HAL open science

Comparing RNA structures using a full set of biologically relevant edit operations is intractable

Guillaume Blin, Sylvie Hamel, Stéphane Vialette

To cite this version:

Guillaume Blin, Sylvie Hamel, Stéphane Vialette. Comparing RNA structures using a full set of biologically relevant edit operations is intractable. 2008. hal-00347464

HAL Id: hal-00347464

https://hal.science/hal-00347464v1

Preprint submitted on 15 Dec 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparing RNA structures using a full set of biologically relevant edit operations is intractable

Guillaume Blin[1] **Sylvie Hamel**[2] **Stéphane Vialette**[1]

[1] Université Paris-Est, IGM-LabInfo - UMR CNRS 8049, France
Email: {gblin,vialette}@univ-mlv.fr

[2] DIRO - Université de Montréal - QC - Canada
Email: hamelsyl@iro.umontreal.ca

Abstract

Arc-annotated sequences are useful for representing structural information of RNAs and have been extensively used for comparing RNA structures in both terms of sequence and structural similarities. Among the many paradigms referring to arc-annotated sequences and RNA structures comparison (see (Blin et al. 2008) for more details), the most important one is the general edit distance. The problem of computing an edit distance between two non-crossing arc-annotated sequences was introduced in (Evans 1999). The introduced model uses edit operations that involve either single letters or pairs of letters (never considered separately) and is solvable in polynomial-time (Zhang & Shasha 1989).

To account for other possible RNA structural evolutionary events, new edit operations, allowing to consider either silmutaneously or separately letters of a pair were introduced in (Jiang et al. 2002); unfortunately at the cost of computational tractability. It has been proved that comparing two RNA secondary structures using a full set of biologically relevant edit operations is **NP**-complete. Nevertheless, in (Guignon et al. 2005), the authors have used a strong combinatorial restriction in order to compare two RNA stem-loops with a full set of biologically relevant edit operations; which have allowed them to design a polynomial-time and space algorithm for comparing general secondary RNA structures.

In this paper we will prove theoretically that comparing two RNA structures using a full set of biologically relevant edit operations cannot be done without strong combinatorial restrictions.

Keywords: RNA structures, Longest Arc-Preserving Subsequence (LAPCS), NP-Hardness, Stem-loops

1 Introduction

In computational biology, comparison of RNA molecules has recently attracted a lot of interest due to the rapidly increasing amount of known RNA molecules, especially non-coding RNAs. Very often, *arc-annotated sequences*, originally introduced in (Evans 1999), are used to represent RNA structures. An arc-annotated sequence is a sequence over a given alphabet together with additional structural information specified by arcs connecting pairs of positions. The arcs determine the way the sequence folds into a three-dimensional space.

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the 15th Computing: The Australasian Theory Symposium (CATS2009), Wellington, New Zealand. Conferences in Research and Practice in Information Technology (CRPIT), Vol. XX, Rod Downey and Prabhu Manyem, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

The problem of computing an edit distance between two arc-annotated sequences was introduced in (Evans 1999) with a model that used only three edit operations (deletion, insertion and substitution) either on single letters (letters in the sequence with no incident arc) or pairs of letters (letters connected by an arc). In this model, the two letters of an arc are never considered separately, and hence the problem of computing the edit distance between two arc-annotated sequences becomes equivalent (when no pair of arcs are crossing) to the tree edit distance problem, that can be solved in polynomial-time (Zhang & Shasha 1989).

To account for other possible RNA structural evolutionary events, new edit operations, such as creation, deletion or modification of arcs between pairs of letters, were introduced in (Jiang et al. 2002) at the cost of computational tractability. Indeed, it has been shown in (Blin, Fertin, Rusu & Sinoquet 2007) that in case of non-crossing arcs, the problem of computing the edit distance between two arc-annotated sequences under this model is **NP**-hard. Playing the game of applying constraints either on the legal edit operations or on the allowed alignments, several papers have shed new light on the borderline between tractability and intractability (Guignon et al. 2005, Blin et al. 2008). Of particular importance, in (Guignon et al. 2005), the authors introduced the notion of *conservative edit distance and mapping* between two RNA stem-loops in order to design a polynomial-time algorithm for comparing general secondary RNA structures using the full set of biological edit operations introduced in (Jiang et al. 2002). This algorithm is based on a decomposition in stem-loop-like substructures that are pairwised compared and used to compare complete RNA secondary structures. As mentionned in (Guignon et al. 2005), whereas in the very restrictive case of conservative distance and mapping, the computation of the general edit distance is polynomial-time solvable, it is not known if the general, *i.e.*, not conservative, edit distance between two stem-loops can be also computed in polynomial-time.

In this paper, we will show that this strong combinatorial restriction was necessary for the problem to become polynomial since it is **NP**-hard in the general case. Despite the fact that this result may be considered as purely theoretical, it proves that comparing two RNA structures using a full set of biologically relevant edit operations cannot be done without strong combinatorial restrictions.

2 Preliminaries

Given a finite alphabet Σ, an arc-annotated sequence is formally defined by a pair (S, P), where S is a string of Σ^* and P is a set of arcs connecting pairs of letters of S. In reference to RNA structures, letters are called

bases. Bases with no incident arc are called *single bases*. In an arc-annotated sequence, two arcs (i_1, j_1) and (i_2, j_2) are crossing, if $i_1 < i_2 < j_1 < j_2$ or $i_2 < i_1 < j_2 < j_1$. An arc (i_1, j_1) is *embedded* into another arc (i_2, j_2) if $i_2 < i_1 < j_1 < j_2$. Evans (Evans 1999) (see (Guignon et al. 2005) for extensions) introduced five different levels of arc structure: UNLIMITED – no restriction at all; CROSSING – there is no base incident to more than one arc; NESTED – there is no base incident to more than one arc and no two arcs are crossing; STEM – there is no base incident to more than one arc and given any two arcs, one is embedded into the other; PLAIN – there is no arc. There is an obvious inclusion relation between those levels: PLAIN $\subset$ STEM $\subset$ NESTED $\subset$ CROSSING $\subset$ UNLIMITED. An arc-annotated sequence (S_1, P_1) is said to *occur* in another arc-annotated sequence (S_2, P_2) if one can obtain the former from the latter by repeatedly deleting bases (deleting a base that is incident to an arc results in the deletion of the arc).

