Thèse de Azadeh Saffarian, Lille 1, 2011

Algorithmes de prédiction et de recherche de multi-structures d'ARN

Prediction and pattern matching algorithms for RNA multi-structures

THÈSE

présentée et soutenue publiquement le 16 novembre 2011

pour l'obtention du

Doctorat de l'Université de Lille 1 – Sciences et Technologies

(spécialité informatique)

par

Azadeh Saffarian

Composition du jury

Rapporteurs :	Pascal Ferraro, Maître de Conférences, HDR	LaBRI, Université Bordeaux
	Robert Giegerich, Professeur	Universität Bielefeld, Allemagne
Examinateurs :	François Boulier, Professeur	LIFL, Université Lille 1
	Yann Ponty, CR CNRS	LRI, Orsay – LIX, Palaiseau
	Hélène Touzet, DR CNRS, *directrice de thèse*	LIFL, CNRS, Univ. Lille 1, INRIA
	Mathieu Giraud, CR CNRS, *co-encadrant de thèse*	LIFL, CNRS, Univ. Lille 1, INRIA

UNIVERSITÉ DE LILLE 1 – SCIENCES ET TECHNOLOGIES
ÉCOLE DOCTORALE SCIENCES POUR L'INGÉNIEUR
Laboratoire d'Informatique Fondamentale de Lille — UMR 8022
U.F.R. d'I.E.E.A. – Bât. M3 – 59655 VILLENEUVE D'ASCQ CEDEX
Tél. : +33 (0)3 28 77 85 41 – Télécopie : +33 (0)3 28 77 85 37 – email : `direction@lifl.fr`

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

History of this document

- 1 August 2011: initial version
- 21 October 2011:
  - updates and clarifications in the sections 4.1, 4.2 and 4.3
  - minor updates throughout the document
- 28 October 2011:
  - added abstract in French
  - minor updates throughout the document
- 23 January 2012:
  - minor updates throughout the document
  - final version

This PhD thesis can be found online at http://www.lifl.fr/bonsai/doc/phd-azadeh-saffarian.pdf.

January 23, 2012

© 2012 Tous droits réservés.

http://doc.univ-lille1.fr

Contents

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Résumé

L'ARN (acide ribonucléique) est une molécule ubiquitaire qui joue plusieurs rôles fondamentaux au sein de la cellule: synthèse des protéines (ARN messagers), activité catalytique ou implication dans la régulation (ARN non-codants). Les nouvelles technologies de séquençage, dites "à haut débit", permettent actuellement de produire des milliards de séquences à moindre coût, posant de manière cruciale la question de l'analyse de ces données. L'objectif de cette thèse est de définir de **nouvelles méthodes computationnelles** pour aider à l'analyse de ces séquences dans le cas des **ARN non-codants**.

Le chapitre 1 est une introduction à ma thèse, commençant, pages 9 et suivantes, par une présentation des objets biologiques utiles à mon travail. Ces pages ne sont pas un exposé en profondeur sur les ARNs non-codants, mais plutôt une introduction destinée aux informaticiens pour comprendre pourquoi les structures d'ARN sont intéressantes à étudier.

Ensuite, pages 13 et suivantes, je détaille les outils bioinformatiques existants pour traiter les structures d'ARN. Je commence par présenter les représentations des structures ARN et leurs propriétés combinatoires (sections 1.2.1 et section 1.2.2). La "structure secondaire" d'un ARN, formée par l'ensemble des appariements entre bases, délivre ainsi des informations utiles pour étudier la fonction de l'ARN.

Je présente enfin les divers problèmes posés par l'étude des structures ARNs et leurs solutions classiques : l'inférence de structures, pouvant être faite sur une séquence unique (sections 1.2.3 et 1.2.4) ou bien en partant d'un ensemble de séquences similaires par analyse comparative (section 1.2.5), la recherche d'ARNs par homologie, motifs, alignements ou modèles (section 1.2.6), la recherche de "gènes ARN" (section 1.2.7), la comparaison de structures d'ARN (section 1.2.8), et enfin l'interaction ARN-ARN et les autres types de repliements d'ARN (section 1.2.9).

Les algorithmes sur la structure secondaire sont bien plus efficaces que ceux sur la structure tertiaire et quaternaire. De plus, la structure secondaire suffit souvent à identifier et classifier les ARN non-codants. Dans cette thèse, je me suis donc intéressée aux **ensembles de structures secondaires** – structures réelles ou potentielles – que j'ai appelés **multi-structures**.

En effet, même si plusieurs outils peuvent prédire plusieurs structures, optimales et "sous-optimales", aucun outil n'existait, jusqu'à maintenant, pour analyser simultanément un ensemble de telles structures. Pour étudier plusieurs "candidats" dans une sortie typique de mfold/unafold ou de RNAsubopt, il faut répéter certaines analyses, bien que plusieurs candidats partagent des hélices communes. Enfin, quelques ARNs ont plusieurs conformations stables : il est là encore très intéressant de pouvoir étudier simultanément plusieurs structures secondaires.

Pour traiter efficacement les multi-structures, je propose de les considérer comme des **imbrications de structures plates** *(flat structures)*. Cette décomposition suit naturellement la

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

représentation en arbre des ARNs et factorise les paires de bases ou les hélices communes entre plusieurs structures. Le chapitre 2 présente mes définitions de structures secondaires, que cela soit sur des paires de bases ou sur des hélices. J'ai ensuite étudié les deux sujets suivants :

- *Quelle est la multi-structure représentant toutes les structures secondaires "pertinentes" d'un ARN ?*

  Il y a souvent beaucoup de structures sous-optimales qui ont une énergie proche de l'optimale, mais qui sont topologiquement très proches. Je choisis donc de garder uniquement les **structures localement optimales** pour décrire toutes les structures possibles. Ces structures sont, en un certain sens, "saturées" : on ne peut leur ajouter aucune paire de bases. Ces structures ont été introduites par Clote dans [22], dans un contexte limité aux paires de bases. La thèse généralise cette définition aux hélices thermodynamiquement stables.

  Le chapitre 3 propose un algorithme efficace pour énumérer les structures localement optimales. Dans une telle structure, chaque structure plate est maximale en un certain sens: l'algorithme manipule donc des structures **maximales par juxtaposition**. Pour des raisons pédagogiques, la thèse présente tout d'abord l'algorithme sur un modèle simple de Nussinov-Jacobson (section 3.2), puis l'étend à des hélices thermodynamiquement stables (section 3.3). J'ai implémenté cet algorithme dans le logiciel Regliss, et j'ai effectué plusieurs expériences montrant comment Regliss peut décrire le paysage énergétique de séquences d'ARN (pages 51 et suivantes).

- *Comment trouver toutes les occurences d'une multi-structure dans une séquence ?*

  Dans ce problème de recherche de motifs, on part d'une multi-structure, donnée comme imbrication de structures plates. Cette multi-structure peut représenter des structures putatives (comme des strucutres sous-optimales produites par un logiciel de prédiction, ou comme les structures localment optimales produites par Regliss) ou des structures réelles (pour des ARNs ayant plusieurs structures stables, ou bien pour plusieurs structures similaires au sein de familles d'ARNs).

  Le chapitre 4 propose un algorithme efficace de programmation dynamique pour localiser toutes les occurences d'une telle multi-structure dans une séquence génomique. Ici encore, la décomposition en imbrication de structures plates est la clé de l'algorithme (section 4.2). La thèse présente aussi comment des contraintes supplémentaires peuvent donner des résultats biologiquement plus pertinents (section 4.3).

  J'ai implémenté cet algorithme dans le logiciel Alterna, et testé plusieurs cas d'utilisation (pages 74 et suivantes). J'ai essayé de montrer que, lorsqu'on connaît un ensemble de structures putatives (même sans connaître la "bonne" structure), il est possible de rechercher de telles structures sur le génome et d'annoter des séquences pouvant être des ARNs avec la même fonction.

Je conclus ma thèse, pages 79 et suivantes, en donnant quelques pistes de recherche: est-ce possible de définir des multi-structures pondérées, où certaines instances seraient plus privilégiées que d'autres ? Comment visualiser, comparer, indexer des multi-structures ? À la fin de cette thèse, je pense pouvoir affirmer que les multi-structures sont un outil prometteur pour explorer la richesse des structures secondaires des ARN non-codants.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

Introduction

Organisms can be composed of one single cell to, as in humans, trillions of cells grouped into specialized organs. Despite this diversity, there are universal principles: the metabolism of these cells is governed by several molecules, including DNA (deoxyribonucleic acid), RNA (ribonucleic acid) and proteins.

The study of RNA is of great interest. In cells, RNA molecules have various functions: just as it can store and deliver the DNA message to the protein (*messenger RNA*, mRNA), it can also directly catalyze chemical reactions and act as a regulator (functional RNAs, also called *non-coding RNAs*, ncRNAs). Moreover, RNA can been studied as a major step in the evolution of cellular life: the "RNA world hypothesis" suggests that pre-cellular life could have been based on ancient RNA, and that DNA may be an evolution of RNA [42].

Nowadays, recent sequencing technologies yield billions of genomic *sequences* at a very small cost. However, sequencing is only the first step. The function of the sequence remains open for investigation. For RNA, knowing the *structure* of the molecule is a relevant information to further study the RNA and its function.

⋆ ⋆ ⋆

The whole thesis is focused on structures of RNA, and, more specifically, on algorithms on *sets of suboptimal secondary structures*.

This document contains the biological and bioinformatical background of my work (Chapter 1) as well as some definitions considering RNA structures on base pair sets then on helix sets (Chapter 2). Then the core of my thesis consists of algorithms to search all "locally optimal structures" of an RNA and occurrences of an "RNA multi-structure" in a genomic sequence (Chapters 3 and 4). All these algorithms have been implemented and tested on real data.

A more detailed content of the thesis is given at the end of Chapter 1, after introducing necessary concepts.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Chapter 1

Background – RNAs, non-coding RNAs, and bioinformatics

This Chapter presents the background of my thesis, first on biology, then on bioinformatics. In Section 1.1, I present some biological facts on RNAs and non-coding RNAs (ncRNAs). This is not a deep presentation of ncRNAs, but rather an introduction for computer scientists to understand why RNA structure is worth of study. Then, in Section 1.2, I present a survey of bioinformatics for RNA structures. Finally, Section 1.3 details the contents of the thesis.

1.1 Ribonucleic acids (RNAs)

1.1.1 RNA and its structures

The **ribonucleic acid (RNA)** is a single-stranded molecule composed of a ribose-phosphate backbone carrying a sequence of *bases* A (Adenine), C (Cytosine), G (Guanine), and U (Uracile). Usual RNAs range from about ten to thousands of bases. This sequence, or "primary structure", is produced by the transcription of genome (see below).

The RNA molecule folds upon itself to form **base pairs**. These pairings are done by hydrogen bonds between two nucleic acids. The most common base pairs are *Watson-Crick base pairs*, or "canonical base pairs", A–U and C–G. Other combinations are possible, such as the *wobble* pair G–U. The "secondary structure" of the RNA is a set of non-crossing base pairs (Figure 1.1), creating *stems-loops*, *helices*, *multi-loops* and other structural elements (Figure 1.2).

In further foldings, some other pairings take place, such as *pseudo-knots*, for example *kissing hairpins*, that have crossing base pairs, or even including other bonds such as *base triplets*. Of course, the RNA molecule finally folds in the 3D space of the cell, and can interact inside a larger complex, with other RNAs, proteins, or other molecules.

1.1.2 The central dogma and messenger RNA (mRNA)

The "central dogma of molecular biology" is depicted on the upper part of Figure 1.4 : DNA is replicated, DNA makes RNA trough transcription, and RNA makes proteins through translation.

The first ideas on this flow of information dates from more than sixty years ago, with papers

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

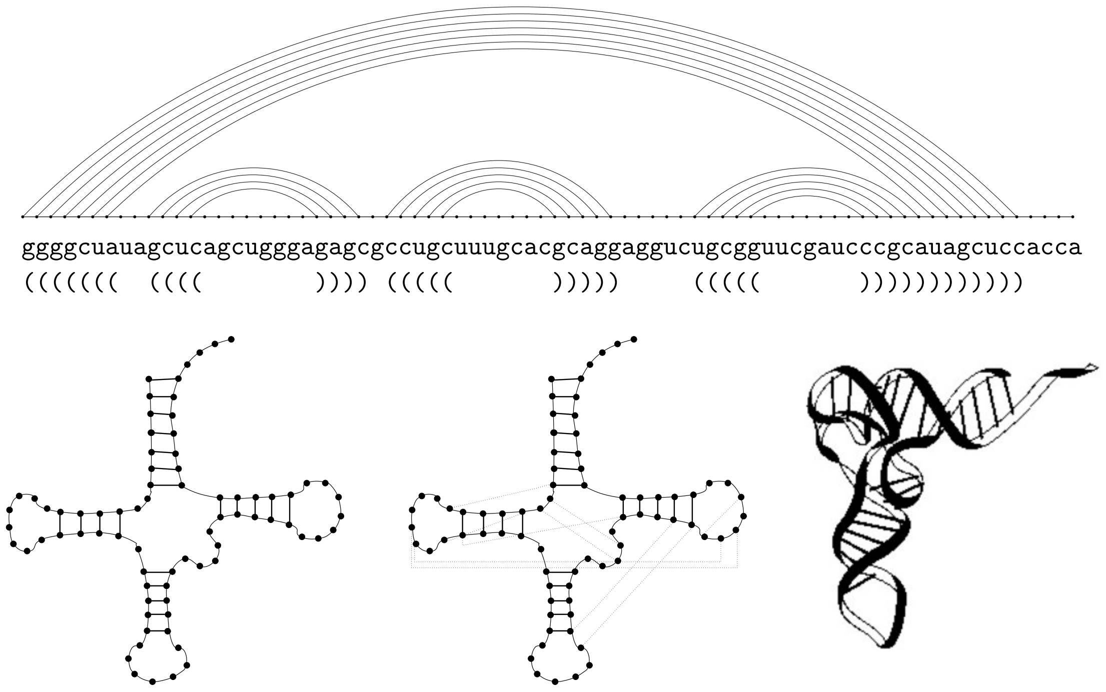

ggggcuauagcucagcugggagagcgccugcuuugcacgcaggaggucugcgguucgaucccgcauagcuccacca
(((((((((((   )))) (((((   )))))   (((((   ))))))))))))

Figure 1.1: (1, top) Primary structure (sequence) of transfer RNA (tRNA) Ala (76 bases), annotated with secondary structure. (2, left) Secondary structure (without pseudo-knots) "in cloverleaf", seen as a planar graph. (3, middle) Secondary structure with pseudo-knots (4, right) "Real" 3D structure in the cell in "L-shape". The global organization of tRNA's structure is well conserved in all forms of life.

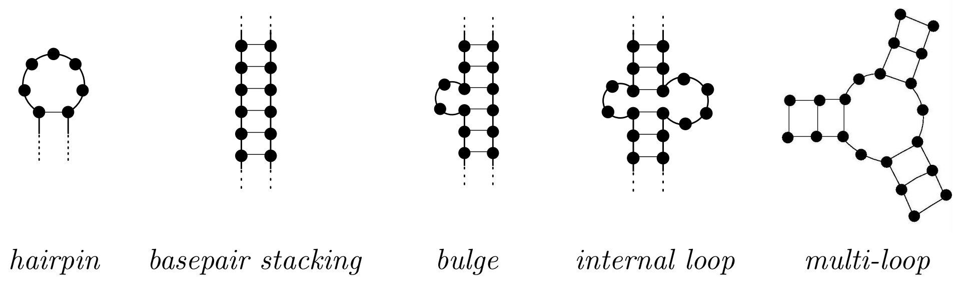


Figure 1.2: Some elements of RNA secondary structures. In this thesis, basepair stacking, bulges and internal loops will be gathered in the notion of "helices" (see Chapter 2).

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

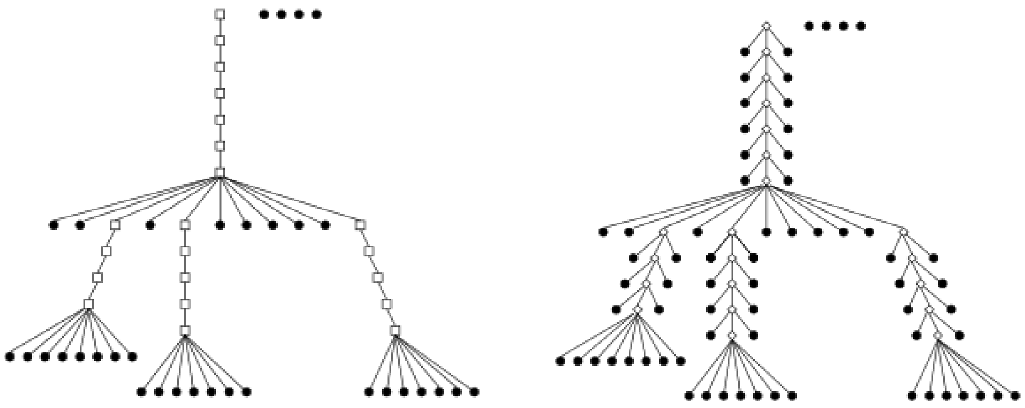


Figure 1.3: Two tree representations of E. coli Ala-tRNA (see Figure 1.1) with different abstraction levels. In both representations, a node represents a base pair. The leaves represent either only the free bases (left) or all the bases, including the paired ones (right).

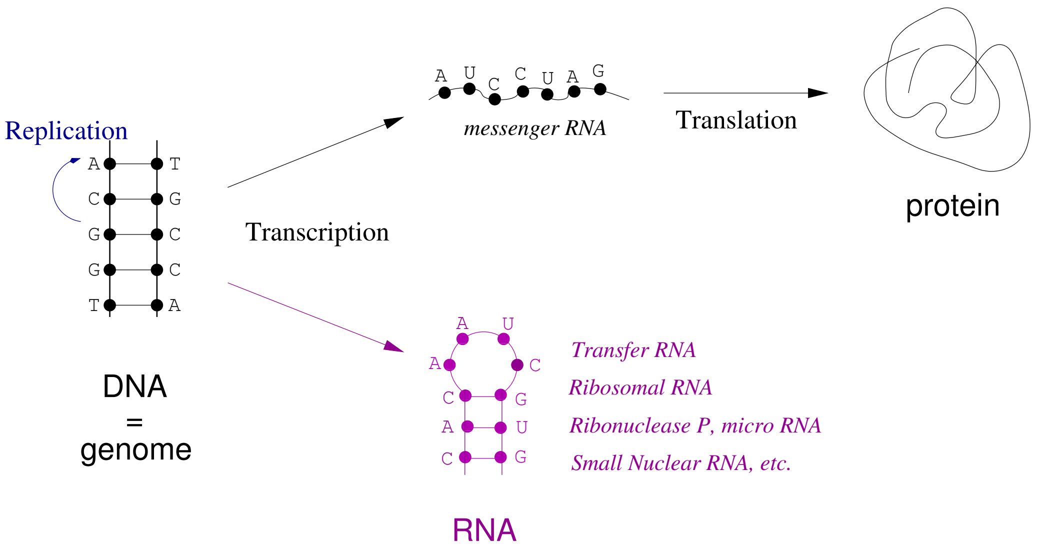


Figure 1.4: RNA and protein synthesis. In the nucleus of the cell, DNA is transcribed to RNA. Outside the nucleus, messenger RNA is translated to proteins within a ribosome. Some RNA are not protein-coding: they act by themselves.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

by Boivin and Vendrely in 1947 [15] and Dounce in 1952 [27]. In his famous 1958 paper, Francis Crick coined the term "central dogma" [25, 26], focusing on the fact that information cannot be transferred back from protein to either protein or nucleic acid. Though the central dogma is respected in the majority of cellular expression mechanisms, some observations are in contradiction with this classic formalism, as inverse transcription by retrovirus (RNA $\to$ DNA), RNA replication (RNA $\to$ RNA), and prion pseudo-replication (protein $\to$ protein). Such exceptions will not be discussed in this thesis.

The RNA has a key role in this central dogma, passing the information between DNA and proteins. In 1961, François Jacob and Jacques Monod assumed that the DNA message contained within a so-called *"structural gene"* is necessary and sufficient to define the structure of a protein via an intermediate *"structural messenger"* [59] [Nobel Prize of Physiology and Medicine, 1965]. They identified this messenger as a small fraction of RNAs, which have been found before in E. coli and yeast, and called it *"messenger RNA"*. Further studies in the 1960's discovered the *genetic code*, mapping every *codon* (three bases of DNA) to one amino acid.

1.1.3 Non-coding RNAs (ncRNAs)

In the eukaryotes, a very small portion of the genome is protein-coding genes [75] with, for example, about only 2% on the 3 GB human genome. Nowadays we know that, even if the other portions do not code for protein, these may have important functionality for the cell metabolism. Focusing on RNAs, it is known today that the majority, if not all, of the genome is transcribed. Some of these RNAs act in various functional roles, either directly or indirectly. They are called *functional RNAs* or *non-coding RNAs (ncRNAs).*

Extensive reviews on ncRNAs can be found in [75, 76]. I present here a brief history of the discovery of some ncRNAs.

- In the already cited 1961 paper, Jacob and Monod discovered "regulator genes" [59], that act on the regulation of other genes. They hypothesized that the product of such genes is not a protein, but rather an RNA which acts directly as a repressor. Even if, at their time, they had no experiences to directly assert the existence of ncRNAs, their ideas were visionary.

- In the 1960's, two fundamental classes of non-coding RNAs were discovered:

  - the **transfer RNAs (tRNAs)** are the key to the genetic code: During translation, they bring to the growing protein the amino acid corresponding to the codon (see Figure 1.4). The structure of Alanine tRNA is described in 1965, linking DNA and protein synthesis [53] [Nobel Prize of Physiology or Medicine, 1968];
  - the **ribosomal RNAs (rRNAs)** are essential parts of the ribosome, which is responsible for the translation of mRNA to proteins.

- In 1977, it was discovered that some protein-coding sequences were not present continuously in the DNA, but were split into several pieces [11, 21] [Nobel Prize of Physiology or Medicine, 1993]. The *introns* present between these "split genes", or exons [44], are thus not translated into the protein. The *splicing* is the process to remove the introns from the RNA. In 1982, Cech discovered that the **intron Group I RNA** can be self-spliced, without any other protein [Nobel Prize of Chemistry, 1989].

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

- In 1978, Altman discovered the **RNase P**, a RNA-cutting enzyme composed of a complex between one protein and one RNA [Nobel Prize of Chemistry, 1989]. He demonstrated that the RNA part of the RNase P is essential for a catalytic reaction, and, later, that the RNA molecule itself could complete the function of the enzyme.

In the following years, many non-coding RNAs have been found in bacteria, archae and eukaryotes. As an example, **micro RNAs (miRNAs)** have been discovered in the majority of eukaryotes. By binding to mRNAs, they increase or decrease the RNA activity and thus the protein production [5]. Furthermore, they are central to **RNA interference (RNAi)**, where a *double-stranded* RNA plays a role in regulating gene activity. The principle of RNAi has been discovered in 1998 by Andrew Z. Fire and Craig C. Mello [35] [Nobel Prize of Physiology or Medicine, 2006]. RNAi is also involved in immunity, and provides some interesting possibilities in gene technology.

Our knowledge on the ncRNA families is thus still expanding. Non-coding RNAs are key points for understanding the metabolism of cells and organisms [76].

1.2 Bioinformatics for RNA structures

Many non-coding RNAs are structured or partially structured, and these structures are intimately tied to their function. As with any biological object that can be modeled, computer science tries to help the biologists in studying RNA structures. Whereas having primary sequence information is cheap, from either genome sequencing or transcriptome sequencing, the experimental identification of RNA structures with NMR or crystallography remains expensive in terms of biological preparation, material and time. Thus lots of methods in RNA bioinformatics will try to *start from primary structures and produce knowledge on secondary or 3D structures*. Ideally, we should all work with 3D foldings, modeling and simulating the complete environment of RNAs in the cell. Of course, such "almost true" foldings are very difficult to compute. However, it should be noted that secondary structures act as a scaffold for the next 3D foldings, and are more computationally tractable than full foldings. Secondary structures are thus the piece of choice for bioinformatics, and they are easily representable.

This section gives thus an overview of RNA structure bioinformatics, focusing mainly on secondary structures. We first begin by presenting main RNA computer representations (Section 1.2.1) as well as combinatorial properties of these structures (Section 1.2.2). Complete studies of structured RNAs include a series of various problems: Structure inference, that can be done on a single RNA sequence (Sections 1.2.3 and 1.2.4) or on a set of related sequences by comparative analysis (Section 1.2.5), RNA search by homology, that can be done through alignment or model-based pattern matching (Section 1.2.6), identification of RNA gene in genomes (Section 1.2.7), RNA structure comparisons (Section 1.2.8), or RNA-RNA interactions and non-secondary RNA foldings (Section 1.2.9).

1.2.1 Computer representations of RNA structures

At the simplest level, an RNA "primary sequence" of length n is a word $x_1 ... x_n$ over the alphabet $\{A, C, G, U\}$. As the RNA folding first happens by pairing nucleic bases (see Section 1.1.1),

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

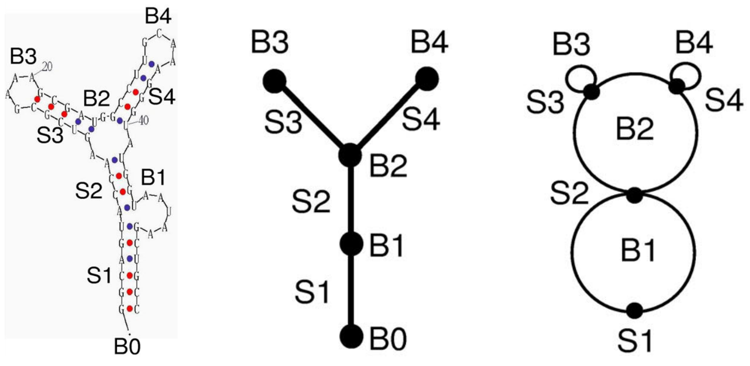


Figure 1.5: Secondary structures of P5abc RNA represented as both tree (middle) and dual (right) graphs, from [34]. Helices (Sx) are represented with edges in tree graphs and vertices in dual graphs. Bulges, loops and junctions (Bx) are represented with vertices in tree graphs and edges in dual graphs. The authors further provide an analysis of RNAs structure as graphs, using network theory.

the most simple approach to describe a secondary structure is just to enumerate these base pairs. The secondary structure being composed only with *nesting* and *juxtaposing* base pairs, three equivalent arborescent descriptions can be used for non-pseudoknotted RNA secondary structures:

- The **bracket notation** (Figure 1.1.(1), bottom), used by Hofacker et al. in the Vienna package[1] [51], sees any RNA secondary structure on a sequence of length n as a sequence $s_1 s_2 \dots s_n$ on the alphabet $\{\texttt{(}, \texttt{)}, \texttt{.}\}$. For each basepair (i, j), $s_i = \texttt{(}$ and $s_j = \texttt{)}$, and, for each unpaired base i, s_i is a dot;
- Any set of non-crossing base pairs can also be seen as a **rooted ordered tree** [36, 49, 129], where each node represents a base pair (Figure 1.3). There is an edge between immediately nesting base pairs;
- **Non-crossing arc-annotated sequences** display all base pairs interactions (Figure 1.1.(1), top), where any node is involved in at most one arc, and where two arcs shall not cross

At a higher level, any secondary structure can be seen as a set of several structural elements, as for example those of Figure 1.2. Such elements can be directly considered into a tree representation (Figure 1.3). Other authors give alternate descriptions of RNA secondary structures, as for example with *dual graphs* in [34] (Figure 1.5).

