Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 10, no. 2, pp. 329–351 (2006)

Efficient drawing of RNA secondary structure

David Auber Maylis Delest
Jean-Philippe Domenger Serge Dulucq
LaBRI - Université Bordeaux 1
www.labri.fr
auber@labri.fr maylis@labri.fr domenger@labri.fr dulucq@labri.fr

Abstract

In this paper, we propose a new layout algorithm that draws the secondary structure of a Ribonucleic Acid (RNA) automatically according to some of the biologists' aesthetic criteria. Such layout insures that two equivalent structures (or sub-structures) are drawn in a same and planar way. In order to allow a visual comparison of two RNAs, we use an heuristic that places the biggest similar part of the two structures in the same position and orientation.

Article Type	Communicated by	Submitted	Revised
Regular paper	M. T. Goodrich	March 2005	June 2006

Research supported in part by ACI Masse de données, Navgraph project.

1 Introduction

Ribonucleic Acid (RNA) is an important molecule, which performs a wide range of functions in biological systems. Some RNA is found in the nucleus (where it is synthesized), and in the cytoplasm, as messenger RNA or mRNA (which carries the genetic information out of the nucleus), transfer RNA or tRNA (which decodes the information), ribosomal RNA or rRNA (which was found in the ribosome of cells). These forms of RNA are involved in the protein synthesis.

RNAs recently became the center of much attention because of its catalytic properties, leading to an increased interest in obtaining structural information. For example, RNA contains genetic information of viruses such as HIV and therefore regulates the functions of such viruses.

An RNA is characterized by its base sequence and higher order structural constraints. It can be considered at three levels of detail :

- its linear sequence of monomers is the primary structure,
- its secondary structure is a two dimensional drawing that reflects major links acting in the RNA,
- its tertiary structure is the three dimensional view where the positions of atoms are obtained using crystallographic method.

The RNA tertiary structure is often much more highly conserved than the sequence during evolution. In addition, secondary and tertiary structural features of RNAs are important in the molecular mechanism involving their functions. The biologists assume that, for a given RNA, a common function to several species corresponds to a preserved molecular conformation of their RNA and, thus, to a preserved secondary and tertiary structure.

Thus, knowledge of RNA secondary structure is increasingly becoming important in molecular phylogenetic studies, particularly in assisting accurate sequence alignment [22] that is detecting similar parts between two linear sequences. Automatic alignment methods that use only primary sequences may misalign RNA sequences [17] while alignments that take secondary structure into consideration can generate improved phylogenetic trees [22].

Therefore the ability to draw and visually compare RNA structures is useful. Figure 1 shows an example of hand-drawn RNA secondary structure coming from a biologic data base [6]. It refers to structural motifs such as stems, hairpins, bulges, interior loops and multi-branch loops. The goal to achieve is to perform an automatic drawing that respects the biologists' habit: there are no crossing edges (planar graph), the structural patterns appear clearly.

In the past ten years, many interactive programs allowing interaction with the RNA drawing were developed. The more recent programs RNAView [32] and RNAViz [26] offer many functionalities such as editing, energy computation, and even three dimensional representation. None of them produces an automatic planar drawing of an RNA secondary structure that corresponds to the biologists' practice. As a consequence, in these programs, the layout is changed manually and thus, even if two RNAs are similar to each other, the drawing can be very different because it is user dependant.

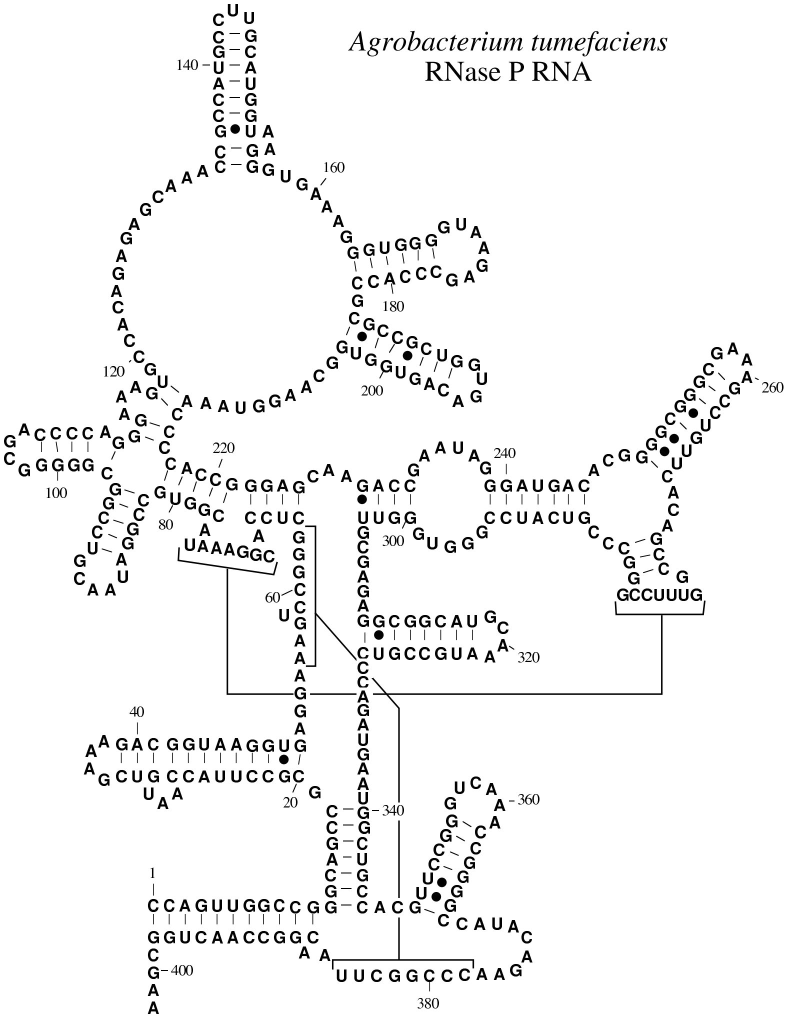


Figure 1: Drawing of a *RNA* coming from a biological data base.

In this paper, we propose an algorithm that partially solves the automatic RNA drawing respecting the biologists' habit. This algorithm is based upon the association of the RNA secondary structure with a tree [29](see Figure 3). This tree encodes the structural motifs of RNA. Note that recently an efficient alignment program RNAForester [18] has appeared, which is based on this association. Due to the underlying tree, it is stable in the sense that small changes on an RNA primary structure will not drastically change the drawing.

Efficient comparison must allow the user to visually compare two RNA secondary structures. One important requirement to facilitate visual comparison of structures is to automatically detect parts of the structure that have the same shape and to place them at the same position and in the same orientation in the final drawing.
This is useful for the user because it will give him reference marks when he will compare the structures. Thus, in order to present the biologist face to two RNAs with the same orientation on the screen, we use an heuristic based on quasi-isomorphic subtrees in a tree. This heuristic has been successfully used at the Infovis'03 contest[3] and is briefly described here. It allows us to place the largest similar part of two RNAs in a similar position on a portion of the screen (for example at a same relative position from the upper left corner).

In what follows, we first describe the biological background. Then, we describe the drawing algorithm and the heuristic for presenting two RNAs. Finally, we conclude with future work to be done in order to match the requirements of biologists.

