Uniform and non-uniform random generation of RNA secondary structures with pseudoknots

Cédric Saule, Claire Wallon, Alain Denise

Uniform and non-uniform random generation of RNA secondary structures with pseudoknots

Cédric Saule [*†] Claire Wallon [*] Alain Denise[*‡†]

Gascom 2010 - Montreal

Abstract

We give an efficient algorithm to generate random RNA secondary structures with pseudoknots, either uniformly or non uniformly in a controllable fashion. Although we consider a restrictive class of pseudoknots, the class of *simple recursive pseudoknots*, it turns out that most of the known real RNA pseudoknotted secondary structures in the biological databases belong to this class.

1 Introduction

Algorithms for generating random biological sequences have been investigated for a long time [9, 3, 11]. Random sequences are of great interest in genome analysis: they provide a way to represent the "background noise" from which the real biological information can be distinguished. A number of softwares for generating random sequences have been made available (see e.g. [5, 17, 16]). More recently, random generation of more complex structures has been investigated. For example, random graphs are useful for retrieving biological information from biological networks such as regulation networks or protein interaction networks [13].

Special attention has been paid for RNA secondary structures. RNA is a major component of cellular processes, as DNA and proteins. Briefly, a RNA molecule is a chain of nucleotides A, C, G, and U (for Adenine, Cytosine, Guanine and Uracil respectively) that folds onto itself to form a three-dimensional conformation, according to chemical bounds between pairs of nucleotides. Algorithms of random generation of RNA structures are used notably for predicting the structure of a given sequence [8, 14] and for evaluating structure comparison softwares [2]. Up to now, random generation algorithms deal only with so-called *secondary structures without pseudoknots*. As will be

[*]LRI, Université Paris-Sud and CNRS. Bat 490, 91405 Orsay cedex, France
[†]INRIA Saclay, Parc Orsay Université, 4 rue Jacques Monod, 91893 Orsay cedex, France
[‡]IGM, Université Paris-Sud and CNRS, bât. 400, 91405 Orsay cedex, France

seen below, such a structure is a partial representation of the whole molecular structure. Most of the RNA structures contain *pseudoknots*, configurations that cannot be taken into account by generation algorithms.

In this paper, we give an efficient algorithm to generate uniformly at random secondary structures with pseudoknots. Although we consider a restrictive class of pseudoknots, the class of *simple recursive pseudoknots*, it turns out that most of the known real RNA secondary structures in the biological databases belong to this class. Our random generation method is based on a context-free encoding of recursive pseudoknots, and involves a linear decoding algorithm for obtaining the final structures. Using the classical recursive method [10] for generating the encoding, it leads to a total complexity in $n \log n$ for generating a structure of size n.

Finally, we give a few experimental results. We present a general context-free grammar for pseudoknotted structures, and we use it to generate uniform random structures. The we show how a particular weighted grammar, as defined and studied in [7], can be designed in order to generate more realistic RNA structures.

2 Definitions and notation.

The structure of a RNA molecule mainly depends on the interactions between its nucleotides. Notably, $A - U$ and $G - C$ form strong hydrogen bounds, they constitute the two *Watson-Crick* pairs of nucleotides (or basepairs, for short). The pair $G - U$, weaker, is called the *Wobble* basepair. Meanwhile, any pair of nucleotides can form a (generally weak) chemical bond [12].

Any RNA structure can be represented by a graph where the nodes are the nucleotides and the edges are the chemical bonds between them. Each node is numbered by its position in the sequence. Several levels of structure have been defined. The *tertiary* structure is the graph that contains all the chemical bonds in the molecule. A *secondary structure* (with or without pseudoknots) is a subgraph of the tertiary structure that contains all the nodes, but only the stronger bounds (Watson-Crick and Wobble). In this kind of structure, every node has one neighbour at most. So we can define the graph of an RNA secondary structure (possibly with pseudoknots) as below. The notion of pseudoknot will be defined later.

Definition 1. *The graph of an RNA secondary structure (possibly with pseudoknots) is a graph G with a set of vertices $V = \{1, 2, \ldots, n\}$ and a set of edges E, such that each vertex has degree at most* 1.