Structures with pseudo-knots are equivalently representable with extended bracket notation (with different symbols for pseudo-knots, otherwise the notation is ambiguous), with some directed acyclic graphs of base pairs, or with non restricted arc-annotated sequences (with crossing arcs).

1.2.2 Combinatorics of bracketed sequences

The number of possible RNA secondary structures grows exponentially with the sequence length. Considering only the paired bases, any RNA secondary structure (without pseudo-knots) can be

[1] http://www.tbi.univie.ac.at/~ivo/RNA/bracket.html

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

seen as a well-parenthesed word (called also Dyck word) over the alphabet {`(`, `)`}. The number of such words with ℓ pairs of parentheses is exactly the ℓ^{th} Catalan number [104], given by

$$C_\ell = \frac{1}{\ell+1}\binom{2\ell}{\ell} = \frac{(2\ell)!}{(\ell+1)!\,\ell!}$$

Adding unpaired bases, the number of well-parenthesed bracketed sequences of length n on the alphabet {`(`, `)`, `.`} is asymptotically equivalent to S_n, given by [106]:

$$S_n = \sqrt{\frac{3}{4\pi}}\, n^{-3/2}\, 3^n$$

Other asymptotic values have been obtained under more stringent conditions, closer to real RNA structures (bounds on the size of the interior loops, no isolated base pair) [52, 106], as well as on pseudo-knotted structures [97].

1.2.3 Minimum free energy (MFE) structure prediction

The problem of *structure prediction* (also called structure inference or folding inference) consists in guessing the secondary structure of a given RNA sequence. Since the 3D structure of an RNA is based on its secondary structure, this latter is a essential step to study ncRNA function. While the secondary structure in turn is mostly determined by the primary structure, specific tools are developed in order to predict secondary structures.

As most of structured RNAs try to stabilize their structures, the basic idea is to predict the *Minimum Free Energy (MFE)* structure. The computation of MFE is based on the Gibbs Free Energy value of a RNA structure [116].

The nearest-neighbor model. The free energy of an RNA structure is usually calculated under the *nearest-neighbor model*. This model assumes that, in a RNA structure, the energy contributed by each structural element is only under the influence of bases of this structural element and of neighbor base interactions. To compute the free energy of a structure, this model requires a set of parameters and rules, in particular the parameters defining the energy associated to a variety of neighbor interactions.

Starting from 1986, the Turner group has presented sets of energy parameters under this nearest-neighbor model for predicting stability of both RNA and DNA secondary structures [37, 115]. These sets of parameters utilize empirical rules, generally derived from optical melting experimental data, as the basis of the predictions. The rules predict both free energy and enthalpy changes of helices through *stacking of base pairs*. There are also rules to predict the stability of helices, hairpin loops, small internal loops, large internal loops, bulge loops, multi-branch loops, exterior loops and pseudoknots [73, 67].

This so-called "Turner energy model"[2] is now widely used by secondary structure prediction programs, such as mfold [132], RNAfold [51, 30] or RNAsubopt [127].

Algorithms and tools. The prediction of secondary structure by energy minimization began with Tinoco et al. in the 1970's [17, 112, 113]. The first specific dynamic programming algorithm

[2]`http://rna.urmc.rochester.edu/NNDB`

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

for secondary structure prediction maximizing the number of base pairs was introduced later, in 1978, by Nussinov et al. [84]. Base pair maximization is a very simple model and does not take into account full energy parameters. The same year, Waterman and Smith [125] presented a method to predict the MFE structure through a simple nearest-neighbor model. In 1981, Zuker and Stiegler [134] proposed a more elaborated dynamic programming algorithm, predicting the MFE structure. Later in the 1980's, these folding algorithms were applied on the Turner energy model [131].

mfold/unafold [132] and RNAfold [51] are some popular implementations of these algorithms. For a sequence of length n, they run in time $O(n^3)$ (or even $O(n^4)$ if one takes into account interior loops of any size) and space $O(n^2)$. Detailed analysis of suboptimal predictions, especially for mfold/unafold, will be done in Chapter 3. Recently, some optimizations in average less time and space have been proposed [8].

1.2.4 Structure prediction, beyond the MFE structure

Most of the time, computing the MFE structure is not enough:

- Even if the Turner model is well established, this empiric thermodynamic data can be incomplete: slight changes in the thermodynamic model could produce very different foldings with similar energy levels;
- The Nearest Neighbor model does not allow for pseudoknots and base triplets, and it does not reflect the interactions of the RNA with other molecules in the cell. Thus the projection of a real RNA structure on its secondary component may not be optimal;
- In the cell, RNA molecules do not take only the most stable structure, but they seem to rapidly change their conformation between structures with similar free energies. These changes can be induced by the interaction of the RNA with other molecules, and mostly concern their 3D structure. Some RNAs, including several mRNAs in their untranslated region, have even been shown to bear conformational switching in their secondary structure.

Thus the MFE structure is not sufficient to fully understand the folding possibilities of an RNA. Instead of getting the optimal structure, a complete backtracking through the dynamic programming table can generate all **suboptimal secondary structures** within a given threshold from minimum free energy. 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. Such suboptimal backtracking is implemented in RNAsubopt [127] using Waterman and Byers algorithm [124], whereas mfold does not implement this full backtrack (see Chapter 3).

However, the set of suboptimal structures is often huge : the number of such structures is exponential in the length of the sequence. RNAshapes [105] suggests "abstract shapes" to classify the secondary structures. During computation of the suboptimal structures, RNAshapes partitions the folding space into sets of structures according to their shapes.

One may be interested by selecting *only relevant structures*, for example those corresponding to a local minimum of energy. The goal is to keep relevant suboptimal structures in the free energy landscape that correspond to local minima. The notions of *saturated structures* and *locally optimal secondary structures* meet these requirements.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

In [32, 133] 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 and, moreover, there is no isolated base pair.

In 2005, Clote defines *locally optimal structures* [22]. A secondary structure is locally optimal when no base pairs can be added without creating a conflict: either crossing pairings, or a base triplet. This last definition is purely topological, and does not involve any consideration of thermodynamic stability on the computed structures. Clote proposes an algorithm to count all these structures, according to the difference in the number of base pairs compared to the optimal structure [22]. The algorithm is implemented in the web server RNALOSS [23], and applied to analyze the folding landscape of some RNA sequences. Finally, in 2007, Waldispühl and Clote proposed an algorithm sampling a collection of saturated structures, again according to the number of base pairs [120].

In these works, the energy model is limited to Nussinov-Jacobson (number of base pairs), and moreover the algorithm does not effectively output the structures. We will argue on Clote's algorithm in Chapter 3. We will also present another definition for the same RNA locally optimal structures, taking stable helices (with respect to the Turner energy model) as the base component.

Finally, some studies give a further understanding of the folding space:

- **Prediction using partition function.** The Boltzmann distribution describes the probability that a system (here an RNA molecule) is in a given state (here a possible RNA structure), including thermodynamic parameters. The partition function is obtained through normalization of the Boltzmann distribution, the sum of probabilities of all states being equal to one. The partition function encodes thus statistical characteristics of the folding space, and can, for example, assert the probability of a specific base pair.

  For RNA, the McCaskill algorithm computes a partition function by dynamic programming in time $O(n^3)$, where n is the length of the sequence [77]. This algorithm is very similar to the algorithms computing the MFE structure.

  The partition function is for example implemented in RNAPDIST [16], and the matrix gathering all base pair probabilities is used in RNAfold [51] as well as in RNASTRUCTURE [72].

- **Prediction of structure switching.** PaRNAss [118] is a software proposed in [43] to predict a non-pseudoknotted structural switching, looking for two local minima corresponding to two stable structures clearly separated by an energy barrier. This approach includes RNA folding and structure comparison.

- **Neighbors of a structure.** To browse the folding space, RNAbor provides the neighbor structures at a given *basepair distance* of an initial given structure [38]. The $\delta-$neighbor structures are the secondary structures yielded by removing or adding exactly δ base pairs from initial structures. RNAbor is also used by their authors to study structure switching.

1.2.5 Structure prediction through comparative analysis

DNA sequences evolve through time, and this evolution is reflected in transcripted RNA sequences. In structured RNA families, some *structural domains* are conserved across species,

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

leading to consensus structures that are related to function of the family. More precisely, RNA homologous sequences often present base pairs *covariations* (also called compensatory mutations): The secondary structure being under selection pressure, mutation at one nucleotide is corrected on the pairing base. Such covariations are often a proof of a conserved structure. Prediction of secondary structure can thus be better performed on a set of homologous RNA than on a single sequence. Tools of this section predict a common structure, or a subset of this structure, from a set of related RNA sequences.

The first possibility consists in using a multiple sequence alignment to build the consensus structure, then to infer a common folding compatible with this alignment. The folding must be supported by covariations and have a good free energy level. This is done for example in RNAalifold [50]. This approach proves very successful on well-conserved sequences, but the correction of the predicted pairings is closely related to the quality of the input alignment.

In 1985, Sankoff proposed a dynamic programming algorithm to align and fold two sequences at the same time, resulting in a low free energy structure common to the two sequences [96]. As a consequence, it does not require any prior multiple sequence alignment, and appears to be able to withstand to noisy datasets. For two sequences of length n, the complexity of Sankoff's algorithm is $O(n^6)$ in time and $O(n^4)$ in space. The algorithm can be extended to multiple sequences, and was adapted with many heuristics using covariations, phylogenetic comparison and sequence conservation, as in DYNALIGN [74], FOLDALIGN [48] or CARNAC [114, 87]. Ziv-Ukelson et al. [130] present an optimization of Sankoff's algorithm that brings, on average, a linear factor gain. Finally, variations of this problem appeared in [86], with RNAprofile that first searches for conserved hairpins to infer the complete common secondary structure, or in [54], with GPRM, based on genetic programming, and in [89] based on stochastic context free grammars that are both approaches for finding small common secondary structure elements within larger sequences.

1.2.6 Homology search

In this section, we present techniques to decide if an unknown sequence belongs to a family of known ncNRAs. Given a family of homologous ncRNAs, a **model** (also called a descriptor, a profile, a signature, a structural motif) is a manual or computed description of the conserved sequence and structure into the family. Ideally, the model should remain specific to the family. Then the problem is to locate all potential occurrences of the model on selected sequences or on large genomic DNA sequences.

- In **alignment based approaches**, both the structure and the sequence of a model are aligned against a tested sequence. These methods are natural extensions of dynamic programming sequence alignment algorithms, such as local alignment of Smith and Waterman [102] or Viterbi's algorithm for HMM [117]. If n is the size of the sequence, and m is the size of the motif, these methods can be in time complexity $O(n^2m^2)$, taking into account base pair interactions.

  In this context, the most popular approach is *stochastic context-free grammars* (SCFGs) [31, 94]. Such grammars learn the RNA pattern from a multiple sequence alignment. They are used in Rfam (see Section 1.2.10) and implemented in Infernal [81].

  The search algorithm uses classical parsing algorithms for grammars, such as CYK [61]. As the underlying dynamic programming algorithm has an high computational complexity,

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

several approaches try to optimize their parsing, either with banded dynamic programming techniques [80], or with filtering heuristics, as in RAVENNA [126] or in [107].

Similar to SCFGs, some methods also model RNA structures with ordered trees [9], or with other models such in ERPIN [40].

- Another solution is **descriptor based methods**, where the RNA model is explicitly defined as an association of sequence and structural elements, sometimes with constraints on distance or on other features.

  Several description languages have been proposed to allow users to define manually such RNA descriptors: RNAmot [41], RNAbob [29], Patsearch [46], Palingol [13], RNAmotif [69], Patscan [28], Locomotif [90], MilPat [111], and DARN! [135].

  The complexity of the underlying scan algorithms depends on the expressiveness of descriptors. The advantage (and the disadvantage) of these methods is that their accuracy is heavily depending on the descriptor, and thus relies on the expertise of the user.

In this Thesis, I will propose, in Chapter 4, a variation of this problem with an algorithm to match a set of alternate RNA structures against a genomic sequence. This algorithm can be linked either to structure alignment methods (when using a structure computed from one or several RNA, or to descriptor based methods (when using a manually described set of structures).

1.2.7 Non-coding RNA gene finders

In this Section, we present tools that are able to locate potential RNA genes in genomic sequences. RNA gene prediction is a difficult question that is still partly unsolved. Unlike protein-coding genes, RNA sequences do not contain simple sequence signal such as open reading frame or codon usage bias. So the problem is addressed by a variety of complementary approaches.

- **Homology search based gene finders**
  - The first method consists in searching a sequence of ncRNA against a trusted ncRNA database. In this context, all tools presented in Section 1.2.6 can also be used as gene prediction. Using homology both at sequence level and at structure level can identify potential occurrence of given ncRNA.
  - There also exists specialized search tools that are specific to an RNA family signature. For example, tRNAscan-SE uses a covariance model (depending on the sequence domain) to identify transfer RNA genes in a DNA sequence [66, 99]. There are other tools dedicated to other families, as for example for miRNAs [71, 122] or rRNAs [63].
- **Comparative analysis based gene finders.** Based on the same idea that structure prediction by comparative analysis: ncRNAs are under a positive selection pressure and should be better conserved than non functional sequences. Historically, this approach was used one of the first time in [93] to find likely structural RNAs regions in *E.coli*. Gene finding through comparative analysis typically combines sequence alignment (to search for similar regions across species) and structure prediction (to detect conserved secondary structures, as described in Section 1.2.5). In this context, it is relevant to be able to classify the common folding found to decide whether it is the hint of a conserved secondary structure, or whether it is found by chance. This is what is done in RNAz [123].

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

The method consists of two measures, one for RNA secondary structure conservation and one for thermodynamic stability of the sequence. The prediction of consensus secondary structure for alignment is proceeded by using RNAalifold. The two independent diagnostic features of structural ncRNAs are used to classify an alignment as "structural RNA" or "other". For this purpose, RNAz uses a support vector machine (SVM) learning algorithm which is trained on a large test set of well known ncRNAs.

- ***Ab initio* gene finders.** Some methods use essential statistical features of data like GC percentage to recognize the region of data respecting this feature. The study in [20] uses a machine learning approach to extract common characteristics (such as base composition bias), amongst known RNAs for prediction of new RNA genes in the unannotated regions of prokaryotic and archaeal genomes. Further work has been carried out in this direction in [98].

1.2.8 Comparison of RNA structures

Comparing two RNA structures involves defining and computing a similarity between two structures. This similarity is often based on a distance between arborescent structures, as trees or arc-annotated sequences, and can take into account both the primary and the secondary structure. Structure comparison can be used to assert or reject the membership of an unknown RNA to a given ncRNA family, hence predicting the function, or to help evolutionary studies.

Comparison of RNA secondary structure is implemented in tools such as RNAdistance [51, 101, 100], RNAforester [55, 56, 57], Migal [3] and Gardenia [14]. To compute a distance value, it is also suggested to compare the whole folding space of two sequences through their partition function and matrix of base pairing probabilities, like in RNAPDIST [16, 51]. Finally, [2] reports a benchmark of several RNA comparison tools.

1.2.9 Tools for non-secondary structures

As mentioned previously, secondary structures are not the real 3D structure of an RNA. We now briefly mention methods and tools that go beyond secondary structures, or that combine secondary structures.

- **Prediction of RNA-RNA interactions.** Some small non-coding RNAs have a post-transcriptional regulation role: by base-pairing with mRNAs, they modify the gene expression. A particular case is miRNA interaction, for which many tools have been proposed (review in [10]). For generic RNA-RNA interactions, some methods consider *only the interaction site*, neglecting intra-RNA interactions, such as in RNAhybrid [91], RNAduplex [51] or RNAplex [109]. Other approaches take into account both intra-RNA structure and inter-RNA interactions:
  - A first idea is to compute the MFE or the partition function of the concatenation of the two RNA sequences, as in Pairfold [7] and RNAcofold [51, 12];
  - Some approaches, as RNAup [79], consider the problem in two steps, first breaking intramolecular binding of both RNAs, then adding intermolecular interaction. The energy gained from hybridization is thus the sum of the energy of the two steps;

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

   – Finally, some programs directly compute the MFE or the partition function for the duplex of the two RNAs, such as RactIP [62] (using integer programming), InteRNA [1] or Agil inteRNA [95].

  These approaches mostly use extension of classical RNA secondary structure prediction algorithms presented in the previous sections, and some of them output both optimal and sub-optimal interactions.

- **Prediction of pseudoknotted structures.** RNA pseudoknots are functionally important. There is evidence for conservation of pseudoknots throughout evolution, as for example in ribosomal RNA.

  The problem of folding a RNA structure with arbitrary pseudoknots is NP-hard [68]. However, there are polynomial algorithms on some reasonable sub-classes of pseudoknots. For example, in [92], Rivas and Eddy proposed a dynamic programming algorithm predicting pseudoknotted RNA structures in $O(n^6)$ time and $O(n^4)$ in space. This algorithm uses the Turner model, with addition of some pseudoknot specific parameters given in [121]. P-DCFold [110], based on comparative analysis, using a greedy approach enable to find simple or complex of pseudoknots in complexity $O(n^2)$.

  Kinefold [128, 58] simulates a sequential folding, taking into account the order of translation of RNA, and can produce pseudo-knots.

- **3D structures.** Some programs predict 3D structures of RNAs or complexes with RNAs. The goal can be to infer the structure with the 3D position of every atom or residue, as it could be obtained through crystallography, or to get other intermediate representation, as the Leontis-Westhof nomenclature which generalizes base pairings [64].

  An example of 3D RNA prediction is the pipeline of MC-fold and MC-Sym [85]. MC-fold takes an RNA sequence and generates a set of sub-optimal secondary structures using a Waterman-Byers algorithm [124], and MC-Sym builds 3D structures from this set of secondary structures. This method uses *nucleotide cyclic motifs* that generalize base pairing and base stacking, taking into account interactions between several nucleotides on the 3' side and several other on the 5' side.

  These programs give a precise understanding of the structure, but they are devoted to the study of short sequences. Their running time prevents their usage in scanning genomes.

1.2.10 RNA Databases

RNA sequences can be found in generic genomic databases, such as Genbank[3]. However, many non-coding RNA databases specifically gather ncRNA sequences, with further information on sequences, sequences, alignments, secondary or 3D models, and functions.

In this Thesis, I mostly used **Rfam** [39][4], a database for RNA families. Release 10.0 of Rfam (january 2010) contains 1446 "RNA families" annotating 3,192,596 genes. Each RNA family is presented in multiple sequence alignment of curated "seed" sequences representative of the family.

[3]`http://www.ncbi.nlm.nih.gov/genbank/`
[4]`http://rfam.janelia.org/`

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

The "full" set of sequences is automatically produced with Infernal [81] (covariance models using SCFGs [31], see Section 1.2.6) on regions selected by sequence similarity scans. Some of these "full" sequences may be "promoted" to "seed" sequences after a "manual verification". All "full" sequences are aligned and annotated with a consensus secondary structure. Note that this consensus structure is based on the alignment, and thus may not be present in each sequence.

There are other more specialized ncRNA databases, as for example:

- **Sprinzel tRNA Database** [103][5], and the **Genomic tRNA Database** [6] both focus on tRNAs. The second database contains predictions obtained with tRNAscan-SE [66];
- **miRBase** [45][7] is a database of microRNA sequences and their corresponding mature sequences;
- **RNaseP Database** [18][8] is a database of RNase P RNAs;
- **RDP-II** [24][9] and **SILVA** [88][10] are databases of ribosomal RNAs. The first one corresponds to bacterial and archaeal small-subunit 16S rRNA sequences. The second one contains datasets of small aligned (16S/18S, SSU) and large subunit (23S/28S, LSU) ribosomal RNA (rRNA) sequences.

1.3 Contents of the thesis

Algorithms on RNA secondary structure are far more efficient than those on pseudoknotted or 3D structures. Moreover, the secondary structure alone is often sufficient to identify and classify ncRNAs. In this thesis, I focused on **sets of secondary structures** – either putative, or real – on a same sequence that I called **RNA multi-structures**.

Indeed, even if tools presented in Section 1.2.4 can predict distinct structures, including "suboptimal" ones, no tool allows to further analyze at once a set of several structures. Anyone interested in several "candidates" in a typical mfold/unafold or RNAsuboptoutput must repeat analysis (including pattern matching) on every candidate, even if these candidates share common parts such as helices.

Moreover, as mentioned in Section 1.2.4, several RNAs, such as riboswitches, seems to change their conformations: in these cases, studying only one structure such as the optimal MFE structure is not enough to have a good understanding of the RNA. Is there any other structure with the same energy value that can yield us better informations?

To efficiently represent multi-structures, I propose to consider them as **nested levels of flat structures** (Chapter 2). This decomposition naturally follows the tree representation of RNA, factorizing common helices of the structures. I studied the two following questions:

[5] http://www.staff.uni-bayreuth.de/~btc914/search/index.html
[6] http://gtrnadb.ucsc.edu/
[7] http://www.mirbase.org/
[8] http://jwbrown.mbio.ncsu.edu/RNaseP
[9] http://rdp.cme.msu.edu/
[10] http://www.arb-silva.de/

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

- *What is the multi-structure of all relevant secondary structures of a RNA ?*

  There are often many suboptimal structures that are close to the optimal or have the same energy value as optimal, but are not topologically different.

  I choose to consider **locally optimal structures** as relevant structures to describe all possible secondary structures. Locally optimal secondary structures are secondary structures to which we can no more add a base pair. Such structures were introduced in [22], in a context limited to base pair interactions, and I generalized this definition to thermodynamically stable helices.

  Chapter 3 presents such structures, and proposes an efficient algorithm to enumerate them, starting from a set of putative helices. In a locally optimal structure, each flat structure in a nesting level is maximal in some sense. The algorithm thus uses **maximal for juxtaposition** structures as a key point.

  This algorithm has been implemented in the Regliss software. I report experimentations showing that Regliss is an useful tool to describe the energical landscape of RNA sequences.

- *How can we find all occurrences of a multi-structure into a sequence ?*

  In this pattern matching problem, a multi-structure is given through its nested levels of flat structures. The multi-structure can gather putative structures (for example suboptimal structures automatically predicted by suboptimal foldings, or locally optimal structures produced by Regliss) or real alternate structures (for RNA supporting several stable foldings).

  Chapter 4 proposes an efficient dynamic programming algorithm to locate all occurrences of such a multi-structure in a genomic sequence. Here again, the decomposition into nested levels of flat structures provide the key point for the algorithm.

  The algorithm has been implemented in the Alterna software, and several tests are reported. I have tried to show that once we know a set of putative structures of an RNA, even without knowing the actual structure, we can further annotate it on the genome and find similar RNA that may have the same function.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

Chapter 2

Defining RNA structures on base pairs and on helices

In this short Chapter, we give principal definitions that will be used in the rest of the thesis. We will show that a structure can be considered such as a set of *putative base pairs* (Section 2.1, and top of Figure 2.1), or on a set of *putative helices* (Section 2.2, and bottom of Figure 2.1). The first definition directly reflects the bracket notation, whereas the second is more abstract, allows us to consider as base elements thermodynamical stable helices.

In both cases, a structure can be decomposed into nested levels of juxtaposed elements – base pairs or helices. Following the tree descriptions of RNA, this decomposition will be essential to provide efficient algorithms to compute all locally optimal structures (Chapter 3) as well as to look for matches of alternate RNA structures (Chapter 4).

2.1 Structure on a set of base pairs

Considering a structure as a set of base pairs is the easiest way to model an RNA structure. For us, it will be mainly interesting for pedagogical purposes, as we will use it in Chapter 3 to present our algorithms that will be used later on helices.

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 $1 \leq x < y \leq n$. Given a set BP of base pairs on α, a *structure* is any subset of BP. We assume that the 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 $y < t$. We denote the empty structure by ε. We define the following relations between base pairs:

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

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

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

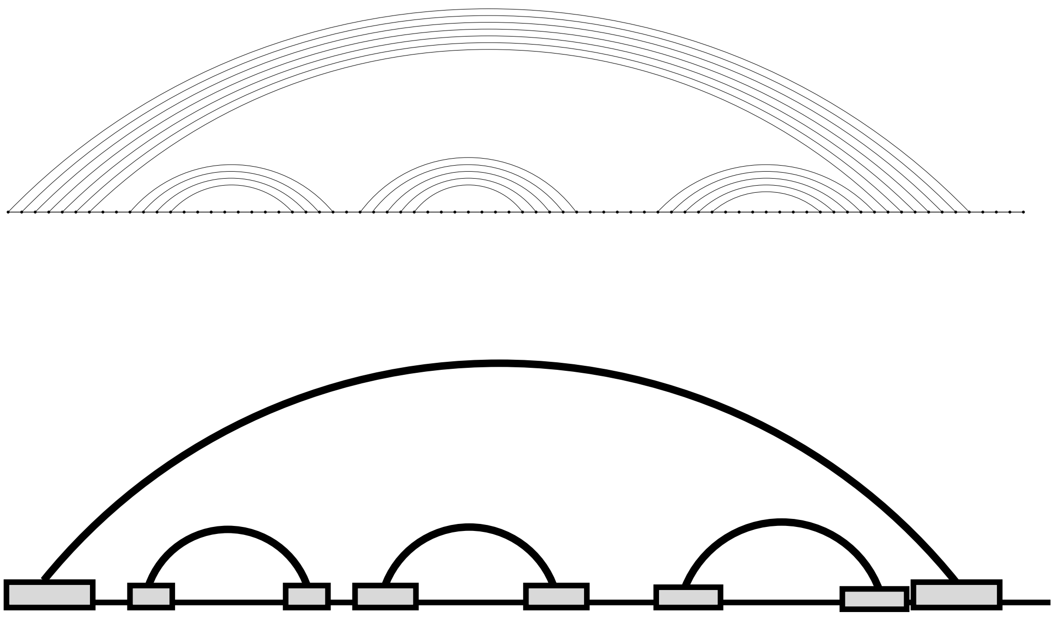


Figure 2.1: Secondary structure of transfer RNA (tRNA) Ala, seen as a set of base pairs (top) or as a set of helices (bottom). Each rectangle shows a group of adjacent bases included in a helix. Between two rectangles belonging to the same helix a gras edge is appeared.

