Efficient drawing and comparison of RNA secondary structure

Category: Research

Abstract

Studying of RNA structures enables to compare evolution between different organisms. One of the main tasks is to compare them in order to build a phylogenetic tree. In this paper we propose a new layout algorithm that draws the secondary structure automatically according to the aesthetic criteria of the biologist. Such layout insures that two equivalent structures (or sub-structures) are drawn in the same way. To speed up visual comparison, we propose a fast heuristic for determining quasi-isomorphic sub-structures. This heuristic enables us to obtain a drawing where similar structures have the same position and the same orientation. Thus, it reduces the cost of the cognitive task of pattern recognition. We argument our results with the experimentation done in a software that integrates all these features.

CR Categories: I.3.3 [Picture/Image Generation]: Viewing algorithms—; I.3.6 [Methodology and Techniques]: Interaction techniques—; J.3 [Life and Medical Sciences]: Biology and genetics—;

Keywords: visualisation, combinatorics, bioinformatic, graph drawing

1 Introduction

Ribonucleic Acid (RNA) is an important molecule which performs a wide range of functions in biological system. 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 cell).

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 virus such as HIV and therefore regulates the functions of such virus.

A RNA is characterized by its base sequence and higher order structural constraints. This is particularly true of non-coding RNAs such as rRNAs and tRNAs and other functional RNAs, such as RNase P. All of these RNAs are characterized by short- and long- range base pair interactions that organize the molecules into domains and provide a framework for functional interactions.

A RNA can be considered at three levels of details :

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

In a sense, a RNA can be viewed as multi-scale molecule in the information visualisation context.

Secondary and tertiary structural features of RNAs are important in the molecular mechanism involving their functions. The presumption, of course, assumes that a preserved function corresponds to a preserved molecular confirmation and, therefore, a preserved secondary and tertiary structure. Therefore the ability to compare RNA structures is useful.

In the RNA world, RNA structure is often much more highly conserved than sequence, especially for structured RNAs, during evolution. So, knowledge of RNA secondary structure is becoming increasingly important in molecular phylogenetic studies, particularly in assisting accurate sequence alignment [Kjer 1995]. Automatic alignment methods that use only primary sequences may misalign RNA sequences [Hickson et al. 2000] and alignments that take secondary structure into consideration can generate improved phylogenetic trees [Kjer 1995].

Thus, consequently, the alignment between two RNAs must be interactive and simultaneous on the primary and secondary structures.

The figure 1 shows an example of RNA drawing coming from a biologic data bases [Brown 1999]. 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 biologist habits : there is no crossing edges (planar graph), the structural patterns appear clearly. In the past ten years, interactive software allowing manual placements has appeared as [Rijk et al. 2003],[Hofacker et al. 1994]. None of them fully fits these biologist requirements and moreover the automatic drawing of an RNA was remaining an open problem .

Regarding the comparison of RNAs, all the actual softwares do not allow to plug in and combine methods for the comparison.

In this paper, we propose an algorithm that solves the RNA drawing respecting the biologist point of view. It is stable in the sense that small changes on an RNA primary structure will not drastically change the drawing. It is based on the fact that one can associate a tree to the RNA secondary structure [Vauchaussade and Viennot 1985](see Figure 2).

In order to present the biologist two RNAs with the same orientation on the screen, we use an heuristic that allows to identify quasi-isomorphic subtrees in a tree. This heuristic has been used successfully at the Infovis'03 contest[Auber et al. 2003] and is described here. It allows us to place the largest similar part of two RNAs in a similar position on a portion of screen (for example at a same relative position from the upper left corner).

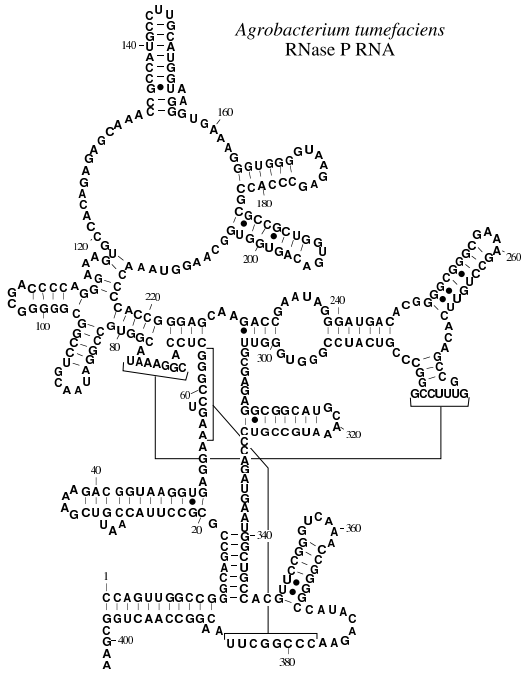

Figure 1: Drawing of a *RNA* coming from a biological data basis.

