RNA Locally Optimal Secondary Structures

Azadeh Saffarian, Mathieu Giraud, Antoine De Monte, Hélène Touzet

RNA Locally Optimal Secondary Structures

Azadeh Saffarian Mathieu Giraud Antoine de Monte Hélène Touzet*
LIFL, UMR CNRS 8022, Université Lille 1 and Inria Lille, France

Authors' Preprint version. The final version is published in:
Azadeh Saffarian, Mathieu Giraud, Antoine de Monte, and Hélène Touzet, *RNA Locally Optimal Secondary Structures*, Journal of Computational Biology, 19(10):1120-1133, 2012, doi:10.1089/cmb.2010.0178.

Abstract

RNA locally optimal secondary structures provide a concise and exhaustive description of all possible secondary structures of a given RNA sequence, and hence a very good representation of the RNA folding space. In this paper, we present an efficient algorithm which computes all locally optimal secondary structures for any folding model that takes into account the stability of helical regions. This algorithm is implemented in a software called regliss that runs on a publicly accessible web server: `http://bioinfo.lifl.fr/RNA/regliss`.

1 Introduction

Noncoding RNAs (ncRNAs) play a wide range of roles in the cell (Mattick & Makunin, 2006), and for many of them, the function is mainly determined by their three dimensional structure. A single stranded RNA folds upon itself to form base pairs, which shape the secondary structure of the molecule, and act as a scaffold for the three dimensional folding. Knowing the secondary structures is thus of critical importance to understand the function of an RNA molecule. The earliest algorithms for the RNA folding problem appeared in the late 70's, see (Eddy, 2004) for a survey. Such algorithms take as input a single stranded RNA sequence and output a single secondary structure possessing *minimum free energy*.

However, it became clear early on that computing the single minimum free energy folding is not enough. For a number of reasons, the biologically correct structure is often not the optimal one, but rather a structure within a small percentage deviation of the minimum free energy. Firstly, slight changes in the thermodynamic model may produce very different foldings with a similar energy level. Secondly, the thermodynamic model does not allow for pseudoknots or base triplets, and it does not reflect the interactions of the RNA with other molecules in the cell. Thirdly, some biological processes involve switches, changes of conformation in RNA structures.

All of these reasons make it important to be able to predict multiple foldings, also called *suboptimal* foldings, that allow for a deeper insight at the RNA folding space. Moreover, being able to compute alternative structures may also be useful in designing RNA sequences which not only have low folding energy, but whose folding landscape would suggest rapid and robust folding.

Several programs produce suboptimal foldings of RNA, including *mfold/unafold* (Zuker, 1989), *RNAsubopt* (Wuchty *et al.*, 1999) and *RNAshapes* (Steffen *et al.*, 2006).

*Corresponding author: `helene.touzet@lifl.fr`

Unfortunately, as we will see in Section 2, none of these tools are fully suitable for an exhaustive enumeration of all possible secondary structures. To address this problem, Clote introduced the concept of *locally optimal secondary structures* (Clote, 2005a). A secondary structure is locally optimal if no base pairs can be added without creating a conflict, such as introducing a pseudoknot or a base triplet. The set of locally optimal secondary structures can be seen as a concise description of the space of all secondary structures, because each secondary structure is included in a locally optimal secondary structure. Clote proposed a dynamic programming algorithm to enumerate such structures. One drawback of this approach is that it uses the Nussinov-Jacobson model (Nussinov & Jacobson, 1980), which does not produce realistic secondary structures. The problem of locally optimal secondary structures with an accurate folding model has been recently addressed in (Lorenz & Clote, 2011). The authors present an algorithm to compute the partition function over all locally optimal secondary structures of a given RNA sequence, extending the McCaskill's classical algorithm (J., 1990). This method however does not effectively produce the set of locally optimal secondary structures.

In this paper, we introduce a novel approach to generate all locally optimal secondary structures assembled from a set of thermodynamically stable helices. We propose an efficient algorithm for this problem, which relies on decomposition of secondary structures into structures *maximal for juxtaposition*. As far as we know, this property has never been formulated or used to study locally optimal secondary structures. The paper is organized as follows. Section 2 presents some background information on suboptimal and locally optimal secondary structures. Section 3 details our folding algorithms for the locally optimal structures. For pedagogical reasons, we first expose the main outlines of the algorithm for the simplistic Nussinov-Jacobson model (Section 3.1). We then explain how to adapt it to deal with thermodynamically stable helices (Section 3.2). Section 4 discusses the implementation. Finally, Section 5 presents some experimental results. All proofs are available in Supplementary Material.

2 Background

We give a brief overview of the main suboptimal and locally optimal RNA folding methods.

mfold/unafold (Zuker, 1989). The algorithm returns a sample set of the foldings by considering all possible base pairs and by computing the best folding that contains this base pair. The *suboptimality level* option further selects the suboptimal candidates to return only those within a given free energy range. The result is that not all possible structures need to be computed, which speeds up computational time. As a counterpart, even with 100% suboptimality level, the algorithm does not provide all possible suboptimal secondary structures. The number of calculated structures is intrinsically bounded by the number of possible base pairs, whatever the suboptimality percentage is: it is quadratic in the length of the input sequence. By construction, secondary structures that contain at least "two different places" of suboptimality are not provided by the algorithm (Figure 1, top). Another consequence is that the algorithm can output secondary structures that contain another secondary structure with a better free energy (Figure 1, bottom).

RNAsubopt (Wuchty *et al.*, 1999). Another possibility to produce suboptimal structures is to modify the standard folding algorithm in order to output *all* secondary structures within a given energy range above the minimum free energy. However, if the threshold is set too low, not much variation is possible, and if it is set too high, too many structures may be generated for the reasonable evaluation. For example, the toy sequence of Figure 1 provides 4 structures within the energy range 10%, and 177 structures within the energy range 30%. The number of structures returned grows exponentially with both sequence length and energy range, and many structures are very similar.

RNAshapes (Steffen *et al.*, 2006) organizes the suboptimal foldings to explore the folding space into classes of *abstract shapes* and reduces the potential exponential number of structures to a few classes. But two secondary structures with no common base pairs can be classified in the same shape.

Locally optimal secondary structures. The critical evaluation of these software programs suggests that there is a need of formal definitions for suboptimal secondary structures, that would correspond to local min-

```
     ACACAAAAGUGUGAAAAACACACAAAAGUGUAAAAUGUGAAACACACAAAAAGUGUGAAACACA
1    ((((....(((((.....)))))....))))....((((...(((((.....)))))...))))   (-17.20)
2    .((((....))))....(((.......))))....((((...(((((.....)))))...))))   (-14.40)
3    ((((....(((((.....)))))....))))....((((.....))))....(((.....))).   (-13.50)
4    ........(((((.....(((.......))))...........(((((.....)))))..))))).   (-13.50)
5    ((((....(((((.....)))))............(((((.....))))).....))))........   (-12.50)
6    ........(((((.....)))))....((((...........(((((.....)))))..))))).   (-12.42)
7    ((((....(((((.....)))))....((((...........))))........))))........   (-12.40)
8    .(((....(((((.....)))))....((.....)))))...(((((.....)))))........   (-12.10)
9    ........(((((.....)))))....((.....))(((...(((((.....)))))...))).   (-11.50)
10   ........((((........((((...........))))...(((((.....)))))..))))).   (-12.10)
11   .((((...((((((....((((((....))))).....))....)))))).......))))....   (-11.30)
12   ........((((......((((((....((((............))))).....))))))..))))).   (-10.30)
13   ........((.......(((((......))))...))(((...(((((.....)))))...))).   (-7.50)
⋆    .((((....))))....(((.......))))....((((.....))))....(((.....))).   (-10.80)


  CACACAUAGGAACCUCCACUAAGGAUUCUAUGGACAGUCGAUGCAGGGAGUUCACAGCUCCCUGCAUCGGCGAUUUU
1 ....(((((((.(((......))).)))))))....((((((((((((((.....))))))))))))))).....  (-40.4)
2 ....(((((((.(((......))).)))))))((..((((((((((((((.....))))))))))))))..))..  (-37.3)
```

Figure 1: (Top) *unafold* output on a toy sequence (64 nt) with 100% suboptimality. This software produces 13 suboptimal secondary structures, displayed in Vienna bracket-dot format, whose free energy ranges from -17.20 kcal/mol to -7.50 kcal/mol. It misses the structure $\star$ of free energy -10.80 kcal/mol, composed of four stem-loops. Each of these stem-loops has been identified by *unafold* (in structures $\#2$ and $\#3$), but the algorithm is not able to recover the four stem-loops in a same structure. (Bottom) *unafold* output on sequence AY545598.5 (37939-38015), RF00107 (77 nt). Structure #2 contains structure #1, and has a higher free energy level. Nevertheless, it is selected in the space of suboptimal structures, because it is the optimal structure containing base pair $(33, 75)$.

ima in the free energy landscape. The notions of *saturated* structures and *locally optimal* secondary structures meet this requirement. In (Zuker & Sankoff, 1984) and (Evers & Giegerich, 2001), a secondary structure is *saturated* when the stacking regions are extended maximally in both directions: No base pairs can be added at the extremity of a stacking region without degrading the free energy. Moreover, there is no isolated base pair. In (Clote, 2005a), a secondary structure is *locally optimal* when no base pairs can be added without creating a conflict: either crossing pairings, or a base triplet. The main drawback of Clote's result is that the algorithm relies on the Nussinov-Jacobson folding model. As a consequence, the number of locally optimal secondary structures is very large, and many of them are not thermodynamically stable in the Nearest Neighbor model. For example, the toy sequence of Figure 1 produces 1107 optimal secondary structures having 18 base pairs, 197,501 locally optimal secondary structures having 17 base pairs, and more than 6 millions of locally optimal secondary structures having 16 base pairs. The work that we present here is inspired by this research. We start with the nice topological definition of locally optimal secondary structures, and extend it to take into account the stability of helical regions. In this context, locally optimal secondary structures are also saturated structures.

