RESEARCH

Reconstruction of ancestral RNA sequences under multiple structural constraints

Olivier Tremblay-Savard[1,2], Vladimir Reinharz[1] and Jérôme Waldispühl.[1*]

*Correspondence:
jeromew@cs.mcgill.ca
[1]School of Computer Science, McGill University, H3A 0E9 Montreal, Canada
Full list of author information is available at the end of the article

Abstract

Motivation: Secondary structures form the scaffold of multiple sequence alignment of non-coding RNA (ncRNA) families. An accurate reconstruction of ancestral ncRNAs must use this structural signal. However, the inference of ancestors of a single ncRNA family with a single consensus structure may bias the results towards sequences with high affinity to this structure, which are far from the true ancestors.

Results: In this paper, we introduce `achARNement`, a maximum parsimony approach that, given two alignments of homologous ncRNA families with consensus secondary structures and a phylogenetic tree, *simultaneously* calculates ancestral RNA sequences for these two families. We test our methodology on simulated data sets, and show that `achARNement` outperforms classical maximum parsimony approaches in terms of accuracy, but also reduces by several orders of magnitude the number of candidate sequences. To conclude this study, we apply our algorithms on the Glm clan and the FinP-traJ clan from the Rfam database. Our results show that our methods reconstruct small sets of high-quality candidate ancestors with better agreement to the two target structures than with classical approaches

Availability: http://csb.cs.mcgill.ca/acharnement

Keywords: RNA; secondary structure; ancestor reconstruction; evolution; phylogeny; algorithm

Introduction

With the development of sequencing technologies emerged the need to elucidate the relationship between sequences from various organisms. The reconstruction of ancestral sequences, which aims to reveal the chain of events that led to the diversity of sequences observed today, became naturally one of the core challenges in this field of research. Since the first attempts to rigorously solve this problem [1], the methods and quality of the data have considerably improved, to the point where the reconstruction of ancient genomes is now feasible [2, 3, 4].

For a long time, most of the attention has been given to the reconstruction of ancient protein and DNA sequences, while RNA molecules remained relatively overlooked. Nonetheless, in the last 20 years, the discovery of the breadth of catalytic and regulatory functions carried by RNA molecules revived our interest for the RNA world hypothesis [5], and resulted in increasing efforts toward a better understanding of the intricate nature of mutational patterns in RNAs [6, 7, 8, 9, 10, 11].

The reconstruction of non-coding RNA (ncRNA) sequences is particularly challenging. Indeed, ncRNA functions are typically carried out by specific molecular structures, and consequently sequences are generally less conserved than structures [12]. This implies that dedicated frameworks must be developed to capture this structural information.

RNA folding is hierarchical. Secondary structures form rapidly and act as a scaffold for the slower formation of tertiary structures [13]. It follows that the stability of secondary structures provides us a relatively accurate signature of the molecular function [14], and thus can be used to guide the reconstruction of ancestral ncRNA sequences.

To date, the most promising approach to infer ncRNA ancestors has been proposed in 2009 by D. Bradley and I. Holmes, who introduced an algorithm to calculate ancestral RNA secondary structures from an alignment [15], and use these structures to infer ancestral sequences using a maximum-likelihood approach on stochastic grammars [16]. Still, the time complexity of inferring ancestral structures can be prohibitive, and the specificity of the functional structure may not accommodate sufficiently large variations of this (secondary) structure to take advantage of this model.

Covariation models are powerful frameworks to model families of structured RNA sequences [17, 18, 19], allowing us to capture dependencies between distant sites. Nevertheless, we argue that the reconstruction of ancestral RNA sequences of a single ncRNA family with a single secondary structure using a covariation model can be hazardous. Indeed, current sequences are most likely uniformly distributed on the entire neutral network of the functional structure [20] (i.e. regions of the sequence landscape with a good affinity to the functional structure), and a strategy aiming to accommodate constraints within a single family will have a tendency to produce ancestors near the core of this network. This bias may result in ancestral sequences potentially far from the first ancestor who acquired the function (*i.e.* the structure). In other words, this first ancestor is likely to be a worse fit to the functional structure than sequences at the core of the neutral network. By contrast, in this paper we adopt a radically different approach. We propose here to solve this problem simultaneously for two ncRNA families that share a common ancestor (See Fig 1). This strategy enables us to make a better estimation of the location of the duplication event at the origin of the two families in the sequence landscape, hence to make a more accurate inference of the ancestors of each family.

Our approach is as follows. Given two alignments of homologous ncRNA families with consensus secondary structures and a phylogenetic tree, we design a maximum parsimony algorithm to *simultaneously* compute ancestral RNA sequences for both families. We test this methodology on simulated data sets, and compare the results to classical (structure-free) maximum parsimony approaches [21, 22], as well as to a customized maximum parsimony algorithm integrating the constraints of a single structure. Finally, we apply our techniques to the reconstruction of ancestral sequences of two Clans (Glm [23] and FinP-traJ [24]) from the Rfam database [25]. Clans are RNA families that "share a common ancestor but are too divergent to be reasonably aligned" [26], and thus illustrate well the signal we aim to capture. Our results on simulated data sets show that our strategy improves the accuracy of the reconstruction. On real data sets, our approach compares favorably to `PAML`, a state-of-the-art maximum likelihood method that considers one structure at a time, and customized versions of the Fitch and Sankoff algorithms. In particular, our data shows that our solutions have a better agreement to the two target structures than the sequences inferred with previous methods. Importantly, we achieve all these results with a significantly smaller set of candidate ancestors, which improves the interpretability of our data.

Our algorithms have been implemented in a software named `achARNement` and are freely available at http://csb.cs.mcgill.ca/acharnement.

Methods

Input data

For the algorithms presented in this paper, we assume that we have two non-coding RNA families that have been identified as a clan [26]. For each of the two ncRNA families, we have the consensus 2D structure it folds into. We also have a set of species that each possess one copy of both ncRNAs (one of each family), and a species tree T that represents the speciation history of the organisms considered. We have the sequences of the two ncRNAs for each of the studied species. Fig. 1 illustrates an example of a species tree.

