Pattern Matching in Arc-Annotated Sequences: New Results for the APS Problem

Guillaume Blin[1], Guillaume Fertin[1], Romeo Rizzi[2], and Stéphane Vialette[3]

[1] LINA FRE CNRS 2729
Université de Nantes, 2 rue de la Houssinière
BP 92208 44322 Nantes Cedex 3 - FRANCE
{blin,fertin}@lina.univ-nantes.fr
[2] Università degli Studi di Trento
Facoltà di Scienze- Dipartimento di Informatica e Telecomunicazioni
Via Sommarive, 14 - I38050 Povo - Trento (TN) - ITALY
Romeo.Rizzi@unitn.it
[3] LRI UMR CNRS 8623
Faculté des Sciences d'Orsay, Université Paris-Sud
Bât 490, 91405 Orsay Cedex - FRANCE
vialette@lri.fr

Abstract. In molecular biology, RNA structure comparison and motif search are of great interest for solving major problems such as phylogeny reconstruction, prediction of molecule folding and identification of common functions. RNA structures can be represented by arc-annotated sequences (base sequence along with arc annotations), and this paper mainly focus on the so-called *arc-preserving subsequence* (APS) problem where, given two arc-annotated sequences (S, P) and (T, Q), we are asking whether (T, Q) can be obtained from (S, P) by deleting some of its bases (together with their incident arcs, if any). In previous studies, this problem has been naturally divided into subproblems reflecting the complexity of the arc structures. We show that APS(Crossing, Plain) is **NP**-complete, thereby answering an open problem posed in [11]. Furthermore, to get more insight into where the actual border between the polynomial and the **NP**-completeness cases lies, we refine the classical subproblems of the APS problem in much the same way as in [18]. We then discuss how previous known results help completing the new complexity table and end this paper by giving some new positive results, namely showing that APS($\{\emptyset\}$, $\emptyset$), APS($\{\emptyset\}$, $\{\frown\}$) and APS($\{\emptyset\}$,$\{\emptyset\}$) are polynomial time solvable.

Keywords: RNA structures, Arc-Preserving Subsequence problem, computational complexity.

1 Introduction

At a molecular state, the understanding of biological mechanisms is subordinated to the discovery and the study of RNA functions. Indeed, it is established that the conformation of a single-stranded RNA molecule (a linear sequence composed of ribonucleotides A, U, C and G, also called primary structure) partly determines the function of the molecule. This conformation results from the folding process due to local pairings between complementary bases ($A-U$ and $C-G$, connected by a hydrogen bond). The secondary structure of an RNA (a simplification of the complex 3 dimensional folding of the sequence) is the collection of folding patterns (stem, hairpin loop, bulge loop, internal loop, branch loop and pseudo-knot) that occur in it.

RNA secondary structure comparison has become a crucial problem in many contexts, such as:

- identification of highly conserved structures during evolution, non detectable in the primary sequence which is often slightly preserved. These structures suggest a significant common function for the studied RNA molecules [16, 17, 13, 8],

- RNA classification of various species (phylogeny)[4, 3, 20],
- RNA folding prediction by considering a set of already known secondary structures [23, 14],
- identification of a consensus structure and consequently of a common role for molecules [21, 5].

At a theoretical level, the RNA structure is often modeled as an *arc-annotated sequence*, that is a pair (S, P) where S is the sequence of ribonucleotides and P represents the hydrogen bonds between pairs of elements of S. Different pattern matching and motif search problems have been investigated in the context of arc-annotated sequences among which we can mention the APS, EDIT DISTANCE, AST and LAPCS problems (see for instance [6, 15, 12, 11, 2]).

In this paper, we focus on the *arc-preserving subsequence* (APS) problem: given two arc-annotated sequences (S, P) and (T, Q), this problem asks whether (T, Q) can be exactly obtained from (S, P) by deleting some of its bases together with their incident arcs, if any. This problem is commonly encountered when one is searching for a given RNA pattern in an RNA database [12]. Moreover, from a theoretical point of view, the APS problem can be seen as a restricted version of the LAPCS problem, and hence has applications in the structural comparison of RNA and protein sequences [6, 10, 22]. The APS problem has been extensively studied in the past few years [11, 12, 6]. Of course, different restrictions on arc-annotation alter the computational complexity of the APS problem, and hence this problem has been naturally divided into subproblems reflecting the complexity of the arc structure of both (S, P) and (T, Q): PLAIN, CHAIN, NESTED, CROSSING or UNLIMITED (see Section 2 for details). All of them but one have been classified as to whether they are polynomial time solvable or **NP**-hard. The problem of the existence of a polynomial time algorithm for the APS(CROSSING,PLAIN) problem was mentioned in [11] as the last open problem in the context of arc-preserving subsequences (cf. Table 1). Unfortunately, as we shall prove in Section 3, the APS(CROSSING,PLAIN) problem is **NP**-complete.

In analyzing the complexity of a problem, we are often trying to define the precise boundary between the polynomial and the **NP**-completeness cases. Therefore, as another step towards establishing the precise complexity landscape of the APS problem, we consider that it would be of great interest to subdivide even more the possible cases, that is to refine the classical complexity levels of the APS problem, for determining more precisely what makes the problem hard. For that purpose, we use the framework introduced by Vialette [18] in the context of 2-intervals (a simple abstract structure for modelling RNA secondary structures). As a consequence, the number of complexity levels rises from 4 to 9, and all the entries of this new complexity table need to be filled. However, previous known results concerning the APS problem, along with our **NP**-completeness proof for the APS(CROSSING, PLAIN) problem allows us to fill most of the entries of this new table. We are left with 8 open problems and we answer 3 of them in this paper.

The paper is organized as follows: in Section 2, we give notations and definitions concerning the APS problem. In Section 3, we show that APS(CROSSING,PLAIN) is **NP**-complete. In Section 4, we introduce and explain the new refinements of the complexity levels we are going to study, and we give first straightforward results, leaving 8 unsolved problems. Finally, in Section 5, we show that 3 of those 8 problems are polynomial time solvable.

