BMC Bioinformatics

Methodology article

Open Access

Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization

Markus Bauer*[1,2], Gunnar W Klau*[1,3] and Knut Reinert[1]

Address: [1]Department of Mathematics and Computer Science, Free University Berlin, 14195 Berlin, Germany, [2]International Max Planck Research School for Computational Biology and Scientific Computing, Berlin, Germany and [3]DFG Research Center Matheon, Berlin, Germany

Email: Markus Bauer* - mbauer@inf.fu-berlin.de; Gunnar W Klau* - gunnar@math.fu-berlin.de; Knut Reinert - reinert@inf.fu-berlin.de

* Corresponding authors

Published: 27 July 2007

BMC Bioinformatics 2007, **8**:271 doi:10.1186/1471-2105-8-271

Received: 30 March 2007

Accepted: 27 July 2007

This article is available from: http://www.biomedcentral.com/1471-2105/8/271

Abstract

Background: The discovery of functional non-coding RNA sequences has led to an increasing interest in algorithms related to RNA analysis. Traditional sequence alignment algorithms, however, fail at computing reliable alignments of low-homology RNA sequences. The spatial conformation of RNA sequences largely determines their function, and therefore RNA alignment algorithms have to take structural information into account.

Results: We present a graph-based representation for sequence-structure alignments, which we model as an integer linear program (ILP). We sketch how we compute an optimal or near-optimal solution to the ILP using methods from combinatorial optimization, and present results on a recently published benchmark set for RNA alignments.

Conclusion: The implementation of our algorithm yields better alignments in terms of two published scores than the other programs that we tested: This is especially the case with an increasing number of input sequences. Our program LARA is freely available for academic purposes from http://www.planet-lisa.net.

1 Background

In recent years, research in RNA sequences and structures has dramatically increased: the discovery of functionally important, not protein-coding, RNA sequences has challenged the traditional picture of the flow of genetic information from DNA via RNA to proteins as functional units. It is now well-established that RNA molecules introduce an additional layer in genetic information processing. They play a significant active role in cell and developmental biology and carry out many tasks that were previously attributed exclusively to proteins. One of the most eminent examples is the class of microRNAs [1,2], an abundant class of small functional RNAs that regulate gene expression by binding to a target in the mRNA. Other examples include snoRNAs, which modify ribosomal RNA [3], *signal recognition particle* RNAs [4], *cis*-acting regulatory elements, and piRNAs [5], a novel class of ncRNAs whose function is still unclear. It is likely that only a small fraction of regulatory RNAs has been identified so far and that many more have yet to be discovered [6].

Computational analyses have contributed largely to the discovery and advancement of biological knowledge. Heuristic methods, such as BLAST [7], or exact approaches based on dynamic programming, such as the Smith-Waterman algorithm [8], are used as everyday tools to analyze DNA and protein sequences. In case of RNA sequences, sequence information alone is not sufficient anymore. An RNA sequence folds back onto itself and forms hydrogen bonds between nucleotides. These bonds lead to the distinctive *secondary structure* of an RNA sequence.

RNA sequences evolve more rapidly than the structure they are forming, because their evolutionary behavior follows the *structure-function* paradigm: RNA molecules with different sequences but same or similar secondary structure are likely to belong to the same functional family, in which the secondary structure is conserved by selective pressure. Hence, computational analysis of RNA molecules inevitably involves considering secondary structure information in addition to the primary sequence. Computing *sequence-structure alignments* is a key step in many important applications. These include finding homologous structures of known ncRNA families [9], phylogenetic fingerprinting (as conducted for example for the ITS2 database [10]), or the computation of a consensus structure of a set of RNA molecules [11]. A recent study shows that pure sequence-based pairwise alignments are unable to produce satisfactory results if the pairwise sequence identity drops below 50 to 60% [12]. Figure 1 illustrates this situation and shows two different alignments of seven tRNA sequences with a pairwise sequence identity of 39%, where the upper alignment is based on sequence information alone and the lower alignment additionally rewards the conservation of structural elements. One can clearly see that the sequence-based alignment is unable to preserve the typical tRNA-cloverleaf structure, whereas the structural alignment conserves the structural features of the input sequences.

Unfortunately, considering structural information adds an additional level of complexity to the problem of aligning two or several sequences. In the remainder of this section, we present a classification of structural alignment problem variants including previous work. Section 2 describes our new approach to multiple sequence-structure alignment. We employ methods from mathematical programming and solve the problem as an integer linear program resulting from a graph-theoretical reformulation. Section 3 is dedicated to an extensive computational study. We describe LARA, the freely available implementation of our novel approach, and present detailed results of a comparative study including state-of-the-art programs on a recently published benchmark database of structural alignments. The results show that on average our software is currently the best program in terms of alignment quality, outperforming other programs with an increasing number of input sequences. Finally, we discuss our results and suggest future research directions in Sect. 4.

In contrast to previous work [13-15] this article describes a full integer linear programming (ILP) formulation that does include arbitrary gap costs and an extensive performance analysis of our implementation for the first time. Due to page limits the mathematical fundament and all proofs are omitted: the interested reader is referred to the companion paper [16] that focuses on an in-depth description of the mathematical properties of the intricate multiple case containing all proofs.