Among the many paradigms referring to arc-annotated sequences (see (Blin et al. 2008) for more details) we focus in this article on the LONGEST ARC-PRESERVING COMMON SUBSEQUENCE (LAPCS for short) (Evans 1999, Jiang et al. 2004, Lin et al. 2002) and the general edit distance (EDIT for short) (Jiang et al. 2002, Blin, Fertin, Herry & Vialette 2007). Indeed, as shown in (Blin et al. 2008), those two paradigms are quite related since the LAPCS problem is a special case of EDIT when considering the complete set of edit operations defined in (Jiang et al. 2002). Therefore, the hardness results for LAPCS stands for EDIT.

Formally, the LONGEST ARC-PRESERVING COMMON SUBSEQUENCE problem is defined as follows: given two arc-annotated sequences (S_1, P_1) and (S_2, P_2), find the longest – in terms of sequence length – common arc-annotated subsequence that occurs in both (S_1, P_1) and (S_2, P_2). It has been shown in (Jiang et al. 2002) that the LAPCS problem is **NP**-hard even for NESTED structures, *i.e.*, LAPCS(NESTED, NESTED). Still focussing on NESTED structures, Alber *et al.* (Alber et al. 2004) proved that the LAPCS(NESTED, NESTED) problem is solvable in $O(3^k\,|\Sigma|^k\,kn)$ time, where n is the maximum length of the two sequences and k is the length of the common subsequence searched for. The $O(3^k\,|\Sigma|^k\,kn)$ time parameterized algorithm by Alber *et al.* is by brute-force enumeration: (i) Generate all possible sequences of length k with all possible NESTED arc annotations, and (ii) For each of these arc-annotated candidate sequences, check whether or not it occurs as a pattern in both S_1 and S_2. At the heart of this approach is the fact that it can be decided in $O(n\,k)$ time whether or not this sequence occurs as an arc-preserving common subsequence (Gramm et al. 2006). It is easily see that the above algorithm reduces to $O(2^{3k-1}\,km)$ time for LAPCS(STEM, STEM). Indeed, there exist $|\Sigma|^k$ sequences of length k and hence, for a given sequence of length k, there exist $\binom{k}{i}$ different arc-annotations with i arcs. Therefore, there exist $\sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{2i} = 2^{k-1}$ arc-annotations of a given sequence of length k.

Here, we focus on the only remaining open problems concerning LAPCS and EDIT over stem-loops by showing, with a unique proof, their hardness. More precisely, we prove that LAPCS(STEM, STEM) - which may be considered as a very restricted problem and thus not interesting - is **NP**-hard **in order to infer** the **NP**-hardness of EDIT(STEM, STEM) - which is for sure, according to (Guignon et al. 2005), an interesting problem that can be used in a very simple way to compare complete RNA secondary structures. This results also prove that in any future work on comparing RNA structures with a full set of edit operations it will be necessary to introduce strong combinatorial restrictions in order to get an exact polynomial-time algorithm since even with the simpliest model, the general edit distance problem is still **NP**-complete.

3 Comparing RNA Stem-Loops is NP-complete

In this section, we prove that LAPCS over stem-loops (LAPCS(STEM, STEM)) is **NP**-complete (in Theorem 1); therefore answering an open question of (Guignon et al. 2005). This last result induces the **NP**-hardness of EDIT over stem-loops.

Theorem 1. LAPCS(STEM, STEM) *is* **NP**-*complete.*

Corollary 1. *Comparing RNA structures with a full set of biologically relevant edit operations cannot be done without introducing strong combinatorial restrictions.*

In the following, we consider the decision version of the problem which corresponds to deciding if there exists an arc-preserving common subsequence of length greater or equal to a given parameter k'.

It is easy to see that the LAPCS problem is in **NP**. In order to prove its $\mathbf{NP} - hardness$, we define a reduction from the **NP**-complete 3SAT problem (Garey & Johnson 1979) which is defined as follows: Given a collection $C_q = \{c_1, c_2, \ldots, c_q\}$ of q clauses, where each clause consists of a set of 3 literals (representing the disjunction of those literals) over a finite set of n boolean variables $V_n = \{x_1, x_2, \ldots, x_n\}$, is there an assignment of truth values to each variable of V_n s.t. at least one of the literals in each clause is true?

Let (C_q, V_n) be any instance of the 3SAT problem s.t. $C_q = \{c_1, c_2, \ldots, c_q\}$ and $V_n = \{x_1, x_2, \ldots, x_n\}$. For convenience, let L_i^j denote the jth literal of the ith clause (i.e. c_i) of C_q. In the following, given a sequence S over an alphabet Σ, let $\chi(i, c, S)$ denote the ith occurrence of the letter c in S.

We build two arc-annotated sequences (S_1, P_1) and (S_2, P_2) as follows. An illustration of a full example is given in Figures 1 and 2, where $n = 4$ and $q = 3$. For readability reasons, the arc-annotated sequences resulting from the construction have been split into several parts and a schematic overview of the overall placement of each part is provided.

Let $S_1 = C_q^1 W_q C_{q-1}^1 \ldots C_2^1 W_2 C_1^1 W_1 S_M^1 V_1 P_1^1 V_2 P_2^1 \ldots P_{q-1}^1 V_q P_q^1$ and $S_2 = C_q^2 W_q C_{q-1}^2 \ldots C_2^2 W_2 C_1^2 W_1 S_M^2\ V_1 P_1^2 V_2 P_2^2 \ldots P_{q-1}^2 V_q P_q^2$ such that for all $1 \leq i \leq q, 1 \leq k \leq n$,