2 Preliminaries

An RNA structure is commonly represented by an arc-annotated sequence (S, P) where S is the sequence of ribonucleotides (or bases) and P is a set of arcs connecting pairs of bases in S. Let (S, P) and (T, Q) be two arc-annotated sequences such that $|S| \geq |T|$ (in the following, $n = |S|$ and $m = |T|$). The APS problem asks whether (T, Q) can be exactly obtained from (S, P) by deleting some of its bases together with their incident arcs, if any.

Since the general problem is intractable [6], the arc structure must be restricted. Evans [6] proposed four possible restrictions on P (resp. Q) which were largely reused in the subsequent literature:

1. there is no base incident to more than one arc,
2. there are no arcs crossing,
3. there is no arc contained in another,
4. there is no arc.

These restriction are used progressively and inclusively to produce five different levels of allowed arc structure:

- UNLIMITED - the general problem with no restrictions
- CROSSING - restriction 1
- NESTED - restrictions 1 and 2
- CHAIN - restrictions 1, 2 and 3
- PLAIN - restriction 4

Guo proved in [12] that the APS(CROSSING, CHAIN) problem is **NP**-complete. Guo et al. observed in [11] that the **NP**-completeness of the APS(CROSSING, CROSSING) and APS(UNLIMITED, PLAIN) easily follows from results of Evans [6] concerning the LAPCS problem. Furthermore, they gave a $O(nm)$ time for the APS(NESTED, NESTED) problem. This algorithm can be applied to easier problems such as APS(NESTED, CHAIN), APS(NESTED, PLAIN), APS(CHAIN, CHAIN) and APS(CHAIN,PLAIN). Finally, Guo et al. mentioned in [11] that APS(CHAIN, PLAIN) can be solved in $O(n + m)$ time. Observe that the UNLIMITED level has no restrictions, and hence is of limited interest in our study. Consequently, from now on we will not be concerned anymore with that level. Until now, the question of the existence of an exact polynomial algorithm for the problem APS(CROSSING, PLAIN) remained open. We will first show in the next section that the problem APS(CROSSING,PLAIN) is **NP**-complete.

APS					
	UNLIMITED	CROSSING	NESTED	CHAIN	PLAIN
UNLIMITED	**NP**-complete [6]				
CROSSING		**NP**-complete [6]	**NP**-complete [12]		**NP**-complete $\star$
NESTED			$O(nm)$[11]		
CHAIN				$O(nm)$ [11]	$O(n+m)$ [11]

Table 1. APS problem complexity where $n = |S|$ and $m = |T|$. $\star$ contribution of this article.

3 APS(CROSSING,PLAIN) is NP-complete

We show in this section that APS(CROSSING, PLAIN) is **NP**-complete, thereby answering on open problem posed in [11]. Note that the present section answers the last open problem concerning the computational complexity of the APS problem with respect to classical complexity levels, *i.e.*, PLAIN, CHAIN, NESTED and CROSSING (cf. Table 1).

We provide a polynomial time reduction from the EXACT 3-CNF-SAT problem: Given a set $\mathcal{V}_n$ of n variables and a set $\mathcal{C}q$ of q clauses (each composed of three literals) over $\mathcal{V}_n$, the problem asks to find a truth assignment for $\mathcal{V}_n$ that satisfies all clauses of $\mathcal{C}_q$. It is well-known that the EXACT 3-CNF-SAT problem is **NP**-complete [9]. For the sake of clarity, we now state formally the APS(CROSSING, PLAIN)) problem.

APS(CROSSING,PLAIN)
INSTANCE: *Two arc-annotated sequences (S, P) and (T, Q) such that $|S| \geq |T|$ and P (resp. Q) is a set of arcs whose structure is of type* CROSSING *(resp. of type* PLAIN*).*
QUESTION: *Can (T, Q) be obtained from (S, P) by deleting in (S, P) some of its bases together with their incident arcs, if any?*

It is easily seen that APS(CROSSING,PLAIN) is in **NP**. The remainder of the section is devoted to proving that it is also **NP**-hard. Let $\mathcal{V}_n = \{x_1, x_2, ...x_n\}$ be a finite set of n variables and $\mathcal{C}_q = \{c_1, c_2, \ldots, c_q\}$ a collection of q clauses. Let us first detail the construction of the sequences S and T. They are defined as follows:

$$S = S^s_{x_1} A\ S^s_{\overline{x_1}}\ S^s_{x_2} A\ S^s_{\overline{x_2}} \ldots S^s_{x_n} A\ S^s_{\overline{x_n}}\ S_{c_1}\ S_{c_2} \ldots S_{c_q}\ S^e_{x_1}\ S^e_{x_2} \ldots S^e_{x_n}$$
$$T = T^s_{x_1}\ T^s_{x_2} \ldots T^s_{x_n}\ T_{c_1}\ T_{c_2} \ldots T_{c_q}\ T^e_{x_1}\ T^e_{x_2} \ldots T^e_{x_n}$$

We now detail the subsequences that compose S and T. Let γ_m (resp. $\gamma_{\overline{m}}$) be the number of occurrences of literal x_m (resp. $\overline{x_m}$) in $\mathcal{C}_q$. For each variable $x_m \in \mathcal{V}_n$, $1 \leq m \leq n$, we construct words $S^s_{x_m} = AC^k$, $S^s_{\overline{x_m}} = C^k A$ and $T^s_{x_m} = AC^k A$ where C^k represents a word of $k = \max(\gamma_m, \gamma_{\overline{m}})$ consecutive bases C. For each clause c_i of $\mathcal{C}_q$, $1 \leq i \leq q$, we construct words $S_{c_i} = UGGGA$ and $T_{c_i} = UGA$. Finally, for each variable $x_m \in \mathcal{V}_n$, $1 \leq m \leq n$, we construct words $S^e_{x_m} = UUA$ and $T^e_{x_m} = UA$.

