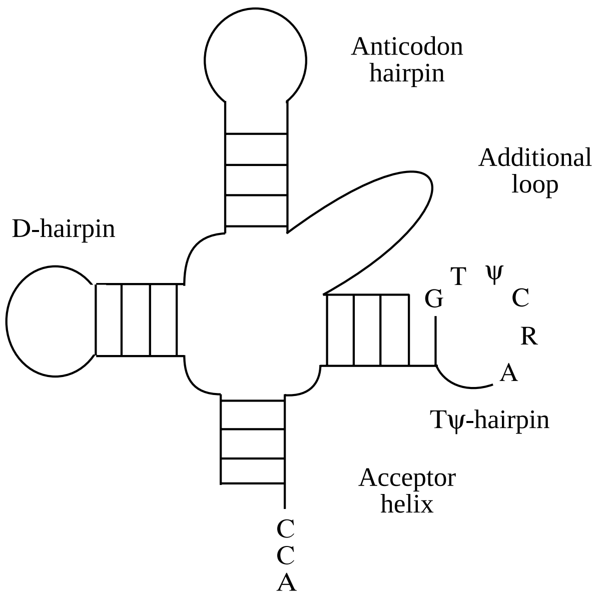

Fig. 2. Cloverleaf tRNA structure. Obviously, tRNA topology can be described by the following relation system: Ac > D, Ac > An, Ac > ψ, D/An, An/ψ (Ac, acceptor helix; D, left-side D helix; An, anticodon helix; ψ, right-side helix).

Sequences of numerous genomes are currently available. Comparative analysis of these sequences allows one to search for new conserved (and alternative) RNA structures. These large-scale studies require special methods, algorithms and software for automated and, importantly, rapid search for secondary structures.

These methods should allow the presence of some "alien" sequences in the initial sequence set, i.e., sequences that surely do not form the needed conserved structure. It is also important that these methods should require no preliminary alignment of the initial sequences. And finally, these methods should be efficient (rapid) enough to run a full-scale genome analysis in reasonable time.

PROBLEM STATEMENT

Let us call putative helix of the given sequence a pair of mutually complementary strands (a strand may contain straight parts and intervening internal and external loops). The pairs of putative helices can be of different mutual disposition. The main forms of mutual location of the helices are shown in Fig. 1. For more detailed classification see [12]. We consider overlapping helices as one of these cases. Let us consider the set of all possible putative helices (or their reasonable part selected. e.g. from energy values) for one of the given sequences. For this set of sequences, a graph is generated with nodes attributed to the putative helixes. Two nodes of the graph (two helices) are joined with an edge if these helices are "compatible within one secondary structure", where the definition of compatibility may change depending on the biological task. The situations (c) and (d) in Fig. 1 are usually considered incompatible. An edge may be considered "painted in certain color" to show mutual position of the helices ascribed to the ends of this edge. We name this graph *graph of secondary structures*.

Here the relation like "helix E contains within its loop the whole helix F" *designated* E > F plays an important role. Another important relation is E/F; it means that the whole helix E is located to the left of the helix F. Obviously, some secondary structures, i.e. mutual orientation of for cloverleaf helices can be completely described by these two relations (Fig. 2).

In most general form the problem to search for secondary structures for the *single given sequence* can be formulated as *a task to search for maximal cliques or sufficiently dense subgraphs in the graph of secondary structures* [20].

Then for the task in which we have *a given set of initial sequences* we consider the respective set of the secondary structure graphs obtained as described above (one graph of secondary structure for any sequence of the given set). *Conserved secondary structure* is a set of isomorphic subgraphs (considering colors and other parameters related with edges and nodes). In this case the task is *to search for these sets*. Obviously, a great number of these sets may be obtained (e.g. if we take one helix from each initial set).

In order to find a biologically reasonable secondary structure a quality criterion should be formulated for a conserved secondary structure; it should depend on the biological task. We can introduce additional parameters for nodes and edges of the graph, e.g. distance between helical strands and/or nucleotide sequences between helical domains (loops) and within the helices, and/or statistical parameters of the helices. In this case, the quality of the conserved structure is defined as a function of these parameters.