2 RNA background

An RNA molecule is a linear polymer in which the monomers - (ribo)nucleotides - are linked together by means of phosphodiester bridges, or bonds. Each nucleotide contains a base: the four different bases are Adenine (A), Cytosine (C), Guanine (G) and Uracil (U). An RNA sequence R of n nucleotides can be represented as a word of length n on the alphabet {A, C, G, U}: $R = r_1.r_2 \ldots r_n$ where r_i is the i-th (ribo)nucleotide belonging to the alphabet. We will refer to i as the i^{th} base of the sequence.
Although each RNA molecule has only a single polynucleotide chain, it is not a smooth linear structure. It has extensive regions of *base pairs*. The complementary bases, A-U and G-C form stable *base pairs* with each other through the creation of hydrogen bonds between donor and acceptor sites on the bases. These are called *Watson-Crick* base pairs whereas the weaker base pair G-U is the *wobble pair*. All of these are called *canonical base pairs*. Other non canonical base pairs occur (e.g. A-C and U-U), some of which are stable.

Thus, the *secondary structure* of an RNA molecule can be viewed as the list of base pairs that occur in its three dimensional structure. In what follows, we will consider a secondary structure on R as a set P of ordered pairs $(i, j), 1 \leq i < j \leq n$, satisfying:

- $j - i > 3$
- if (i, j) and (k, l) are two base pairs, (assuming without loss of generality that $i \leq k$), then either:
  - $i = k$ and $j = l$ (they are the same base pair),
  - $i < j < k < l$ that is (i, j) precedes (k, l), or
  - $i < k < l < j$ that is (i, j) includes (k, l).

Note that the last condition excludes base pairs (i, j) and (k, l) such that $i < k < j < l$, that is when the two base pairs overlap. A set of base pairs $((i+k, j-k))_{0 \leq k \leq m}$ overlaping a pair (k, l) is called *pseudo-knot*. Pseudo-knots are often considered as belonging to tertiary structure. However, pseudo-knots are real and important structural features.

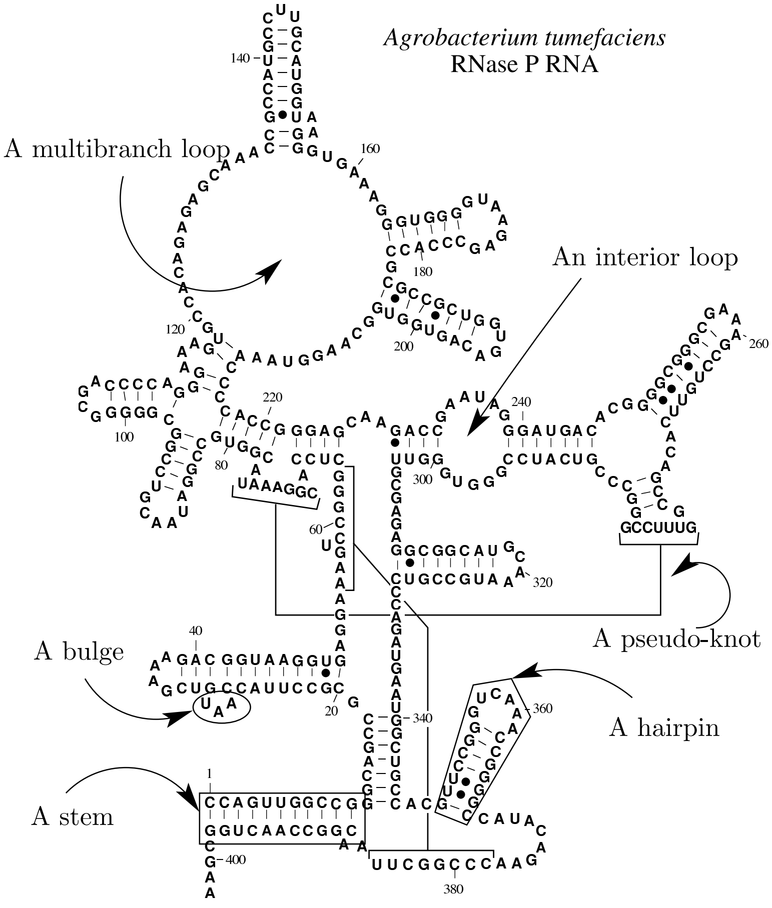


Figure 2: Motifs in a secondary structure.

The RNA secondary structure refers to **structural motifs** such as stems(**S**), hairpins(**H**), bulges (**B**), interior loops (**I**) and multi-branch loops(**M**). RNA stems are self-complementary base-paired regions (A-U, U-A, G-C, C-G), whereas hairpins and bulges are regions with unpaired bases; RNA *junctions* (interior and multi-branch loops) are the place where two or more stems *meet*, and they contain unmatched bases. The overall molecular architecture of the secondary structure is mainly stabilized by the canonical base pairs A-U, G-C, and G-U. See Figure 2.

Following [21], a consequence is that an RNA secondary structure without pseudo-knots can be represented as an ordered tree in which each node is labeled and the left to right order among the sibling nodes is significant. The labels of the nodes can represent:

- either structural motifs,
- or nucleotides A, C, G, U, Watson-Crick pairs, wobble pairs, ...

Represented below are (see Figure 3):

1. on the left, an RNA secondary structure, the vertices represent the nucleotides,
2. in the center, a relatively rough tree representation of this structure where the labels refer to its structural motifs,
3. on the right, a coding of the same structure, the tree with appropriate labeling of the nodes that makes it possible to come back to the secondary structure [29, 27].

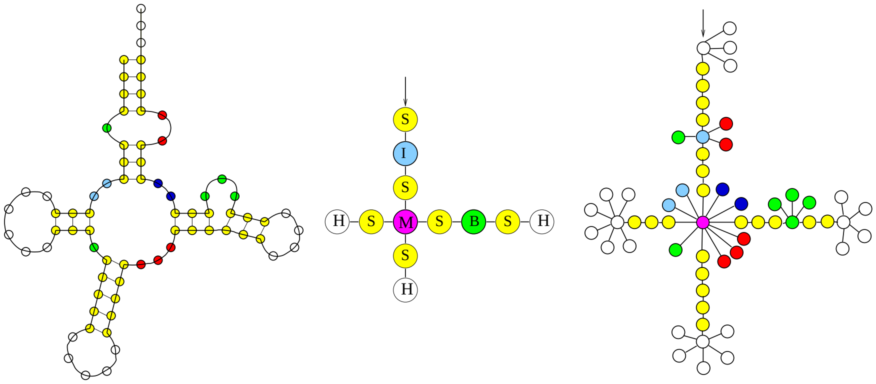


Figure 3: A secondary structure and its tree representations.

Note that this tree representation can be rough (case 2) or refined (case 3).

Since the RNA secondary structure appears as a tree-like structure, there exist works comparing them using tree comparison. Many measures have been proposed for the similarity of two trees, e.g. tree edit distance, constrained edit distance and alignment of trees [9, 10, 21, 23, 34].

Other related measures can be found in [20, 24, 30]. Alignment of trees is a straightforward extension of sequence alignment that was proved to be different from tree edit distance [21].