Having disposed of the two sequences, we now turn to defining the corresponding two arc structures (see Figure 1). In the following, $\mathcal{S}eq[i]$ will denote the i^{th} base of a sequence $\mathcal{S}eq$ and, for any $1 \leq m \leq n$, $l_m = |S^s_{x_m}|$ and $l_{\overline{m}} = |S^s_{\overline{x_m}}|$. For all $1 \leq m \leq n$, we create the two following arcs: $(S^s_{x_m}[1], S^e_{x_m}[1])$ and $(S^s_{\overline{x_m}}[l_{\overline{m}}], S^e_{x_m}[2])$. For each clause c_i of $\mathcal{C}_q$, $1 \leq i \leq q$, and for each $1 \leq m \leq n$, if a literal of x_m (resp. $\overline{x_m}$) is in c_i, then we create an arc between any free (i.e. not already incident to an arc) base C of $S^s_{\overline{x_m}}$ (resp. $S^s_{x_m}$) and a free base G of S_{c_i} (note that this is possible by definition of $S^s_{\overline{x_m}}$, $S^s_{x_m}$ and S_{c_i}). On the whole, the instance we have constructed is composed of $3q + 2n$ arcs. We denote by APS-CP-construction any construction of this type. In the following, we will distinguish arcs between bases A and U, denoted by AU-arcs, from arcs between bases C and G, denoted by CG-arcs. An illustration of an APS-CP-construction is given in Figure 1. Clearly, our construction can be carried on in polynomial time. Moreover, the result of such a construction is indeed an instance of APS(CROSSING,PLAIN), since no arc is added to T and any base of S is incident to at most one arc.

We begin by proving a canonicity lemma of an APS-CP-construction.

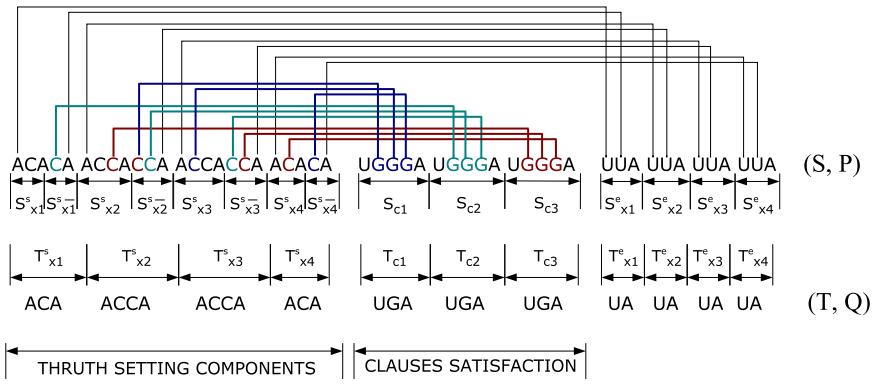

Fig. 1. Example of an APS-CP-construction with $\mathcal{C}_q = (x_2 \vee \overline{x_3} \vee x_4) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4})$

Lemma 1. *Let (S, P) and (T, Q) be any two arc-annotated sequences obtained from an* APS-CP*-construction. If (T, Q) can be obtained from (S, P) by deleting some of its bases together with their incident arcs, if any, then for each $1 \leq i \leq q$ and $1 \leq m \leq n$:*

1. *T_{c_i} is obtained from S_{c_i} by deleting two of its three bases G,*
2. *$T^e_{x_m}$ is obtained from $S^e_{x_m}$ by deleting one of its two bases U,*
3. *$T^s_{x_m}$ is obtained from $S^s_{x_m} A S^s_{\overline{x_m}}$, by deleting either $S^s_{x_m}$ or $S^s_{\overline{x_m}}$.*

Proof. Let (S, P) and (T, Q) be two arc-annotated sequences resulting from an APS-CP-construction.
(1) By construction, the first base U appearing in S (resp. T) is $S_{c_1}[1]$ (resp. $T_{c_1}[1]$). Thus, $T_{c_1}[1]$ is obtained from a base U of S at, or after, $S_{c_1}[1]$. Moreover, the number of bases A appearing after $S_{c_1}[1]$ in S is equal to the number of bases A appearing after $T_{c_1}[1]$ in T. Therefore, every base A appearing after $S_{c_1}[1]$ and $T_{c_1}[1]$ must be matched. That is, for each $1 \leq i \leq q$, $T_{c_i}[3]$ is matched to $S_{c_i}[5]$. In particular, $T_{c_q}[3]$ is matched to $S_{c_q}[5]$. But since there are as many bases U between $S_{c_1}[1]$ and $S_{c_q}[5]$ as there are between $T_{c_1}[1]$ and $T_{c_q}[3]$, any base U in this interval in S must be matched to any base U in this interval in T ; that is, for any $1 \leq i \leq q$, $T_{c_i}[1]$ is matched to $S_{c_i}[1]$. Thus, we conclude that for any $1 \leq i \leq q$, T_{c_i} is obtained by deleting two of the three bases G of S_{c_i}.
(2) By the above argument concerning the bases A appearing after $S_{c_1}[1]$ and $T_{c_1}[1]$, we know that if (T, Q) can be obtained from (S, P), then $T^e_{x_m}[2]$ is matched to $S^e_{x_m}[3]$ for any $1 \leq m \leq n$. Thus, for any $1 \leq m \leq n$, $T^e_{x_m}$ is obtained from $S^e_{x_m}$, and in particular $T^e_{x_m}[1]$ is matched to either $S^e_{x_m}[1]$ or $S^e_{x_m}[2]$.
(3) By definition, as there is no arc incident to bases of T, at least one base incident to every arc of P has to be deleted. We just mentioned that $T^e_{x_m}[1]$ is matched to either $S^e_{x_m}[1]$ or $S^e_{x_m}[2]$ for any $1 \leq m \leq n$. Thus, since by construction there is an arc between $S^e_{x_m}[1]$ and $S^s_{x_m}[1]$ (resp. $S^e_{x_m}[2]$ and $S^s_{\overline{x_m}}[l_{\overline{m}}]$), for any $1 \leq m \leq n$ either $S^s_{x_m}[1]$ or $S^s_{\overline{x_m}}[l_{\overline{m}}]$ has to be deleted ; and all these arcs connect a base A appearing before $S_{c_1}[1]$ to a base U appearing after $S_{c_q}[5]$. Therefore, for any $1 \leq m \leq n$ a base A appearing before $S_{c_1}[1]$ in S is deleted. Originally, there are $3n$ bases A appearing before $S_{c_1}[1]$ in S and $2n$ appearing before the first base of $T_{c_1}[1]$ in T. Thus, the number of bases A not deleted in S and appearing before $S_{c_1}[1]$ is equal to the number of bases A appearing before $T_{c_1}[1]$ in T. But since, for each $1 \leq m \leq n$, a base A of either $S^s_{x_m}$ or $S^s_{\overline{x_m}}$ is deleted, we conclude that for each $1 \leq m \leq n$, $T^s_{x_m}$ is obtained from $S^s_{x_m} A S^s_{\overline{x_m}}$, by deleting either $S^s_{x_m}$ or $S^s_{\overline{x_m}}$. □

We now turn to proving that our construction is a polynomial time reduction from EXACT 3-CNF-SAT to APS(CROSSING, PLAIN).