The following two definitions are required before defining the pseudoknot.

Definition 2 (Crossing arcs). *Let (i, j) and (k, l) two edges of G, with $i < j$ and $k < l$. We say that (i, j) and (k, l) are* crossing *if $i < k < j < l$.*

Definition 3 (Crossing graph). *The* crossing graph *of the graph G of an RNA structure is a graph C defined as follows: the vertices of C are the edges of G, and two vertices of C are connected by an edge if and only if their two corresponding arcs in G are crossing.*

Definition 4 (Pseudoknot)**.** *A* pseudoknot *is a set of arcs that is not a singleton and that corresponds to a maximal connected component in the crossing graph.*

Figure 1 shows two representations of the graph of a RNA secondary structure without pseudoknot, where the vertices are labelled by their corresponding nucleotide. It is

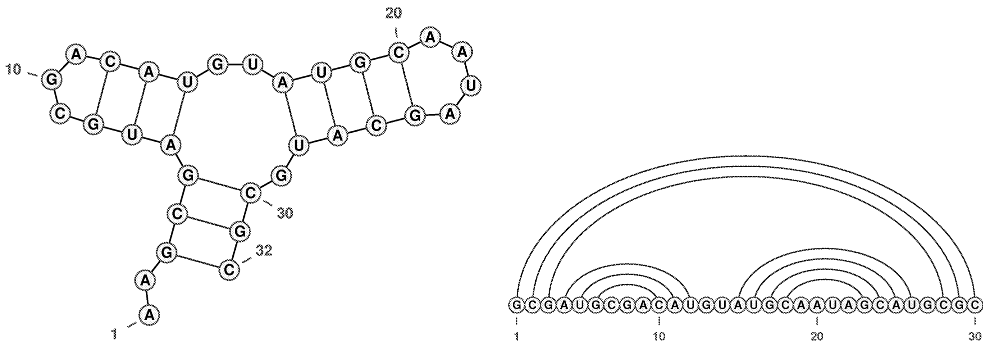


Figure 1: Left: classical "biological" representation of a secondary structure without pseudoknot. Right: representation by an *arc-annotated sequence. Drawing done with VARNA [6].*

well known that the set of secondary structures without pseudoknot are in one-to-one correspondence with Motzkin words [19]. Indeed, the two extremities of a basepair can be encoded, respectively, by an opening parenthese and a closing parenthese, and the unpaired base can be encoded by a dot. For example, the code for the structure in Figure 1 is ((((((...)))..((((....)))).))). On the other hand, the set of secondary structures with pseudoknots on $\{1, 2, \ldots, n\}$ is in one-to-one correspondence with the set of involutions on $\{1, 2, \ldots, n\}$.

As will be seen later, the notions of *simple pseudoknot* and *H-type pseudoknot* are important for our purpose.

Definition 5 (Simple pseudoknot [1])**.** *A pseudoknot P is* simple *if there exist two numbers j_1 and j_2, with $j_1 < j_2$, such that:*

- *each edge (i, j) in P satisfies either $i < j_1 < j \leq j_2$ or $j_1 \leq i < j_2 < j$,*
- *and if two edges (i, j) and (i', j') satisfy $i < i' < j_1$ or $j_1 \leq i < i'$, then $j > j'$.*

The first property ensures that, for each edge of P, one of its ends exactly is between j_1 and j_2. And the edges are divided in two sets: those having their other end smaller than j_1, and those having their other end greater than j_2. We call these two sets, respectively, the *left part* and the *right part* of the pseudoknot. The second property of the definition ensures that two edges in the same set cannot intersect each other. Figure 2 shows a simple pseudoknot.