An important requirement in a drawing process is to use the knowledge of users in order to have no conflict with existing representation of the data in the mind of the users. In this particular case, biologists have a long habit of a standard representation of such structures. The drawing of the secondary structure must be planar, the loops (bulges, hairpins, interior and multi-branch) in the structure should be drawn on circles, and the stems should be drawn on a straight line. Furthermore, the edge length should be constant. Figure 1 shows a hand drawing of a secondary structure made by a biologist. These criteria are equivalent to minimizing three well-known graph drawing criteria which are: the angular resolution, the standard deviation of the edge length, and the number of crossings. For more information about graph drawing aesthetic criteria one can refer to [8]. Classical planar drawing algorithms such as the one proposed in [16] do not correspond to the intuitive representation of biologists.

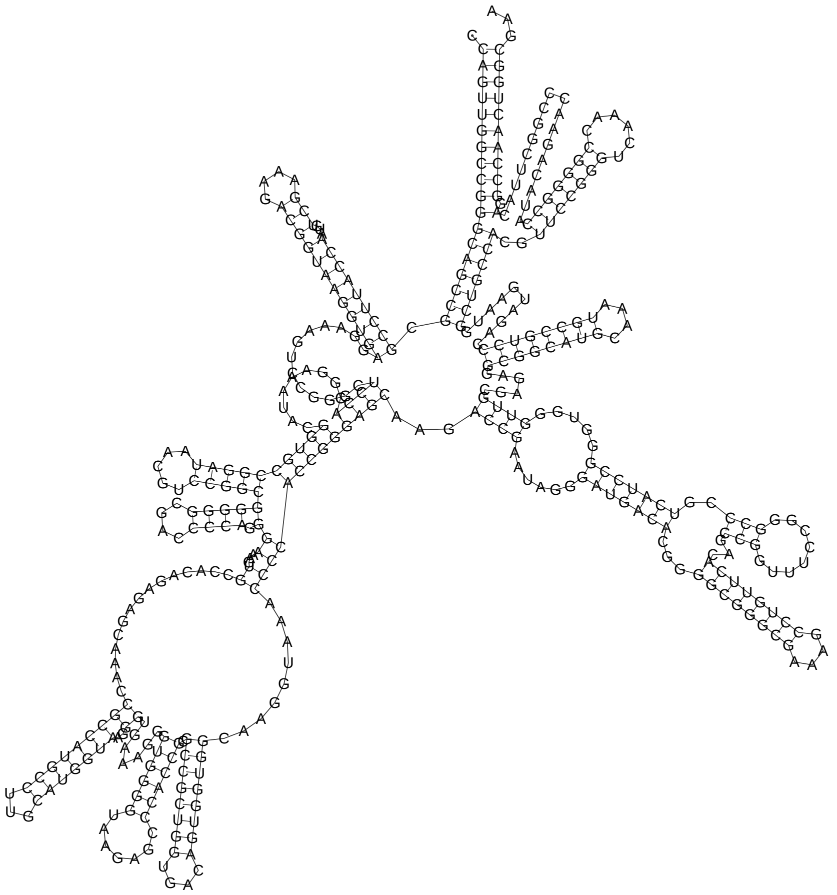


Figure 4: Drawing the Agrobacterium tumefaciens with Vienna RNA software.

Other representations can be found in RNA visualization programs [26, 32], however, even if these representations are closer to the biologists' requirements, some important criteria such as the size of the drawing and the number of crossings are not properly taken into account. Figure 4 shows the result of the layout proposed by one of the latest program, the Vienna package. On this drawing, one can see that there are crossings and that loops are not always drawn on a circle. Thus RNAViz has some tools allowing the user to edit the drawing.

3 Drawing RNA secondary structure

In order to highlight our algorithm, we have labelled by X, Y and Z the three main multibranch loops on the RNA hand-drawing (see Figure 5). As described in section 2, secondary structure of RNA is a chain of nucleotides n_1, $n_2, \cdots, n_{k-1}, n_k$ where all n_i, n_{i+1} are linked together and where several links can exist between n_i, n_j with $j \neq i+1$. Figure 6 shows a sequence of nucleotides and links. We call P_{link} (primary structure) the links belonging to the sequence (horizontal links), S_{link} (secondary structure) the links that form stems and T_{link} (tertiary structure) the links that create crossings. In Figure 6, the set of P_{links} are the horizontal edges, $S_{link} = 2, 3, 5, 6$ and $T_{link} = 1, 4$. If one looks to the structure of RNAs, one can see that the set of nucleotides lie on the outer face of the graph. Thus, the union of P_{link} and S_{link} is an outer-planar biconnected graph. Outer-planar graphs are planar graphs where all vertices lies on a same face (here the external one). One of the nice properties of such a graph is that one can extract a tree as shown in paragraph 2. Thus, after transforming the graph in an outer planar graph (building of the T_{link} set) our algorithm uses that property in order to reduce the problem of RNA drawing to a problem of tree drawing with a specific set of aesthetic criteria. In what follows, we detail the steps of the algorithm.

3.1 Outer-planarization

In the first step, we remove the edges (T_{link}) in order to get an outer-planar graph. This can be done using the biologic data basis which point out pseudo-knots or using RNAView. But, knowing that the P_{link} set forms a Hamiltonian path in the graph, we can deduce that this path must be the outer-face of the outer-planar graph. Thus, the problem is transformed to the problem of finding a minimum set of edges that enables us to draw the graph without crossing. In Figure 6 one can see the drawing of a graph on one page (that is each edge is drawn up the sequence). Finding this minimum set consists of building a conflict graph. In this graph a node represents an edge and an edge represents a conflict in the one page drawing. We say that an edge e_1 is in conflict with another edge e_2 if and only if e_1 intersects e_2 in the one page drawing.

If the conflict graph is bipartite, one can compute T_{link} easily. Let S_1 and S_2 be the sets of vertices such that for all vertices $u \in S_1$, there exists a vertex

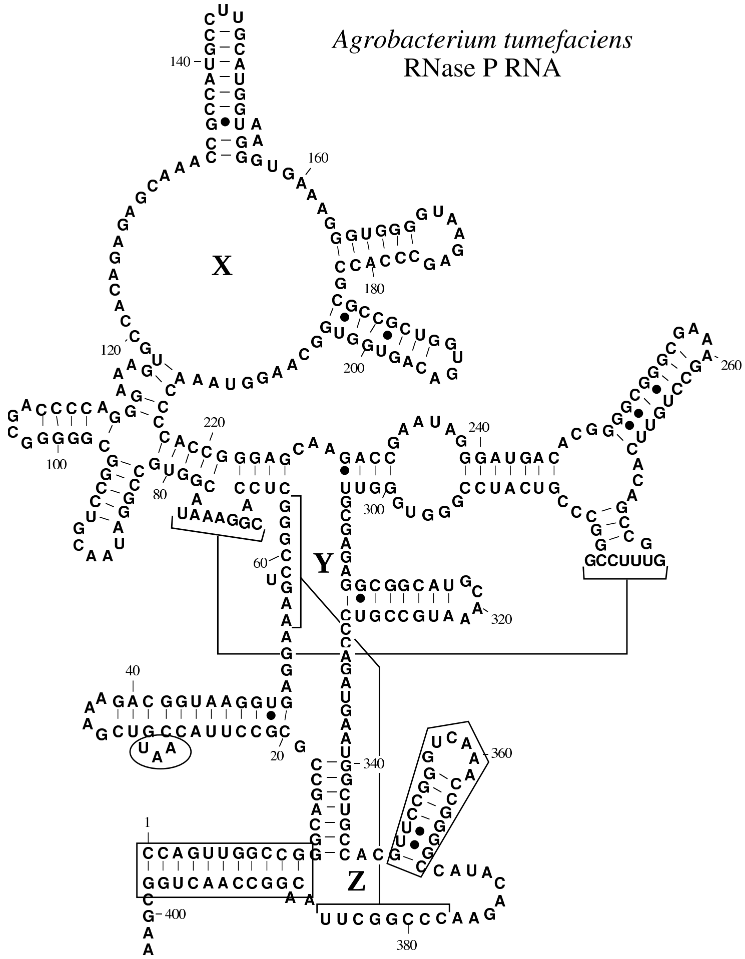


Figure 5: The three main multibranch loops.