Lemma 2. *Let I be an instance of the problem* EXACT 3-CNF-SAT *with n variables and q clauses, and I' an instance $((S,P);(T,Q))$ of* APS(CROSSING,PLAIN) *obtained by an* APS-CP*-construction from I. An assignment of the variables exists and satisfies the boolean formula of I iff T is an Arc-Preserving Subsequence of S.*

Proof. ($\Rightarrow$) Suppose we have an assignment AS of the n variables that satisfies the boolean formula of I. By definition, for each clause there is at least one literal that satisfies it. In the following, j_i will define, for any $1 \leq i \leq q$, the smallest index of the literal of c_i (i.e. 1, 2 or 3) which, by its assignment, satisfies c_i. Let (S,P) and (T,Q) be two sequences obtained from an APS-CP-construction from I. We look for a set $\mathcal{B}$ of bases to delete from S in order to obtain T. For each variable $x_m \in AS$ with $1 \leq m \leq n$, we define $\mathcal{B}$ as follows:

- if $x_m = True$ then $\mathcal{B}$ contains each base of $S^s_{\overline{x_m}}$ and $S^e_{x_m}[1]$,
- if $x_m = False$ then $\mathcal{B}$ contains each base of $S^s_{x_m}$ and $S^e_{x_m}[2]$,
- if $j_i = 1$ then $\mathcal{B}$ contains $S_{c_i}[3]$ and $S_{c_i}[4]$,
- if $j_i = 2$ then $\mathcal{B}$ contains $S_{c_i}[2]$ and $S_{c_i}[4]$,
- if $j_i = 3$ then $\mathcal{B}$ contains $S_{c_i}[2]$ and $S_{c_i}[3]$.

Since a variable has a unique value (i.e. True or False), either each base of $S^s_{\overline{x_m}}$ and $S^e_{x_m}[1]$ or each base of $S^s_{x_m}$ and $S^e_{x_m}[2]$ are in $\mathcal{B}$ for all $1 \leq m \leq n$. Thus, $\mathcal{B}$ contains at least one base in S of any AU-arc of P.

For any $1 \leq i \leq q$, two of the three bases G of S_{c_i} are in $\mathcal{B}$. Thus, $\mathcal{B}$ contains at least one base in S of two thirds of the CG-arcs of P. Moreover, $S_{c_i}[j_i+1]$ is the base G that is not in $\mathcal{B}$. We suppose in the following that the j_i^{th} literal of the clause c_i is x_m, with $1 \leq m \leq n$. Thus, by the way we build the APS-CP-construction, there is an arc between a base C of $S^s_{\overline{x_m}}$ and $S_{c_i}[j_i+1]$ in P. By definition, if AS is an assignment of the n variables that satisfies the boolean formula, AS satisfies c_i and thus $x_m = True$. We mentioned, in the definition of $\mathcal{B}$ that if $x_m = True$ then each base of $S^s_{\overline{x_m}}$ is in $\mathcal{B}$. Thus, the base C of $S^s_{\overline{x_m}}$ incident to the CG-arc in P with $S_{c_i}[j_i+1]$ is in $\mathcal{B}$. A similar result can be found if the j_i^{th} literal of the clause c_i is $\overline{x_m}$. Thus, $\mathcal{B}$ contains at least one base in S of any CG-arc of P.

If S' is the sequence obtained from S by deleting all the bases of $\mathcal{B}$ together with their incident arcs, if any, then there is no arc in S' (i.e. neither AU-arcs or CG-arcs). By the way we define $\mathcal{B}$, S' is obtained from S by deleting all the bases of either $S^s_{x_m}$ or $S^s_{\overline{x_m}}$, two bases G of S_{c_i} and either $S^e_{x_m}[1]$ or $S^e_{x_m}[2]$, for $1 \leq i \leq q$ and $1 \leq m \leq n$. It is easily seen that the sequence S' obtained is similar to T.

($\Leftarrow$) Let I be an instance of the problem EXACT 3-CNF-SAT with n variables and q clauses. Let I' be an instance $((S,P);(T,Q))$ of APS(CROSSING,PLAIN) obtained by an APS-CP-construction from I such that (T,Q) can be obtained from (S,P) by deleting some of its bases (i.e. a set of bases $\mathcal{B}$) together with their incident arcs, if any. By Lemma 1, either all bases of $S^s_{\overline{x_m}}$ or all bases of $S^s_{x_m}$ are in $\mathcal{B}$. Consequently, for $1 \leq m \leq n$, we define an assignment AS of the n variables of I as follows:

- if all bases of $S^s_{\overline{x_m}}$ are in $\mathcal{B}$ then $x_m = True$,
- if all bases of $S^s_{x_m}$ are in $\mathcal{B}$ then $x_m = False$.

Now, let us prove that for any $1 \leq i \leq q$ the clause c_i is satisfied by AS. By Lemma 1, for any $1 \leq i \leq q$ there is a base G of the segment S_{c_i} (say the $j_i + 1^{th}$) that is not in $\mathcal{B}$. By the the way we build the APS-CP-construction, there is a CG-arc in P between $S_{c_i}[j_i + 1]$ and a base C of $S^s_{\overline{x_m}}$ (resp. $S^s_{x_m}$) if the j_i^{th} literal of c_i is x_m (resp. $\overline{x_m}$).

Suppose, *w.l.o.g.*, that the j_i^{th} literal of c_i is x_m. Since Q is an empty set, at least one base of any arc of P is in $\mathcal{B}$. Thus, the base C of $S^s_{\overline{x_m}}$ incident to the CG-arc in P with $S_{c_i}[j_i + 1]$ is in $\mathcal{B}$ (since $S_{c_i}[j_i + 1] \notin \mathcal{B}$). Therefore, by Lemma 1, all the bases of $S^s_{\overline{x_m}}$ are in $\mathcal{B}$. By the way we define AS, $x_m = True$ and thus c_i is satisfied. A similar result can be obtained if the j_i^{th} literal of c_i is $\overline{x_m}$. □

We have thus proved the following.