3 Algorithms

3.1 Folding at base pair resolution

We begin by considering locally optimal secondary structures for a simple model: All base pairs are independent, like in the Nussinov-Jacobson model. This model is mainly interesting for pedagogical purposes, because it allows us to provide basic definitions and ideas. We shall explain in Section 3.2 how to extend this to energetically stable helices to take into account interactions between adjacent base pairs.

3.1.1 Definitions

Let α be an RNA sequence of length n over the alphabet $\{A, C, G, U\}$: $\alpha = \alpha_1\alpha_2 \dots \alpha_n$. A *base pair* (x, y) on α is an ordered pair of natural numbers such that $1 \leq x < y \leq n$. Base pairs are sorted according to the lexicographical order on their positions on the sequence: (x, y) is smaller than (z, t) if $x < z$, or if $x = z$ and $y < t$. Given a set BP of base pairs on α, a *structure* is any subset of BP. We denote the empty structure by ε. A *secondary structure* on BP is a subset of BP such that any two distinct base pairs of S are either *nested* or *juxtaposed*:

- (x, y) is nested in (z, t) if $z < x < y < t$,
- (x, y) is juxtaposed with (z, t) if $z < t < x < y$.

The base pairs (x, y) and (z, t) are nested, if (x, y) is nested in (z, t) or (z, t) is nested in (x, y). The base pairs (x, y) and (z, t) are juxtaposed, if (x, y) is juxtaposed with (z, t) or (z, t) is juxtaposed with (x, y). Two base pairs that are neither nested nor juxtaposed are said to be *conflicting*.

Definition 1. *Let S and T be two structures on* BP*. S is* strictly included *in T, or T is a* strict extension *of S, if any base pair of S is present in T, and there exists a base pair of T that is not in S.*

Definition 2. *Let S be a secondary structure structure on* BP*. S is* locally optimal *if it satisfies the following condition: If T is a structure that is strict extension of S, then T is not a secondary structure.*

In other words, a secondary structure is locally optimal if no base pairs can be added without producing conflict. It follows that any secondary structure is included in a locally optimal structure. We give in Figure 2 an example of a set of base pairs and all its locally optimal secondary structures, that will serve as a running example throughout this paper.

The set of all locally optimal secondary structures is potentially very large: It can be exponential in ℓ, the number of base pairs in BP. The exact upper bound, $3^{\ell/3}$, can be calculated by rephrasing the problem in terms of Maximal Independent Sets. The set of vertices is BP and an edge links two conflicting base pairs. The locally optimal structures are exactly the Maximal Independent Sets of the graph (see Figure 1 in Supplementary Materials).

The idea of our algorithm is to reduce the combinatorics by taking advantage of properties of nested and juxtaposed relations, such as transitivity, to achieve a good running time in practice. We divide the construction of locally optimal structures into two steps: First applying only juxtaposition operations, then applying only nesting operations. Structures are thus decomposed into horizontal levels of juxtaposed base pairs. For that we need two more notations. Given a structure S, $\mathsf{Toplevel}(S)$ is defined as the set of base pairs of S that are not nested in any base pair of S. Given a base pair (x, y) in S, $\mathsf{Nested}(x, y, S)$ is the set of base pairs of S that are nested in (x, y) and that are not nested in any base pair nested in (x, y). These levels induce a partition of S: $S = \mathsf{Toplevel}(S) \cup \bigcup_{(x,y)\in S} \mathsf{Nested}(x, y, S)$

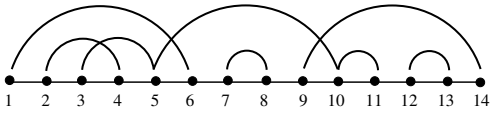
It is routine to verify that S is a secondary structure if, and only if, any two base pairs of $\mathsf{Toplevel}(S)$ are juxtaposed, and for each (x, y) of S, any two base pairs of $\mathsf{Nested}(x, y, S)$ are juxtaposed. One important result for our algorithm is that the property for a secondary structure to be locally optimal can be testified by looking only at the Toplevel and Nested subsets. We will show that these subsets must be *maximal for juxtaposition*.

Definition 3. *Let S be a structure on* BP*. S is* maximal for juxtaposition *if it satisfies the two following conditions:*

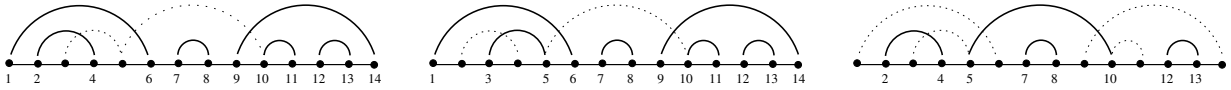
(i) if b and b' are two distinct base pairs in S, then b and b' are juxtaposed,
(ii) if b is a base pair of BP *not present in S such that $\{b\} \cup S$ is a secondary structure, then b is nested in some base pair of S.*

Figure 2-(c) gives examples of structures maximal for juxtaposition. The link between structures maximal for juxtaposition and locally optimal secondary structures is established by Theorem 5.

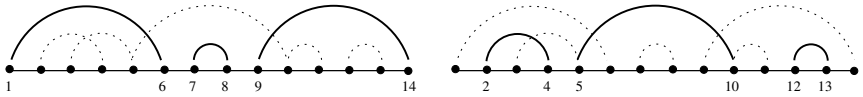
(a) Initial set of base pairs



(b) Locally optimal secondary structures



(c) Structures maximal for juxtaposition



(d) From structures maximal for juxtaposition to locally optimal secondary structures

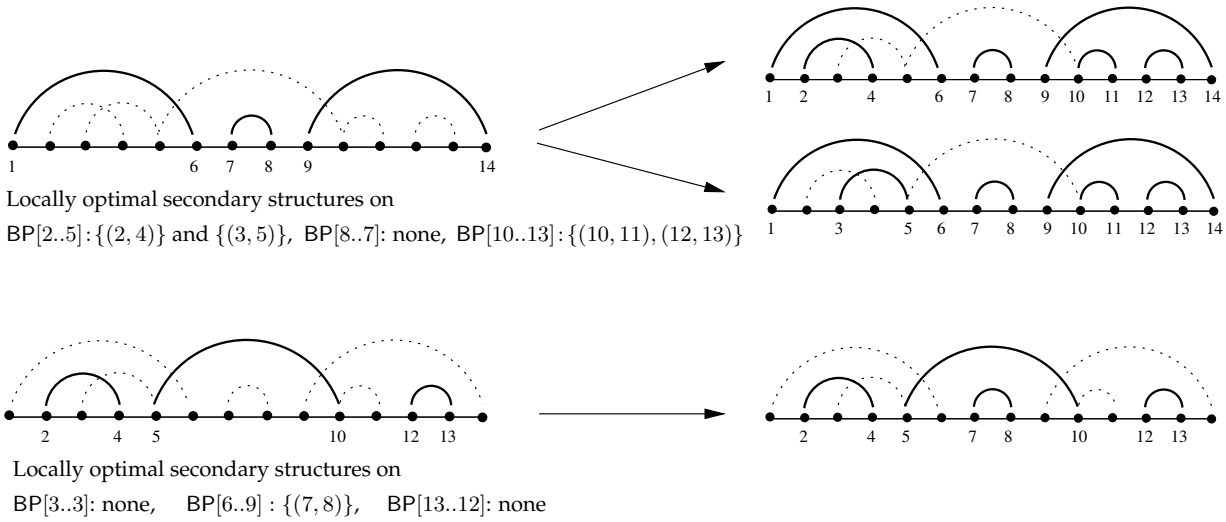


Figure 2: Example and construction of locally optimal secondary structures. (a) The sequence has 8 base pairs. (b) This set of base pairs contains three locally optimal secondary structures: $\{(1,6),(2,4),(7,8),(9,14),(10,11),(12,13)\}$, $\{(1,6),(3,5),(7,8),(9,14),(10,11),(12,13)\}$ and $\{(2,4),(5,10),(7,8),(12,13)\}$. (c) It can form two structures maximal for juxtaposition, $\{(1,6),(7,8),(9,14)\}$ and $\{(2,4),(5,10),(12,13)\}$. (d) The first structure maximal for juxtaposition extends to the two first locally optimal secondary structures. The second one extends to a single locally optimal secondary structure.

Theorem 1. *A structure S on* BP *is a locally optimal secondary structure if, and only if,*

(i) $\mathsf{Toplevel}(S)$ *is maximal for juxtaposition on* $\mathsf{BP}[1..n]$,
(ii) for each base pair (x, y) *of* S, $\mathsf{Nested}(x, y, S)$ *is maximal for juxtaposition in* $\mathsf{BP}[x + 1..y - 1]$

$\mathsf{BP}[x..y]$ denotes the subset of BP composed of base pairs (z, t) such that $x \leq z < t \leq y$. The proof of Theorem 5 is given in Supplementary materials. Figure 2-(d) gives an illustration of the Theorem.