$v \in S_2$ such that u crosses v. We obtain T_{link} by choosing the smallest set. If the conflict graph is not bipartite the problem of finding a maximum induced bipartite subgraph is NP-complete. In order to solve this problem one can use the heuristic proposed in [2].

Figure 6 shows a conflict graph. From this graph, we remove the edges 1 and 4 ($T_{link} = \{1, 4\}$). One can see in Figure 7 that if the user wants to visualize pseudo-knots (T_{link}), these edges can be placed using the third dimension after the execution of the drawing algorithm.

After removing the T_{link} edges one can compute the tree representation shown on the right of the figure 3. In what follows, we explain how to draw this tree in order to obtain a layout that respects biologist aesthetic criteria. During

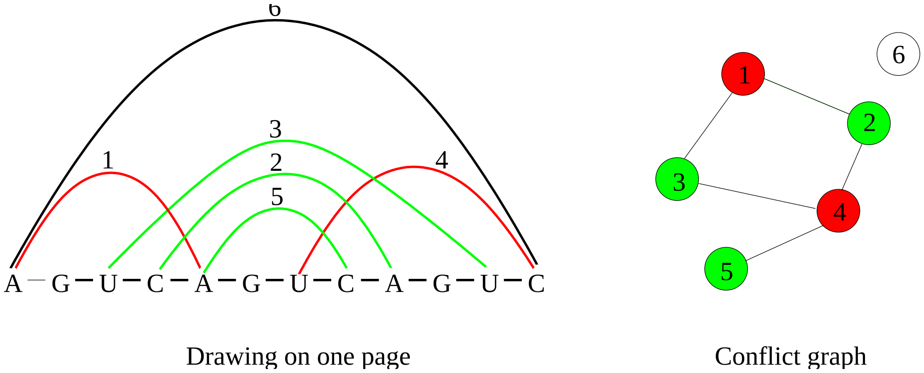


Figure 6: One page drawing and its conflict graph.

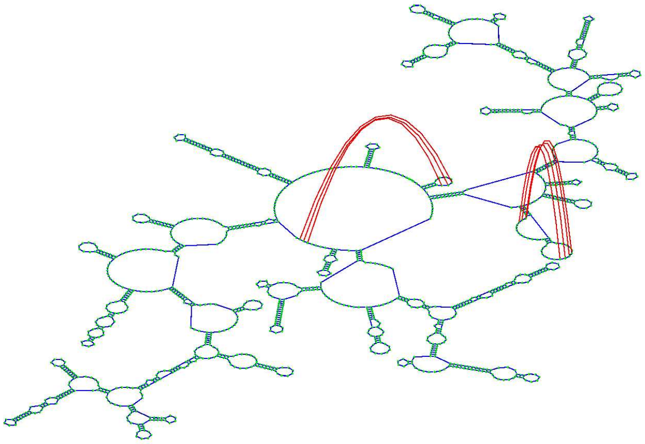


Figure 7: Visualization of pseudo-knots.

our experimentation with data coming from biological data base, the conflict graph was always bipartite and moreover our algorithm agrees in all cases with biological data basis on the pseudo-knots.

3.2 Tree drawing

The tree drawing algorithm we have designed is recursive. At each step we first layout all the sub-trees induced by the children of a node. Then, we place all these subtrees drawings on a circle. Literature about similar drawing algorithms can be found in [7, 14]. In our approach, instead of using a circle hull, we use a packing method inspired by Reingold and Tilford [25] for an efficient hierarchical tree drawing. However, in our case, the packing is more complex because it is a circular drawing and not a hierarchical one. The main problem is to determine a first coarse radius r of the circle on which one places the subtree drawings. In order to compute this radius, we compute the circumference of the circle on which the sub-trees are placed. To find the circumference, we first pack the sub-drawing as a hierarchical drawing does. Then, the width of the bounding box gives this first circumference. This circumference can be used to obtain the drawing. However a lot of space is lost because changing from a line to a circle gives more space to use (see Figure 8). Figure 9 shows the result of the packing at the root level and the result on the drawing of the RNA of Figure 1.

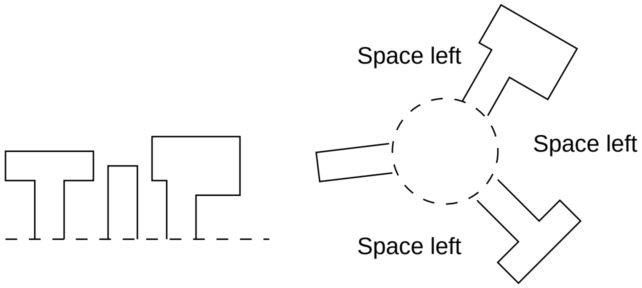


Figure 8: Lost space in a simple packing.

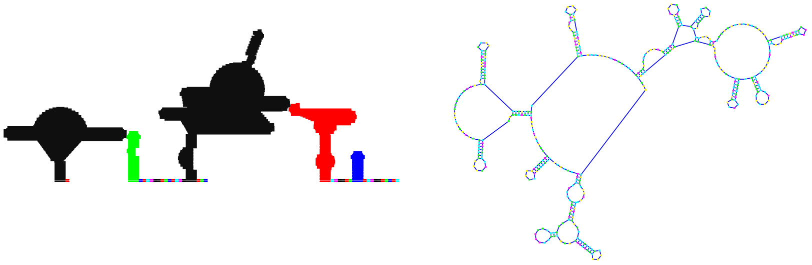


Figure 9: Simple packing.

To take this free space into account, we then pack the sub-drawing on a circle instead of a line. This operation can be done using the same method of packing but with a transformation from Cartesian coordinates to polar coordinates.

At some level, the drawing can be seen as a sequence of trees whose basis is on a line (see Figure 10 left). For each motif, we compute the real occupied sector in order to deform the motif. Let us consider the two points $M(x_M, 0)$ and $N(x_N, 0)$ which are on the basis of a motif. Let $I(x_I, 0)$ the middle of the segment $[M, N]$. For each point $P(x_P, y_p)$ that delimitates the motif, we compute the transformed point $TP(x_{TP}, y_{TP})$ by

$$x_{TP} = r * arcsin((x_I - x)/(y_{TP} + r)) + x_I, \; y_{TP} = \sqrt{(x_I - x_P)^2 + (y_P + r)^2} - r.$$

One can see in Figure 10 right the effect of the transformation. This transformation then allows to compute a new bounding box that gives in turn a new radius r_T which is smaller than r. We then pack the transformated motifs and apply this method until a stable circumference is obtained. Figure 11 shows the packing of the tree and the result on the entire structure. To obtain an efficient solution we have implemented this packing algorithm with a scan line algorithm which enables us to obtain drawings of the biggest RNA structures in a few seconds.