- $C_i^1 = R_i^3 Q_i R_i^2 Q_i X_1^1 X_2^1 \ldots\ X_n^1 Q_i R_i^2 Q_i R_i^1$ with $X_k^1 = x_k s_j \overline{x_k}$ if $x_k = L_i^j$ or $\overline{x_k} = L_i^j$; $X_k^1 = x_k \overline{x_k}$ otherwise;
- $P_i^1 = Q_{q+i} Q_{q+i} R_{q+i}^3 X_n^1 \ldots X_{\frac{n}{2}+1}^1 R_{q+i}^2 X_{\frac{n}{2}}^1 \ldots X_1^1\ R_{q+i}^1 Q_{q+i} Q_{q+i}$ such that $X_k^1 = \overline{x_k} x_k$;
- $C_i^2 = X_1^2 \ldots X_n^2 R_i^3 Q_i X_1^2 \ldots\ X_{\frac{n}{2}}^2 R_i^2 X_{\frac{n}{2}+1}^2 \ldots X_1^2\ Q_i R_i^1 X_1^2 \ldots X_n^2$ such that for $1 \leq j \leq 3$, $\chi(j, X_k^2, C_i^2) = x_k \overline{x_k} s_j$ (resp. $s_j x_k \overline{x_k}$) if $x_k = L_i^j$ (resp. $\overline{x_k} = L_i^j$); $\chi(j, X_k^2, C_i^2) = x_k \overline{x_k}$ otherwise;
- $P_i^2 = X_n^2 \ldots X_1^2 R_{q+i}^1 Q_{q+i} X_n^2 \ldots X_{\frac{n}{2}+1}^2 R_{q+i}^2\ X_{\frac{n}{2}}^2 \ldots X_1^2 Q_{q+i} R_{q+i}^3 X_n^2 \ldots X_1^2$ with $X_k^2 = \overline{x_k} x_k$.

Moreover, let $S_M^1 = x_1\overline{x_1}x_2\overline{x_2}\dots x_n\overline{x_n}$ and $S_M^2 = \overline{x_1}x_1\overline{x_2}x_2\dots\overline{x_n}x_n$. Notice that, by construction, there is only one occurrence of each $\{s_1, s_2, s_3\}$ in C_i^2.

For all $1 \le i \le q$, let Q_i (resp. Q_{q+i}) be a segment of $n+1$ symbols y_i (resp. y_{q+i}). Moreover, for all $1 \le i \le q$, let W_i (resp. V_i) be a segment of $20(max\{q,n\}^2)$ symbols w_i (resp. v_i). Let us now define P_1 and P_2.

For all $1 \le i \le q-1$, (1) add an arc in P_1 between $\chi(1, x_k, C_i^1)$ (resp. $\chi(1, \overline{x_k}, C_i^1)$) and $\chi(1, x_k, P_{i+1}^1)$ (resp. $\chi(1, \overline{x_k}, P_{i+1}^1)$), $\forall 1 \le k \le n$ (see Figure 1.d and 2.b); (2) add an arc in P_2 between $\chi(j, x_k, C_i^2)$ (resp. $\chi(j, \overline{x_k}, C_i^2)$) and $\chi((4-j), x_k, P_i^2)$ (resp. $\chi((4-j), \overline{x_k}, P_i^2)$), $\forall 1 \le k \le n$ (see Figure 1.c, 2.a and 2.c); (3) add an arc in P_2 between $\chi(1, R_i^j, C_i^2)$ and $\chi(1, R_{q+i}^j, P_i^2)$, $\forall 1 \le j \le 3$ (see Figure 1.c, 2.a and 2.c).

Clearly, this construction can be achieved in polynomial-time, and yields to sequences (S_1, P_1) and (S_2, P_2) that are both of type STEM. We now give an intuitive description of the different elements of this construction.

Each clause $c_i \in C_q$ is represented by a pair (C_i^1, C_i^2) of sequences. The sequence C_i^2 is composed of three subsequences representing a selection mechanism of one of the three literals of c_i. The pair (S_M^1, S_M^2) of sequences is a control mechanism that will guarantee that a variable x_k cannot be true and false simultaneously. Finally, for each clause $c_i \in C_q$, the pair (P_i^1, P_i^2) of sequences is a propagation mechanism which aim is to propagate the selection of the assignment (i.e. true or false) of any literal x_k all over C_q. Notice that all the previous intuitive notions will be detailed and clarified afterwards.

In the rest of this article, we will refer to any such construction as a *snail-construction*. In order to complete the instance of the LAPCS(STEM, STEM) problem, we define the parameter $k' = 40q(max\{q,n\}^2) + 6qn + 8q + n$ which corresponds to the desired length of the solution. In the following, let (S_1, P_1) and (S_2, P_2) denote the arc-annotated sequences obtained by a snail-construction. We will denote S_d the set of symbols deleted in a solution of LAPCS problem on (S_1, P_1) and (S_2, P_2) (i.e. the symbols that do not belong to the common subsequence).

We start the proof that the reduction from 3SAT to LAPCS(STEM, STEM) is correct by giving some properties about any optimal solution.

Lemma 1. *In any optimal solution of* LAPCS *problem on (S_1, P_1) and (S_2, P_2), at least one symbol incident to any arc would be deleted. Moreover, all the symbols of V_i and W_i, for $1 \le i \le q$, will not be deleted.*

Proof. By contradiction, let us suppose that there exist at least one arc s.t. the two symbols incident to this last are not deleted in a solution of LAPCS problem on (S_1, P_1) and (S_2, P_2). Then, by construction, it induces that at least one complete sequence V_j or W_j, for a given $1 \le j \le q$, has been deleted. Since they have the same length, we will consider w.l.o.g. afterwards that V_i has been deleted. Therefore, since S_1 is, by construction, smaller than S_2 the length of this optimal solution is at most $|S_1| - |V_j| = \sum_{i=1}^{q}(|C_i^1| + |P_i^1| + |V_i| + |W_i|) + |S_M^1| - |V_j| = \sum_{i=1}^{q}((6n+11) + (6n+7) + (20(max\{q,n\}^2)) + (20(max\{q,n\}^2))) + 2n - (20(max\{q,n\}^2)) = q[12n + 18 + 40(max\{q,n\}^2)] + 2n - (20(max\{q,n\}^2))$. Then, in order for this solution to be optimal, one should have $q[12n + 18 + 40(max\{q,n\}^2)] + 2n - (20(max\{q,n\}^2)) \ge 40q(max\{q,n\}^2) + 6qn + 8q + n$. This can be reduced to $6qn + 10q - 20(max\{q,n\}^2) + n \ge 0$. But, one can easily check that for any $n \ge 3$ (which is always the case in 3SAT instances), this is not true; a contradiction. □

Lemma 2. *Any optimal solution of* LAPCS *problem on (S_1, P_1) and (S_2, P_2) is of length $40q(max\{q,n\}^2) + 6qn + 8q + n$.*

Proof. By construction, in S_1 there is (1) $\forall 1 \le i \le n$, $2q+1$ occurrences of x_i (resp. $\overline{x_i}$); (2) $\forall 1 \le i \le q$, 4 occurrences of Q_i (resp. Q_{q+i}); (3) $\forall 1 \le i \le q$, 1 occurrence of each $\{R_i^1, R_{q+i}^2, R_i^3, R_{q+i}^1, R_{q+i}^3, W_i, V_i, s_1, s_2, s_3\}$; (4) $\forall 1 \le i \le q$, 2 occurrences of R_i^2.

