A new method to predict the consensus secondary structure of a set of unaligned RNA sequences

D. Bouthinon and H. Soldano

LIPN, UPRES-A 7030 CNRS, Av. J.B. Clément, 93430 VILLETANEUSE (France)
Atelier de BioInformatique, Université Paris VI, 12 rue Cuvier, 75005 PARIS (France)

Abstract

Motivation: Predict the consensus secondary structure, possibly including pseudoknots, of a set of RNA unaligned sequences

Results: We have designed a method based on a new representation of any RNA secondary structure as a set of structural relations between the helices of the structure. We refer to this representation as a *structural pattern*. In a first step we use thermodynamic parameters to select, for each sequence, the best secondary structures according to energy minimisation and we represent each of them using its corresponding *structural pattern*. In a second step we search for the *repeated structural patterns* i.e. the largest structural patterns that occur in at least one sequence, i.e. included in at least one of the structural patterns associated to each sequence. Thanks to an efficient encoding of structural patterns this search comes down to identify the largest repeated word suffixes in a dictionary. In a third step we compute the plausibility of each repeated structural pattern by checking if it occurs more frequently in the studied sequences than in random RNA sequences. We then suppose that the consensus secondary structure corresponds to the repeated structural pattern that displays the highest plausibility. We present several experiments concerning tRNA, fragments of 16S rRNA and 10Sa RNA (including pseudoknots): in each of them we found the putative consensus secondary structure.

Availability: The program corresponding to the algorithms described hereafter is written in ansi-C and is available upon request.

Contact: Dominique.Bouthinon@snv.jussieu.fr

Introduction

The prediction of RNA secondary structures is mainly based on two techniques: energy minimisation and phylogenetic comparisons.

The first one comprises combinatorial and dynamic programming approaches, and is based on computation of the lowest free energy structure(s) of a sequence. The combinatorial approach first generates all the possible helices of a sequence and associates to each possible helix a weight representing its energy. In a second step, a branch and bound algorithm combines compatible helices until optimal or suboptimal structures are formed (Pipas & McMahon, 1975 ; Ninio, 1979 ; Dumas & Ninio, 1982 ; Sankoff and al., 1983 ; Papanicolaou et al., 1984 ; Gouy et al., 1985 ; Gouy, 1987). This technique, that exhaustively search the solution space, cannot deal with sequences much longer than 200 nucleotides. Dynamic programming approach computes the lowest free energy structure by mean of an energy base-pairing optimisation based on a recursive relation between the best structures of length k and the best structures of length k-1 (Nussinov et al., 1978 ; Waterman & Smith, 1978 ; Nussinov & Jacobson, 1980 ; Zuker & Stiegler, 1981 ;

Zuker & Sankoff, 1984 ; Zuker, 1986). Dynamic programming allows to treat sequences containing up to 2000 nucleotides, however such methods method neither consider pseudo-knots nor find suboptimal solutions. A more recent method (Zuker, 1989 ; Zuker et al., 1991 ; Zuker, 1991) partially solves the last problem.

The phylogenetic methods use covariation analysis to identify conserved paired bases among a set of homologous sequences (Noller & Woese, 1981 ; Olsen, 1983 ; Gutell et al., 1986 ; Stern et al., 1988 ; Haselman et al., 1989 ; Han & Kim, 1993 ; Woese & Gutell, 1989 ; Chiu & Kolodziejczak, 1991 ; Gutell et al., 1992). This is a satisfying procedure that gives excellent results, including pseudo-knots identification. However the procedure requires a prior alignment of sequences and multiple alignment is, in turn, a difficult problem.

One must also quote miscellaneous methods based either on parallel algorithms (Nakaya et al., 1995 ; Shapiro and Wu, 1997), or formal grammar (Sakakibara et al., 1994 ; Grate, 1995 ; Lefebvre, 1995 ; Lefebvre, 1996), graph theory (Cary and Stormo 1995) and simulation of the RNA folding process (Martinez 1983 ; Martinez 1988 ; Wuju and JiaJin, 1998).

Recent methods intend both to align RNA sequences and to predict their consensus secondary structure (Eddy and Durbin, 1994 ; Gorodkin et al., 1997 ; Tabaska and Stormo 1997). Finally two recent methods have been proposed in order to predict the secondary structure, possibly included pseudoknots, of a single RNA sequence : (Tabaska et al., 1998) is based on graph theory approach (Maximum Weighted Matching) while (Rivas and Eddy, 1999) relies on dynamic programming.

In this paper we propose a new method to predict the consensus secondary structure of a set of RNA sequences. It is based on a new representation of a secondary structure as a set of structural relations between the helices it contains. We propose an efficient encoding of such a set of relations as a word (of integers), and so reduce the search of the consensus secondary structure to the search for repeated suffixes in a dictionary. Each entry of the dictionary is a word encoding a partial secondary structure. The main purpose of the method is to deal with unaligned RNA sequences while considering all secondary structures patterns, including those corresponding to pseudo-knots.

In the following of this paper we first present an overview of our method. Then we discuss the representation of RNA secondary structures. In the third part we describe a fast algorithm that builds the dictionary mentioned above and searches for repeated suffixes. In part four we discuss the selection of the consensus structure among a set of candidates. Finally we show that the method find the putative consensus secondary structures of different sets of tRNAs, fragments of 16S rRNA and 10Sa RNA.

Overview of the method

The goal of this method is to predict the consensus secondary structure of a set S containing n unaligned RNA sequences.