Finally, thanks to the Tulip framework [Auber 2003], we have set up a software that implements these algorithms. It allows to plug in alignments methods and combine them in order to construct interactively a solution. Interactively means that the user can simultaneously act on the primary and secondary structure, visually detect *good* alignments and construct step by step a complete alignment. This last interaction is a major one because alignment algorithms give usually many solutions for a part of two RNAs among which one must be chosen.

In what follows we first describe the biological background and list the major tasks to achieve. Then, we describe the drawing algorithm and the heuristic for preparing the alignment. At the final, a short experimentation of the software is proposed.

2 RNA background

A 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`). 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. In the base paired region, the RNA molecule adopts a helical structure as in DNA. Therefore, the linear RNA chain molecule, running from a 5? to a 3? end, folds on itself to form (2D) secondary and tertiary (3D) structures.

The RNA secondary structure refers to structural motifs such as stems, hairpins, bulges, interior loops and multi-branch loops. 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`.

The secondary structural motifs can interact to form 3D folds of biologically active (functional) RNA molecules [Moore 1999]. These interactions produce a complex hierarchical relationship between secondary and tertiary structures. According to the hierarchical view of RNA folding , the secondary structure is stabilized relatively fast and is followed by slow folding of the tertiary structure [Thirumalai et al. 2001].

Three-dimensional folding of RNAs is necessary for their in vivo functions. Consequently, the inference of secondary structures of RNA is a crucial step in the study of structure- function relationships and provides a major constraint in determining the three-dimensional models of the structures of these molecules [Gaspin and Westhof 1995]. The two main approaches of determining secondary structures from their primary sequence are comparative analysis and energy minimization [Zuker 2003; Mathews et al. 1999; Hofacker et al. 1994].

Because of its decomposition into stems (**S**), hairpins (**H**), bulges (**B**), interior loops (**I**), and multi-branch loops (**M**), a RNA secondary structure can be represented as an ordered tree in which each node is labeled by a letter **S**, **H**, **B**, **I** or **M** and the left to right order among siblings is significant [Jiang et al. 1995]. Since the RNA secondary structure appears as tree-like structure, there exist works comparing them using tree comparison [Le et al. 1989; Shapiro and Zhang 1990; Corpet and Michot 1995]. However these methods do not directly use base-paired nucleotides and unpaired nucleotides.

However, this tree representation which is rough (considering only the structural patterns) can be refined until an exact coding of the secondary structure is obtained [Vauchaussade and Viennot 1985; Schmitt and Waterman 1994] (see fig. 2). Many measures have been proposed for the similarity of two trees, e.g. tree edit distance, constrained edit distance and alignment of trees [Zhang and Shasha 1989; Jiang et al. 1995; Klein 1998; Dulucq and Tichit 2003; Dulucq and Touzet 2003].

Other related measures can be found in [Wang et al. 1999; Jiang et al. 2002; Ma et al. 2002]. Alignment of trees is a straightforward extension of sequence alignment that was proved to be different from tree edit distance [Jiang et al. 1995].

3 Definitions and notations

The *secondary structure* of a RNA molecule is the list of base pairs that occur in its three dimensional structure. A 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. A secondary structure on R is 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 lost in 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).

The last condition excludes *pseudo-knots*. These occur when two base pairs (i, j) and (k,l) satisfy $i < k < j < l$, that is when the two base pairs overlap. Pseudo-knots are often considered as belonging to tertiary structure. However, pseudo- knots are real and important structural features.

A 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 elements (stems, hairpins, loops, bulges, stems, ...),
- or nucleotides `A`, `C`, `G`, `U`, WatsonCrick pairs, wobble pairs, ...

In that way, considering only the structural patterns, this tree representation can be rough (situation (1)) or refined (situation (2)). Represented below are (see Figure 2):

- a RNA secondary structure on the left, the dots represent the nucleotides,
- a relatively rough tree representation of this structure on the center where the following labels are used: a stem node labeled **S**, a bulge node labeled **B** , an inside loop node labeled **I** , a multi-branch loop node labeled **M** and a an hairpin node labeled **H**,
- a coding of the same structure on the right: *coding* in the way that the tree (with an appropriate labeling of its nodes) makes it possible to come back to the secondary structure [Vauchaussade and Viennot 1985; Schmitt and Waterman 1994].