Whereas, in S_2, there is (1) $\forall 1 \le i \le n$, $6q+1$ occurrences of x_i (resp. $\overline{x_i}$); (2) $\forall 1 \le i \le q$, 2 occurrences of Q_i (resp. Q_{q+i}); (3) $\forall 1 \le i \le q$, 1 occurrence of each $\{R_i^1, R_i^2, R_i^3, R_{q+i}^1, R_{q+i}^2, R_{q+i}^3, W_i, V_i, s_1, s_2, s_3\}$.

Therefore, in any optimal solution there may be only (1) $\forall 1 \le i \le n$, $2q+1$ occurrences of x_i (resp. $\overline{x_i}$); (2) $\forall 1 \le i \le q$, 2 occurrences of Q_i (resp. Q_{q+i}); (3) $\forall 1 \le i \le q$, 1 occurrence of each $\{R_i^1, R_i^2, R_i^3, R_{q+i}^1, R_{q+i}^2, R_{q+i}^3, W_i, V_i, s_1, s_2, s_3\}$.

More precisely, by Lemma 1, and since, by construction, there is an arc in P_2 between $\chi(1, R_i^j, C_i^2)$ and $\chi(1, R_{q+i}^j, P_i^2)$, $\forall j \in \{1,2,3\}$, in any optimal solution, $\forall 1 \le i \le q$, only half of the $\{R_i^1, R_i^2, R_i^3, R_{q+i}^1, R_{q+i}^2, R_{q+i}^3\}$ may be conserved.

Moreover, any x_i (resp. $\overline{x_i}$) of S_1 except in C_q^1, is linked by an arc to another x_i (resp. $\overline{x_i}$), therefore by Lemma 1, in any optimal solution, $\forall 1 \le i \le q-1$, only half of the occurrences of x_i (resp. $\overline{x_i}$) may be conserved.

Finally, in any optimal solution, only half of the occurrences of $\{x_i, \overline{x_i}\}$ and one over $\{s_1, s_2, s_3\}$ in C_q^1 and S_M^1 may be conserved. Indeed, by construction, if this is not the case in C_q^1 (resp. S_M^1), it implies that at least one complete sequence Q_q (resp. V_1 or W_1) is totally deleted – which is not optimal since it is of length $n+1$ (resp. $20(max\{q,n\}^2)$).

On the whole, the maximal total length of any solution is thus equal to $40q(max\{q,n\}^2) + 6qn + 8q + n$. Moreover, this solution is composed of (1) $\forall 1 \le i \le n$, $2q+1$ occurrences of either x_i or $\overline{x_i}$, (2) $\forall 1 \le i \le q$, 2 occurrences of Q_i and Q_{q+i}, (3) $\forall 1 \le i \le q$, 1 occurrence of each $\{W_i, V_i\}$ and either s_1, s_2 or s_3 and (4) $\forall 1 \le i \le q$, $R_i^{j_1}, R_i^{j_2}, R_{q+i}^{j_3}$ s.t. $\{j_1, j_2, j_3\} = \{1,2,3\}$. □

Lemma 3. *In any optimal solution of* LAPCS *problem on (S_1, P_1) and (S_2, P_2), if $\chi(1, x_k, S_M^1)$ (resp. $\chi(1, \overline{x_k}, S_M^1)$) for a given $1 \le k \le n$ is deleted then, $\forall 1 \le j \le q$, $\chi(1, x_k, C_j^1)$ (resp. $\chi(1, \overline{x_k}, C_j^1)$) is deleted.*

Proof. By construction, $\forall 1 \le k \le n$ only one of $\{x_k, \overline{x_k}\}$ may be conserved between S_M^1 and S_M^2 since $\chi(1, x_k, S_M^1) < \chi(1, \overline{x_k}, S_M^1)$ whereas $\chi(1, \overline{x_k}, S_M^2) < \chi(1, x_k, S_M^2)$. By Lemma 1, at least one symbol incident to any arc is deleted. Therefore, $\forall 1 \le k \le n$ only one of $\{x_k, \overline{x_k}\}$ may be conserved between C_1^1 and C_1^2.

Let us suppose that for a given $1 \le k \le n$, $\chi(1, \overline{x_k}, S_M^1)$ is deleted. According to the proof of Lemma 2, in any optimal solution, $\forall 1 \le k \le n$ exactly