Our approach mainly relies on a new representation of a RNA secondary structure as a set of structural relations, referred to as *structural pattern*, between the *helices* of the secondary structure (for instance the tRNA structure corresponds to a large helix containing three contiguous helices). Note that the structural pattern associated with a secondary structure made of a given set of helices includes the structural pattern associated with any subset of these helices. Thus, we define the *consensus structural pattern* of a set of RNA sequences as the largest structural pattern included in all (or almost all) structural patterns associated with these sequences.
The main difficulty with determining this consensus structural pattern is the fact that there is no unique secondary structure a priori associated with a given sequence. Thereby there is no unique structural pattern associated with the sequence. A related difficulty is that because of this indeterminacy we need to consider various candidates for the consensus structural pattern. The method we propose is divided into three steps sketched as follows:

1) Compute the structural patterns of each sequence

Because we do not know the secondary structure of a given sequence *s* we cannot identify the helices that compose the structure. Fortunately a RNA sequence contains various string patterns called *palindromes* that represent *virtual helices* (an helix always corresponds to a *palindrome* but the converse is false). As a consequence from the set of palindromes found in *s* we obtain several *virtual secondary structures*, each corresponding to a maximal subset of palindrome*s*. Then we select the set of virtual secondary structures displaying the lowest free-energy (as the energy minimisation techniques do not reliably predict the secondary structure, we cannot reliably select a unique virtual secondary structure corresponding to the lowest free-energy). We assume that the secondary structure of the sequence belongs to this selection. Finally we associate the corresponding structural pattern with each selected structure. We will say that such structural patterns, and all the structural patterns they includes, *occur* in the sequence. This procedure is performed for each sequence *s* belonging to *S*.

2) Compute the repeated patterns

The second step aims to build the set of *repeated structural patterns*. We define a repeated structural pattern as the largest structural patterns that occur in at least *q* (smaller than *n*) sequences of *S*. As a consequence, a repeated pattern can be found even when the secondary structure of some sequence unfortunately has not been selected during the previous step. This allows also to find a consensus structural pattern even when the consensus to be found does not occur in the secondary structure of some sequence in *S*. To reduce the number of repeated structural patterns to consider we will only focus on those that describe sets of *contiguous palindrom*e (see definition below). The consensus structural pattern we search for is supposed to be one of the repeated structural patterns.

3) ***Predict the consensus structural pattern***

Our last step is devoted to select the consensus structural pattern among the repeated structural patterns. To do this we associate to each repeated structural pattern p a number $M(p)$ that intuitively represents the *plausibility* of p to be the consensus structural pattern : not only p has to occur in at least q of the sequences of S, but it should also have few occurrences in a set of random sequences similar to the sequences of S. In order to reduce the number of choices to few (hopefully one) candidates, we rank repeated patterns according to their plausibility.

In what follows we first discuss our representation of structural patterns and then we describe the algorithms computing the three steps presented above.

A new representation of RNA secondary structures as a structural pattern of palindromes

Given a RNA sequence, the RNA secondary structure is basically defined as a sorted set of base-pairings (A-U, G-C, G-U). A *base-pairing* is represented as the couple (i,j) of indexes pointing out the bases paired. An helix is then represented as a sorted set of contiguous base-pairings $((i,j),(i+1,j-1), \ldots, (i+k,j-k))$. Such a set is denoted as a *palindrome*. Figure 1 displays the three parts of the sequence concerned by a palindrome : the *head*, corresponding to the indexes $(i, i+1, \ldots, i+k)$, the *tail* $(j-k, j-k+1 \ldots, j)$ and the *loop* $(i+k+1, \ldots j-k-1)$.

Figure 1

We first give some definitions concerning palindromes and structures :

- *two palindromes are compatible if they do not share any base* (otherwise they would have base-triplets*)*.
- *a palindrome **a** precedes a palindrome **b**, denoted as **a** < **b**, whenever the head of **a*** precedes *the head of **b** on the sequence they come from.*
- *a structure is a sorted set of palindromes* (we will note $\boldsymbol{t}$*:* $\boldsymbol{<p_1p_2\ldots p_m>}$, or simply $\boldsymbol{<p_1p_2\ldots p_m>}$, the structure ***t*** corresponding to the set of palindromes $\{\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_m}\}$).
- *a substructure of a given structure **t** is a subset of **t**.*
- *two palindromes **a** and **b** (with **a** < **b**) belonging to a structure **t**, are contiguous if and only if there is no palindrome **c** in **t**, such that **a** < **c** < **b*** (In other words ***b*** is the successor of ***a*** within the structure ***t****)*.
- *Let **t** be a structure. We will say that **t'** is a contiguous substructure of **t** iff*
  *1) **t'** is a substructure of **t** and*
  *2) each pair of contiguous palindromes in **t'** are contiguous in **t**.*

We see in Figure 3 that the palindromes ***b*** and ***c*** are contiguous, while the palindromes ***b*** and ***d*** are not. Thus ***<bc>*** is a contiguous substructure of ***<abcd>*** while ***<bd>*** is not.

In what follows we will consider that a given palindrome is described only using its structural relations with other palindromes. We will discuss in conclusion issues about considering other attributes describing the palindrome, e.g. attributes regarding the length of its head or its primary sequence.

We denote as *G(**t**)* the *structural pattern* associated to the structure $t\!:<p_1 \ldots p_m>$ and defined as the list of all binary structural relations between pair of palindromes belonging to ***t***. Given a pair of palindromes (p_1, p_2) (we will always suppose that p_1 precedes p_2, i.e. the head of p_1 precedes the head of p_2) p_1 is related to p_2 in exactly one way when considering the position of the tail of p_1 with respect to the head and tail of p_2. The three possible structural relations (see Figure 2) are denoted as :

$J(p_1, p_2)$: p_1 is juxtaposed to p_2 (i.e. the tail of p_1 is before the head of p_2)
$O(p_1, p_2)$: p_1 and p_2 overlap (i.e. the tail of p_1 is between the head and tail of p_2
$I(p_1, p_2)$: p_1 includes p_2 (i.e. the tail of p_1 is after the tail of p_2)

Figure 2

Moreover we consider here that any palindrome ***p*** has a structural relation with itself, noted *Pal(**p**)*. In the present work this relation has no particular meaning but is useful for presentation purpose and allows further addition of non structural information in our representation.

To exemplify this representation we show in Figure 3 the structure ***<abcd>*** that can be described as the following structural pattern:

$$G(<abcd>) = \{Pal(a), I(a,b), I(a,c), O(a,d), Pal(b), J(b,c), J(b,d), Pal(c), O(c,d), Pal(d)\}.$$

Figure 3

Moreover the structural pattern $G(t\!:<p_1 \ldots p_m>)$ is the concatenation of m *partial structural patterns* $P(p_j, <p_j \ldots p_m>)$ $(j \in \{1,\ldots,m\})$, each defined as the sorted set of the binary relations between the palindrome p_j and its successors $\{p_j\, p_{j+1} \ldots p_m\}$[1] .