3.1.2 Construction of structures maximal for juxtaposition

We show how to efficiently construct the structures maximal for juxtaposition. For each pair of positions i and j of α, we define the set of secondary structures $\mathsf{MJ}(i, j)$ as follows.

1. If $i \geq j$, then $\mathsf{MJ}(i, j) = \{\varepsilon\}$

2. otherwise, if there is no base pair (i, y), $i < y \leq j$, in BP, then $\mathsf{MJ}(i, j) = \mathsf{MJ}(i + 1, j)$.

3. otherwise

$$\mathsf{MJ}(i,j) = \bigcup \begin{cases} \bigcup_{(i,y)\in \mathsf{BP}[i..j]} \{(i,y)\} \oplus \mathsf{MJ}(y+1,j) & (a) \\ \bigcap_{(i,y)\in \mathsf{BP}[i..j]} \mathit{Filter}((i,y), \mathsf{MJ}(i+1,j)) & (b) \end{cases}$$

The operator $\oplus$ denotes the concatenation of a base pair to a set of structures: S is in $\{(i, y)\} \oplus \mathsf{MJ}(y+1, j)$ if, and only if, there exists S' in $\mathsf{MJ}(y + 1, j)$ such that $S = \{(i, y)\} \cup S'$. In rule (3b), a Filter function is used to check the maximality of structures. It is defined as follows: Given a base pair b, and a set of secondary structures R, the secondary structure S of R is in $\mathit{Filter}(b, \mathsf{R})$ if, and only if, there exists a base pair b' in S such that b and b' are conflicting. We have the following Theorem.

Theorem 2. *Let i and j be two positions on α.* $\mathsf{MJ}(i, j)$ *is exactly the set of all structures maximal for juxtaposition on* $\mathsf{BP}[i..j]$.

The question is now how to implement the formula to compute $\mathsf{MJ}(i, j)$. The recurrence relation naturally suggests to use dynamic programming with a two dimensional table, indexed by i and j. This can be further refined. A close inspection at Theorem 5 shows that not all pairs of positions i and j are useful for the computation of locally optimal secondary structures: We only need $\mathsf{MJ}(x + 1, y - 1)$ for all base pairs (x, y) of BP, and intermediate values necessary to obtain $\mathsf{MJ}(x + 1, y - 1)$. So we should only consider pairs of positions of the form $(k, y - 1)$ with $x < k < y$ and (x, y) in BP. The last point that we want to make here is that in the rule (3b), the computation of $\mathit{Filter}((x, y), \mathsf{MJ}(i, j))$ requires at most $O(y - x)$ tests for every structure $S \in \mathsf{MJ}(i, j)$. Indeed, given a structure S in $\mathsf{MJ}(i, j)$, let b be the first base pair of S not nested in (x, y). S belongs to $\mathit{Filter}((x, y), \mathsf{MJ}(i, j))$ if, and only if, such a b exists and is conflicting with (x, y).

3.1.3 Construction of locally optimal secondary structures

We now explain how to compute the set of locally optimal secondary structures from the set of structures maximal for juxtaposition. The stepping stone is Theorem 5, stated in Section 3.1.1. This result allows us to view the set of locally optimal secondary structures as the set of ordered rooted tree whose vertices are labeled by structures maximal for juxtaposition. More precisely, each tree is such that:

- The root is labeled by an element of $\mathsf{MJ}(1, n)$,
- Each node is labeled by an element w of $\mathsf{MJ}(x + 1, y - 1)$ for some base pair (x, y) of BP,
- The out-degree of a node labeled by w is the number of base pairs of w,
- The ith child of a node labeled by w is labeled by an element of $\mathsf{MJ}(x + 1, y - 1)$, where (x, y) is the ith base pair of w.

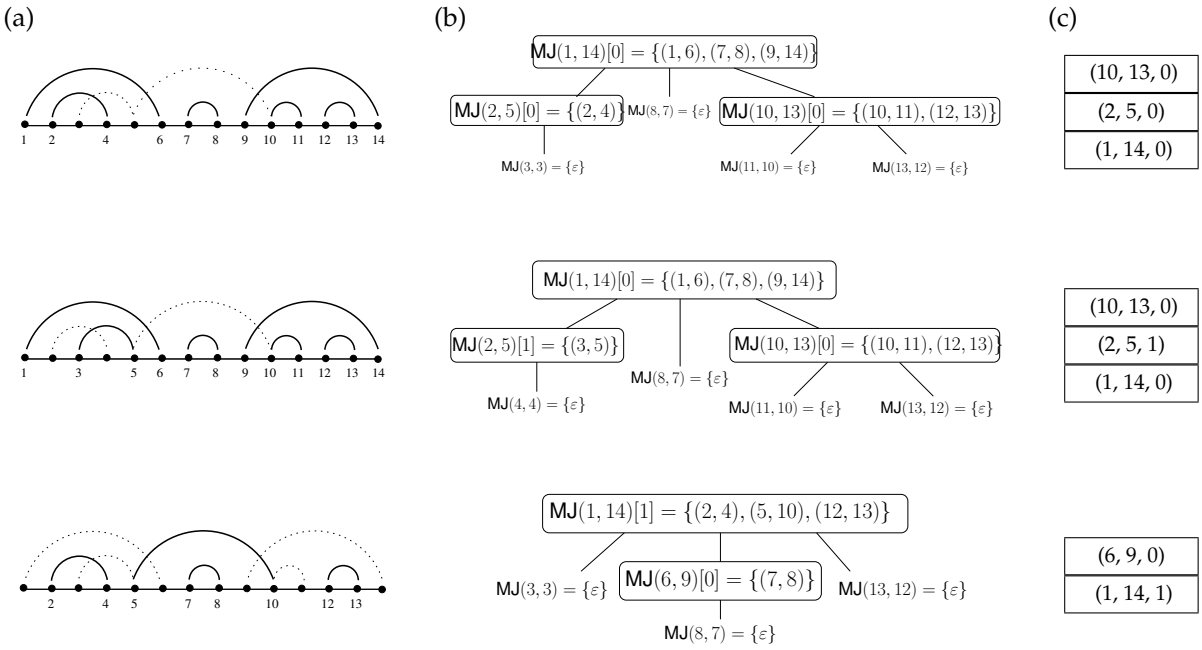


Figure 3: (a) Locally optimal secondary structures of Figure 2. (b) Associated ordered trees. $\mathsf{MJ}(i,j)[k]$ denotes the kth element of $\mathsf{MJ}(i,j)$. (c) Content of the stack at the end of each iteration of the algorithm of Figure 4 (lines 5, 6 and 7). $\mathsf{MJ}(i,j)[k]$ is symbolized by the triplet (i,j,k). Each cell of the stack corresponds to an internal node of the underlying tree depicted in (b). For example, at iteration 1, the line 5 pushes the triplet (1, 14, 0), corresponding to $\mathsf{MJ}(1,14)[0] = \{(1,6),(7,8),(9,14)\}$, then the nested structures are pushed.

Figure 3-(c) gives the three possible trees for the locally optimal secondary structures of the example of Figure 2. This representation brings an effective way to compute all locally optimal secondary structures. The enumeration of all possible such trees can be done easily with a push-down stack whose elements are structures maximal for juxtaposition. The pseudo-code of the algorithm is given in Figure 4, and an example of run is given on Figure 3-(d). At each iteration of the algorithm, the stack contains a different locally optimal secondary structure. The height of the stack is bounded by ℓ', the maximal number of structures maximal for juxtaposition present in the locally optimal secondary structure. This value is much smaller than the number of base pairs of the output structure, and thus smaller than the total size of BP. Each iteration of the loop is then done in time $O(\ell')$. Subsequently, the construction of all locally secondary structures can be performed in time linear in the size of the output, that is the total number of base pairs of all locally optimal secondary structures.

3.1.4 Back to Clote's algorithm

In Section 2, we mentioned the seminal work of (Clote, 2005a) on counting locally optimal secondary structures for the Nussinov-Jacobson model. This work uses a clever optimization based on the notion of *visible* bases and *visible* positions. Given a secondary structure S, a *visible* position p in S is a position outside any base pair of S: $\forall (x,y) \in S, \quad p < x \ \vee \ y < p$. By extension, a character $c \in \{A,C,G,U\}$ is visible in S if there exists a visible position p such that $c = \alpha_p$. Let $\mathbf{v} \subset \{A,C,G,U\}$ be a subset of the alphabet, and let $Loc(i,j)[\mathbf{v}]$ be the set of locally optimal structures between positions i and j where the bases $\mathbf{v}$, and only these bases, are visible. Then $Loc(i,j)$, the set of all locally optimal structures between positions i and j, is the union of the different $Loc(i,j)[\mathbf{v}]$ for all $\mathbf{v} \subset \{A,C,G,U\}$.