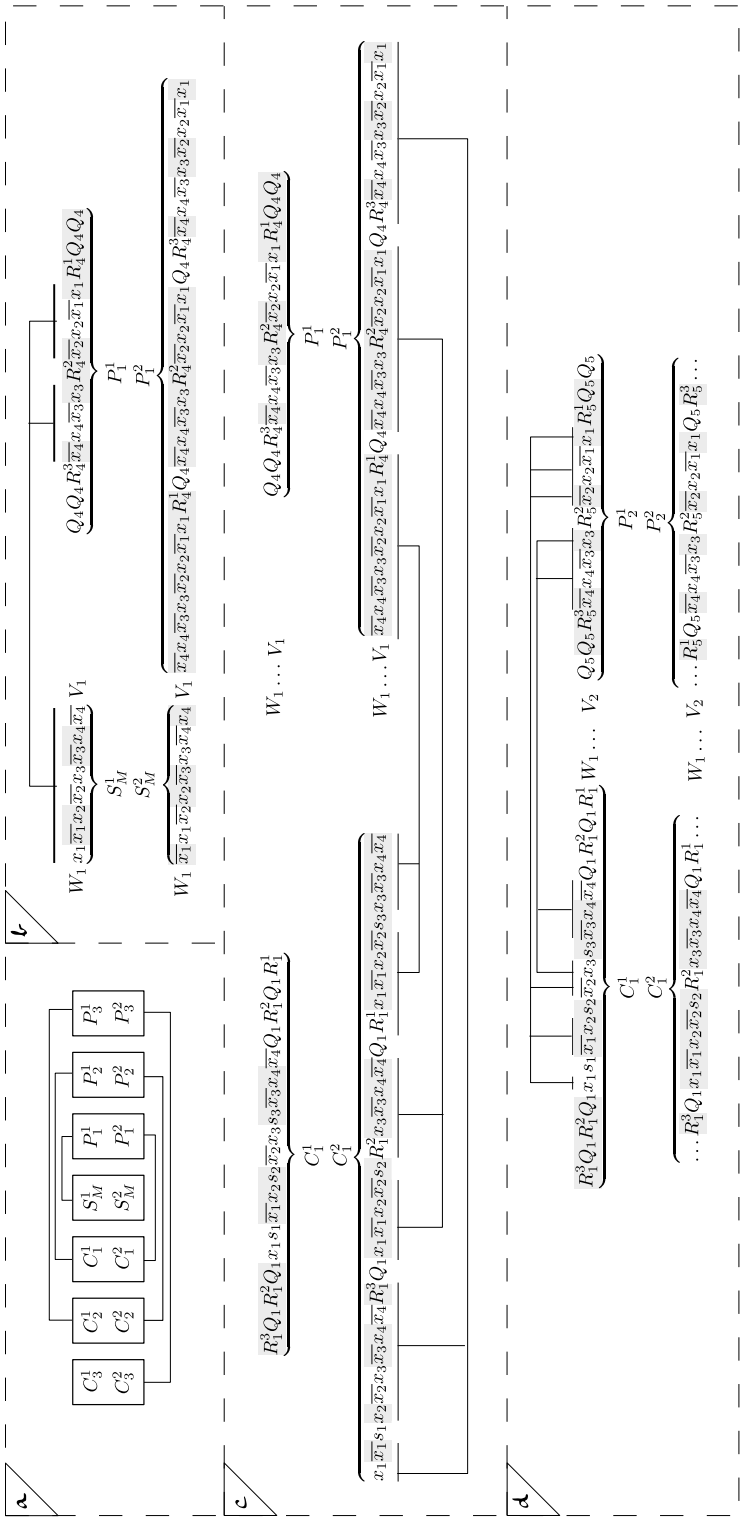

Figure 1: Considering $C_q = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4})$. For readability, all the arcs have not been drawn, consecutive arcs are representing by a unique arc with lines for endpoints. Symbols over a grey background may be deleted to obtain an optimal LAPCS. a) A schematic view of the overall arrangement of the components of the two a.a. sequences. b) Description of S_M^1, S_M^2, P_1^1, P_1^2 and the corresponding arcs in P_1. c) Description of C_1^1, C_1^2, P_1^1, P_1^2 and the corresponding arcs in P_2. d) Description of C_1^1, C_1^2, P_2^1, P_2^2 and the corresponding arcs in P_1.

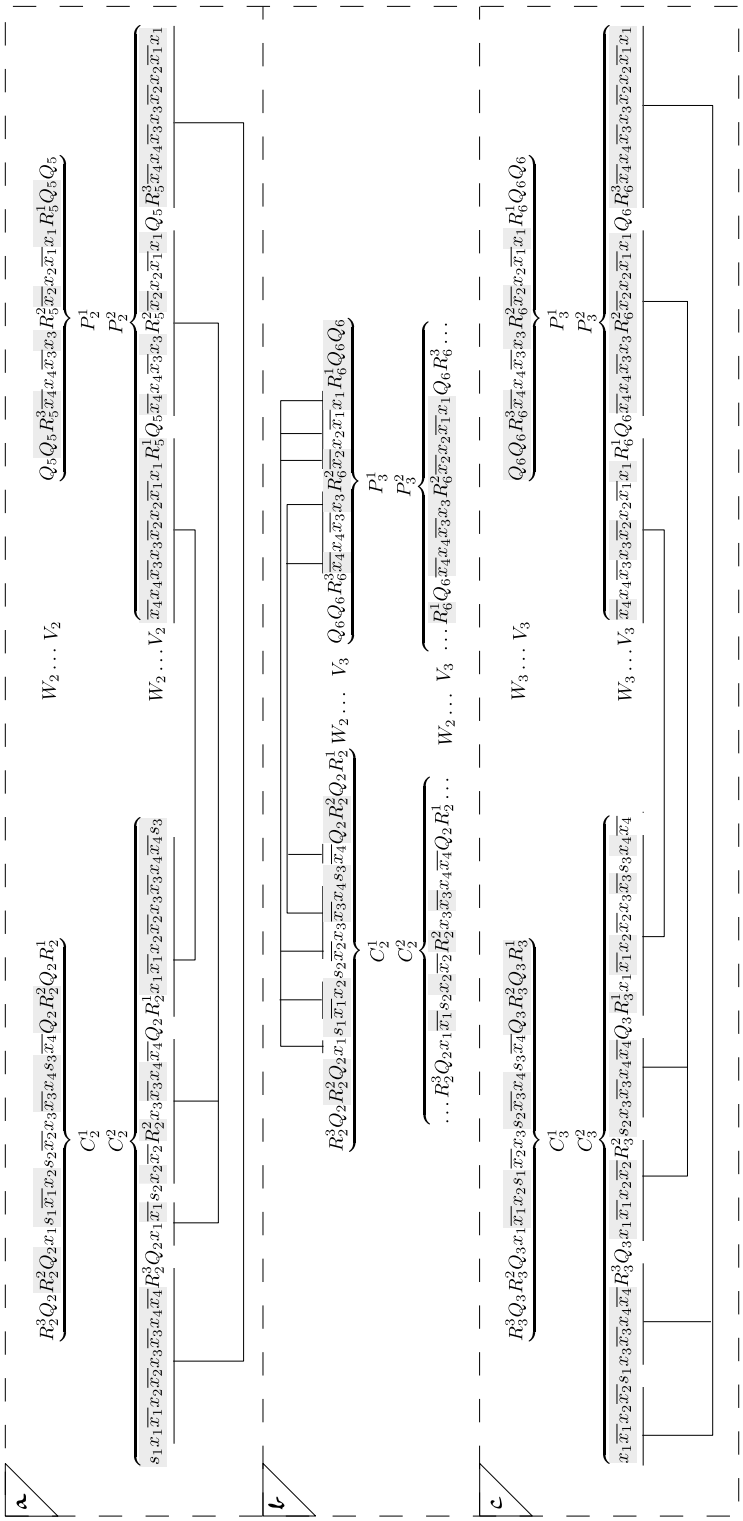

Figure 2: Considering $C_q = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4})$. For readability all the arcs have not been drawn, consecutive arcs are representing by a unique arc with lines for endpoints. Symbols over a grey background may be deleted to obtain an optimal LAPCS. a) Description of C_2^1, C_2^2, P_2^1, P_2^2 and the corresponding arcs in P_2. c) Description of C_2^1, C_2^2, P_3^1, P_3^2 and the corresponding arcs in P_1. d) Description of C_3^1, C_3^2, P_3^1, P_3^2 and the corresponding arcs in P_2.