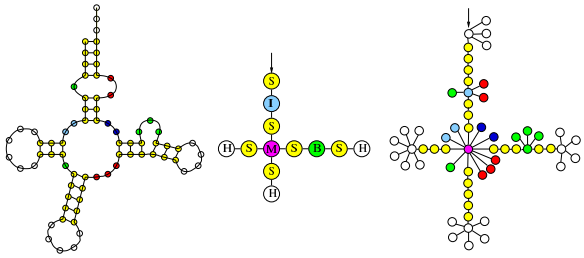

Figure 2: A secondary structure and its tree representations.

4 Visualization of RNA

A lot of methods for comparing RNA already exist (see section 2). However, no one is perfect and thus, human interaction is required in order to select a solution or to merge solutions from different algorithms. The purpose of this research is to design visualization tools that enable these operations. In order to experiment our methods, we have designed and developed a complete software called *Arna* that includes all the features presented in the following. The result of each alignment methods for RNA comparison gives the same kind of results. Thus, changing the comparison algorithm will not change the visualization task. Therefore, in our experimentation we have chosen to use only one algorithm based on the Levenshtein local distance [Levenshtein 1966]. This algorithm first builds a matrix of distance between the letters of the two primary sequences and then looks for paths in this matrix. Let $S_1 = [s_1, \cdots, s_n]$ and $S_2 = [s_1, \cdots, s_k]$ be two RNA words. The result of the algorithm

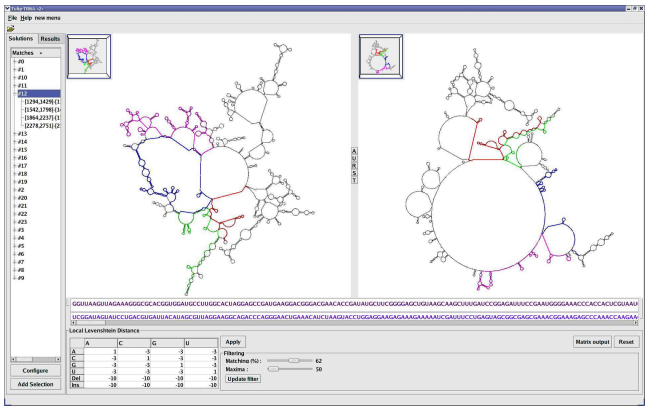

Figure 3: Arna software without alignment

is a list of list of position pairs in the form $((b_1,e_1),(b_2,e_2))$ indicating that $S'_1 = [s_{b_1}, \cdots, s_{e_1}]$ could match on $S'_2 = [s_{b_2}, \cdots, s_{e_2}]$. The size of the computed list depends on the initialisation of the method. It is usually high (40 is a mean).

4.1 Rendering and alignment

RNA structure is often much more highly conserved, thus for an efficient comparison, we must allow the user to compare visually the RNA structure. In that case, one of the most important requirement of the system is that isomorphic structure or sub-structure are drawn in the same way. Thus, if we show side by side two secondary structures one can detect easily similarities. On figure 3 are displayed a *Bacillus subtilis W D26185* and a *Listeria grayi X92948M*. On this example, one can visually detect the common patterns to both structures. The algorithm we have set-up in order to enable this visual comparison is described in the section 5.

One important requirement to facilitate visual comparison of structure is to detect automatically parts of the structure that have the same shape and to place it at the same position and in the same orientation in the final drawing. This is useful for the user because it gives him reference marks when he is comparing the structures. In the figure 4 one can see the result of the alignment. One can compare to the figure 3 to perceive the help that automatic alignment procures. In this example, the two drawings have been rotated in order to give the same orientation to the two biggest parts identified by our heuristic (see section 6).

4.2 Interaction and navigation

One can divide the navigation task in two different sets. The first one must allow the user to focus on a part of the structure and the second one enables to navigate through the solution space of the comparison algorithm.

We solve the first navigation task by allowing the user to zoom (zoom box, wheel mouse) in a part of the drawing and by providing an overview of the entire structure. One can notice in the figure 3 that in order to keep the context, we connect the box to the corner of the overview with lines. This simple trick enables to know quickly where we are in the overview even when the zoom factor is high.