1.1 Previous approaches

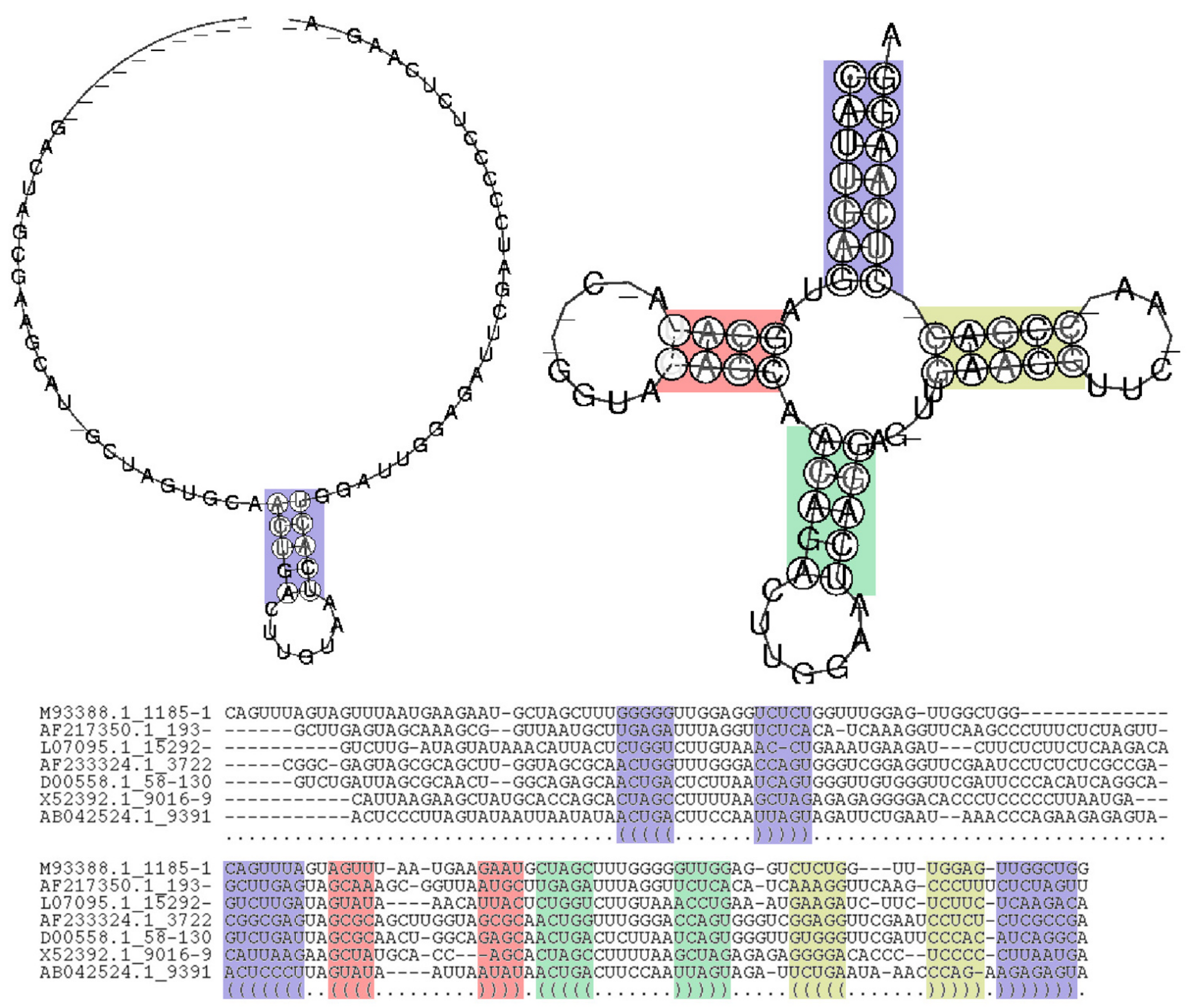
Depending on the available knowledge about the (putative) structures that we want to align, there are three different alignment scenarios for two RNA structures, which readily extend to the multiple case. Figure 2 gives a cartoon illustration of the three scenarios.

1. *Structure-to-structure* alignments align two known secondary structures, typically the *minimum free energy* structures. This scenario applies if one searches for common structural motifs that are shared by both structures and there is reason to believe that the secondary structures are correct.

2. *Structure-to-unknown* alignments align a given structure to a sequence with unknown structure. Applications are finding homologous sequences by inferring a consensus structure to a sequence (this is done, for example, in the verification phase of the FASTR package [9]), or finding new family members of ncRNA families: This problem has recently sparked considerable interest in the context of searching homologous structures of noncoding-RNAs in large genomic sequences. See [17] for a survey.

3. In the *unknown-to-unknown* alignment problem, no previous structural information is given. It applies when two RNA sequences are suspected to share a common, but still unknown, structure. We constrain the space of possible structures by the entire set of possible Watson-Crick and wobble pairs. A reduction of the size of this space is possible, for instance, by applying a folding algorithm to obtain the base pair probabilities [18] and then considering only those interactions whose probabilities are above a certain threshold.

There are four major alignment models for RNA structures that tackle the previous described alignment scenarios: *annotated sequences*, *tree models*, *probabilistic models*, and *graph-based models*. We give small examples for each model in Fig. 3: Note that we did not show an example of probabilistic models because the representation of probabilistic and tree models are the same. The underlying algorithms, however, are completely different. Table 1 classifies previous work in the area of structural RNA alignment according to the different models and scenarios.



```
M93388.1_1185-1 CAGUUUAGUAGUUUAAUGAAGAAU-GCUAGCUUUGGGGGUUGGAGGUCUCUGGUUUGGAG-UUGGCUGG-------------
AF217350.1_193- ------GCUUGAGUAGCAAAGCG--GUUAAUGCUUGAGAUUUAGGUUCUCACA-UCAAAGGUUCAAGCCCUUUCUCUAGUU-
L07095.1_15292- ----------GUCUUG-AUAGUAUAAACAUUACUCUGGUCUUGUAAAC-CUGAAAUGAAGAU---CUUCUCUUCUCAAGACA
AF233324.1_3722 -----CGGC-GAGUAGCGCAGCUU-GGUAGCGCAACUGGUUUGGGACCAGUGGGUCGGAGGUUCGAAUCCUCUCUCGCCGA-
D00558.1_58-130 ------GUCUGAUUAGCGCAACU--GGCAGAGCAACUGACUCUUAAUCAGUGGGUUGUGGGUUCGAUUCCCACAUCAGGCA-
X52392.1_9016-9 -----------CAUUAAGAAGCUAUGCACCAGCACUAGCCUUUUAAGCUAGAGAGAGGGGACACCCUCCCCCUUAAUGA---
AB042524.1_9391 -----------ACUCCCUUAGUAUAAUUAAUAUAACUGACUUCCAAUUAGUAGAUUCUGAAU--AAACCCAGAAGAGAGUA-
                ..................................(((((.......))))).............................

M93388.1_1185-1 CAGUUUAGUAGUUU-AA-UGAAGAAUGCUAGCUUUGGGGGUUGGAG-GUCUCUGG---UU-UGGAG-UUGGCUGG
AF217350.1_193- GCUUGAGUAGCAAAGC-GGUUAAUGCUUGAGAUUUAGGUUCUCACA-UCAAAGGUUCAAG-CCCUUUCUCUAGUU
L07095.1_15292- GUCUUGAUAGUAUA----AACAUUACUCUGGUCUUGUAAACCUGAA-AUGAAGAUC-UUC-UCUUC-UCAAGACA
AF233324.1_3722 CGGCGAGUAGCGCAGCUUGGUAGCGCAACUGGUUUGGGACCAGUGGGUCGGAGGUUCGAAUCCUCU-CUCGCCGA
D00558.1_58-130 GUCUGAUUAGCGCAACU-GGCAGAGCAACUGACUCUUAAUCAGUGGGUUGUGGGUUCGAUUCCCAC-AUCAGGCA
X52392.1_9016-9 CAUUAAGAAGCUAUGCA-CC---AGCACUAGCCUUUUAAGCUAGAGAGAGGGGACACCC--UCCCC-CUUAAUGA
AB042524.1_9391 ACUCCCUUAGUAUA----AUUAAUAUAACUGACUUCCAAUUAGUAGA-UUCUGAAUA-AACCCAG-AAGAGAGUA
                (((((((..((((.........)))).(((((.......))))).....(((((.......))))).))))))).
```

