



HAL
open science

Comparing RNA structures with biologically relevant operations cannot be done without strong combinatorial restrictions

Guillaume Blin, Sylvie Hamel, Stéphane Vialette

► **To cite this version:**

Guillaume Blin, Sylvie Hamel, Stéphane Vialette. Comparing RNA structures with biologically relevant operations cannot be done without strong combinatorial restrictions. 4th Workshop on Algorithms and Computation (WALCOM'10), Feb 2010, Dhaka, Bangladesh, Bangladesh. pp.149-160. hal-00620327

HAL Id: hal-00620327

<https://hal.science/hal-00620327v1>

Submitted on 30 Sep 2011

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 with biologically relevant operations cannot be done without strong combinatorial restrictions

Guillaume Blin¹, Sylvie Hamel², and Stéphane Vialette¹

¹ Université Paris-Est, LIGM - UMR CNRS 8049, France
`{gblin,vialette}@univ-mlv.fr`

² DIRO - Université de Montréal - QC - Canada
`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 [2] 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 [5]. The introduced model uses edit operations that involve either single letters or pairs of letters (never considered separately) and is solvable in polynomial-time [12].

To account for other possible RNA structural evolutionary events, new edit operations, allowing to consider either simultaneously or separately letters of a pair were introduced in [9]; 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 [8], 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.

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 [5], 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.

The problem of computing an edit distance between two arc-annotated sequences was introduced in [5] 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 [12].

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 [9] at the cost of computational tractability. Indeed, it has been shown in [4] 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 [8, 2]. Of particular importance, in [8], 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 [9]. This algorithm is based on a decomposition in stem-loop-like substructures that are pairwise compared and used to compare complete RNA secondary structures. As mentioned in [8], 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 is actually 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 [5] (see [8] 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 [2] for more details) we focus in this article on the LONGEST ARC-PRESERVING COMMON SUBSEQUENCE (LAPCS for short) [5, 10, 11] and the general edit distance (EDIT for short) [9, 3]. Indeed, as shown in [2], 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 [9]. 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 [9] that the LAPCS problem is **NP**-hard even for NESTED structures, *i.e.*, LAPCS(NESTED, NESTED). Still focussing on NESTED structures, Alber *et al.* [1] 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.

While LAPCS(STEM, STEM) might appear at first sight as yet another refinement of LAPCS, we emphasize here that our hardness result implies hardness of EDIT(STEM, STEM) [2]. This latter problem has been shown to be of particular interest to compare complete RNA secondary structures [8]. Another consequence of our result is that further research on comparing RNA structures with a full set of edit operations aiming at polynomial-time solutions will have to introduce strong combinatorial restrictions.

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 [8]. 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 **NP**-hardness, we define a reduction from the **NP**-complete 3SAT problem [6] which is defined as follows: Given a collection $C_q = \{c_1, c_2, \dots, c_q\}$ 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 $V_n = \{x_1, x_2, \dots, x_n\}$, 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. $C_q = \{c_1, c_2, \dots, c_q\}$ and $V_n = \{x_1, x_2, \dots, x_n\}$. For convenience, let L_i^j denote the j^{th} literal of the i^{th} 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 i^{th} 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 $S_1 = C_q^1 W_q C_{q-1}^1 \dots C_2^1 W_2 C_1^1 W_1 S_M^1 V_1 P_1^1 V_2 P_2^1 \dots P_{q-1}^1 V_q P_q^1$ and $S_2 = C_q^2 W_q C_{q-1}^2 \dots C_2^2 W_2 C_1^2 W_1 S_M^2 V_1 P_1^2 V_2 P_2^2 \dots P_{q-1}^2 V_q P_q^2$ s.t. for all $1 \leq i \leq q, 1 \leq k \leq n$,

