Optimisation problems for pairwise RNA sequence and structure comparison: a brief survey

Alain Denise[1,2,3] and Philippe Rinaudo[1,3,4]

[1] LRI, Université Paris-Sud and CNRS, France
[2] IGM, Université Paris-Sud and CNRS, France
[3] INRIA Saclay, France
[4] PRISM, Université Versailles-St-Quentin and CNRS, France

Abstract. RNA molecules play major roles in all cell processes, and therefore have been subject to a great attention by biologists, biochemists and bioinformaticians in the recent years. From a computational optimization point of view, two interrelated major issues are on one hand the problem of structure prediction, and the problem of comparing two or several RNA sequences or structures. We present a brief survey of the latter, its variants, its computational complexity issues, and optimization algorithms that have been developed up to now.

Keywords: Optimization problems, RNA structure comparison, edition, alignment

1 Introduction

RNA is among the three major molecules that are involved in life processes at cell scale. Up to the 1990's, RNA molecules were mainly considered as simple intermediates between DNA, that contains the genetic information, and proteins, that were considered as the almost unique actors in the cell processes (with some notable exceptions, as *e.g.* transfer RNAs and ribosomal RNAs). Nowadays it is known that RNA molecules play major roles in almost all cell processes [35]. The biological function of a RNA molecule is intimately related to its structure, that is its three-dimensional shape. Therefore many efforts have been done during the last years by biologists and bioinformaticians in order to study RNA structure, notably relations between sequence, structure and function. The following two problems are among the most important ones in RNA bioinformatics: structure prediction and structure comparison. They are closely related, notably they are used in combination in order to find and to fold RNA genes in whole genomes [30].

This paper focuses on RNA structure comparison, which reduces to a very challenging optimization problem from the computer science point of view. It has been subject to a large number of works in the last twenty years. We outline the state-of-the-art in this area. We notably focus on computational complexity results, and we briefly present the algorithmic and heuristic solutions that have been developed so far in order to tackle with this problem.

2 RNA edition and alignment problems

An RNA molecule is a linear polymer, composed of a succession of nucleotides, directed from the first nucleotide (5′ end) to the last one (3′ end). Each nucleotide is composed of a sugar (ribose), a phosphate, and a base, and is linked to its predecessor and its successor in the sequence by so-called phosphodiester bonds. There are four kinds of bases: Adenine (A), Cytosine (C), Guanine (G) and Uracil (U). The linear RNA molecule folds into itself according to chemical bonds that can link nucleotides together. Most of these links are made of one or several hydrogen bonds. Among these interactions between nucleotides, the so-called *canonical Watson-Crick* basepairs are the most common ones: A-U and G-C, followed by the *wobble* basepair G-U. However, any two nucleotides can form a basepair. In [27], Leontis and Westhof give an inventory and a classification of all possible base-base interactions in RNA molecules. Moreover, any base may have more than one such interaction with other bases (up to 5 in known RNA molecules), and other kinds of interactions are possible, notably between a base and a phosphate. All these interactions give the molecule a complex three-dimensional conformation that is called the *tertiary structure* of the molecule. Meanwhile, as they are hydrogen bounds, they are far weaker than the strong covalent bounds that link any two successive nucleotides in the sequence.

Using the Leontis-Westhof basepair classification, an RNA molecule can be represented at a fairly practical abstraction level by a graph whose nodes are nucleotides and edges are interactions between them. Nodes are labelled by a letter (A,C,G or U), and edges, that can be directed or not, are labelled by the kind of interaction. The graph has maximum degree 7 (for each nucleotide, one or two phosphodiester bonds, and up to five other interactions). Among the interactions, the phosphodiester bounds between two successive nucleotides in the sequence are particular: they are directed, and all together they form an Hamiltonian path in the graph. For this reason, the graph is often represented by an *arc-annotated sequence*, as shown in Figure 1. The sequence is written as it, and the pairings are denoted by so-called *arcs*.