Figure 1
Comparison between sequence and sequence-structure alignments. Comparison between sequence-based (upper, computed by the CLUSTALW program [60]) and sequence-structure-based alignment (lower, computed by LARA, an implementation of our new approach). The left and right consensus structures are based on the ClustalW and LaRA alignment, respectively (consensus structures generated using RNAalifold [11]).

Tree-based models
Tree-based structural alignment algorithms view an RNA secondary structure as a tree. Depending on the particular model (either tree-editing [19] or tree alignment [20]), one either searches for the minimal number of operations (node inserting, node deletion, and node substitution) to transform one tree into the other, or into a common supertree. Algorithms employing the model from [20] have time complexities in $O(n^4)$, thus making the computation expensive. Here and in the following, n denotes the size of the longest sequence. Tree-alignment algorithms have complexities that are on average only slightly worse than conventional sequence alignment. More precisely, their running time is in $O(n^2 \cdot \Delta^2)$, where Δ denotes the maximum number of branches of a multiloop in the input structures.

A tool that builds upon the tree paradigm is RNAFORESTER [21]. It computes multiple structure-to-structure alignments of RNA sequences by performing tree-alignment in a progressive fashion.

Annotated sequences
We call a sequence that is augmented by structural information an *annotated sequence*. Classical dynamic programming (DP) algorithms can be extended to annotated

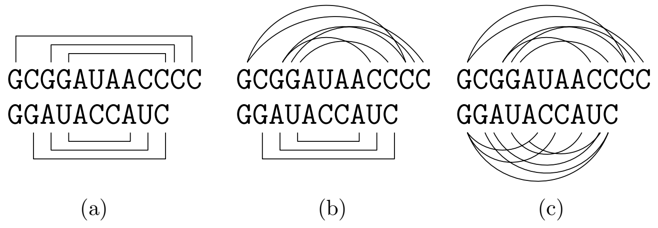


Figure 2
Input scenarios. Different input alignment scenarios of RNA sequences (pairwise case): (a) the alignment of two known structures, (b) of one known and one unknown structure, and (c) of two unknown structures.

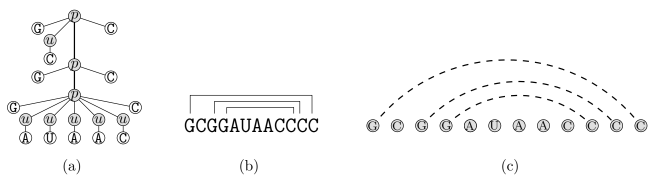


Figure 3
RNA representations. Different models representing RNA structures: (a) tree representation, (b) annotated sequences, and (c) graph-based models.

sequences. The DP solution for the structure-to-structure and structure-to-unknown problem then typically requires $O(n^4)$ and $O(n^3)$ in time and space, respectively. Bafna, Muthukrishnan, and Ravi describe an algorithm that simultaneously aligns the sequence and secondary structure of two RNA sequences [22]. Their method runs in time $O(n^4)$, which still does not make it applicable to instances of realistic size. Eddy [23] proposes an algorithm that reduces the memory consumption to $O(n^2 \log n)$. The STRAL tool [24] uses the values of the *base pair probability matrices*, as given by the partition function [18], to compute the maximal pairing probability of a single nucleotide and to align the sequences in a CLUSTALW-like fashion.

In the restricted structure-to-structure scenario, one can resort to more sophisticated edit-models like the one proposed by Jiang in [25] where the authors specify operations both on the sequence and the structure level. The dynamic programming algorithm is in $O(n^4)$, making the computation rather tedious for longer sequences. A program that implements the Jiang model is MARNA [26]: it computes pairwise sequence-structure alignments, but is additionally able to compute multiple alignments. To this end, MARNA computes all pairwise structural alignment and uses T-COFFEE to compute the actual multiple alignment incorporating the structural information of the pairwise alignments.