Problem statement

Given the input data described in the previous subsection, the problem is to infer a most parsimonious set of ancestral sequences for each of the two ncRNAs at each ancestral node of the input species tree. Although this is a very classical problem in comparative genomics, our goal is to achieve that using a new evolutionary model that simultaneously considers sequence and 2D structure information, as described previously.

Evolutionary model

Our evolutionary model assumes that the two ncRNA families are the result of an ancient duplication of an ancestral ncRNA that was able to fold into two different structures. Following the duplication, a subfunctionalization process took place: a series of neutral mutations occurred and gave rise to both extant families that can only fold into one specific structure (see Fig. 1). Here, we assume that the ancestor of all studied species already possessed both ncRNAs, but that the duplication event occurred not too long before that (near the root of the species tree T representing the studied organisms). Only point mutations are allowed in our evolutionary model (no indels).

As mentioned earlier, ncRNA sequences are more constrained by their structure than their sequence during evolution. Since we have only access to the 2D structure of the two extant families (and not the ancestral 2D structure), our model considers both of these structures during the inference process. Near the root of the species tree, our model suggests that the sequences were still likely to be able to fold into both structures. However, as time passes, each ncRNA starts to specialize into only one structure and loses affinity to the other. We represent that gradual transition into our model using a gradient G which varies from 50% (near the root) to 100% (near the leaves). This gradient is going to be used in our algorithm to calculate the "weight" that each of the structures must have in the global score of the inferred ancestral sequences.

We developed two novel algorithms, implemented in a package called `achARNement` and freely available at http://csb.cs.mcgill.ca/acharnement.

Algorithms

We propose a new tool, `achARNement`, composed of two exact algorithms (`CalculateScores-1struct` and `CalculateScores-2structs`) based on the Fitch [21] and Sankoff [22] parsimony methods for the inference of ancestral sequences in a phylogeny (note that we are focusing on the inference of ancestral sequences, and not only on the calculation of parsimony scores). Our algorithms use a three-step approach (see Fig. 2): (i) a bottom-up step in which minimal costs for every possible nucleotide at every site are calculated, (ii) a middle step where we link the minimal cost matrices for both families at the root of the phylogeny, and (iii) a top-down step that enumerates all the optimal sequences based on the calculated costs. Our algorithms have the same running-time complexity than the Sankoff algorithm ($O(Nk)$, where N is the number of nodes and k is the sequence length); the only difference being a constant number of additional calculations that depends on the basepairs in the two structures.

For the substitutions, we use a cost matrix that has a different weight for transitions and transversions, since transversions normally occur less frequently than transitions (see Tab. 1(a)). In addition to the substitution cost, we also consider a basepair cost, as shown in Tab. 1(b). The basepair cost is 0 for the G-C basepairs and 0.001 for the A-U basepairs, that are not as strong as G-C basepairs. Compared to an A-U basepair, a G-U pair costs twice as much, while all the others are penalized three times as much. We have also experimented with a more complex scoring system for the basepairs, one that reflects the geometry and isostericity of the basepairs. We performed tests using the IsoDiscrepancy Index (IDI) table, as described in [27]. However, since this table represents a transition from the initial basepair to a mutated basepair, more calculations were required by our `CalculateScores-2structs` algorithm. The results obtained with the IDI table and our simpler table (Tab. 1(b)) were very similar (results not shown), but at the cost of a 4-fold increase in computation time. Consequently, we decided to use the simpler table.

The difference between the two algorithms we propose (`CalculateScores-1struct` and `CalculateScores-2structs`) resides in the first step (bottom-up), where we calculate the minimal costs for every possible nucleotide at every site. Let f be one of the two families and $\bar{f}$ be the other one. When calculating the costs for family f at the internal node a, algorithm `CalculateScores-1struct` considers only the structure associated with family f. On the other hand, algorithm `CalculateScores-2structs` considers both structures, but with a weight G that varies along the depth of the tree. For example, at the level of the leaves, for the family f, we consider 100% of the structure f and 0% of the structure $\bar{f}$. At the level of the root, we consider 50% of the structure of family f and 50% of the structure of family $\bar{f}$. We use a linear gradient to set the values of G on the different depths of the tree (from 50% to 100%).

For space reasons, the full description of the algorithms was placed in the supplementary materials. In the following paragraphs, an overview of the algorithms will be presented.

Bottom-up step

The first step of the algorithms consists of doing a post-order traversal of the species tree (as shown in Algorithm 1, supp. mat.), to calculate the most parsimonious costs for each possible nucleotide at every site. In the following paragraphs, we explain the differences between `CalculateScores-1struct` and `CalculateScores-2structs` in the calculation of those costs.

CalculateScores-1struct: Let a_i be the nucleotide at position i in the parent (ancestral) node, $\bar{a_i}$ be the nucleotide that is paired with a_i in the current structure, l_i (resp. r_i) be the nucleotide at position i in the left (resp. right) child, and $\bar{l_i}$ (resp. $\bar{r_i}$) be the nucleotide that is paired with l_i (resp. $\bar{r_i}$) in the left (resp. right) child in the current structure.

In the case that the position i is part of a basepair in the current structure, the cost of having a specific dinucleotide $a_i, \bar{a_i}$ is equal to:

$$\begin{aligned} &\min_{l_i \text{and} \bar{l_i} \in \{A,C,G,U\}} \{c(l_i) + s(l_i, a_i) + c(\bar{l_i}) + s(\bar{l_i}, \bar{a_i}) + bpc(a_i, \bar{a_i})\} + \\ &\min_{r_i \text{and} \bar{r_i} \in \{A,C,G,U\}} \{c(r_i) + s(r_i, a_i) + c(\bar{r_i}) + s(\bar{r_i}, \bar{a_i}) + bpc(a_i, \bar{a_i})\} \end{aligned} \tag{1}$$

where $c(x)$ is the previously calculated optimal cost of having the nucleotide x, $s(x, y)$ is the cost of substituting nucleotide x for y and $bpc(x, y)$ is the cost of having the basepair (x, y). In the other case where i is not part of a basepair, we simply calculate the substitution costs.