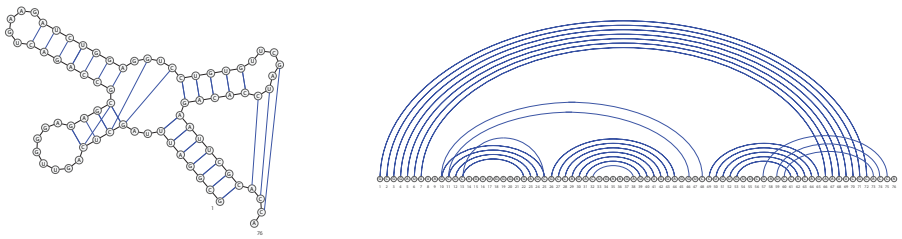

Fig. 1. Classical drawing of an RNA structure (left) and corresponding arc-annotated sequence (right). Edge labels and orientations are omitted.

For comparing combinatorial structures such as sequences, trees or graphs, a classical approach is to first define a set of basic and "atomic" operations, called edition operations, that allow to change a structure into another. Each operation is given a cost, depending on the operation and the values of its arguments. The cost of a sequence of operations that changes a combinatorial structure S_1 into another structure S_2, denoted $Cost(S_1, S_2)$, is the sum of the scores of the involved operations.

For RNA structures, a set of biologically relevant operations has been defined in [21]. Let us define a *free* base as a base that is unpaired. Each operation involves either a free base, or a basepair. They are as follows:

- Base-mismatch: replace a free base by another one.
- Arc-mismatch: replace the two bases incident to an arc, by other bases.
- Base-deletion: delete a free base.
- Arc-removing: delete an arc and its two incident bases.
- Arc-breaking: delete an arc.
- Arc-altering: delete an arc and one if its incident letters.

All these operations can be visualized in an alignment of the two arc-annotated sequences, as shown in Figure 2. Using these operations, a partial ordering can

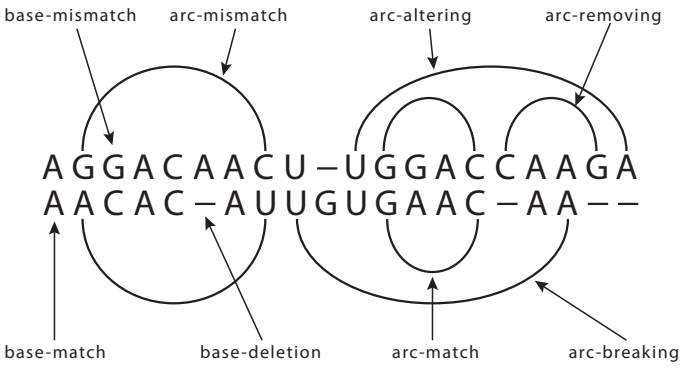

Fig. 2. Edit operations on RNA structures (from [21]).

be defined within the set of arc-annotated sequences. Let S and S' be two arc-annotated sequences. We say that S is a substructure of S', or equivalently that S' is a superstructure of S, and we write $S \trianglelefteq S'$, if S can be obtained from S' by applying a sequence of the above operations.

Now the problem of comparing two arc-annotated sequences S_1 and S_2 has two main variants, that are classical for comparing any kind of combinatorial structures:

- *Edit problem* (EDIT): find a *substructure* S of both S_1 and S_2, in such a way that $Cost(S_1, S)+Cost(S_2, S)$ is minimized. When the score function has the properties of a distance (which occurs in most applications), the cost is called the *edition distance*. Originally, the problem was described in a different way: for each deletion operation, the symmetric operation was allowed, with the same cost. And the problem was to find a sequence of operations that changes S_1 into S_2, in such a way that $Cost(S_1, S_2)$ is minimized. It has been shown in [5, 7] that both formulations are equivalent.
- *Alignment problem* (ALIGN): Find a *superstructure* S of both S_1 and S_2, in such a way that $Cost(S_1, S) + Cost(S_2, S)$ is minimized.

Equivalently, these problems can be defined in terms of scores in place of costs. In this case, the minimization problem becomes a maximisation problem.

3 Refined problem and complexity results

In [13], Evans defined four classes of structures for arc-annotated sequences, depending on constraints on their arcs:

- *Unlimited* (UNLIM): no constraint
- *Crossing* (CROS): any letter has at most one incident arc.
- *Nested* (NEST): any letter has at most one incident arc, and there is no crossing between arcs.
- *Plain* (PLAIN): there is no arc.