The unknown-to-unknown scenario requires the simultaneous computation of the alignment and consensus structure. The computational problem of simultaneously considering sequence and structure of an RNA molecule was initially addressed by Sankoff in [27], where the author proposed a DP algorithm to align and fold a set of RNA sequences at the same time. The CPU and memory requirements of the original algorithm are $O(n^{3k})$ and $O(n^{2k})$, respectively, where k is the number of sequences and n is their maximal length. Current implementations modify Sankoff's algorithm by imposing limits on the size or shape of substructures, e.g., DYNALIGN [28,29], or FOLDALIGN [30] that combine a sliding window and banded alignment approach. Hofacker, Bernhart, and Stadler [31] have presented the PMMULTI software to align base pair probability matrices. Their recursions are essentially the same as the ones given by Sankoff in [27] and subsequently used for sequence-structure alignment by Bafna et al. in [22] with the only difference that they consider probabilities instead of fixed structures. By banding the range of possible alignment positions they bring the time and space complexity of the pairwise case down to $O(n^4)$ and $O(n^3)$, respectively. For the multiple case, they align consensus base pair probability matrices in a progressive fashion. Similar in spirit are FOLDALIGNM [32] or LOCARNA [33], two recent reimplementations of the PMMULTI approach. FOLDALIGNM provides both several restrictions on the alignment and a two-stage procedure to fill the DP matrix: this further reduces the running time to $O(n^2\delta^2)$ where n is the length of the longer sequence and d is the maximal length difference of the alignment of two subsequences. LOCARNA on the other hand takes advantage of the sparse base pair probabilities matrices to reduce the running time.

Table 1: Classification of previous work

	tree-based	annotated sequences	probabilistic	graph-based
structure-to-structure	[20-22,61]	[22,25,26,62]	[34]	[14,41,43]
structure-to-unknown	--	[22]	[34,38]	[14,41,43]
unknown-to-unknown	--	[22,27–29,31]	[37]	[14,41,43]

Probabilistic models

Eddy and Durbin [34] describe *covariance models* for measuring the secondary structure and primary sequence consensus of RNA sequence families. They present algorithms for analyzing and comparing RNA sequences as well as database search techniques. Since the basic operation in their approach is an expensive dynamic programming algorithm, their algorithms cannot analyze sequences longer than 150–200 nucleotides. Therefore, recent approaches reduce the running time by incorporating additional information, e.g. Holmes et al.'s STEMLOC [35,36] where the authors propose the concept of *alignment/fold envelopes* that constrain possible alignments. Along these lines, in [37] the authors keep a set of probabilistically derived alignment positions fixed: these alignment positions serve subsequently as anchors for the structural alignment which prune away large parts of the search space. The authors of [38] describe a method based on conditional random fields to align an RNA sequence with known structure to one with unknown structure. They estimate their parameters using conditional random fields and compute the alignment using the recursions from [39].

Graph-based models

Kececioglu [40] has introduced a graph-theoretical model for the classical primary sequence alignment problem. In [41] the authors present a first extension of this model to RNA structures and propose a branch-and-cut approach based on an integer linear programming formulation. Based on this formulation and inspired by the successful application of Lagrangian relaxation by Lancia and Caprara [42] to the related *contact map overlap problem*, in [43] the authors switch from branch-and-cut to the Lagrangian relaxation technique. They are able to solve instances a magnitude larger by simultaneously reducing the running time significantly. In [44] the authors give a graph-theoretic model for the computation of multiple sequence alignments with arbitrary gap costs. In the next section we will combine the formulations given in [43] and [44], resulting in a novel graph-based formulation for sequence-structure alignment with arbitrary gap costs.

Note that the graph-based model naturally deals with all three alignment scenarios. In addition, unlike other algorithmic approaches, the graph-based algorithms do not restrict the input in any way and hence can handle arbitrary *pseudoknots*: Pseudoknots have been shown to play important roles in a variety of biological processes, see [45] for a recent review. Most DP-based algorithms assume nested secondary structures to compute subproblems efficiently. Few exceptions exist, for example [46], but these algorithms are always restricted to certain classes of pseudoknots (like H-type pseudoknots) and do not handle the general case.

2 Results

This section deals with our novel graph-based approach to structural RNA alignment. We first give the problem definition and then describe the graph-theoretical model we use, which combines the models presented in [43] and [44]. We convert the nucleotides of the input sequences into vertices of a graph, and we add edges between the vertices that represent either structural information or possible alignments of pairs of nucleotides. Based on the graph model we develop an integer linear programming formulation. We find solutions using an algorithmic approach employing methods from combinatorial optimization. For sake of simplicity, we will limit the description to the two-sequence case. We want to stress, however, that the model can be extended to the multiple case without changing the core algorithms and ideas. The interested reader is referred to an extensive theoretical description including proofs and a computational complexity discussion appearing elsewhere [16].

2.1 Graph-theoretical model for structural RNA alignment

Problem definition

Given two RNA sequences, we denote by $\mathcal{A}$ an alignment of the two sequences. Let $s_S(\mathcal{A})$ be the sequence score of alignment $\mathcal{A}$ including gap penalties, and let $s_P(\mathcal{A})$ be the score of structural features that are conserved by the alignment $\mathcal{A}$. We now aim at maximizing the combined sequence-structure score, that is, we search for an alignment $\mathcal{A}^*$ with

$$s_S(\mathcal{A}^*) + s_P(\mathcal{A}^*) = \max_{\mathcal{A}} s_S(\mathcal{A}) + s_P(\mathcal{A}).$$

Figure 4 gives a toy example showing two annotated sequences and two possible alignments, one maximizing the score of sequence and structure, and the other one just the sequence score alone. This problem definition comprises the one addressed by Bafna et al. in [22]: Our

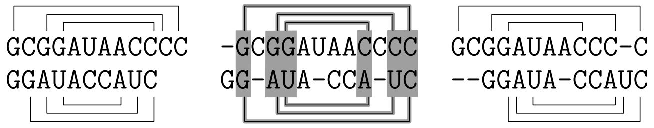


Figure 4
Problem statement. Given the annotated sequences on the left side as the input, we search for an alignment maximizing the sequence plus the induced structure score. The alignment in the middle conserves the entire annotation (highlighted in grey), whereas the alignment on the right hand side maximizes the sequence score and does not conserve any structure.

model, however, also allows tertiary elements, which is not covered by their recursions.

Basic model

Let $s = s_1, \ldots, s_n$ be a sequence of length n over the alphabet $\Sigma = \{A, C, G, U\}$. A pair (s_i, s_j) is called an *interaction* if $i < j$, and nucleotide i pairs with j. In most cases, these pairs will be Watson-Crick or wobble base pairs. The set p of interactions is called the *annotation* of sequence s. Two interactions (s_k, s_l) and (s_m, s_o) are said to be *inconsistent*, if they share one base; they form a *pseudoknot* if they "cross" each other, that is, if $k < m < l < o$ or $m < k < o < l$. A pair (s, p) is called an *annotated sequence*. Note that a structure where no pair of interactions is inconsistent with each other forms a valid secondary structure of an RNA sequence, possibly with pseudoknots.