Theorem 1. APS(CROSSING, PLAIN) *is* **NP**-*complete.*

4 Refinement of APS

In this section, we propose a refinement of the APS problem. We first state formally our approach and explain why such a refinement is relevant for both theoretical and experimental studies. We end the section by giving easy properties of the proposed refinement that will prove extremely useful in Section 5.

4.1 Splitting the levels

Theorem 1 answers the last open problem concerning the computational complexity of the APS problem with respect to classical complexity levels, *i.e.*, PLAIN, CHAIN, NESTED and CROSSING (cf. Table 1). However, we are mainly interested in the elaboration of the precise border between **NP**-hard and polynomially solvable cases. Indeed, both theorists and practitioners might naturally ask for more information concerning the hard cases of the APS problem in order to get valuable insight into what makes the problem difficult.

As a next step towards better understanding what makes the APS problem hard, we decide to refine the models which are classically used for classifying arc-annotated sequences. Our refinement consists in splitting those models of arc-annotated sequences into more precise relations between arcs. For example, such a refinement provides a general framework for investigating polynomial time solvable and hard restricted instances of APS(CROSSING, PLAIN), thereby refining in many ways Theorem 1 (see Section 5).

We use the three relations first introduced by Vialette [18, 19] in the context of 2-*intervals* (a simple abstract structure for modelling RNA secondary structures). Actually, his definition of 2-intervals could almost apply in this paper (the main difference lies in the fact that Vialette used 2-intervals for representing sets of contiguous arcs). Vialette defined three possible relations between 2-intervals that can be used for arc-annotated sequences as well. They are the following: for any two arcs $A_1 = (i, j)$ and $A_2 = (k, l)$ in P, we will write $A_1 < A_2$ if $i < j < k < l$ (precedence relation), $A_1 \sqsubset A_2$ if $k < i < j < l$ (nested relation) and $A_1 \between A_2$ if $i < k < j < l$ (crossing relation). Two arcs A_1 and A_2 are τ-comparable for some $\tau \in \{<, \sqsubset, \between\}$ if $A_1 \tau A_2$ or $A_2 \tau A_1$. Let $\mathcal{P}$ be a set of arcs and R be a non-empty subset of $\{<, \sqsubset, \between\}$. The set $\mathcal{P}$ is said to be *R-comparable* if any two distinct arcs of $\mathcal{P}$ are τ-comparable for some $\tau \in R$. An arc-annotated sequence (P, A) is said to be an R-arc annotated sequence for some non-empty subset R of $\{<, \sqsubset, \between\}$ if A is R comparable. Because $<$, $\sqsubset$ and $\between$ are

binary relations, it is convenient to introduce one additional symbol, written $\frown$, to deal with arc-annotated sequences which contain only one arc.

As a straightforward illustration of the above definitions, classical complexity levels for the APS problem can be expressed in terms of our new relations: PLAIN is fully described by $R = \emptyset$, CHAIN is fully described by $R = \{<\}$, NESTED is fully described by $R = \{<, \sqsubset\}$ and CROSSING is fully described by $R = \{<, \sqsubset, \between\}$. The key point is to observe that our refinement allows us to consider new structures for arc-annotated sequences, namely $R = \{\sqsubset\}$, $R = \{\between\}$, $R = \{<, \between\}$ and $R = \{\sqsubset, \between\}$, which could not be considered using the classical complexity levels. Although other refinements may be possible (in particular well-suited for parameterized complexity analysis), we do believe that such an approach allows a more precise analysis of the complexity of the APS problem.

Of course one might object that some of these subdivisions are unlikely to appear in RNA secondary structures. While it is true, it is also true that it is of great interest to answer, at least partly, the following question: Where is the precise boundary between the polynomial and the **NP**-completeness cases ? Indeed, such a question is relevant for both theoretical and experimental studies.

For one, many important optimization problems are known to be **NP**-hard. That is unless **P** = **NP**, there is no polynomial time algorithm that optimally solves these on every input instance. Hence proving a problem to be **NP**-hard is generally accepted as a proof of its difficulty. However the problem to be solved may be much more specialized than the general one that was proved to be **NP**-complete. Therefore, during the past three decades, many studies have been devoted to proving **NP**-hardness for highly restricted instances in order to precisely define the border between tractable and intractable problems. Our refinements have thus to be seen as another step towards establishing the precise complexity landscape of the APS problem.

For another, it is worthwhile keeping in mind that intractability must be coped with and problems must be solved in practical applications. Computer science theory has articulated a few general programs for systematically coping with the ubiquitous phenomena of computational intractability: average case analysis, approximation algorithm, randomized algorithm or fixed parameter complexity. Fully understanding where the boundary lies between efficiently solvable formulations and intractable ones is another important approach. Indeed, from an engineering point of view for which the emphasis is on efficiency, that precise boundary might be a good starting point for designing efficient heuristics or for exploring fixed-parameter tractability. The better our understanding of the problem, the better our ability in defining efficient algorithms for practical applications.

4.2 Immediate results

First, observe that, as in Table 1, we only have to consider cases of APS(R_1,R_2) where $R_1 \subseteq R_2$. Indeed, if this is not the case, we can immediately answer negatively since there exists two arcs in T which satisfy a relation in R_2 which is not in R_1, and hence T simply cannot be obtained from S by deleting bases of S. Moreover, in addition to that, other useless cases appear in our refined model; they are simply denoted by "/////" in Table 2.

Some known results allow us to fill many entries of the new complexity table derived from our refinement. The remainder of this subsection is devoted to detailing these first easy statements. We begin with an easy observation concerning complexity propagation properties of the APS problems in our refined model.