Clearly, these classes define a hierarchy. They correspond rather well to the levels of structure that had been defined by biologists: The PLAIN level is the so-called primary structure, NEST corresponds to what is called the *secondary structure* by biologists, while CROS corresponds to the *secondary structure with pseudoknots*. We say that there is a pseudoknot when two arcs at least are crossing (see Figures 4 and 5 for two examples of pseudoknots). Finally, the UNLIM level can contain all interactions of the *tertiary structure* of the molecule.

Given the four classes, the alignment problem can be refined by considering constraints on the class of S, the superstructure of S_1 and S_2. In the general problem S has no constraint, that is it belongs to the UNLIM class. However, the case where S must belong to a given class can be considered. This is what was done in [5, 7] by defining the *alignment hierarchy*. Given three classes C_1, C_2, C_3 among PLAIN, NEST, CROS, and UNLIM, the expression ALIGN($C_1 \times C_2 \to C_3$) denotes the problem of aligning two arc-annotated sequences $S_1 \in C_1$ and $S_2 \in C_2$, in such a way that the superstructure S belongs to C_3. This defines fourteen different problems, whose complexities are presented in Table 1. It can be shown that, in most cases, the problem ALIGN($C_1 \times C_2 \to C_3$) is equivalent to the problem EDIT($C_1 \times C_2$). These cases are indicated by a "×" in the EDIT column of the table.

Three problems are still open, to our knowledge. Among the others, the problem is NP-hard has soon as one of the sequence belongs to the CROS class.

In most cases, the problem has even been proved to be Max-SNP-hard, that means that no polynomial time approximation scheme (PTAS) can be found for it, unless P=NP.

In the table, the most complex configuration where the problem is known to be polynomial is the ALIGN(NEST × NEST → NEST) case. However, in [34] a new and more general class is considered : the class NMULT of structures that are nested but where each base can be involved in several basepairs, and it is proved that ALIGN(NEST × NEST → NMULT) can be solved in $O(n^4)$. Remark that, by contrast, the NEST × NEST → UNLIM case is Max-SNP-hard. These problems will be detailed in Section 4. The ALIGN(PLAIN × NEST → NEST) problem can be solved by a dynamic programming algorithm in $O(nm^3)$ time, where n is the size of the nested sequence and m the size of the plain sequence [21]. Finally, the case ALIGN(PLAIN × PLAIN → PLAIN) is the classical problem of sequence alignment, for which $O(n^2)$ algorithms have been known for a long time (the original algorithm [32] was in $O(n^3)$). The more recent algorithm in $O(n^2/\log n)$ makes use of sequence decompositions related to the Lempel-Ziv compression algorithm [25]. Detailed complexity results on all these problems and related ones are presented in [4].

$A \times B \to C$	EDIT	Complexity
PLAIN × PLAIN → PLAIN	×	$O(n^2/\log n)$ [8]
PLAIN × NEST → NEST	×	$O(n^4)$ [21]
PLAIN × CROS → CROS	×	Max SNP-hard [21]
PLAIN × UNLIM → UNLIM	×	Max SNP-hard [21]
NEST × NEST → NEST		$O(n^4)$ [5, 7]
NEST × NEST → CROS		unknown
NEST × NEST → UNLIM	×	NP-hard [6]
NEST × CROS → CROS		unknown
NEST × CROS → UNLIM	×	Max SNP-hard [21]
NEST × UNLIM → UNLIM	×	
CROS × CROS → CROS		unknown
CROS × CROS → UNLIM	×	Max SNP-hard [21]
CROS × UNLIM → UNLIM	×	
UNLIM × UNLIM → UNLIM	×	

Table 1. Complexities of the 14 alignment problems (adapted from [5, 7])

We indicate problems that can be formulated as edit distance problems in the second column. The other problems are specific to the ALIGN hierarchy. Complexity results are indicated for two arc-annotated sequences S_1 and S_2 s.t. $\max(|S_1|, |S_2|) = n$.