Definition 6 (H-type Pseudoknot)**.** *A* H-type pseudoknot *is a simple pseudoknot having the following additional property: each arc in one of the two above sets crosses all the arcs of the other set.*

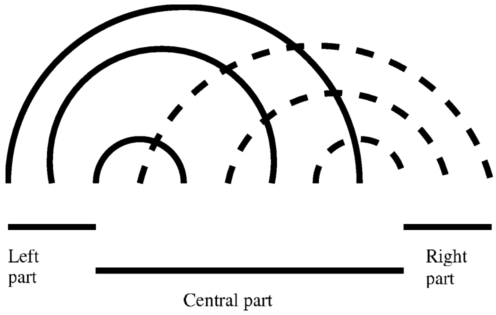


Figure 2: A simple pseudoknot.

3 Random generation of pseudoknotted structures.

As remarked in the previous section, the set of all theoretical pseudoknotted structures is in bijection with the set of involutions. Meanwhile, due to the steric and thermodynamic constraints on RNA molecules, only a negligeible fraction of the set of theoretical pseudoknotted structures can be produced in real molecules. Notably, it was shown in [4] that more than 87% of the RNA structures in the biological databases contain only H-type pseudoknots. For this reason, we need to consider restricted classes of pseudoknots in order to generate "biologically realistic" pseudoknotted structures. In the following, we consider two classes of pseudoknotted structures: those containing only simple pseudoknots, and the subclass of those containing only H-type pseudoknots. Meanwhile, any pseudoknot can embed substructures that can be pseudoknotted in turn. For this reason, the structures are sayed to be *recursively pseudoknotted*.

In section 3.1, we present an encoding of these two classes by a context-free language. Although this encoding was first stated in [18] by two aof the authors together with Mireille Régnier and Jean-Marc Steyaert, we present it here for the sake of self-containment. Then in section 3.2 we give a linear algorithm that constructs a pseudoknot given its encoding. Finally, we present in section 3.3 some preliminary results in generating realistic RNA pseudoknotted structures.

3.1 A context-free encoding for simple and H-type pseudoknots

Let us first recall some definitions. Let L be a language on a given alphabet A, and $w = w_1 w_2 \dots w_n$ a word of L, where the w_i's are the letters of w. A word v is a *subword* of w if $v = w_{i_1} w_{i_2} \dots w_{i_k}$, where $1 \leq i_1 < i_2 < \dots < i_k \leq n$. The *projection* of w onto an alphabet $A' \in A$ is the subword w' obtained by erasing in w all letters that do not belong to A'. The projection of L onto A' is the set of projections of the words of L onto A'. Finally, let us recall that the Dyck language on any two-letter alphabet $\{d, \bar{d}\}$ is the language of balanced parentheses strings, where d and $\bar{d}$ stand, respectively, for opening and closing parentheses. Now we can state twe two following straightforward lemmas:

Lemma 1. *Any class of pseudoknotted structures where all pseudoknots are simple can be represented by the words of a language L on the alphabet $\{c, d, \bar{d}, x, \bar{x}, y, \bar{y}\}$ where*

- *(i) c encodes the unpaired nucleotides;*
- *(i) d and $\bar{d}$ encode, respectively, the left and right ends of edges that are not involved in pseudoknots;*
- *(ii) x and $\bar{x}$ encode, respectively, the left and right ends of edges that are involved in the left parts of pseudoknots;*
- *(iii) y and $\bar{y}$ encode, respectively, the left and right ends of edges that are involved in the right parts of pseudoknots.*

Additionally, the projection of the language to the alphabet $\{d, \bar{d}\}$ (resp. $\{x, \bar{x}\}$, $\{y, \bar{y}\}$) is a sublanguage of the Dyck language on the same alphabet.

Lemma 2. *Let S be a pseudoknotted structure, and w be the word on $\{c, d, \bar{d}, x, \bar{x}, y, \bar{y}\}$ that represents S. Then every simple pseudoknot in S is represented by a subword v of w, such that*

$$v = x^n\, y^{m_1}\, \bar{x}^{n_1}\, y^{m_2}\, \bar{x}^{n_2}\, \ldots\, y^{m_k}\, \bar{x}^{n_k}\, \bar{y}^m\,,$$

where $n_1 + n_2 + \ldots + n_k = n$ and $m_1 + m_2 + \ldots + m_k = m$.