Observation 1 *Let R_1, R_2, R'_1 and R'_2 be four subsets of $\{\frown, <, \sqsubset, \between\}$ such that $R'_2 \subseteq R_2 \subseteq R_1$ and $R'_2 \subseteq R'_1 \subseteq R_1$. If* APS$(R'_1, R'_2)$ *is* **NP***-hard (resp.* APS(R_1, R_2) *is polynomial time solvable) then* APS(R_1, R_2) *is* **NP***-hard (resp.* APS(R'_1, R'_2) *is polynomial time solvable) as well.*

On the positive side, Gramm *et al.* have shown that APS(NESTED, NESTED) is solvable in $O(nm)$ time [11]. Another way of stating this is to say that APS$(\{<, \sqsubset\}, \{<, \sqsubset\})$ is solvable in $O(mn)$ time. That result together with Observation 1 may be summarized by saying that APS(R_1, R_2) for some compatible R_1 and R_2 such that $\between \notin R_1$ and $\between \notin R_2$ is polynomial time solvable.

Conversely, Evans has proved that APS(CROSSING,CROSSING) is **NP**-complete [6]. A simple reading shows that her proof is concerned with $\{<, \sqsubset, \between\}$-annotated sequences, and hence she actually proved that APS$(\{<, \sqsubset, \between\}, \{<, \sqsubset, \between\})$ is **NP**-complete. Similarly, in proving that APS(CROSSING,CHAIN) is **NP**-complete [12], Guo actually proved that APS$(\{<, \sqsubset, \between\}, \{<\})$ is **NP**-complete. Note that according to Observation 1, this latter result implies that APS$(\{<, \sqsubset, \between\},\{<, \sqsubset\})$ and APS$(\{<, \sqsubset, \between\},\{<, \between\})$ are **NP**-complete. We end by another way of stating Theorem 1.

Proposition 1. APS$(\{\sqsubset, \between\}, \emptyset)$ *is* **NP***-complete.*

Proof. A careful examination of the proof of Theorem 1 shows that the construction is actually an instance of APS$(\{\sqsubset, \between\}, \emptyset)$. □

Furthermore, it easily follows from Proposition 1 and Observation 1 that APS(R_1, R_2) for some compatible R_1 and R_2 such that $\{\sqsubset, \between\} \subseteq R_1$ is **NP**-complete. Summarizing we are now left with eight new problems of unknown complexity, each of them being a subcase of a **NP**-hard case of the APS problem. We will provide in the next section polynomial time algorithms for three of them.

APS									
R_1 vs R_2	$\{<, \sqsubset, \between\}$	$\{\sqsubset, \between\}$	$\{<, \between\}$	$\{\between\}$	$\{<, \sqsubset\}$	$\{\sqsubset\}$	$\{<\}$	$\{\frown\}$	$\emptyset$
$\{<, \sqsubset, \between\}$	**NP**-C$\circ$	**NP**-C$*$	**NP**-C$\bullet$	**NP**-C$*$	**NP**-C$\bullet$	**NP**-C$*$	**NP**-C$\bullet$	**NP**-C$*$	**NP**-C$*$
$\{\sqsubset, \between\}$		**NP**-C$*$	/////	**NP**-C$*$	/////	**NP**-C$*$	/////	**NP**-C$*$	**NP**-C$*$
$\{<, \between\}$			**?**	**?**	/////	/////	**?**	**?**	**?**
$\{\between\}$				$O(nm^2)*$	/////	/////	/////	$O(nm^2)*$	$O(nm^2)*$
$\{<, \sqsubset\}$					$O(nm)\star$	$O(nm)\star$	$O(nm)\star$	$O(nm)\star$	$O(nm)\star$
$\{\sqsubset\}$						$O(nm)\star$	/////	$O(nm)\star$	$O(nm)\star$
$\{<\}$							$O(nm)\star$	$O(nm)\star$	$O(n+m)\star$
$\{\frown\}$								$O(n+m)\star$	$O(n+m)\star$
$\emptyset$									$O(n+m)\star$

Table 2. Complexity results after refinement of the complexity levels : $*$: results from this paper ; $\circ$: result from [6] ; $\bullet$: results from [12] ; $\star$: results from [11]

5 Three polynomial time solvable APS problems

We prove in this section that APS$(\{\between\}, \emptyset)$, APS$(\{\between\}, \{\frown\})$ and APS$(\{\between\},\{\between\})$ are polynomial time solvable. In other words, the relation $\between$ alone does not imply **NP**-completeness.

We need the following notations. Sequences are the concatenation of zero or more elements from an alphabet. We use the period . as the concatenation operator, but frequently the two operands are simply put side by side. Let $T = T[1]\,T[2]\ldots T[m]$ be a sequence of length m. For all $1 \leq i \leq j \leq m$, we write $T[i:j]$ to denote $T[i]\,T[i+1]\ldots T[j]$ The *reverse* of T is the sequence $T^R = T[m]\ldots T[2]\,T[1]$. A *factorization* of T is any decomposition $T = x_1\,x_2\ldots x_q$ where $x_1, x_2, \ldots x_q$ are (possibly empty) sequences. Let (T, A) be a $\{\langle\rangle\}$-arc annotated sequence and $(i,j) \in A$, $i < j$, be an arc. We call $T[i]$ a *forward base* and $T[j]$ a *backward base*. We will denote by LF_T the position of the last forward base in (T, A) and by FB_T the position of the first backward base in (T, A), *i.e.*, $\mathsf{LF}_T = \max\{i : (i,j) \in A\}$ and $\mathsf{FB}_T = \min\{j : (i,j) \in A\}$. By convention, we let $\mathsf{LF}_T = 0$ and $\mathsf{FB}_T = |T| + 1$ if $A = \emptyset$. Observe that $\mathsf{LF}_T < \mathsf{FB}_T$.

We begin by proving a factorization result on $\{\langle\rangle\}$-annotated sequences.