For instance the structural pattern in Figure 3 can be rewritten as :

[1] When Considering $G(<p_1 \ldots p_m>)$ as a graph, $P(p_j, <p_j p_{j+1} \ldots p_m>)$ is a subgraph containing only edges connected to p_j

$$G(\langle \mathbf{abcd} \rangle) = P(\mathbf{a}, \langle \mathbf{abcd} \rangle) \text{"+"} P(\mathbf{b}, \langle \mathbf{bcd} \rangle) \text{"+"} P(\mathbf{c}, \langle \mathbf{cd} \rangle) \text{"+"} P(\mathbf{d}, \langle \mathbf{d} \rangle)$$

where "+" is the operator of concatenation and where the partial structural patterns are :

$P(\mathbf{a}, \langle \mathbf{abcd} \rangle) = \{Pal(\mathbf{a}), I(\mathbf{a},\mathbf{b}), I(\mathbf{a},\mathbf{c}), O(\mathbf{a},\mathbf{d})\}$
$P(\mathbf{b}, \langle \mathbf{bcd} \rangle) = \{Pal(\mathbf{b}), J(\mathbf{b},\mathbf{c}), J(\mathbf{b},\mathbf{d})\}$
$P(\mathbf{c}, \langle \mathbf{cd} \rangle) = \{Pal(\mathbf{c}), O(\mathbf{c},\mathbf{d})\}$
$P(\mathbf{d}, \langle \mathbf{d} \rangle) = \{Pal(\mathbf{d})\}$

Now we present an efficient encoding of structural patterns in which integers serve as a substitute for structural relations. This encoding relies on the following remark: any partial structural pattern $P(\mathbf{p}_j, \langle \mathbf{p}_j \ldots \mathbf{p}_m \rangle)$ is characterised by the position of the tail of $\mathbf{p}_j$ with respect to the heads and tails of its successors $\{\mathbf{p}_{j+1}, \ldots, \mathbf{p}_m\}$:

a) As $\mathbf{p}_j < \mathbf{p}_k$ for all k belonging to $\{\mathbf{j+1}, \ldots, \mathbf{m}\}$ the head of $\mathbf{p}_j$ is located before all the heads of its successors.
b) The tail of $\mathbf{p}_j$ is preceded by a single number of heads and tails belonging to its successors, thus we can code each $P(\mathbf{p}_j, \langle \mathbf{p}_j \ldots \mathbf{p}_m \rangle)$ with this number (note that $P(\mathbf{p}_j, \langle \mathbf{p}_j \rangle) = \mathbf{0}$).

An example of such an encoding is given Figure 4.

Figure 4

Following this encoding:

$$P(\mathbf{p}_j, \langle \mathbf{p}_j \ldots \mathbf{p}_m \rangle) = \sum_{j<k\leq m} R(\mathbf{p}_j, \mathbf{p}_k)$$

where $R(\mathbf{p}_j,\mathbf{p}_k)$ is a function which value belongs to $\{0, 1, 2\}$. Indeed the tail of $\mathbf{p}_j$ is preceded either by

- no element of $\mathbf{p}_k$ ($R(\mathbf{p}_j,\mathbf{p}_k) = 0$ whenever $J(\mathbf{p}_j, \mathbf{p}_k)$ is true), or by
- the head of $\mathbf{p}_k$ ($R(\mathbf{p}_j,\mathbf{p}_k) = 1$ whenever $O(\mathbf{p}_j, \mathbf{p}_k)$ is true), or by
- the head and the tail of $\mathbf{p}_k$ ($R(\mathbf{p}_j,\mathbf{p}_k) = 2$ whenever $I(\mathbf{p}_j, \mathbf{p}_k)$ is true).

For instance in Figure 4, we obtain $P(\mathbf{a}, \langle \mathbf{abcd} \rangle) = R(\mathbf{a},\mathbf{b}) + R(\mathbf{a},\mathbf{c}) + R(\mathbf{a},\mathbf{d}) = 2 + 2 + 2 = \mathbf{6}$.

Note that there is no ambiguity about the number encoding the partial structural pattern $P(\mathbf{p}_j, \langle \mathbf{p}_j \ldots \mathbf{p}_m \rangle)$. Let us suppose that $\mathbf{p}_j$ has t successors : there are $2t$ heads and tails of these successors that can precede the tail of $\mathbf{p}_j$ and as a consequence $0 \leq P(\mathbf{p}_j, \langle \mathbf{p}_j .. \mathbf{p}_m \rangle) \leq 2t$. It is clear that $P(\mathbf{p}_j, \langle \mathbf{p}_j .. \mathbf{p}_m \rangle)$ corresponds to a single situation of $\mathbf{p}_j$ with respect to its successors and that any another number in the valid range corresponds to a different situation.

As a consequence any structural pattern,

$G(<p_1p_2\ldots p_m>) = P(p_1, <p_1\ldots p_m>)$ "+" $P(p_2, <p_2\ldots p_m>)$ … "+" $P(p_m, <p_m>)$,

will be encoded by a word of m integers, each of them encoding a partial structural pattern. For instance in the case of the structure described Figure 4 :

$G(<abcd>) =$

$P(a,<abcd>)$ "+" $P(b,<bcd>)$ "+" $P(c,<cd>)$ "+" $P(d,<d>) = 6$ "+" 0 "+" 1 "+" 0, *i.e. the word of integers* **6 0 1 0.**

This encoding is not ambiguous : one structural pattern of any RNA structure has exactly one code, two different RNA structures have two different codes. This comes from the fact that the codes of the $P(p_j, <p_j\ldots p_m>)$ are not ambiguous and that there are sorted $(P(p_1, <p_1\ldots p_m>)$ is before $(P(p_2, <p_2\ldots p_m>)$ and so on), so there is one single way ("+") to associate them.

We can use this encoding together with a simple indexing scheme in order to efficiently build the structural patterns of all contiguous substructure*s* of a structure. Let us consider the structure $t: <p_1p_2\ldots p_m>$. We associate to this structure a dictionary *Dt* that contains *m* words. Each word *Dt(i)* encodes the structural pattern associated with the largest contiguous substructure of ***t*** whose last palindrome is p_i as it is exemplified in Figure 5.