For the second task we first filter the algorithm solutions (list of pairs) into groups that do not overlap. Like this we can display a group without occlusion. For the displaying of the solution, we map

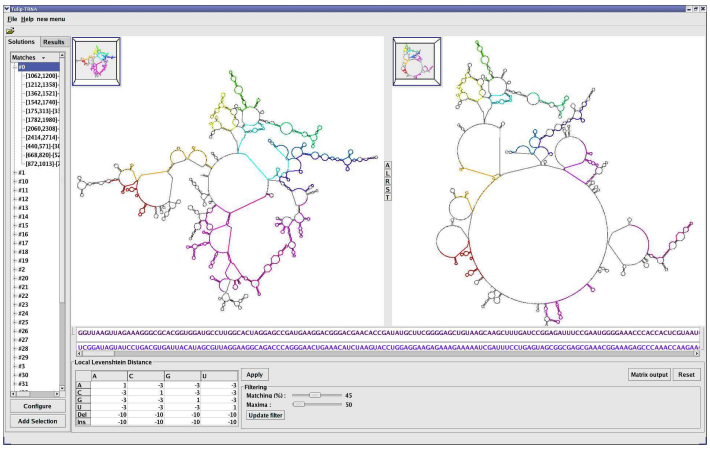

Figure 4: Arna software with alignment

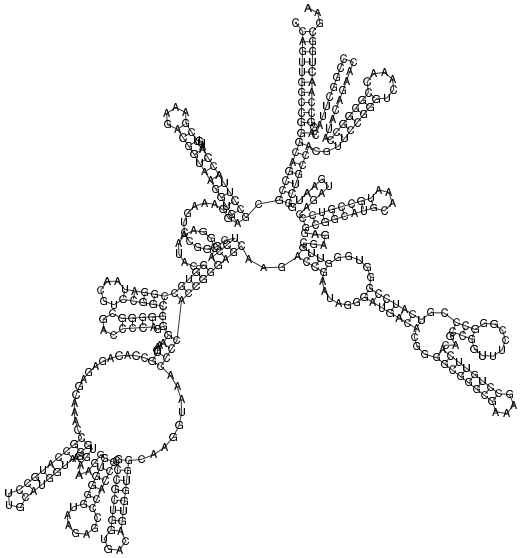

Figure 5: Drawing the Agrobacterium tumefaciens with Vienna RNA software.

each pair to a different color in the view. Choosing the set of colors is often a problem in visualization, in our experimentation we have noticed that groups are quite small (less than ten elements) so it stills possible to detect visually the different solutions. Furthermore, the solution indicates position in the sequence and we know that if the positions are near they should be near in the drawing. Thus, one can choose the color to be as different as possible between two solutions that are near in the sequence. One can see on the figure 4 the displaying of a group of solutions.

4.3 Biologist aesthetic criteria

An important requirements in a visualization system is to use the knowledge of users in order to speed up the learning of the software and not to have conflicts 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. The figure 1 shows a hand drawing of a secondary structure made by a biologist. These criteria are equivalent to maximize 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 [Battista et al. 1999]. Classical planar drawing algorithms such as the one proposed in [Gutwenger and Mutzel 1998] do not correspond to the intuitive representation of biologist.

Other representations can be found in RNA visualization softwares [Rijk et al. 2003; Hofacker et al. 1994], however, even if these representations are closer to the biologist's requirements, some important criteria such as the size of the drawing and the number of crossings are not properly taken into account. The figure 5 shows the result of the layout proposed by the software of the Vienna team. On this drawing one can see that there is crossings and that loops are not always drawn on a circle.

5 Drawing RNA secondary structure

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 $j \neq i+1$. The figure 2 shows a sequence of nucleotides and some link. We call P_{link} (primary structure) the links belonging to the sequence, S_{link} (secondary structure) the links that form stems and T_{link} (tertiary structure) the links that create crossings. If one looks to the structure of *RNA*, one can see that the set of nucleotides and the union of P_{link} and S_{link} is an outer-planar biconnected graph. One of the nice properties of such a graph is that one can extract a tree as shown in the 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 the following, we briefly detail the steps of the algorithm.

5.1 Outer-polarization

In the first step, we remove the edges (T_{link}) in order to get an outer-planar graph. Knowing that the P_{link} set forms a Hamiltonian path in the graph we know that this path must be the outer-face of the outer-planar graph. Thus, the problem is transformed to the problem of finding a set of edges that enables to draw the graph without crossing. In the figure 6 one can see the drawing of a graph on one page (each edge is drawn up to the sequence). Finding this minimum set consists in 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 the sets S_1 *and* S_2 such that for all vertices $u \in S_1$, it exists a vertex $v \in S_2$ such that u cross v. We obtain T_{link} by choosing the set with minimum size. 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 [Asano et al. 1991]. During our experimentation with data coming from biological data base, the conflict graph was always bipartite.