Let now fix a given set of allowed pairings. For example, one can consider only Watson-Crick base pairs, taking for BP, the set of possible base pairs, the following WC set:

```
1  push(1, n, -1)
2  repeat
3        pop(i, j, k) until MJ(i,j)[k+1] exists or stack is empty
4        if the stack is empty, and no MJ(i,j)[k+1] have be found, then exit
5        push(i, j, k+1)
            \\ consider MJ(i,j)[k+1]
6        init-down(i, j, k+1)
            \\ recursively push on the stack a structure nested in MJ(i,j)[k+1]
            \\ (initialisation by choosing the first MJs)
7        init-right(i, j)
            \\ recursively push on the stack a structure juxtaposed to MJ(i,j)[k+1],
            \\ and compatible with the elements in the stack
            \\ (initialisation by choosing the first MJs)
8        output stack content
9  end repeat
```

Figure 4: Enumeration of all locally optimal secondary structures from the set of all structures maximal for juxtaposition. Each iteration of the loop outputs one locally optimal structure. $\mathsf{MJ}(i,j)[k]$ denotes the kth element of $\mathsf{MJ}(i,j)$.

$$\mathsf{WC} = \{(x,y) \mid 1 \leq x < y \leq n \text{ and } (\{\alpha_x, \alpha_y\} = \{A,U\} \text{ or } \{\alpha_x, \alpha_y\} = \{C,G\})\}$$

The sets of locally optimal structures can then be computed in a very efficient way:

$$Loc(i,j)[\mathbf{v}] = \bigcup \left\{ \begin{array}{l} \bigcup_{(i,y)\in \mathsf{WC}[i..j]} \{(i,y)\} \oplus Loc(i+1,y-1) \oplus Loc(y+1,j)[\mathbf{v}] \\ Loc(i+1,j)[\mathbf{v} - \{\overline{\alpha_i}\}] \cup Loc(i+1,j)[\mathbf{v} - \{\alpha_i, \overline{\alpha_i}\}] \end{array} \right.$$

In the second line, $\overline{\alpha_i}$ is the complementary base of α_i. As this base $\overline{\alpha_i}$ is never visible in the locally optimal structures in $Loc(i+1,j)[\mathbf{v} - \{\overline{\alpha_i}\}]$ and $Loc(i+1,j)[\mathbf{v} - \{\alpha_i, \overline{\alpha_i}\}]$, that guarantees that all such structures are also locally optimal on $\mathsf{WC}[i..j]$: no *Filter* function is further required.

It is possible to take advantage of this optimization and to combine it with our construction method through structures maximal for juxtaposition. We obtain the following recurrence relation for this construction:

$$\mathsf{MJ}(i,j)[\mathbf{v}] = \bigcup \left\{ \begin{array}{l} \bigcup_{(i,y)\in \mathsf{WC}[i..j]} \{(i,y)\} \oplus \mathsf{MJ}(y+1,j)[\mathbf{v}] \\ \mathsf{MJ}(i+1,j)[\mathbf{v} - \{\overline{\alpha_i}\}] \cup \mathsf{MJ}(i+1,j)[\mathbf{v} - \{\alpha_i, \overline{\alpha_i}\}] \end{array} \right.$$

The construction of locally optimal secondary structures from structures maximal for juxtaposition (as described in Theorem 5) is then unchanged. The same optimization can be adapted to some larger BP sets, including for example the wobble G–U pairs. However, the efficiency of the method relies on a set of fixed base pairs, independently of their positions: our algorithm allows far more flexibility, constructing locally optimal structures on any initial set of base pairs BP.

3.2 Folding at helix resolution

In this section, we extend the construction of locally optimal secondary structures to the framework of energetically favorable helices. This model is likely to produce more biologically realistic structures, because it takes into account the stacking energy between base pairs, such as introduced in the Nearest Neighbor model (Matthews *et al.*, 1999) for example.

3.2.1 Definitions

We admit a generic definition for *helices*. It is an ordered set of base pairs $\{(x_1,y_1),\ldots,(x_k,y_k)\}$ such that $x_1 < \ldots < x_k$, and $y_1 > \ldots > y_k$. It can contain bulges and internal loops. The *5′ arm* of the helix is the

set of positions $x_1, \ldots, x_k$, and the *3′ arm* is the set of positions $y_1, \ldots, y_k$. We denote by $f.5start$, $f.5end$, $f.3start$ and $f.3end$ the first position of the 5′ arm, the last position of the 5′ arm, the first position of the 3′ arm and the last position of the 3′ arm respectively. Given two distinct helices f and g, we define four different relations between f and g:

- g is *nested* in f if $f.5end < g.5start$ and $g.3end < f.3start$, (in this case, any base pair of g is nested in any base pair of f),
- g is *juxtaposed* with f if $f.3end < g.5start$, (in this case, any base pair of g is juxtaposed with any base pair of f),
- g is *embedded* in f if any base pair of g is also a base pair of f,
- otherwise, f and g are said to be *conflicting*,

The concepts of structures, secondary structures, strict inclusion (Definition 1) and locally optimal secondary structures (Definition 2) on base pairs can easily be adapted to helices:

- A *structure* is any subset of H. Given a structure S on BP, S is *described* by the structure $\{f_1, \ldots, f_k\}$ on H, if $S = f_1 \cup \cdots \cup f_k$.
- A *secondary structure* is any subset of H such that any two helices are either nested or juxtaposed.
- Given two structures $\{f_1, \ldots, f_k\}$ and $\{g_1, \ldots, g_j\}$ of H, we say that $\{f_1, \ldots, f_k\}$ is strictly included in $\{g_1, \ldots, g_j\}$ if the set of base pairs $f_1 \cup \cdots \cup f_k$ is strictly included in $g_1 \cup \cdots \cup g_j$.
- A secondary structure $\{f_1, \ldots, f_k\}$ of H is *locally optimal* if it satisfies the following condition: If $\{g_1, \ldots, g_j\}$ is a structure on H that is a strict extension of $\{f_1, \ldots, f_k\}$, then $\{g_1, \ldots, g_j\}$ is not a secondary structure on H.

From now on, we assume that we have a set H of helices of size ℓ, and we work with structures defined on H. We also assume that helices of H are ranked from 1 to ℓ according to a total *helix ordering* $<$, and that the order verifies $f < g \Rightarrow f.5start < g.5start$. Given two helices f and g in H, H$[f..g]$ denotes the subset of H composed of helices whose all base pairs are in the interval $[f.5start..g.3end]$, and H$]f..g[$ the subset of H composed of helices whose all base pairs are in the interval $]f.5end..g.3start[$. As before, for a structure F on H, Toplevel(F) is defined as the set of helices of F that are not nested in any helix of F and Nested(f, F) is the set of helices of F that are nested in the helix F, and that are not nested in any helix nested in f.

We now turn to the problem of constructing all locally optimal secondary structures for a set of helices. The two-step method described in Section 3.1 is still valid: First considering structures maximal for juxtaposition and constructing them by dynamic programming, then recovering locally optimal secondary structures on the fly with a push-down stack. However, the algorithm needs some adaptation to take into account the existence of embedded helices and the fact that some helices can combine to form other helices present in the input set.

3.2.2 Construction of structures maximal for juxtaposition

Definition 3 for base pairs can be adapted to helices.

Definition 4. *Given a set of helices* H*, and a structure F on* H*, F is* maximal for juxtaposition *if it satisfies the two following conditions:*

(i) if f and g are two distinct helices of F, then f and g are juxtaposed,
(ii) if f is a helix of H *not present in F such that $\{f\} \cup F$ is a secondary structure on* H*, then f is nested in some helix of F.*

As in Section 3.1, we define for each pair of helices f and g of H a set of secondary structures MJ(f, g) that contains all structures maximal for juxtaposition for H$[f..g]$.

1. If $f.5start > g.3end$, then $\mathsf{MJ}(f, g) = \{\varepsilon\}$

2. otherwise, if $f.3end > g.3end$, then $\mathsf{MJ}(f, g) = \mathsf{MJ}(f + 1, g)$

3. otherwise

$$\mathsf{MJ}(f, g) = \bigcup \begin{cases} \{f\} \oplus \mathsf{MJ}(\mathsf{nextJuxt}(f), g) & (3a) \\ \mathit{Filter}(f, \mathsf{MJ}(f + 1, g)) & (3b) \end{cases}$$

Now $\oplus$ denotes the concatenation of a helix to a set of structures, $f + 1$ denotes the next helix (wrt the helix ordering) after f and $\mathsf{nextJuxt}(f)$ denotes the smallest helix (wrt the helix ordering) juxtaposed with f. The definition of Filter is a straightforward translation from the definition on base pairs to helices: Given a helix h and a set of secondary structures R on H, the secondary structure S of R is in $\mathit{Filter}(h, \mathsf{R})$ if, and only if, there exists a helix h' in S such that h and h' are neither nested nor juxtaposed. We then have a result analogous to the Theorem 6.

Theorem 3. *For each pair of helices f and g of H, $\mathsf{MJ}(f, g)$ is exactly the set of structures maximal for juxtaposition on $\mathsf{H}[f..g]$.*

3.2.3 Construction of locally optimal secondary structures

The construction of locally optimal secondary structures must take into account the fact that different sets of helices can describe the same base pair secondary structure in some cases. This happens when two nested helices can combine to form a new helix. See Figure 5-a. Of course, the algorithm should output only one structure. To address this problem, we introduce the definition of *strong nestedness*. Intuitively, two helices of are strongly nested, if each time they occur simultaneously in a locally optimal secondary structure, they can be merged into a single helix.

Definition 5. *Let f and g be two helices, such that g is nested in f. Helix g is* strongly nested *in f, if for any helix h which is either juxtaposed with g, or in which g is nested, then h is not nested in f. The set of helices H is* closed under strong nestedness *if for any two helices f and g of H such that g is strongly nested in f, then $f \cup g$ is also in H.*

For any set of helices H, it is easy to construct its closure under strong nestedness by iteratively adding a new helix obtained by merging two strongly nested helices until the set is closed (see Figure 5-a). The set of locally optimal secondary structures is unchanged. From now on, we assume that the set of input helices H is closed under strong nestedness. In this context, we show that each locally optimal secondary structure can be written in a unique way as the combination of helices that are mutually not strongly nested. We call such structures *canonical* structures.