Figure 5

Let us denote as *Dt(i,j)* the j^{th} value (numbered from the right to the left) of *Dt(i)*. *Dt(i,j)* is the number encoding the partial structural pattern $P(p_{i-j+1}, <p_{i-j+1}\ldots p_{i-1}p_i>)$. In Figure 5, for instance, $Dt(3,3) = 4$ represents the partial structural pattern $P(a, <abc>)$ between the palindrome **a** and all its successors (*a*, ***b*** and ***c***) in the contiguous substructure *<abc>*. In a same way $Dt(4,3) = 0$ encodes the partial structural pattern $P(b,<bcd>)$ concerns the palindrome ***b*** within the contiguous substructure *<abcd>*.

As shown later, we do not really need to entirely build this dictionary. More precisely as far as we are concerned with *k*-length substructures, we only need the last *k* columns of the dictionary. Moreover we will describe an efficient way to build this dictionary by stepwise adding columns on the left of the dictionary. We will name Dt_k the restricted dictionary containing the *k* last columns of ***Dt***. This dictionary has the same properties as ***Dt*** except that it considers only the contiguous substructures of ***t*** that contains at most *k* palindromes.

It is easy to compute Dt_{k+1} starting from Dt_k together with the binary relations between palindromes, using the following system :

$$\text{(I)}\ \begin{cases} Dt_1(i) = 0 \qquad (1 \le i \le m) \\ \\ (1 \le k < m) \\ \begin{cases} Dt_{k+1}(i) = Dt_k(i) & (1 \le i \le k) \\ Dt_{k+1}(i) = (Dt_k(i-1,k) + R(p_{i-k}, p_i))\ \text{"+"}\ Dt_k(i) & (k < i \le m) \end{cases} \end{cases}$$

In Figure 5, for instance, building Dt_3 from Dt_2 simply consists in adding a column containing the two elements $Dt_3(3,3) = \mathbf{4}$ *and* $Dt_3(4,3) = \mathbf{0}$) at the left of Dt_2. Following the system (I):

$Dt_3(3) = (Dt_2(2,2) + R(\mathbf{a},\mathbf{c}))$ "+" $Dt_2(2) = (2 + \mathbf{2})$ "+" $\mathbf{0\ 0} = \mathbf{4\ 0\ 0}$, and

$Dt_3(4) = (Dt_2(3,2) + R(\mathbf{b},\mathbf{d}))$ "+" $Dt_2(4) = (0 + \mathbf{0})$ "+" $\mathbf{1\ 0} = \mathbf{0\ 1\ 0}$.

Representation of most plausible structures of the sequences belonging to S.

Now recall our original problem: in order to find the consensus structural pattern of a set *S* of sequences, we have first selected a set of virtual secondary structures for each sequence and then we have built their structural patterns. Now we have to build the set of repeated structural patterns i.e. the largest structural patterns included in at least one structural pattern of (at least) *q* sequences. Then we will select a best candidate among these repeated structural patterns.

So as we know how to represent the structural pattern of a virtual secondary structure as a dictionary, it is not difficult, given a sequence, to build a dictionary representing the set of structural patterns of all the virtual secondary structures of the sequence :

Let P(**s**) be the set of structural patterns of a given sequence ***s*** of ***S***. We denote as a *structural pattern dictionary*, the dictionary ***Dt*** associated with a given structural pattern ***t***. We encode all structural patterns of ***s*** by concatenating all structural pattern dictionaries. We call *sequence dictionary*, denoted as D**s,** such a dictionary :

$$(D_s = \bigcup_{t \in P(s)} D_t).$$

In a similar way we form the *general dictionary D* representing the structural patterns of all the sequences in S by concatenating all the sequence dictionaries:

$$(D = \bigcup_{s \in S} D_{P(s)}).$$

Each suffix in this dictionary represents a structural pattern occurring in (at least) one sequence. Searching for the repeated structural patterns comes down to search for the largest suffixes that are present in at least *q* of the sequence dictionaries. Such suffixes will be called r*epeated suffixes* .

Figure 6 shows an example of a general dictionary (and the repeated suffixes it contains) associated with two sequences.

Figure 6

Algorithms

We give hereafter the algorithms corresponding to the three steps of our method :

1) Compute the virtual secondary structures of each sequence

The first step consists in identifying virtual helices (palindromes) in each sequence ***s***. This is performed using a program that belongs to Palingol (Billoud et al., 1996), a package designed to

describe secondary structures and to detect their possible occurrences in banks of sequences. We only consider palindromes that contain at least three nucleotides in each structural element (head, tail and loop)and that meet the usual base-pairings constraints (A-U, G-C, G-U) in a flexible way (i.e. other pairings are occasionally allowed).The worst case time complexity of this step is quadratic with respect to the length of ***s***. The computation of the binary structural relation (either I, J or O) that links each pair of compatible palindromes is performed at the end of this step.
The second step consists in selecting, as virtual secondary structures of each sequence **s**, the **g**[2] lowest free-energy saturated structures according to Salser's thermodynamical parameters (Salser, 1977). This is done using a branch and bound algorithm that searches for the **g** lowest weighted set of *compatible palindromes*. The worst case complexity of this algorithm, as usual regarding branch and bound algorithms, is exponential with respect to the numbers of palindromes in **s**.
The general dictionary D mentioned above is not built at the end of this step. As shown in the next paragraph it is more efficient to only build the part of the dictionary that demonstrates as useful when searching for repeated structural patterns.

2) Compute the repeated structural patterns

The goal of this step is to compute the set of repeated structural patterns represented by the largest repeated suffixes (let us assume that they are of length k_{max})found in the general dictionary D associated with the set of sequences $\boldsymbol{S}$. As a consequence we must both build D and search for the repeated suffixes.

In what follows D_k stands for the general *subdictionary* representing the k last column of D. As a consequence $D_k(i)$ represents the k-length suffix of thei^{th} entry of the general dictionary D, and stands for a k-length structural pattern (i.e. a pattern concerning k palindromes) occurring in at last one sequence. Also recall that $D_k(i,j)$represent*s* the jt^{h} value (numbered from the right to the left) of this suffix).