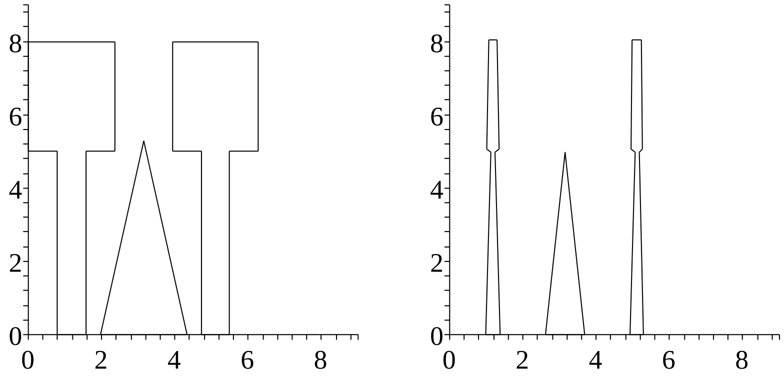


Figure 10: Polar transformation.

To prevent overlapping in the final drawing one must force sub-trees to be drawn outside of the circle. This restriction introduces a lot of long edges in the final drawing but is necessary in order to have a good angular resolution in the drawing. In Figure 12 one can see this phenomena on two stems. In order to obtain a drawing that matches the aesthetic criteria of the biologist one must make a trade-off between these two aesthetic criteria (edge length, angular resolution). We obtain this trade-off by forcing the drawing of the subtree to be on a semi circle. This is done by adding in the packing algorithm two sub-trees (or shape) at the left and at the right. This way also take into account the particular case of sub-trees composed of one node. In Figure 12 one can see these two shapes in black at the left and at right of the drawing. Even if the semi circle seems to be the best solution according to our experimentation one

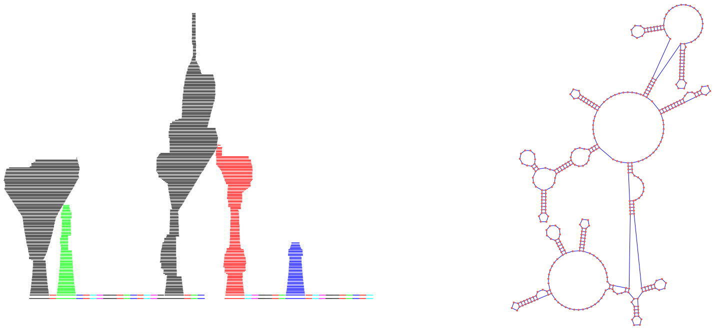


Figure 11: Polar packing.

can use this as a parameter to choose the trade-off between the edge length and the angular resolution.

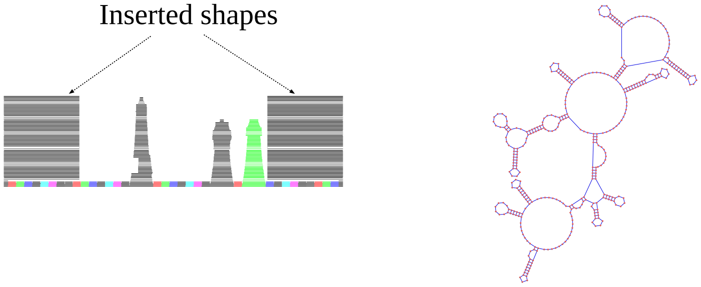


Figure 12: (left) Shape insertion, (right) Bad edge length.

3.3 Final drawing

In Figure 13, the tree is displayed in blue and the sequences in red. One can see that several nodes are shared between the tree and the sequence (green), thus to obtain the final drawing we have to set coordinates for the nodes that form stems (red). Knowing that each of these nodes is associated with a node η of the tree (grey box), we can compute the position of these nodes by placing them on a line orthogonal to the line formed by the coordinates of η and $father(\eta)$. This operation is straightforward using a cross product. The main problem now is to determine the distance between the node and η. In order to prevent the nodes from overlaping, we set the size of η three times bigger (one node, a space and one node) in the tree drawing algorithm. Thus, we are sure that if a node is at distance $\frac{3}{2}$ from η, no overlapping will occur. Figure 14 shows the result of our algorithm on the graph of the Figure 1 and the Figure 15 shows both side by side. Note that it would be easy now to deforme the circles in order to fit the hand drawing.

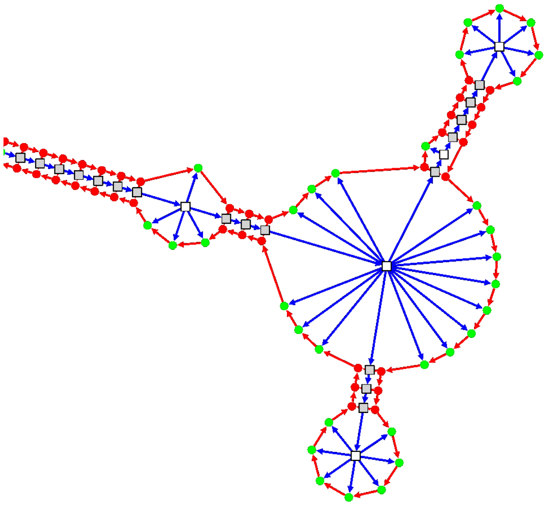


Figure 13: Sets of nodes.

4 RNAs pairwise Placement

In this section, we use a combination of metrics in order to predict common parts in several RNAs in order to help the user in the alignment process. We have shown in section 2 that the secondary structure of an RNA is associated with a tree. At the Infovis'03 Conference contest [12] on pairwise comparison of trees, an assigned task was to find similar subtrees that have moved :

- the subtrees are not in the same place in the hierarchy,
- slight changes occure between the two subtrees

First note that "finding isomorphic subtrees in a tree" or "common subtrees in several trees" are one and the same task. In the last case, one just needs to construct a tree with a vertex (its root), which has subtrees that are the trees to be compared. Thus, using the underlying tree of two RNAs, "find similar parts on RNAs" can be turned to "find similar subtrees in a tree". Works has already been done based on vertices degrees by Zemlyachenko [33] and then by Dinitz et al. [31]. These algorithms give a partition of subtrees into isomorphism equivalence classes. They are proved to be linear. However, they only detect isomorphism and do not provide a measure of similarity for subtrees. More

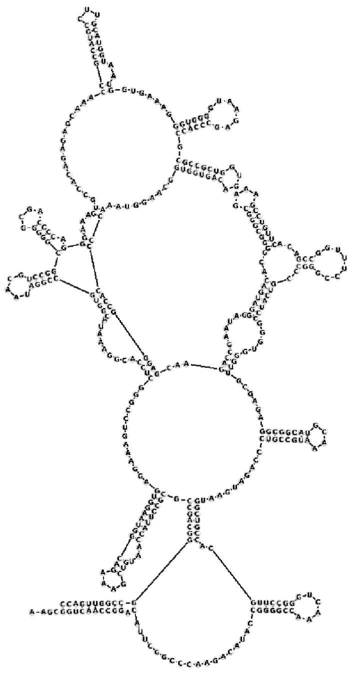


Figure 14: Drawing Agrobacterium tumefaciens with our algorithm

recently, Gupta et al. [15] gave a nice algorithm for determining the largest tree embeddable in two trees. It would be usable in our context but the complexity of their algorithm is $O(n^2)$ for non parallel computer. In our case, we want a faster algorithm because the RNA placement is computed on data bases (several thousands of RNAs) and this algorithm must also work on a personal computer. Moreover, we do not want the largest one but mainly the largest similar one, which means that we accept some slight changes. In order to give a response to the Infovis'03 task, we have designed an heuristic [3] that can suggest by colors meaning similar parts in a tree (similar subtrees have same colors). Thus, we are able to use the graph drawing algorithm of section 3 to automatically display two RNAs and (using our heuristic) to color them in order to suggest to the user which parts of the RNAs are similar. The last task is to present the final images of the two RNAs such that the user can easily identify the common parts. We decided to place the center of the biggest (in term of nodes of the underlying subtrees) similar parts at the same coordinate. If there are several choices then we choose one at random. Thus we apply a rotation to the drawing. Below, we briefly describe the heuristic. Let s be a vertex. We compute the four following metrics :