The figure 6 shows a conflict graph. For this graph we remove the edges 1 and 4 ($T_{link} = \{1,4\}$). One can see in red on the 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.

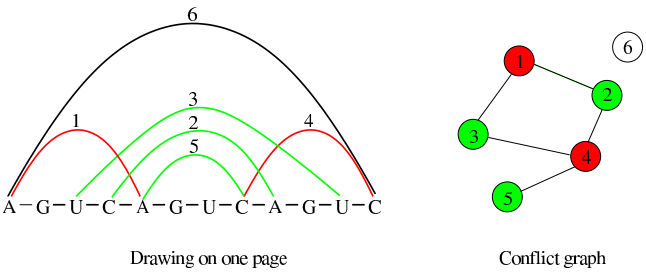

Figure 6: One page drawing and its conflict graph.

After removing the T_{link} edges one can compute the tree representation shown on the right of the figure 2. In the following, we explain how to draw this tree in order to obtain a layout that respects biologist aesthetic criteria.

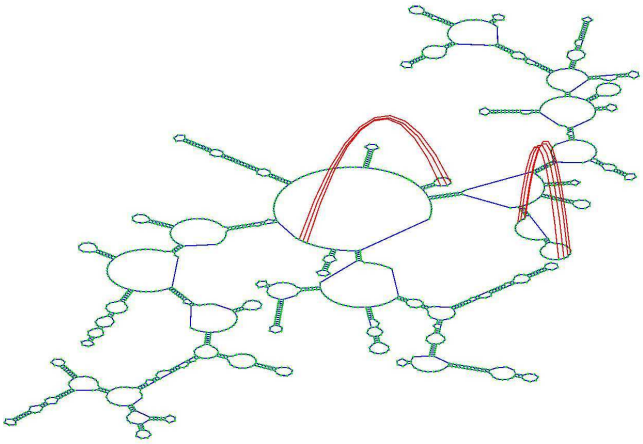

Figure 7: Visualization of pseudo-knots.

5.2 Tree drawing

The tree drawing algorithm we have set-up is a recursive one. At each step we first layout all the sub-trees induced by children of a node. Then, we place all these sub-trees drawing on a circle. Literature about similar drawing algorithm can be found in [Carriere and Kazman 1995; Grivet et al. 2004]. In our approach, instead of using a circle hull, we use a packing method such as the one used by [Reingold and Tilford 1981]. However, in the case of circular drawing, the packing is more complex than in the hierarchical case. The main problem is to determine the radius of the circle on which one places the sub-trees drawing. 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 do. Then, by computing the bounding box of this drawing we obtain the 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. Figure 8 shows the result packing of the root of the tree and the result on the final drawing.

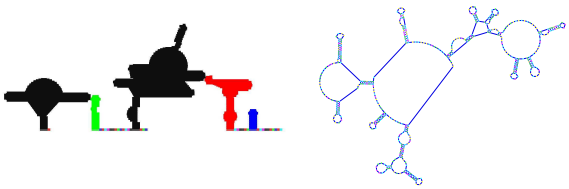

Figure 8: Simple packing.

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. We apply this method until a stable circumference is obtained. Figure 9 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 to obtain drawings of the biggest RNA structures in a few seconds.

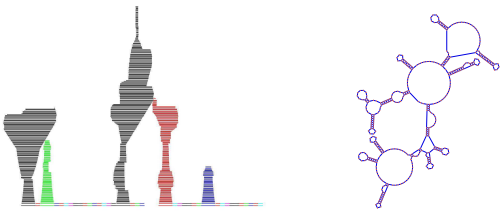

Figure 9: Polar packing.

To prevent overlapping in the final drawing one must force sub-trees drawing to be drawn outside of the circle. This restriction introduces a lot of long edges in the final drawing but is necessary to have a good angular resolution in the drawing. On the figure 10 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 sub-tree to be on the half circle. To do this and to take into account the particular case of sub-trees composed of one node. We add in the packing algorithm to sub-trees (or shape) at the left and at the right. On the figure 10 one can see these two shapes in black at the left and at right of the drawing. Even if the half circle seems to be the best solution according to our experimentation one can use this as a parameter to choose the trade off between the edge length and the angular resolution.

5.3 Final drawing

In the figure 11 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 to 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

Figure 10: (left) Shape insertion, (right) Bad edge length.

$father(\eta)$. This operation is straightforward and is done by using a cross product. The main problem is to determine the distance between the node and η. If one do not take care about it the node could overlap with other nodes of the drawing. We solve this problem by setting the size of η three times bigger (one node, a space and one node) in the tree drawing algorithm. Like this we are sure that if one places a node at distance $\frac{3}{2}$ from η; no overlapping can appear. Figure 12 shows the result of our algorithm on the graph of the figure 1.