4 Nested structure-structure comparison

Pairwise comparison of nested arc-annotated sequences (that correspond to RNA secondary structures without pseudoknots) has been subject to a number of

works in the recent years. One reason is that it is much easier to know the secondary structure of a RNA molecule, by experimental biology as well as by computer prediction, than its tertiary structure. Another reason is that there exists a well known one-to-one correspondence between arc-annotated sequences and a particular family of *trees*, and comparison of trees is a well studied topic. Indeed, any nested arc-annotated sequence can be modeled by a labelled ordered tree [47, 39], where each inner node corresponds to a basepair (*i.e.* two bases with an arc between them), and each leaf corresponds to an unpaired base. The transformation algorithm, of linear complexity, is quite simple [18]. Figure 3 shows an example of an arc-annotated sequence and its corresponding tree. The

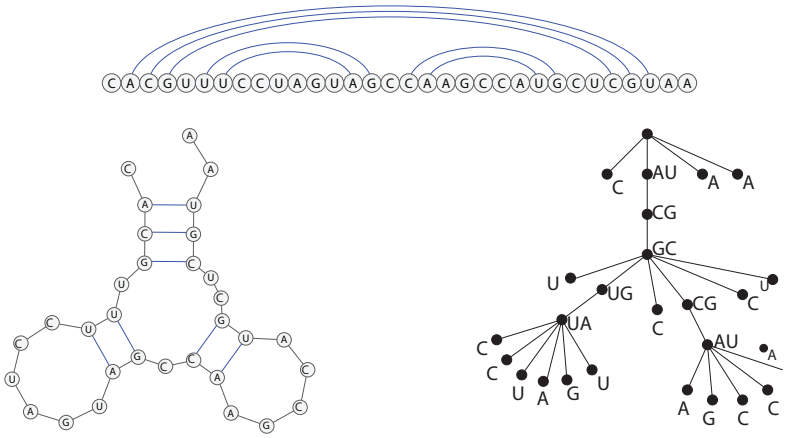

Fig. 3. A nested arc-annotated sequence (top) its classical drawing (left) and its corresponding labelled ordered tree (right).

classical edition operations for comparing trees are the following:

- node-substitution: the label of a node is changed,
- node-deletion: a node is deleted, and its children become the children of its former parent.

Given these operations, one can define, as in the case of arc-annotated sequences, a partial order relation between trees, and then consider the alignment problem and the edition problem. It turns out that these two problems are not equivalent for trees. Meanwhile, both can be solved in polynomial time by dynamic programming algorithms. The first efficient edition algorithm for ordered rooted trees is due to Zhang and Shasha [46]. It runs in $O(n^2m^2)$ worst-case complexity, and in $O(n^{3/2}m^{3/2})$ average-case complexity [10], where n and m stand for the numbers of nodes of the two trees, respectively. Some authors have given variants of the algorithm which improve the worst-case complexity [11, 23]. Alignment of

trees was first investigated by Jiang, Wang and Zhang [22]. They gave an algorithm whose worst-case complexity is in $O(n^2m^2)$. The average complexity was later proven in $O(nm)$ [18].

However, the edit operations on trees are not enough to compare RNA secondary structures in a biologically relevant way. Indeed, node-substitution corresponds to base-mismatch if the node is a leaf, or to arc-mismatch if the node is an inner node. Node deletion corresponds to base-deletion if the node is a leaf, or to arc-removing if the node is an inner node. There remain two operations on arc-annotated sequences that have no counterpart on trees: arc-breaking and arc-altering. However it is possible to define two new operations on trees that correspond to them [5, 7]. If these operations are added to the former ones, the problem of alignment (resp. edition) of trees becomes equivalent to the problem of alignment (resp. edition) of nested arc-annotated sequences. It was shown in *Blin et al.* [6] that the edition problem of two nested arc-annotated sequences (that is EDIT(NEST $\times$ NEST) or, equivalently, ALIGN(NEST $\times$ NEST $\rightarrow$ UNLIM)) is NP-hard. In contrast, it turns out that the refined alignment problem ALIGN(NEST$\times$ NEST $\rightarrow$ NEST) is polynomial, in $O(n^2m^2)$ worst-case complexity [5, 7] and in $O(nm)$ average-case complexity [10]. Roughly, the algorithm consists in a generalisation of the dynamic programming scheme for the tree alignment algorithm [22] to the new operations. The problem ALIGN(NEST $\times$ NEST $\rightarrow$ NMULT) defined and solved in $O(n^4)$ worst-case complexity in [34] cannot be directly related to tree alignment.