$$\begin{aligned} & - C_i^1 = R_i^3 Q_i R_i^2 Q_i X_1^1 X_2^1 \dots X_n^1 Q_i R_i^2 Q_i R_i^1 \text{ with } X_k^1 = x_k s_j \bar{x}_k \text{ if } x_k = L_i^j \text{ or } \\ & \quad \bar{x}_k = L_i^j; X_k^1 = x_k \bar{x}_k \text{ otherwise;} \\ & - P_i^1 = Q_{q+i} Q_{q+i} R_{q+i}^3 X_n^1 \dots X_{\frac{n}{2}+1}^1 R_{q+i}^2 X_{\frac{n}{2}}^1 \dots X_1^1 R_{q+i}^1 Q_{q+i} Q_{q+i} \text{ s.t. } X_k^1 = \bar{x}_k x_k; \\ & - C_i^2 = X_1^2 \dots X_n^2 R_i^3 Q_i X_1^2 \dots X_{\frac{n}{2}}^2 R_i^2 X_{\frac{n}{2}+1}^2 \dots X_1^2 Q_i R_i^1 X_1^2 \dots X_n^2 \text{ s.t. } \forall 1 \leq j \leq 3, \\ & \quad \chi(j, X_k^2, C_i^2) = x_k \bar{x}_k s_j \text{ (resp. } s_j x_k \bar{x}_k) \text{ if } x_k = L_i^j \text{ (resp. } \bar{x}_k = L_i^j); \\ & \quad \chi(j, X_k^2, C_i^2) = x_k \bar{x}_k \text{ otherwise;} \\ & - P_i^2 = X_n^2 \dots X_1^2 R_{q+i}^1 Q_{q+i} X_n^2 \dots X_{\frac{n}{2}+1}^2 R_{q+i}^2 X_{\frac{n}{2}}^2 \dots X_1^2 Q_{q+i} R_{q+i}^3 X_n^2 \dots X_1^2 \text{ with} \\ & \quad X_k^2 = \bar{x}_k x_k. \end{aligned}$$

Moreover, let $S_M^1 = x_1 \bar{x}_1 x_2 \bar{x}_2 \dots x_n \bar{x}_n$ and $S_M^2 = \bar{x}_1 x_1 \bar{x}_2 x_2 \dots \bar{x}_n x_n$. Notice that, by construction, there is only one occurrence of each $\{s_1, s_2, s_3\}$ in C_i^2 .

For all $1 \leq i \leq q$, let Q_i (resp. Q_{q+i}) be a segment of $n+1$ symbols y_i (resp. y_{q+i}). Moreover, for all $1 \leq i \leq q$, 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 $1 \leq i \leq q-1$, (1) add an arc in P_1 between $\chi(1, x_k, C_i^1)$ (resp. $\chi(1, \bar{x}_k, C_i^1)$) and $\chi(1, x_k, P_{i+1}^1)$ (resp. $\chi(1, \bar{x}_k, P_{i+1}^1)$), $\forall 1 \leq k \leq n$ (see Figure 1.d and 2.b); (2) add an arc in P_2 between $\chi(j, x_k, C_i^2)$ (resp. $\chi(j, \bar{x}_k, C_i^2)$) and $\chi((4-j), x_k, P_i^2)$ (resp. $\chi((4-j), \bar{x}_k, P_i^2)$), $\forall 1 \leq k \leq n$ (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 $\chi(1, R_{q+i}^j, P_i^2)$, $\forall 1 \leq j \leq 3$ (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 $1 \leq i \leq q$, 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 $1 \leq j \leq q$, 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 $q[12n + 18 + 40(\max\{q, n\}^2)] + 2n - (20(\max\{q, n\}^2)) \geq 40q(\max\{q, n\}^2) + 6qn + 8q + n$. This can be reduced to $6qn + 10q - 20(\max\{q, n\}^2) + n \geq 0$. But, one can easily check that for any $n \geq 3$ (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) $\forall 1 \leq i \leq n$, $2q + 1$ occurrences of x_i (resp. \bar{x}_i); (2) $\forall 1 \leq i \leq q$, 4 occurrences of Q_i (resp. Q_{q+i}); (3) $\forall 1 \leq i \leq q$, 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) $\forall 1 \leq i \leq q$, 2 occurrences of R_i^2 .