We are given two annotated sequences (s^A, p^A) and (s^B, p^B) and model the input as a *structural graph* $G_S = (V, L)$. The set V denotes the vertices of the graph, in this case the bases of the sequences, and we write v_i^A and v_i^B for the ith base in sequence A and B, respectively. The set L contains undirected *alignment edges* between vertices of sequences A and B, for sake of better distinction called *lines*. A line $l \in L$ with $l = (v_k^A, v_l^B)$ represents the alignment of the k-th character in sequence A with the l-th character in sequence B. By $s(l)$ and $t(l)$ we refer to the adjacent vertices of line l in sequence A and B, respectively. A subset $\mathcal{L} \subset L$ represents a *valid sequence alignment* of sequence A and B, if there are no two lines $k, l \in \mathcal{L}$ such that k and l cross or touch each other [40]. Crossing or touching lines induce ordering conflicts in the alignment (see Fig. 5 for an illustration). We denote with the set $\mathcal{C}_L$ the collection of all maximal sets of mutually conflicting lines.

We extend the original graph $G_S = (V, L)$ by the edge set I to model the annotation of the input sequences in our graph. Consequently, we have *interaction edges* between vertices of the same sequence, i.e., an edge (v_i^A, v_j^A) representing the interaction between nucleotides i and j in sequence A. Figure 6 illustrates these definitions by means of an example. Note that at this stage gaps are not modelled in our formulation. Hence, we have to extend our model to incorporate gap penalties in our model.

Gap edges

The initial model containing only lines (the set L) and interaction edges (the set I) is augmented by a set of *gap edges* G, which represents gaps in the alignment. For sake of compactness, we just describe the gap edges of sequence A, the gap edges of sequence B are defined analogously: We have an edge e_{kl}^A from v_k^A to v_l^A with $k, l \in 1, \ldots, |s^A|$ representing the fact that no character of the substring $s_k^A \ldots s_l^A$ is aligned to any character of the sequence B, whereas s_{k-1}^A (if $k - 1 > 0$) and s_{l+1}^A (if $l + 1 \leq |s^B|$) are aligned with some characters in sequence B. We say that $v_k^A, \ldots, v_l^A$ are *spanned* by the gap edge e_{kl}^A. Figure 7 shows the graph extended by gap edges.

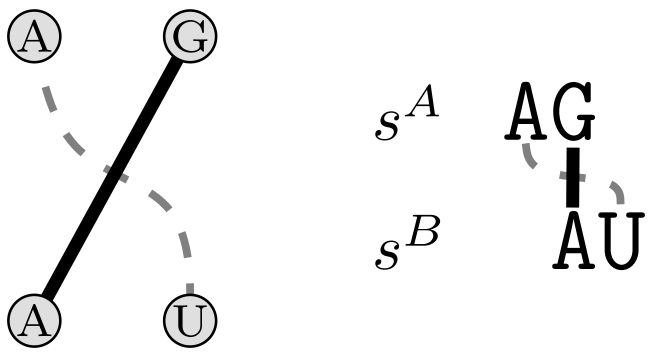


Figure 5
Crossing lines. Sequences $s^A = AG$ and $s^B = AU$ are given. The solid line between G and A represent the alignment of these two nucleotides. If we added the gray dashed line, this would induce an ordering conflict.

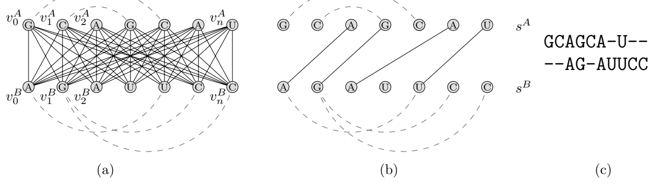


Figure 6
Initial graph model. (a) Initial graph model representing two annotated sequences s^A = GCAGCAU and s^B = AGAUUCC. Solid lines represent lines, dashed lines represent interaction edges. Please note that in this toy example minimum loop lengths constraints on the interaction edges are violated for sake of compactness of the illustration. Interactions (v_1^B, v_{n-1}^B) and (v_1^B, v_n^B) are in conflict with each other, (v_0^B, v_{n-2}^B) and (v_1^B, v_n^B) form a pseudoknot. Sequence s^A contains only nested interactions. (b) A subset of all possible lines is shown representing the alignment (c).

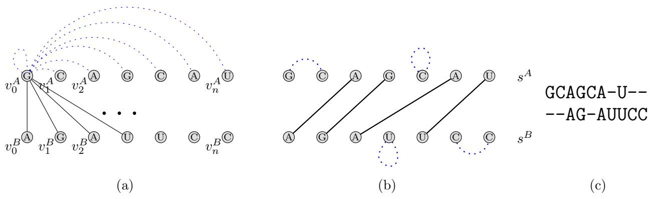


Figure 7
Graph model augmented gap edges. (a) Initial model additionally augmented with gap edges. The figure shows possible alignments edges and all gap edges starting from v_0^A (for sake of clarity, all other gap edges and interaction edges are not displayed). Note, however, that every node has outgoing gap edges to all other nodes in the sequence. The subset of lines and gap edges (b) corresponds to the alignment (c).

Two gap edges e_{kl}^A and $e_{mn}^A \in G$ are *in conflict* with each other if $\{k, ..., l + 1\} \cap \{m, ..., n\} \neq \emptyset$, that is, if they overlap or touch. This is intuitively clear, because we do not want to split a longer gap into two separate gaps: Consequently, there has to be at least one aligned character between any two realized gap edges. See Fig. 8 for an example. We denote with the set C_G the collection of all maximal sets of mutually conflicting gap edges. Finally, we define $G_{v_k^A \leftrightarrow v_l^A}$ as the set of gap edges that span the nodes $v_k^A,...,v_l^A$.