According to system (I), we build D_{k+1} starting from D_k using for each sequence both the binary structural relations between (compatible) palindromes and the sets of compatible palindromes representing the virtual secondary structures selected during the previous step. Moreover we show hereafter that we can construct all repeated *(k+1)*-length suffixes of D_{k+1} starting from those of length k. We can safely stop the process as soon as there are no repeated suffixes in D_{k+1}, and as a consequence, we avoid to entirely build D when it is not necessary.
Before describing our algorithm to identify the repeated suffixes we present two definitions and one lemma concerning the entries of the subdictionaries D_k and D_{k+1} :

---

[2] g is a user parameter.

Definition 1 Two entries i and i' of D_k satisfy the relation R_k iff $D_k(i,k)=D_k(i',k)$.

Definition 1 states that two entries are related by $\underline{\underline{\mathbf{R_k}}}$ whenever the $\underline{k}^{th}$ palindromes (from right to left) of the two structural patterns are such that the positions of their tails with respect to their k-1 successors are identical.

For instance consider D_3 the subdictionary composed of the 3 last columns of the general dictionary shown in figure 6. We have 8 R_3 11 as the 3-length suffixes **410** and **400** of the entries 8 and 11 have the same 3rd value (from the right). This means that the tails of the 3rd palindromes of the corresponding structures (<***cde***> for entry 8 and <***a'b'c'***> for entry 11) have the same position with respect to their 2 successors.

Definition 2 Two indexes i and i' of D_k are said k-equivalent, noted $i\ E_k\ i'$, whenever their corresponding k-length suffixes are equal (i.e $D_k(i)=D_k(i')$).

Definition 2 states that two entries are k-equivalent whenever their corresponding k-length structural patterns are equal.

For instance consider D_2 the subdictionary composed of the 2 last columns of the general dictionary shown in figure 6. We have 4 E_2 14 because the 2 length suffixes (**10**) of the entries 4 and 14 are equal.

Note that $D_{k+1}(i)=D_{k+1}(i')$ if and only if $D_k(i)=D_k(i')$ and $D_{k+1}(i,k+1)=D_{k+1}(i',k+1)$, and that as a consequence we have the following lemma :

Lemma 1 *$i\ E_{k+1}\ i' \Leftrightarrow i\ E_k\ i'$ and $i\ R_{k+1}\ i'$.*

More formally, given D_{k+1} let us denote as:

- $C(E_k)$ the equivalence classes of the relation E_k. Each class contains the indexes of identical k-length suffixes of D_{k+1} .These suffixes stand for a k-length structural pattern occurring in at least one sequence.
- $C_q(E_k)$ a subset of $C(E_k)$ such that each class of $C_q(E_k)$ contains the indexes of repeated k-length suffixes. These suffixes stand for a k-length structural pattern occuring in at least q sequences.

Thus the goal of this step is to compute $C_q(E_{kmax})$, such that $C_q(E_{kmax}) \neq \emptyset$ and $C_q(E_{kmax+1}) = \emptyset$. Each class of $C_q(E_{kmax})$ points out a largest structural pattern occuring in at least q sequences.

Computing $C_q(E_{kmax})$ is performed using a variant of the algorithm KMR[3], named KMRSTR (Bouthinon and Soldano 1996a ; Bouthinon, 1996b ; Bouthinon & Soldano, 1998a ; Bouthinon and Soldano, 1998b). The main idea of KMRST is to build $C_q(E_{k+1})$ from $C_q(E_k)$. We give hereafter the main lines of this algorithm (details can be found in (Bouthinon, 1996b)) :

KMRSTR Algorithm

input :

- the set S containing **n** RNA sequences
- for each sequence s :

the set V(s) containing the g best free energy virtual secondary structures, each one being simply described as a set of palindromes.

- the number m(s,u) of palindromes in each virtual secondary structure u.
- the couples of compatible palindromes together with the corresponding structural relations.

output :

$C_q(E_{kmax})$, and as a consequence, the repeated structural patterns.

Begin

/* initialisation */

Build a dictionary D_1 containing r entries $(r = \sum_{s\in S} \sum_{u \in V(s)} m(s,u))$ and

fill all cells of D_1 with value **0** ;

/ D_1 contains the unique structural pattern corresponding to Pal(p) */*

Constitute the unique class of $C_q(E_1)$ with all indexes of D_1

/ each index of this class points out the repeated structural pattern Pal(p) */*

k := 1 ; */*k stands for the current length */*

repeat

Build D_{k+1} from D_k using system (I);

/ D_{k+1} contains the (k+1)-length structural patterns */*

Build $C(E_{k+1})$ from $C_q(E_k)$ and D_{k+1} :

1) Dispatch the indexes *i* belonging to the classes of $C_q(E_k)$ in subsets such that in each subset $D(i,k+1)$ has a constant value *c*. We also further refer to such a subset as *c*.

[3] KMR (Karp et al., 1972) is an algorithm searching for repeated words in a string.

*/*note that in each set c the indexes coming from a same class of $C_q(E_k)$ are contiguous (as we scan $C_q(E_k)$ each class in turn) and form a class of $C(E_{k+1})$ (see Lemma 1). */*

2) For each set *c*, gather the contiguous indexes forming a class of $C(E_{k+1})$ in a new class.

Build $C_q(E_{k+1})$ from $C(E_{k+1})$:
eliminate classes of $C(E_{k+1})$ whose indexes correspond to suffixes that do not occur in at least *q* sequence dictionaries ;

k := k + 1 ;

Until $C_q(E_k) = \varnothing$
/ $C_q(E_{kmax})$ is computed ($k_{max} = k-1$). */*
End

So each class of $C_q(E_{kmax})$ contains the indexes corresponding to a particular repeated structural pattern.

Note that the worst case time and space complexity of this algorithm is O($\mathbf{r.k_{max}}$), where **r** is the number of entries in **D**. In other words the algorithm spend a CPU time that is linear with the product of the number of entries and of the maximum length of a repeated pattern. As **S** contains **n** sequences, as each sequence is represented with **g** virtual secondary structures, and considering that each virtual secondary structure contains at most $\underline{\mathbf{p}}$ palindromes, this complexity is O($\mathbf{n.g.\underline{p}.k_{max}}$). Note that **kmax** is less than $\underline{\mathbf{p}}$.