one of $\{x_k, \overline{x_k}\}$ has to be deleted. Then $\chi(1, x_k, P_1^1)$ is deleted whereas $\chi(1, \overline{x_k}, P_1^1)$ is conserved.

By construction, in P_1^2, since according to the proof of Lemma 2, both occurrences of Q_{q+1} and $R_1^{j_1}, R_1^{j_2}, R_{q+1}^{j_3}$ s.t. $\{j_1, j_2, j_3\} = \{1, 2, 3\}$ have to be conserved, either (1) $\{R_1^1, R_1^2, R_{q+1}^3\}$, (2) $\{R_1^1, R_1^3, R_{q+1}^2\}$ or (3) $\{R_1^2, R_1^3, R_{q+1}^1\}$ are conserved.

Let us first consider that $\{R_1^1, R_1^2, R_{q+1}^3\}$ are conserved. Then one can check that the only solution is to conserve $\chi(2, R_1^2, C_1^1)$ since otherwise at least half of the x_k's would not be conserved. Consequently, the only solution is to conserve, $\forall 1 \leq k \leq n$, the first (resp. last) occurrence of any x_k or $\overline{x_k}$ in C_1^2 (resp. P_1^2) – i.e. the occurrences appearing before $\chi(1, Q_1, C_1^2)$ (resp. after $\chi(2, Q_{q+1}, P_1^2)$). Since by construction, there is an arc between $\chi(1, x_k, C_1^2)$ (resp. $\chi(1, \overline{x_k}, C_1^2)$) and $\chi(3, x_k, P_1^2)$ (resp. $\chi(3, \overline{x_k}, P_1^2)$), in order for $\chi(1, \overline{x_k}, P_1^1)$ to be conserved, one has to conserved $\chi(3, \overline{x_k}, P_1^2)$. Thus, by Lemma 1, $\chi(1, \overline{x_k}, C_1^2)$ has to be deleted and, according to the proof of Lemma 2, $\chi(1, x_k, C_1^2)$ has to be conserved.

Let us now consider that $\{R_1^1, R_1^3, R_{q+1}^2\}$ are conserved. By a similar reasoning, one can check that the only solution is to conserve, $\forall 1 \leq k \leq n$, the second occurrence of any x_k or $\overline{x_k}$ in C_1^2 (resp. P_1^2) – i.e. the occurrences appearing between $\chi(1, Q_1, C_1^2)$ and $\chi(2, Q_1, C_1^2)$ (resp. $\chi(1, Q_{q+1}, P_1^2)$ and $\chi(2, Q_{q+1}, P_1^2)$). Since by construction, there is an arc between $\chi(2, x_k, C_1^2)$ (resp. $\chi(2, \overline{x_k}, C_1^2)$) and $\chi(2, x_k, P_1^2)$ (resp. $\chi(2, \overline{x_k}, P_1^2)$), in order to $\chi(1, \overline{x_k}, P_1^1)$ to be conserved, one has to conserved $\chi(2, \overline{x_k}, P_1^2)$. Thus, by Lemma 1, $\chi(2, \overline{x_k}, C_1^2)$ has to be deleted and, according to the proof of Lemma 2, $\chi(2, x_k, C_1^2)$ has to be conserved.

Finally, let us consider that $\{R_1^2, R_1^3, R_{q+1}^1\}$ are conserved. Once again, by a similar reasoning, one can check that the only solution is to conserve $\chi(1, R_1^2, C_1^1)$ since otherwise at least half of the x_k's would not be conserved. Consequently, the only solution is to conserve, $\forall 1 \leq k \leq n$, the last (resp. first) occurrence of any x_k or $\overline{x_k}$ in C_1^2 (resp. P_1^2) – i.e. the occurrences appearing after $\chi(2, Q_1, C_1^2)$ (resp. before $\chi(1, Q_{q+1}, P_1^2)$). Since by construction, there is an arc between $\chi(3, x_k, C_1^2)$ (resp. $\chi(3, \overline{x_k}, C_1^2)$) and $\chi(1, x_k, P_1^2)$ (resp. $\chi(1, \overline{x_k}, P_1^2)$), in order to $\chi(1, \overline{x_k}, P_1^1)$ to be conserved, one has to conserved $\chi(1, \overline{x_k}, P_1^2)$. Thus, by Lemma 1, $\chi(3, \overline{x_k}, C_1^2)$ has to be deleted and, according to the proof of Lemma 2, $\chi(3, x_k, C_1^2)$ has to be conserved.

Therefore, in the three cases, if for a given $1 \leq k \leq n$, $\chi(1, x_k, S_M^1)$ is conserved then so does $\chi(1, x_k, C_1^1)$. It is easy to see that, by a similar reasoning, if for a given $1 \leq k \leq n$, $\chi(1, \overline{x_k}, S_M^1)$ is conserved then so does $\chi(1, \overline{x_k}, C_1^1)$.