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

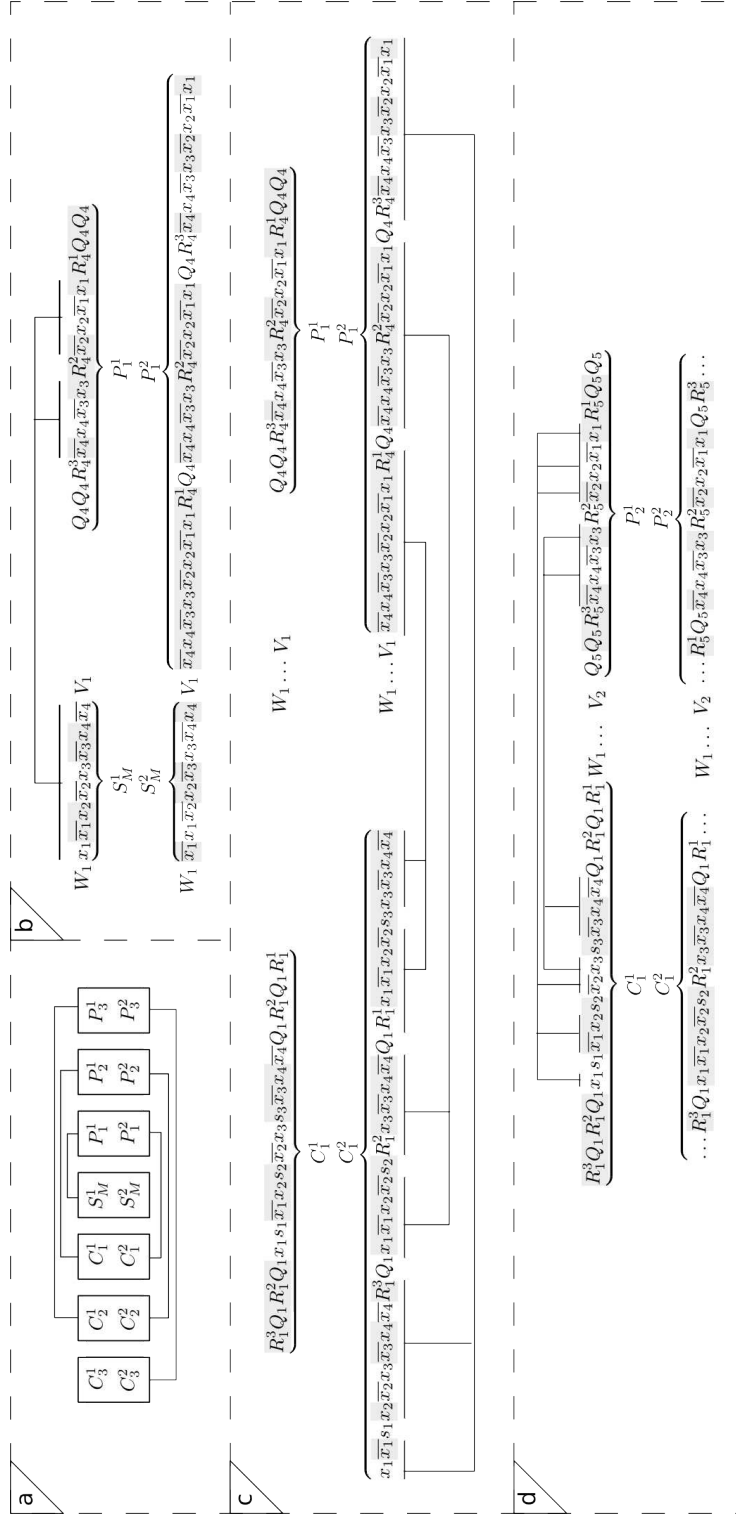


Fig. 1. Considering $C_q = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$. For readability, all the arcs have not been drawn, consecutive arcs are representing by a unique arc with lines for endpoints. Symbols over a grey background 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 .