Interaction match

We call two interactions $(s_k^A, s_l^A) \in p^A$ and $(s_m^B, s_n^B) \in p^B$ an *interaction match* if there exist two alignment edges $a = (v_k^A, v_m^B)$ and $b = (v_l^A, v_n^B)$ that do not cross each other. We say that a subset $\mathcal{S} \subseteq L$ *realizes* the interaction match if $\{a, b\} \subseteq \mathcal{S}$. Interaction matches realized by a set $\mathcal{S}$ represent common interactions that are preserved by aligning the begin and end nucleotides of the interaction. Figure 9 illustrates the definitions.

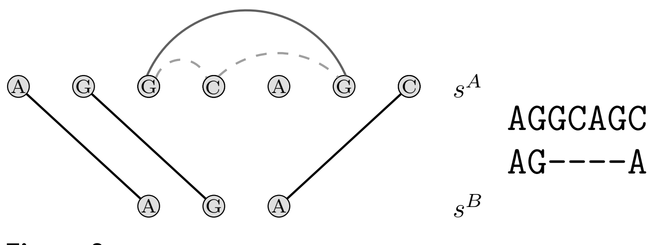


Figure 8
Crossing gaps. Gaps have to be realized by exactly one gap edge (in this example represented by the solid gray line), and cannot be split into two separate smaller gaps (the two dotted gap edges in this example).

Gapped structural trace

A triple ($\mathcal{L}$, $\mathcal{I}$, $\mathcal{G}$) with $\mathcal{L} \subseteq L$, $\mathcal{I} \subseteq I$, and $\mathcal{G} \subseteq G$ is called a valid *gapped structural trace* if and only if the following constraints are satisfied:

1. The vertices v_l^A and v_k^B of sequences A and B are either incident to exactly one alignment edge $e \in \mathcal{L}$ or spanned by a gap edge $g \in \mathcal{G}$. In other words, a nucleotide is either aligned or "aligned" to a gap.

2. A line l can realize at most one interaction match (l, m), because a nucleotide can pair with at most one other nucleotide in a valid RNA secondary structure.

3. There are no two lines k, $l \in L$ that cross or touch each other: Crossing lines induce ordering conflicts in the alignment, whereas touching lines imply that two different nucleotides are mapped to the same nucleotide in the other sequence.

4. There are no two gaps edges $e_{kl}^A, e_{mn}^A \in \mathcal{G}$ such that e_{kl}^A is in conflict with e_{mn}^A, and there are no two gaps edges $e_{kl}^B, e_{mn}^B \in \mathcal{G}$ such that e_{kl}^B is in conflict with e_{mn}^B.

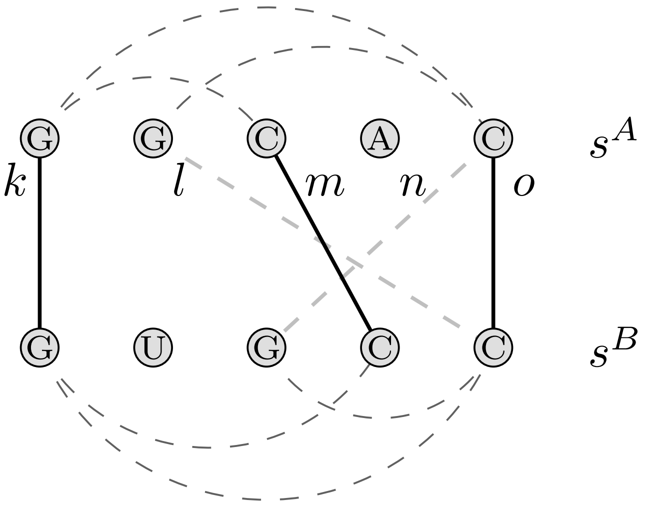


Figure 9
Interaction match. The pairs (k, m) and (k, o) are valid interaction matches. The pair (l, n), however, is not a valid interaction match since l and n cross each other.

Figure 10 visualizes these properties by showing a toy example for a gapped structural trace.

We assign weights w_l and w_{kl} for each line l and interaction match (k, l) that represents the benefit of realizing l or (k, l). By default, we set these scores along the lines of standard scoring methods, e.g., BLOSUM matrices for the weight of the lines, *base pair probabilities* [18] for the interaction match scores, or by using the RIBOSUM scoring matrices derived from alignments of ribosomal RNAs [47]. Our model, however, is not limited to standard scoring schemes. Since we can set each (sequence or structure) weight separately, the user can assign completely arbitrary scores to each line or interaction match which makes the incorporation of expert knowledge into the computation of structural alignments easy. Furthermore, we assign negative weights to gap edges a_{kl}^A with representing the gap penalty for aligning substring $s_k^A, \ldots, s_l^A$ with gap characters. Note that the model allows for arbitrary, position-dependent gap scoring.

Approaches for traditional sequence alignment aim at maximizing the score of edges in an alignment $\mathcal{L}$. Structural alignments, however, must also take the structural information encoded in the interaction edges into account. The problem of structurally aligning two annotated sequences (s^A, p^A) and (s^B, p^B) corresponds to finding an alignment such that the weight of the sequence part (i.e., the weight of selected lines plus gap penalties) plus the weight of the realized interaction matches is maximal. More formally, we seek to maximize $\sum_{l \in \mathcal{L}} w_l + \sum_{g \in \mathcal{G}} w_g + \sum_{(i,j) \in \mathcal{I}} w_{ij}$, where $(\mathcal{L}, \mathcal{G})$ represents an alignment with arbitrary gap costs, and $\mathcal{I}$ contains the interaction matches realized by $\mathcal{L}$. Observe that this graph-theoretical reformulation matches the problem statement given at the beginning of this section.