Unfortunately, an attempt to directly follow this task formulation by the algorithm to solve it has no chance for success, since by no doubt we would produce an ordinary trial-and-error algorithm.

Consider a set of n RNA fragments, e.g. a set of regulatory sequences from orthologous genes. Variables i and j pass through the numbers of these fragments changing from 1 to n. Our task is to find in these sequences some similar (homologous) secondary structures, remembering that some sequences may lack these structures (these sequences are named *“alien”*).

In many cases, it is often interesting and sufficient to analyze the ranking of potential helices in all non-alien sequences; lower rank within sequence i should indicate lower claims for a part of the putative secondary structure of this sequence i. This sort of task does not consider a problem to find the secondary structure of the sequence i (this problem can be however solved aiming to check the helix ranking within the sequence set by comparison of the “optimal” secondary structures).

The present work describes the simplified task and the algorithm to solve it; the results of testing are presented. This work is based on our earlier publications [21–23].

ALGORITHM TO SEARCH FOR CONSERVED SECONDARY STRUCTURE

Simplified Problem Statement and Algorithm Scheme

First we search each nucleotide sequences for all or appropriate part of the potential helices (the criterion for this selection is described below). Then each sequence is compared with each of the remaining sequences to find all analogous helices (the criterion is defined below). Then the rank is estimated for each helix as a number of sequences with helices analogous to the given one. In each of the sequences the helices with the rank above certain value are marked (these helices undergo linear arrangement). The algorithm attributes these sequences to the conserved secondary structure for the sequence under study. In order to work efficiently, the algorithm often requires consideration of the helix context, i.e. the presence in the neighborhood of the given helix of any helices of certain orientation and the presence of the conserved blocks (nucleotide words in certain positions of this neighborhood), etc.

The search for these blocks can be run independently of the algorithm described above, e.g. using an algorithm of multiple alignment.

However, the search for analogous helices in two given nucleotide sequences appears a complicated task. Some further simplifications can be introduced into the initial task. Any helix is composed of the two strands: left strand and right strand. If we consider the strands independent of each other (keeping in mind, however, whether it is originally ‘left’ or ‘right’ and from what helix it came, then the task to search for analogous helices in two nucleotide sequences may be reduced to the task to align new special sequences (the sequences of helix strands). This problem is efficiently solved using the dynamic programming methods. We name these sequences *strand sequences* (complete definition is given below).

Now we can define more exactly the helix ranking (the helix is represented by two strands in the strand sequence i which is obtained from nucleotide sequence i) as a sum (for two strands and all j) of the weights obtained from each of these strands after alignment of the strand sequence i with each of the other strand sequence j (obtained from nucleotide sequence j). Now we have the simplified statement of the problem and the scheme of our algorithm. Evidently, the rankings allow us to select the optimal helices in each of the given nucleotide sequences, to rank these helices, and to reveal alien sequences as those containing helices with low total rank.

The Algorithm

Following the scheme described in the above section, the algorithm contains seven steps.

1. Selection of putative helices for each of the nucleotide sequences (the sequences are indexed i or j, and the helices for one sequence are indexed k or l).

2. Generation of the helix strand sequences selected from sequence i for each i.

3. Multiple alignment of all strand sequences or pair-based alignment of each sequence pair.

4. Selection and ranking of the helices for each nucleotide sequence i. At this step the sequences with total rank of the selected helices below the given critical value can be discarded (the algorithm considers these sequences as “alien sequences”).

5. Generation of putative optimal (“consensus”) secondary structure for each nucleotide sequence i.

6. Generation of the consensus secondary structure from the set of secondary structures obtained as described above. Practically, step 5 is often inessential, and step 6 in principle can be replaced with Zuker’s algorithm. For these reasons, these two steps are not described here. Their description can be found in [23].

7. Steps 1–6 are repeated as a cycle with changing input parameters till certain quality of the secondary structures (or certain quality of the consensus structure) is obtained.

Realization of steps 1–4 is discussed below.