Since pairwise secondary structure comparison is of particular interest in RNA bioinformatics and genomics, a number of softwares have been developed so far by several teams. RNAdistance, part of the Vienna RNA package [20] and RNA_align [28] use the classical edition operations on trees only (node-substitution and node-deletion). RNAforester [19] uses the same operations, but the tree corresponding to the RNA structure is different from the classical one, so that the more complex operations on RNA structures can be 'mimicked', at the price of some constraints on the scores of the operations. Both Gardenia [5] and NestedAlign [17] implement the alignment algorithm given in [5, 7], involving all relevant RNA operations. The four above programs are based on dynamic programming and provide an optimal solution. Besides, some heuristic approaches have been developed. MiGaL [1] and TreeMatching [33] are based on the fact that RNA structures may be subject to more global modifications than the simple atomic operations described above. These two softwares use multilevel tree representations on RNA structures that allows to consider more general operations. Both approaches are heuristic but use, at the most detailed representation level, classical operations and dynamic programming algorithms.

Other heuristic approaches have been developed, mainly in order to speed-up the processing. Indeed, the $O(n^4)$ complexity, even if it reduces to $O(n^2)$ in average, is too high for efficiently comparing large sets of large molecules. Among these heuristics, RNAStrAT [15] decomposes the secondary structures into simple substructures (stem-loops), and uses an original dynamic programming algorithm for comparing pairs of stem-loops. ExpaRNA [40] first searches for a

combination of sequence-structure motifs common to the two RNAs, then uses dynamic programming to compute a longest common sequence of substructures. Some other approaches have been developed, notably in [29, 38]. In particular, in [38] an original approach using conditional random fields is presented. Finally, recently an on-line benchmark has been developed in order to offer tools for evaluating and comparing any (nested) structure-structure comparison softwares on real and synthetic datasets [2].

5 Sequence-structure comparison

The sequence-structure comparison problem is important in bioinformatics for two purposes at least:

- Searching for *non coding RNA genes* in genomes. The data are one or several RNA sequences with a known common structure on one hand (the query), and a generally very long sequence on the other hand (the target). The problem is to find, in the target, parts that could fold nearly as the given structure(s).
- Three-dimensional modeling. The data are two RNA sequences, and the tertiary structure of one of them. The other one is supposed to fold nearly the same. The problem is to predict the tertiary structure of the latter.

The differences between these two very similar problems are the following. In the first one, the target sequence is very long compared to the known query structure(s), so the algorithms must be as fast as possible; on the other hand, the result of the comparison does not need to be extremely precise. In the second one, the two sequences generally have about the same size and the result (the alignment) must be very precise.

As seen in Table 1, the Edit and Align problems are equivalent if one of the arc-annotated sequences is plain (*i.e.* without arcs). The Edit(Plain, Nest) problem is polynomial in $O(nm^3)$ where n is the length of the sequence with known structure and m the length of the sequence of unknown structure [21]. When the latter is very long, heuristics are currently used based on filters like the HMM filtering techniques described in [43], as in the famous Infernal software that is based on stochastic context free grammar [12]. In fact Infernal takes as request not a sole structured sequence, but a representation of a set of structured sequences by a special stochastic context-free grammar called a *covariance model*.

Unfortunately, answers to the problem Edit(Plain, Nest) are not precise enough for three-dimensional modeling of RNA structures. It is necessary to consider higher structure levels. The Edit(Plain, Crossing) problem is Max-SNP-hard (Table 1).

A way to tackle with the intrinsic time complexity is to use adapted heuristics. A first group of methods represent the structures at a much lower precision level, by considering structural elements (such as stems and loops) as the nodes of the structural graph. This allows to run exponential algorithms on small graphs. Notably, in [41], a dynamic programming approach has been developed over a

tree decomposition of a coarse grained structure graph. Some other approaches consist in giving only near optimum solutions. For example, in [26] the problem is changed into an integer linear program and solved by a branch-and-cut approach. However, experiments show that this approach is far to be efficient in terms of computation time. A similar representation is used in [3], where Lagrangian relaxation is used to obtain a near optimal solution in more reasonable time. The heuristic in [31] is based on the fact that, in most realistic cases, only a small part of the graph contains crossing interactions. These parts can be solved by an exponential algorithm, while the others are processed in polynomial time.