Definition 6. *A structure F is* canonical *if any two helices of F are not strongly nested.*

Property 1. *Let H be a set of helices closed under strong nestedness, and let G be a locally optimal secondary structure on H. There exists a unique canonical structure F, such that F and G describe the same base pairs structure.*

So the problem of computing all locally optimal secondary structures reduces to construct all canonical locally optimal secondary structures. How to solve it ? In Section 3.1, we saw that locally optimal secondary structures for base pairs could be obtained exactly from structures maximal for juxtaposition. Here, each locally optimal secondary structure can still be decomposed into levels of helices that are maximal for juxtaposition. However, the reciprocal result is no longer true. Figure 5-b shows an example where some combination of structures maximal for juxtaposition gives a secondary structure that is not locally optimal. This fact comes from the existence of embedded helices. So we have to identify which combinations of structures maximal for juxtaposition lead to locally optimal secondary structures. With canonical secondary structures, the local optimality could be established by looking at all helices not present in the structure. This allows us to formulate a simple condition that guarantees that a given helix in a secondary structure cannot be replaced by an embedding helix.

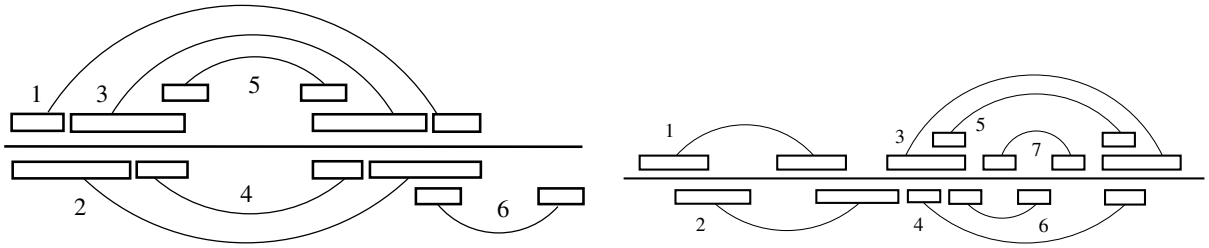


Figure 5: (Left, a.) This helix set contains six helices, numbered from 1 to 6. The union of helices 1 and 3 gives the same set of base pairs as the union of helices 2 and 4. Thus $\{1, 3\}$ and $\{2, 4\}$ are two descriptions of the same structure on base pairs. Helix 3 is strongly nested in helix 1, and helix 4 is strongly nested in helix 2. The closure is obtained by adding the helix $1 \cup 3$. The locally optimal secondary structures are $\{1 \cup 3\}$, $\{2, 5\}$, $\{4, 6\}$, and $\{5, 6\}$. (Right, b.) Structures maximal for juxtaposition and locally optimal secondary structures on helices. The set of helices H contains seven elements, ranked according to a helix ordering. Helix 5 is embedded in helix 3. There are five structures maximal for juxtaposition for $\mathsf{H}[1..3]$: $\{1, 3\}$, $\{1, 4\}$, $\{1, 5\}$, $\{2, 4\}$, $\{2, 5\}$. There are six locally optimal secondary structures: $\{1, 3, 7\}$, $\{1, 4, 6\}$, $\{1, 4, 7\}$, $\{2, 4, 6\}$, $\{2, 4, 7\}$, $\{2, 5, 7\}$. Importantly, the structure $\{1, 5, 7\}$ is not locally optimal, even if its substructures at Toplevel and Nested levels are maximal for juxtaposition. The reason is that it is strictly included in $\{1, 3, 7\}$.

Definition 7. *Let f be a helix of* H*, and T be a subset of* H*. We say that f fulfills the condition $(\star)$ in T if for any helix h of* H *such that f is embedded in h, h is conflicting with some helix of T.*

On Figure 5-b, the helix 5 does not fulfill the condition $(\star)$ in the structure $\{1, 5, 7\}$, as the helix 3 is not conflicting with any helix of the structure. Finally, Theorem 5 on base pairs is replaced by Theorem 8 on helices.

Theorem 4. *Let F be a canonical secondary structure on* H*. F is locally optimal if, and only if, it fulfills the two following properties:*

(i) $\mathsf{Toplevel}(F)$ *is maximal for juxtaposition,*

(ii) for each helix f of F, $\mathsf{Nested}(f, F)$ *is maximal for juxtaposition on* $\mathsf{H}]f..f[$*,* $\mathsf{Nested}(f, F)$ *is not a single helix, and f fulfills the condition $(\star)$ in F.*

It follows that the construction of locally optimal secondary structures from the sets of structures maximal for juxtaposition can be performed using the same algorithm described in Section 3.1.3 and on Figure 4, based on a push-down stack whose elements are structures maximal for juxtaposition. The only difference, at line 5 (push of an element) and at lines 6 and 7 (push of nested and juxtaposed structures), is that structures containing exactly one helix (Condition (ii)-b of Theorem 8), and the structures that do not meet the $(\star)$ condition (Condition (ii)-c of Theorem 8) are not pushed on the stack.

4 Implementation and availability

The algorithm for locally optimal secondary structures with helices was implemented in C in a software called *regliss* (for *RNA energy landscape and secondary structures*). It is freely available on the server: `http://bioinfo.lifl.fr/RNA/regliss`. The input of *regliss* is an RNA sequence together with a set of putative helices given by the user. The helices can also be computed directly by the server from the RNA sequence. The output is the set of all locally optimal structures, sorted according to the free energy as computed with *rnaeval* (Hofacker *et al.*, 1994). We also produce an *energy landscape graph*, useful for visualizing at a glance all found structures.

Running times. We show on Table 1 the running times of *regliss* for a selection of RNA sequences. The program was run on a Athlon Core 2 Duo with 2 GB RAM. The running time mainly depends of

sequence family	species	sequence length	number of helices	number of structures	running time
tRNA – RF00005	*S. pombe*	76 nt	54	511	< 0.2 s
GcvB – RF00022	*Enterobacter sp.*1	208 nt	62	3663	0.08 s
SRP-euk-arch – RF00017	*M. voltae*	298 nt	76	49775	0.62 s
RNase P – RF00010	*P. marinus*	405 nt	76	93142	1.49 s
5S rRNA – RF00001	*D. radiophilus*	119 nt	124	304059	2.51 s
RNase P – RF00010	*S. usitatus*	358 nt	104	1071968	20.92 s

Table 1: Running times and output size of *regliss* for some RNA sequences.

the number of output structures. When there are only some hundred stuctures, `regliss` runs almost instantaneoulsy. However, as the number of structures can be exponential is the number of putative helices, `regliss` can be longer for some sequences.

5 Experimentations

5.1 Example on a SECIS element

SECIS elements (selenocysteine insertion sequence) occur in messenger RNAs encoding selenoproteins (Walczak *et al.*, 1996) and direct the cell machinery to translate UGA stop codons as selenocysteines. They are around 60 nucleotides in length and adopt a stem-loop structure. Here we work with sequence Y11109.1/1272-1330, from *Oreochromis niloticus* (RFAM RF00031 (Gardner *et al.*, 2009)). We first ran *unafold* asking "all" suboptimal structures (100% suboptimality, option `-P 100`). This gives 30 structures, displayed in Figure 6. The expected consensus secondary structure is not present in this set of structures. We also observe that several predictions are redundant: structure #2 is a strict extension of structure #1, structures #4 and #6 are both strict extensions of structure #3, and structure #20 is a strict extension of structure #11. We kept all non-redundant structures, from them we extracted all putative helices. By doing so, we obtained 39 helices and launched *regliss* on this helix set. *regliss* generates 192 locally optimal secondary structures. Structure #14 found by *regliss* is consistent with the consensus structure provided in RFAM for this family.

5.2 Comparison between *regliss* and *unafold*

We generalized the experiment of the previous paragraph, analyzing the size of the output of *regliss* and comparing it to *unafold* on a large number of RNA sequences from RFAM database. We selected all families of RFAM having sequences shorter than 200nt, then picked up five sequences randomly for each family. This gives 5308 sequences. As in the preceding example, we run *unafold* with 100%-suboptimality, and we provide *regliss* with helices coming from non-redundant suboptimal *unafold* structures. Figure 7 shows the number of structures found with *regliss* compared to the theoretical upper bound of Section 3.1.1, as well as the number of structures produced by *unafold* on the same data. As expected, *unafold* generates at most a quadratic number of suboptimal structures, even with a 100% suboptimality level, whereas *regliss* produces an exponential number of locally optimal structures.