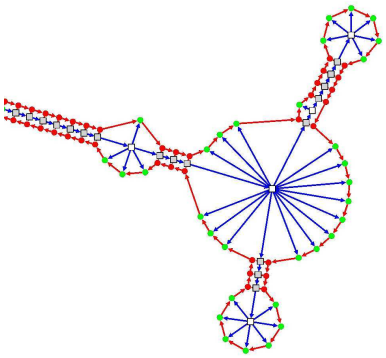

Figure 11: Sets of nodes.

6 RNA Alignment

In this section, we use a combination of metrics in order to predict common parts in several RNA's. We have shown in section 2 that the secondary structure of an RNA is associated to a tree. At the Infovis'03 contest on pairwise comparison of trees, an assigned task was to find parts of trees that have moved. First note that "finding isomorphic subtrees in a tree" or "common subtrees in two trees" are the same task. In the last case, one just needs to construct a tree with a vertex (its root) which has two subtrees that are the trees to be compared. Thus "find similar parts in two RNAs" is equivalent to "find similar subtrees in a tree". Work has already been done based on arity by Zemlyachenko [Zemlyachenko 1971] and more recently by Dinitz and *alt.* [Y. Dinitz 1999]. Their algorithm gives a partition of subtrees into isomorphism equivalence classes. It is proved to be linear. However, it only detects isomorphism and does not provide a measure of similarity for subtrees. In order to give a response to the Infovis'03 task, we have designed an heuristic [Auber et al. 2003] that can suggest by colors meaning similar parts in a tree. Thus, we are able using the graph drawing algorithm of section 5 to display automatically two RNAs and (using our heuristic) to color them in order to suggest the user which parts of the RNAs are similar. The last task to do is to present the final images of the two RNAs such that the user can easily identify the common parts. We decide to

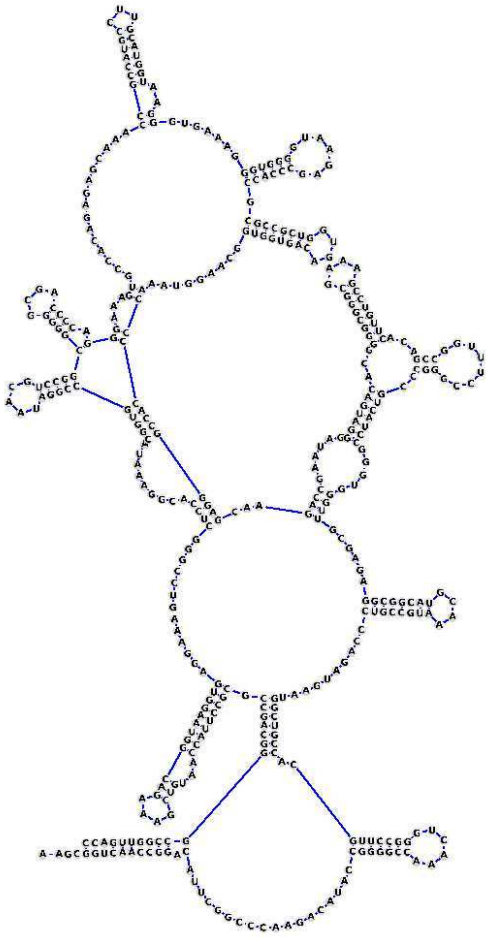

Figure 12: Drawing Agrobacterium tumefaciens with our algorithm

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. On figure 4, one can see two RNAs drawn by this algorithm. Below, we briefly describe the heuristic which is not yet published. Let s a vertex. It is based on the three metrics :

- the arity of the vertex denoted by $\alpha(s)$,
- the number of nodes of the subtree with root s denoted by $\nu(s)$,
- the hight of the subtree with root s denoted by $\eta(s)$

and an other one the so-called Strahler numbers. We now 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 [Horton 1945; Strahler]. A generalization on planar trees has been made [Auber et al. 2004] using the nice interpretation by Ershov [Ershov 1958] who has 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. Each internal vertex s have $k+1$ 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 :

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

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

$$(\overline{\alpha}(s)-\overline{\alpha}(s'))^2+(\overline{\nu}(s)-\overline{\nu}(s'))^2+(\overline{\eta}(s)-\overline{\eta}(s'))^2 + (\overline{\sigma}(s)-\overline{\sigma}(s'))^2 \le \varepsilon.$$

The second one aggregates the classes of vertices that have equivalent subtrees. Let $(K_i)_{0\le i\le k}$ the set of classes. Let $(s_i)_{0\le i\le n}$ the sorted list of vertices in decreasing order according to their Strahler value. For a vertex s, we denote by $C(s)$ the set of its children. Let s and s' in a given K_t, we consider the distribution of the indices over $C(s)$ and $C(s')$. For each indice, if the number of children are not too far away each other then we aggregate these children to the class K_t. The algorithm is a recursive one that means that the same process is applied to the grandchildren until no vertex can be aggregated.