CalculateScores-2structs: As mentioned earlier, `CalculateScores-2structs` takes into account both structures, using a weight G. Calculating the costs on the left and right branches is a little bit different depending on if we are dealing with a paired position or an unpaired one. The general idea is that for each position i, we are going to measure the cost of the basepair formed with position $\bar{i}$ (if it exists) in the structure of the current family, and we are also going to consider the positions paired with both i and $\bar{i}$ in the other structure. Note that each position can be paired to two different positions in the two structures; we will focus on that case here, because if the basepairs are the same in both structures, then we do not need the gradient G and simply consider 100% of the basepair cost. Fig. 3 shows three examples.

The simpler case is when the position i is unpaired. Then, only the position paired with i in the other structure needs to be considered, if it is paired (see Fig. 3 a)). Since that position (e.g. position #6 in Fig. 3 a)) is not necessarily fixed, we consider an average basepair cost over all possible nucleotides at that position.

The more complex case is when position i is paired. In this case, we also have to check for the position paired with $\bar{i}$ in the other structure (see Fig. 3 b) and c)).

More precisely, using the same definitions as above for `CalculateScores-1struct`, and considering that position i is paired in both structures (and position $\bar{i}$ too), the cost of having a specific dinucleotide $a_i, \bar{a_i}$ is equal to:

$$G * Eq.(1) + (1 - G) * (\sum_{nc \in \{A,C,G,U\}} bpc(a_i, nc)/4 + \sum_{nc \in \{A,C,G,U\}} bpc(\bar{a_i}, nc)/4) \tag{2}$$

Note that it is possible to get cycles of "interdependent" positions when considering both structures. As you can observe in Fig. 3 c), positions 4 and 6 are paired together in fam1. In fam2, position 4 is paired with 3, which is paired with position 7 in fam1. Finally, position 6 is paired with position 8 in fam2. Thus all of those positions are "interdependent". To simplify the algorithm, instead of considering the complete cycles, we chose to stop at one "level", that is looking only at one paired position in the other structure for each position in the first structure.

Once the costs are calculated for every site at every node of the species tree, we can simply do the middle and top-down steps.

Middle and top-down steps

The top-down step is the part where we start from the root of the tree, we select the nucleotides of minimum cost at every position and construct the optimal sequences. Once all the optimal sequences are enumerated at an internal node of the tree, we go down in the tree and enumerate the optimal sequences that gave rise to them in the child nodes and so on. Algorithm 6 (supp. mat.) describes this process. Note that before starting to select the nucleotides at the root, we do a simple Fitch (algorithm not shown here) on both cost matrices of family 1 and 2. This middle step is necessary to make the link between the two families, *i.e* reconnect both matrices of optimal scores, and reconstruct the original ancestral sequences (before the duplication).

Generalizing to more than two families

This problem can easily be generalized to $F > 2$ structure families, as long as we maintain the same assumption that all ancestors represented in the tree possess one copy of each F number of ncRNAs. The only part of the algorithm that would change is the bottom-up step: it would be similar to `CalculateScores-2structs`, except that we would be considering the basepairs in all the F structures instead of just two. The gradient G would also be different: it would range from $\frac{100}{F}\%$ (near the root, where the same weight is given to all structures) to 100% (near the leaves).

Results

Simulated data generation

To evaluate our method, we generated in silico twenty different phylogenetic trees for three different pairs of secondary structures as follows. First, three secondary structures of size 100 were randomly designed such that the two first have a similar shape, and the last, a different one. Those structures are the following

```
0: ...(..((((((((.(.(((.(((((((.((.((.(((..(((....((....))..)))
                           ..))))).))..))))))).))).).))))))))..)...
1: ...........((((((.(((((((((((((((.((..((((..((........)).)))
                           )..)).))))))))))..))))).)))))).........
2: .(((.((((.......))))))).((((((...(((......)))))))))...((((((
                           ((((((((((.((((....)))))))))))))).))))))
```

The base pair distances evaluated with `RNAdistance` [28] between the structures `0` and `1`, `0` and `2`, and `1` and `2` are respectively 40, 96 and 86.

For every pair of secondary structures, a set of twenty bi-stable sequences was generated with `Frnakenstein` [29], such that the best scoring sequence of each run was kept. For each pair of structures (s_1, s_2), and each sequence z designed on these structures, a complete binary tree T of depth 6 was populated. The root r of T is initialized with $(r^1, r^2) \equiv (z, z)$. Each internal node n of T is composed of a pair of sequences, (n^1, n^2), such that the sequence n^1 is associated with the structure s_1 and n^2 with the structure s_2.

The sequences for each internal node are generated in a bottom up fashion. Given a node w, its sequences (w^1, w^2), its two child nodes c^1, c^2, a mutation probability α, and a substitution matrix β. From (w^1, s_1) (resp (w^2, s_2)), a set W of a thousand mutants of w^1 is generated as follows. Each sequence w_i in W is created by applying a probability of mutation α to each position in the original sequence w^1. Each nucleotide x can be substituted to $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\} \setminus \{x\}$ following the distribution $\beta(x)$. We used for β: $\mathbb{P}(\mathrm{A} \leftrightarrow \mathrm{G}) = \mathbb{P}(\mathrm{C} \leftrightarrow \mathrm{U}) = 50\%$, all others are set to 25%. We used those probabilities for the mutational events based on the observation [30] that transitions are more frequent than transversions.

We define a free energy $E(w_i, s)$ as the base pair distance between the minimal free energy structure of w_i and s, i.e. $\Delta(\mathrm{MFE}(w_i), s)$. The MFE and base pair distance are computed with `RNAfold` and `RNAdistance` [28].

A Boltzmann distribution is induced such that the weight of any sequence w_i is

$$\mathcal{B}(w_i, s) = e^{\frac{-E(w_i,s)}{RT}}$$

where R is the Boltzmann constant and T the temperature in Kelvin. The partition function $\mathcal{Z}^s_W$ is obtained by summing the weights of all sequences $w_i \in W$ and we defined the Boltzmann probability of each sequence $\mathbb{P}^s_W(w_i)$ such that

$$\mathcal{Z}^s_W = \sum_{w_i \in W} \mathcal{B}(w_i, s) \qquad \text{and} \qquad \mathbb{P}^s_W(w_i) = \frac{\mathcal{B}(w_i, s)}{\mathcal{Z}^s}.$$

We sample two sequences from this distribution to populate c^1_1 and c^2_1 (resp. c^1_2 and c^2_2). We re-apply recursively. The generator was implemented in python and is bundled with our `achARNement` package.