We then evaluated the free energy of each structure with *rnaeval*, and selected structures whose energy is greater than or equal to 80% of the optimal energy (we call them "20%-suboptimality structures"). The 5308 sequences divide in three groups:

- 10% sequences: *unafold* finds more structures than *regliss*. Typically, some of these structures are redundant, and are discarded by *regliss*;
- 25% sequences: *unafold* and *regliss* find the same number of 20%-suboptimality structures. In this

(a) All suboptimal structures found by *unafold*

```
     GUUUCUCAGUGAAGGCUACAGAUUAAACCUCUGGCCUCUGGAGCCAGAUGCAUUGAAAC
RFAM <<<<<........<<<<.<<<<.....<<...>>..>>>>.>>>>.........>>>>>
1    .....((((((..(((((.((.......)).)))))((((....)))).))))))... (-15.7)
2    (((..((((((..(((((.((.......)).)))))((((....)))).))))))))) (-14.6)
3    .....((((((..((((.((((.............)))).)))).....))))))... (-14.14)
4    .....((((((..((((.((((.....((...)).))))).)))).....))))))... (-13.9)
5    (((((.(((...((((((.((.......)).)))))))))..((.....)).)))))) (-13.2)
6    .....((((((..((((.((((.((......)).)))).)))).....))))))... (-12.7)
7    .....((((((..((............))((((((.......))))))..))))))... (-12.1)
8    (((((.((....((((((.((.......)).))))))((....)).))....))))) (-11.9)
9    (((((...(((.....)))..........((((((.......))))))......))))) (-11.6)
10   (((((.(((...(((............))))))((.((((....)))).))...))))) (-11.2)
11   ..(((.(((...((((((.((.......)).)))))))))))).............. (-11.1)
12   (((((((...((.(((((.((.......)).)))))))..))))..)))).......... (-10.7)
13   .....((((((.....((.....))....((((((.......))))))..))))))... (-10.6)
14   .....((((((.((........)).....((((((.......))))))..))))))... (-10.4)
15   .((((.....)))).....(((((.......)))).((.((((....)))).)).)..... (-10.2)
16   (..(((..((..((((((.((.......)).))))))).....)).)))..)..... (-9.9)
17   .....(((((....((..(((((.......))))...((((....)))).)))))))... (-9.7)
18   .((((.....)))).................((..((.((((....)))).)))...))... (-9.3)
19   .....((((((.......((.........((((((.......))))))))))))))... (-9.1)
20   ..(((.(((...((((((.((.......)).))))))))))))).(((.....)))).... (-9.1)
21   (((((...((.................)).((((((.......))))))......))))) (-9.09)
22   ((((.((.(((.....))).))..)))).((((((.......))))))..... (-9.0)
23   .((((.....)))).........((((...((((((.......))))))....))))... (-8.9)
24   .........((..((((((.((.......)).))))))(((....)))..........)) (-8.6)
25   .....((((((..((((..........((..........))))))).....))))))... (-8.6)
26   .....((((((.............((....((((((.......))))))))))))))... (-8.2)
27   ((.(((......)))..))..........((((((.......))))))..... (-8.1)
28   .((((.....)))).....(((........((((((.......))))))....))).... (-7.4)
29   ((((...(((....)))..)))).......((((((.......))))))..... (-7.1)
30   .....((((((..((((......((..((...))....)).))))).....))))))... (-5.1)
```

(b) Structure found with *regliss*

```
     GUUUCUCAGUGAAGGCUACAGAUUAAACCUCUGGCCUCUGGAGCCAGAUGCAUUGAAAC
RFAM <<<<<........<<<<.<<<<.....<<...>>..>>>>.>>>>.........>>>>>
14   (((((.((.....((((.((((.....((...)).))))).))))...))....))))) (-10.5)
```

Figure 6: Results for SECIS element (Y11109.1/1272-1330). (a) *unafold* results, all structures. (b) Structure # 14, not found by *unafold* and found by *regliss*.

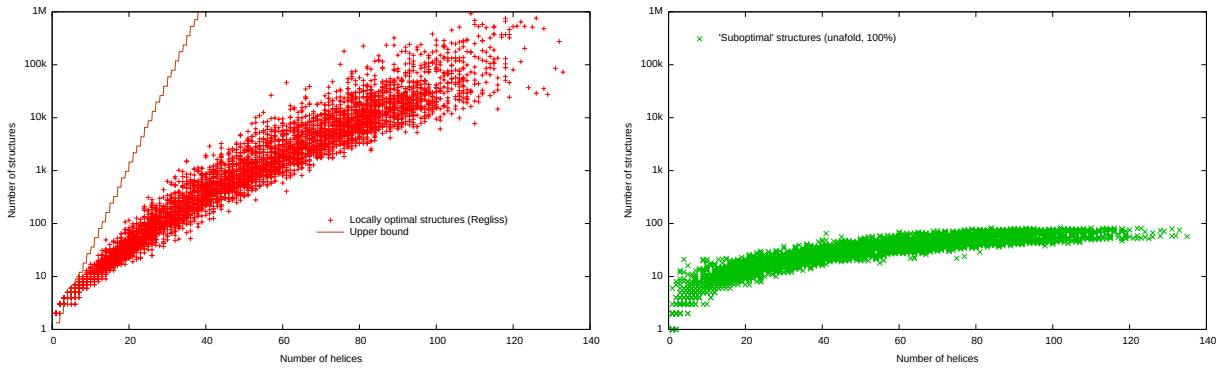


Figure 7: Left: Number of structures produced by *regliss* on 5308 sequences from RFAM. The line is the $3^{\ell/3}$ theoretical upper bound. On average, there are 2.98 times more locally optimal structures than structures maximal for juxtaposition. Right: Number of structures produced by *unafold* (100% suboptimality) on the same set on helices. *unafold* never generates more than 84 structures.

group, almost all sequences have few putative helices and consequently a very small number of 20%-suboptimality structures (1205 sequences have at most 10 different 20%-suboptimality structures). Both programs often find exactly the same 20%-suboptimality structures;

- 65% sequences: *regliss* finds more 20%-suboptimality structures than *unafold*, and so offers a larger variety of structures.

5.3 Structured versus random sequences

In (Clote, 2005a), it is proved that for some families, structured RNA has a different folding landscape than random RNA of the same dinucleotide frequency. We reproduce here this experimentation using *regliss*. We used the sequence of a Hammerhead type III ribozyme sequence, that is also used in (Clote, 2005a). For this sequence, we generated 100 randomized sequences with the same length and the same dinucleotide composition. This computation has been performed with the dishuffle program[1] that implements the dinucleotide shuffle algorithm described in (Altschul & Erickson, 1985). We then compared the distributions of locally optimal secondary structures between these randomized sequences and the initial sequence. Result is shown on Figure 8. Graphs obtained with *regliss* are even more convincing that those obtained with *RNALOSS*. Figure 8 also shows graphs obtained on a 5S rRNA and on a tRNA sequence. Again, these tend to confirm that the folding landcapes, seen as the distribution of locally optimal structures, are different between structured RNAs and random sequences.

6 Conclusion

We introduced a novel approach to produce locally optimal secondary structures of an RNA sequence, which enables us to break down the complexity of the problem into simpler steps. This work shows that all locally optimal secondary structures of a given RNA can effectively be computed. From a practical point of view, these structures can also be filtered out using some post-processing criterium such as the free energy or the shape of the structure. This is a fruitful alternative to existing software programs, and the set of locally optimal secondary structures brings a new look into the folding space of an RNA sequence. Another advantage of the method is that the user can provide its own set of helices, based on the thermodynamic Nearest Neighbour model or any other model.

[1] `http://clavius.bc.edu/clotelab/RNAdinucleotideShuffle`

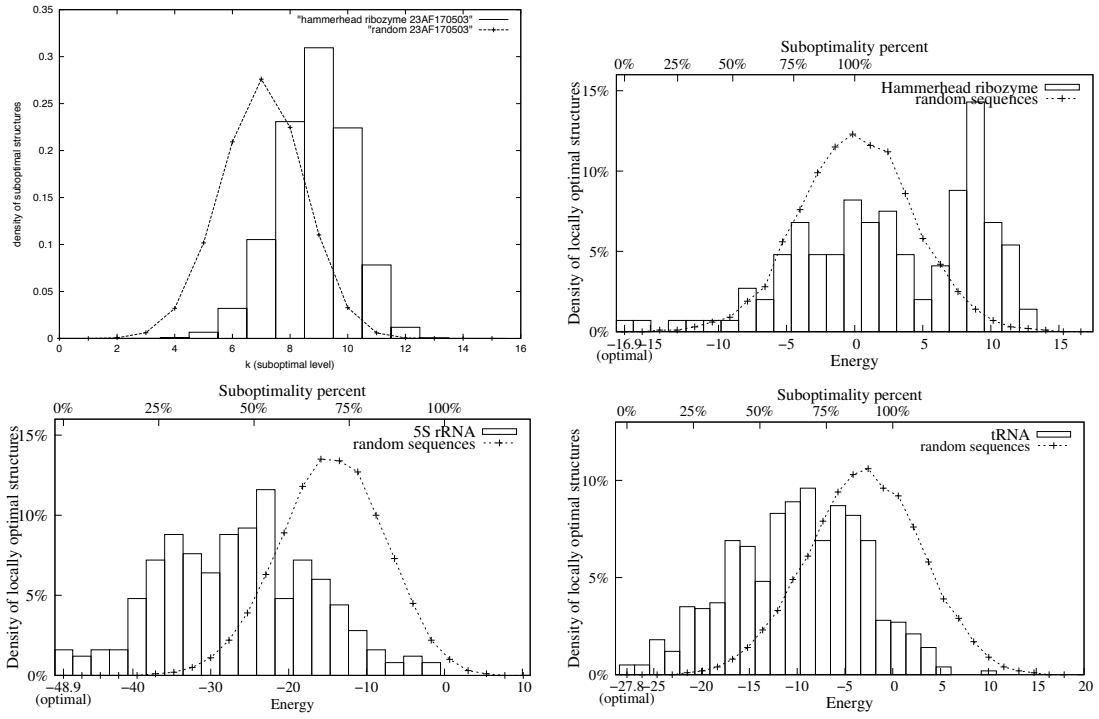


Figure 8: Top: Density of locally optimal secondary structures of Hammerhead type III ribozyme (54 nt, RFAM RF0008, AF170503) versus average density of all locally optimal secondary structures of 100 random sequences of same dinucleotide frequency and same length. Left: *RNALOSS* results (Figure from (Clote, 2005a)), right: *regliss* results. Below: same experiment with 5S rRNA (RFAM RF00001, DQ397844.1/16860-16979), and tRNA (*E.coli* PDB_00313).