7 Experimentation and Conclusion

In order to check our choices, we have done a first user experiment with some biologists involved in research on RNA. They have to

1. score the drawings small RNA as in Figure 1
2. score the drawing of large RNA as in Figure 4,
3. validate the process when one is going through the construction of an alignment.

The first task leads to a score four over five. The main criticism invoked is that on circles the stems between bases have not all the same length as it is usually done on other softwares. In a certain sense, the reason of this problem is intrinsic to our algorithm. Usually, biologists deformed the circles manually in order to get such a result. The deformations are progressively from ellipse to rectangle. Our algorithm use a method for packing parts of the RNA. Mixing forms in such a process is extremely difficult. The second task leads to a score of five over five because no public software exists giving a planar drawing for large (about 3000 bases) RNA. Validating the process of construction was more difficult because we have implemented only one method while biologist are in use with several one. Thus, they were not able to score the software itself. Anyway, they said that they were convinced by the efficiency of such methods for exploring RNAs alignments. We plan to enlarge our software with several methods for instance the Zhang one and tree alignment that seem major to us. Other research will be to improve the drawing algorithm using other shapes (ellipses). A track is surely in enlarging the set of interactive tools provided to the user.

8 Acknowledgment

We thanks Alain Denise, Yan Dupérier, Gérald Gainant and Nicolas Parysey.

References

ASANO, T., IMAI, H., AND MUKAIYAMA, A. 1991. Finding a maximum weight independent set of a circle graph. *IEICE Transactions E74*, 4, 681–683.

AUBER, D., DELEST, M., DOMENGER, J., FERRARO, P., AND STRANDH, R. 2003. EVAT : Environment for visualization and analysis of trees. In *IEEE Symposition on Information Visualisation Contest*, vol. www.cs.umd.edu/hcil/iv03contest/, 124–126.

AUBER, D., DELEST, M., FÉDOU, J., DOMENGER, J., AND DUCHON, P. 2004. New strahler numbers for rooted plane trees. In *Third Colloquium on Mathematics and Computer Science, Algorithms, Trees, Combinatorics and Probabilities*, Birkhuser, M. Drmota, Ed., Vienna University of Technology, To appear.

AUBER, D. 2002. Using strahler numbers for real time visual exploration of huge graphs. In *International Conference on Computer Vision and Graphics*, vol. 1-3 of *Journal of WSCG*, 56–69.

AUBER, D. 2003. Tulip : A huge graph visualisation framework. In *Graph Drawing Softwares*, P. Mutzel and M. Jünger, Eds., Mathematics and Visualization. Springer-Verlag, 105–126.

BATTISTA, G., EADES, P., TAMASSIA, R., TOLLIS, I., AND TOLLIS, G. 1999. *Graph Drawing : Algorithms for the Visualization of Graphs*. Prentice-Hall.

BROWN, J. W. 1999. The ribonuclease p database. *Nucleic Acids Research 27*, 314.

CARRIERE, J., AND KAZMAN, R. 1995. Interacting with huge hierarchies: Beyond cone trees. In *IEEE Symposium on Information Visualization*, Atlanta, Georgia Institute for Electrical and Electronics Engineers, G. and S. Eick, Eds., 74–78.

CORPET, F., AND MICHOT, B. 1995. Rnalign program: alignment of rna sequences using both primary and secondary structures. *Comput. Appl. Biosci. 10*, 389–399.

DULUCQ, S., AND TICHIT, L. 2003. Rna secondary structur comparison: exact analysis of the zhang-shasha tree edit algorithm. *Theoretical Computer Science 306*, 471–484.

DULUCQ, S., AND TOUZET, H. 2003. Analysis of tree edit distance algorithms. In *14th Annual Symposium on Combinatorial Pattern Matching*, Springer-Verlag, Ed., vol. 2676 of *Lecture Notes in Computer Science*, 83–95.

ERSHOV, A. 1958. On programming of arithmetic operations. *Communication of the A.C.M 1*, 8, 3–6.