1. Search for putative helices creates no special problem. It may use a vocabulary procedure (similar to BLAST, [24]), which allows one to run fast search for the helices, though with some losses, or a slower Smith–Waterman algorithm. We counted the energy of each of the helices as the total energy of all its complementary bases, subtracting charges for looping, internal loops, and too long external loop. The energy of the two complementary fragments was calculated as a sum of energy values for each of the interacting complementary nucleotide pairs. These energy values depend on the temperature, and for this work we took energies for 37°C. For each possible combination of the four parameters A, B, C, and D (strand ends) one more parameter is calculated and stored: maximal energy E for all helices with these ends; a helix with these values of the five parameters is also generated. After completion of the list, the helices are ordered for increasing energy, and then the list is cut from the end to the given length (we left 40–85% of the whole list).

To summarize, at this step the algorithm generates the sets of putative helices $F_i = \{h_k\}_i$ for each of the initial nucleotide sequences S_i.

2. Generation of the helix strand sequences. For each set F_i we generate the set F_i', composed of the strands from all helices from the set F_i. More precisely, F_i' contains all pairs of the type $\langle k, d\rangle$, where variable k numbers all helices from F_i, and the discrete variable d is once equal l, and once equal r for each k. The pair $\langle k, l\rangle$ means left strand of the helix k, and the pair $<k, r>$ means the right arm of the same helix k. Each of the sets $\langle k, r\rangle$ is linearly arranged, i.e., transformed into the sequence named *strand sequence i* and corresponding to the initial nucleotide sequence *i*. We tested two ways of arrangement: by coordinates of the centers of the strands (most often), bit also by first coordinate for the left strand and by last coordinate of the right arm.

To summarize, index *i* numbers all strand sequences (similarly to all initial nucleotide sequences when *i* runs from 1 to *n*).

3. Alignment of the two strand sequences. The following common dynamic programming procedure was applied at this step.

Consider the recursive formula of dynamic programming applied to the set of the helical arms:

$$f(k, I) = \max\{ f(k-1, I-1) + W(k, I),$$
$$f(k-1, I) - d, f(k, I-1) - d, 0\},$$

where f is alignment quality for the subset ending with the pair (k, I), $W(k, I)$ is the weight of similarity for the strands k and I (explained below), and d is charge for deletion.

Then we define the *function of similarity (identity)* $W(h_1, h_2)$ for any pair of helices h_1 and h_2 from different sets of the helices. Let A_i, D_i be the external ends of, respectively, right and left strands of helix h_i, and the internal ends of the two strands be B_i, C_i. Now two words can be formed for each of the helices $S_{il} = [A_i - o_A, B_i + o_B]$ and $S_{ir} = [C_i - o_C, D_i + o_D]$; these are two strands of helix i with some flanks. Here $-o$ and $+o$ mean shift left or right by o nucleotides. The algorithm has a possibility to delete all external and internal loops from the strands and the opposite possibility to delete all straight interacting strand fragments. Flank sizes o_A, o_B, o_C, o_D (left and right) are algorithm parameters (their numeric values usually were not higher than 15).

Now consider

$$W(h_1, h_2) = W(S_{1l}, S_{2l}) + W(S_{1r}, S_{2r}),$$

where S is the weight of local alignment of the two words in brackets, calculated using, e.g., the Smith–Waterman algorithm [23]. If the resulting similarity of the two helices is below certain critical value W^*, than this value is substituted with a big negative number (1000 in our calculations). Then the similarity $W(h_1, h_2)$ is corrected by introducing charges for the different length of the external loops of the helices h_1 and h_2 (charges for the long external loops can also be introduced, as well as charges for the difference in strand lengths, charges and bonuses for presence or absence of the conserved nucleotides in certain position of the strands, etc.)

This approach needs some explanations. At the step of the strand sequence alignment *similarity function* $W(h_1, h_2)$ of the two helices (playing a role similar to that of the common charges and bonuses for alignment of the two sequences) is corrected to allow only alignment between left or between right arms. If aligning two strand sequences we align both strands of the helix k with respective strands of the same helix r (obviously, quite possible for this not to happen), then we name the helices k and r *aligned*. A bonus is added to the weight of the strand $l_p(i)$ if not only the strand is aligned, but also the corresponding helix.

Similarly, if helix k is aligned with helix r and helix r is aligned with helix s (from one more nucleotide sequence), then in the case when helix k is aligned with helix s, all the involved strands receive additional bonus (the *"triangle rule"*).