Disclosure statement

No competing financial interests exist.

References

Altschul, S F, & Erickson, B W. 1985. Significance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleotide and codon usage. *Molecular biology and evolution*, **2**(6), 526–38.

Clote, P. 2005a. An efficient algorithm to compute the landscape of locally optimal RNA secondary structures with respect to the Nussinov-Jacobson energy model. *J. Computational Biology*, **1**, 83–101.

Clote, P. 2005b. RNALOSS: A web server for RNA locally optimal secondary structures. *Nucleic Acids Res.*, **33**, W600–604.

Eddy, S.R. 2004. How do RNA folding algorithms work? *Nat Biotechnol*, **22**(11), 1457–1458.

Evers, D.J., & Giegerich, R. 2001. Reducing the Conformation Space in RNA Structure Prediction. *In: German Conference on Bioinformatics*.

Gardner, P.P., Daub, J., Tate, J.G., Nawrocki, E.P., Kolbe, D.L., Lindgreen, S., Wilkinson, A.C., Finn, R.D., Griffiths-Jones, S., Eddy, S.R., & Bateman, A. 2009. Rfam: updates to the RNA families database. *Nucleic Acids Research*, **37**, D136–D140.

Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, S., Tacker, M., & Schuster, P. 1994. Fast Folding and Comparison of RNA Secondary Structures. *Monatshefte f. Chemie*, **125**, 167–188.

J., McCaskill. 1990. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 11051119.

Lorenz, W.A., & Clote, P. 2011. Computing the Partition Function for Kinetically Trapped RNA Secondary Structures. *PLoS ONE*, **6(1)**.

Markham, N.R., & Zuker, M;. 2008. UNAFold: software for nucleic acid folding and hybriziation. *Bioinformatics: Structure, Function and Applications*, **453**, 3–31.

Matthews, D.H., Sabrina, J., Zuker, M., & Turner, D. H. 1999. Expanded Sequence Dependence of Thermodynamic Parameters Improves Prediction of RNA Secondary Structure. *J. Mol. Biol.*, **288**, 911–940.

Mattick, J. S., & Makunin, I.V. 2006. Non-coding RNA. *Hum Mol Genet*, **15 (1)**, R17–29.

Moon, J., & Moser, L. 1965. On cliques in graphs. *Israel Journal of Mathematics*, **3**, 23–28.

Nussinov, R., & Jacobson, A.B. 1980. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Nat. Acad. Sci. USA, Biochemistry*, **77**(11), 6309–6313.

Nussinov, R., Pieczenik, G., Griggs, J.R., & Kleitman, D.J. 1978. Algorithms for loop matchings. *SIAM J. Appl. Math.*, **35**, 68–82.

Steffen, P., Voss, B., Rehmsmeier, M., Reeder, J., & Giegerich, R. 2006. RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, **22**(4), 500–503.

Voss, B., Meyer, C., & Giegerich, R. 2004. Evaluating the predictability of conformational switching in RNA. *Bioinformatics*, **20**(10), 1573–1582.

Walczak, R., Westhof, E., P, P. Carbon, & Krol, A. 1996. A novel RNA structural motif in the selenocysteine insertion element of eukaryotic selenoprotein mRNAs. *RNA*, **2**, 367–379.

Waterman, M. S., & Smith, T. F. 1978. RNA secondary structure: a complete mathematical analysis. *Mathematical Biosciences*, **42 (3-4)**, 257–266.

Wuchty, S, Fontana, W, Hofacker, I L, & Schuster, P. 1999. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**(2), 145–165.

Zuker, M. 1989. On finding all suboptimal foldings of an RNA molecule. *Science*, **244**(4900), 48–52.

Zuker, M., & Sankoff, D. 1984. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, **46**(4), 591–621.

Zuker, M., & Stiegler, P. 1981. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, **9(1)**, 133–148.

Supplementary Materials

Proof of Theorem 1 – Section 3.1.1

Theorem 5. *A structure S on* BP *is a locally optimal secondary structure if, and only if,*

(i) $\mathsf{Toplevel}(S)$ *is maximal for juxtaposition on* $\mathsf{BP}[1..n]$,
(ii) for each base pair (x, y) *of* S, $\mathsf{Nested}(x, y, S)$ *is maximal for juxtaposition in* $\mathsf{BP}[x + 1..y - 1]$

The proof of this Theorem comes from the two following lemmas.

Lemma 1. *Let S be a locally optimal secondary structure on* BP. *S fulfils the following properties.*

(i) $\mathsf{Toplevel}(S)$ *is maximal for juxtaposition on* $\mathsf{BP}[1..n]$,
(ii) *for each* (x, y) *in* S, $\mathsf{Nested}(x, y, S)$ *is maximal for juxtaposition on* $\mathsf{BP}[x + 1..y - 1]$.

Proof. (i) Assume that $\mathsf{Toplevel}(S)$ is not in $\mathsf{MJ}(1, n)$: It means that there exists a base pair b in $\mathsf{BP} - \mathsf{Toplevel}(S)$ such that $\{b\} \cup \mathsf{Toplevel}(S)$ is a secondary structure and b is not nested in any base pair of $\mathsf{Toplevel}(S)$. The latter point implies that b is not in S. So $\{b\} \cup S$ is a secondary structure on BP that is a strict extension of S. This contradicts the hypothesis that S is locally optimal.

(ii) Assume there exists (x, y) in S such that $\mathsf{Nested}(x, y, S)$ is not maximal for juxtaposition on $\mathsf{BP}[x + 1..y - 1]$. By Definition 3, this implies that there exists a base pair b in $\mathsf{BP}[x+1..y-1] - \mathsf{Nested}(x, y, S)$ such that $\{b\} \cup \mathsf{Nested}(x, y, S)$ is a secondary structure and b is not nested in any base pair of $\mathsf{Nested}(x, y, S)$. Again, it follows that $\{b\} \cup S$ is a secondary structure on BP that is a strict extension of S, which contradicts the hypothesis that S is locally optimal. □

Lemma 2. *Let S be a structure on* BP *such that*

(i) $\mathsf{Toplevel}(S)$ *is maximal for juxtaposition on* $\mathsf{BP}[1..n]$,
(ii) *for each* (x, y) *in* S, $\mathsf{Nested}(x, y, S)$ *is maximal for juxtaposition on* $\mathsf{BP}[x + 1..y - 1]$.

Then S is a locally optimal secondary structure.

Proof. It is clear that S is a secondary structure. Indeed, any two base pairs of $\mathsf{Toplevel}(S)$ are juxtaposed, and for each (x, y) in S, any two base pairs of $\mathsf{Nested}(x, y, S)$ are juxtaposed.

Now assume that S is not locally optimal on BP. It implies that there exists a base pair b of BP which does not belong to S such that $\{b\} \cup S$ is a secondary structure. As $\mathsf{Toplevel}(S)$ is maximal for juxtaposition, b is necessary nested in some base pair of $\mathsf{Toplevel}(S)$. If (x, y) is the base pair of S immediately nesting b, this contradicts the assumption that $\mathsf{Nested}(x, y, S)$ is maximal for juxtaposition on $\mathsf{BP}[x + 1..y - 1]$. □

Proof of Theorem 2 – Section 3.1.2

Theorem 6. *Let i and j be two positions on α. $\mathsf{MJ}(i, j)$ is exactly the set of all structures maximal for juxtaposition on $\mathsf{BP}[i..j]$.*

Proof. We first establish that for all i and j, all elements of $\mathsf{MJ}(i, j)$ are structures maximal for juxtaposition. The proof is by recurrence on $j - i$.

If $i \geq j$, then, following rule (1), $\mathsf{MJ}(i, j)$ contains only the empty structure ε, and this structure is locally optimal. This is the initial case of the recurrence.

Otherwise, let S be a structure on $\mathsf{MJ}(i, j)$. It is easy to verify that any two distinct base pairs of S are juxtaposed. So it remains to prove that for any base pair b of $\mathsf{BP}[i..j]$ not present in S such that $\{b\} \cup S$ is a secondary structure, b is nested in some base pair of S.

– If S is obtained by rule (2): There is no base pair of $\mathsf{BP}[i..j]$ starting at position i. It follows that b is in $\mathsf{BP}[i + 1..j]$. On the other hand, S belongs to $\mathsf{MJ}(i + 1, j)$. So, by recurrence hypothesis, it is maximal for juxtaposition on $\mathsf{BP}[i + 1..j]$ It follows directly that b is nested in some base pair of S.

– If S is obtained by rule (3a): Let y, $i < y \leq j$, and $S' \in \mathsf{MJ}(y + 1, j)$ such that $S = (i, y) \cup S'$. There are exactly two possibilities for b: it is nested in (i, y), or it is juxtaposed with (i, y). In the first case, we have directly the expected conclusion. In the second case, b is necessarily in $\mathsf{BP}[y + 1..j]$. On the other hand, S' is maximal for juxtaposition on $\mathsf{BP}[y + 1..j]$ by recurrence hypothesis. It follows that b is nested in some base pair of S', and thus is some base pair of S.

– If S is obtained by rule (3b): b cannot start at position i, because it would be conflicting with some base pair of S, by definition of the *Filter* function. So b is in $\mathsf{BP}[i + 1..j]$. We also know that S is maximal for juxtaposition for $\mathsf{BP}[i + 1..j]$, by recurrence hypothesis. The expected result follows.