- the degree of the vertex denoted by $\delta(s)$,

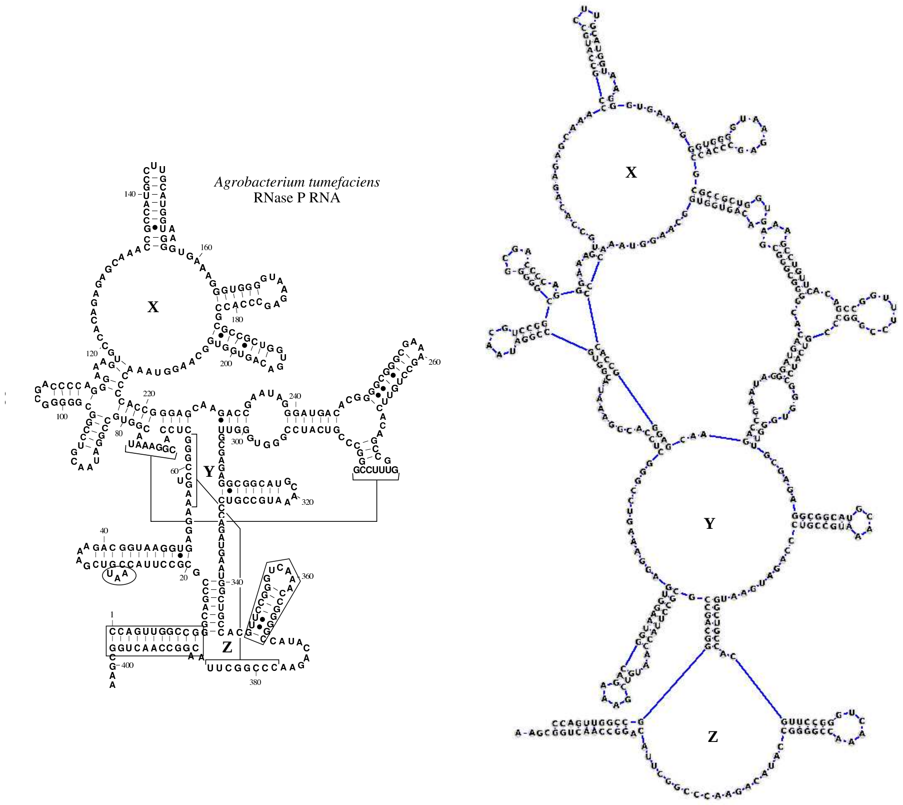


Figure 15: Hand drawing and automatic drawing.

- the number of nodes of the subtree with root s denoted by $\nu(s)$,
- the height of the subtree with root s denoted by $\eta(s)$
- the so-called Strahler number of a vertex denotes by $\sigma(s)$.

We briefly explain this last metric. The Strahler number σ_b has first been introduced on binary trees in some work about the morphological structure of rivers [19, 28]. A generalization on planar trees has been set up [4] using the nice interpretation by Ershov [11]. He proved that the Strahler number of the root of the binary tree incremented by one is exactly the minimal number of registers needed to compute an arithmetical expression given by the tree output by the syntactical analysis. Following this interpretation, for each internal vertex s having $k+1$ subtrees whose roots are $\{s_i\}_{0 \leq i \leq k}$ such that if $i \leq j$ then $\sigma(s_i) \geq$

$\sigma(s_j)$, the Strahler number $\sigma(s)$ is given by :

$$\sigma(s) = \begin{cases} 1 \text{ if } s \text{ is a leaf} \\ \\ \underset{0\leq i\leq k}{Max}(\sigma(s_i)+i) \text{ if } s \text{ has } k+1 \text{ subtrees with root } s_i \end{cases}$$

Of course, these four metrics are not in the same scale, thus we map them to the range [0..1] by a linear normalization. We will denoted by $\overline{\alpha}, \overline{\nu}, \overline{\eta}$ and $\overline{\sigma}$ the normalized metrics. The heuristic is now in two steps. The first one consists in classifying roughly the vertices of the tree on the basis of a function of the four metrics. Let ϵ be a real positive number. Two vertices s and s' are in the same ϵ-class if

$$\begin{aligned}&\left(\overline{\delta}(s)-\overline{\delta}(s')\right)^2+\left(\overline{\nu}(s)-\overline{\nu}(s')\right)^2+\left(\overline{\eta}(s)-\overline{\eta}(s')\right)^2\\&+\left(\overline{\sigma}(s)-\overline{\sigma}(s')\right)^2\leq\varepsilon\end{aligned}$$

Note that this is not an equivalence relation. All the leaves are in the same ϵ-class. Thus, in the algorithm we omit them. If ϵ is enough small, the ϵ-class of the root is reduced to itself. The second step aggregates the ϵ-classes of vertices that are part of similar subtrees. In order to do this, we construct a new integer metric π such that if two subtrees have quasi-similar parts then all the vertices s of these parts have the same value $\pi(s)$. The strategy consists in a top-down traversal on subtrees that stops as soon as there is no π valuation possible. Let $(K_i)_{0\leq i\leq p}$ be the set of all ϵ-classes not reduced to one element. Assume that

$$\forall i \in [0..p], K_i = s_{i_1}, s_{i_2}, \ldots, s_{i_{p_i}} \text{ with } \sigma(s_{i_1}) \geq \sigma(s_{i_2}) \geq \ldots \geq \sigma(s_{i_{p_i}}).$$

First, we define the τ-sets between two sets of vertices $\mathcal{S}$ and $\mathcal{S}'$. Let τ be a positive integer threshold, the τ-sets are given by the following algorithm.

```
Function $\tau$-sets($\mathcal{S}$, $\mathcal{S}'$):two sets of vertices
    $\mathcal{S}_1 = \emptyset$
    $\mathcal{S}_2 = \emptyset$
    for $i$ in $[0..p]$
        $\mathcal{S}'_1 = \mathcal{S} \cap K_i$
        $\mathcal{S}'_2 = \mathcal{S}' \cap K_i$
        if $|\mathcal{S}'_1| - |\mathcal{S}'_2| \leq \tau$
            $\mathcal{S}_1 = \mathcal{S}_1 \cup \mathcal{S}'_1$
            $\mathcal{S}_2 = \mathcal{S}_2 \cup \mathcal{S}'_2$
    return ($\mathcal{S}_1$,$\mathcal{S}_2$)
```

