AN ALGORITHM FOR SEARCHING FOR COMMON SECONDARY STRUCTURES IN A SET OF RNA SEQUENCES

* ***Gorbunov K.Yu.***, ***Lyubetsky V.A.***

Institute for Information Transmission Problems RAS, Moscow, Russia, e-mail: lyubetsk@iitp.ru, gorbunov@iitp.ru

Key words: *secondary RNA structure, common structure, alignment, tRNA*

Resume

Motivation: We propose an algorithm for searching for conservative secondary structures in a set of RNA sequences. Its complexity is quadratic in the sum of lengths of the input sequences. The main idea of the algorithm is a concurrent alignment of sequences of possible structure elements.

Results: The algorithm was tested on various kinds of conservative secondary RNA structures. Practical applicability of the algorithm was demonstrated—about 70–80% of the biological hairpins were found by this algorithm.

Availability: The software is available on request directed to authors.

Introduction

Regulatory RNA secondary structures are often similar in related genomes. This raises the problem of predicting such structures in a family of RNA sequences. This problem seems to be far from effective algorithmic solution in general situation. Secondary structure consists of hairpins. Every hairpin consists of two ordered sequences of segments located from left to right at each sequence having some bulges between neighbor segments. Under hairpin loop, we mean the segment between these ordered sequences, which are called half-stems of the hairpin. Each *i*th segment from the beginning of the left half-stem serves as a complement of the *i*th segment from the end of the right half-stem. The pair of such *i*th segments is called a helix. In other words, helix is an equivalent of a hairpin having only one segment in each sequence. Thus, we can consider helices as elementary parts of a secondary structure. Often, secondary structure contains quite long helices.

The known algorithms for prediction of secondary structures are based on comparative retrieval of the corresponding structures in a given family of RNA sequences. For example, the method of dynamic programming is used (Gorodkin et al., 1997) to construct secondary structures that are both similar and maximally powerful for every pair of sequences and for every pair of their subsequences (starting with short subsequences). An inference of structures in stochastic context-free grammars is used with the same object (Eddy, Durbin, 1994), and so on.

Our algorithm is based on a different approach. We start from a representation of secondary structures as a linearly ordered set of (left and right) half-stems of hairpins, not as a tree of hairpins. The hairpins are placed in ascending order of their coordinates. The coordinate of a left half-stem is defined as the number of position of the rightmost nucleotide in it. The coordinate of a right half-stem is defined as the number of position of the leftmost nucleotide in it. The half-stems with the same coordinates are ordered arbitrarily. In conservative structures, homologies of half-stems are ordered similarly. This demonstrates the main idea of our algorithm—a concurrent alignment of half-stems of hairpins.

Methods and Algorithms

For any n given RNA sequences, we construct a large list of possible helices with a length not less than a prescribed value and with the distance between half-stems lying in a prescribed range (we construct only the helices continuing in both directions as far as possible taking into account that helices often have this property in secondary structures). We join into hairpins those helices whose left and, respectively, right half-stems are located at the small distance from each other. So, n lists of hairpins $L1,\ldots, Ln$ will be obtained.

For each pair of hairpins from different lists, we estimate their similarity. Thus, the base of similarities of hairpins is created. In biological hairpins, similar regions can lie in both helices and bulges, or even outside of a hairpin at a small distance from it. Our program enables us to take into account these possibilities: there are parameters defining the way of constructing a word from a hairpin. The words constructed above are compared by the Smith–Waterman method (see, for example, Waterman, 1989). To be more precise, a variant of this method aimed to find the most similar subwords in given words was

* Corresponding author

used (we also used other algorithms of the same type). We can correct similarities estimated by this method by setting certain parameters: penalty for difference in length of hairpin loops, penalty for long hairpin loops, and so on. All the similarities that are less than a chosen threshold *t* are ignored (they are replaced with 0).

Now, our aim is to refine the lists *L*1,…, *Ln* so that they would contain only the hairpins that form the desired structure. At the first (rough) stage, we delete each hairpin *h* from each list so that that the number of lists containing a hairpin *h1* with similarity (*h,h1*)≥ *t* is small.

The second (main) stage of the refinement demonstrates the principal idea of the algorithm. We transform each list of hairpins into a list of their left and right half-stems ordered according to increase in their location (coordinate) in RNA sequence (for a left half-stem, we choose its end as coordinate; for a right half-stem, we choose its beginning as coordinate). We consider a list of half-stems as a word whose letters are half-stems. Thus, each pair of lists can be aligned with the above-mentioned variant of the Smith–Waterman algorithm (or by any other algorithm of that type). It is natural that we allow the left half-stems to match only with left half-stems and, analogously, for right half-stems. When two half-stems are matched, we take their similarity from the above-mentioned base—this is the similarity between the corresponding hairpins. Taking into account that, as a rule, many superfluous hairpins are present in our lists, it is reasonable to choose null or small penalty for deletion of a half-stem.

After each pair of lists has been aligned, we count the quality of every hairpin—a value that reflects how often its half-stems were matching. While counting the quality, we assign a special price for *complete matching* of the hairpin. The complete matching occurs when half-stems of a hairpin is matched with half-stems of the same hairpin. Some price is also assigned to a hairpin *h* when two hairpins being completely matching with *h* are completely matching to each other.