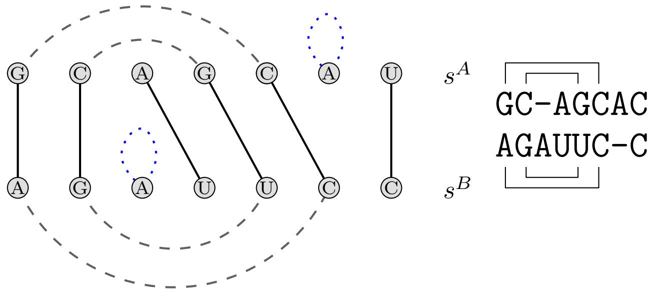


Figure 10
Valid gapped structural trace. Valid gapped structural trace: every vertex is incident to exactly one line or is spanned by a gap edge. There are no crossing lines, and every line is incident to at most one interaction match.

Biological aspects
The basic entities of our model are the alignment, interaction, and gap edges in the structural graph, which contribute to the objective function rather independently. Hence, one could argue that the model does not capture important features of RNA structures, like the incorporation of stacking energies or loop scores that depend on the actual size of the loop. We are aware of these limitations.

Nevertheless, the results of our computational experiments presented in Sect. 3 show that this approach yields high-quality structural alignments. In the pairwise case, our graph-based model is competitive with state-of-the-art approaches and develops its strength with an increasing number of sequences, outperforming all other programs that we tested (for details see Sect. 3). Additionally, the authors of [48] showed that models that do not capture stacking energies and loops are still competitive.

Beyond, our graph-based approach offers the possibility to change the model from nucleotides as the working entities to stems: Instead of taking single nucleotides as the vertices of the structural graph, we could search for *candidate stems* in the sequences and introduce a vertex for each half-stem. This would allow us to incorporate energy-based scoring into our model, which then, however, will have to be adapted to take into account overlapping stem candidates.

2.2 Integer linear program and Lagrangian relaxation
Given the graph-theoretical model it is straightforward to transform it to an *integer linear program* (ILP). We associate binary variables with each line, interaction match, and gap edge, and model the constraints of a valid gapped structural trace by adding inequalities to the linear program.

The handling of lines and gap edges is straightforward: We associate a x and z variable to each line and gap edge, respectively. We set $x_l = 1$ if and only if line $l \in L$ is part of the alignment $\mathcal{L}$, and $z_a = 1$ if and only if gap edge $a \in G$ is part of the alignment.

Interaction matches, however, are treated slightly differently: Instead of assigning an ILP variable to each interaction edge, we split an interaction match (l, m) into two separate *directed interaction matches* (l, m) and (m, l) that are detached from each other. A directed interaction match (l, m) is *realized* by the line set $\mathcal{L}$ if $l \in \mathcal{L}$. We then have $y_{lm} = 1$ if and only if the directed interaction match (l, m) is realized (note again that y_{lm} and y_{ml} are distinct variables). Figure 11 gives an illustration of the variable splitting. Note that this does not change the underlying model, it just makes the ILP formulation more convenient for further processing. Splitting interaction matches has first been proposed by Caprara and Lancia in the context of contact map overlap [42].

As described in Sect. 2.1, the sets L, I, and G refer to lines, interaction edges, and gap edges, and the sets $\mathcal{C}_L$ and $\mathcal{C}_G$ contain subsets of mutually conflicting lines or gap edges.

We then give the following ILP formulation for the gapped structural trace problem:

$$\max \sum_{l \in L} w_l x_l + \sum_{l \in L} \sum_{m \in L} w_{lm} y_{lm} + \sum_{g \in G} w_g z_g \tag{1}$$

$$\text{s.t.} \sum_{l \in C_L} x_l \le 1 \quad \forall C_L \in \mathcal{C}_L \tag{2}$$

$$\sum_{a \in C_G} z_a \le 1 \quad \forall C_G \in \mathcal{C}_G \tag{3}$$

$$x_l + \sum_{a \in G_{s(l) \leftrightarrow s(l)}} z_a = 1 \quad \forall l \in L \tag{4}$$

$$x_l + \sum_{a \in G_{t(l) \leftrightarrow t(l)}} z_a = 1 \quad \forall l \in L \tag{5}$$

$$\sum_{m \in L} y_{lm} \le x_l \quad \forall l \in L \tag{6}$$

$$y_{lm} = y_{ml} \quad \forall l, m \in L \tag{7}$$

$$x \in \{0, 1\}^L \, y \in \{0, 1\}^{L \times L} z \in \{0, 1\}^G \tag{8}$$

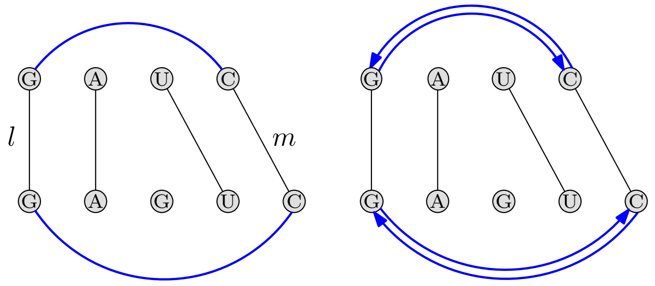


Figure 11
Splitting of interaction matches. One interaction match is split into two *directed* interaction matches.

Lemma 2.1 (Proof in [16])
A feasible solution to the ILP (1)–(8) *corresponds to a valid gapped structural trace of weight equal to the objective function and vice versa.*

Observe that constraints (2)–(6) exactly correspond to the properties of a gapped structural trace as described in Sect. 2.1.

In [49] the authors show that the problem of computing an optimal gapped structural trace is already NP-hard, even without considering gap costs. Hence, we cannot hope to find an optimal solution to the problem in polynomial time.

Commonly used mathematical programming techniques for NP-hard problems therefore resort to various *relaxation techniques* that are the basis for further processing. A relaxation results from the removal of constraints from the original ILP formulation, and is often solvable in polynomial time. A popular relaxation is the so called *LP relaxation* where the integrality constraints on the variables are dropped, yielding a standard linear program, for which solutions can be found efficiently.

Another possible relaxation technique is *Lagrangian relaxation*: Instead of just dropping certain inequalities, we move them to the objective function, associated with a penalty term that becomes active if the dropped constraint is violated. By iteratively adapting those penalty terms using, for instance, *subgradient optimization*, we get better solutions with each iteration. A crucial parameter is therefore the number of iterations that we perform: the higher the number, the more likely it is to end up with an optimal or near-optimal solution.