A *secondary structure* on BP is a subset of BP such that any two distinct base pairs of S are either nested or juxtaposed. Figure 2.2 gives an example of a secondary structure with 6 base pairs.

In the rest of this thesis, a secondary structure is decomposed into nested levels of juxtaposed base pairs. For this perspective, we define the two following notations:

- Given a structure S, $\texttt{toplevel}(S)$ 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, $\texttt{nested}(x, y, S)$ 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:

$$S = \texttt{toplevel}(S) \cup \bigcup_{(x,y)\in S} \texttt{nested}(x, y, S)$$

For example, on the Figure 2.2, $\texttt{toplevel}(S) = \{(1, 6), (5, 10), (9, 14)\}$, $\texttt{nested}(1, 6, S) = \{(2, 4), (3, 5)\}$, $\texttt{nested}(5, 10, S) = \{(7, 8)\}$, $\texttt{nested}(9, 14, S) = \{(10, 11), (12, 13)\}$, and all other $\texttt{nested}$ sets are empty.

It is routine to verify that S is a secondary structure if, and only if, any two base pairs of $\texttt{toplevel}(S)$ are juxtaposed, and for each (x, y) of S, any two base pairs of $\texttt{nested}(x, y, S)$ are juxtaposed.

Finally, we also define intervals which are subsets of the base pair set. Given a set BP of base pairs, $\mathsf{BP}[x..y]$ denotes the subset of BP composed of base pairs (z, t) such that $x \le z < t \le y$.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

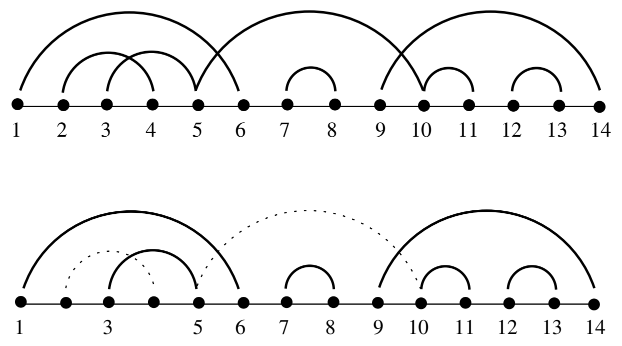


Figure 2.2: Top: sequence with a set of 8 putative base pairs. BP $= \{(1,6),(2,4),(3,5),(5,10),(7,8),(9,14),(10,11),(12,13)\}$. Bottom: a secondary structures on this BP set. $S = \{(1,6),(3,5),(7,8),(9,14),(10,11),(12,13)\}$.

2.2 Structure on a set of helices

I admit here a very general formal definition for *helices*. A helix can refer to a contiguous set of base pairs, but it can also contain bulges and internal loops (Figure 2.3). We do not explicitly impose constraints on the thermodynamic stability of a helix. A *helix* is simply an ordered set of base pairs $\{(x_1,y_1),(x_2,y_2),\ldots,(x_k,y_k)\}$ such that $x_1 < x_2 < \cdots < x_k$, and $y_1 > y_2 > \cdots > y_k$. The *5' side* of the helix is the set of positions $x_1,\ldots,x_k$, and the *3' side* 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' side, the last position of the 5' side, the first position of the 3' side and the last position of the 3' side respectively: $f.5start = x_1$, $f.5end = x_k$, $f.3start = y_k$ and $f.3end = y_1$.

Given two distinct helices f and g, we now define 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).

The concepts of structures and secondary structures can easily be adapted on helices:

Definition 1 (Structures on helices) *Let α be a sequence,* BP *be a set of base pairs on α, and* H *be a set of helices on* BP*.*

- *A* structure on H *is any subset of* H*. Given a structure S on* BP*, we say that S is* described *by the structure $\{f_1,\ldots,f_k\}$ of* H*, if $S = f_1 \cup \cdots \cup f_k$;*
- *A* secondary structure on H *is any subset $\{f_1,\ldots,f_k\}$ of* H *such that any two helices of $\{f_1,\ldots,f_k\}$ are either nested or juxtaposed.*

In this definition, it is worth emphasizing that each secondary structure on H describes a unique secondary structure on the set of base pairs BP. But the correspondence between helix secondary structures and base pair secondary structures is not always a one-to-one mapping.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

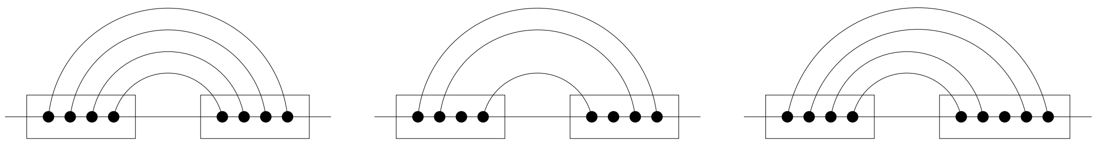


Figure 2.3: Examples of helices: contiguous set of base pairs, with an internal loop, and with a bulge. We gather all these cases in an unique type of helices, keeping only the positions (.5*start*, .5*end*, .3*start* and .3*end*).

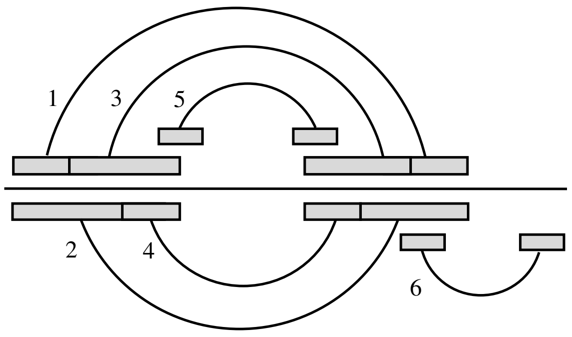


Figure 2.4: Ambiguous set of helices. This helices set contains five helices, numbered from 1 to 5. Each rectangle shows a group of adjacent bases included in a helix. Between two rectangles belonging to the same helix a gras edge is appeared. The union of helices 1 and 3 gives the same set of base pairs as the union of helices 2 and 4. Thus $\{1, 3\}$ and $\{2, 4\}$ are two descriptions of the same secondary structure (on base pairs).

Indeed, some secondary structures on BP can be described by several different secondary structures on H. This is due to the fact that several nested helices can combine to form a new helix. We give such an example in Figure 2.4.

As before, for a structure F on H, $\texttt{toplevel}(F)$ is defined as the set of helices of F that are not nested in any helix of F and $\texttt{nested}(f, F)$ is the set of helices of F that are nested in the helix f, and are not nested in any other helix which is already nested in f.

Finally, we consider structures in which all helices are pairwise juxtaposed:

Definition 2 (Flat structure) *A set of helices $w \subset H$ is a* flat structure *if, for any pair of distinct helices f and g in w, f and g are juxtaposed.*

Given a secondary structure F, the $\texttt{toplevel}(F)$ set and all $\texttt{nested}(f, F)$ sets are flat structures. In Chapter 3, we will consider “maximal” flat structures – that we call maximal by juxtaposition structures – to build locally optimal structures. In Chapter 4, we will use flat structures as a base element to describe RNA multi-structures.

2.3 Orders and intervals on a set of helices

The algorithms of the two following chapters directly work with helices. To have well-founded computation, we need to process the helix set in some order, guaranteeing that some helices are seen before others. For this, we define two total orders between helices:

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

- $\preccurlyeq$ is such that the start positions of the helices are sorted : $f \preccurlyeq g \iff f.5start \leq g.5start$,
- $\sqsubseteq$ is such that the end positions of the helices are sorted : $f \sqsubseteq g \iff f.3end \leq g.3end$.

There may be several orders satisfying these conditions. For the $\preccurlyeq$ order, one may order arbitrarily helices which have the same $5start$ position. We suppose that one of the possible order has been chosen. The same remark applies for the $\sqsubseteq$ order on helices with the same $3end$ position. Note that g is nested in f, then $f \prec g \sqsubset f$, and if g is juxtaposed with f, then $f \prec g$ and $f \sqsubset g$.

Finally, if H is an helix set, we call $H[k..\ell]$ the set of helices composed of helices whose all base pairs are in the interval $[k.5start..\ell.3end]$. Another equivalent definition is that $H[k..\ell]$ is the set of helices that are greater than k for the $\preccurlyeq$ ordering and smaller than ℓ for the $\sqsubseteq$ ordering: $H[k..\ell] = \{f \in H \mid k \preccurlyeq f \sqsubseteq \ell\}$. We also define by $\mathsf{H}]f..g[$ the subset of H composed of helices whose base pairs are all in the interval $]f.5end..g.3start[$.

For example, on the Figure 2.4, we have $\mathsf{H}[1..3] = \{3, 4, 5\}$ and $\mathsf{H}]2..2[= \{4, 5\}$.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Chapter 3

Regliss – Locally optimal structures prediction

An article with the content of this Chapter, "RNA Locally Optimal Secondary Structures", was submitted to Journal of Computational Biology *in 2010. This paper is now in second revision.*

In this Chapter, I deal with the problem of computing all locally optimal secondary structures of an RNA sequence in a non-trivial thermodynamic model that takes into account adjacent nucleotide interactions. As written in the introduction (see p. 16), computing the minimum free energy (MFE) is not enough to study the folding landscape of a RNA. It is important to be able to consider *suboptimal* foldings, that allow for a deeper insight at the RNA folding space. Several programs produce suboptimal foldings of RNA, including mfold/unafold [131, 132] and RNAsubopt [127]. However, we will see in Section 3.1 that none of these tools is suitable for an exhaustive enumeration of all relevant secondary structures. In practice, these tools are very good at finding secondary structures whose free energy is optimal, or close to the optimal. However they are not suitable to give a complete picture of folding landscape of an RNA.

We will use and extend *locally optimal secondary structures* introduced by Peter Clote in [22] (see p. 41). Locally optimal secondary structures are structures maximal for inclusion. They cannot be extended without creating 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: By definition, each secondary structure is included in a locally optimal structure. For a given RNA sequence, the set of locally optimal structures is the smallest set of secondary structures encapsulating all secondary structures. Clote proposed a dynamic programming algorithm to count all locally optimal secondary structures of a given RNA sequence within a range of numbers of base pairs less than the number of base pairs of optimal structure [22].

The locally optimal structures are a fruitful alternative to suboptimal structures and yield a very good picture of the RNA folding space. One main limitation of Clote's work is that the algorithm is not well-suited to effectively construct the locally optimal secondary structures, and it uses the Nussinov-Jacobson model [83]. By nature this RNA folding model does not produce realistic secondary structures, and the method can not be easily extended to the Nearest Neighbor model (see p. 15) [32].

In this Chapter, we will thus introduce a novel approach to produce locally optimal secondary structures. Our locally optimal secondary structures are built up from thermodynamically stable

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

helices (that could have been produced considering any energy model). but the approach is then purely combinatorial (without considering any energy model). We suggest an efficient algorithm for this problem. This algorithm is non-trivial: it relies on decomposition into structures *maximal for juxtaposition*. As far as we know, this property has never been formulated and used to study locally optimal secondary structures.

The Chapter is organized as follows. Section 3.1 presents some background on suboptimal and locally optimal secondary structures. We then propose an efficient algorithm that computes all locally optimal structures for an RNA sequence. For pedagogical reasons, we first expose the main outlines of the algorithm for the simplistic Nussinov-Jacobson model (Section 3.2). We then explain how to adapt the algorithm to deal with thermodynamically stable helices (Section 3.3). Section 3.4 discusses the implementation of the Regliss software. The software takes, as an input, thermodynamically stable helices, and produces locally optimal structures. It runs on a publicly accessible web server: `http://bioinfo.lifl.fr/RNA/regliss`. Finally, Section 3.5 presents some experimental results.

3.1 Suboptimal foldings in popular programs

I detail here the main existing suboptimal and locally optimal RNA folding methods for secondary structure prediction. All of these methods are variations of dynamic programming algorithms presented on page 15.

mfold/unafold. One of the first methods for predicting suboptimal foldings was proposed in [131], and is implemented in the mfold and unafold [70] suite. To obtain the suboptimal folds, the algorithm returns a sample set of the foldings by considering all possible base pairs (i, j), 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 suboptimal secondary structures is intrinsically bounded by the number of possible base pairs (i, j), whatever the suboptimality percentage is. So 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 3.1 exhibits such an example. Here unafold produces 13 suboptimal secondary structures, whose free energy ranges from -17.20 to -7.30. It misses some structure of free energy -10.80, even if this structure is composed of four stable stem-loops and each of these stem-loops has been identified individually by unafold.

Another drawback of the method is that the algorithm can output secondary structures that are non-relevant, because they embed an other secondary structure with a lower free energy. Figure 3.2 gives such an example. In this example, structure #2 is obtained from structure #1 with two additional base pairs, which increases the free energy level. Nevertheless, structure #2 appears in the space of suboptimal structures, because it is the optimal structure containing base pair $(33, 75)$.

RNAsubopt. 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

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

Structures output by unafold with "100% suboptimality" (free energy in kcal/mol)

```
   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)
```

Missing suboptimal structure, not found by unafold

```
   ACACAAAAGUGUGAAAAACACACAAAAGUGUAAAAUGUGAAACACACAAAAAGUGUGAAACACA
*  .((((....))))....((((......))))....((((.....))))....(((.....))). (-10.80)
```

Figure 3.1: unafold (version 3.6) output on a toy sequence (64 nt) with 100% suboptimality. 13 suboptimal structures are displayed in Vienna bracket-dot format. Even with 100% suboptimality, unafold does not find the secondary structure *, whose free energy is better than those of structure #12. This secondary structure is composed of four stem-loops. The two first stem-loops appear in suboptimal structure #2, and the two last stem-loops in suboptimal structure #3. The unafold algorithm is not able to recover the four stem-loops in a same structure, because it is composed of two suboptimal juxtaposed substructures.

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

Figure 3.2: unafold (version 3.6) output on sequence AY545598.5 (37939-38015), RF00107 (77 nt). Structure #2 contains structure #1, and has a higher energy level.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

the minimum free energy. This is done by keeping all suboptimal structures in a given energy range when backtracking in the dynamic programming matrix. This solution is implemented in RNAsubopt [127] (see p. 16). However, if the minimal energy threshold is set too low, a few number of alternate structures is produced, and if it is set too high, too many structures may be generated for the reasonable evaluation. For example, the sample sequence of Figure 3.1 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. To cope with this, one can use RNAshapes [105] (see p. 16). During the computation of the suboptimal structures, RNAshapes reduces the potential exponential number of suboptimal structures to a few *abstract shapes* that are classes of similar structures. The program returns as a representative of each class the structure with the minimum free energy.

Locally optimal secondary structures. The critical evaluation of mfold/unafold and RNA-subopt suggests that there is a need for a formal definition of relevant suboptimal secondary structures and effective algorithms to compute them. The goal is to keep relevant suboptimal structures in the free energy landscape that correspond to local minima. The notions of *saturated* structures and *locally optimal* secondary structures meet these requirements. In [133] and [33], 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. Moreover, there is no isolated base pair. In [22], a secondary structure is *locally optimal* when no base pairs can be added without creating a conflict: either crossing pairings, or a base triplet. This last definition is purely topological, and did not involve any consideration for thermodynamic stability on the computed structures. The main drawback of Clote's result [22], 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 64 nt in Figure 3.1 gives 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. Very recently, Lorentz and Clote proposed an algorithm to compute the partition function of all locally optimal structures on the Turner model [65].

The work I present in this Chapter is at the crossroad of these two lines of research. I introduce a nice topological definition of locally optimal secondary structures, which is extended for taking into account the stability of helical regions. In this context, locally optimal secondary structures are also saturated structures.

3.2 Folding at base pair resolution

We begin by considering locally optimal secondary structures for a very simple folding model: All base pairs are supposed to be independent, like in Nussinov-Jacobson model [83] implemented in Clote's algorithm [22]. This model is mainly interesting for pedagogical purposes, because it allows us to state fundamental notions and algorithms. We shall explain in Section 3.3 how to extend this approach to thermodynamically stable helices to take into account interactions between base pairs.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

3.2.1 Definitions

We use here secondary structures on base pair sets, as defined on page 25. A secondary structure is said to be *locally optimal* if no base pairs can be added without producing conflict.

Definition 3 (Strict inclusion, strict extension) *Let a sequence α, a set of base pairs* BP *on α, and 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 4 (Locally optimal secondary structure) *Given a sequence α, a set of base pairs* BP *on α and a secondary structure S, 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.*

Thus any secondary structure is included in a locally optimal structure. We give in Figure 3.3 an example of a set of base pairs and all its locally optimal secondary structures. This running example will serve to illustrate definitions and algorithms throughout this Chapter.

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 can be calculated by rephrasing the problem in terms of maximal independent sets (MIS) in graph theory. Let $\mathcal{G}$ be the graph whose vertex set is BP and where an edge links two conflicting base pairs. Then the locally optimal structures are exactly the maximum independent sets of $\mathcal{G}$. The upper bound of the number of MIS is $3^{\ell/3} = 1,44...^{\ell}$ [78]. 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, as illustrated in Figure 3.4.

However note that both problems are not strictly equivalent, as some incompatibility graphs do not encode a set of base pairs. We give a proof by counterexample in Figure 3.5. Moreover, we shall see in Section 3.3.3 that the extension of this approach to thermodynamically stable helices can not be fully seen in terms of MIS.

Our algorithm takes advantage of properties of nested and juxtaposed relations, such as transitivity, to achieve a good running time in practice. Experimental results on real RNA sequences are reported in Section 3.5, Figure 3.17. The idea is to reduce the combinatorics and to divide the construction of locally optimal structures into two steps: First applying only juxtaposition operations, then applying only nesting operations. For this, we follow the decomposition in `toplevel` and `nested` subsets presented in the previous Chapter.

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 5 (Maximal for juxtaposition) *Given a sequence α, a set of base pairs* BP *on α, and a structure S on* BP*, S is* maximal for juxtaposition *if it satisfies the two following conditions:*

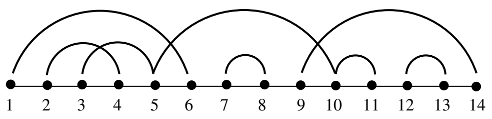
(i) if b and b' are two distinct base pairs in S, then b and b' are juxtaposed,

(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.*

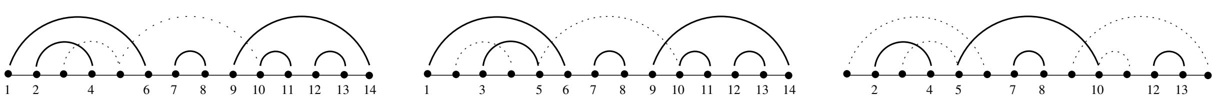
© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

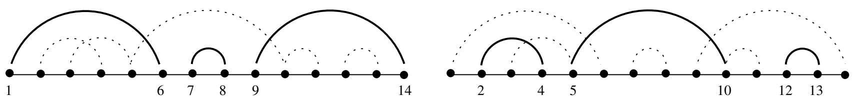
(a) Initial set of base pairs



(b) Locally optimal secondary structures



(c) Structures maximal for juxtaposition



(d) Illustration of Theorem 1. Left: structures maximal for juxtaposition, right: locally optimal structures

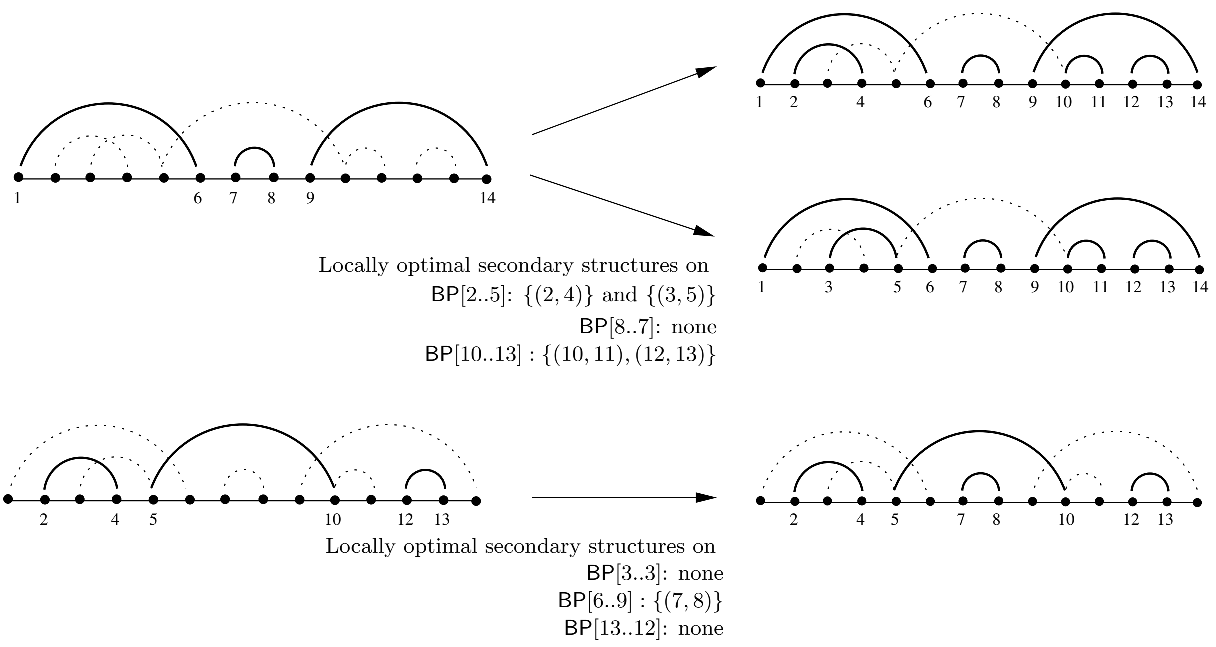

Locally optimal secondary structures on
BP[2..5]: $\{(2,4)\}$ and $\{(3,5)\}$
BP[8..7]: none
BP[10..13] : $\{(10,11),(12,13)\}$
Locally optimal secondary structures on
BP[3..3]: none
BP[6..9] : $\{(7,8)\}$
BP[13..12]: none

Figure 3.3: Example and construction of locally optimal secondary structures. (a) The sequence has eight base pairs: $(1,6),(2,4),(3,5),(5,10),(7,8),(9,14),(10,11)$ and $(12,13)$. (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) This diagram shows how to obtain locally optimal secondary structures from structures maximal for juxtaposition. 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.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

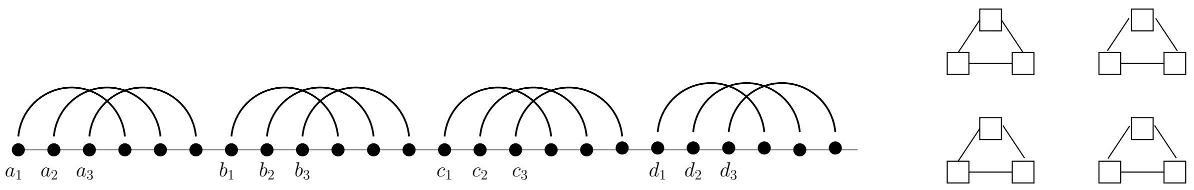


Figure 3.4: Worst-case of the number of locally optimal secondary structures. Left: In this BP set, the 12 base pairs are conflicting 3 by 3. There are exactly $3^{12/3} = 81$ locally optimal secondary structures: Any set of the form $\{a_i, b_j, c_k, d_e\}$, with $1 \leq i \leq 3$, $1 \leq j \leq 3$, $1 \leq k \leq 3$, $1 \leq e \leq 3$. Right: These locally optimal secondary structures correspond exactly to all maximal independent sets of the triangle graph. This example can be extended to any number ℓ of base pairs that is divisible by three.

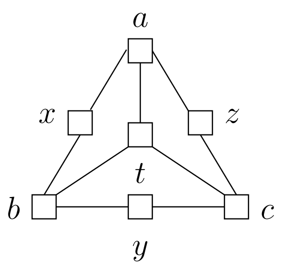


Figure 3.5: Incompatibility graph with no counterpart in terms of set of base pairs. This graph is composed of seven vertices, and no set of seven base pairs can be encoded by this graph. The three vertices a, b and c are mutually compatible. So the underlying base pairs should be either pairwise nested, or pairwise juxtaposed. Since the graph is symmetrical, we can assume without loss of generality four cases for a, b and c. Case 1: a is nested in b, and b is nested in c. Then it is not possible to have z. Case 2: a is nested in b, and b is juxtaposed with c. Then it is not possible to have z. Case 3: a is juxtaposed with b, and b is juxtaposed with c. Then it is not possible to have t. Case 4: a is juxtaposed with b, and a and b are nested in c. Then it is not possible to have t.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

Figure 3.3-(c) gives an example of structures maximal for juxtaposition. The theoretical upper bound of number of structures maximal for juxtaposition is still $3^{\ell/3}$. In the example of Figure 3.4, all locally optimal secondary structures are also maximal for juxtaposition. However in practice, there will be much fewer structures maximal for juxtaposition than locally optimal structures. We provide in Section 3.5 experimental measures on a large selection of RNA sequences (see Figure 3.17).

The link between structures maximal for juxtaposition and locally optimal secondary structures is established by Theorem 1.

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

(i) $\texttt{toplevel}(S)$ *is maximal for juxtaposition on* $\mathsf{BP}[1..n]$,

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

The proof of Theorem 1 will be given in Section 3.2.3, with Lemmas 2 and 3. Figure 3.3-(d) gives an illustration of the Theorem.