Remark that a H-type pseudoknots is a simple pseudoknot where $k = 1$. Thus every H-type pseudoknot in S is represented by a subword $v = x^n\, y^m\, \bar{x}^n\, \bar{y}^m$.

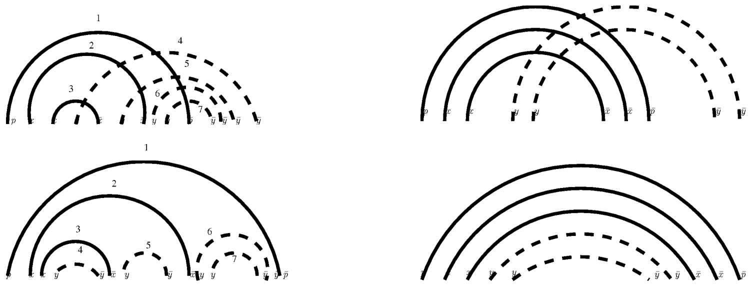
The following Proposition gives a way to encode any pseudoknotted structure where all pseudoknots are simple by a variant of the Motzkin language with for kinds of pairs of parentheses, that is on the alphabet $\{c, p, \bar{p}, d, \bar{d}, x, \bar{x}, y, \bar{y}\}$.

Proposition 1. *Let S be a pseudoknotted structure, and w be the word on $\{c, d, \bar{d}, x, \bar{x}, y, \bar{y}\}$ that encodes S. Then w can be encoded by a word on the alphabet $\{d, \bar{d}, x, \bar{x}, y, \bar{y}\} \cup \{p, \bar{p}\}$ where every subword $v = x^n\, y^{m_1}\, \bar{x}^{n_1}\, y^{m_2}\, \bar{x}^{n_2}\, \ldots\, y^{m_k}\, \bar{x}^{n_k}\, \bar{y}^m$, corresponding to a H-type pseudoknot is replaced with $v' = px^{n-1}\, y^{m_1}\bar{y}^{m_1}\, \bar{x}^{n_1}\, y^{m_2}\bar{y}^{m_2}\, \bar{x}^{n_2}\, \ldots\, y^{m_k}\bar{y}^{m_k}\, \bar{x}^{n_k-1}\bar{p}$.*

In particular , every subword $v = x^n\, y^m\, \bar{x}^n\, \bar{y}^m$ corresponding to a simple pseudoknot is replaced with $v' = px^{n-1}\, y^m\, \bar{y}^m\, \bar{x}^{n-1}\bar{p}$. The new letters p and $\bar{p}$ mark the beginning and the end of a pseudoknot. They are necessary to avoid ambiguity in the case of nested pseudoknots.

The proof is straightforward, as there is an immediate one-to-one correspondance between the two kinds of words below. The transformation is illustrated in Figure 3(a) and Figure 3(b), respectively, for simple pseudoknots and for the particular case of H-type pseudoknots.

Now we can define an unambiguous context-free grammar that generates the language which encodes the recursively pseudoknotted structures, with simple pseudoknots.



(a) Simple pseudoknot (b) H-type pseudoknot

Figure 3: Top: the pseudoknot and its encoding v. Bottom: the corresponding nested structure and its encoding v' given by Proposition 1.

Proposition 2. *The following unambiguous context-free grammar generates the language which encodes the recursively pseudoknotted structures, with simple pseudoknots:*

$$\begin{array}{rcl} S & \rightarrow & dS\bar{d}S|cS|P \\ P & \rightarrow & pSX\bar{p}S|\epsilon \\ X & \rightarrow & xSX\bar{x}SY|yYS\bar{y}S \\ Y & \rightarrow & ySY\bar{y}S|\epsilon \end{array}$$

The three rules in the first line allow to generate unpaired bases and non crossing edges, and to place pseudoknots anywhere. The other rules generate words which correspond to the code for a simple pseudoknot.

3.2 A linear decoding algorithm

At first, we present an algorithm that takes as input the word u that represents a unique simple pseudoknot, and constructs the pseudoknot. In other words, the algorithms constructs $f^{-1}(u)$, where f is the encoding function defined in Proposition 1. The words are considered as arrays of characters. The principle of the algorithm is very simple, in two steps. During the first step the word u is read from left to right, and the algorithm writes in the same order all the letters but the $\bar{y}$'s, and replaces the p and the $\bar{p}$ by x and $\bar{x}$, respectively. During the second step, the $\bar{y}$'s are written at the end of the word.