Evaluation on simulated data

We first evaluated `achARNement` using simulated data, as described in Sec. Simulated data generation. The mutational rates of bacterias (bacterial genomes are studied in Sec. Evaluation on biological data) are known to vary greatly between species and it is difficult to find indisputable reference points to evaluate them [31]. We thus approximate many generations in each step ($i.e$ level of the tree) by using as the mutation rate α three values: $\{1\%, 5\%, 10\%\}$. This enables us to obtain diverse enough sequences at the leafs of a complete binary tree of depth 6.

For every pair of structures and mutation rate, twenty trees were generated. In Fig. 4, we show the average error percentage over all optimal sequences inferred for both families in all nodes of the trees. We divided the results by structural features; the first row is the average error percentage for positions involved in an interaction, while the second row is for unstructured positions. Each column represents a different pair of secondary structures, annotated `01`, `02` and `12` following the notation defined in Sec. Simulated data generation. For each sequence of a family fam, we consider a position to be in a structured region, if the structure of fam has a base pair at that position.

A first observation is that `CalculateScores-2structs` always performs the best, followed by `CalculateScores-1struct`, and the Fitch and Sankoff algorithms whose performances are equivalent. In all cases, `achARNement` methods always perform better, even in unstructured regions. For `CalculateScores-1struct`, although the other structure is ignored during the parallel ancestral reconstructions, some constraints from the other structure are implicitly taken into account during the middle step when solutions from both families are merged. The higher quality in unstructured regions when using `CalculateScores-2structs` was expected because we always consider structures from both families, and one unstructured position in one family can be structured in the other. Finally, although the two structures `0` and `1` are much closer to each other than to `2`, the basepair distance does not seem to affect the quality of the results.

We then examine the number of optimal solutions, for each pair of secondary structures and mutation rate α. As can be observed in Fig. 5 and Fig. 6, the average number of optimal sequences inferred both in the whole tree and for the root only is always smaller for algorithms `CalculateScores-1struct` and `CalculateScores-2structs`, compared to Fitch and Sankoff. In the case of the pair of structures `01`, the average number of optimal sequences is even several orders of magnitude lower for our two algorithms. An important observation is that, in every case, all sequences at the root reconstructed by `CalculateScores-2structs` are a **subset** of the optimal sequences obtained with the classical Sankoff algorithm (*i.e.* `CalculateScores-Sankoff`). This shows that the additional structural constraints defined in our method help to reduce the initial solution space produced by traditional approaches.

Running times for the four methods are shown in Sup. Mat. Sec. Running times.

Evaluation on biological data

We analyzed the Glm and FinP-traJ clans from the Rfam database. A clan contains two RNA families, that are homologous but functionally and structurally distinct [26]. These clans, with their two functional families with distinct consensus structures, constitute good candidates to test our algorithms.

Glm clan

The Glm clan contains two bacterial small non-coding RNAs, GlmY and GlmZ, that are homologous but functionally distinct. They act in a hierarchical manner to activate the translation of the glmS mRNA [23]. We selected 74 bacterial genomes for which Rfam alignments were available for both families (see the complete list in Sup. Mat. Sec. Biological Data). The phylogeny of the 74 studied bacterial strains was taken from the Pathosystems Ressource Integration Center (PATRIC) [32], and Rfam seed alignements of both families were aligned together with `CARNA` [33]. The sequences in the full Rfam alignments were then re-aligned to the alignment obtained with `CARNA` simply by mapping their corresponding positions. The sequences and structures were subsequently trimmed to remove the gapped columns, and if only one side of an interaction was removed the other position was marked as unstructured.

We used the basic Fitch and Sankoff methods, and our algorithms `CalculateScores-1struct` and `CalculateScores-2structs` to infer the ancestral sequences at the root of the species tree. Both Fitch and Sankoff inferred the same set of 786 432 sequences at the root of the species tree, whereas `CalculateScores-1struct` inferred 393 216 and `CalculateScores-2structs` 196 608. The ancestral sequences reconstructed by our methods are subsets of the ones produced by Fitch and Sankoff: `CalculateScores-1struct` cut the solution space in half and `CalculateScores-2structs` by another half. Running times were of 19 seconds for both Fitch and Sankoff, and 14 seconds for both `CalculateScores-1struct` and `CalculateScores-2structs`. The lower running times for `achARNement` methods could be explained by the smaller numbers of ancestral sequences inferred.

We look at two different measures to evaluate the quality of the ancestral sequences. First, we simply look at the percentage of all structured positions, for each family, that can actually form canonical basepairs in the ancestral sequences. The goal is to see if the reconstructed sequences can form the required basepairs in both structures. Second, we compute the harmonic mean (H-mean) between the frequencies in the ensemble of structures for each structure family (representing GlmY and GlmZ). In a statistical physics framework, an RNA sequence can adopt all structures and its frequency represents the fraction of time that the sequence adopts a particular structure. The harmonic mean is defined as

$$2 \cdot \frac{FreqS1 \times FreqS2}{FreqS1 + FreqS2}$$

and is maximized when both frequencies are at 0.5, given that the structures are different. Thus the H-mean will be equal to 0.5 if the two structures are different and share the complete structure space. Another important feature is the energy of a sequence in a particular configuration. Although that sequence could have other more favorable structures, it gives another idea of the stability of a particular configuration. In order to calculate this mean for a sequence, we compute the free energy of the sequence when folded in the 2 different structures and their frequencies using `RNAfold`, and the non-canonical base pairs are ignored for these computations.

To compare the different ensembles of solutions, we sampled 200 000 distinct sequences from each of them. We present in the first 6 lines of Tab. 2 the maximum and average values of the percentage of canonical basepairs and H-mean for: sequences inferred by `Sankoff` (or Fitch) only, those inferred by `CalculateScores-1struct` but not `CalculateScores-2structs`, and those inferred by `CalculateScores-2structs` only. We also present the values of energy and frequencies in regards to the secondary structure of each family. The standard deviations are shown in the Sup. Mat. Tab. 7.