3.2.2 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 exists no base pair of the form (i, y), $i < y \leq j$, in BP, then $\mathsf{MJ}(i, j) = \mathsf{MJ}(i + 1, j)$.

3. otherwise

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

$\oplus$ 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.

Definition 6 (Filter) *Given a base pair b, and a set of secondary structures* R*, the secondary structure S of* R *is in Filter(b, R) if, and only if, there exists a base pair b' in S such that b and b' are conflicting.*

In Figure 3.6 for a set of structure R and a base pair b, there is one structure in *Filter*(b, R).

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* $\mathsf{BP}[i..j]$.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

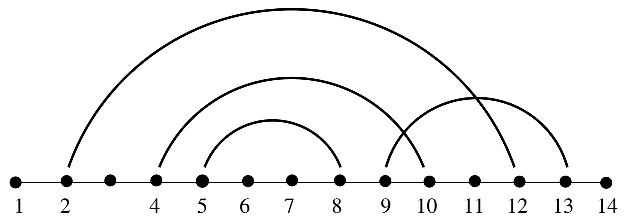


Figure 3.6: This BP set has four base pairs: $(2,12),(4,10),(5,8)$ and $(9,13)$. Let consider the set of two secondary structures $\mathsf{R} = \{\{(4,10),(5,8)\},\ \{(5,8),(9,13)\}\}$ and the base pair $b = (2,12)$. The only structure in $\mathit{Filter}(b,\mathsf{R})$ is $\{(5,8),(9,13)\}$, because $(9,13)$ conflicts with $(2,12)$. Here $\mathsf{MJ}(1,14)$ contains two maximal for juxtaposition structures: $\{(2,12)\}$ and $\{(5,8),(9,13)\}$.

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 6. 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 5 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 5. 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 $\mathsf{MJ}(i,j)$.

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 1 shows that not all n^2

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

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. In practice, this reduces the search space and running time.

The last point that we want to make here is that in the rule (3b), $\mathit{Filter}(b, \mathsf{MJ}(i,j))$ can be computed efficiently. For this purpose, we show the following Lemma.

Lemma 1 *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 $\mathit{Filter}(b, \mathsf{MJ}(i,j))$ 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 $\mathit{Filter}(b, \mathsf{MJ}(i,j))$.

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

– 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 $\mathit{Filter}(b, \mathsf{MJ}(i,j))$.

With this Lemma, if $b = (x,y)$, the computation of $\mathit{Filter}(b, \mathsf{MJ}(i,j))$ requires at most $O(y-x)$ tests for every structure $S \in \mathsf{MJ}(i,j)$.

3.2.3 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 1, stated in Section 3.2.1. The proof of this Theorem comes from the two following lemmas.

Lemma 2 *Let S be a locally optimal secondary structure on BP. Then we have the two following properties.*

(i) *`toplevel`(S) is maximal for juxtaposition on $\mathsf{BP}[1..n]$,*

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

Proof.

(i) Assume that `toplevel`(S) is not in $\mathsf{MJ}(1,n)$: It means that there exists a base pair b in BP − `toplevel`(S) such that $\{b\} \cup$ `toplevel`(S) is a secondary structure and b is not nested in any base pair of `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 `nested`(x,y,S) is not maximal for juxtaposition on $\mathsf{BP}[x+1..y-1]$. By Definition 5, this implies that there exists a base pair b in $\mathsf{BP}[x+1..y-1]$ − `nested`(x,y,S) such that $\{b\} \cup$ `nested`(x,y,S) is a secondary structure and b is not nested in any base pair of `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.

Conversely, each assembly of structures maximal for juxtaposition leads to a locally optimal secondary structure.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

Lemma 3 *Let S be a structure on* BP *such that*

(i) `toplevel`(S) *is maximal for juxtaposition on* BP$[1..n]$,

(ii) *for each* (x, y) *in* S, `nested`(x, y, S) *is maximal for juxtaposition on* 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 `toplevel`(S) are juxtaposed, and for each (x, y) in S, any two base pairs of `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 `toplevel`(S) is maximal for juxtaposition, b is necessary nested in some base pair of `toplevel`(S). If (x, y) is the base pair of S immediately nesting b, this contradicts the assumption that `nested`(x, y, S) is maximal for juxtaposition on BP$[x + 1..y - 1]$.

This establishes the proof of Theorem 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 MJ$(1, n)$,
- Each node is labeled by an element w of 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 MJ$(x + 1, y - 1)$, where (x, y) is the ith base pair of w.

This gives an effective way to compute all locally optimal secondary structures. Figure 3.7-(c) gives the three possible trees associated to the three locally optimal secondary structures of the running example of Figure 3.3. 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 3.8, and an example of run is given on Figure 3.7-(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', defined as 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 ℓ. 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.

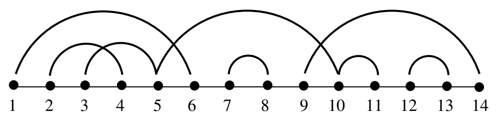
3.2.4 Back to Clote's algorithm

In Section 3.1, we mentioned the seminal work of Clote on counting locally optimal secondary structures for the Nussinov-Jacobson model [22]. This work uses a clever optimization based on

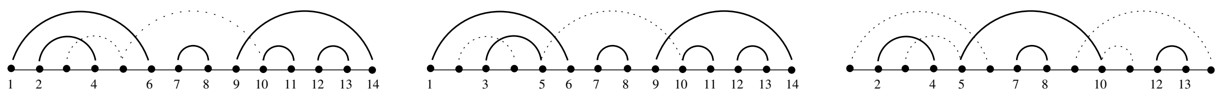
© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

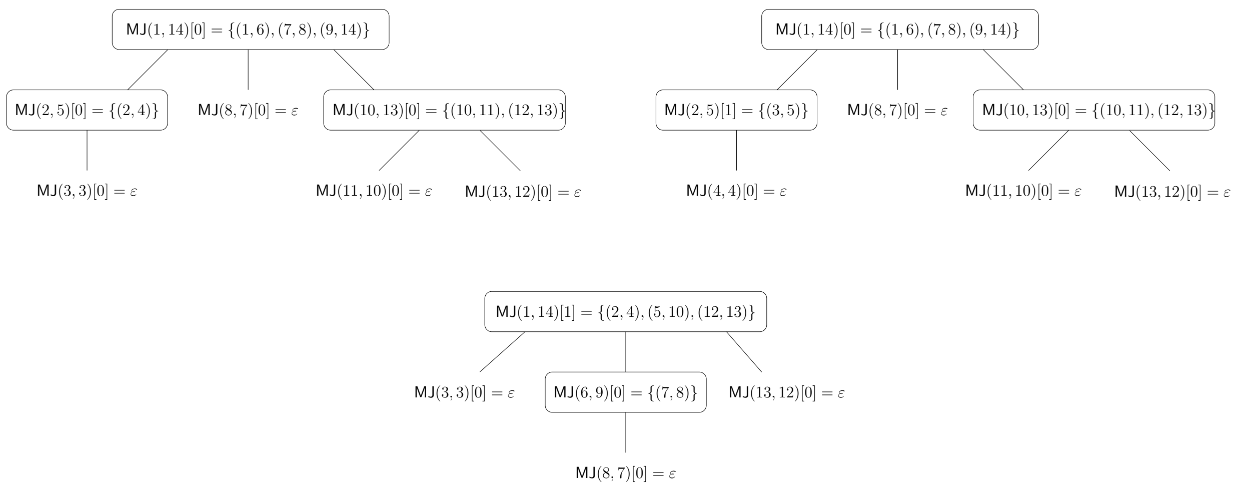
(a) Initial set of base pairs



(b) Locally optimal secondary structures



(c) All possible trees corresponding to the construction of locally optimal structures from structures maximal for juxtaposition



(d) States of the push-down stack

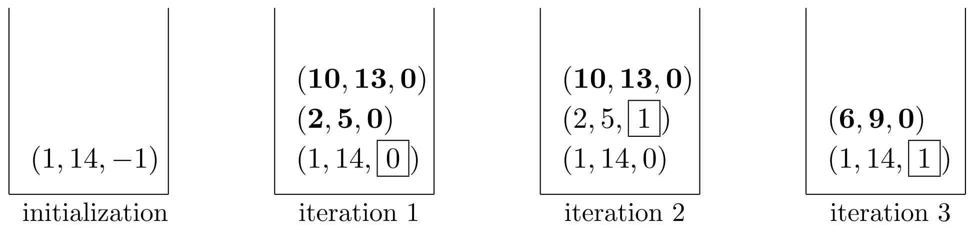


Figure 3.7: (a) and (b) come from Figure 3.3. (c) Trees associated to the three locally optimal secondary structures. If (i, j) is a base pair of BP, then MJ$(i, j)[k]$ denotes the kth element of MJ(i, j). (d) Contents of the stack at the end of each iteration of the algorithm of Figure 3.8. MJ$(i, j)[k]$ is symbolized by the triplet (i, j, k). Each iteration outputs a locally optimal structure. Each cell of the stack corresponds to an internal node of the underlying tree depicted in (c). In each iteration, the boxed element is the new MJ$(i, j)[k + 1]$, pushed by line 5 of the algorithm. All bold elements are then pushed through initializations of lines 6 and 7: their last component is always 0. For example, at iteration 1, the line 5 pushes the triplet (1, 14, 0), corresponding to MJ$(1, 14)[0] = \{(1, 6), (7, 8), (9, 14)\}$, then the nested structures are pushed.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

```
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 3.8: Enumeration, in prefix order, 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. If (i, j) is a base pair of BP, then $\mathsf{MJ}(i, j)[k]$ denotes the kth element of $\mathsf{MJ}(i, j)$ in the stack.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

the notion of *visible* bases and *visible* positions. Given a secondary structure S on α, 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 $\mathbf{v} \subset \{A, C, G, U\}$. In our framework, these sets can be computed with the following recurrence:

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

Let now consider only Watson-Crick base pairs, taking for BP, the set of possible base pairs, the following WC set:

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

The advantage of this WC set is that locally optimal structures can be counted or computed in a more 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 1) is then unchanged. The same optimization can be adapted to some larger WC 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.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

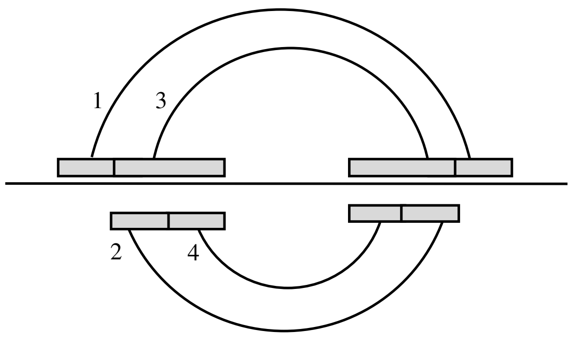


Figure 3.9: The structure $\{2, 4\}$ is strictly included in the structure $\{1, 3\}$, because any base pair in helices 2 and 4 can be found in the helix 1 or in the helix 3. Here the only locally optimal structure is the structure $\{1, 3\}$.

3.3 Folding at helix resolution

In this Section, we extend the problem 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, as introduced in the Nearest Neighbor model [73].

3.3.1 Definitions

To complete the definitions given on page 27, we introduce two more relations between helices. We suppose that two distinct helices f and g are given.

- g is *embedded* in f if any base pair of g is also a base pair of f,
- if f and g are neither nested, nor juxtaposed, nor embedded, then f and g are said to be *conflicting*.

We now use secondary structures on *helix sets*, as defined on page 27. The concepts of strict inclusion (Definition 3) and locally optimal secondary structures (Definition 4) on base pairs can easily be adapted to helices (Figure 3.9).

Definition 7 (Strict inclusion of structures) *Given two structures* $\{f_1, \ldots, f_k\}$ *and* $\{g_1, \ldots, g_j\}$ *on* 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* $g_1 \cup \cdots \cup g_j$*.*

Definition 8 (Locally optimal secondary structure on helices) *A secondary structure* $\{f_1, \ldots, f_k\}$ *on* 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, and we work with structures defined on H. We also assume that helices of H are sorted according to the $\preccurlyeq$ order defined on page 28 (according to their start position on the sequence): The helices are thus ranked from 1 to ℓ, where ℓ is the number of helices in H.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

We now turn to the problem of the construction of all locally optimal secondary structures for a set of helices. From an algorithmic viewpoint, the two-step approach described in Section 3.2 is still relevant: 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. The main outline is unchanged. However, the algorithm presented in Section 3.2 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.

We first explain how to construct structures maximal for juxtaposition in the context of structures formed by helices, and then how to derive locally optimal secondary structures.

3.3.2 Construction of structures maximal for juxtaposition

The maximal for juxtaposition structures on H are maximal flat structures, similar to maximal for juxtaposition structures on BP in Definition 5 :

Definition 9 (Maximal for juxtaposition on helices) *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 3.2, 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 $\mathsf{H}[f..g]$.

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

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

Now $\oplus$ denotes the concatenation of a helix to a set of structures, $f + 1$ denotes the next helix (wrt the helix ordering $\preccurlyeq$) after f and $\mathsf{nextJuxt}(f)$ denotes the smallest helix (wrt the helix ordering $\preccurlyeq$) juxtaposed with f. The definition of *Filter* is a straightforward translation from Definition 6 on base pairs to helices.

Definition 10 (Filter on 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 2. The proof is identical.

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* $\mathsf{H}[f..g]$*.*

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

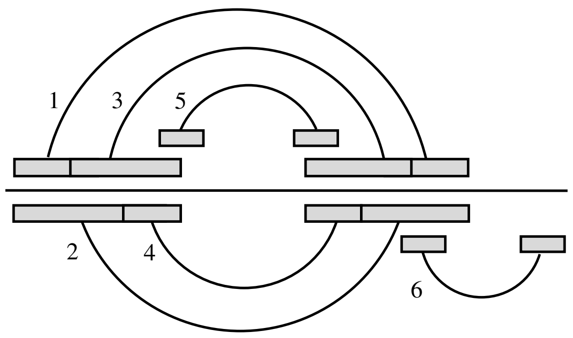


Figure 3.10: Same example than Figure 2.4 (page 28). In this ambiguous set of helices, $\{1, 3\}$ and $\{2, 4\}$ are two descriptions of the same locally optimal secondary structure (on base pairs). The other locally optimal secondary structures are $\{4, 6\}$, $\{2, 5\}$ and $\{5, 6\}$.

3.3.3 Construction of locally optimal secondary structures

We now come to the construction of locally optimal secondary structures from the set of structures maximal for juxtaposition which are composed of helices.

As noted in the previous Chapter, the first pitfall is that some base pair structures on BP can have several descriptions on the set of helices H.

For example, on Figure 3.10 $\{1, 3\}$ and $\{2, 4\}$ are two descriptions of the same locally optimal secondary structure (on base pairs). Of course, the algorithm should output only one structure.

To address this problem, we introduce the definition of *strong nestedness*. Intuitively, two helices of H are strongly nested, if each time they occur simultaneously in a locally optimal secondary structure, they can be merged into a single helix.

Definition 11 (Strong nestedness) *Let H be a set of helices, and let f and g be two helices of H, such that g is nested in f. The helix g is said to be* strongly nested *in f, if for any helix h of H which is either juxtaposed with g, or in which g is nested, then h is not nested in f.*

On the Figure 3.10, helix 3 is strongly nested in helix 1, and helix 4 is strongly nested in helix 2. Helix 5 is strongly nested in 1 and in 2 but neither in 3 nor in 4.

Definition 12 (Closure under strong nestedness) *Let H be a set of helices. We say that 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 a helix of H.*

For any set of helices H, it is easy to construct H', the closure of H under strong nestedness by iteratively adding a new helix obtained by merging two strongly nested helices until the set is closed. For example, on the Figure 3.10, the closure is obtained by adding the helix $1 \cup 3$. By construction, the set of base pair secondary structures described by helices on H is exactly the same as the set of base pair secondary structures described by helices on H', and subsequently, the set of base pair locally optimal secondary structures are identical. From now on, we assume that the set of input helices 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* locally optimal secondary structures.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

Definition 13 (Canonical structures) *Let* H *be a set of helices, and let* $\{f_1, \ldots, f_k\}$ *be a structure on* H. $\{f_1, \ldots, f_k\}$ *is a* canonical structure *on* H *if for any* i, j, $1 \le i \le k$, $1 \le j \le k$, f_i *is not strongly nested in* f_j.

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

Proof.

Assume g is not strongly nested in f. By Definition 11, 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 $\mathtt{nested}(f, F) = \{g\}$, 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.

Lemma 5 *Let* H *be a set of helices that is closed under strong nestedness, and let* $\{g_1, \ldots, g_j\}$ *be a locally optimal secondary structure on* H. *Then there exists a unique canonical structure* $\{f_1, \ldots, f_k\}$ *on* H, *such that* $\{g_1, \ldots, g_j\}$ *and* $\{f_1, \ldots, f_k\}$ *describe the same base pairs structure:* $g_1 \cup \cdots \cup g_j = f_1 \cup \cdots \cup f_k$.

Proof.

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, $\mathtt{nested}(f, F)$ contains exactly one helix: Helices should be as long as possible.

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 3.2, 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.

Lemma 6 *Let* F *be a locally optimal secondary structure on* H. *Then we have the two following properties.*

(i) $\mathtt{toplevel}(F)$ *is maximal for juxtaposition on* H,

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

The proof of Lemma 6 is basically the same as the one of Lemma 2. The point with structures formed from helices is that the reciprocal result is no longer true. Figure 3.11 shows an example where some combination of structures maximal for juxtaposition gives a secondary structure that is not locally optimal. If we consider a graph whose vertex set is H and where an edge links two conflicting helices, that means that some maximal independent sets in this graph are not locally optimal.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

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. For that, we demonstrate that for canonical secondary structures, the local optimality could be established by looking at all helices not present in the structure.

Lemma 7 *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 8, 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 Lemma 5 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.

Note that this Lemma is no longer true if the set of helices is not closed under strong nestedness, as shown in Figure 3.12. In the case of canonical secondary structures, the Lemma allows us to formulate a simple condition that guarantees that a given helix in a secondary structure cannot be replaced by an embedding helix.

Definition 14 ((⋆) condition) *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 3.11, 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 1 on base pairs is replaced by Theorem 4 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) `toplevel`(F) *is maximal for juxtaposition,*

(ii) for each helix f *of* F,

(a) `nested`(f, F) *is maximal for juxtaposition on* $\mathsf{H}]f..f[$,

(b) `nested`(f, F) *is not a single helix,*

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

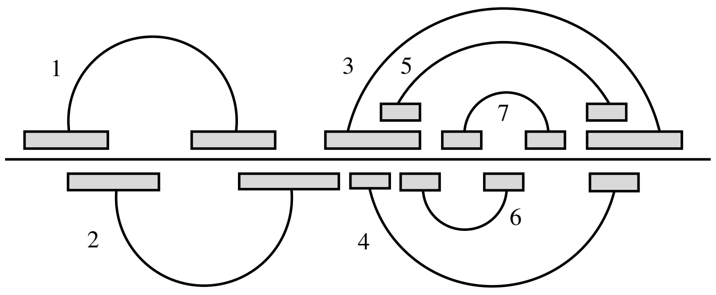


Figure 3.11: Structures maximal for juxtaposition and locally optimal secondary structures on helices. In this example, the initial set of helices H contains seven elements, that are ranked according to a helix ordering $\preccurlyeq$. Helix 5 is embedded in helix 3. There are five structures maximal for juxtaposition for 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 $\{1,3,7\}$.

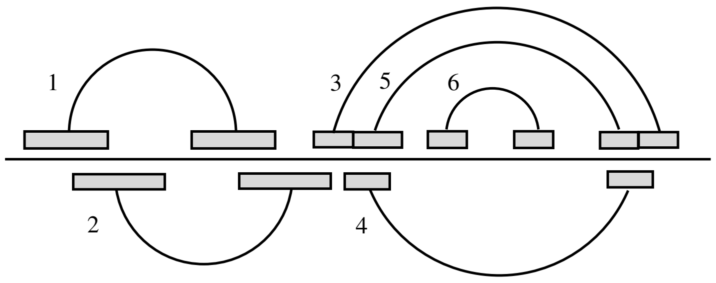


Figure 3.12: Structures maximal for juxtaposition and locally optimal secondary structures on helices. This example displays six helices. All base pairs of helix 4 are either present in helix 3 or in helix 5. The set of helices is not closed under strong nestedness: Helix 5 is strongly nested in helix 3, but $3 \cup 5$ is not present in the set of helices. Locally optimal secondary structures are $\{1,3,5,6\}$, $\{2,4,6\}$ and $\{2,5,6\}$. Note that $\{1,4,6\}$ is not locally optimal because it is strictly included in $\{1,3,5,6\}$, even if all helices not in $\{1,4,6\}$ fulfill condition of Lemma 7. This shows that the hypothesis of the helix set to be closed under strong nestedness is mandatory in this Lemma.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

(c) f fulfills the condition $(\star)$ in F.

Condition (ii)-b derives from Lemma 4 and condition (ii)-c from Lemma 7. It follows that the construction of locally optimal secondary structures from the set of structures maximal for juxtaposition can be performed using the same algorithm described in Section 3.2.3 and on Figure 3.8, based on a push-down stack whose elements are structures maximal for juxtaposition. Again, at each step of the algorithm, the stack contains a different locally optimal secondary structure. The only difference resides in lines 5 (push of an element), 6 and 7 (push of nested and juxtaposed structures), where two kind of structures maximal for juxtaposition are never pushed on the stack. These two kind of structures are, first, the structures containing exactly one helix (Condition (ii)-b of Theorem 4), and, second, the structures that do not meet the $(\star)$ condition (Condition (ii)-c of Theorem 4).

Note that checking the $(\star)$ condition can be done on a small subset of at most five helices. This subset is composed by, if they exist, the helix immediately nesting f, the immediately juxtaposed helices in the same nesting level of f, and the first and the last helices of $\texttt{nested}(f, F)$.

3.4 Implementation and availability

The main algorithm of Section 3.3 (folding at helix resolution) was implemented in C in a software called Regliss (for *RNA energy landscape and secondary structures*). The software runs on a publicly accessible web server: `http://bioinfo.lifl.fr/RNA/regliss`.

The input of Regliss is an RNA sequence in Fasta format together with a set of putative helices. The helices can be specified explicitly by the user, or computed from the RNA sequence through several methods, including detection in mfold/unafold output [132] or through a custom program.

The output is the set of all locally optimal structures, sorted according to the free energy as computed with rnaeval [51]. The user has the possibility to select only the best structures according to this energy level. For each structure, the result is available in `.ct` and Vienna bracket-dot format, and in 2D images generated by naview [19].

Finally, we produce a *energy landscape graph*, useful for visualizing at a glance all found structures. This graph yields an insight into the full energy landscape of the RNA sequence. Each vertex represents a locally optimal structure. An edge links two structures if they differ by at most two helices. The folding space presented by this graph can easily be clusterised, where each connex component is acluster of structures.

Running times. We show on Table 3.1 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 the number of output structures. When there are only some hundred structures, `regliss` runs almost instantaneously. However, as the number of structures can be exponential is the number of putative helices, `regliss` can be longer for some sequences.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

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 3.1: Running times and output size of Regliss for some RNA sequences.

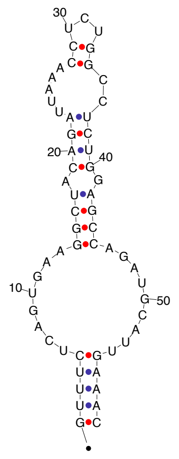


Figure 3.13: Consensus secondary structure for SECIS element Y11109.1/1272-1330, from *Oreochromis niloticus*. Source: RFAM RF00031

3.5 Experimentations

3.5.1 Example on a SECIS element

In this example, we compare the suboptimal structures of unafold and the locally optimal secondary structures of Regliss on a SECIS element. SECIS elements (selenocysteine insertion sequence) occur in messenger RNAs encoding selenoproteins [119] 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 [39]). The consensus secondary structure for this sequence is showed in Figure 3.13.

The Regliss output is highly dependent on the set of helices given in input. To fairly compare unafold and Regliss, we proceeded as follows. We first ran unafold asking "all" suboptimal structures (100% suboptimality, option `-P 100`). This gives 30 suboptimal structures. We then selected non-redundant suboptimal structures. A structure is considered as redundant if it contains another suboptimal structure output by unafold. 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.

The 30 suboptimal structures found by unafold (version 3.6) are displayed in Figure 3.14. First, we observe several predictions are redundant: structure #2 is a strict extension of structure

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

#1, structures #4 and #6 are both strict extensions of structure 3, and structure #20 is a strict extension of structure #11. A closer inspection also shows that the expected consensus secondary structure is not present in the set of suboptimal structures. With the 39 helices of unafold, Regliss generates 192 locally optimal secondary structures. Figure 3.15 shows the best 30 structures, with lower free energy as computed with rnaeval [51]. It appears that structure #14 found by Regliss is very close to the consensus structure provided in RFAM for this family. It contains exactly all base pairs of the consensus secondary structure, and contains two additional base pairs, that are likely to form with this sequence.

3.5.2 Comparison between Regliss and unafold

In this section, we analyze the output of Regliss and compare to unafold on a large number of RNA sequences from RFAM database, version 9.1 [39]. We selected all families of RFAM having sequences shorter than 200 nt, then selected five sequences randomly for each family. This gives 5308 sequences. As in the preceding example, we run unafold (version 3.6) with 100%-suboptimality, and we provide Regliss with helices coming from non-redundant suboptimal unafold structures.

Figure 3.16 compares the number of structures produced by unafold and Regliss on the 5308 sequences from RFAM. As expected, the unafold algorithm generate at most a quadratic number of suboptimal structures, even with a 100% suboptimality level, whereas Regliss produces an exponential number of locally optimal structures.

Figure 3.17 compares this exponential number against the upper bound of Section 3.2.1 and the total number of maximal per juxtaposition structures : in these real sequences, the actual number of locally optimal structures is far below the upper bound.

We evaluated the free energy of all structures with rnaeval, and looked at 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 have less 20%-suboptimality structures obtained by Regliss than by unafold,
- 25% sequences have the same number of 20%-suboptimality structures,
- 65% sequences have more 20%-suboptimality structures obtained by Regliss than by unafold.

In the first group, unafold finds more structures than Regliss. Typically, some of these structures are redundant, and are discarded by Regliss. In the second 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, and many of these structures are very simple, sometimes reduced to a single helix. Finally, the third group, the largest, shows how Regliss can, in some situations, give relevant structures which are not predicted by unafold by exploring the combinatorics of the helix input set. Regliss offers here a larger variety of structures.

3.5.3 Comparison on random sequences

In [22], Clote's results tend to prove that for some families, structural RNA has a different folding landscape than random RNA of the same dinucleotide frequency. However, even if these

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