Conversely, let S be a structure maximal for juxtaposition on $\mathsf{BP}[i..j]$. We show that S is in $\mathsf{MJ}(i, j)$. The proof is by recurrence on $j - i$. If $i \geq j$, the only such S is the empty structure. Otherwise, let (x, y) be the smallest base pair of S and S' the subset of $\mathsf{BP}[y + 1..j]$ such that $S = \{(x, y)\} \cup S'$.

– If $x = i$: Definition 3 implies that S' is maximal for juxtaposition on $\mathsf{BP}[y+1..j]$. Thus S' is in $\mathsf{MJ}(y+1, j)$ by recurrence hypothesis. Then rule (3a) applies and S is in $\mathsf{MJ}(i, j)$.

– If $x > i$: Then S is included in $\mathsf{BP}[i + 1..j]$. Moreover, it is maximal for juxtaposition on $\mathsf{BP}[i + 1..j]$ by Definition 3. So, by recurrence hypothesis, S is in $\mathsf{MJ}(i + 1, j)$. It remains to show that it is transferred to $\mathsf{MJ}(i, j)$. If no base pair of BP starts at position i, then rule (2) applies, and S is also in $\mathsf{MJ}(i, j)$. If not, consider any base pair of the form (i, y) in BP. Since $x > i$, (i, y) does not belong to S. As S is maximal for juxtaposition on $\mathsf{BP}[i..j]$, (i, y) is necessarily conflicting with some base pair of S. So rule (3b) applies and S is in $\mathsf{MJ}(i, j)$. □

Lemma 3. *Given a structure S in $\mathsf{MJ}(i, j)$, let b' be the first base pair of S not nested in b. S belongs to $Filter(b, \mathsf{MJ}(i, j))$ if, and only if, such a b' exists and is conflicting with b.*

Proof. – If b' does not exist, then by definition, S does not belong to $Filter(b, \mathsf{MJ}(i, j))$.

– If b' exists and is conflicting with b, then by definition S belongs to $Filter(b, \mathsf{MJ}(i, j))$.

– If b' exists and is not conflicting with b, then no element of S is conflicting with b. Indeed, in this case, b' is juxtaposed with b. This implies that all base pairs greater than b' are juxtaposed with b. By construction of b', we also know that all base pairs smaller that b' are not conflicting with b. So S does not belong to $Filter(b, \mathsf{MJ}(i,j))$. □

Proof of Theorem 3 – Section 3.2.2

Theorem 7. *For each pair of helices f and g of* H, $\mathsf{MJ}(f,g)$ *is exactly the set of structures maximal for juxtaposition on* $\mathsf{H}[f..g]$.

Proof. The proof is identical to the proof of Theorem 6. □

Proof of Property 1 – Section 3.2.3

We first establish the following Lemma.

Lemma 4. *Let F be a locally optimal secondary structure on* H. *If there are some helices f and g of F, such that* $\mathsf{Nested}(f,F) = \{g\}$, *then g is strongly nested in f.*

Proof. Assume g is not strongly nested in f. By Definition 5, there exists h in H such that h is nested in f, and h is juxtaposed with g or g is nested in h. Since $\mathsf{Nested}(f,F) = \{g\}$, this implies in all cases that $\{h\} \cup F$ is a secondary structure that is a strict extension of F, and thus F is not locally optimal. □

Property 2. *Let* H *be a set of helices closed under strong nestedness, and let G be a locally optimal secondary structure on* H. *There exists a unique canonical structure F, such that F and G describe the same base pairs structure.*

Proof. Let $G = \{g_1, \ldots, g_j\}$ and $F = \{f_1, \ldots, f_k\}$. Assume $\{g_1, \ldots, g_j\}$ is not a canonical structure: There exist two indices i and e in $[1..j]$ such that g_i is strongly nested in g_e. Since H is closed under strong nestedness, by definition, the helix $g_i \cup g_e$ is also present in H. So we can replace g_i and g_e with $g_i \cup g_e$ in $\{g_1, \ldots, g_j\}$, and so forth until all strongly nested helices have been merged. This guarantees the existence of $\{f_1, \ldots, f_k\}$. The unicity comes from Lemma 4. Indeed, given any base pair secondary structure that can be described by a helix structure F, there is a unique way to assemble base pairs into helices such that for no helix f, $\mathsf{Nested}(f,F)$ contains exactly one helix: Helices should be as long as possible. □

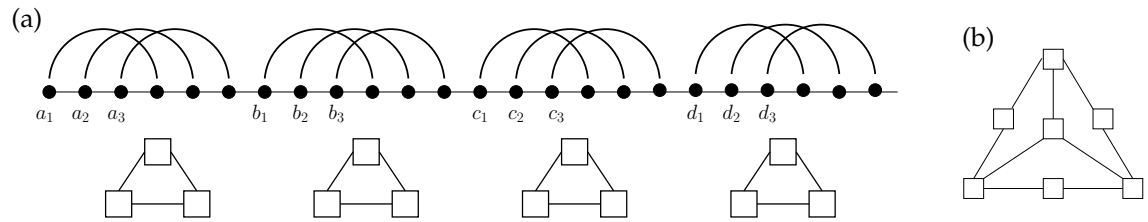
Proof of Theorem 4 – Section 3.2.3

Theorem 8. *Let F be a canonical secondary structure on* H. *F yis locally optimal if, and only if, it fulfills the two following properties:*

(i) $\mathsf{Toplevel}(F)$ *is maximal for juxtaposition,*
(ii) for each helix f of F, $\mathsf{Nested}(f,F)$ *is maximal for juxtaposition on* $\mathsf{H}]f..f[$, $\mathsf{Nested}(f,F)$ *is not a single helix, and f fulfills the condition $(\star)$ in F.*

Lemma 5. *Let F be a locally optimal secondary structure on* H. *We have the following properties.*

(i) $\mathsf{Toplevel}(F)$ *is maximal for juxtaposition on* H,
(ii) *for each helix f in F,* $\mathsf{Nested}(f,F)$ *is maximal for juxtaposition on* $\mathsf{H}]f,f[$.



Supplementary Figure 1: Locally optimal secondary structures and MIS (Maximal Independent Sets). It is possible to encode any set of base pairs into an undirected graph $\mathcal{G}$: The vertex set is BP and there is an edge between two nodes if, and only if, the corresponding base pairs are conflicting. The locally optimal secondary structures are exactly the maximum independent sets of $\mathcal{G}$. (a) The upper bound of the number of MIS is known to be $3^{\ell/3} = 1.44\ldots^{\ell}$ (Moon & Moser, 1965). This upper bound is obtained when the graph is a disjoint union of triangle graphs and this union corresponds to a set of base pairs. In this Figure, the 12 base pairs are conflicting 3 by 3. There are exactly $3^{12/3} = 81$ locally optimal secondary structures: Any set of the form $\{a_i, b_j, c_k, d_e\}$, with $1 \leq i \leq 3$, $1 \leq j \leq 3$, $1 \leq k \leq 3$, $1 \leq e \leq 3$. This example can be extended to any number ℓ of base pairs that is divisible by three. (b) The two problems, locally optimal secondary structures and MIS, are not equivalent. The graph in (b) is composed of seven vertices, and it is easy to verify that no set of seven base pairs can be encoded by this graph.

Proof. The proof of Lemma 5 is basically the same as the one of Lemma 1. □

Lemma 6. *Let* H *be a set of helices that is closed under strong nestedness, and let* $\{f_1, \ldots, f_k\}$ *be a canonical structure on* H. $\{f_1, \ldots, f_k\}$ *is a locally optimal secondary structure on* H *if, and only if, it satisfies the following property: If* g *is a helix of* H *that is not present in* $\{f_1, \ldots, f_k\}$*, then there exists* i *in* $[1..k]$ *such that* g *is conflicting with* f_i*, or* g *is embedded in* f_i*.*

Proof. – If $\{f_1, \ldots, f_k\}$ is locally optimal: Let g be a helix of H that is not in $\{f_1, \ldots, f_k\}$ and that is not embedded in any f_i, $1 \leq i \leq k$. Since $\{f_1, \ldots, f_k\}$ is a canonical structure, this means that there exists a base pair of g that does not appear in $f_1 \cup \ldots \cup f_k$. So $\{f_1, \ldots, f_k, g\}$ is a strict extension of $\{f_1, \ldots, f_k\}$. Since $\{f_1, \ldots, f_k\}$ is locally optimal, this implies that $\{f_1, \ldots, f_k, g\}$ is not a secondary structure: g is conflicting with some helix of $\{f_1, \ldots, f_k\}$.

– If $\{f_1, \ldots, f_k\}$ is not locally optimal: We show that there exists a helix g not present in $\{f_1, \ldots, f_k\}$ such that g is not conflicting with any f_i and g is not embedded in any f_i. Since $\{f_1, \ldots, f_k\}$ is not locally optimal, by definition, there exists a secondary structure $\{g_1, \ldots, g_j\}$ in which $\{f_1, \ldots, f_k\}$ is strictly included. Since H is closed under strong nestedness, we can assume by Property 2 that $\{g_1, \ldots, g_j\}$ is a canonical structure on H. There exists at least one helix g of $\{g_1, \ldots, g_j\}$ such that g contains a base pair that is not present in $f_1 \cup \cdots \cup f_k$. Now let consider a f_i such that f_i and g are neither nested nor juxtaposed. As $\{g_1, \ldots, g_j\}$ is canonical, f_i is embedded in g. So g is not embedded in f_i.

□