Finally, an essential addition to our algorithm is the following. Beside the set F_i' of the strands consider the set G_i of conserved blocks within the same sequence *i*, which also can be defined F_i'. This sequence reflects the mutual location of the strands and of the blocks. We name it *strand-box sequence*. This sequence can be analyzed as described above, alignment allowed for blocks only with blocks, and for strands only with strands. This results in simultaneous alignment of the helix strands and of the conserved blocks.

The main idea of our algorithm is based on the fact that homologous helices within the conserved structures often have similar strand sequences; therefore, alignment of the "sequences" of helices can be substituted with alignment of the strand sequences (or better, with alignment of the strand-box sequences).

To summarize, the problem described in this section is much simpler than the initial graph-based problem statement. However, this simplification has an obvious drawback: it becomes more difficult to consider the relation of "being the strands of the same helix."

4. Ranking of the helices. We define the helix quality *h* as follows. Consider pairwise alignment of the sequence 1 with each sequence *j* (where *j* runs from 2 to *n*). The total weight for all these alignments can be ascribed to the strand *l* of the sequence 1; we name this sum the *weight* of the strand *l*. We define $l_0(1)$ the strand l_0 from the sequence 1 for which the weight is maximal.

Replacing number 1 with sequence number *i*, similar procedure is used to find the strand $l_0(i)$. It is natural to suppose that function $l_0(i)$ provides good candidate secondary structures for initial sequence *i*. Moreover, we can declare alien the sequences with the strand weight $l_0(i)$ below the given critical value; these sequences are then deleted from the initial nucleotide sequence set. The next strands after $l_0(i)$ are found similarly and named $l_1(i)$, etc. In each of the remaining (non-alien) sequences we obtain a fixed number of the "optimal" strands $l_0(i)$, $l_1(i)$, $l_2(i)$, …, $l_p(i)$. This will result in selection and ranking of some "optimal" helices for each of the initial nucleotide sequences.

The algorithm described here has the following parameters: maximal size of the external loop, minimal size of the external loop, maximal size of the internal loop, minimal length of the uninterrupted paired fragment of the helix, maximal number of GT pairs within the fragment, size of the strand flanks used to estimate helix similarity, coefficients of penalty for too long external loops and for their varying lengths, maximal number of helices left for alignment, critical similarity value, minimal number of the sequences having a helix similar to the given one, maximal number of the helices left after alignment to generate secondary structure, indicator of possibility for the structural overlaps, alignment parameters (bonus for same letters, penalty for deletion, and penalty for different letters); number of iterations for alignment.

The algorithm is implemented as console application for Windows 9x/NT/2000. The executive version of the program and detailed testing results are available online (http://www.iitp.ru/lyubetsky), and can be requested via E-mail (gorbunov@iitp.ru).

TESTING RESULTS AND DISCUSSION

Testing for a tRNA Case

First testing results for the algorithm presented here were obtained using 18 tRNA fragments from *Escherichia coli*, each 75-base long. These fragments represent bacterial genes and are taken from: http://ncbi.nlm.nih.gov/. Names of these tRNA genes from complete genome of *E. coli*, fragment lengths, and the secondary structure helices found with our algorithm are shown in Table 1.

Some false helices generated by the algorithm should be noted. Their strands are usually located near the strands of the biological confirmed helices. The false helix either forms a pseudo knot with the real helix (in this case, obviously, the real helix cannot be found), or is a variant of the true helix (in this case both helices are found). In the latter case, the weight of the true helix usually (more than in 68% of our analyses) is higher than that of the false helix.

In a real problem one cannot be sure that all the studied sequences form the given secondary structure. For this reason we run a special test to evaluate stability of the algorithm toward dilution of the initial sequence set with some "alien" sequences. We added random Bernoulli sequences in varying numbers (1, 3, 5, 7, 9, etc.) and applied the algorithm to the obtained extended samples. Similarly, we added random flanks to the original sequences. The results showed high stability of the algorithm toward both addition of the "alien" sequences and toward addition of random sequence 3' and 5' flanks.