```
     GUUUCUCAGUGAAGGCUACAGAUUAAACCUCUGGCCUCUGGAGCCAGAUGCAUUGAAAC
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)

     GUUUCUCAGUGAAGGCUACAGAUUAAACCUCUGGCCUCUGGAGCCAGAUGCAUUGAAAC
RFAM <<<<<........<<<<.<<<<.....<<...>>..>>>>.>>>>.........>>>>>
```

Figure 3.14: unafold results (all structures) for SECIS element Y11109.1/1272-1330. No structure is close to the RFAM consensus structure.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

```
     GUUUCUCAGUGAAGGCUACAGAUUAAACCUCUGGCCUCUGGAGCCAGAUGCAUUGAAAC
1    .....((((((..(((((.((.......)).)))))((((....))))..))))))...  (-15.7)
2    .....((((((.((((((.((.......)).))))))(((....)))...))))))...  (-15.7)
3    .....((((((..((((.((((.....((...))..)))).)))).....))))))...  (-13.9)
4    (((((.(((...((((((.((.......)).)))))))))..((.....))...)))))  (-13.2)
5    .....((((((.......((((.......))))...((((....))))..))))))...  (-12.6)
6    .....((((((..((((.((((.......))))........)))).....))))))...  (-12.3)
7    (((((.((....((((((.((.......)).))))))(((....))).))....)))))  (-11.9)
8    (((((..(((....))).((((.......))))((.((((....)))).))...)))))  (-11.3)
9    (((((.....((.(((((.((.......)).)))))))....((.....))...)))))  (-11.2)
10   ..(((.(((...((((((.((.......)).))))))))))))................  (-11.1)
11   (((((.((.....(((((.((.......)).)))))((((....))))))....)))))  (-11.0)
12   (((((...(((.....)))..............((.((((....)))).))...)))))  (-10.9)
13   (((((((...((.(((((.((.......)).)))))))..)))..))))..........  (-10.7)
14   (((((.((.....((((.((((.....((...))..)))).))))...))....)))))  (-10.5)
15   .....((((((..((.((.....))..))((((((.......))))))..))))))...  (-10.5)
16   .....((((((.((........)).....((((((.......))))))..))))))...  (-10.4)
17   .((((.....))))....((((.......))))((.((((....)))).))........  (-10.2)
18   (..(((..((..((((((.((.......)).)))))).....)).)))..)........  (-9.9)
19   (((((((.....((((((.((.......)).))))))...)))..))))..........  (-9.7)
20   .....(((((....((..((((.......))))...((((....)))).)))))))...  (-9.7)
21   (((((.(((...(((.((.....))..))))))((.((((....)))).))...)))))  (-9.6)
22   .....((((((.......((.........((((((.......))))))))))))))...  (-9.11)
23   ((((.((.(((.....))).))..)))).((((((.......))))))...........  (-9.0)
24   (((((.((.....((((.((((.......))))........))))...))....)))))  (-8.9)
25   ((((.((.(((.....))).))..)))).((..((.((((....)))).))...))...  (-8.9)
26   .((((.....))))........((((...((((((.......))))))....))))...  (-8.9)
27   (((((.(((((.....))).........((((((.......))))))))....)))))  (-8.6)
28   ........((..((((((.((.......)).))))))(((....)))..........))  (-8.6)
29   .....((((((..((((..........((..........)))))).....))))))...  (-8.6)
30   (..(((..((((.(((((.((.......)).)))))))....)).)))..)........  (-8.4)

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

Figure 3.15: Regliss results (30 best structures) for SECIS element Y11109.1/1272-1330 with helices coming from unafold prediction. Structure # 14, not found by unafold, exhibits a folding that is very close to the consensus structure provided by RFAM for this family.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

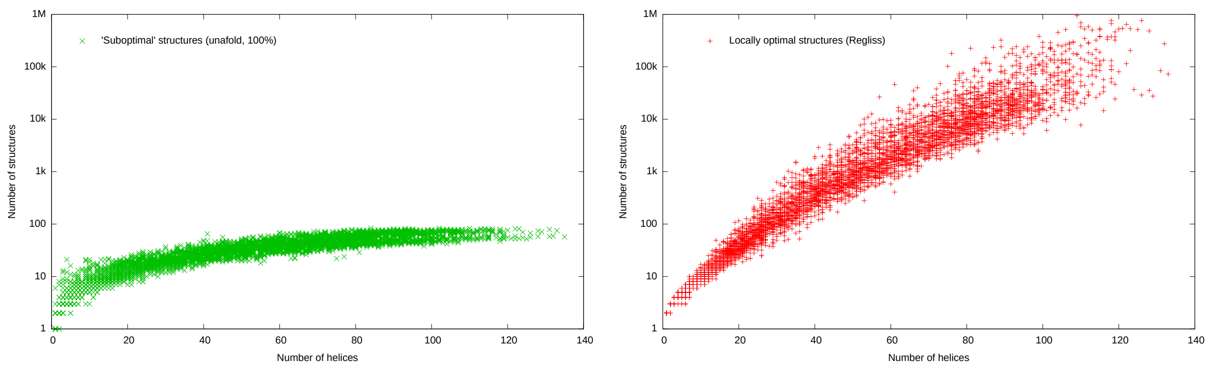


Figure 3.16: Number of structures produced by unafold (100% suboptimality, left) and Regliss (putative helices extracted from unafold output, right) on 5308 sequences from RFAM. unafold never generates more than 84 structures.

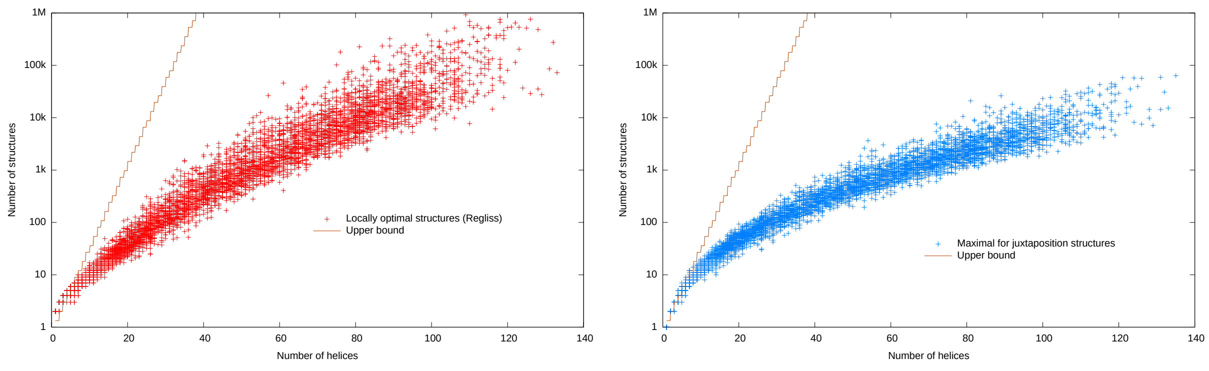


Figure 3.17: Comparison between the $3^{\ell/3}$ upper bound and the actual number of locally optimal structures (left, same data than Figure 3.16) and maximal per juxtaposition structures (right) on 5308 sequences from RFAM. On average, there are 2.98 times more locally optimal structures than maximal for juxtaposition structures.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

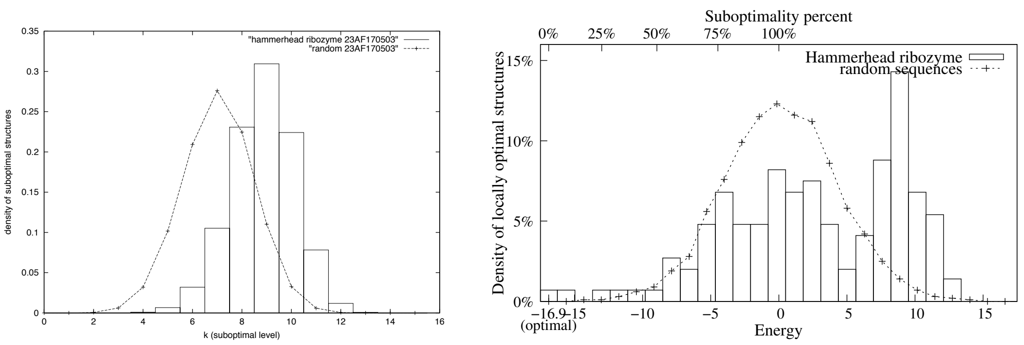


Figure 3.18: Density of locally optimal secondary structures of Hammerhead type III ribozyme 23AF170503 from RFAM (54 nt) versus average density off all locally optimal secondary structures of 100 random RNAs of same dinucleotide frequency and same length. Left: RNALOSS results (Figure from (Clote, 2005a)), right: Regliss results.

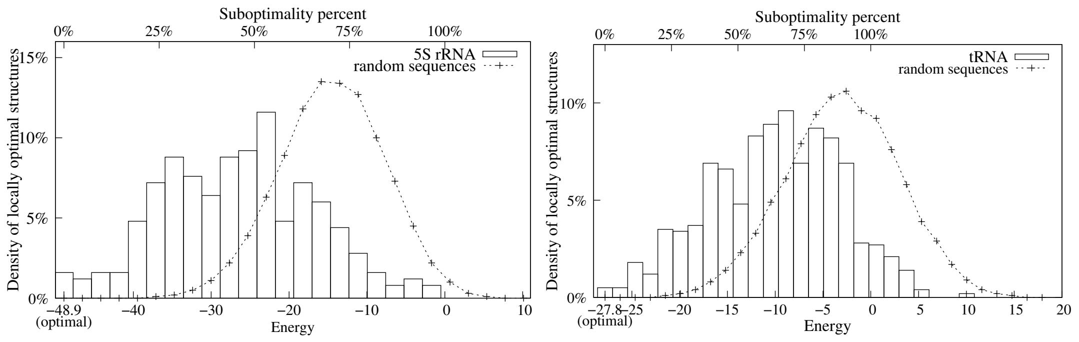


Figure 3.19: Density of locally optimal secondary structures versus average density off all locally optimal secondary structures of random RNAs of same dinucleotide frequency and same length (Regliss results). Left: 5S (C. symbiosum, RFAM RF000001, DQ397844.1/16860-16979), right: tRNA (E. coli Cysteinyl-tRNA, RNA STRAND PDB_00313).

distributions are different, this distribution of structures is not enough, in the general case, to predict that a sequence has a true RNA structure.

We reproduce here this experimentation using Regliss. We used the sequence of a Hammerhead type III ribozyme sequence, that is also used in [22]. 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[11] that implements the dinucleotide shuffle algorithm described in [4]. We then compared the distributions of locally optimal secondary structures between these randomized sequences and the initial sequence. Result is shown on Figure 3.18. Graphs obtained with Regliss are even more convincing that those obtained with RNALOSS. Figure 3.19 shows other graphs obtained with Regliss, on a 5S and on a tRNA sequence. Again, this tend to confirms that the folding landscapes, seen as the distribution of locally optimal structures, are different between structural RNAs and random sequences.

[11]http://clavius.bc.edu/clotelab/RNAdinucleotideShuffle

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

3.6 Conclusion

In this Chapter, I introduced a novel approach to produce locally optimal secondary structures of an RNA sequence. Our algorithm is based on a topological definition of locally optimal structures. The dynamic programming algorithm is based on the intrinsic relations of RNA helices: juxtaposition and nesting, which enable us to break down the complexity of 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 mfold/unafold suboptimal structures. Another advantage of the method is that the user can provide its own set of helices, based on the thermodynamic Nearest Neighbor model or any other model. The set of locally optimal secondary structures produced by our tool, Regliss, brings a new look into the folding space of an RNA sequence.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Chapter 4

Alterna – Alternate RNA structure pattern matching

An article with the content of this chapter, "Searching for Alternate RNA Secondary Structures in Genomic Sequences", is in preparation to submit to Algorithms for Molecular Biology.

In this Chapter, we address the problem of finding homologs for an RNA sequence when the signature for the family is given by *a set of alternate secondary structures.* This set of secondary structures can represent variations of the consensus structure in several species. It may also consist of alternative stable folding states: For example, riboswitches undergo structural changes upon binding with other molecules [6]. It can be a set of putative suboptimal secondary structures obtained by free energy minimization or any other folding method. In this last case, the set of candidate secondary structures helps to identify homologous sequences although the correct structure is not known, which happens for example when the number of available sequences for the RNA family is too low to determine an accurate common secondary structure by comparative analysis. Finally, the set can even be the set of all locally optimal structures, as computed by Regliss in the previous Chapter.

To deal with the set of alternate secondary structures, we introduce the novel concept of *RNA multi-structures* that represent a set of RNA secondary structures in a compact and non redundant way. These multi-structures can be seen as flexible pattern descriptors that take into account structural information, as well as constraints on the distance between helices or the length of sequence.

I thus propose here a dynamic programming algorithm to search for RNA multi-structures in genomic sequences, considering both the pattern and the text as an ensemble of helices that could be provided by any folding method. As in the previous Chapter, the algorithm relies on the decomposition of structures in nesting sets of flat helices, and processes at once common helices in the different structures of the multi-structure.

The Chapter is organized as follows. Section 4.1 gives some definitions and states the problem. The algorithm to match a RNA multi-structure into a set of text helices is presented in the Section 4.2. The Section 4.3 proposes further constraints that give more relevant results. Section 4.4 discusses the implementation of the Alterna software, whose source code is freely available. Finally, in the Section 4.5, I report experiments on the search of several non-coding RNA families on the chromosome 1 of the human genome. In some cases, the algorithm is able

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

to detect sequences with similar foldings of a given RNA, even if the actual structure of this RNA is not known.

4.1 Multi-structure matching

In this chapter, we consider RNA patterns built on helices and that contain alternate secondary structures generated from combinations of *flat structures*, as defined on page 28. We denote $w = h_1 \circ \cdots \circ h_q$ the flat structure containing the helices $h_1, \ldots, h_q$ such that h_{i+1} is juxtaposed with h_i for any $1 \leq i \leq q-1$. A *suffix* of w is any flat structure of the form $h_i \circ \cdots \circ h_q$ for any $1 \leq i \leq q$. A *proper suffix* of w is a suffix of w not equal to w.

It is clear that any secondary structure can be decomposed into a union of disjoint flat structures. Moreover, this decomposition is unique as soon as we ask that any helix h of the secondary structure has at most one flat structure nested in h and not nested in any helix nested in h. A flat structure w is *nested* into a helix h if all helices of w are nested into h. In this situation, the flat structure contains all helices connected to the multibranch loop closed by h. Note that a flat structure can be empty (denoted by λ), and by convention the empty flat structure is nested into any helix. A *multi-structure* is defined as a set of alternative flat structures.

Definition 15 (Multi-structure) *A* multi-structure *is a triplet $\mathcal{M} = (H, W, N)$, where*

- *H is a set of helices,*
- *W is a set of flat structures on H, containing the empty flat structure λ,*
- *N is the* nesting function*, mapping every helix h of H to a non-empty subset $N(h)$ of W such that every flat structure of $N(h)$ is nested in h.*

We denote by `toplevel`$(\mathcal{M})$ the set of flat structures of W that are not found in any $N(h)$ set. A multi-structure represents a set of secondary structures in the following way: Start from a flat structure w in `toplevel`$(\mathcal{M})$, for each helix h of w, choose a flat structure in $N(h)$, then go on recursively until reaching the empty flat structure λ. We call these secondary structures *instances* of the multi-structure.

Definition 16 (Instance of a multi-structure) *Let $\mathcal{M} = (H, W, N)$ be a multi-structure. An* instance *of $\mathcal{M}$ is a subset W' of W such that W' contains exactly one flat structure of* `toplevel`$(\mathcal{M})$ *and, for every helix h appearing in a flat structure of W', $N(h) \cap W'$ contains exactly one flat structure. By extension, the set of helices composing the set of flat structures is also called an instance of the multi-structure.*

It is straightforward to verify that any instance of a multi-structure is a secondary structure. We give a series of examples to explain how to build multi-structures for real sequences in practice.

- The first possibility is to start from a set of thermodynamically stable helices H, and to consider *all possible secondary structures* that can be built from this set. This is what is shown on Figure 4.1 for the human mitochondrial tRNA sequence. W_0 is the set of all possible flat structures built from H and the nesting function N_0 is defined by: For each

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

helix h in H, w of W_0 is in $N_0(h)$ if, and only if, w is nested in h. By construction, the set of instances coincides exactly with the set of secondary structures on H. In many cases however, this set is too large and may contain non-relevant instances that are very unlikely to form, because they contain very few stable helices. Using the results of the previous chapter, we can naturally restrict the set of flat structures to *maximal for juxtaposition* structures. This is what is done on Figure 4.2, where the number of instances is reduced to 5. Then all the locally optimal structures are included in the set of instances of the multi-structure. The inclusion may be not strict, if some conditions of Theorem 4 are not met for some instances.

- Multi-structures can also be used to encode a *pre-defined set of secondary structures*. Figure 4.3 shows how to encode the set of suboptimal secondary structures for the same tRNA sequence. The initial set of structures contains 4 suboptimal secondary structures (given by mfold with 20% suboptimality), instead of 5 in Figure 4.2: The secondary structure $\{1, 4, 8\}$ is lacking. To build the new multi-structure, we retrieved all flat structures present in the 4 input secondary structures. By doing this, the input secondary structures are guaranteed to appear in the set of instances. The multi-structure $\mathcal{M}_2$ has 4 instances, which are exactly the 4 input secondary structures.

- Lastly, Figure 4.4 explains how to model an *RNA family* with some variation in the consensus secondary structure. It displays the multi-structure built for bacterial RNase P RNAs based on the two main types, classes A and B. It is worth mentionning that the multi-structure allows for instances that are not present in the initial structures (class A and class B). However, these variants are functional since they are observed in some species (see Figure 4.5).

We are now interested in identifying RNA multi-structures in genomic sequences. We suppose that the text is given by a set of putative helices K, that have been obtained by preprocessing the genomic sequence. The pattern is a multi-structure $\mathcal{M} = (H, W, N)$. We define formally what is an *occurrence* of a multi-structure in the text.

Definition 17 (Structural mapping) *Let H and K be two sets of helices. A* structural mapping *from H to K is an injective function $\phi : H \mapsto K$ such that, for any two helices h and h' in H,*

- *if h is nested in h', then $\phi(h)$ is nested in $\phi(h')$,*
- *if h is juxtaposed with h', then $\phi(h)$ is juxtaposed with $\phi(h')$.*

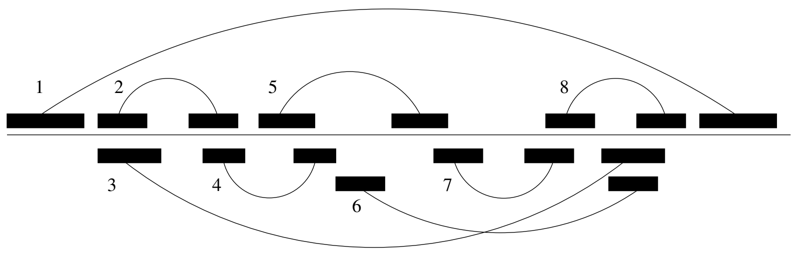
Definition 18 (Occurrence of a multi-structure) *Let $\mathcal{M} = (H, W, N)$ be a multi-structure, and let K be a text defined by a set of helices. An* occurrence *of $\mathcal{M}$ in K is a pair (H', ϕ), where H' is a subset of H and ϕ a structural mapping from H' to K.*

Note that this definition allows for approximate occurrences with errors, since we do not require that all helices of the pattern appear in the text. Note also that in this general definition, we do not specify whether one instance or all instances of the multi-structure should appear in the text. This gives rise to two distinct searching problems, that are formalized as follows.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

(a) Set of helices H



```
           GUAAAUAUAGUUUAACCAAAACAUCAGAUUGUGAAUCUGACAACAGAGGCUUACGACCCCUUAUUUACC
helix 1    (((((((......................................................))))))).
helix 2    .........((((.....))))...............................................
helix 3    .........((((((((..............................)).))).)))............
helix 4    ....................(((......))).....................................
helix 5    .......................(((((((...))))))).............................
helix 6    ............................((((((................)))))).............
helix 7    ...................................((((....))))......................
helix 8    ..........................................((.((.......)).))..........
```

(b) Set of (non empty) flat structures W_0

```
{1}        (((((((......................................................))))))).
{2}        .........((((.....))))...............................................
{2,5}      .........((((.....))))..(((((((...)))))))............................
{2,5,8}    .........((((.....))))..(((((((...)))))))..((.((.......)).)).........
{2,6}      .........((((.....))))......((((((................)))))).............
{2,7}      .........((((.....)))).............((((....))))......................
{2,8}      .........((((.....)))).....................((.((.......)).)).........
{3}        .........((((((((..............................)).))).)))............
{4}        ....................(((......))).....................................
{4,7}      ....................(((......)))...((((....))))......................
{4,8}      ....................(((......)))..........((.((.......)).))..........
{5}        .......................(((((((...))))))).............................
{5,8}      .......................(((((((...)))))))..((.((.......)).))..........
{6}        ............................((((((................)))))).............
{7}        ...................................((((....))))......................
{8}        ..........................................((.((.......)).))..........
```

(c) Nesting function N_0

$$
\begin{aligned}
N_0(1) &= \{\{2\},\{2,5\},\{2,5,8\},\{2,6\},\{2,7\},\{2,8\},\{3\},\\
&\quad\ \{4\},\{4,7\},\{4,8\},\{5\},\{5,8\},\{6\},\{7\},\{8\}\}\\
N_0(2) &= \{\lambda\}\\
N_0(3) &= \{\{4\},\{4,7\},\{5\},\{7\}\}\\
N_0(4) &= \{\lambda\}
\end{aligned}
\qquad
\begin{aligned}
N_0(5) &= \{\lambda\}\\
N_0(6) &= \{\{7\}\}\\
N_0(7) &= \{\lambda\}\\
N_0(8) &= \{\lambda\}
\end{aligned}
$$

Figure 4.1: Multi-structure for the Human mitochondrial tRNA sequence (FJ004809.1/12137-12205 – source RFAM, RF0005). In this example, we start from a set of thermodynamically stable helices, and we want to encode all possible secondary structures. (a) The set of helices H contains eight elements, numbered from 1 to 8 from 5' to 3', following the $\preccurlyeq$ ordering: $H = \{1,2,3,4,5,6,7,8\}$. Helices are written in Vienna bracket-dot format. (b) W_0 is the set of all possible flat structures that can be built form H: This gives 16 flat structures. (c) The nesting function N_0 maps every helix h of H to all flat structures that are nested in h.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

(a) Set of helices H

```
          GUAAAUAUAGUUUAACCAAAACAUCAGAUUGUGAAUCUGACAACAGAGGCUUACGACCCCUUAUUUACC
helix 1   (((((((......................................................))))))).
helix 2   .........((((.....)))).......................................................
helix 3   .........((((((((..............................)).))).)))............
helix 4   ....................(((......))).....................................
helix 5   .......................(((((((...))))))).............................
helix 6   ............................((((((................)))))).............
helix 7   ...................................((((....))))......................
helix 8   ..........................................((.((.......)).))..........
```

(b) Set of (non empty) flat structures W_1

```
{1}       (((((((......................................................))))))).
{2,5,8}   .........((((.....)))).(((((((...)))))))..((.((.......)).))..........
{2,6}     .........((((.....))))......((((((................)))))).............
{3}       .........((((((((..............................)).))).)))............
{4,7}     ....................(((......)))...((((....))))......................
{4,8}     ....................(((......)))..........((.((.......)).))..........
{5}       .......................(((((((...))))))).............................
{7}       ...................................((((....))))......................
```

(c) Nesting function N_1

$$\begin{array}{lcl} N_1(1) & = & \{\{2,5,8\},\{2,6\},\{3\},\{4,8\}\} \\ N_1(2) & = & \{\lambda\} \\ N_1(3) & = & \{\{4,7\},\{5\}\} \end{array} \quad \begin{array}{lcl} N_1(4) & = & \{\lambda\} \\ N_1(5) & = & \{\lambda\} \\ N_1(6) & = & \{7\} \end{array} \quad \begin{array}{lcl} N_1(7) & = & \{\lambda\} \\ N_1(8) & = & \{\lambda\} \end{array}$$

(d) All possible instances of the multi-structure $\mathcal{M}_1$

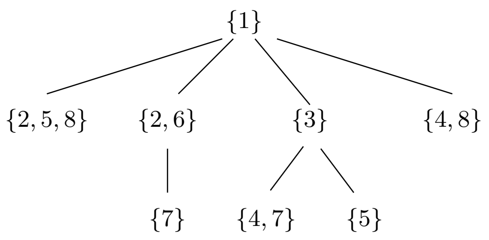
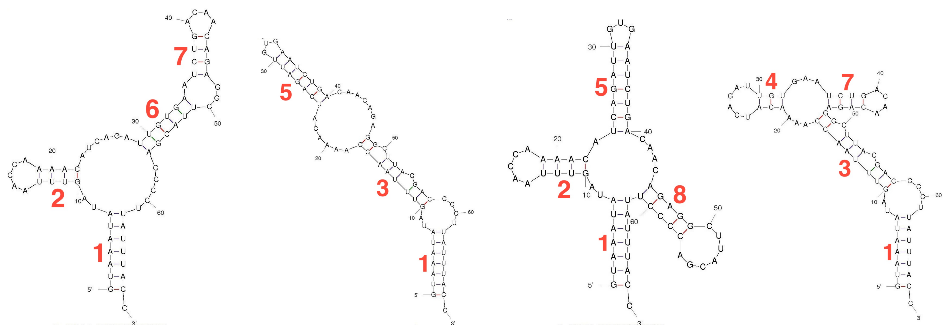


Figure 4.2: Multi-structure for the Human mitochondrial tRNA sequence, continued. (a) The set of helices H is unchanged. (b) The new set of flat structures W_1 is restricted to the maximal for juxtaposition structures. This gives 8 flat structures. (c) The new nesting function N_1 is obtained as a restriction of N_0. For a helix h, $N_1(h)$ is the subset of W_1 such that any element of $N_1(h)$ is *directly* nested in h. (d) The multi-structure $\mathcal{M}_1 = (H, W_1, N_1)$ has five distinct instances.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