With a similar reasoning, by reccurence, since, $\forall 1 \leq i \leq q, 1 \leq k \leq n$, there is an arc in P_1 between $\chi(1, x_k, C_i^1)$ (resp. $\chi(1, \overline{x_k}, C_i^1)$) and $\chi(1, x_k, P_{i+1}^1)$ (resp. $\chi(1, \overline{x_k}, P_{i+1}^1)$), if $\chi(1, x_k, C_i^1)$ is conserved then $\chi(1, x_k, P_{i+1}^1)$ is deleted. And therefore, with similar arguments, $\chi(1, x_k, C_{i+1}^1)$ is conserved. Once more, it is easy to see that this result still holds if $\chi(1, \overline{x_k}, C_i^1)$ is conserved. □

Theorem 2. *Given an instance of the problem 3SAT with n variables and q clauses, there exists a satisfying truth assignment iff the* LAPCS *of (S_1, P_1) and (S_2, P_2) is of length $k' = 40q(max\{q, n\}^2) + 6qn + 8q + n$.*

Proof. ($\Rightarrow$) An optimal solution for $C_q = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4})$ – i.e. $x_1 = x_3 = true$ and $x_2 = x_4 = false$ – is illustrated in Figures 1 and 2 where any symbol over a grey background have to be deleted. Suppose we have a solution of 3SAT, that is an assignment of each variable of V_n satisfying C_q. Let us first list all the symbols to delete in S_1.

For all $1 \leq k \leq n$, if $x_k = false$ then delete, $\forall 1 \leq j \leq q$, $\{\chi(1, x_k, C_j^1), \chi(1, \overline{x_k}, P_j^1)\}$ and $\chi(1, x_k, S_M^1)$; otherwise delete, $\forall 1 \leq j \leq q$, $\{\chi(1, \overline{x_k}, C_j^1), \chi(1, x_k, P_j^1)\}$ and $\chi(1, \overline{x_k}, S_M^1)$.

For each L_i^j satisfying c_i with the biggest index j with $1 \leq i \leq q$,

if (1) $j = 1$ then delete $\{\chi(1, R_i^3, C_i^1), \chi(1, Q_i, C_i^1), \chi(1, R_i^2, C_i^1), \chi(2, Q_i, C_i^1), \chi(1, s_2, C_i^1), \chi(1, s_3, C_i^1), \chi(1, R_{q+i}^2, P_i^1), \chi(1, R_{q+i}^1, P_i^1), \chi(3, Q_{q+i}, P_i^1), \chi(4, Q_{q+i}, P_i^1)\}$ (cf Figure 1.a);

if (2) $j = 2$ then delete $\{\chi(1, R_i^2, C_i^1), \chi(2, Q_i, C_i^1), \chi(1, s_1, C_i^1), \chi(1, s_3, C_i^1), \chi(3, Q_i, C_i^1), \chi(2, R_i^2, C_i^1), \chi(2, Q_{q+i}, P_i^1), \chi(1, R_{q+i}^3, P_i^1), \chi(1, R_{q+i}^1, P_i^1), \chi(3, Q_{q+i}, P_i^1)\}$ (cf Figure 2.a);

if (3) $j = 3$ then delete $\{\chi(1, s_1, C_i^1), \chi(1, s_2, C_i^1), \chi(3, Q_i, C_i^1), \chi(2, R_i^2, C_i^1), \chi(4, Q_i, C_i^1), \chi(1, R_i^1, C_i^1), \chi(1, Q_{q+i}, P_i^1), \chi(2, Q_{q+i}, P_i^1), \chi(1, R_{q+i}^3, P_i^1), \chi(1, R_{q+i}^2, P_i^1)\}$ (cf Figure 2.c);

Let us now list all the symbols in S_2 to be deleted.

For all $1 \leq k \leq n$, if $x_k = false$ then delete $\chi(1, x_k, S_M^2)$; otherwise delete $\chi(1, \overline{x_k}, S_M^2)$.

For each L_i^j satisfying c_i with the biggest index j with $1 \leq i \leq q$,

if (1) $j = 1$ then delete $\forall 1 \leq k \leq n$ $\{\chi(1, R_i^3, C_i^2), \chi(1, s_2, C_i^2), \chi(2, x_k, C_i^2), \chi(2, \overline{x_k}, C_i^2), \chi(1, s_3, C_i^2), \chi(3, x_k, C_i^2), \chi(3, \overline{x_k}, C_i^2), \chi(1, x_k, P_i^2), \chi(1, \overline{x_k}, P_i^2), \chi(1, R_{q+i}^1, P_i^2), \chi(1, R_{q+i}^2, P_i^2), \chi(2, x_k, P_i^2), \chi(2, \overline{x_k}, P_i^2)\}$. Moreover, if $x_k = false$ with $1 \leq k \leq n$ then delete, $\{\chi(1, x_k, C_i^2), \chi(3, \overline{x_k}, P_i^2)\}$; otherwise delete $\{\chi(1, \overline{x_k}, C_i^2), \chi(3, x_k, P_i^2)\}$ (cf Figure 1.a);

if (2) $j = 2$ then delete $\forall 1 \leq k \leq n$ $\{\chi(1, R_i^2, C_i^2), \chi(1, s_1, C_i^2), \chi(1, x_k, C_i^2), \chi(1, \overline{x_k}, C_i^2), \chi(1, s_3, C_i^2), \chi(3, x_k, C_i^2), \chi(3, \overline{x_k}, C_i^2), \chi(1, x_k, P_i^2), \chi(1, \overline{x_k}, P_i^2), \chi(1, R_{q+i}^1, P_i^2), \chi(1, R_{q+i}^3, P_i^2), \chi(3, x_k, P_i^2), \chi(3, \overline{x_k}, P_i^2)\}$. Moreover, if $x_k = false$ with $1 \leq k \leq n$ then delete, $\{\chi(2, x_k, C_i^2), \chi(2, \overline{x_k}, P_i^2)\}$; otherwise delete $\{\chi(2, \overline{x_k}, C_i^2), \chi(2, x_k, P_i^2)\}$ (cf Figure 2.a);

if (3) $j = 3$ then delete $\forall 1 \leq k \leq n$ $\{\chi(1, R_i^1, C_i^2), \chi(1, s_1, C_i^2), \chi(1, x_k, C_i^2), \chi(1, \overline{x_k}, C_i^2), \chi(1, s_2, C_i^2), \chi(2, x_k, C_i^2), \chi(2, \overline{x_k}, C_i^2), \chi(2, x_k, P_i^2), \chi(2, \overline{x_k}, P_i^2), \chi(1, R_{q+i}^2, P_i^2), \chi(1, R_{q+i}^3, P_i^2), \chi(3, x_k, P_i^2), \chi(3, \overline{x_k}, P_i^2)\}$. Moreover, if $x_k = false$ with $1 \leq k \leq n$ then delete, $\{\chi(3, x_k, C_i^2), \chi(1, \overline{x_k}, P_i^2)\}$; otherwise delete $\{\chi(3, \overline{x_k}, C_i^2), \chi(1, x_k, P_i^2)\}$ (cf Figure 2.c);