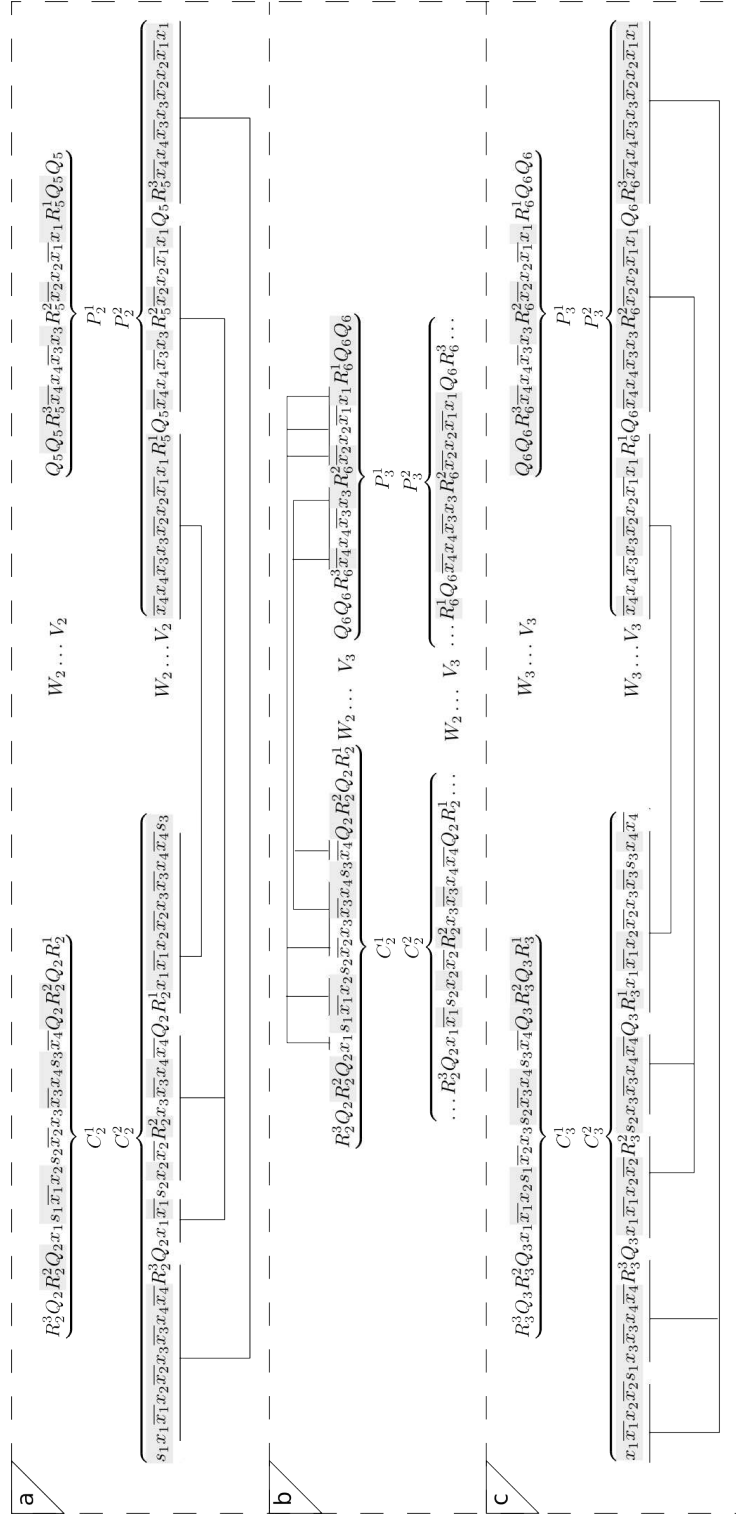


Fig. 2. Considering $C_q = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$. For readability all the arcs have not been drawn, consecutive arcs are representing by a unique arc with lines for endpoints. Symbols over a grey background 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 .

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

Finally, in any optimal solution, only half of the occurrences of $\{x_i, \bar{x}_i\}$ 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) $\forall 1 \leq i \leq n$, $2q+1$ occurrences of either x_i or \bar{x}_i , (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. $\{j_1, j_2, j_3\} = \{1, 2, 3\}$.

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. $\chi(1, \bar{x}_k, S_M^1)$) for a given $1 \leq k \leq n$ is deleted then, $\forall 1 \leq j \leq q$, $\chi(1, x_k, C_j^1)$ (resp. $\chi(1, \bar{x}_k, C_j^1)$) is deleted.*

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

Let us suppose that for a given $1 \leq k \leq n$, $\chi(1, \bar{x}_k, S_M^1)$ is deleted. According to the proof of Lemma 2, in any optimal solution, $\forall 1 \leq k \leq n$ exactly one of $\{x_k, \bar{x}_k\}$ has to be deleted. Then $\chi(1, x_k, P_1^1)$ is deleted whereas $\chi(1, \bar{x}_k, P_1^1)$ 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 \bar{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. $\chi(1, \bar{x}_k, C_1^2)$) and $\chi(3, x_k, P_1^2)$ (resp. $\chi(3, \bar{x}_k, P_1^2)$), in order for $\chi(1, \bar{x}_k, P_1^1)$ to be conserved, one has to conserve $\chi(3, \bar{x}_k, P_1^2)$. Thus, by Lemma 1, $\chi(1, \bar{x}_k, C_1^2)$ 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 \bar{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. $\chi(2, \bar{x}_k, C_1^2)$) and $\chi(2, x_k, P_1^2)$ (resp. $\chi(2, \bar{x}_k, P_1^2)$), in order to $\chi(1, \bar{x}_k, P_1^1)$ to be conserved, one has to conserve $\chi(2, \bar{x}_k, P_1^2)$. Thus, by Lemma 1, $\chi(2, \bar{x}_k, C_1^2)$ 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 \bar{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. $\chi(3, \bar{x}_k, C_1^2)$) and $\chi(1, x_k, P_1^2)$ (resp. $\chi(1, \bar{x}_k, P_1^2)$), in order to $\chi(1, \bar{x}_k, P_1^1)$ to be conserved, one has to conserve $\chi(1, \bar{x}_k, P_1^2)$. Thus, by Lemma 1, $\chi(3, \bar{x}_k, C_1^2)$ 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 $1 \leq k \leq n$, $\chi(1, \bar{x}_k, S_M^1)$ is conserved then so does $\chi(1, \bar{x}_k, C_1^1)$.