(a) Input secondary structures



(b) Set of helices H

```
         GUAAAUAUAGUUUAACCAAAACAUCAGAUUGUGAAUCUGACAACAGAGGCUUACGACCCCUUAUUUACC
helix 1  (((((((......................................................))))))).
helix 2  .........((((.....))))...............................................
helix 3  .........((((((((..............................)).))).)))............
helix 4  ....................(((......)))).....................................
helix 5  .......................(((((((...)))))))............................
helix 6  ............................((((((................))))))).............
helix 7  ...................................((((....))))......................
helix 8  ..........................................((.((.......)).))..........
```

(c) Set of (non empty) flat structures W_2

```
{1}      (((((((......................................................))))))).
{2,5,8}  .........((((.....))))..(((((((...)))))))..((.((.......)).)).........
{2,6}    .........((((.....))))......((((((................))))))).............
{3}      .........((((((((..............................)).))).)))............
{4,7}    ....................(((......)))...((((....))))......................
{5}      .......................(((((((...)))))))............................
{7}      ...................................((((....))))......................
```

(d) Nesting relation N_2

$$
\begin{array}{lcl}
N_2(1) & = & \{\{2,6\},\{2,5,8\},\{3\}\} \\
N_2(2) & = & \{\lambda\} \\
N_2(3) & = & \{\{4,7\},\{5\}\}
\end{array}
\qquad
\begin{array}{lcl}
N_2(4) & = & \{\lambda\} \\
N_2(5) & = & \{\lambda\} \\
N_2(6) & = & \{7\}
\end{array}
\qquad
\begin{array}{lcl}
N_2(7) & = & \{\lambda\} \\
N_2(8) & = & \{\lambda\}
\end{array}
$$

(e) All possible instances of the multi-structure $\mathcal{M}_2$

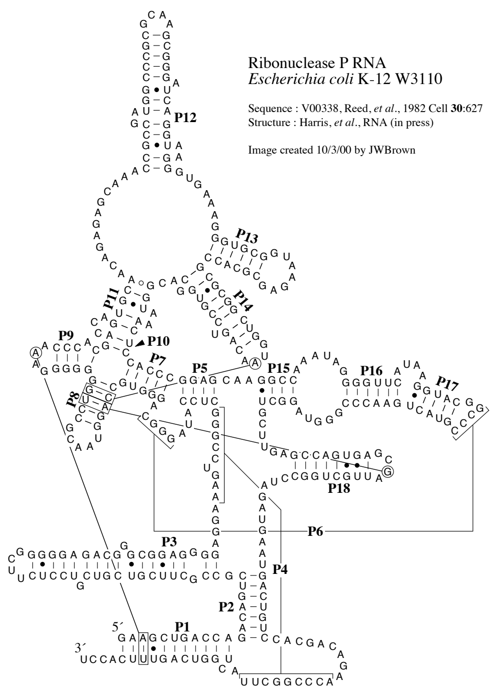
$$\begin{tikzcd}
 & \{1\} \arrow[dl, no head] \arrow[d, no head] \arrow[drr, no head] & & & \\
\{2,5,8\} & \{2,6\} \arrow[d, no head] & & \{3\} \arrow[dl, no head] \arrow[dr, no head] & \\
 & \{7\} & \{4,7\} & & \{5\}
\end{tikzcd}$$

Figure 4.3: Multi-structure for the Human mitochondrial tRNA sequence, continued. This time, we start from a set of of candidate secondary structures with low free energy level. (a) We ran the mfold software with 20% suboptimality [132], and get four suboptimal structures. The set of helices is unchanged. (b) These helices combine into seven non-empty flat structures present in the input seconday structures: $W_2 = \{\{1\},\{2,5,8\},\{2,6\},\{3\},\{4,7\},\{5\},\{7\}\}$. The flat structure $\{4,8\}$ that was present in W_1 is not in W_2, since no input secondary structure contains this combination of helices. (c) To build the nesting function N_2, we consider for each helix h the possible flat structures that are nested in h in one of the input secondary structures. (d) The obtained multi-structure $\mathcal{M}_2 = (H, W_2, N_2)$ has four distinct instances, that correspond exactly to the four initial secondary structures proposed by mfold.

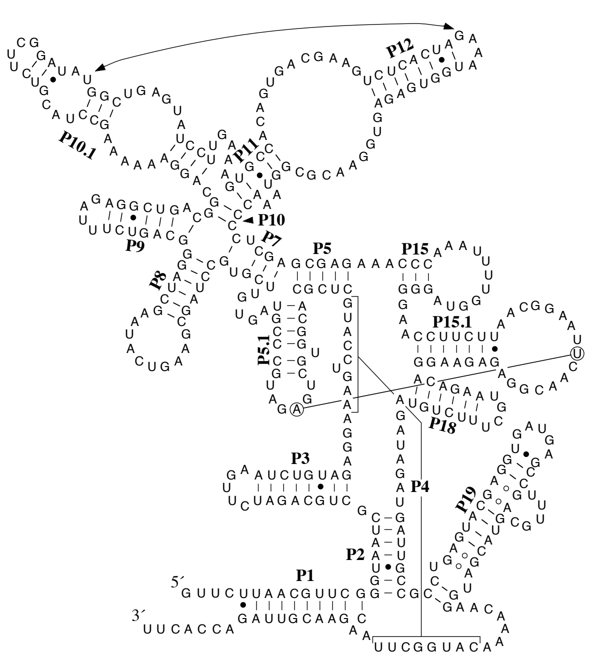
© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

(a) Secondary structures for bacterial RNase P, types A and B, from RNase P database [60]


Ribonuclease P RNA
Escherichia coli K-12 W3110
Sequence : V00338, Reed, *et al.*, 1982 Cell **30**:627
Structure : Harris, *et al.*, RNA (in press)
Image created 10/3/00 by JWBrown
P1
P2
P3
P4
P5
P6
P7
P8
P9
P10
P11
P12
P13
P14
P15
P16
P17
P18
5′
3′

Type A: *E. coli*



Type B: *B. subtilis*

(b) Nesting function N for types A and B structures

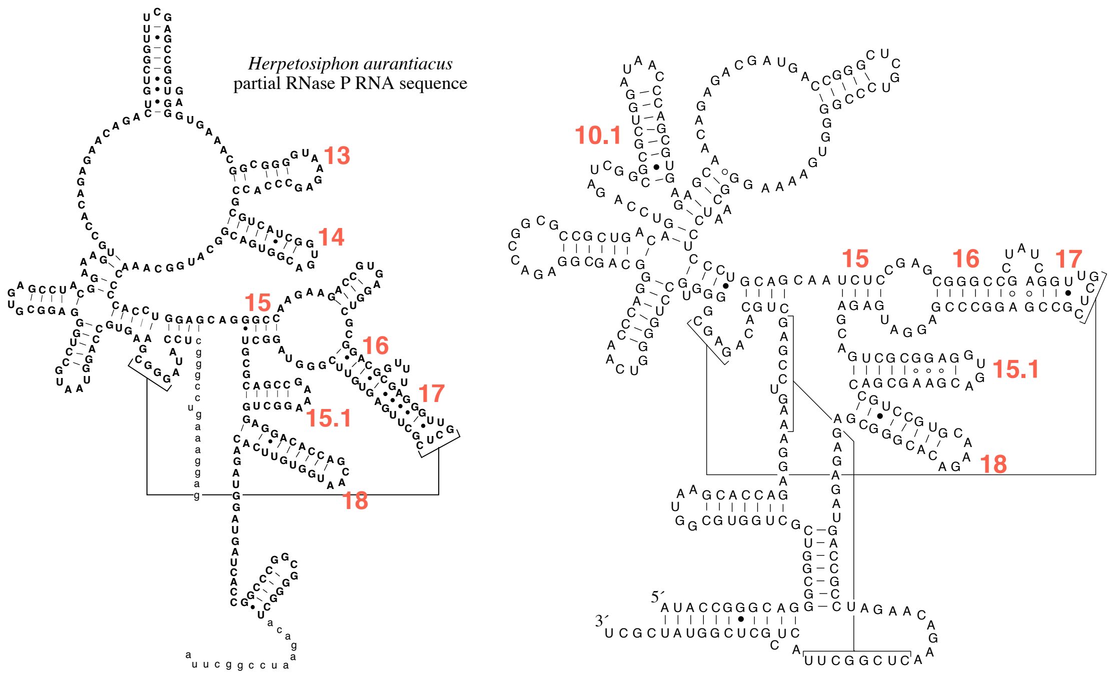
$$
\begin{aligned}
N(P_1) &= \{\{P_2\},\{P_2,P_{19}\}\} \\
N(P_2) &= \{\{P_3,P_5,P_{15},P_{18}\}, \\
&\qquad \{P_3,P_5,P_{15},P_{15.1},P_{18}\}\} \\
N(P_3) &= \{\lambda\} \\
N(P_5) &= \{\{P_7\},\{P_{5.1},P_7\}\} \\
N(P_{5.1}) &= \{\lambda\} \\
N(P_7) &= \{\{P_8,P_9,P_{10}\}\} \\
N(P_8) &= \{\lambda\} \\
N(P_{10}) &= \{\{P_{10.1},P_{11}\},\{P_{11}\}\} \\
N(P_{10.1}) &= \{\lambda\}
\end{aligned}
\qquad
\begin{aligned}
N(P_{11}) &= \{\{P_{12}\},\{P_{12},P_{13},P_{14}\}\} \\
N(P_{12}) &= \{\lambda\} \\
N(P_{13}) &= \{\lambda\} \\
N(P_{14}) &= \{\lambda\} \\
N(P_{15}) &= \{\lambda,\{P_{16}\}\} \\
N(P_{15.1}) &= \{\lambda\} \\
N(P_{16}) &= \{\{P_{17}\}\} \\
N(P_{17}) &= \{\lambda\} \\
N(P_{18}) &= \{\lambda\} \\
N(P_{19}) &= \{\lambda\}
\end{aligned}
$$

Figure 4.4: Multi-structure of the bacterial ribonuclease P RNA (RNase P). (a) Bacterial RNase P RNAs fall into two major classes that share a common catalytic core, but also show some distinct structural modules: type A (represented here by *Escherichia coli*) is the common and ancestral form found in most bacteria, and type B (represented here by *Bacillus subtilis*) is found in the low-GC content gram-positive bacteria [47]. (b) We gather the two structures (that are on different sequences) in a same multi-structure. Following the nomenclature of the RNase P database, helices are labeled $P_1 - P_{18}$ from 5' to 3' in the *E. coli* structure. Helices P_4 and P_6 are not listed, because they do not belong to the secondary structure: They form pseudoknots. *B. subtilis* features some additional helices, denoted $P_{5.1}$, $P_{10.1}$, $P_{15.1}$ and P_{19}, and lacks helices P_{13}, P_{14}, P_{16} and P_{17}. Flat structures for the multi-structure are obtained by taking flat structures of each of the two input secondary structures.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

(c) Secondary structures of green non-sulfur bacteria RNase P, from RNase P database [60]



H. aurantiacus, partial sequence *T. roseum*

Figure 4.5: Multi-structure of the bacterial ribonuclease P RNA (continued). (c) Green non-sulfur Bacteria RNase P RNAs are known to show some notable variation against the forms A and B, depicted in Figure 4.4. The majority of them (represented here by *H. auriantacus*) are of the type A class, except for the structural module $P_{18}/P_{15.1}$ that is instead quite similar to that of the type B. One exception is the represented by the sequence of *T. roseum*, which has independently converged with the class B RNAs (presence of helices $P_{10.1}$ and $P_{15.1}$, and absence of helices P_{13} and P_{14}). However, *T. roseum* RNA retains P_{16}/P_{17} and does not contain $P_{5.1}$ [47]. These two variant forms also appear as instances of the multi-structure designed to encode types A and B of Figure 4.4.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

$$T(h,k,\ell) \;=\; \boxed{\min}_{w\in N(h)} S(w,k,\ell)$$

$$\begin{aligned}
S(\lambda,k,\ell) &= 0\\
S(h_1\circ h_2\circ\ldots\circ h_q,k,\ell) &=
\end{aligned}$$

Case 1: if $K[k..\ell]$ is empty, then $||\mathcal{M}[h_1\circ\ldots\circ h_q]||$

Case 2: otherwise, if $\ell \sqsubset k$, then $S(h_1\circ\ldots\circ h_q,k+1,\ell)$,

otherwise,

$$\min\begin{cases}
\textit{Case 3-a: helix } k \textit{ of the text does not match the pattern}\\
S(h_1\circ\ldots\circ h_q,k+1,\ell)\\
\textit{Case 3-b: helix } h_1 \textit{ is not present in the text, but some helix of } \mathcal{M}[h_1] \textit{ is found in the text}\\
\min\limits_{k\preccurlyeq p\sqsubseteq\ell} 1 + T(h_1,k,p) + S(h_2\circ\ldots\circ h_q,\mathsf{firstJuxt}(p),\ell)\\
\textit{Case 3-c: No helix of } \mathcal{M}[h_1] \textit{ is found in the text}\\
1 + ||\mathcal{M}[h_1]|| + S(h_2\circ\ldots\circ h_q,k,\ell)\\
\textit{Case 3-d: helix } h_1 \textit{ of the pattern is matched with helix } k \textit{ of the text}\\
\quad T(h_1,\mathsf{firstNested}(k),\mathsf{lastNested}(k)) + S(h_2\circ\ldots\circ h_q,\mathsf{firstJuxt}(k),\ell)
\end{cases}$$

$$\begin{aligned}
||\lambda|| &= 0\\
||\mathcal{M}[h]|| &= \boxed{\min}_{w\in N(h)}||\mathcal{M}[w]||\\
||\mathcal{M}[h_1\circ\ldots\circ h_q]|| &= 1+||\mathcal{M}[h_1]||+||\mathcal{M}[h_2\circ\ldots\circ h_q]||
\end{aligned}$$

Figure 4.6: Recurrence equations to compute T and S values for **Problem A**. $k+1$ denotes the helix succeeding k for $\preccurlyeq$, $\mathsf{firstJuxt}(k)$ the first helix according to $\preccurlyeq$ that is juxtaposed with k, $\mathsf{firstNested}(k)$ the first helix according to $\preccurlyeq$ that is nested in k, and $\mathsf{lastNested}(k)$ the last helix according to $\sqsubseteq$ that is nested in k (see Figure 4.7 for an example). To solve the **Problem B**, the two $\boxed{\min}$ operators have to be replaced by $\boxed{\max}$ operators.

Problem A. Let err_A be a natural number. Find in the text K all occurrences of $\mathcal{M}$ with at most err_A errors, where the number of errors of an occurrence (H',ϕ) is defined as the *minimal* number of helices appearing in some instance of $\mathcal{M}$ and that are not in H'.

Problem B. Let err_B be a natural number. Find in the text K all occurrences of $\mathcal{M}$ with at most err_B errors, where the number of errors of an occurrence (H',ϕ) is defined as the *maximal* number of helices appearing in some instance of $\mathcal{M}$ and that are not in H'.

Problem A consists in finding in the text all positions where at least one possible instance of the multi-structure matches the putative helices of the text. This applies typically to RNA sequences that are provided with a set of potential secondary structures, such as suboptimal secondary structures automatically predicted by free energy minimization (illustrated in Figure 4.3), or to RNA families that show a variety of consensus secondary structures with some conserved features (such as RNase P RNAs in Figure 4.4). Problem B asks that all instances of the multi-structure match the same location in the text. This applies for example to RNA sequences that support several stable folding states, such as riboswitches.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

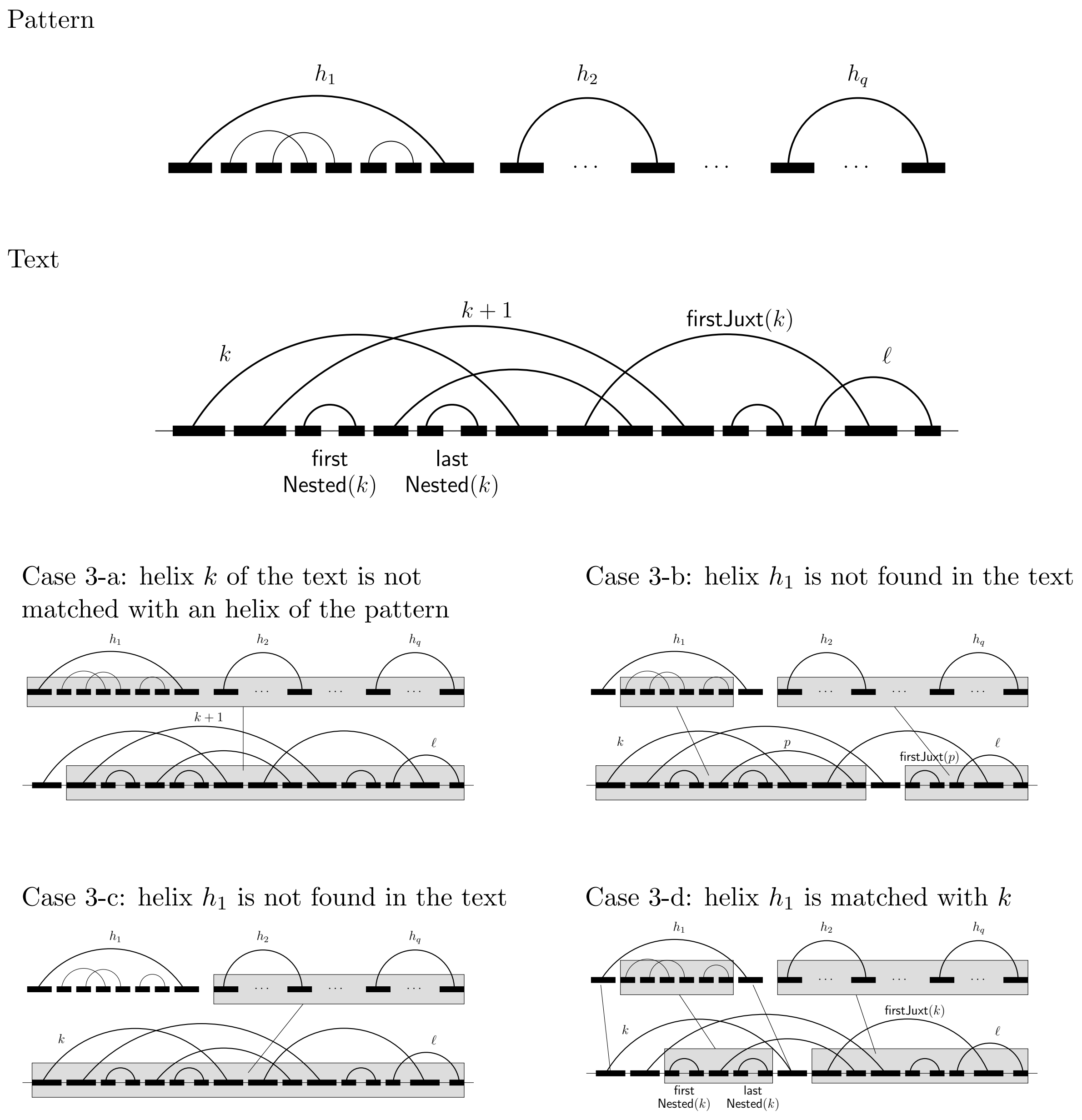

Pattern
h_1
h_2
h_q
Text
k
$k+1$
firstJuxt(k)
ℓ
first Nested(k)
last Nested(k)
Case 3-a: helix k of the text is not matched with an helix of the pattern
Case 3-b: helix h_1 is not found in the text
p
firstJuxt(p)
Case 3-c: helix h_1 is not found in the text
Case 3-d: helix h_1 is matched with k

Figure 4.7: Illustration of recurrence formulas of Figure 4.6.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

4.2 Algorithm

We present here algorithms for Problems A and B.

Restriction of a multi-structure. Given a multi-structure $\mathcal{M} = (H, W, N)$, we consider for each helix h of H the multi-structure $\mathcal{M}[h]$ obtained by restricting $\mathcal{M}$ to any flat structures nested in h that are reached by successive iterations of N. Note that the helix h does not appear in $\mathcal{M}[h]$. We also consider, for each flat structure w composed of helices of H, the multi-structure $\mathcal{M}[w]$ that is obtained by restricting $\mathcal{M}$ to any flat structures that are reached by successive iterations of N from w, including w.

As an example, if $\mathcal{M}$ is the multi-structure described on the Figure 4.3, then $\mathcal{M}[3]$ is a multi-structure with helix set $\{4, 5, 7\}$ and the two flat structures $\{4, 7\}$ and $\{5\}$.

Equation formulas. To solve the **Problem A**, let call $T(h, k, \ell)$ the minimal number of errors to match $\mathcal{M}[h]$ with text helices in the set $K[k..\ell]$, and $S(w, k, \ell)$ the minimal number of errors to match $\mathcal{M}[w]$ with text helices in the set $K[k..\ell]$. The values of S and T can be computed recursively with the equations given in Figure 4.6.

Property 1: The value

$$\min\{S(w, k_0, \ell_0), w \in \texttt{toplevel}(\mathcal{M})\},$$

where k_0 is the smallet helix of K for the $\preccurlyeq$ ordering and ℓ_0 the largest helix of K for the $\sqsubseteq$ ordering, gives the number of errors to match $\mathcal{M}$ against K, such as defined in the **Problem A**.

Proof.

We show that $S(w, k, \ell)$ is the minimal number of errors to match $\mathcal{M}[w]$ with $K[k \dots \ell]$, and $T(h, k, \ell)$ is the minimal number of error to match $\mathcal{M}[h]$ with $K[k \dots \ell]$. The proof is by recurrence on the total number of helices in $\mathcal{M}[w]$ and $\ell.3end - k.5start$.

$S(\lambda, k, \ell)$ is a basis case, where the multi-structure is empty. $S(\lambda, k, \ell)$ is the cost of matching the empty flat structure against any text.

Case 1 is another basis case, where the text $K[k..\ell]$ is empty. Then $S(h_1 \circ \dots \circ h_q, k, \ell)$ is the cost of deleting $\mathcal{M}[h_1 \circ \dots \circ h_q]$, that is one error to remove h_1, $T(h_1, k, \ell)$ errors to delete $\mathcal{M}[h_1]$ and $S(h_2 \circ \dots \circ h_q, k, \ell)$ errors to delete $\mathcal{M}[h_2 \circ \dots \circ h_q]$.

Case 2 is a degenerate case where the helix k does not belong to the set $K[k..\ell]$, and thus $K[k..\ell] = K[k + 1..\ell]$.

Case 3 correponds to the main case, and is divided into four possibilities.

Case 3-a: helix k of the text does not match the pattern. The search goes on with the remaining helices of the text: $K[k + 1..\ell]$.

Case 3-b: helix h_1 is not present in the text, but some helix nested in h_1 matches. The deletion of h_1 costs one error. The substructure $\mathcal{M}[h_1]$ is matched with $K[k..p]$ for some helix p giving $T(h_1, k, p)$ errors, and the substructure $\mathcal{M}[h_2 \circ \dots \circ h_q]$ matches the remaining of the text, $K[\mathsf{firstjuxt}(p)..\ell]$ giving $S(h_2 \circ \dots \circ h_q, \mathsf{firstJuxt}(p), \ell)$ errors.

Case 3-c: neither h_1; nor any helix of of $\mathcal{M}[h_1]$ is found in the text. The whole substructure is deleted, which costs $1 + ||\mathcal{M}[h_1]||$ errors. Then the substructure $\mathcal{M}[h_2 \circ \dots \circ h_q]$ is matched against $K[k..\ell]$.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

```
Initialization of all values in $T$ to $+\infty$
For each helix $\ell$ in the text $K$, enumerated in increasing order for $\sqsubseteq$
    $w_{\mathsf{old}} := \lambda$
    For each flat structure $w = h_1 \circ h_2 \circ \ldots \circ h_q$ in $W$, enumerated in the lexicographic ordering $\sqsubseteq_{\mathsf{lex}}$
        $s :=$ length of the longest common suffix between $w$ and $w_{\mathsf{old}}$
        For each helix $k$ in $K[1..\ell]$, enumerated in decreasing order for $\preccurlyeq$
            For $i := s + 1$ to $q$
                compute $S'[i, k] := S(h_{q-(i-1)} \circ \ldots \circ h_q, k, \ell)$ with Equations of Figure 4.6        $(\star)$
            End for $i$
            For each helix $h$ in $H$ such that $w \in N(h)$
                $T[h, k, \ell] := \min(T[h, k, \ell], S'[q-1, k])$
            End for $h$
        End for $k$
        $w_{\mathsf{old}} := w$
    End for $w$
End for $\ell$
```

Figure 4.8: Space-efficient implementation of the dynamic programming equation of Figure 4.6

Case 3-d: h_1 is matched with k. The substructure $\mathcal{M}[h_1]$ is matched with the subset $K[\mathsf{firstNested}(k)..\mathsf{lastNested}(k)]$ giving $T(h_1, \mathsf{firstNested}(k), \mathsf{lastNested}(k))$ errors, and the substructure $\mathcal{M}[h_2 \circ \ldots \circ h_q]$ is matched against the remaining of the text, $K[\mathsf{firstJuxt}(k)..\ell]$, giving $S(h_2 \circ \ldots \circ h_q, \mathsf{firstJuxt}(k), \ell)$ errors.

For the **Problem B**, similar equations hold, where the values of T are now the maximal values of the corresponding values of S (see the bottom of Figure 4.6). The number of errors to match $\mathcal{M}$ against K is now given by $\max\{S(w, k_0, \ell_0), w \in \mathtt{toplevel}(\mathcal{M})\}$.

Space-efficient implementation. It is possible to efficiently implement equations of Figure 4.6 using dynamic programming for T and S. There are mn^2 values to compute for function T, where m and n are the numbers of helices in H and K respectively. Considering S, one can note that the first parameter w is always a *suffix* of some flat structure of W. The number of all different suffixes of all elements of W is bounded by $\sigma = \sum_{w \in W} |w|$. So the total number of different values to compute for S and T is in $O(mn^2 + \sigma n^2)$.

We still have to discuss the order of execution to compute T and S. When computing $T(h, k, \ell)$, we need to access values of S of the form $S(w, k, \ell)$, where w is a flat structure of W nested in h. When computing $S(w, k, \ell)$, and assuming that all required values for T are known, we only need to access values of S that are of the form $S(w', k', \ell)$, where w' is always a suffix of w. More precisely:

- either $w' = w$ and $k \prec k'$;
- or w' is a proper suffix of w and $k \preccurlyeq k'$.