Now we can write the Algorithm 2 that takes as input the encoding u of a recursively pseudokotted structure, and gives the pseudoknot $f^{-1}(u)$. As any pseudoknot can embed other pseudoknotted structures, a stack is used. Each element of the stack will contain the list of the positions of the vertices of a given pseudoknot in the structure. When the end of the encoding of a pseudoknot is reached, it is popped out and the procedure *CrossSubword* is called (Algorithm 3). This procedure is quite similar to the Algorithm 1, the only difference is that the positions of the vertices of the pseudoknot in the whole word u that contains the pseudoknot are taken into account.

Algorithm 1 Constructing a simple pseudoknot from its noncrossing encoding

Require: a word u of length n that encodes a pseudoknot
Ensure: $u \leftarrow f^{-1}(u)$
  $j \leftarrow 1$
  for $i = 1$ to n **do**
    if $u_i \in \{x, \bar{x}, y\}$ **then**
      $u_j \leftarrow u_i$
      $j \leftarrow j + 1$
    else if $u_i = p$ **then**
      $u_j \leftarrow x$
      $j \leftarrow j + 1$
    else if $u_i = \bar{p}$ **then**
      $u_j \leftarrow \bar{x}$
      $j \leftarrow j + 1$
    end if
  end for
  for $k = j$ to n **do**
    $u_k \leftarrow \bar{y}$
  end for

Algorithm 2 Constructing a simple recursive pseudoknotted structure from its noncrossing encoding

Require: a word u of length n
Ensure: $u \leftarrow f^{-1}(u)$
  for $i = 1$ to n **do**
    if $u_i = p$ **then** {Beginning of a pseudoknot}
      Create new list L
      Add i to L
      Push(L)
    else if $u_i \in \{x, \bar{x}, y, \bar{y}\}$ **then** {Inside a pseudoknot}
      Pop(L)
      Add i to L
      Push(L)
    else if $u_i = \bar{p}$ **then** {End of a pseudoknot}
      Pop(L)
      CrossSubword(u,L)
    end if
  end for

3.3 Experiments

Now it is easy to generate random pseudoknotted structures of a given size. Starting from a non ambiguous grammar, we use the *GenRGenS* software [15] to generate words

Algorithm 3 CrossSubword

Require: a word u of length n, a list $L = (q_1, q_2, \ldots, q_m)$ of positions in u
Ensure: $u_{q_1} u_{q_2} \ldots u_{q_m} \leftarrow f^{-1}(u_{q_1} u_{q_2} \ldots u_{q_m})$

```
j ← 1
for i = 1 to m do
    if u_{q_i} ∈ {x, x̄, y} then
        u_{q_j} ← u_{q_i}
        j ← j + 1
    else if u_{q_i} = p then
        u_{q_j} ← x
        j ← j + 1
    else if u_{q_i} = p̄ then
        u_{q_j} ← x̄
        j ← j + 1
    end if
end for
for k = j to n do
    u_{q_k} ← ȳ
end for
```

that encode pseudoknots, then we decode them with the Algorithm 3. The generation is uniform if we use a classical context-free grammar. It can also be non-uniform (in a controllable fashion) if we use a *weighted* context-free grammar [7].

Figure 4 shows some examples of random structures that have been generated with the grammar of Proposition 2.