We observe that, on average, the percentages of canonical basepairs are all the same on the GlmZ structure (99.1%), but it is 1.5% higher for the solutions of `CalculateScores-2structs` on the GlmY structure. Although this is not a huge difference, the fact that we get more canonical basepairs on average by inferring a lot less ancestral sequences is interesting. As for the maximums, in all subsets of solutions we get sequences that have 100% of the canonical basepairs for both structures. The average (resp. max) H-means for the distinct sets of ancestors produced by Sankoff, CalculateScores-1struct and CalculateScores-2structs are roughly similar, indicating that by cutting the solution space with CalculateScores-1struct and CalculateScores-2structs, we do not lose sequences that have significantly better folding properties in regards to both structures.

We then proceeded to do a comparison of our method with the state-of-the-art maximum likelihood ancestral reconstruction program PAML [34]. For clarity, we remind the reader that PAML considers only one family at the time, and returns one ancestor per node. We generated the ancestors using both families separately, GlmZ and GlmY, and compared the two predicted ancestors percentage of canonical basepairs and H-mean with those obtained with the other methods, as shown in Tab. 2. The two ancestral sequences produced by PAML have percentages of canonical basepairs (and H-mean of PAML GlmY) that are significantly lower than the best and average values of all the other methods.

FinP-traJ clan

We also ran the experiment on the FinP-traJ clan. FinP is an antisense ncRNA that can bind to the 5' UTR region of the traJ mRNA. The binding of those two RNAs represses the the translation of traJ, which in turn represses F-plasmid transfer [35]. Similarly to the Glm clan, we selected bacterial genomes for which the Rfam alignments were available for both families (54 genomes; see the complete list in Sup. Mat. Sec. Biological Data) and we did the same preprocessing to prepare the alignments. The phylogeny for the 54 bacterial strains was also taken from PATRIC [32].

Noticeably, both families in this clan are sequentially and structurally more different than with the Glm clan. The Fitch, Sankoff method, and CalculateScores-1struct inferred the same ensemble of 12 582 912 sequences at the root. In contrast, CalculateScores-2structs inferred a strict subset (4x smaller) of 3 145 728 sequences. Running times were of 17 seconds for both Fitch and Sankoff, and 19 seconds for both CalculateScores-1struct and CalculateScores-2structs.

As with Glm, we sampled 200 000 distinct sequences to compare the two sets. We present in the first two rows of Tab. 3 the maximum and average results for the sampled sequences in Others, the set inferred by Fitch, Sankoff, and CalculateScores-1struct but not by CalculateScores-2structs. The following two rows present those sampled in the subset inferred by CalculateScores-2structs. We show the results with their standard deviation in Sup. Mat. Tab. 8.

We observe that on average, the solutions from the “others” group can form 86.8% of the basepairs of the FinP structure and 88.2% of the ones of traJ. On the other hand, the subset of ancestors produced by CalculateScores-2structs can form on average 85.5% (1.3% less) of the basepairs of the FinP structure and 91.2% (3% more) of the ones of traJ, which, overall, seems to be a better compromise. Note that this was achieved by inferring 4 times fewer ancestors.

Regarding the H-mean, the samples taken from the smaller subset of ancestral sequences reconstructed by CalculateScores-2structs show similar results for the maximum and slightly better for the average H-mean than the bigger sets inferred by the other algorithms, which tends to show that our method is not discarding sequences with better folding properties.

We also compared our results with PAML, for each family separately. We observe a stark contrast with our results when comparing the percentage of canonical basepairs for both families. While PAML can get 100% on the considered structure, it gets only about 80% of the basepairs of the other structure. When looking at the stability of the functional structures of the two families on the reconstructed ancestral sequences, we observe that our solutions offer a better trade-off (i.e. the average harmonic mean is several degrees of magnitude better that the ones obtained by PAML).

These results suggest that our methods are indeed capable to retrieve ancestral sequences with better fitness to both functional structures of the homologous RNA families. Since RNA families are known to favour the conservation of structures over sequences, we argue that achARNement solutions are better ancestral candidates.

Discussion

In this paper, we presented two novel maximum parsimony algorithms, implemented in achARNement, to solve the simultaneous ancestral reconstruction of two ncRNA families sharing a common ancestor. We first evaluated our methods on simulated data, as described in Sec. Simulated data generation, then on two Rfam clans, the Glm and FinP-traJ clan (Sec. Evaluation on biological data).

We first showed that on simulated data, achARNement produces smaller sets of ancestral sequences with fewer errors on average than the classical Fitch and Sankoff algorithms. Since all the ancestral sequences reconstructed at the root by achARNement are included in those produced by the Sankoff algorithm, it indicates that considering the secondary structures does not generate superfluous mutations. Most importantly, considering both structures in CalculateScores-2structs produces orders of magnitudes fewer sequences while always improving on the other algorithms in terms of the average percentage of errors.

The biological data can not be validated directly, yet some interesting observations can be made. To the best of our knowledge, achARNement has the first implementations of complete parsimonious models able to reconstruct ancestral sequences of large alignments with multiple structures. On both the Glm and FinP-traj clans, CalculateScores-2structs has been shown to infer smaller sets of ancestral sequences than Fitch and Sankoff, while offering a better compromise in terms of the percentage of canonical basepairs for both structures (without penalizing the folding properties, as shown with the similar values of H-mean). Also, the comparison with PAML highlights the benefits of our approach, especially on the FinP-traJ clan, where it is clear that we are able to infer sequences that have better folding properties in both considered structures.

The evolutionary model and algorithms presented here constitute a first attempt at tackling this specific problem. Although the results are encouraging, a lot more work needs to be done in the future to improve our approach: analyzing more in depth the different parameters of our method, reducing even more the number of ancestral sequences inferred and testing on more Rfam clans are just a few examples. The frequency in the ensemble also raises important questions in regards to how we view neutral networks. Given a ncRNA and its functional conformation, what is the minimal frequency in the structures ensemble needed in order for it to be able to fulfill its function? Equally for the RNA design problem, most methods are based on local searches, from a random search as in RNAinverse [28] or with an ant algorithm as in antaRNA-ant [36]. The observed diversity in the quality of sequences at a minimal distance from each other demonstrates the need of more global tools, like the one of IncaRNAtion [37] for example.