The complete results of these calculations are available (http://www.iitp.ru/lyubetsky). Here we show only a small part of the results and only general statistics for dilution of the original sample (Tables 2–4).

The first case to analyze is the "pure" tRNA set (same as in Table 1) loaded with nine random sequences. The set size increased 1.5-fold. As seen from Table 2 and similar results available on the web (http://www.iitp.ru/lyubetsky), increasing of the random sequence number induces gradual decrease of the helices found by the algorithm: from 82 to 73.5%. One can also note that it becomes difficult to find a D-helix; this is explained by its small length and low conservation of the respective nucleotide fragment.

Table 3 shows percent efficiency for finding each of the four tRNA helices depending on the number of the added random sequences (0, 1, 3, 5, 7, or 9).

It should be noted that the algorithm does not consider any specificity of the tRNA structure, and no specificity for the structures of RFN, T-boxes and S-boxes analyzed below.

The decreasing in efficiency is not monotonous, since the number of false helices appearing at addition of the random sequences or flanks is highly variable.

Table 1. The results of algorithm testing

Gene *E. coli*	Length, nt	ACC	D	A	Ψ	False helices
alaV	76	+	+	+	+	2
alaX	76	+	+	+	+	1
cysT	74	+	–	+	+	2
aspU	77	+	–	+	+	2
gltU	76	+	–	+	+1, –2	1
pheV	76	+	+	+	–	1
glyT	75	+	–	–	+	3
glyV	76	+	+	+	+	1
glyU	74	+	+	+	+	1
hisR	77	+	+	+	+1, –2	0
IleV	77	+	+	+2, –1	+	0
IleX	76	+	+	+1, +7	–	1
lysT	76	+	+	+	–	1
leuQ	87	+	–	–2, +2	+	0
leuW	85	+	–	–	+	2
leuX	85	+	+	+	+	1
leuU	87	+	–	+	+	0
leuZ	87	+	–	+	+	2
%		100	56	89	83	

Note: Columns 3 to 6 show the results for the following helix types: ACC, acceptor helix; D, D-helix; A, anticodon helix; Ψ, pseudouridine helix (see Fig. 2). Plus indicates that a helix is detected *precisely*, two numbers mean an error of helix detection (their modules show the errors for the two ends of the external loop, while the signs show direction of the shifts); minus means that no true helix was detected by the algorithm. The last column shows the number of false helices detected by the algorithm, and the last row shows mean detection efficiency (%) for each of the helix types.

Obviously, in a real situation one does not know the exact location of the secondary structure searched for. That is why we tested the pure set of tRNAs (same set as above) adding biological flanks of 10–40 nucleotides to each of the sequences from this set. In this case the algorithm found additional biologically significant helices (from 1 to 3), which belong to the upstream or to the downstream tRNA genes. This is not surprising since the tRNA genes are known to be clustered.

The situations when a secondary structure is tandemly repeated several times in the genome is extremely rare (except for the tRNAs). Therefore, we performed another test series adding 10–40-nucleotide-long right and left flanks of the random Bernoulli sequences. Table 4 shows the results for the 40-nt random flanks. Complete results of this analysis are available online (http://www.iitp.ru/lyubetsky).

The prediction efficiency in this case slowly decreases with increasing flank length: from 82% to 65% at average for all helices found by the algorithm.

Case of the RFN Structure

The case of the tRNAs is a common test for secondary structure analysis. However, the real structures often are much longer, much less conserved, and have more helices and more complicated helix configuration. Therefore we further tested the algorithm for prediction of the RFN element.

The RFN structure [1] regulates expression of the genes involved in riboflavin biosynthesis and transport. It contains one stem and four helices (Fig. 3); other helices are less conserved and were not searched for. Compared with the case of tRNA, the RFN element contains loops of considerably more diverse size (mainly from 5 to 50), and this affects working of the algorithm. The sequence lengths varied from 119 to 170.