If τ is small enough then this function consists in selecting the vertices which have a same distribution of K-indices over the ϵ-classes. In what follows,we denote by $C(s)$ the set of children of a vertex s. Now, the next part of the agorithm consists from a τ-sets to evaluate π according to a given value α. In order to speed up the process in the last part of the algorithm the π metric is initialised to 0 for all vertices.

```
Function $\pi$-prolongation($\mathcal{S}$, $\mathcal{S}'$,$\alpha$):void
   for $s$ in $\mathcal{S} \cup \mathcal{S}'$
      $\pi(s) = \alpha$
   endfor
   $\mathcal{S}_1 = (\bigcup_{s \in \mathcal{S}} C(s))$
   $\mathcal{S}_2 = (\bigcup_{s \in \mathcal{S}'} C(s))$
   $(\mathcal{S}_1, \mathcal{S}_2) = \tau - sets(\mathcal{S}_1, \mathcal{S}_2)$
   if $(\mathcal{S}_1, \mathcal{S}_2) \neq (\emptyset, \emptyset)$
      $\pi$-prolongation($\mathcal{S}_1$,$\mathcal{S}_2$,$\alpha$)
```

Now, the general algorithm consist for each pair of vertices belonging to the same ϵ-classes to compute the τ-sets of their children and then to evaluate π.

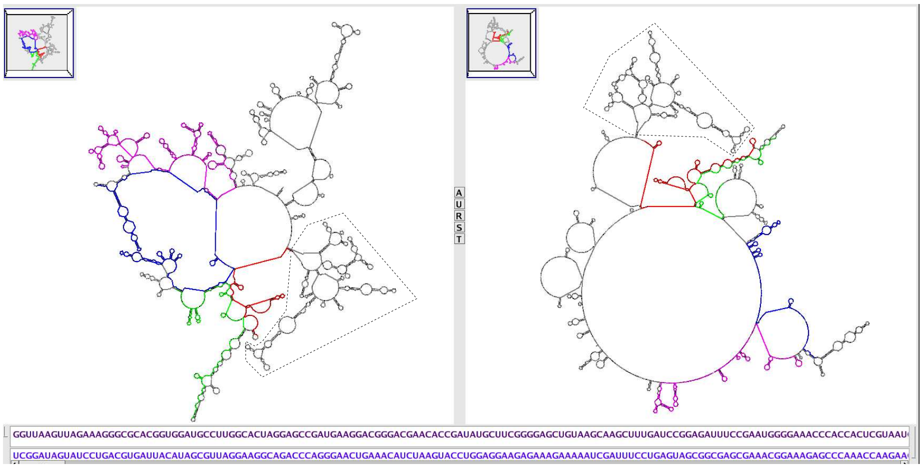
```
Function $\pi$-evaluation:void
 $\alpha = 0$
 for $i$ in $[0..p]$
  for $j$ in $[i_1..i_{p_{i-1}}]$
   if $\pi(s_j) \neq 0$
    $\alpha = \alpha + 1$
    $\pi(s_j) = \alpha$
    for $k$ in $[j+1..i_{p_i}]$
     if $\pi(s_k) \neq 0$
      $(\mathcal{S}_1, \mathcal{S}_2) = \tau - sets(C(s_j), C(s_k))$
      if $(\mathcal{S}_1, \mathcal{S}_2) \neq (\emptyset, \emptyset)$
       $\pi$-prolongation($\mathcal{S}_1$,$\mathcal{S}_2$,$\alpha$)
```

In Figure 16 are displayed a *Bacillus subtilis W D26185* and a *Listeria grayi X92948M*. In this example, one can visually detect common patterns to both structures and one of the biggest (dash boxes).

In Figure 17, one can see the result of the rotation. One can compare to Figure 16 to perceive the help that automatic placement procures.

5 Conclusion

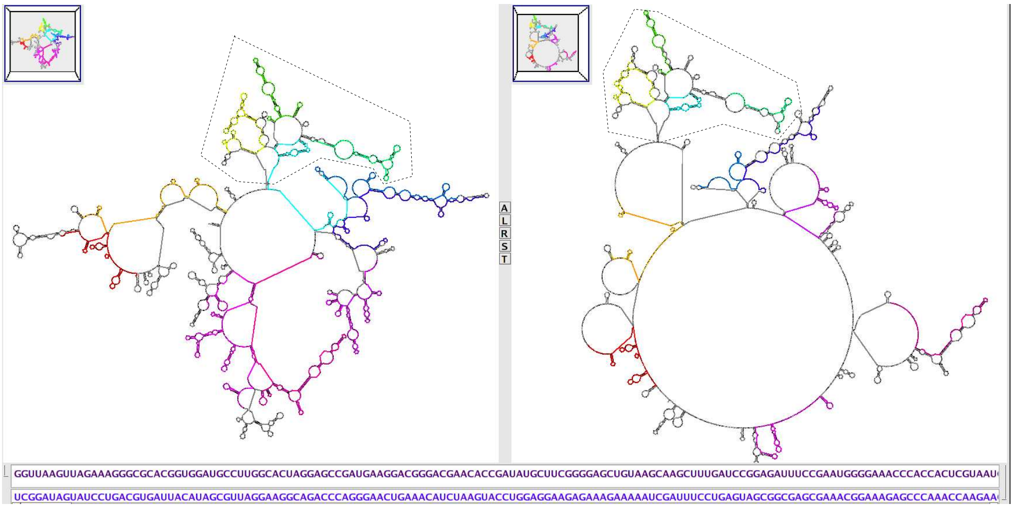
In order to check the efficiency, we have drawn about 1100 RNAs [5]. They were presented at the meeting of the ARENA french project [1] that involved biologists and bioinformatics french researchers focussed on RNA studies. About 40 persons were there. The community was split in two parts. Half agreed with the drawing because of its stability under small changes in the secondary structure. Half were less convinced because of their habits. The main criticism invoked was that on circles the edges between the nucleotids do not all have the same length as it is usually done by other interactive programs. In a certain sense, the reason of this problem is intrinsic to our algorithm. Usually, biologists



GGUUAAGUUAGAAAGGGCGCACGGUGGAUGCCUUGGCACUAGGAGCCGAUGAAGGACGGGACGAACACCGAUAUGCUUCGGGGAGCUGUAAGCAAGCUUUGAUCCGGAGAUUUCCGAAUGGGGAAACCCACCACUCGUAAU

UCGGAUAGUAUCCUGACGUGAUUACAUAGCGUUAGGAAGGCAGACCCAGGGAACUGAAACAUCUAAGUACCUGGAGGAAGAGAAAGAAAAAUCGAUUUCCUGAGUAGCGGCGAGCGAAACGGAAAGAGCCCAAACCAAGAA

Figure 16: Two RNAs before rotation.



GGUUAAGUUAGAAAGGGCGCACGGUGGAUGCCUUGGCACUAGGAGCCGAUGAAGGACGGGACGAACACCGAUAUGCUUCGGGGAGCUGUAAGCAAGCUUUGAUCCGGAGAUUUCCGAAUGGGGAAACCCACCACUCGUAAU

UCGGAUAGUAUCCUGACGUGAUUACAUAGCGUUAGGAAGGCAGACCCAGGGAACUGAAACAUCUAAGUACCUGGAGGAAGAGAAAGAAAAAUCGAUUUCCUGAGUAGCGGCGAGCGAAACGGAAAGAGCCCAAACCAAGAA

Figure 17: Two RNAs after rotation.

deform the circles manually in order to get such a result. The deformations are progressively from ellipse to rectangle. Our algorithm uses a method for packing parts of the RNA. Mixing forms in such a process is extremely difficult. They had thought that their habits were primordial but anyway looking to our drawing they can recognize the RNA in a blind process (that is they did not know the title of the RNA).