GASPIN, C., AND WESTHOF, E. 1995. An interactive framework for rna secondary structure prediction with a dynamical treatment of constraints. *J. Mol Biol. 254*, 2, 163–174.

GRIVET, S., AUBER, D., DOMENGER, J., AND MELANÇON, G. 2004. Bubble tree drawing algorithm. Tech. Rep. RR-1323-04, LaBRI.

GUTWENGER, C., AND MUTZEL, P. 1998. Planar polyline drawings with good angular resolution. In *6th Symp. Graph Drawing*, Springer-Verlag, S. Whitesides, Ed., Lecture Notes in Computer Science, 1547, 167–182.

HICKSON, R., SIMON, C., , AND PERREY, S. W. 2000. 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.

HOFACKER, I., FONTANA, W., STADLER, P., BONHOEFFER, L., TACKER, M., , AND SCHUSTER, P. 1994. Fast folding and comparison of rna secondary structures (the vienna rna package). *Monatsheffe fr Chemie 125*, 167-188.

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

JIANG, T., WANG, L., AND ZHANG, K. 1995. Alignment of trees - an alternative to tree edit. *Theoret. Comput. Sci. 143*, 137–148.

JIANG, T., LIN, G., MA, B., AND ZHANG, K. 2002. A general edit distance between rna structures. *J. Comput. Biol. 9*, 371–388.

KJER, K. 1995. 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.

KLEIN, P. 1998. Computing the edit-distance between unrooted ordered trees. In *Proc. Annual European Symposium on Algorithms*, Springer-Verlag, Ed., vol. 1461 of *Lect. Notes in Computer Science*, 91–102.

LE, S., NUSSINOV, R., AND MAIZEL, J. 1989. Tree graphs of rna secondary structures and their comparisons. *Comput. Biomed. Res. 22*, 461–473.

LEVENSHTEIN, V. I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl. 6*, 707–710.

MA, B., WANG, L., AND ZHANG, K. 2002. Computing similarity between rna structures. *Theoret. Comput. Sci. 276*, 111–132.

MATHEWS, D., SABINA, J., ZUKER, M., AND TURNER, D. 1999. Expanded sequence dependence of thermodynamic parameters improves prediction of rna secondary structure. *J. Mol. Bio. 288*, 911–940.

MOORE, P. 1999. Structural motifs in rna. *Annu. Rev. Biochem. 68*, 287–300.

REINGOLD, E., AND TILFORD, J. 1981. Tidier drawings of trees. *IEEE Transactions on Software Engineering 7*, 2, 223–228.

RIJK, P. D., WUYTS, J., AND WACHTER, R. D. 2003. Rnaviz2: an improved representation of rna secondary structure. *Bioinformatics 19*, 299–300.

SCHMITT, W., AND WATERMAN, M. 1994. Linear trees and rna secondary structures. *Discrete Appl. Math. 51*, 317–323.

SHAPIRO, B., AND ZHANG, K. 1990. Comparing multiple rna secondary structures using tree comparisons. *Comput. Appl. Biosci. 6*, 309–318.

STRAHLER, A. Hypsomic analysis of erosional topography. *Bulletin Geological Society of America 63*, 1117–1142.

THIRUMALAI, D., LEE, N., WOODSON, S., AND KLIMOV, D. 2001. Early events in rna folding. *Annu. Rev. Phys. Chem. 52*, 751–762.

VAUCHAUSSADE, M., AND VIENNOT, X. 1985. Enumeration of rna secondary structures by complexity. Springer-Verlag, Ed., vol. 57 of *Lecture Notes in Biomathematics*, 360–365.

WANG, J., ZHANG, K., , AND CHANG, C. 1999. Identifying approximately common substructures in trees based on a restricted edit distance. *Inform. Sci. 121*, 367–386.

WATERMAN, M. 1978. Secondary structure of singled stranded nucleic acids. *Adv. in Maths Suppl. Studies I*, 167–212.

Y. DINITZ, A. ITAI, M. R. 1999. On an algorithm of Zemlyachenko for subtree isomorphism. *Information Processing Letters 703*, 141–146.

ZEMLYACHENKO, V. 1971. Determining tree isomorphism. *Seminar on Combinatorial Mathematics*, 54–60.

ZHANG, K., AND SHASHA, D. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput. 18*, 1245–1262.

ZUKER, M. 1989. On finding all sub-optimal foldings of an rna molecule. *Science 244*, 48–52.

ZUKER, M. 2003. Mfold web server for nucleic acid folding and hybridisation prediction. *Nucleic Acids Res. 31*, 13, 1–10.