The results of the two test runs obtained using our algorithms are shown below. Same 39 RFN fragments were analyzed. We considered the nucleotides in both left and right flanks of each helix (ten nucleotides from each side of the arm). No restriction for the

Table 2. The results of testing the algorithm by addition of nine random sequences (for designations see Table 1)

Gene	Length, nt	ACC	D	A	Ψ	False helices
alaV	76	+	+	+	+	0
alaX	76	+	+	+	+	1
cysT	74	+	–	+	+	1
aspU	77	–	–	–	+	1
gltU	76	+	–	+	+1, –2	1
pheV	76	+	+	+	–	1
glyT	75	+	–	–	+	2
glyV	76	+	+	+	+	0
glyU	74	+	–	+	+	0
hisR	77	+	+	+	+1, –2	0
IleV	77	+	+	–	+	0
IleX	76	+	+	+1, +7	–	1
lysT	76	+	+	+	–	1
leuQ	87	–	–	–2, +2	+	1
leuW	85	+	–	–	+	2
leuX	85	+	+	+	+	1
leuU	87	–	–	+	+	0
leuZ	87	+	–	+	+	1
%		83	50	78	83	

length of the external loop was introduced for the first test (column 2 of Table 5).

For the second test (column 3 of Table 5) the restriction from above for the external loop length (35) was introduced into the algorithm. Obviously, the stem could not be found in this case, and the stem was not searched for. However, in this case the algorithm efficiency is higher because of the considerably low number of helices to analyze, the time needed for the analysis is much shorter, so that much larger samples can be analyzed.

Estimates of the helix predictions are shown in Table 5. Plus sign means almost exact prediction of the helix, with the possible shift for each strand not larger that half strand length; plus/minus means that the predicted helix differs from the real one by not more than six nucleotides, and minus sign means that the helix is either not found or found with a large shift. However, it should be noted that we have only preliminary information about the positioning of helices within the RFN structure; therefore, the shifts are indicated relative to a *putative* biological response; in some cases our results considered imprecise here can actually prove true. This problem requires special biochemical analysis.

In the case when the paired fragments of the helices are mostly not less than four nucleotides long (as, e.g., in T-box or S-box structure), it appears reasonable to make larger the algorithm parameter that defines the minimal length of paired fragment for any helix considered in alignment. This results in significant reduction of the number of the helices considered. Complete results of this analysis are available online (http://www.iitp.ru/lyubetsky).

Table 3. Detection of the four helix types (%) with different number of added random sequences

Number of random sequences	0	1	3	5	7	9
ACC-helix	83.3	72.2	83.3	83.3	83.3	66.7
D-helix	66.7	44.4	50	44.4	44.4	44.4
A-helix	77.8	88.9	83.3	77.8	66.7	83.3
Ψ-helix	94.4	88.9	83.3	83.3	72.2	72.2
Mean, %	80.6	73.6	75	72.2	66.7	66.7

Table 4. The results of testing the algorithm by addition of random 40-nt flanks on both sides (for designations see Table 1)

Gene	Length, nt	ACC	D	A	Ψ	False helices
AlaV	76	+	+	+	+	2
AlaX	76	+	–	+	+	2
Cyst	74	+	–	+	+	2
AspU	77	–	–	–	+	1
GltU	76	–	–	+	+	3
PheV	76	+	+	+	–	3
GlyT	75	+	–	–	+	4
GlyV	76	+	+	+	+	2
GlyU	74	+	–	+	+	2
HisR	77	+	–	+	+	2
IleV	77	–	–	+	+	1
IleX	76	+	+	+	+	0
LysT	76	+	+	+	+	0
LeuQ	87	–	–	–	+	5
LeuW	85	–	–	+	+	1
LeuX	85	–	–	+	+	2
LeuU	87	–	–	+	+	2
LeuZ	87	–	–	+	+	3
%		56	28	83	94	

Table 5. Prediction efficiency (%) for five helix types of the RFN element

Helix number	No upper limit for external loop (set to 170)			Upper limit for external loop set to 35		
	+	±	–	+	±	–
1	82	8	10	0	0	0
2	64	23	13	33	62	5
3	72	7	21	62	10	28
4	77	0	23	77	5	18
5	0	41	59	28	49	23

Note: (+) Helix predicted precisely (shift for each strand not exceeding half of the strand length); (±) found helix differs from the true one by no more than six nucleotides; (–) helix not detected or detected with a large shift.