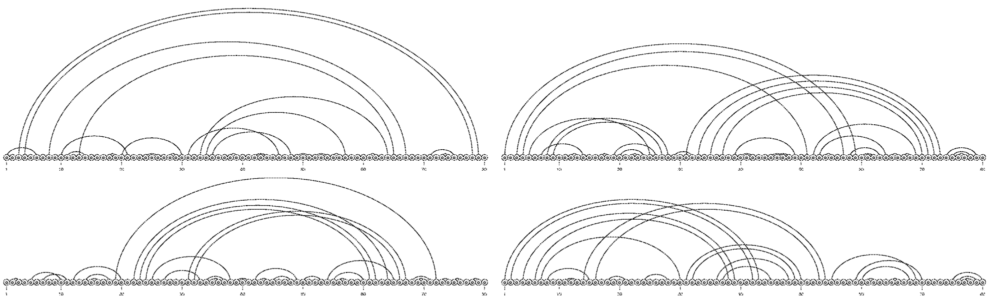


Figure 4: Four random structures generated with the grammar of Proposition 2.

In order to generate more realistic structures, we designed a more complex grammar in such a way to favour long stems (i.e. series of consecutive basepairs):

$$\begin{array}{rcl}
S & \rightarrow & T|cS|P \\
T & \rightarrow & N|Q \\
N & \rightarrow & dN\bar{d}S|QT \\
Q & \rightarrow & dQ\bar{d}|dddccc\bar{d}\bar{d}\bar{d} \\
P & \rightarrow & pxxXS\bar{x}\bar{x}\bar{p}S|\varepsilon \\
X & \rightarrow & xSX\bar{x}S|yyySY\bar{y}\bar{y}\bar{y}S \\
Y & \rightarrow & ySY\bar{y}S|\varepsilon
\end{array}$$

And we weighted the grammar in order to obtain, in average, more pseudoknots than in the uniform model. Two examples of generated structures are shown in Figure 5. Other weighted grammars are being investigated for getting more realistic structures

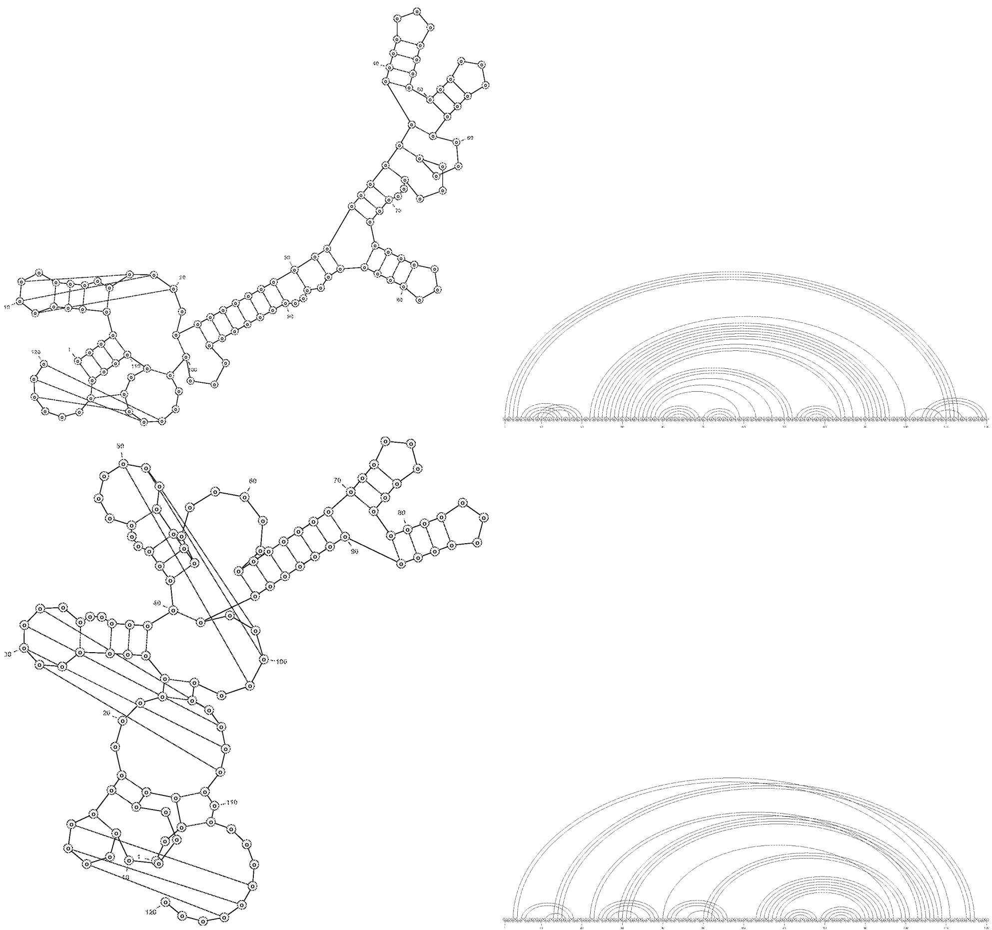


Figure 5: Two random structures generated with a weigthed grammar. Left: classical biological representation. Right: arc-annotated representation.