Through the annotation of Rfam families, manual curation is needed to distinguish between families of similar sequences with known distinct function or structure, which are joined into clans [26]. In practice, achARNement could be used for the classification of sequences to the correct clan member. achARNement could also be customized to detect families of sequences folding into multiple structures, as those exhibited in [38, 39].

Competing interests
The authors declare that they have no competing interests.

Author's contributions
All the authors have participated in the conception of the algorithms, the conception of the experiments, the analysis of the results and the writing of the paper. Olivier Tremblay-Savard implemented the algorithms. Vladimir Reinharz implemented the simulated data generator. Olivier Tremblay-Savard and Vladimir Reinharz realized the experiments.

Declarations
This work was made possible by grants from FRQNT, NSERC, and the NSERC Discovery (386596-10) and FRQNT new researchers (FQ-175959) programs. The publication charges for this article were funded by NSERC Discovery grant 386596-10.

Acknowledgements
The authors would like to thank Sebastian Will, Paul Gardner and Alain Laederach for helpful discussions.

Author details
[1]School of Computer Science, McGill University, H3A 0E9 Montreal, Canada. [2]Department of Computer Science, University of Manitoba, R3T 2N2 Winnipeg, Canada.

References

1. Pauling, L., Zuckerkandl, E.: Chemical paleogenetics, molecular restoration studies of extinct forms of life. Acta Chemica Scandinavica **17**(9-16) (1963)
2. Blanchette, M., Green, E.D., Miller, W., Haussler, D.: Reconstructing large regions of an ancestral mammalian genome in silico. Genome Res **14**(12), 2412–23 (2004). doi:10.1101/gr.2800104
3. Snir, S., Pachter, L.: Phylogenetic profiling of insertions and deletions in vertebrate genomes. In: Research in Computational Molecular Biology, 10th Annual International Conference, RECOMB 2006, Venice, Italy, April 2-5, 2006, Proceedings, pp. 265–280 (2006). doi:10.1007/11732990_23. http://dx.doi.org/10.1007/11732990_23
4. Paten, B., Herrero, J., Fitzgerald, S., Beal, K., Flicek, P., Holmes, I., Birney, E.: Genome-wide nucleotide-level mammalian ancestor reconstruction. Genome Res **18**(11), 1829–43 (2008). doi:10.1101/gr.076521.108
5. Higgs, P.G., Lehman, N.: The rna world: molecular cooperation at the origins of life. Nat Rev Genet **16**(1), 7–17 (2015). doi:10.1038/nrg3841
6. Chen, Y., Carlini, D.B., Baines, J.F., Parsch, J., Braverman, J.M., Tanda, S., Stephan, W.: Rna secondary structure and compensatory evolution. Genes Genet Syst **74**(6), 271–86 (1999)
7. Gruber, A.R., Bernhart, S.H., Hofacker, I.L., Washietl, S.: Strategies for measuring evolutionary conservation of rna secondary structures. BMC Bioinformatics **9**, 122 (2008). doi:10.1186/1471-2105-9-122
8. Knies, J.L., Dang, K.K., Vision, T.J., Hoffman, N.G., Swanstrom, R., Burch, C.L.: Compensatory evolution in rna secondary structures increases substitution rate variation among sites. Mol Biol Evol **25**(8), 1778–87 (2008). doi:10.1093/molbev/msn130
9. Srivastava, A., Cai, L., Mrázek, J., Malmberg, R.L.: Mutational patterns in rna secondary structure evolution examined in three rna families. PLoS One **6**(6), 20484 (2011). doi:10.1371/journal.pone.0020484
10. Nasrallah, C.A.: The dynamics of alternative pathways to compensatory substitution. BMC Bioinformatics **14 Suppl 15**, 2 (2013). doi:10.1186/1471-2105-14-S15-S2
11. de Boer, F.K., Hogeweg, P.: Mutation rates and evolution of multiple coding in rna-based protocells. J Mol Evol **79**(5-6), 193–203 (2014). doi:10.1007/s00239-014-9648-6
12. Hoeppner, M.P., Gardner, P.P., Poole, A.M.: Comparative analysis of rna families reveals distinct repertoires for each domain of life. PLoS Comput Biol **8**(11), 1002752 (2012). doi:10.1371/journal.pcbi.1002752
13. Tinoco, I. Jr, Bustamante, C.: How rna folds. J Mol Biol **293**(2), 271–81 (1999). doi:10.1006/jmbi.1999.3001
14. Gutell, R.R., Lee, J.C., Cannone, J.J.: The accuracy of ribosomal rna comparative structure models. Curr Opin Struct Biol **12**(3), 301–10 (2002)