Note also that the last parameter ℓ is unchanged. This shows that it is not necessary to store all values of S: to compute $S(w, k, \ell)$, we can forget all previously computed values for S whose first parameter is not a suffix of w and whose last parameter is not ℓ. To do that, we must pay attention that some flat structures of W may share common suffixes: some $S(w', k', \ell)$ values are

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

used for distinct elements of W. To prevent redundant useless computations, it is thus necessary to order flat structures of W according to their suffixes. We define the $\sqsubseteq_{\mathsf{lex}}$ order on W as the lexicographic ordering built on $\sqsubseteq$, starting from the rightmost helices: $h_1 \circ \cdots \circ h_{q-1} \circ h_q \sqsubseteq_{\mathsf{lex}} h'_1 \circ \cdots \circ h'_{q'-1} \circ h'_{q'}$ if, and only if, $h_q \sqsubset h'_{q'}$, or $h_q = h'_{q'}$ and $h_1 \circ \cdots \circ h_{q-1} \sqsubseteq_{\mathsf{lex}} h'_1 \circ \cdots \circ h'_{q'-1}$.

It follows from these remarks helices ℓ should be enumerated in increasing order for $\sqsubseteq$, flat structures w of W in increasing order for the lexicographic ordering $\sqsubseteq_{\mathsf{lex}}$, helices k in decreasing order for $\preccurlyeq$, and suffixes of a given flat structure in decreasing order for $\preccurlyeq$.

Figure 4.8 shows the resulting algorithm, using a permanent two-dimensional table for T of size mn^2 and a temporary two-dimensional table S' for S. For a given helix ℓ and a flat structure w, we eventually store in $S'[j,k]$ the value of S for the suffix of w of size j, compared to the text $K[k..\ell]$, that is $S'[j,k] = S(h_{q-(j-1)} \circ \ldots \circ h_q, k, \ell)$. The table S' is of size $y \times n$, where $y \leq m$ is the maximal length of a flat structure in W, and n is the number of helices in K. The algorithm depicted on Figure 4.8 thus requires space $O(mn^2)$. The two following properties hold.

Property 2: In Algorithm of Figure 4.8, before the execution of the $(\star)$ line, all values $T(h_{q-(i-1)}, k, \ell)$, $T(h_{q-(i-1)}, k, p)$ with $k \preccurlyeq p \sqsubseteq \ell$, and $T(h_{q-(i-1)}, \mathsf{firstNested}(k), \mathsf{lastNested}(k))$ were previously computed.

Proof.

The computation of $T(h_{q-(i-1)}, .., ...)$ is finished when for all flat structures w' of $N(h_{q-(i-1)})$, the value of $S(w', ..., ...)$ has been previously computed.

For $S(w', k, p)$: we have either $p \sqsubset \ell$, or $p = \ell$. In the first case, the loop on ℓ ensures that $S(w', k, p)$ has been computed before. In the latter case, we have $w' \sqsubseteq_{\mathsf{lex}} h_{q_i}$, and thus $w' \sqsubseteq_{\mathsf{lex}} h_1 \circ \ldots h_q$, which implies that $S(w', k, \ell)$ has already been computed in the loop on w.

For $S(w', \mathsf{firstNested}(k), \mathsf{lastNested}(k))$: we have $\mathsf{lastNested}(k) \sqsubset \ell$. So the loop on ℓ ensures that the value has been computed before.

Property 3: In Algorithm of Figure 4.8, before the execution of the $(\star)$ line, for any j and k' such that

- either $k \prec k' \sqsubseteq \ell$ and $j \in [1, q]$,
- or $k = k'$ and $j \in [1, i-1]$,

then $S'[j, k']$ has been computed and is equal to $S(h_{q-(j-1)} \circ \ldots \circ h_q, k', \ell)$.

Proof.

At the start of each iteration of the loop on ℓ, the property is true at the first execution of the $(\star)$ line (k is the last helix for $\preccurlyeq$, and $s = 0$, thus $i = 1$). The helix ℓ being fixed, we now prove by recurrence that the property remains true for all the next executions. Suppose that the property was true at a given execution of the $(\star)$ line, and let consider the next execution of the $(\star)$ line. Let be $k' \in K[k, \ell]$ and $1 \leq j \leq q$. As shown on the Figure 4.9, there are three cases:

- If $s + 1 \leq j \leq q$:

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

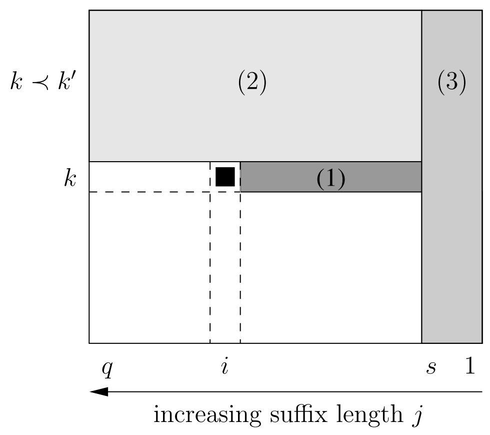


Figure 4.9: Data dependencies when computing $S'[i,k]$. The three cases (1), (2) and (3) are covered by the proof of Property 3.

  – *Case (1):* If $k = k'$, and $s+1 \leq j \leq i-1$, then the previous computation of $S'[j,k']$ was done at a previous iteration of the loop on i and concerned the correct suffix of w of size j;

  – *Case (2):* If $k \prec k'$, then no previous computation of $S'[j,k']$ was done in any iteration of this loop on i. The value of $S'[j,k']$ is thus the same as the one in a previous iteration of the loop on k, but in the same iteration of the loop on w, thus again for the correct suffix of w of size j;

- *Case (3):* If $1 \leq j \leq s$, then no previous computation of $S'[j,k']$ was done in any iteration of this loop on i, nor on k: the value $S'[j,k']$ comes from a previous iteration of the loop on w, on a different $w' = h'_1 \circ \ldots \circ h'_{q'}$ flat structure. By recurrence, this value was $S'[j,k'] = S(h'_{q'-(j-1)} \circ \ldots \circ h'_{q'}, k', \ell)$. As w and w' have a common suffix of length s, and as $j \leq s$, their suffixes of size j are the same: $h'_{q'-(j-1)} \circ \ldots \circ h'_{q'} = h_{q-(j-1)} \circ \ldots \circ h_q$. Thus $S'[j,k'] = S(h_{q-(j-1)} \circ \ldots \circ h_q, k', \ell)$.

By recurrence, this property is thus always true.

The consequence of Properties 2 and 3 is that all values needed for the computation of $S'[i,k]$ in the ($\star$) line are already computed. Therefore, the Algorithm of Figure 4.8 ultimately computes all correct values for T.

The time complexity is as follows. The loop ℓ has $O(n)$ iterations, where n is the number of helices in K, and, for each ℓ and w, the loop on k has also $O(n)$ iterations. In the worst case, there is no common suffixes between flat structures of W: for each flat structure w, the loop on i covers all helices of w, in $O(y)$ executions of the ($\star$) line. For any fixed ℓ and k, the total number of executions of the ($\star$) line is thus bounded by $\sigma = \sum_{w \in W} |w|$. Finally, each execution of the ($\star$) line takes at worst $O(n)$ time (loop on p with $k \preccurlyeq p \sqsubseteq \ell$, see Figure 4.6). Taking all together, the algorithm runs is in $O(\sigma n^3)$ worst-case time.

Note that the algorithm of Figure 4.8 computes the minimal number of errors: retrieving the actual best alignments of the pattern with the text is done through backtracking in the T table, recomputing only the relevant S' tables.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

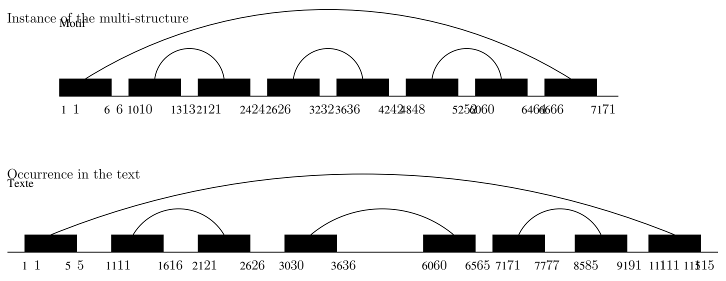


Figure 4.10: Top: second instance of the pattern for tRNA of the Figure 4.3. This instance has one helix nesting three juxtaposed helices. Bottom: exact occurrence of the pattern in the text. This occurrence may be forbidden by the helix compatibility constraint, because the opening of the first helix is significantly larger than what is expected in a tRNA.

4.3 Adding positional constraints

Until now, the text and the pattern are both defined as sets of helices, that can be nested or juxtaposed. We now explain how to enrich this topological information by taking into account supplementary constraints such as the relative location of these helices on the genomic sequence.

4.3.1 Constraint on helix compatibility

The first idea is to add compatibility constraints between helices in the text and helices in the multi-structure. For this, we suppose that a *compatibility function* γ is defined between helices of H in the multi-structure and helices of the text K: $\gamma(h, k)$ is true if one considers that the pattern helix $h \in H$ could match the putative text helix $k \in K$.

Compatibility could come from intrinsecal structural properties of the helix, such as the presence or the absence of a bulge, the presence of a motif mandatory for the function of the molecule, as the AA platform.

Compatibility could also derive from the relative positions of the helices, *h.5start*, *h.5end*, *h.3start* and *h.3end*, that can refer to the locations of helices on an actual RNA sequence or on an alignment of sequences. We then add new parameters to the problem: L_m, the length of the sequence for the pattern, L_n, the length of the sequence for the text, and d some natural number that describes the allowed variation.

$$\begin{cases} h.5start - d \leq k.5start \\ (L_m - h.3end) - d \leq (L_n - k.3end) \end{cases}$$

An additional constraint can be given on the opening of helices:

$$(h.3start - h.5end) - d' \leq k.3start - k.5end \leq (h.3start - h.5end) + d'$$

The helix-compatibility constraints can be easily implemented by a simple test on the Case 3-d on the equations of Figure 4.6. The worst-case complexity is unchanged.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

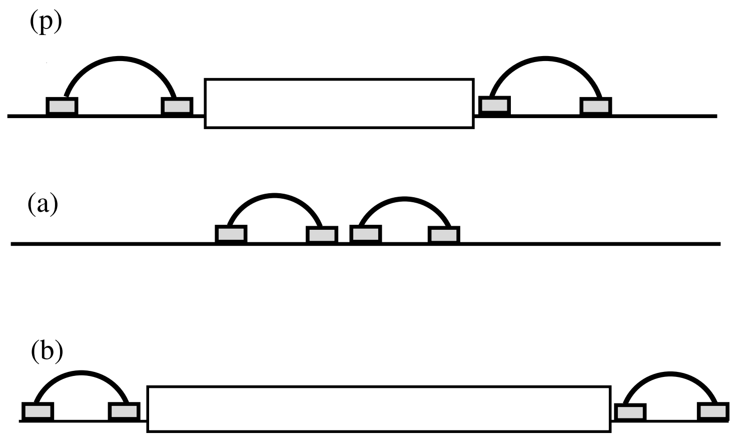


Figure 4.11: The pattern (p) has two juxtaposed helices and an unpaired region. To forbid occurrences (a) (no unpaired region) and (b) (too large unpaired region), one solution is considering constraints on the length of matching sequences.

4.3.2 Constraint on sequence length compatibility

In some RNA structures with no nesting helices or with large parts of unpaired bases, the positional constraint on helix compatibility is not enough to eliminate spurious occurrences. For example, on Figure 4.11, one may want to forbid occurrences (a) and (b). Thus the second idea is to also compare lengths of all aligned segments between the pattern and the text, although they are not formed by helices. Occurences in the text should have a sequence length smilar to that of the multi-structure. For that, we introduce the *length compatibility constraint.*

Let (H', ϕ) be an occurrence of the multi-structure in the text. Let call h_{left} the smallest helix of H' for $\preccurlyeq$, and h_{right} the largest helix of H' for $\sqsubseteq$ (note that h_{left} and h_{right} may be the same helix). We say that an occurrence satisfies the length compatibility constraint for d if

- $h_{\mathsf{left}}.5start - \phi(h_{\mathsf{left}}).5start \;\leq\; d$
- for any h in H', for any x in *5start, 5end, 3start, 3end*,
$$|(\phi(h).x - h.x) \;-\; (\phi(h_{\mathsf{left}}).5start - h_{\mathsf{left}}.5start)| \;\leq\; d$$
- $(L_m - h_{\mathsf{right}}.3end) - (L_n - \phi(h_{\mathsf{right}}).3end) \;\leq\; d$

These constraints can be implemented by refining the loop bounds of the algorithm of Figure 4.8, discarding the computation of some $S(w, k, \ell)$ values, in the same manner as is implemented the Needleman-Wunsh dynamic programming alignment between two sequences with d errors (details not shown).

4.4 Implementation and availability

The Algorithm of Section 4.2, together with an implementation of constraints on helix and segment length compatibility, is prototyped in C in a software called Alterna (for *Alternate RNA*

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

					s_{93}		s_{128}	
sequence family	species	length	hel.	alt.	5%	20%	5%	20%
SNORA61 – RF00420	*B. taurus*	127 nt	57	12	0.51	0.71	1.51	1.77
tRNA – RF00005	*D. melanogaster*	72 nt	60	20	0.90	1.02	1.78	1.94
U2 – RF00004	*M. mulatta*	192 nt	414	51	9.19	14.06	41.30	50.17

Figure 4.12: Running times, in seconds, of Alterna for patterns computed from unafold/20%, on a genomic sequence from the human chromosome 1. The "hel." and "alt." columns indicate the number of helices and the number of alternate structures for the pattern. Putative helices of the text are extracted from structures computed by unafold (options -W 0 and -P 100). The two text sequences, from the human chromosome 1, are here s_{93} (sequence of 190 bp, with 93 putative helices) and s_{128} (sequence of 330 bp, with 128 putative helices). Tests are done by using the thresholds $d = 5\%$ and 20% on sequence length compatibility.

Structure pattern matching). The sources can be downloaded on `http://bioinfo.lifl.fr/RNA/alterna`. The inputs of Alterna are:

- for the pattern, a set of helices H, together with a description of the function N;
- for the text, a set of helices K.

The program outputs the minimal number of errors, and, by backtracking, prints one alignment corresponding to the minimal error between the pattern and the text.

It is worth to note that this basic mode operation does not require any genomic sequence, but only relies on the helix sets. Options are given to take into account helix or sequence length compatibilities.

Running times. We show on Table 4.12 the running times of Alterna for a selection of RNA sequences, with pattern multi-structure produced from all structures in 20% unafold. The program was run on a Athlon Core 2 Duo with 2 GB RAM. The running time mainly depends of the number of helices in the pattern, and are also influenced by the threshold on sequence length compatibility.

4.5 Experimentations

In this section, we tested Alterna on real data. The Section 4.5.1 deals with tests on manually crafted patterns. However, the use of Alterna makes sense when the structure is not known. In Section 4.5.2, we designed experiments to simulate the situation where an unknown RNA, but of similar function/sequence, is queried against a large genomic sequence.

4.5.1 Tests on manually defined tRNA patterns

We tested two patterns : one being the RFAM consensus of the tRNA family, the other one composed by several structures, taking into account large tRNA such as Leu-tRNA.

We looked for the patterns in a 140 kbp region of *E. coli* containing 9 tRNAs, according to predictions by tRNAscan-SE[12] [66, 99]. The region was split in 1001 files of 280bp, overlapping

[12] `http://gtrnadb.ucsc.edu/Esch_coli_536/Esch_coli_536-by-locus-txt.html`

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

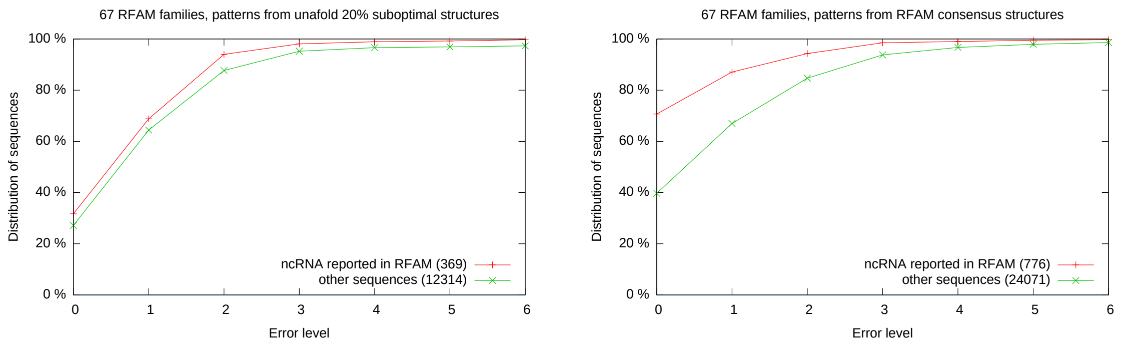


Figure 4.13: Average distribution (on 67 RFAM families) of errors when trying to match multi-pattern against 300 sequences of yass. Left: pattern gathering all 20%-suboptimal structures found by unafold on the query sequence. Right: control pattern, made by one unique structure (consensus RFAM structure).

every 140 base pairs. With the overlaps, 11 files contains a full tRNA. We ran Alterna on the 1001 files. With $d = 15\%$, 10 out of 11 files are recognized without errors (91% sensibility), with a specificity of 90%.

4.5.2 Tests with sequence filtering

To test Alterna in real use cases, on large genomic sequences, we propose to use a sequence similarity filtering as a first step. The sequence was the 249 Mbp human chromosome 1 (Genbank sequence CM000663.1, assembly GRCh37).

We ran tests on ncRNAs reported in RFAM 10.1. The RFAM `CM000663.1.gff` file contains annotations for 842 ncRNAs from 125 RFAM families[13]. To remove microRNAs and other RNAs with few structural elements, we selected families whose RFAM consensus structure possesses at least 2 helices, and 30% paired bases. To simulate the case where the structure is ignored, we used the following protocol: for any ncRNA of the human chromosome 1, we selected a homologous sequence from other organisms at a chosen similarity distance. For example, we kept 67 families presenting a seed sequence having an identity with a corresponding human ncRNA of 80%. We then built a pattern gathering the suboptimal structures found by mfold/unafold [132] with 20% suboptimality.

For the text, we filtered the chromosome 1 using yass [82], keeping only the 300 best hits, according to their Evalue. For each family, we then ran Alterna on these 300 sequences, and plotted the distribution of recognized files (Figure 4.13, left). As a control, we also built a pattern made by only one RFAM consensus structure. (Figure 4.13, right).

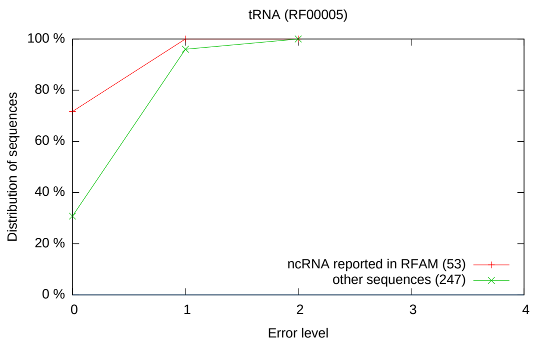
When one knows the structure, the method gives the best results. However, even when the actual structure is unknown, some families whose sequences are well structurated provide interesting results:

- There are many tRNAs (RF00005) in the human chromosome 1. We look for them using a query tRNA sequence from the fruit fly, at about 70% sequence identity of the human tRNAs. Figure 4.15 displays the structures obtained from mfold. They are sorted on increasing MFE value computed by rnaeval [51].

[13] `http://rfam.sanger.ac.uk/genome/9606`

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011



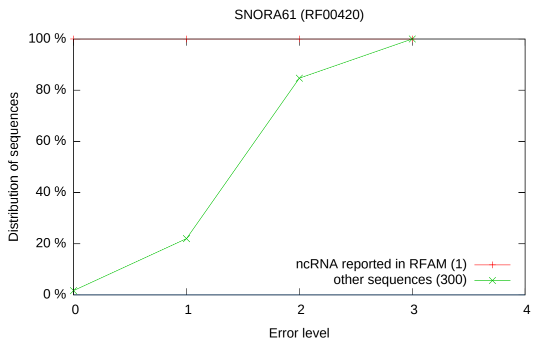


Figure 4.14: Focus on families tRNA (RF00005, left) and SNORA61 (RF00420, right). In the case of SNORA61, the unique sequence is recognized without errors, and only 5 false positives are reported without errors.

As in the other tRNA example of Figure 4.3, the real structure is not the MFE structure. On the 300 best sequences returned by yass, 53 tRNAs has found by yass (from 160 tRNAs existing in RFAM), and 114 sequences give no errors when searching with Alterna(with a sequence length compatibility constraint of $d = 5\%$), including 38 true human tRNAs (Figure 4.14, left).

- The SNORA61 element, also called H/ACA box (RF00420), interacts with the 28S ribosomal RNA. According to RFAM annotations, it is found once in the human chromosome 1, at positions 28,906,405 – 28,906,276. The query sequence was the SNORA61 element in cattle (at 79.8% identity of the human sequence). Figure 4.16 displays the structures obtained from mfold (sorted by MFE value associated by rnaeval). Here again, the real structure is not the first one. On the 300 best sequences returned by yass, only 6 sequences give no errors when searching with Alterna (with $d = 5\%$), including the only true human sequence (Figure 4.14, right).

Therefore in both cases, while we suppose that the real structure is not known, considering a multi-structure – here a set of putative structures yielded by unafold– allows us to include the real structure in the ensemble of structures of the pattern, and thus lead to increase the sensibility of the matching.

4.6 Conclusions

In this Chapter, I proposed an algorithm to search for patterns representing a set of putative or real RNA secondary structures in an efficient way. The advantage of deferring the thermodynamical computations to a external folding program is that our algorithm remains purely topological. The algorithm may be useful to study unknown RNA families. In our test cases on the human chromosome 1, Alterna is able to recognize some good tRNA and SNORA61 sequences, without the knowledge of the actual structures.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

```
    >X07778.1/115-45 RF00005-7   (60 putative helices in a region of 72 bp)
    GCAUCGGUGGUUCAGUGGUAGAAUGCUCGCCUGCCACGCGGGCGGCCCGGGUUCGAUUCCCGACCGAUGCA
C   (((((((..<<<<.......>>>>.<<<<<.......>>>>>....<<<<<.......>>>>>))))))).

#1  ((((((((.(..(.(((((((.........)))))))..)..)....((((.......)))))))))))).
#2  ((((((((................((.((((((((...))))))))).((((.......)))))))))))).
#3  ((((((((.((((.......))))((.((((((((...))))))))).((((.......)))))))))))).
#4  ((((((((((.((.(((((((.........))))))).((((...))))......))...)).)))))))).
#5  ((((((((((....(((((((.........))))))).((((...))))..........)).)))))))).
```

```
    >00113.fa         (111 putative helices in an extended region of 132 bp)
    TCCTTGTTACTATAGTGGTAAGTATCTCTGCCTGTCATGCATGAGAGAGGGGGTCGATTCCCTGACGGGGAG
C   (((((((..<<<<........>>>>.<<<<<.......>>>>>....<<<<<.......>>>>>))))))).
```

```
    >00058.fa          (129 putative helices in an extended region of 133 bp)
    TCCCTGGTGGTCTAGTGGCTAGGATTCGGCGCTTTCACCGCCTGCAGCTCGAGTTCGATTCCTGGTCAGGGAA
C   (((((((..<<<<........>>>>.<<<<<.......>>>>>.....<<<<<.......>>>>>))))))).
```

Figure 4.15: Scan of the chromosome 1 for tRNA (RF00005). For each sequence, C indicates the consensus structure found by Infernal (RFAM / Sequence Search). Top: query sequence X07778.1/115-45 *(Drosophila melanogaster)*. Structures #1 to #5 are the 5 best structures found by unafold. The RFAM consensus structure is close to unafold structure #3. Middle and bottom: text sequences from the human chromosome 1, detected by similarity search against X07778.1/115-45 (using yass). Both sequences yield no errors with Alternaand can probably identified as homologous of query sequence.

```
   >AAFC03087627.1/39513-39638 RF00420-80                                         (57 putative helices in a region of 127 bp)
   ACCCUCCAGUUCCCUUUCCCAUCGGAUCUAAACCCUGGUCUUGGUGGUCAUAAAAGGAGGAAUAGUAGUACUGAAACUGUGCUCGUGGUGUGUGCUGUACAGCACGUGGGCUAGUUUCAGACAAGG
C  ..<<<<<...........<<<<<.<<<<........>>>>..>>>>>........>>>>>...............<<<.<<<<.<<<............>>>...>>>>>>>............

#1 ..(((((..((.......((((((((((........)))).))))))....))..)))))......((.(((((((((.(((((.....((((((...)))))))))))))))))))))))....
#2 ..(((((..((.......((((((((((........)))).))))))....))..)))))......((.(((((((((.(((((((.((((.....)))).)))).)))))))))))))))....
#3 ..(((...(((((((((.((((((((((........)))).))))))......)))).)))))....((.(((((((((.(((((.....((((((...)))))))))))))))))))))).)))
#4 ..(((((.(((.....(((....)))....)))..((..(.....)..)).....))))).......((.(((((((((.(((((.....((((((...)))))))))))))))))))))))....
#5 ..(((((.........(((....))).....(((((......)).))).......))))).......((.(((((((((.(((((.....((((((...)))))))))))))))))))))))....
```

```
   >00001.fa                                                                       (93 putative helices in an extended region of 190 bp)
   TCCTCCTGATCCCTTTCCCATCGGATCTGAACACTGGTCTTGGTGGTCGTAAAAGGAGGAAAAGTAATAGTGAAGCTGGCCTAAATGTTGTAATCTGGTATATGGCATGTGGGCTAGTTTCAGACAGGTTTCA
C   ..<<<<<...........<<<<<.<<<<........>>>>..>>>>>........>>>>>...............<<<<<<<........<<<............>>>...>>>>>>>............
```

Figure 4.16: Scan of the chromosome 1 for SNORA61 (RF00420). For each sequence, C indicates the consensus structure found by Infernal (RFAM / Sequence Search). Top: query sequence AAFC03087627.1/39513-39638 X07778.1/115-45 *(Bos taurus)*. Structures #1 to #5 are the 5 best structures found by unafold. Bottom: Sequence from the human chromosome 1, detected by similarity search against X07778.1/115-45 (identity: 79.8%) by yass. The multi-structure is found by Alterna without errors.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

Chapter 5

Conclusion

Predicting and browsing sets of RNA suboptimal structures is of critical importance to study ncRNA functions. Moreover, these ensembles may play a significant role in the evolution, as stated by Bonhoeffer and al. [16]:

"*The genealogy of a single sequence represents a trajectory through sequence space. The quasispecies* [14] *concept broadens the trajectory to a band, the concept of conformational ensembles widens this band through sequence space further. (...) Neglecting non-equilibrium structures means an underestimation of the power Darwinian selection in molecular evolution.*"

⋆ ⋆ ⋆

My work is based up on dealing with sets of RNA structures on a same sequence. Chapter 3 ("Locally Optimal Structures") details how such sets are produced and Chapter 4 ("Alternate RNA Structure Pattern Matching") tells us how they can be used as a pattern.