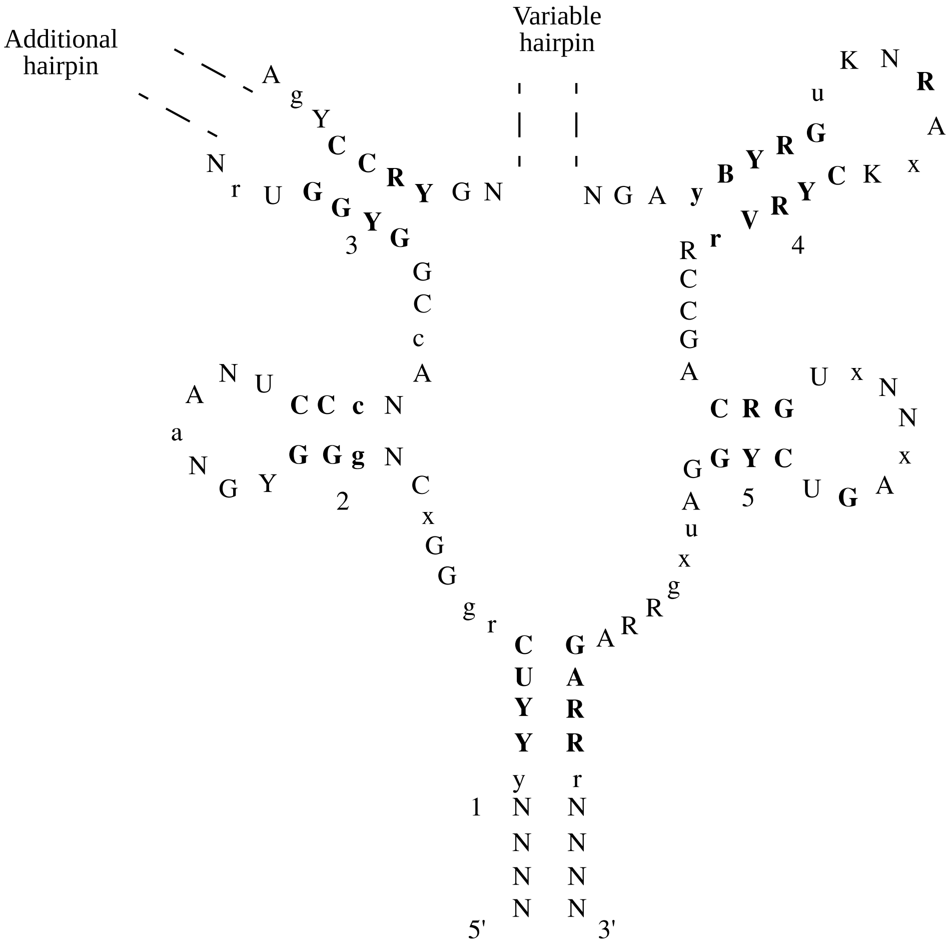

Fig. 3. RFN element. Five helices are shown.

In addition, false helices can often be distinguished from the true helices by the quality (total weight for all alignments) gained by these helices. This is evident from an example presented at our website; similar testing results for the cases of T and S boxes are also shown there.

Beside good quality of prediction, the algorithm proved to be rather fast. Testing of the tRNA set described above on PC Pentium-4 (2.4 GHz) took 1–3 s, and testing of the RFN files described above took 40–60 min.