3) Predict the consensus structural pattern

The consensus structural pattern should be one of the repeated structural patterns found by the previous algorithm. As mentioned in the overview, we rank the repeated structural patterns according to a plausibility measure in order to remove random effects. We first introduce some definitions:

Definition 3 We call *biological covering* of a repeated structural pattern T the ratio $F_S(T) = \frac{N_S(T)}{n}$, where $N_S(T)$ is the number of sequences in which T occurs[4], and n is the total number of sequences.

We further call *random sequence* a RNA sequence whose letters have been randomly shuffled.

Definition 4 Let S be a set of RNA sequences and $\boldsymbol{S_R}$ be a set of random sequences, such that each sequence in S has a constant number of random sequences in $\boldsymbol{S_R}$. In such a case $\boldsymbol{S_R}$ is called a *random model* of S.

Let S_R be a random model of S_R, then $F_{SR}(T)$, that we call the *random covering* of the structural pattern T, denotes the proportion of random sequences in which T occurs. Then $F_{SR}(T)$ measures the *noise* concerning T**,** and $(F_S(T)-F_{SR}(T))$ measures the *signal* concerning T, i.e. the proportion of sequences in S in which occurrences of T do not result from random effects.

Definition 5 The *plausibility* of a repeated structural pattern T is defined as : $M_S(t) = \frac{F_S(t) - F_{S_R}(t)}{F_S(t) + F_{S_R}(t)}$

- Note that plausibility is high (~1) whenever the biological covering is high and the random covering is low. On the contrary high values of both biological and random covering results in low plausibility (~0). We will further consider as best candidates the repeated structural patterns corresponding to highest values of plausibility.

Computing the plausibility of each repeated structural pattern T corresponding to a set of sequences S requires to apply the two preceding steps (- search for the $\boldsymbol{g}$ best free-energy virtual secondary structures corresponding to each sequence, 2- build their structural patterns and search for the repeated ones) to a random model S_R of S , and to compute the random covering $F_{SR}(T)$.

Results

The method described in the previous section was applied to different sets of tRNAs, fragments of 16S rRNA and tmRNA.

tRNA

First we present the results of an experiment concerning a set S_{130} of 150 sequences, each containing a tRNA corresponding to 80 nucleotides, plus 25 nucleotides located at the beginning and 25 nucleotides located at the end of each sequence.

[4] Note that Ns(T) ≥ q (because of the constraint of repetition).

Starting from S_{130}, we have created different groups of sequences. In each group S_j, the sequences are those of S_{130}, however part of the nucleotides at the beginning and at the end of each sequence has been deleted, thus leaving *j* nucleotides. In the experiment *j* belongs to{120, 110, 100, 90, 80} (e.g. S_{80} only contains nucleotides corresponding to the tRNA). In all cases we have selected ***g***=200 virtual secondary structures for each sequence.

For each group we present the results concerning three repeated structural patterns (among about fifty ones). In Figure 7 we see that the first one, coded **6 0 0 0,** corresponds to the tRNA structure, the second one, coded **1 4 0 0,** has a biological covering close to that of the repeated structural pattern **6 0 0 0**, and is somewhat close to this structure, the third one, coded **4 4 2 0,** is a structural pattern corresponding to a high random covering.

Figure 7

This experiment results in several conclusions:

1) the noise level increases with the length of the sequences

Figure 8 shows that, as expected, the random covering of any structural pattern increases with the length of the sequences. This is the case for the repeated structural patterns **6 0 0 0** and **4 4 2 0** (up to 90% on 130-length sequences).

Figure 8

2)No relevant repeated structural pattern displays a biological covering close to the random one

Figure 9 shows that the biological covering curve of the repeated structural pattern **4 4 2 0** is very close to the random one, and so its plausibility is low. On the other hand, the random and biological covering curves of the repeated structural pattern **6 0 0 0** look very different. This suggest that the plausibility measure allows to distinguish relevant structural pattern(s). Indeed this was the case in all experiments we have done.

Figure 9

3) The repeated structural pattern corresponding to the tRNA is the most plausible one

We see in Figure 10 that the repeated structural pattern **6 0 0 0** has the highest plausibility (among fifty or so repeated structural patterns). Note that the plausibility of the repeated structural pattern **1 4 0 0** is close to that of the repeated structural pattern **6 0 0 0**. On the contrary, the score of the repeated structural pattern **4 4 2 0** is low because of its important random covering. One can observe that in this experiment the plausibility decreases as the length of the sequences increases[5]. Furthermore, we have localised, on each sequence, the sets of palindromes that correspond to occurrences of the repeated structural pattern **6 0 0 0** (see Figure 11). There can be several occurrences of a given repeated structural pattern within a sequence and we note that the actual occurrence, in this example, had one of the best free energy. These results have been confirmed in all other experiments concerning several sets of bacterial tRNAs.

Figure 10
Figure 11

16S rRNA

We also made experiments on two sets of fragments of 16S rRNA sequences chosen as examples by (Kim et al., 1996). The first set contained 15 aligned sequences of approximate length 55, the second one 10 aligned sequences of approximate length 100. As our method does not require any alignment we have unaligned the sequences discarding the gaps. As shown in Figure 12 the theoretical consensus structural pattern of the first set is coded by the word **4 0 0**, and the theoretical consensus structural pattern of the second set is coded by the word **6 0 0 0**.

Figure 12

In our experiment the repeated structural pattern coded **4 0 0** (respectively **6 0 0 0**) has the highest plausibility among the repeated structural patterns. Figure 13 shows the occurrences of these repeated structural patterns on some sequences of the two sets.

Figure 13

10Sa RNA (tmRNA)

Our last experiment concerns 35 aligned sequences of 10Sa RNA (also known as tmRNA) taken from the tmRNA website (Williams, 1999).
We have discarded the gaps and focussed on the fragments containing the secondary structure showed figure 14. This structure consists in two pseudoknots (involving palindromes ***p8***, ***p9***, ***p10***, ***p11***) corresponds to a repeated structural pattern coded **1 0 1 0**.