15. Bradley, R.K., Holmes, I.: Evolutionary triplet models of structured rna. PLoS Comput Biol **5**(8), 1000483 (2009). doi:10.1371/journal.pcbi.1000483
16. Klosterman, P.S., Uzilov, A.V., Bendaña, Y.R., Bradley, R.K., Chao, S., Kosiol, C., Goldman, N., Holmes, I.: Xrate: a fast prototyping, training and annotation tool for phylo-grammars. BMC Bioinformatics **7**, 428 (2006). doi:10.1186/1471-2105-7-428
17. Yao, Z., Weinberg, Z., Ruzzo, W.L.: Cmfinder–a covariance model based rna motif finding algorithm. Bioinformatics **22**(4), 445–52 (2006). doi:10.1093/bioinformatics/btk008
18. Knudsen, B., Hein, J.: Rna secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics **15**(6), 446–54 (1999)
19. Eddy, S.R., Durbin, R.: Rna sequence analysis using covariance models. Nucleic Acids Res **22**(11), 2079–88 (1994)
20. Schuster, P., Fontana, W., Stadler, P.F., Hofacker, I.L.: From sequences to shapes and back: a case study in RNA secondary structures. Proc Biol Sci **255**(1344), 279–84 (1994). doi:10.1098/rspb.1994.0040
21. Fitch, W.M.: Toward defining the course of evolution: Minimum change for a specific tree topology. Systematic Biology **20**(4), 406–416 (1971)
22. Sankoff, D.: Minimal mutation trees of sequences. SIAM Journal on Applied Mathematics **28**(1), 35–42 (1975)
23. Urban, J.H., Vogel, J.: Two seemingly homologous noncoding rnas act hierarchically to activate glms mrna translation. PLoS Biol **6**(3), 64 (2008). doi:10.1371/journal.pbio.0060064
24. Biesen, T., Soderbom, F., Wagner, E.G.H., Frost, L.S.: Structural and functional analyses of the finp antisense rna regulatory system of the f conjugative piasmid. Molecular microbiology **10**(1), 35–43 (1993)
25. Nawrocki, E.P., Burge, S.W., Bateman, A., Daub, J., Eberhardt, R.Y., Eddy, S.R., Floden, E.W., Gardner, P.P., Jones, T.A., Tate, J., et al.: Rfam 12.0: updates to the RNA families database. Nucleic acids research, 1063 (2014)
26. Gardner, P.P., Daub, J., Tate, J., Moore, B.L., Osuch, I.H., Griffiths-Jones, S., Finn, R.D., Nawrocki, E.P., Kolbe, D.L., Eddy, S.R., Bateman, A.: Rfam: Wikipedia, clans and the "decimal" release. Nucleic Acids Res **39**(Database issue), 141–5 (2011). doi:10.1093/nar/gkq1129
27. Stombaugh, J., Zirbel, C.L., Westhof, E., Leontis, N.B.: Frequency and isostericity of rna base pairs. Nucleic acids research **37**(7), 2294–2312 (2009)
28. Lorenz, R., Bernhart, S.H., Zu Siederdissen, C.H., Tafer, H., Flamm, C., Stadler, P.F., Hofacker, I.L., *et al.*: ViennaRNA package 2.0. Algorithms for Molecular Biology **6**(1), 26 (2011)
29. Lyngsø, R.B., Anderson, J.W., Sizikova, E., Badugu, A., Hyland, T., Hein, J.: Frnakenstein: multiple target inverse RNA folding. BMC bioinformatics **13**(1), 260 (2012)
30. Vogel, F., Kopun, M.: Higher frequencies of transitions among point mutations. Journal of molecular evolution **9**(2), 159–180 (1977)
31. Tremblay-Savard, O., Benzaid, B., Lang, B.F., El-Mabrouk, N.: Evolution of trna repertoires in bacillus inferred with orthoalign. Molecular Biology and Evolution **32**(6), 1643–1656 (2015). doi:10.1093/molbev/msv029. http://mbe.oxfordjournals.org/content/32/6/1643.full.pdf+html
32. Gillespie, J.J., Wattam, A.R., Cammer, S.A., Gabbard, J.L., Shukla, M.P., Dalay, O., Driscoll, T., Hix, D., Mane, S.P., Mao, C., Nordberg, E.K., Scott, M., Schulman, J.R., Snyder, E.E., Sullivan, D.E., Wang, C., Warren, A., Williams, K.P., Xue, T., Seung Yoo, H., Zhang, C., Zhang, Y., Will, R., Kenyon, R.W., Sobral, B.W.: Patric: the comprehensive bacterial bioinformatics resource with a focus on human pathogenic species. Infection and Immunity **79**(11), 4286–4298 (2011). doi:10.1128/IAI.00207-11. http://iai.asm.org/content/79/11/4286.full.pdf+html
33. Sorescu, D.A., Möhl, M., Mann, M., Backofen, R., Will, S.: CARNA—alignment of RNA structure ensembles. Nucleic acids research, 491 (2012)
34. Yang, Z.: Paml 4: phylogenetic analysis by maximum likelihood. Molecular biology and evolution **24**(8), 1586–1591 (2007)
35. Jerome, L.J., van Biesen, T., Frost, L.S.: Degradation of finp antisense {RNA} from f-like plasmids: the rna-binding protein, fino, protects finp from ribonuclease {E1}. Journal of Molecular Biology **285**(4), 1457–1473 (1999). doi:10.1006/jmbi.1998.2404
36. Robert, K., Martin, M., Rolf, B.: antarna-ant colony based rna sequence design. Bioinformatics, 319 (2015)
37. Reinharz, V., Ponty, Y., Waldispühl, J.: A weighted sampling algorithm for the design of rna sequences with targeted secondary structure and nucleotide distribution. Bioinformatics **29**(13), 308–315 (2013)
38. Kutchko, K.M., Sanders, W., Ziehr, B., Phillips, G., Solem, A., Halvorsen, M., Weeks, K.M., Moorman, N., Laederach, A.: Multiple conformations are a conserved and regulatory feature of the rb1 5' utr. RNA **21**(7), 1274–85 (2015). doi:10.1261/rna.049221.114
39. Rogers, E., Heitsch, C.E.: Profiling small rna reveals multimodal substructural signals in a boltzmann ensemble. Nucleic Acids Res **42**(22), 171 (2014). doi:10.1093/nar/gku959

Figure legends

Figure 1

Our approach. Up: The red and blue areas represent regions of the sequence landscape of sequences with "good" affinity (i.e. sufficient to carry the associated function) to the target structures $\mathcal{S}$ (red) and $\mathcal{S}'$ (blue). Here, α and α' are paralogous sequences, as well as β and β', γ and γ' and δ and δ'. Using classical reconstruction approaches, $\mathcal{A}$ would be the inferred ancestor of the orthologous sequences α, β, γ and δ, and $\mathcal{A}'$ would be the inferred ancestor of the orthologous sequences α', β', γ' and δ'. Shaded trees represent the classical ancestral reconstructions performed separately, while the main tree rooted at $\mathcal{AA}'$ represents the *simultaneous* ancestral reconstruction approach introduced in this contribution. The rationale of this work is that ancestors inferred from a single family and structure may have a tendency to be located in the core of the affinity regions, and might end up with ancestral sequences that would be hard to reconcile. By contrast, a simultaneous reconstruction of orthologous families ensures the coherency of the process and a better inference of the ancestors (which are not necessarily located in the core of the affinity regions). Down: An example of a species tree T (dashed lines) of four species A,

B, Γ and Δ corresponding to the neutral networks shown on top. A duplication event is shown at the root, creating the two ncRNA families (represented by colored lines). Each node of the species tree contains a copy of each ncRNA family (one red, one blue). At the leaves of the species tree T, we find the two extant ncRNAs for which we have the sequence and the structure information. The linear gradient G is also shown: it represents the weight that is given to each structure when calculating the costs (G for one structure and (1-G) for the other).