In parallel, we have developed a software system called ARNA [13] which includes the classical alignment method based on Levenshtein distance. Then, we have done a first user experiment with some biologists involved in research on RNA. The next step will be to compare the motifs that can be visually extracted on some well-known RNA pairs in order to check the exact efficiency of the presentation. This work is continuing with the "Institut Europen de Chimie et Biologie" in Bordeaux.

6 Acknowledgment

We thanks Alain Denise, Gérald Gainant and Nicolas Parysey for their help in setting up the software ARNA and Robert Strandh for improving the English of this paper.

References

[1] AReNa: Groupe de travail pluridisciplinaire sur la structure et la fonction des ARN. LRI.

[2] T. Asano, H. Imai, and A. Mukaiyama. Finding a maximum weight independent set of a circle graph. *IEICE Transactions*, E74(4):681–683, 1991.

[3] D. Auber, M. Delest, J. Domenger, P. Ferraro, and R. Strandh. EVAT: Environment for visualization and analysis of trees. In *IEEE Symposition on Information Visualisation Contest*, volume www.cs.umd.edu/hcil/iv03contest/, pages 124–126, 2003.

[4] D. Auber, M. Delest, J. Fédou, J. Domenger, and P. Duchon. New Strahler numbers for rooted plane trees. In M. Drmota, P. Flajolet, D. Gardy, and B. Gittenberger, editors, *Third Colloquium on Mathematics and Computer Science, Algorithms, Trees, Combinatorics and Probabilities*, Trends in Mathematics, pages 203–215. Vienna University of Technology, Birkhauser, 2004.

[5] D. Auber and L. Jézéquel. Automatic drawings of secondary structure of RNA. Technical Report RR-140406, LaBRI, 2006.

[6] J. Brown. The ribonuclease P database. *Nucleic Acids Research*, 27(314), 1999.

[7] J. Carriere and R. Kazman. Interacting with huge hierarchies: Beyond cone trees. In N. Gershon and S. Eick, editors, *IEEE Symposium on Information Visualization*, pages 74–78. IEEE Press, 1995.

[8] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

[9] S. Dulucq and L. Tichit. RNA secondary structure comparison: exact analysis of the Zhang-Shasha tree edit algorithm. *Theoretical Computer Science*, 306:471–484, 2003.

[10] S. Dulucq and H. Touzet. Analysis of tree edit distance algorithms. In R. Baeza-Yates, E. Chvez, and M. Crochemore, editors, *14th Annual Symposium on Combinatorial Pattern Matching*, volume 2676 of *Lecture Notes in Computer Science*, pages 83–95. Springer-Verlag, 2003.

[11] A. P. Ershov. On programming of arithmetic operations. *Communication of the ACM*, 1(8):3–6, 1958.

[12] J. Fekete, C. Plaisant, and S. Tafresh. Information visualization benchmarks repository. http://www.cs.umd.edu/hcil/InfovisRepository/, 2003.

[13] G. Gainant and D. Auber. ARNA: Interactive comparison and alignment of RNA secondary structure. In M. Wards and T. Munzner, editors, *IEEE Information Visualization Symposium 2003*, pages 8–9, Austin, USA, 2003. IEEE Computer Society.

[14] S. Grivet, D. Auber, J. Domenger, and G. Melançon. Bubble tree drawing algorithm. In K. Wojciechowski, editor, *Computer Vision and Graphics*, page to appear. Kluwer, 2004.

[15] A. Gupta and N. Nishimura. Finding largest subtrees and smallest supertrees. *Algorithmica*, 21(2):183–210, 1998.

[16] C. Gutwenger and P. Mutzel. Planar polyline drawings with good angular resolution. In S. Whitesides, editor, *6th Symp. Graph Drawing*, Lecture Notes in Computer Science, 1547, pages 167–182. Springer-Verlag, 1998.

[17] R. Hickson, C. Simon, and S. W. Perrey. The performance of several multiple-sequence alignment programs in relation to secondary-structure features for an rRNA sequence. *Mol. Biol. Evol.*, 17(4):530–539, 2000.

[18] M. Hø̈chsmann, T. Töller, R. Giegerich, and S. Kurtz. Local similarity in RNA secondary structures. In P. Blauvelt, editor, *IEEE Bioinformatics Conference*, pages 159–168. Standford University, 2003.

[19] R. Horton. Erosioned development of systems and their drainage basins, hydrophysical approach to quantitative morphomology. *Bulletin Geological Society of America*, 56:275–370, 1945.

[20] T. Jiang, G. Lin, B. Ma, and K. Zhang. A general edit distance between RNA structures. *J. Comput. Biol.*, 9:371–388, 2002.

[21] T. Jiang, L. Wang, and K. Zhang. Alignment of trees - an alternative to tree edit. *Theoret. Comput. Sci.*, 143:137–148, 1995.

[22] K. Kjer. Use of rRNA secondary structure in phylogenetic studies to identify homologous positions: an example of alignment and data presentation from the frogs. *Mol. Phylogenet. Evol.*, 4:314–330, 1995.

[23] P. N. Klein. Computing the edit-distance between unrooted ordered trees. In G. Bilardi, G. F. Italiano, A. Pietracaprina, and G. Pucci, editors, *Proceedings of the 6th Annual European Symposium*, volume 1461 of *Lecture Notes in Computer Science*, pages 91–102. Springer-Verlag, 1998.

[24] B. Ma, L. Wang, and K. Zhang. Computing similarity between RNA structures. *Theoret. Comput. Sci.*, 276:111–132, 2002.

[25] E. Reingold and J. Tilford. Tidier drawings of trees. *IEEE Transactions on Software Engineering*, 7(2):223–228, 1981.

[26] P. D. Rijk, J. Wuyts, and R. D. Wachter. RnaViz2: an improved representation of RNA secondary structure. *Bioinformatics*, 19:299–300, 2003.

[27] W. Schmitt and M. Waterman. Linear trees and RNA secondary structures. *Discrete Appl. Math.*, 51:317–323, 1994.

[28] A. Strahler. Hypsomic analysis of erosional topography. *Bulletin Geological Society of America*, 63:1117–1142, 1952.

[29] M. Vauchaussade and X. Viennot. Enumeration of RNA secondary structures by complexity. In Springer-Verlag, editor, *Mathematics in Medecine and Biology*, volume 57 of *Lecture Notes in Biomathematics*, pages 360–365, 1985.

[30] J. T. L. Wang, K. Zhang, and C. Chang. Identifying approximately common substructures in trees based on a restricted edit distance. *Inform. Sci.*, 121:367–386, 1999.

[31] M. R. Y. Dinitz, A. Itai. On an algorithm of Zemlyachenko for subtree isomorphism. *Information Processing Letters*, 703:141–146, 1999.

[32] H. Yang, F. Jossinet, N. Leontis, L. Chen, J. Westbrook, H. Berman, and E. Westhof. Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Res.*, 31:3450–3460, 2003.

[33] V. Zemlyachenko. Determining tree isomorphism. *Seminar on Combinatorial Mathematics*, pages 54–60, 1971.

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