On the other hand, putting strong constraints on the costs of some operations allows to extend the dynamic programming scheme of Edit(Plain, Nest) to Edit(Nested, Crossing), hence to Edit(Plain, Crossing), leading to a $O(nm^3)$ complexity [21]. Almost all other approaches that have been developed up to now focus on restricted classes of crossing structures. It has been observed that the so-called *H-type* and *kissing-hairpin* pseudoknots (Figures 4 and 5) represent more than 80% of the pseudoknots in known structures [37]. When

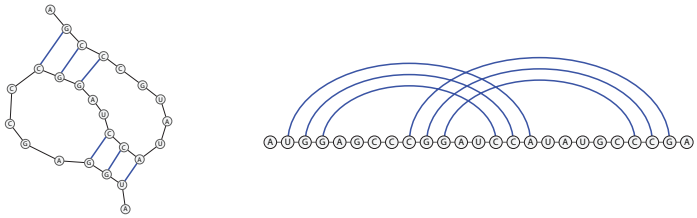

Fig. 4. An H-type pseudoknot.

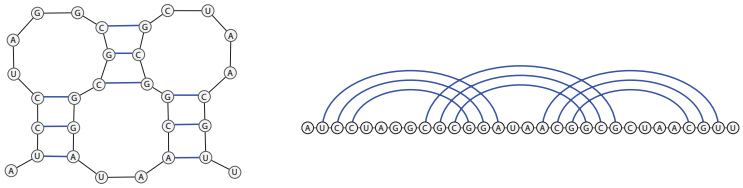

Fig. 5. A kissing-hairpin pseudoknot.

considering structures that contain only these kinds of pseudoknots, the problem can be solved in polynomial time with a worst-case complexity in $O(nm^5)$, or $O(nm^4)$ if only H-Type pseudoknots are allowed [16, 45]. The key idea of the

algorithms is that in these RNA structures, the interactions can be partially ordered in such a way that a dynamic programming scheme can be developed. Finally, it has been shown that some variants of the problem are fixed-parameter tractable, e.g when parametrized by cut-width or bandwidth [14]. This leads to polynomial algorithms (possibly of high degree) for limited classes of crossing structures.

Some works address the general sequence-structure alignment problem. In [42, 44], the problem is solved for two particular classes of unlimited structures. Recently, in [36], a parameterized structure-sequence alignment algorithm has been given for all possible structures in the UNLIM class. This work unifies all previous exact algorithms [21, 16, 45, 44] in the sense that it consists in an unique algorithm, which has the same complexity as previous approaches in their respective fields of resolution. This algorithm uses tree decompositions, it transforms the given structure into a tree-decomposition, which is then aligned with the sequence. This leads to a complexity of $O(nm^k)$ or $O(nm^{k+1})$, where k is the width of the tree-decomposition. It turns out that, for most known biological structures, a tree-decomposition with small width can be found efficiently.

6 Conclusion

Despite of the numerous works on these subjects, the sequence-structure and structure-structure problems are still not totally solved in a fully relevant manner for biological purposes. As numerous problems in bioinformatics, one crucial issue is to find a good trade-off between precision and speed. Sequence-structure alignment has become a major method for detecting RNA genes in whole genomes. Clearly, time consuming algorithms are not appropriate for this kind of research. Even when considering secondary structures only, the problem is still too time consuming. In order to scan complete genomes in a systematic way, new accurate and time performing methods have to be built. This still open question will certainly be answered by new kinds of heuristics for sequence-structure alignments, based on solid theoretical foundations. Searching for biologically relevant scoring schemes is also a very important and difficult task to be addressed, despite some promising works in this direction [17, 24].

Acknowledgments. This work was funded by the ANR (Agence Nationale pour la Recherche) project AMIS ARN (ANR-09-BLAN-0160). Figures 1, 3, 4 and 5 have been drawn with the Varna software [9].

References

1. J Allali and M-F Sagot. A multiple layer model to compare RNA secondary structures. *Software: Practice and Experience*, 38:775–792, 2008.
2. J Allali, C Saule, C Chauve, Y D'Aubenton-Carafa, A Denise, C Drevet, P Ferraro, D Gautheret, C Herrbach, F Leclerc, A De Monte, A Ouangraoua, M-F Sagot, M Termier, C Thermes, and H Touzet. BRASERO: A resource for benchmarking RNA secondary structure comparison algorithms. *Submitted for publication*, 2012.