compared to real biological ones.

Acknowledgements

This research was supported in part by the ANR project BRASERO ANR-06-BLAN-0045, and by the Digiteo project "RNAomics".

References

[1] T. Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104:45–62, 2000.

[2] J. Allali, Y. d'Aubenton Carafa, C. Chauve, A. Denise, Ch. 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 *Actes des Journées Ouvertes de Biologie, Informatique et Mathématiques - JOBIM'08*, pages 67–68, 2008.

[3] S.F. Altschul and B.W. Erickson. Signifiance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleotide and codon usage. *Mol. Biol. Evol.*, 2(6):256–538, 1985.

[4] A. Condon, B. Davy, B. Rastegari, S. Zhao, and F. Tarrant. Classifying RNA pseuknotted structures. *Theorical computer science*, 320:35–50, 2004.

[5] E. Coward. Shufflet: Shuffling sequences while conserving the k-let counts. *Bioinformatics*, 15:1058–1059, 1999. Programme : http://www.genetique.uvsq.fr/eivind/shufflet.html.

[6] K. Darty, A. Denise, and Y. Ponty. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25(15):1974–1975, Apr 2009.

[7] A. Denise, Y. Ponty, and M. Termier. Controlled non uniform random generation of decomposable structures. *Theoretical Computer Science*, (to appear), 2010.

[8] Y. Ding and E. Lawrence. A statistical sampling algorithm for rna secondary structure prediction. *Nucleic Acids Research*, 31(24):7280–7301, 2003.

[9] W.M. Fitch. Random sequences. *Journal of Molecular Biology*, 163:171–176, 1983.

[10] Ph. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theor. Comput. Sci.*, 132(2):1–35, 1994.

[11] D. Kandel, Y. Matias, R. Unger, and P. Winkler. Shuffling biological sequences. *Discrete Applied Mathematics*, 71:171–185, 1996.

[12] N. Leontis and E. Westhof. Geometric nomenclature and classification of RNA base pairs. *RNA*, 7:499–512, 2001.

[13] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Networks motifs : Simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[14] Y. Ponty. Efficient sampling of rna secondary structures from the boltzmann ensemble of low-energy: The boustrophedon method. *Journal of Mathematical Biology*, 56(1-2):107–127, Jan 2008.

[15] Y. Ponty, M. Termier, and A. Denise. Genrgens: Software for generating random genomic sequences and structures. *Bioinformatics*, 22(12):1534–1535, June 2006.

[16] Pere Puigbo, Ignacio Bravo, and Santiago Garcia-Vallve. E-CAI: a novel server to estimate an expected value of Codon Adaptation Index (eCAI). *BMC Bioinformatics*, 9(1):65, 2008.

[17] E. Rouchka and C. T. Hardin. rMotifGen: random motif generator for DNA and protein sequences. *BMC Bioinformatics*, 8(1):292, 2007.

[18] C. Saule, M. Régnier, J.-M. Steyaert, and A. Denise. Counting RNA pseudoknotted structures (extended abstract). In *Proc. 22th Conference on Formal Power series and Algebraic Combinatorics (FPSAC)*. San Francisco State University, 2010.

[19] M. Vauchaussade de Chaumont and X.G. Viennot. Enumeration of RNA's secondary structures by complexity. In V. Capasso, E. Grosso, and S.L. Paven-Fontana, editors, *Mathematics in Medecine and Biology*, volume 57 of *Lecture Notes in Biomathematics*, pages 360–365, 1985.