By construction, the natural order of the symbols of S_1 and S_2 allows the corresponding set of undeleted symbols to be conserved in a common arc-preserving common subsequence between (S_1, P_1) and (S_2, P_2). Let us now prove that the length of this last is k'. One can easily check that this solution is composed of $\forall 1 \leq k \leq n$, (1) $2q + 1$ occurrences of either x_k or $\overline{x_k}$, (2) $\forall 1 \leq i \leq q$, 2 occurrences of Q_i and Q_{q+i}, (3) $\forall 1 \leq i \leq q$, 1 occurrence of each $\{W_i, V_i\}$ and

either s_1, s_2 or s_3 and (4) $\forall 1 \leq i \leq q$, $R_i^{j_1}, R_i^{j_2}, R_{q+i}^{j_3}$ s.t. $\{j_1, j_2, j_3\} = \{1, 2, 3\}$. Thus, the length of the solution is $40q(max\{q, n\}^2) + 6qn + 8q + n$.

($\Leftarrow$) Suppose we have an optimal solution – i.e. a set of symbols S_d to delete – for LAPCS of (S_1, P_1) and (S_2, P_2). Let us define the truth assignment of V_n s.t., $\forall 1 \leq i \leq q$, if $\chi(1, s_j, C_i^1) \notin S_d$ then L_i^j is true. Let us prove that it is a solution of 3SAT.

By construction, if $L_i^j = x_k$ (resp. $\overline{x_k}$) then in C_i^1, s_j appears between x_k and $\overline{x_k}$ whereas in C_j^2 it appears after $\overline{x_k}$ (resp. before x_k). Thus, if $\chi(1, s_j, C_i^1)$ is not deleted then $\overline{x_k}$ (resp. x_k) in C_i^1 is deleted if $L_i^j = x_k$ (resp. $\overline{x_k}$). Consequently, according to the proof of Lemma 3, if $\chi(1, s_j, C_i^1)$ is not deleted then $\overline{x_k}$ (resp. x_k) in all $C_{i'}^1$, with $1 \leq i' \leq q$ is deleted if $L_i^j = x_k$ (resp. $\overline{x_k}$). Therefore, we can ensure that one cannot obtain L_i^j and $L_{i'}^{j'}$ being true whereas $L_i^j = \overline{L_{i'}^{j'}}$ (that is a variable cannot be simultaneously true and false). By Lemma 2, we can ensure that for any $1 \leq i \leq q$ exactly one of $\{s_1, s_2, s_3\}$ is conserved in C_i^1. Therefore, for any clause c_i at least one of its literal is set to true. This ensures that our solution is a solution of 3SAT. □

4 Future work

From a computational biology point of view, especially for comparing stems, one may, however, be mostly interested in the case k (length of the common subsequence searched) might not be assumed to small compared to n. A first approach is provided in (Alber et al. 2004) where it is proved that, given two sequences of length at most n and nested arc structure, an arc-preserving common subsequence can be determined (if it exists) in $O(3.31^{k_1+k_2}\, n)$ time; obtained by deleting (together with corresponding arcs) k_1 letters from the first and k_2 letters from the second sequence. Improving the running time of the parameterization in case of stem arc structures appears to be a promising line of research.

References

Alber, J., Gramm, J., Guo, J. & Niedermeier, R. (2004), 'Computing the similarity of two sequences with nested arc annotations', *Theoretical Computer Science* **312**(2-3), 337–358.

Blin, G., Denise, A., Dulucq, S., Herrbach, C. & Touzet, H. (2008), 'Alignment of RNA structures', *IEEE/ACM Transactions on Computational Biology and Bioinformatics* . To appear.

Blin, G., Fertin, G., Herry, G. & Vialette, S. (2007), Comparing rna structures: towards an intermediate model between the edit and the lapcs problems, *in* M.-F. Sagot & M. E. Telles Walter, eds, '1st Brazilian Symposium on Bioinformatics (BSB'07)', Vol. 4643 of *Lecture Notes in Bioinformatics*, Springer-Verlag, Angra dos Reis, Brazil, pp. 101–112.

Blin, G., Fertin, G., Rusu, I. & Sinoquet, C. (2007), Extending the hardness of rna secondary structu, *in* B. Chen, M. Paterson & G. Zhang, eds, '1st intErnational Symposium on Combinatorics, Algorithms, Probabilistic and Experimental methodologies (ESCAPE'07)', Vol. 4614 of *LNCS*, Springer-Verlag, Hangzhou, China, pp. 140–151.

Evans, P. (1999), Algorithms and Complexity for Annotated Sequences Analysis, PhD thesis, University of Victoria.

Garey, M. & Johnson, D. (1979), *Computers and Intractability: a guide to the theory of NP-completeness*, W.H. Freeman, San Franciso.

Gramm, J., Guo, J. & Niedermeier, R. (2006), 'Pattern matching for arc-annotated sequences', *ACM Transactions on Algorithms* **2**(1), 44–65. To appear.

Guignon, V., Chauve, C. & Hamel, S. (2005), An edit distance between rna stem-loops, *in* M. P. Consens & G. Navarro, eds, '12th International Conference SPIRE', Vol. 3772 of *LNCS*, pp. 335–347.

Jiang, T., Lin, G., Ma, B. & Zhang, K. (2002), 'A general edit distance between RNA structures', *Journal of Computational Biology* **9(2)**, 371–388.

Jiang, T., Lin, G., Ma, B. & Zhang, K. (2004), 'The longest common subsequence problem for arc-annotated sequences', *Journal of Dicrete Algorithms* pp. 257–270.

Lin, G., Chen, Z.-Z., jiang, T. & Wen, J. (2002), 'The longest common subsequence problem for sequences with nested arc annotations', *Journal of Computer and System Sciences* **65**, 465–480.

Zhang, K. & Shasha, D. (1989), 'Simple fast algorithms for the editing distance between trees and related problems', *SIAM journal of computing* **18**(6), 1245–1262.