Figure 14

Again we found **1 0 1 0** as the repeated structural pattern with highest plausibility among 44 repeated structural patterns occurring in at least 80% of the sequences. We can see Figure 15 the occurrences of this pattern in a sequence.

---

[5] in these experiments the nucleotides involved in the tRNA structure are included in the central 80-length sequences, whereas additional nucleotides add noise. As a result the random covering of 6000 increases when sequences grow longer.

Perspectives

The method presented here only relies on three binary structural relations between helices, namely inclusion, overlap and juxtaposition. In other words we only use here topological information whereas consensus secondary structures should include numerical characteristics regarding length of heads, tails and loops as well as pairing constraints and lexical patterns to be matched on the primary sequence. Still using such limited information not only is very efficient regarding CPU-time and space complexity, but also demonstrates to be surprisingly efficient regarding the results, i.e. the proposal of consensus secondary structures. However a major drawback of the algorithm presented here is that only rather small sequences and, as a consequence, small secondary structures can be addressed. The technical reason of this limitation is the exponential complexity of the branch and bound algorithm used during the first step to predict possible secondary structures of each sequence. However a more fundamental reason is that as sequences grow longer, using only relational information results in poor signal to noise ratio for short structures. More precisely, as sequences grow longer and as the number of possible secondary structures for each sequence increase, most short structures are found during the second step, thus mixing unexpected similarities (that should induce consensus proposals) and random similarities. This last effect could be avoided by adding non relational information when describing palindromes. However in this case similarities could no more be reduced to identity: we should add some flexibility (e.g. 7-length heads should be considered as similar to 6-length heads) at the cost of loosing the linear complexity of the second step that builds the repeated structural patterns. From a technical point of view this last modification relies on previous works extending the KMR algorithm to compute flexible lexical similarities (Soldano et al., 1995 ; Sagot et al., 1995). Still we would be limited by the first step of the method. When considering the high number of possible structures, it can be reduced when a priori knowledge is available. Anyway when working on long sequences the best strategy probably is to first divide sequences in short fragments, and then to search for consensus secondary structures on fragments. Note that this suppose some kind of rough multiple alignment between primary sequences, in order to determine the corresponding fragments in each sequence.

To conclude a future work concerns a one-step method mainly based on the following observation: the search of the g lowest free-energy structures of each sequence is not necessary for structures whose corresponding structural patterns do not occur in at least q sequences. We hope it will help addressing long sequences with low time and space complexity, as the first step computations will be partially avoided.

Acknowledgements

We thank very much Alain Viari and Bernard Billoud for useful discussion and assistance, Edouardo Rocha, Vincent Escalier and Andreij Konopka for comments on a draft of this paper, and the anonymous reviewers for their excellent comments and suggestions.

References

Billoud, B., Kontic, M. and Viari, A. (1996) Palingol: a declarative programming language to describe nucleic acids' secondary structures and to scan sequence databases, *Nucleic Acids Research,* **24**(8), 1395-1403.

Bouthinon, D., Soldano, H. and Billoud, B. (1996a) , Apprentissage d'un concept commun à un ensemble d'objets dont la description est hypothétique : application à la découverte de structures secondaires d'ARN, *Journées Francophones d'Apprentissage (JFA '96),* 8-10 Mai 1996, Sètes (France), pp. 206-220.

Bouthinon, D. (1996b), Apprentissage à partir d'exemples ambigus : étude théorique et application à la découverte de structures communes à un ensemble de séquences d'ARN, Doctorat de l'université Paris XIII-Institut Galilée.

Bouthinon, D. and Soldano, H. (1998a) , Un cadre de programmation logique inductive pour apprendre un concept à partir d'exemples ambigus, *11ème Congrès Reconnaissance des Formes et Intelligence Artificielle (RFIA '98)*, 20-22 Janvier 1998, Clermont-Ferrand (France), pp. 137-147.

Bouthinon D., Soldano H. (1998b), An Inductive Logic Programming Framework to learn a Concept from Ambiguous Examples, 10th European Conference on Machine Learning, Chemintz, Germany, April 1998, *Lecture Notes in Artificial Intelligence*, **1398**, pp. 238-249.

Cary, R. B. and Stormo, G. D. (1995) Graph-Theoretic Approach to RNA Modeling Using Comparative Data, *Proceedings of the third International Conference on Intelligent Sytems for Molecular Biology,* AAAI Press, Menlo Park, CA, pp. 75-80.

Chiu, D. K. Y. and Kolodziejczak, T. (1991) Inferring consensus structure from nucleic acid sequences, *Computer Applications in the Biosciences*, **7**, 347-352.

Dumas, J.-P. and Ninio, J. (1982) Efficient algorithms for folding and comparing nucleic acid sequences, *Nucleic Acids Research*, 10, 197-206.

Eddy, S.R. and Durbin R. (1994) RNA sequence analysis using covariance modeles, *Nucleic Acids Research*, Vol 22, 11, 2079-2088.

Gouy, M., Marliere, P., Papanicolaou, C. and Ninio, J. (1985) Prediction des structures secondaires dans les acides nucléiques: aspects algorithmiques et physiques, *Biochimie*, **67**, 523-531.

Gouy, M. (1987) In Bishop, M. J., and Rawlings, C. J. (eds.), Nucleic acid and protein sequence analysis a practical approach. IRL Press, Oxford.

Grate, L. (1995) Automatic RNA Secondary Structure Determination with Stochastic Context-Free Grammars, *Intelligent Systems for Molecular Biology*, 136-144.

Gutell, R. R., Noller, H. F. and Woese, C. R. (1986) Higher order structure in ribosomal RNA, *Embo J.*, **5**, 1111-1113.

Gutell, R. R., Power, A., Hertz, G. Z., Putz, E. J. and Stormo, G. D. (1992) Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods, *Nucleic Acids Res.*, **20**, 5785-5795.

Han, K. and Kim, H.-J. (1993) Prediction of Common folding structures of homologous RNA's, *Nucleic Acids Res.,* **21**, 1251-1257.

Haselman, T., Gutell, R. R., Jurka, J. and Fox, G. E. (1989) Additional watson-crick interactions suggest a structureal core in large subunit ribosomal RNA, *J. Biomol. Struct. Dynam.*, **7**, 181-186.