Lemma 3. *Let S and T be two $\{\langle\rangle\}$-annotated sequences of length n and m, respectively. If T occurs as an arc preserving subsequence in S, then there exists a factorization (possibly trivial) $T[\mathsf{LF}_T+1 : \mathsf{FB}_T-1] = xy$ such that $T[1 : \mathsf{LF}_T] \cdot x \cdot (y \cdot T[\mathsf{FB}_T : m])^R$ occurs as an arc preserving subsequence in $S[1 : \mathsf{FB}_S-1] \cdot S[\mathsf{FB}_S : n]^R$.*

Proof. Suppose that T occurs as an arc preserving subsequence in S. Since both S and T are $\{\langle\rangle\}$-annotated sequences, then there exist two factorizations $S[1 : \mathsf{LF}_S] = uw$ and $S[\mathsf{FB}_S : n] = zv$ such that: (i) $T[1 : \mathsf{LF}_T]$ occurs in u, (ii) $T[\mathsf{LF}_T+1 : \mathsf{FB}_T-1]$ occurs in $w \cdot S[\mathsf{LF}_S+1 : \mathsf{FB}_S-1] \cdot z$ and (iii) $T[\mathsf{FB}_T : m]$ occurs in v. Then it follows that there exists a factorization $T[\mathsf{LF}_T+1 : \mathsf{FB}_T-1] = xy$ such that x occurs in $w \cdot S[\mathsf{LF}_S+1 : \mathsf{FB}_S-1]$ and y occurs in z, and hence $T' = T[1 : \mathsf{LF}_T] \cdot x \cdot (y \cdot T[\mathsf{FB}_T : m])^R$ occurs as an arc preserving subsequence in $S' = S[1 : \mathsf{FB}_S-1] \cdot S[\mathsf{FB}_S : n]^R$ (see Figure 2). □

Theorem 2. APS($\{\langle\rangle\}$,$\{\langle\rangle\}$) *is solvable in $O(m^2n)$ time.*

Proof. The algorithm is as follows:

```
   Data   : Two {⟨⟩}-annotated sequences S and T of length n and m, respectively
   Result : true iff T occurs as an arc preserving subsequence in S
   begin
1      S' = S[1 : FB_S −1] · S[FB_S : n]^R
2      foreach factorization T[LF_T +1 : FB_T −1]| = xy do
3          T' = T[1 : LF_T] · x · (y · T[FB_T : m])^R
4          if T' occurs as an arc preserving subsequence in S' then
5              return true
6      return false
   end
```

Correctness of the algorithm follows from Lemma 3. What is left is to prove the time complexity. Clearly, $S' = S[1 : \mathsf{FB}_S-1] \cdot S[\mathsf{FB}_S : n]^R$ is a $\{\sqsubset\}$-annotated sequence. The key point is to note that, for any factorization $T[\mathsf{LF}_T+1 : \mathsf{FB}_T-1]| = xy$, the obtained $T' = T[1 : \mathsf{LF}_T] \cdot x \cdot (y \cdot T[\mathsf{FB}_T : m])^R$ is a $\{\sqsubset\}$-annotated sequence as-well. Now let k be the number of arcs in T. So there are at most $m - 2k$ iterations to go before eventually returning false. According to the above, Line 4 constitutes an instance of APS($\{\sqsubset\}$,$\{\sqsubset\}$). But APS($\{\sqsubset\}$,$\{\sqsubset\}$) is a special case of APS($\{<, \sqsubset\}$,$\{<, \sqsubset\}$), and hence is solvable in $O(mn)$ time [11]. Then it follows that the algorithm as a whole runs in $O(mn(m-2k)) = O(m^2n)$ time. □

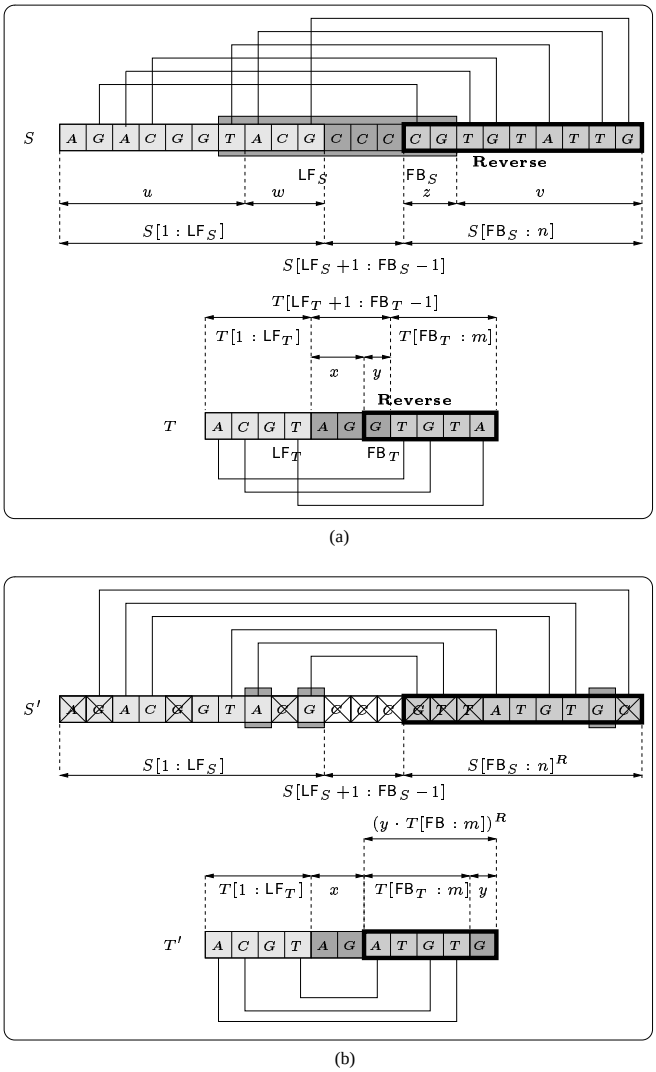

Fig. 2. Illustration of Lemma 3.

Clearly, proof of Theorem 2 relies on an efficient algorithm for solving APS($\{\sqsubset\}$,$\{\sqsubset\}$): the better the complexity for APS($\{\sqsubset\}$,$\{\sqsubset\}$), the better the complexity for APS($\{\not\sqsubset\}$,$\{\not\sqsubset\}$). We have used only the fact that APS($\{\sqsubset\}$,$\{\sqsubset\}$) is a special case of APS($\{<,\sqsubset\}$,$\{<,\sqsubset\}$). It remains open, however, wether a better complexity can be achieved for APS($\{\sqsubset\}$,$\{\sqsubset\}$).

Theorem 2 carries out easily to restricted versions.