3. M Bauer, G W Klau, and K Reinert. Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. *BMC Bioinformatics*, 8(271), 2007.
4. G Blin, M Crochemore, and S Vialette. Algorithmic Aspects of Arc-Annotated Sequences. In Zomaya Albert Y. Elloumi Mourad, editor, *Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications*, pages 113–126. Wiley, February 2011.
5. G Blin, A Denise, S Dulucq, C Herrbach, and H Touzet. Alignments of RNA structures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(2):309–322, 2010.
6. G Blin, G Fertin, I Rusu, and C Sinoquet. Extending the Hardness of RNA Secondary Structure Comparison. In Bo Chen, Mike Paterson, and Guochuan Zhang, editors, *The intErnational Symposium on Combinatorics, Algorithms, Probabilistic and Experimental methodologies (ESCAPE 2007)*, volume 4614, pages 140–151, Hangzhou, China, April 2007.
7. G Blin and H Touzet. How to Compare Arc-Annotated Sequences: The Alignment Hierarchy. In Fabio Crestani, Paolo Ferragina, and Mark Sanderson, editors, *13th International Symposium on String Processing and Information Retrieval (SPIRE 2006)*, volume 4209, pages 291–303, Glasgow, UK, October 2006.
8. M Crochemore, G M Landau, and M Ziv-Ukelson. A subquadratic sequence alignment algorithm for unrestricted scoring matrices. *SIAM Journal on Computing*, 32(6):1654–1676, 2003.
9. K Darty, A Denise, and Y Ponty. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25(15):1974–1975, 2009.
10. S Dulucq and L Tichit. RNA secondary structure comparison: exact analysis of the Zhang-Shasha tree edit algorithm. *Theoretical Computer Science*, 306(1-3):471–484, 2003.
11. S Dulucq and H Touzet. Analysis of tree edit distance algorithms. In *Proc. 14th Annual Symposium on Combinatorial Pattern Matching (CPM 2003)*, pages 83–95, 2003.
12. R Eddy and R Durbin. RNA sequence analysis using covariance models. *Nucleic Acid Research*, 22(11), 1994.
13. P A Evans. *Algorithms and Complexity for Annotated Sequence Analysis*. PhD thesis, University of Victoria, 1999.
14. P A Evans. Finding common subsequences with arc and pseudoknots. *CPM'99, LNCS*, (1645), 1999.
15. V Guignon, C Chauve, and S Hamel. An edit distance between RNA stem-loops. In *String Processing and Information Retrieval (SPIRE 2005)*, pages 335–347. Springer, Berlin/Heildeberg, 2005.
16. B Han, B Dost, V Bafna, and S Zhang. Structural alignment of pseudoknotted RNA. *Journal of Computational Biology*, 15(5), 2008.
17. C Herrbach. *Étude algorithmique et statistique de la comparaison de structures secondaires d'ARN*. PhD thesis, Université Bordeaux 1, 2007.
18. C Herrbach, A Denise, and S Dulucq. Average complexity of the Jiang-Wang-Zhang pairwise tree alignment algorithm and of a RNA secondary structure alignment algorithm. *Theoretical Computer Science*, 411(26-28):2423 – 2432, 2010.
19. M Höchsmann, T Töller, R Giegerich, and S Kurtz. Local similarity in RNA secondary structures. *Proc IEEE Comput Soc Bioinform Conf*, pages 159–168, 2003.