Hairpins with null quality are deleted from the lists. The remaining hairpins are involved in the second iteration of alignments, after which the qualities of the hairpins are calculated again. The second iteration proceeds similarly to the first iteration but with two differences. First, the similarity of two half-stems is not merely taken from the base but is updated with regard to the qualities ascribed to the corresponding hairpins at the first iteration. Second, calculation of the qualities after the second iteration is more rigorous: we take into account only the complete matching. The computer program allows any chosen number of iterations to be performed; however, testing showed that two iterations suffice as a rule.

After the second stage of refinement of the lists, we form a joined (for all the sequences) list *L* of hairpins in descending order of their qualities (we can bound the length of *L*). The last stage of the algorithm is constructing of secondary structures. In each sequence, a structure is built independently of other sequences by our modification of Nussinov–Jacobson algorithm. It is known (Nussinov, Jacobson, 1980) that this algorithm constructs the most powerful structure on a given sequence (and on its each subsequence) by the method of dynamic programming (starting with short subsequences). Our modification of this algorithm consists in the following. First, we use half-stems of hairpins of the list *L,* not nucleotides, as primary elements. Second, instead of the most powerful structure, we build the structure with a maximal sum of qualities of the hairpins.

Let us seek a structure of "clover leaf" kind, that is, one helix with a long loop containing several helices with short loops (for example, as in the structure of tRNA) within this loop. Our algorithm can be amplified as follows (analogous possibility is provided for other kinds of secondary structures). At the last stage of the algorithm, we allow pairing of hairpin half-stems only if this hairpin has short loop or has long loop with desired number of helices in it already constructed. Similar improvement is provided for the stage of half-stem alignment. Certainly, particular conserved nucleotides can also be taken into account.

Implementation and Results

Let us describe the result of testing of the algorithm on 18 fragments of *Escherichia coli* tRNA. Below, after the number of organism and the anticodon, we cite the helices of real (biological) structures that were found by the algorithm (i.e. that are present in the structure suggested by the algorithm). The letter H denotes the lower helix (handle); L, the left helix; U, the upper helix; and R, the right helix. The number in brackets indicates how many superfluous helices were output (absent in the real structures). Sometimes when a false helix F lies near a real helix H, the algorithm may output F instead of H. The results below contain one such sample; the distances between the left and right ends of the hairpin loops of the false helix H and the real helix R are indicated in square brackets.

DA1660 TGC: H,L,U,R(1); DA1661 GGC: H,L,U,R(1); DC1660 GCA: H,U,R(0);

DD1660 GTC: H,U,R(1); DE1660 TTC: H,R(2); DF1660 GAA: H,L,U,R(0);

DG1660 TCC: H,U,R(1); DG1661 GCC: H,L,U,R(1); DG1662 CCC: H,L,U,R(0);

DH1660 GTG: H,L,U,R[1,2](1); DI1660 GAT: H,L,U,R(0); DI1661CAT: H,L,U,R(1);

DK1660 TTT: H,L,U,R(0); DL1660 CAG: U,R(2); DL1661 TAG: H,R(2);

DL1662 CAA: H,U,R(2); DL1663 GAG: H,U,R(1); DL1664 TAA: H,U,R(0).

We also carried out an extensive testing of the algorithm for other kinds of regulatory secondary RNA structures including RFN structures, regulating riboflavin biosynthesis and transport genes in various bacteria (Vitreschak et al, 2002). The

detailed results of this testing are submitted for publication in the electronic Journal *Information Processes* (http://www.jip.ru).

Discussion

The program admits one more stage: comparison of the structures constructed with each other and indication of the consensus structure together with its (partial) maps for the given structures. Such maps provide a possibility to predict the hairpins of real structures that for some reasons were not found by the algorithm.

Let us remark that all the stages of this algorithm except for the last stage can work even in the case when a real structure contains pseudoknots, that is, hairpins containing only one half-stem of another hairpin in their loop. It seems natural to use at the last stage of our algorithm a recently suggested algorithm of Rivas & Eddy (1999) with the corresponding modifications; this algorithm is designed for the same purpose as Nussinov–Jacobson algorithm but admit existence of pseudoknots. Though a time bound of this algorithm in the worst case is quite high (sixth power) but due to the fact that a few hairpins usually remain for the last stage, the algorithm of Rivas & Eddy (1999) works fast (as our computations showed).

Acknowledgments

The authors thank M.S.Gelfand and A.A.Mironov for help and for numerous explanations of biological content of the problem.

References

1. Eddy S., Durbin R. (1994). RNA sequence analysis using covariance models. Nucl. Acids Res. 22:2079–2088.
2. Gorodkin J., Heyer L.J., Stormo G.D. (1997). Finding the most significant common sequence and structure motifs in a set of RNA sequences. Nucl. Acids Res. 25:3724-3732.
3. 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.
4. Rivas E., Eddy S.R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. J. Mol. Biol. 285:2053–2068.
5. Vitreschak A.G., Rodionov D.A., Mironov A.A., Gelfand M.S. (2002). Regulation of riboflavin biosynthesis and transport genes by a conserved RNA structural element. This volume.
6. Mathematical Methods for DNA Sequences. Waterman M.S. (ed), CRC Press, Inc., Boca Raton, Florida, 1989 (translation into Russian, M.: Mir, 1999).