Corollary 1. APS($\{\not\sqsubset\}$,$\{\frown\}$) *and* APS($\{\not\sqsubset\}$,$\emptyset$) *are solvable in* $O(m^2n)$ *time.*

Proof. Theorem 2 and Observation 1. □

6 Conclusion

We investigated the time complexity of the Arc-Preserving Subsequence problem (APS). We proved that APS(Crossing,Plain) is **NP**-complete thereby answering an open problem

posed in [11]. Note that this result answers the last open problem concerning the computational complexity of the APS problem with respect to complexity levels, *i.e.*, PLAIN, CHAIN, NESTED and CROSSING (cf. Table 1).

Also, we refined the four above mentioned levels for exploring the border between polynomial time problems and **NP**-complete problems. Some previous known results concerning APS (the one concerning APS(CROSSING,PLAIN) included) allowed us to fill most of the entries of the new table. Moreover, we answered three more problems by showing that the relation $\between$ alone does not imply **NP**-completeness. There are five remaining open problems, which are of the form APS($\{<, \between\}$,R), for any relation R compatible with $\{<, \between\}$. Not surprisingly, those five open problems can be seen as standing on the border between **NP**-completeness and polynomiality. In some sense, this justifies the refinement we decided to adopt, although at the present time, we do not know yet where the gap between **NP**-completeness and polynomiality lies exactly. It is of course a challenging problem to determine the complexities of those five remaining open problems.

References

1. J. Alber, J. Gramm, J. Guo, and R. Niedermeier. Towards optimally solving the longest common subsequence problem for sequences with nested arc annotations in linear time. In *Proc. of the 13th Annual Symposium on Combinatorial Pattern Matching (CPM 2002)*, volume 2373 of *LNCS*, pages 99–114. Springer-Verlag, 2002.
2. 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.
3. B. Billoud, M.-A. Guerrucci, M. Masselot, and J.S. Deutsch. Cirripede phylogeny using a novel approach: Molecular morphometrics. *Molecular Biology and Evolution*, 19:138–148, 2000.
4. G. Caetano-Anollés. Tracing the evolution of RNA structure in ribosomes. *Nucl. Acids. Res.*, 30:2575–2587, 2002.
5. W. Chaia and V. Stewart. RNA Sequence Requirements for NasR-mediated, Nitrate-responsive Transcription Antitermination of the Klebsiella oxytoca M5al nasF Operon Leader. *Journal of Molecular Biology*, 292:203–216, 1999.
6. P. Evans. *Algorithms and Complexity for Annotated Sequence Analysis.* PhD thesis, University of Victoria, 1999.
7. P. Evans. Finding common subsequences with arcs and pseudoknots. In *Proc. of the 10th Annual Symposium Combinatorial Pattern Matching (CPM 1999)*, volume 1645 of *LNCS*, pages 270–280. Springer-Verlag, 1999.
8. A.D. Farris, G. Koelsch, G.J. Pruijn, W.J. van Venrooij, and J.B. Harley. Conserved features of Y RNAs revealed by automated phylogenetic secondary structure analysis. *Nucl. Acids. Res.*, 27:1070–1078, 1999.
9. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.
10. D. Goldman, S. Istrail, and C.H. Papadimitriou. Algorithmic aspects of protein structure similarity. In *Proc. of the 40th Annual Symposium of Foundations of Computer Science (FOCS99)*, pages 512–522, 1999.
11. J. Gramm, J. Guo, and R. Niedermeier. Pattern matching for arc-annotated sequences. In *Proc. of the the 22nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2002)*, volume 2556 of *LNCS*, pages 182–193, 2002.
12. J. Guo. Exact algorithms for the longest common subsequence problem for arc-annotated sequences. Master's Thesis, Universitat Tubingen, Fed. Rep. of Germany, 2002.
13. K. Hellendoorn, P.J. Michiels, R. Buitenhuis, and C.W. Pleij. Protonatable hairpins are conserved in the 5'-untranslated region of tymovirus RNAs. *Nucl. Acids. Res.*, 24:4910–4917, 1996.
14. L. Hofacker, M. Fekete, C. Flamm, M.A. Huynen, S. Rauscher, P.E. Stolorz, and P.F. Stadler. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. *Nucl. Acids. Res.*, 26:3825–3836, 1998.
15. T. Jiang, G.-H. Lin, B. Ma, and K. Zhang. The longest common subsequence problem for arc-annotated sequences. In *Proc. 11th Annual Symposium on Combinatorial Pattern Matching (CPM 2000)*, volume 1848 of *LNCS*, pages 154–165. Springer-Verlag, 2000.

16. V. Juan, C. Crain, and S. Wilson. Evidence for evolutionarily conserved secondary structure in the H19 tumor suppressor RNA. *Nucl. Acids. Res.*, 28:1221–1227, 2000.
17. S.W.M. Teunissen, M.J.M. Kruithof, A.D. Farris, J.B. Harley, W.J. van Venrooij, and G.J.M. Pruijn. Conserved features of Y RNAs: a comparison of experimentally derived secondary structures. *Nucl. Acids. Res.*, 28:610–619, 2000.
18. S. Vialette. Pattern matching over 2-intervals sets. In *Proc. 13th Annual Symposium Combinatorial Pattern Matching (CPM 2002)*, volume 2373 of *LNCS*, pages 53–63. Springer-Verlag, 2002.
19. S. Vialette. On the computational complexity of 2-interval pattern matching. *Theoretical Computer Science*, 312(2-3):223–249, 2004.
20. H.-Y. Wang and S.-C. Lee. Secondary structure of mitochondrial 12S rRNA among fish and its phylogenetic applications. *Molecular Biology and Evolution*, 19:138–148, 2002.
21. J. Wuyts, P. De Rijk, Y. Van de Peer, G. Pison, P. Rousseeuw, and R. De Wachter. Comparative analysis of more than 3000 sequences reveals the existence of two pseudoknots in area V4 of eukaryotic small subunit ribosomal RNA. *Nucl. Acids. Res.*, 28:4698–4708, 2000.
22. K. Zhang, L. Wang, and B. Ma. Computing the similarity between RNA structures. In *Proc. 10th Annual Symposium on Combinatorial Pattern Matching (CPM 1999)*, volume 1645 of *LNCS*, pages 281–293. Springer-Verlag, 1999.
23. M. Zuker. RNA folding. *Meth. Enzymology*, 180:262–288, 1989.