20. I L Hofacker, W Fontana, P F Stadler, S L Bonhoeffer, M Tacker, and P Schuster. Fast Folding and Comparison of RNA Secondary Structures. *Monatsh. Chem.*, 125:167–188, 1994.
21. T Jiang, G-H Lin, B Ma, and K Zhang. A general edit distance between RNA structures. *Journal of Computational Biology*, 9(2):371–388, 2002.
22. T Jiang, L Wang, and K Zhang. Alignment of trees - an alternative to tree edit. *Theoretical Computer Science*, 143:137–148, 1995.
23. P N Klein. Computing the edit-distance between unrooted ordered trees. In *Proc. 6th Annual European Symposium on Algorithms (ESA '98)*, pages 91–102, 1998.
24. R. J. Klein and S. R. Eddy. RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics*, 4(1):44, 2003.
25. A Lempel and J Ziv. On the complexity of finite sequences. *IEEE Trans. Inform. Theory*, 22:75–81, 1976.
26. H Lenhof, K Reinert, and M Vingron. A polyhedral approach to RNA sequence structure alignment. *Proc. 2nd Ann. Int. Conf. Computational Molecular Biology (RECOMB'98)*, pages 153–159, 1998.
27. N B Leontis and E Westhof. Geometric nomenclature and classification of RNA base pairs. *RNA*, 7:499–512, 2001.
28. G-H Lin, B Ma, and K Zhang. Edit distance between two rna structures. In *Proceedings of the fifth annual international conference on Computational biology*, RECOMB '01, pages 211–220, New York, NY, USA, 2001. ACM.
29. J Liu, J T Wang, J Hu, and B Tian. A method for aligning RNA secondary structures and its application to RNA motif detection. *BMC bioinformatics*, 6(89), 2005.
30. D H Mathews, W N Moss, and D H Turner. Folding and finding RNA secondary structure. *Cold Spring Harbor perspectives in biology*, 2(12), December 2010.
31. M Möhl, S Will, and R Backofen. Fixed parameter tractable alignment of RNA structures including arbitrary pseudoknots. *Lecture note in computer science*, 5029, 2008.
32. S B Needleman and C D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–453, March 1970.
33. A Ouangraoua, P Ferraro, L Tichit, and S Dulucq. Local similarity between quotiented ordered trees. *J Discrete Algorithms*, 5:23–35, 2007.
34. A Ouangraoua, V Guignon, S Hamel, and C Chauve. A new algorithm for aligning nested arc-annotated sequences under arbitrary weight schemes. *Theoretical Computer Science*, 412(8-10):753–764, 2011.
35. G Riddihough. In the forests of RNA dark matter. *Science*, 309(5740):1507–1507, 2005.
36. Ph Rinaudo, Y Ponty, D Barth, and A Denise. Tree decomposition and parameterized algorithms for rna structure-sequence alignment including tertiary interactions and pseudoknots (extended abstract). In *WABI 2012: 12th Workshop on Algorithms in Bioinformatics*, Lecture Notes in Computer Science, 2012.
37. Einar Andreas A. Rødland. Pseudoknots in RNA secondary structures: representation, enumeration, and prevalence. *Journal of Computational Biology*, 13(6):1197–1213, 2006.
38. K Sato and Y Sakakibara. RNA secondary structural alignment with conditional random fields. *Bioinformatics*, 21(Suppl. 2):ii237–ii242, 2005.
39. B A Shapiro. An algorithm for comparing multiple RNA secondary structures. *Computer Applications in the Biosciences*, 4(3):387–393, 1988.

40. Cameron Smith, Steffen Heyne, Andreas S. Richter, Sebastian Will, and Rolf Backofen. Freiburg rna tools: a web server integrating intarna, exparna and locarna. *Nucleic Acids Research*, 38(suppl 2):W373–W377, 2010.
41. Y Song, C Liu, X Huang, R L Malmberg, Y Xu, and L Cai. Efficient parameterized algorithms for biopolymer structure-sequence alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4), 2006.
42. K St-Onge, P Thibault, S Hamel, and F Major. Modeling RNA tertiary structure motifs by graph-grammars. *Nucleic Acids Research*, 35(5), 2007.
43. Z Weinberg and W.L Ruzzo. Sequence-based heuristics for faster annotation of non-coding RNA families. *Bioninformatics*, 22:35–39, 2006.
44. T K Wong and S M Yiu. Structural alignment of RNA with triple helix structure. *Journal of Computational Biology*, 19(4):365–78, 2012.
45. T K F Wong, T W Lam, W K Sung, B W Y Cheung, and S M Yiu. Structural alignment of RNA with complex pseudoknot structure. *Journal of Computational Biology*, 18(1), 2011.
46. K Zhang and D Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262, 1989.
47. M Zuker and D Sankoff. RNA secondary structures and their prediction. *Bull. Math. Biol.*, 46:591–621, 1984.