REFERENCES

1. Vitreschak A.G., Rodionov D.A., Mironov A.A., Gelfand M.S. 2002. Regulation of riboflavin biosynthesis and transport genes in bacteria by transcriptional and translational attenuation. *Nucleic Acids Res.* **30**, 3141–3151.
2. Rodionov D.A., Vitreschak A.G., Mironov A.A., Gelfand M.S. 2002. Comparative genomics of thiamin biosynthesis in procaryotes. New genes and regulatory mechanisms. *J. Biol. Chem.* **277**, 48949–48959.
3. Vitreschak A.G. Computer analysis of regulation of genes, encoding aminoacyl-tRNA synthetases and amino acid biosynthetic proteins in Gram positive bacteria: T-box RNA regulatory element. Prediction of regulation of new genes, including amino acid transporters. In *The Proceedings of International School “Artificial Intelligence and Heuristic Methods for Bioinformatics.”* 2001, October 1–11. Italy, San-Miniato. 63.
4. Grundy F.J., Henkin T.M. 1998. The S box regulon: a new global transcription termination control system for methionine and cysteine biosynthesis genes in gram-positive bacteria. *Mol. Microbiol.* **30**, 737–749.
5. Tumanyan V.G., Sotnikova L.E., Kholopov A.E. 1966. On identification of secondary RNA structure from the nucleotide sequence. *Dokl. Akad. Nauk SSSR.* **166**, 1465–1468.
6. Nussinov R., Jacobson A.B. 1980. Fast Algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. USA.* **77**, 6309–6313.
7. Zuker M., Stiegler P. 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.* **9**, 133–148.
8. Zuker M. 1989. Computer prediction of RNA structure. *Meth. Enzymol.* **180**, 262–288.
9. Zuker M. 1991. Suboptimal sequence alignment in molecular biology. Alignment with error analysis. *J. Mol. Biol.* **221**, 403–420.

10. Mironov A.A., Dyakonova L.P., Kister A.E. 1985. A theoretical analysis of the kinetics of RNA secondary structure formation. *J. Biomol. Struct. Dynam*. **2**, 953–962.
11. Mironov A.A., Kister A.E. 1985. Theoretical analysis of structural rearrangements in the process of RNA secondary structure formation. *Mol. Biol.* **23**, 61–71.
12. Mironov A.A., Lebedev V.F. 1993. A kinetic model of RNA folding. *Biosystems*. **30**, 49–56.
13. Woese C.R., Magrum L.J., Gupta R., Siegel R.B., Stahl D.A., Kop J., Crawford N., Brosius J., Gutell R., Hogan J.J., Noller H.F. 1980. Secondary structure model for bacterial 16S ribosomal RNA: phylogenetic, enzymatic and chemical evidence. *Nucleic Acids Res*. **8**, 2275–2293.
14. Tahi F., Gouy M., Regnier M. 2002. Related Articles, Books, Link Out Automatic RNA secondary structure prediction with a comparative approach. *Computer Chem*. **26**, 521–530.
15. Hofacker I.L., Fekete M., Stadler P.F. 2002. Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol*. **319**, 1059–1066.
16. Gorodkin J., Stricklin S.L., Stormo G.D. 2001. Discovering common stem-loop motifs in unaligned RNA sequences. *Nucleic Acids Res*. **29**, 2135–2144.
17. Akmaev V.R., Kelley S.T., Stormo G.D. 2000. Phylogenetically enhanced statistical tools for RNA structure prediction. *Bioinformatics*. **16**, 501–512.
18. Chen J.H., Le S.Y., Maizel J.V. 2000. Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucleic Acids Res*. **28**, 991–999.
19. Titov I.I., Ivanisenko V.A., Kolchanov N.A. 2000. FITNESS-A WWW-resource for RNA folding simulation based on genetic algorithm with local optimization. *Comput. Technol*. **5**, 48–56.
20. Mironov A.A., Diakonova L.P., Kister A.E. 1984. Theoretical analysis of secondary RNA structure formation kinetics. *Dokl. Akad. Nauk SSSR*. **259**, 725–728.
21. Gorbunov K.Yu., Lyubetsky V.A. An algorithm for searching common secondary structures in a set of RNA sequences. In *The Proceedings of the third international conference of bioinformatics of genome regulation and structure, BGRS'2002*. 2002, July 14–20, Novosibirsk, Russia. **3**, 21–23.
22. Gorbunov K.Yu, Lyubetskaya E.V., Lyubetsky V.A. 2001. On two algorithms to search for alternative secondary RNA structure. *Informatsionnye Protsessy.* **1**, 178–187. (http://www.jip.ru/).
23. Gorbunov K.Yu., Lyubetsky V.A. 2002. An algorithm to search for conserved secondary RNA structures within a set of RNA fragments. *Informatsionnye Protsessy.* **2**, 55–58. (http://www.jip.ru/).
24. Altschul S.F., Gish W., Miller W., Myers E.W., Lipman D.J. 1990. Basic local alignment search tool. *J. Mol. Biol*. **215**, 403–410.