Karp, R. M., Miller, R. E. and Rosenberg, A. L. (1972) , *Proc. 4th Annu. ACM Symp. Theory of Computing*, pp. 125-136.

Kim, J., Cole, J. R. and Prmanik, S. (1 996) Alignment of possible secondary structures in multiple RNA sequences using smulated annealing, *CABIOS*, **12**(4), 259-267.

Lefebvre, F. (1995) , *Third International Conference on Intelligent Systems for Molecular Biology,* July 16-19, 1995, Robinson College, Cambridge England, pp. 222-230.

Lefebvre, F. (1996) , *Fourth International Conference on Intelligent Systems for Molecular Biology,* June 12-15, 1996, Whashington University, St Louis Missouri, USA.

Martinez,H.M. (1983) An efficient method for finding repeats in molecular sequences, *Nucl. Acids Res.*, **11**, n° 13, 4629-4634.

Martinez,H.M. (1988) An RNA secondary structure workbench, *Nucl. Acids Res.*, **16**, n° 5.

Nakaya, A., Yamamoto, K. and Yonesawa, A. (1995) RNA secondary structure prediction using highly parallel computers, *Comput. Applic. Biosci.*, **11**, 685-692.

Ninio, J. (1979) Prediction of pairing schemes in RNA molecule-loop contributions and energy of wobble and non-wobble pairs, *Biochimie*, **61**, 1133-1150.

Noller, H. F. and Woese, C. R. (1981) Secondary Structure of 16S ribosomal RNA, *Science*, **212**, 403-411.

Nussinov, R., Pieczenik, G., Griggs, J. R. and Kleitman, D. J. (1978) Algorithm for loop matchings, *SIAM J. Appl. Math.*, **35**, 68-82.

Nussinov, R. and Jacobson, A. B. (1980) Fast algorithm for predicting the secondary structure of single- stranded RNA, *Proceedings of the National Academy of Science USA,* **77**, 6309-6313.

Olsen, G. (1983) . University of Colorado.

Papanicolaou, C., Gouy, M. and Ninio, J. (1984) An energy model that predicts the correct folding of both the tRNA and the 5S RNA molecules, *Nucleic Acids Res.,* **12**, 31-44.

Pipas, J. M. and McMahon, J. E. (1975) Method for predicting RNA secondary structure, *Proceedings of the National Academy of Science USA*, **72,** 2017-2021.

Sagot, M. F., Viari, A., Pothier, J. and Soldano, H. (1995) Finding flexible patterns in a text - An application to 3D molecular matching., *CABIOS,* **11**(1), 59-70.

Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjoelander, K., Underwood, R. and Haussler, D. (1994) Stochastic context-free grammars for tRNA modelling, *Nucleic Acids Research,* **22**, 5112-5120.

Salser, W. (1977) Globin mRNA sequences: analysis of base pairing and evolutionary implications, *Cold Spring Harbor Symp. Quant. Biol.,* **42**, 985-1002.

Sankoff, D., Kruskal, J. B., Mainville, S. and Cedergren, R. J. (1983) , Time warps, string edits, and macromolecules: the theory and practice of sequence comparison, Sankoff D., Kruskal J.B., Eds., Addison-Wesley, Reading, pp. 93-120.

Shapiro,B.A. and Wu,J.C. (1997), Predicting RNA H-type pseudoknots with massively paralel genetic algorithm, *Comput. Applic. Biosci.*, **13**, 459-471.

Soldano,H., Viari,A., Champesme,M. (1995) Searching for flexible repeated patterns using a non-transitive similarity relation, *Pattern Recognition Letters*, **16**, 243-246.

Stern, S., Weiser, B. and Noller, H. F. (1988) A model for the three-dimensional folding of 16S ribosomal RNA, *Journal of Molecular Biology,* 204, 447-481.

Rivas E. and Eddy S.R. (1999) A Dynamic Programming Algorithm for RNA Structure Prediction including Pseudoknots, *Journal of Molecular Biology*, **285**, 2053-2068.

Tabaska J.E. and Stormo G.D. (1997) Automated Alignment of RNA Sequences to Pseudoknotted Structures, *Fifth International Conference on Intelligent Systems for Molecular Biology*, The AAAI Press Menlo Park, California (USA), 311-318.

Tabaska, J.E., Cary R.B., Gabow H. N. and Stormo G.D. (1998) an RNA folding method capable of identifying pseudoknots and base triples, *BioInformatics*, **14**, n° 8, pp. 691-699.

Waterman, M. S. and Smith, T. F. (1978) RNA secondary structure: a complete mathematical analysis, *Math. Biosci.,* **42**, 257-266.

Williams, K. (1999), The tmRNA website, *Nucleic Acids Res.,* **27**, 165-166.

Woese, C. R. and Gutell, R. R. (1989) Evidence for several higher-order structural elements in ribosomal RNA, *Proc. Natl. Acad. Sci. USA 86,* 3119-3122.

Wuju, Li., and JiaJin W. (1998) Prediction of RNA secondary structure based on helical regions distribution, Bioinformatics, **14**, n° 8, pp 700-706.

Zuker, M. and Sankoff, D. (1984) RNA secondary structures and their prediction, *Bulletin of Mathematical Biology,* **46**, 591-621.

Zuker, M. (1986) RNA folding prediction : The continued need for interaction between biologists and mathematicians, *Lecture Mathematics Life Sci.,* **17**, 87-124.

Zuker, M. and Stiegler, P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information, *Nucleic Acids Res.,* **9**, 133-148.

Zuker, M. (1989) Computer prediction of RNA structure, Methods in Enzymology, 180, 262-288.

Zuker, M. (1989) , Mathematical methods for DNA sequences, *Waterman M.S., Ed., CRC Press,* Boca Raton, pp. 159-184.

Zuker, M., Jaeger, J. A. and Turner, D. H. (1991) A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison, *Nucleic Acids Res.,* **19**, 2707-2714.

Zuker, M. (1991) Suboptimal sequence alignment in molecular biology. Alignment with error analysis, *Journal of Molecular Biology,* **221**, 403-420.