Since the conservation of ncRNA structure against evolution pressure has been happening via conformation of stable helices, I choose to work directly with helices. Both Regliss and Alterna take putative or real *helices* as input, enabling to be, with topological algorithms, as close as possible to thermodynamical information of the RNA molecule.

Both methods rely on the decomposition of flat structures: Regliss provides a set of maximal for juxtaposition structures (which is a subset of the set of flat structures) and combines them by using a stack to produce locally optimal structures on the fly, whereas Alterna requires, as a pattern, a set of flat structures. The flat structures allow to factorize the computations: Each operation being carried out just one time for each flat structure, results are used several times.

My two new methods have been implemented and tested on real data. Regliss can not really be categorized in any prediction methods described in Section 1.2. Regliss discovers the set of all possible structures for a given set of putative helices. Is does not use energy minimization, partition function, nor any other classic methods used to generate suboptimal structures, but starts directly from putative helices produced by any folding method.

[14] group of genomes with many mutations

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

Neither can Alterna be categorized in any pattern matching methods presented in the introduction, because our multi-structure pattern offers us several choices for matching different structures at a same time. In combination with some sequence filtering tools, one may use Alterna to search for RNA pattern on large genomes, without knowing the actual RNA structure.

⋆ ⋆ ⋆

I conclude this thesis by proposing several research themes on sets of alternate secondary structures:

- *Weighted multi-structures.* A multi-structure could be weighted to favor an instance of a structure, if there is a priority for some kind of structures over other structures. This can bring a large flexibility into the model.
- *Browsing and visualizing multi-structures.* On the website of Regliss, we suggested a neighborhood graph, which provides a tool to browse the folding space represented by the neighboring locally optimal structures. This kind of tool could be improved, for instance by computing and browsing connected components. Visualizing an occurrence of multi-structure found by Alterna is also challenging.
- *Statistics on multi-structures.* Many statistic of properties of RNA have already been studied [108]. We could extend these statistics on multi-structures, or on specific multi-structures such as the locally optimal secondary structures.
- *Comparison of multi-structures.* Comparing sets of RNA structures could be done as an extension to the RNA comparisons presented in the introduction. Given a group of multi-structures, are there identical (or similar) instances in them ? What are the unique instances of every multi-structure ?
- *Indexation of multi-structures.* The search method behind Alterna could be optimized using a filtering based on indexation on some tree elements. This problem is very challenging, as even the indexation of single RNA structures remain an open issue.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Bibliography

[1] C. Alkan, E. Karakoc, J. Nadeau, C. Sahinalp, and K. Zhang. RNA-RNA interaction prediction and antisense RNA target search. 13(2):267–283, 2006.

[2] J. Allali, Y. d'Aubenton Carafa, C. Chauve, A. Denise, C. Drevet, P. Ferraro, D. Gautheret, C. Herrbach, F. Leclerc, A. De Monte, A. Ouangraoua, M.-F. Sagot, C. Saule, M. Termier, C. Thermes, and H. Touzet. Benchmarking RNA secondary structure comparison algorithms. In *Journées Ouvertes de Biologie, Informatique et Mathématiques JOBIM'08*, pages 67–68, 2008.

[3] J. Allali and M. F. Sagot. A multiple layer model to compare RNA secondary structures. *Software: Practice and Experience*, 2007.

[4] S. F. Altschul and B. W. Erickson. 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, 1985.

[5] V. Ambros. microRNAs: Tiny regulators with great potential. 107:862–864, 2001.

[6] B. Andrea L. Edwards and R. T. Batey. Riboswitches: A common RNA regulatory element. *Nature Education*, 3(9):9, 2010.

[7] M. Andronescu, Z. Zhang, and A. Condon. Secondary structure prediction of interacting RNA molecules. *Journal of Molecular Biology*, 345:987–1001, 2005.

[8] R. Backofen, D. Tsur, S. Zakov, and M. Ziv-Ukelson. Sparse RNA folding: Time and space efficient algorithms. pages 249–262, 2009.

[9] V. Bafna, H. Tang, and S. Zhang. Consensus folding of unaligned RNA sequences revisited. *RECOMB*, pages 172–187, 2005.

[10] I. Bentwich, A. Avniel, Y. Karov, R. Aharonov, S. Gilad, O. Barad, A. Barzilai, P. Einat, U. Einav, E. Meiri, E. Sharon, Y. Spector, and Z. Bentwich. Identification of hundreds of conserved and nonconserved human microRNAs. *Nature Genetics*, 37(7):766–770, 2005.

[11] S. M. Berget and P. A. Sharp. A spliced sequence at the 5'-terminus of adenovirus late mRNA. *Brookhaven Symp Biol*, (29):332–44, 1977.

[12] S. H. Bernhart, C. H. Flamm, P. F. Stadler, and I. L. Hofacker. Partition function and base pairing probabilities of RNA heterodimers algorithms. 22(10):1177–1182, 2006.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

[13] B. Billoud, M. Kontic, and A. Viari. Palingol: a declarative programming language to describe nucleic acids' secondary structures and to scan sequence database. *Nucleic Acids Research*, 24:1395–1403, 1996.

[14] G. Blin, A. Denise, S. Dulucq, C. Herrbach, and H. Touzet. Alignment of RNA structures. 2008.

[15] A. Boivin and R. Vendrely. Sur le rôle possible de des deux acides nucléiques dans la cellule vivante. *Experimentia*, 3:32–34, 1947.

[16] S. Bonhoeffer, J. S. McCaskill, P. F. Stadler, and P. Schuster. RNA multi-structure landscapes. *Euro Biophys J*, 22:13–24, 1993.

[17] P. N. Borer, B. Dengler, I. Tinoco, and O. C. Uhlenbeck. Stability of ribonucleic acid double-stranded helices. *Journal of Molecular Biology*, 86:843–53, 1974.

[18] J. Brown. The ribonuclease P database. *Nucleic Acids Research*, 27:314, 1999.

[19] R. E. Bruccoleri and G. Heinrich. An improved algorithm for nucleic acid secondary structure display. *Computer Applications in Biosciences*, 4:167–173, 1988.

[20] R. J. Carter, I. Dubchak, and S. R. Holbrook. A computational approach to identify genes for functional RNAs in genomic sequences. *Nucleic Acids Research*, 29:3928–3938, 2001.

[21] L. T. Chow, J. M. Roberts, J. B. Lewis, and T. R. Broker. A map of cytoplasmic RNA transcripts from lytic adenovirus type 2, determined by electron microscopy of RNA:DNA hybrids. *Cell*, 11(4):819–36, 1977.

[22] P. Clote. 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, 2005.

[23] P. Clote. RNALOSS: A web server for RNA locally optimal secondary structures. *Nucleic Acids Research*, 33:W600–604, 2005.

[24] J. R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R. J. Farris, A. S. Kulam-Syed-Mohideen, D. M. McGarrell, T. Marsh, G. M. Garrity, and J. M. Tiedje. The ribosomal database project: improved alignments and new tools for rRNA analysis. *Nucleic Acids Research*, 37:D141–D145, 2008.

[25] F. Crick. Central dogma of molecular biology. *Nature*, 227:561–563, 1970.

[26] F. H. Crick. On protein synthesis. *Symposia of the Society for Experimental Biology*, 12:138–63, 1958.

[27] A. L. Dounce. Duplicating mechanism for peptide chain and nucleic acid synthesis. *Enzymologia*, 15:251–258, 1952.

[28] M. Dsouza, N. Larsen, and R. Overbeek. Searching for patterns in genomic data. *Trends in Genetics*, 13:497–498, 1997.

[29] S. R. Eddy. Rnabob: a program to search for RNA secondary structure motifs in sequence databases. ftp://selab.janelia.org/pub/software/rnabob/, 1996.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

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

[31] S. R. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Research*, 22:2079–2088, 1994.

[32] D. J. Evers and R. Giegerich. Reducing the conformation space in RNA structure prediction. In *German Conference on Bioinformatics (GCB 2001)*, 2001.

[33] D. J. Evers and R. Giegerich. Reducing the conformation space in RNA structure prediction. *Bioinformatics*, 20(10):1573–1582, 2004.

[34] D. Fera, N. Kim, N. Shiffeldrim, J. Zorn, U. Laserson, H. H. Gan, and T. Schilk. RAG: RNA-As-Graphs web resource. *BMC Bioinformatics*, 2004.

[35] A. Fire, S. Xu, M. Montgomery, S. Kostas, S. Driver, and C. Mello. Potent and specific genetic interference by double-stranded RNA in caenorhabditis elegans. *Nature*, 391(6669):806–811, 1998.

[36] W. Fontana, T. Griesmacher, W. Schnabl, P. F. Stadler, and P. Schuster. Statistics of landscapes based on free energy, replication and degradation rate constants of RNA secondary structures. *Chemical Monthly*, 122:795–819, 1991.

[37] S. M. Freier, R. Kierzek, J. A. Jaeger, N. Sugimoto, M. H. Caruthers, T. Neilson, and D. H. Turner. Improved free-energy parameters for predictions of RNA duplex stability. *Proc. Natl. Acad. Sci.*, 83:9373–9377, 1986.

[38] E. Freyhult, V. Moulton, and P. Clote. RNAbor: A web server for RNA structural neighbors. *Nucleic Acids Research*, 35:W305–309, 2007.

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

[40] D. Gautheret and A. Lambert. Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. *Journal of Molecular Biology*, 313:1003–1011, 2001.

[41] D. Gautheret, F. Major, and R. Cedergren. Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for trna. *Comput Appl Biosci*, 6(4):325–331, 1990.

[42] R. Gesteland, T. Cech, and J. Atkins. *The RNA World*. Cold Spring Harbor Laboratory Press, New York, third edition, 1999.

[43] R. Giegerich. Prediction and visualization of structural switches in RNA. *Pacific Symposium on Biocomputing*, 4:126–137, 1999.

[44] G. Gilbert. Why gene in pieces? *Nature*, 271:501, 1978.

[45] S. Griffiths-Jones, H. Saini, S. van Dongen, and A. Enright. mirbase: tools for microRNA genomics. *Database Issue*, pages D154–D158, 2008.

© 2012 Tous droits réservés.

http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

[46] G. Grillo, F. Licciulli, S. Liuni, E. Sbisà, and G. Pesole. Patsearch: a program for the detection of patterns and structural motifs in nucleotide sequences. *Nucleic Acids Research*, 31:3608–3612.

[47] B. J. Haas ES. Evolutionary variation in bacterial RNase P RNAs. *Nucleic Acids Res.*, 26(18):4093–9, 1998.

[48] J. Havgaard, E. Torarinsson, and J. Gorodkin. Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. *PLOS computational biology*, (3), 2007.

[49] M. Höchsmann, T. Töller, R. Giegerich, and S. Kurtz. Local similarity in RNA secondary structures. In *Proc. 2nd IEEE Computer Society Bioinformatics Conference (CSB 2003), 11-14 August 2003, Stanford, CA, USA*, pages 159–168, 2003.

[50] I. L. Hofacker, M. Fekete, and P. F. Stadler. Secondary structure prediction for aligned RNA sequences. *Journal of Molecular Biology*, 319:1059–1066, 2002.

[51] I. L. Hofacker, W. Fontana, P. F. Stadler, S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, 125:167–188, 1994.

[52] I. L. Hofacker, P. Schuster, and P. F. Stadler. Combinatorics of RNA secondary structures. *Discrete Appl. Math.*, 88:207–237, November 1998.

[53] R. W. Holley. Structure of an alanine transfer ribonucleic acid. *JAMA*, 194:868–871, 1965.

[54] Y. Hu. Prediction of consensus structural motifs in a family of coregulated RNA sequences. *Nucleic Acids Research*, 30(17):3886–3893, 2002.

[55] M. Höchsmann. *The Tree Alignment Model: Algorithms, Implementations and Applications for the Analysis of RNA Secondary Structures.* PhD thesis.

[56] M. Höchsmann, T. Töller, R. Giegerich, and S. Kurtz. Local similarity in RNA secondary structures. *Proceedings of the IEEE Bioinformatics Conference 2003(CSB 2003)*, pages 159–168, 2003.

[57] M. Höchsmann, B. Voss, and R. Giegerich. Pure multiple RNA secondary structure alignments: A progressive profile approach. *in IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1:53–62, 2004.

[58] H. Isambert and E. Siggia. Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme. *Proceedings of the National Academy of Sciences USA*, 97(12):6515–6520, 2000.

[59] F. Jacoba and J. Monoda. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3:318–356, 1961.

[60] B. JW. The Ribonuclease P Database. *Nucleic Acids Research*, 27(1):314, 1999.

[61] T. Kasami. An efficient recognition and syntax-analysis algorithm for context-free languages. *Scientific report AFCRL*, 65:758, 1965.

[62] Y. Kato, K. Sato, M. Hamada, Y. Watanabe, K. Asai, and T. Akutsu. fast and accurate prediction of RNA-RNA interaction using integer programming. 26 ECCB:i1–i7, 2010.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

[63] K. Lagesen, P. Hallin, E. A. Rodland, H. H. Staerfeldt, T. Rognes, and D. W. Ussery. RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucl. Acids Res.*, 35(9):3100–3108, 2007.

[64] N. B. Leontis and E. Westhof. Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(4):499–512, 2001.

[65] W. A. Lorenz and P. Clote. Computing the partition function for kinetically trapped rna secondary structures. *PLoS ONE*, 6(1):e16178, 01 2011.

[66] T. M. Lowe and S. R. Eddy. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Research*, 25:955–964, 1997.

[67] Z. J. Lu, D. H. Turner, and D. H. Mathews. A set of nearest neighbor parameters for predicting the enthalpy change of RNA secondary structure formation. *Nucleic Acids Research*, (34):4912–4924, 2006.

[68] R. B. Lyngsø and C. N. S. Pedersen. Pseudoknots in RNA secondary structures. In *Proceedings of the fourth annual international conference on Computational molecular biology*, RECOMB '00, pages 201–209, New York, NY, USA, 2000. ACM.

[69] T. J. Macke, D. J. Ecker, R. R. Gutell, D. Gautheret, D. A. Case, and R. Sampath. RNAmotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Research*, 29:4724–4735, 2001.

[70] N. Markham and M. Zuker. UNAFold: software for nucleic acid folding and hybriziation. *matics: Structure, Function and Applications*, 453:3–31, 2008.

[71] A. Mathelier and A. Carbone. Mirena: finding micrornas with high accuracy and no learning at genome scale and from deep sequencing data. *Bioinformatics*, 26(18):2226–2234, 2010.

[72] D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci.*, 101:7287–7292, 2004.

[73] D. H. Mathews, J. Sabrina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940, 1999.

[74] D. H. Mathews and D. H. Turner. Dynalign: An algorithm for finding the secondary structure common to two RNA sequences. *Journal of Molecular Biology*, 191(317):810–825, 2002.

[75] J. S. Mattick. Non-coding RNAs: the architects of eukaryotic complexity. *EMBO reports*, 2:986–991, 2001.

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

[77] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

[78] J. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965.

[79] U. Mueckstein, H. Tafer, J. Hackermueller, S. H. Bernhart, P. F. Stadler, and I. L. Hofacker. Thermodynamics of RNA-RNA binding. 22(10):1177–1182, 2006.

[80] E. P. Nawrocki and S. R. Eddy. Query-dependent banding (qdb) for faster RNA similarity searches. *PLoS Computational Biology*, 3:e56, 2007.

[81] E. P. Nawrocki, D. L. Kolbe, and S. R. Eddy. Infernal 1.0: Inference of RNA alignments. *Bioinformatics*, 25:1335–1337, 2009.

[82] L. Noé and G. Kucherov. YASS: enhancing the sensitivity of DNA similarity search. *Nucleic Acids Research*, 33:W540–W543, 2005.

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

[84] R. Nussinov, G. Pieczenik, J. Griggs, and D. Kleitman. Algorithms for loop matchings. *SIAM J. Appl. Math.*, 35:68–82, 1978.

[85] M. Parisien and F. Major. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452(7183):51–55, 03 2008.

[86] G. Pavesi, G. Mauri, M. Stefani, and G. Pesole. RNAprofile: an algorithm for finding conserved secondary structure motifs in unaligned RNA sequences. *Oxford Journals*, 32(10):3258–3269, 2004.

[87] O. Perriquet, H. Touzet, and M. Dauchet. Finding the common structure shared by two homologous RNAs. 19:108–116, 2003.

[88] E. Pruesse, C. Quast, K. Knittel, B. Fuchs, W. Ludwig, J. Peplies, and F. Gloeckner. SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic Acids Research*, 35:7188–7196, 2007.

[89] M. Rabani, M. Kertesz, and E. Segal. Computational prediction of RNA structural motifs involved in posttranscriptional regulatory processes. *Proc Natl Acad Sci*, 105:14885–14890, 2008.

[90] J. Reeder, J. Reeder, and R. Giegerich. Locomotif: From graphical motif description to RNA motif search in bioinformatics. *Nucleic Acids Research*, 23(13):392–400, 2007.

[91] M. Rehmsmeier, P. Steffen, M. Höchsmann, and R. Giegerich. Fast and effective prediction of microRNA/target duplexes. 10:1507–1517, 2004.

[92] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. 285:2053–2068, 1999.

[93] E. Rivas, R. Klein, T. Jones, and S. Eddy. Computational identification of noncoding RNAs in e. coli by comparative genomics. *Current biology*, 11:1369–1373, 2001.

[94] Y. Sakakibara, M. Brown, R. Hughey, I. Mian, K. Sjolander, R. Underwood, and D. Haussler. Stochastic context-free grammars for tRNA modeling. 22:5112–5120, 1994.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

[95] R. Salari, M. Mohl, S. Will, S. Sahinalp, and R. Backofen. Time and space efficient RNA-RNA interaction prediction via sparse folding. 2010.

[96] D. Sankoff. Simultaneous solution of the RNA folding, alignement and protosequence problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825, 1985.

[97] C. Saule, M. Régnier, J. M. Steyaert, and A. Denise. Counting RNA pseudoknotted structures. *Journal of Computational Biology*, 2010.

[98] P. Schattner. Searching for RNA genes using base-composition statistics. *Nucleic Acids Res.*, 30(9):2076–82, 2002.

[99] P. Schattner, A. Brooks, and T. Lowe. The tRNAscan-SE, snoscan and snoGPS web servers for the detection of tRNAs and snoRNAs. *Nucleic Acids Research*, 33:W686–689, 2008.

[100] B. A. Shapiro. An algorithm for comparing multiple RNA secondary structures. *CABIOS*, 4:381–393, 1988.

[101] B. A. Shapiro and K. Zhang. Comparing multiple RNA secondary structures using tree comparison. *CABIOS*, 6:309–318, 1990.

[102] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[103] M. Sprinzl and K. Vassilenko. Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Research*, 33:139–140, 2005.

[104] R. P. Stanley. *Enumerative combinatorics*, volume 2. Cambridge Studies in Advanced Mathematics, 1999.

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

[106] P. R. Stein and M. S. Waterman. On some new sequences generalizing the Catalan and Motzkin numbers. *Discrete Mathematics*, 26(3):261 – 272, 1979.

[107] Y. Sun and J. Buhler. Designing patterns and profile for faster HMM search. *IEEE Transactions on Computational Biology and Bioinformatics*, 6:232–234, 2008.

[108] M. Tacker, P. F. Stadler, E. G. Bornberg-Bauer, I. L. Hofacker, and P. Schuster. Algorithm independent properties of RNA secondary structure predictions. *European Biophysics Journal*, 25(2):115–130, 1996.

[109] H. Tafer and I. L. Hofacker. RNAplex: a fast tool for RNA–RNA interaction search . 24:2657–2663, 2008.

[110] F. Tahi, S. Engelen, and M. Régnier. A fast algorithm for RNA secondary structure prediction including pseudoknots. In *Proceedings of the 3rd IEEE Symposium on BioInformatics and BioEngineering*, BIBE '03, pages 11–, Washington, DC, USA, 2003. IEEE Computer Society.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

[111] P. Thébault, S. de Givry, T. Schiex, and C. Gaspin. Searching RNA motifs and their intermolecular contacts with constraint networks. 22:2074–2080, 2006.

[112] I. Tinoco, P. N. Borer, B. Dengler, M. D. Levin, O. C. Uhlenbeck, D. M. Crothers, and J. Bralla. Improved estimation of secondary structure in ribonucleic acids. *Nature*, 150:40–41, 1973.

[113] I. Tinoco, O. C. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[114] H. Touzet and O. Perriquet. Carnac: folding families of non coding RNAs. 142, 2004.

[115] D. H. Turner and D. H. Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, 38:280–282, 2010.

[116] D. H. Turner, N. Sugimoto, and S. M. Freier. RNA structure prediction. *Ann. Rev. Biophys. Biophys. Chem*, 17:167–192, 1988.

[117] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

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

[119] R. Walczak, E. Westhof, P. Carbon, and A. Krol. A novel RNA structural motif in the selenocysteine insertion element of eukaryotic selenoprotein mRNAs. *RNA*, 2:367—379, 1996.

[120] J. Waldispühl and P. Clote. Computing the partition function and sampling for saturated secondary structures of RNA, with respect to the turner energy model. *J. Computational Biology*, 14:190–215, 2007.

[121] A. Walter, D. Turner, J. Kim, M. Lyttle, P. Müller, D. Mathews, and M. Zuker. Coaxial stacking of helices enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proc. Natl. Acad. Sci.*, 91:9218–9222, 1994.

[122] X. Wang, J. Zhang, F. Li, J. Gu, T. He, X. Zhang, and Y. Li. Microrna identification based on sequence and structure alignment. *Bioinformatics*, 21:3610–3614, September 2005.

[123] S. Washietl, I. L. Hofacker, and P. F. Stadler. Fast and reliable prediction of noncoding RNAs. 102:2454–2459, 2005.

[124] M. S. Waterman and T. H. Byers. A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Mathematical Biosciences*, 77(2):179–188, 1985.

[125] M. S. Waterman and T. F. Smith. RNA secondary structure: a complete mathematical analysis. *Mathematical Biosciences*, 42:257–266, 1978.

[126] Z. Weinberg and W. Ruzzo. Sequence-based heuristics for faster annotation of non-coding RNA families. *Bioinformatics*, 22:35–39, 2006.

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

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Thèse de Azadeh Saffarian, Lille 1, 2011

[128] A. Xayaphoummine, T. Bucher, and H. Isambert. Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots. *Nucleic Acids Research*, 33:W605–W610, 2005.

[129] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18:1245–1262, December 1989.

[130] M. Ziv-Ukelson, I. Gat-Viks, Y. Wexler, and R. Shamir. A faster algorithm for RNA co-folding. *Algorithms in Bioinformatics*, pages 174–185, 2008.

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

[132] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, 2003.

[133] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, 07 1984.

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

[135] M. Zytnicki, C. Gaspin, and T. Schiex. DARN! a weighted constraint solver for RNA motif localization. *Constraints*, 13:91–109, June 2008.

© 2012 Tous droits réservés. http://doc.univ-lille1.fr

Résumé

L'ARN (acide ribonucléique) est une molécule ubiquitaire qui joue plusieurs rôles fondamentaux au sein de la cellule: synthèse des protéines (ARN messagers), activité catalytique ou implication dans la régulation (ARN non-codants). Les nouvelles technologies de séquençage, dites "à haut débit", permettent de produire des milliards de séquences à moindre coût, posant de manière cruciale la question de l'analyse de ces données.

L'objectif de cette thèse est de définir de nouvelles méthodes computationnelles pour aider à l'analyse de ces séquences dans le cas des ARN non-codants. Dans cette perspective, la "structure secondaire" d'un ARN, formée par l'ensemble des appariements entre bases, délivre des informations utiles pour étudier la fonction de l'ARN. Notre travail se concentre plus particulièrement sur l'ensemble des structures potentielles que peut adopter une séquence d'ARN donnée, ensemble que nous appelons "multi-structure". Nous apportons deux contributions: un algorithme pour générer systématiquement toutes les structures localement optimales composant une multi-structure, et un algorithme basé sur la recherche d'une multi-structure pour identifier un ARN non-codant dans une séquence génomique. Ces résultats ont été mis en oeuvre dans deux logiciels, Alterna et Regliss, appliqués avec succès à des ensembles de test.

Mots-clés: bio-informatique, biologie computationnelle, algorithmique discrète, analyse des génomes, ARNs non-codants.

Abstract

RNA (ribonucleic acid) molecules have various functions in cells. Just as they can store and deliver the DNA message for the protein synthesis (messenger RNAs), they can also directly catalyze chemical reactions or act as a regulator (functional RNAs, also called non-coding RNAs).

Nowadays, recent sequencing technologies yield billions of genomic sequences – DNA, RNA – at a very small cost. However, sequencing is only the first step: The function of the sequence remains open for investigation. The objective of the thesis is to define new computational methods to help sequence and structure analysis of non-coding RNAs. In this perspective, the "secondary structure" of an RNA, made with base pairs, provides useful hints to further study its function. Our work is focused on sets of all possible RNA structures for a given sequence, introducing the concept of "RNA multi-structures".

The thesis details how such sets can be constructed systematically to generate all locally optimal secondary structures, and how they can be used as a pattern to identify non-coding RNAs in genomic sequences. We provide efficient algorithms for these two problems. These algorithms have been implementated in the software tools Alterna and Regliss and tested on real data, providing new insight into RNA structures.

Keywords: bioinformatics, computational biology, discrete algorithmics, genome analysis, non-coding RNAs.