With a similar reasoning, by recurrence, 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. $\chi(1, \bar{x}_k, C_i^1)$) and $\chi(1, x_k, P_{i+1}^1)$ (resp. $\chi(1, \bar{x}_k, P_{i+1}^1)$), 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 $\chi(1, \bar{x}_k, C_i^1)$ 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 $C_q = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$ – i.e. $x_1 = x_3 = \text{true}$ and $x_2 = x_4 = \text{false}$ – 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 $x_k = \text{false}$ then delete, $\forall 1 \leq j \leq q$, $\{\chi(1, x_k, C_j^1), \chi(1, \bar{x}_k, P_j^1)\}$ and $\chi(1, x_k, S_M^1)$; otherwise delete, $\forall 1 \leq j \leq q$, $\{\chi(1, \bar{x}_k, C_j^1), \chi(1, x_k, P_j^1)\}$ and $\chi(1, \bar{x}_k, S_M^1)$.

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_i^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.

Lemma 4. $\text{LAPCS}(\text{STEM}, \text{STEM})$ is solvable in $O(2^{k-1} |\Sigma|^k kn)$

Proof. We use a straightforward brute-force algorithm for arc-annotated sequences [1]: (i) generate all possible sequences of length k with all possible STEM 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 $O(nk)$ time whether or not this sequence occurs as an arc-preserving common subsequence [7]. It is easily seen that the above algorithm reduces to $O(2^{k-1} |\Sigma|^k km)$ time for $\text{LAPCS}(\text{STEM}, \text{STEM})$. Indeed, there exist $|\Sigma|^k$ sequences of length k and hence, for a given sequence of length k , there exist $\binom{k}{2i}$ 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 .

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 too small compared to n . A first approach is provided in [1] 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

1. J. Alber, J. Gramm, J. Guo, and R. Niedermeier. Computing the similarity of two sequences with nested arc annotations. *Theoretical Computer Science*, 312(2-3):337–358, 2004.
2. G. Blin, A. Denise, S. Dulucq, C. Herrbach, and H. Touzet. Alignment of RNA structures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2008. To appear.
3. G. Blin, G. Fertin, G. Herry, and S. Vialette. Comparing rna structures: towards an intermediate model between the edit and the lapcs problems. In Marie-France Sagot and Maria Emilia Telles Walter, editors, *1st Brazilian Symposium on Bioinformatics (BSB'07)*, volume 4643 of *Lecture Notes in Bioinformatics*, pages 101–112, Angra dos Reis, Brazil, August 2007. Springer-Verlag.

4. Guillaume Blin, Guillaume Fertin, Irena Rusu, and Christine Sinoquet. Extending the hardness of rna secondary structure. In B. Chen, M. Paterson, and G. Zhang, editors, *1st international Symposium on Combinatorics, Algorithms, Probabilistic and Experimental methodologies (ESCAPE'07)*, volume 4614 of *LNCS*, pages 140–151, Hangzhou, China, April 2007. Springer-Verlag.
5. P. Evans. *Algorithms and Complexity for Annotated Sequences Analysis*. PhD thesis, University of Victoria, 1999.
6. M.R. Garey and D.S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. W.H. Freeman, San Francisco, 1979.
7. J. Gramm, J. Guo, and R. Niedermeier. Pattern matching for arc-annotated sequences. *ACM Transactions on Algorithms*, 2(1):44–65, 2006. To appear.
8. V. Guignon, C. Chauve, and S. Hamel. An edit distance between rna stem-loops. In Mariano P. Consens and Gonzalo Navarro, editors, *12th International Conference SPIRE*, volume 3772 of *LNCS*, pages 335–347, 2005.
9. T. Jiang, G. Lin, B. Ma, and K. Zhang. A general edit distance between RNA structures. *Journal of Computational Biology*, 9(2):371–388, 2002.
10. T. Jiang, G. Lin, B. Ma, and K. Zhang. The longest common subsequence problem for arc-annotated sequences. *Journal of Discrete Algorithms*, pages 257–270, 2004.
11. G. Lin, Z.-Z. Chen, T. jiang, and J. Wen. The longest common subsequence problem for sequences with nested arc annotations. *Journal of Computer and System Sciences*, 65:465–480, 2002.
12. K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal of computing*, 18(6):1245–1262, 1989.