Figure 2
Graphical representation of the algorithm `CalculateScores-2structs`. In this example, we have four species (A, B, C and D) and for each species, we have two extant RNAs (for family 1, in red, and family 2, in blue). The three major steps of the algorithm are presented. 1) The bottom-up step, where minimum scores are calculated at every node of the tree for each family. The scores take into account the substitutions, but also the basepair cost for the current family, and for the other family. 2) The middle step. Here we link the minimum score matrices for families 1 and 2 by doing a simple Fitch on the two matrices. This allows us to reconstruct the original ancestral sequences (before the duplication), taking into account both families. 3) The top-down step, where we start from the root and select the nucleotides of minimum cost at every position and construct the optimal sequences.

Figure 3
Three examples of the positions that need to be considered when using information from both structures. Note that in those examples, we consider that we are working on the sequence of family 1, and fam1 and fam2 represent the 2D structures of family 1 and 2 respectively. a) The easier case when the position (8 here) is not paired in fam1, and we only have to consider the position paired with it in fam2. b) The case where only one of the two paired positions of fam1 is paired in fam2. b) The case where both paired positions of fam1 are paired in fam2.

Figure 4
The average error percentage of all optimal sequences for both families in a tree. Each column represents a pair of secondary structures. The first row is for positions in structured regions, and the second row for unstructured regions. For three mutation rates: 1%, 5% and 10%.

Figure 5
Average number of optimal sequences in the **tree**, y-axis logscale. Each column represents a different pair of secondary structures. For three mutation rates: 1%, 5% and 10%.

Figure 6
Average number of optimal sequences in the **root**, y-axis logscale. Each column represents a different pair of secondary structures. For three mutation rates: 1%, 5% and 10%.

Tables

	A	C	G	U
A	0	2	1	2
C	2	0	2	1
G	1	2	0	2
U	2	1	2	0

(a)

	A	C	G	U
A	0.003	0.003	0.003	0.001
C	0.003	0.003	0	0.003
G	0.003	0	0.003	0.002
U	0.001	0.003	0.002	0.003

(b)

Table 1: (a) Nucleotide substitution matrix. Each cell $C_{i,j}$ represents the cost of mutating the nucleotide i into j. (b) Basepair cost matrix. Each cell $C_{i,j}$ represents the cost of having the basepair i-j.

	%Z	%Y	H-mean	EnSZ	FreqSZ	EnSY	FreqSY
`Sankoff`	100	100	1.40e-03	-18.7	1.40e-03	-18.7	1.40e-03
average	99.1	93.0	6.60e-06	-15.5	4.09e-06	-17.8	1.55e-04
`1struct`	100	100	8.76e-03	-19.3	8.76e-03	-19.3	8.76e-03
average	99.1	93.0	9.02e-05	-16.1	5.54e-05	-17.4	5.68e-04
`2struct`	100	100	2.45e-03	-19.3	1.27e-03	-21.3	3.26e-02
average	99.1	94.5	7.45e-06	-16.1	3.78e-06	-19.4	9.27e-04
`PAML` GlmZ	92.9	90.6	1.53e-05	-18.1	7.66e-06	-22.6	1.14e-02
`PAML` GlmY	92.9	90.6	1.98e-07	-17.5	9.90e-08	-22.5	3.30e-04

Table 2: Maximum and average results for the Glm Clan. The %Z (resp. %Y) column shows the percentage of all structured positions in the GlmZ (resp. GlmY) family for which the ancestral sequences can form canonical basepairs. The H-mean column represents the harmonic mean. The EnSZ column (resp. EnSY) shows the energy of the sequence when folded in the secondary structure of the family GlmZ (resp. GlmY). The FreqSZ column (resp. FreqSY) shows the frequency in the ensemble of the secondary structure of GlmZ (resp. GlmY). The first six rows show maximum and average results for `Sankoff`, `CalculateScores-1struct` and `CalculateScores-2structs` algorithms. The last two rows represent values obtained for the `PAML` root ancestral sequence reconstructed on the GlmZ family and on the GlmY family.

	%F	%t	H-mean	EnSF	FreqSF	EnSt	FreqSt
`Others`	94.7*	100*	6.25e-01	-23.7	6.25e-01	-23.7	6.25e-01
average	86.8	88.2	3.17e-03	-27.5	1.97e-02	-24.4	3.65e-03
`2struct`	94.7*	100*	5.58e-01	-23.7	5.58e-01	-23.7	5.58e-01
average	85.5	91.2	3.86e-03	-27.0	9.62e-03	-25.3	1.39e-02
`PAML` FinP	100	82.4	7.13e-08	-28.3	2.59e-03	-21.4	3.56e-08
`PAML` traJ	78.9	100	2.61e-07	-26.3	2.84e-07	-26.2	2.42e-07

Table 3: Maximum and average results for the FinP-traJ Clan. The %F (resp. %t) column shows the percentage of all structured positions in the FinP (resp. traJ) family for which the ancestral sequences can form canonical basepairs. The H-mean column represents the harmonic mean. The EnSF column (resp. EnSt) shows the energy of the sequence when folded in the secondary structure of the family FinP (resp. traJ). The FreqSF column (resp. FreqSt) shows the frequency in the ensemble of the secondary structure of FinP (resp. traJ). The first four rows show maximum and average results for the first three algorithms (`Others`) and `CalculateScores-2structs`. The last two rows represent values obtained for the `PAML` root ancestral sequence reconstructed on the FinP family and on the traJ family.
*: The maximum values showed are the ones maximizing the basepairs in traJ; the ones maximizing FinP are 100 and 94.1 for %F and %t respectively.

Additional files

The file `supplemantarymaterial.pdf` contains the algorithms of `CalculateScores-1struct` and `CalculateScores-2structs`. It is followed by the running times of Fitch, Sankoff, `CalculateScores-1struct` and `CalculateScores-2structs` on the simulated data sets. It also contains the list of RNA strains used in the biological analysis. We also present additional results on the Glm and FinP-traJ clans reconstructions.