Inspired by the successful approach of Lancia and Caprara for the contact map overlap problem, we consider the relaxation resulting from moving constraint (7) into the objective function.

Lemma 2.2 (Proof in [16])
The relaxed problem is equivalent to the pairwise sequence alignment problem with arbitrary gap costs.

2.3 Algorithms for the pairwise and multiple case

Our algorithm for the pairwise RNA structural alignment problem consists of iteratively solving the primary sequence alignment problem associated with the relaxation. The penalization of the relaxed inequality is reflected in an adapted scoring matrix for the primary alignment. Intuitively, these weights incorporate also the structural information. In each iteration we get a new lower bound for the problem by analyzing the primary sequence alignments and inferring the best *structural completion* of this alignment. In fact, this corresponds to solving a maximum weighted matching problem in a general graph. For details see [16]. In the course of the algorithm, these solutions get better and better. Furthermore, the value of the relaxation itself constitutes an upper bound on the problem, which decreases with an increasing number of iterations. When these bounds coincide, we have provably found an optimal solution, otherwise, we get near-optimal solutions with a quality guarantee. Assuming an upper bound on the number of interaction matches per line, which is typically the case with base pair probability matrices of RNA sequences, we get a running time of $O(n^2)$ for each Lagrangian iteration. Since we fix the number of iterations, this leads to an overall time complexity of $O(n^2)$.

For the multiple case, similar in spirit to the MARNA software, we combine our pairwise method with the popular progressive alignment software T-COFFEE [50]. Progressive methods build multiple alignments from pairwise alignments. The pairwise distances are usually used to compute a guide tree which in turn determines the order in which the sequences are aligned to the evolving multiple alignment.

Progressive approaches often suffer from their sensitivity to the order in which the sequences are chosen during the alignment process. T-COFFEE reduces this effect by making use of local alignment information from *all* pairwise sequence alignments during its progressive alignment phase. We supply such local alignment information based on all-against-all structural alignments computed with our pairwise approach, assigning a high score to conserved interaction matches. The structural information is subsequently passed on to T-COFFEE that computes a multiple alignment, taking into account the additional structural information.

3 Experiments

The basis of our computational experiments is the recently published benchmark set BRALIBASE 2.1 [51]. We compared our program to four other alignment programs (MARNA, FOLDALIGNM, MAFFT, and STRAL) using two established measures for the quality of structural alignments (Compalign and SCI score). We performed all experiments with default parameters.

3.1 BRAliBase 2.1

We chose this data set, which is available from [52], as our test set, since it covers a greater range of typical noncoding-RNA families than the original BRALIBASE data set [12]. BRALIBASE 2.1 contains 36 different RNA families, ranging from approximately 26 nucleotides long Histone 3'UTR stem-loop motifs to approximately 300 nucleotides long eukaryotic SRP RNAs. See [51] for a detailed listing of all instances. BRALIBASE 2.1 reference alignments are based on manually curated seed alignments of the *Rfam 7.0* database [53]. Out of the pool of all ncRNA families that have more than 50 sequences in their seed alignment, either 2, 3, 5, 7, 10 or 15 sequences were randomly drawn considering constraints on the sequences (e.g., average pairwise sequence identity or structural conservation). These subsets of the original seed alignments form the instances of BRALIBASE: in the following we stick to the BRALIBASE naming convention and refer to the sets of instances by $k2$, $k3$, $k5$, $k7$, $k10$, and $k15$, depending on the number of sequences per instance.

3.2 Compalign and SCI

We use two different scores to measure the quality of the computed alignments: the *Compalign* value codes the degree of similarity to a given reference alignment as given by the percentage of columns that are identically aligned as in the reference alignment. A value of 1 states that the reference and test alignment are the same, whereas 0 denotes that no column was correctly aligned with respect to the reference alignment.

The second score is the so called *structural conservation index* [54] (or SCI in short). The SCI basically gives the degree of conservation of a consensus structure induced by a multiple alignment in relation to the *minimum free energy* structure of each sequence (to be more precise, not the actual structures are compared but their respective energy values). A SCI value of ≈ 1 indicates very high structural conservation, whereas a value around 0 indicates no structural conservation at all. Note that the SCI score can be greater than 1, because covariance information is additionally rewarded in the computation.

We have used the programs *compalignp* and *scif* to compute the Compalign and SCI score. Both tools are freely available from the BRAliBase website.

3.3 Other structural alignment programs

We implemented our approach called LARA in C++ within the LISA framework. LISA (*Li*brary of *S*tructural *A*lignment algorithms) contains various methods for aligning protein and RNA structures as well as biological networks.

Furthermore, we selected several other multiple structural alignment programs to compare the results. We used MARNA [26] (available from [55]) using an ensemble of three suboptimal structures as its input, STRAL [24] (a sequence based algorithm incorporating McCaskill's base pair probabilities, available from [56]), and a reimplementation of the PMCOMP approach called FOLDALIGNM [32] (a banded variant of Sankoff's algorithm that aligns base pair probability matrices, available from [57]).

Furthermore, to compare the performance of the structure-based alignment programs to purely sequence-based ones, we performed the same tests with MAFFT [58], a recent multiple sequence alignment program which is available from [59]. We want to emphasize that we did not perform any parameter tuning for any program (this includes LARA), i.e., we downloaded the programs from the respective websites and performed the computations out of the box without specifying any optional parameters.

Since earlier studies [12,51] showed that structural alignments only contribute an additional benefit – compared to sequence-based approaches – if the pairwise sequence identity drops below ≈ 50 – 60%, we restricted the test set to instances of low homology, i.e., instances having a pairwise sequence identity below 50%.

3.4 LaRA

A scoring system for structural alignments has to provide two different kinds of scores: scores for the sequence and the structure part (in case of LARA, these correspond to weights for the alignment and interaction edges, respectively). Since the structure is considered to contain the necessary information for "correct" alignments, we have to make sure that the structure scores contribute the major part to the overall score.

We do not generate the complete annotation for our input sequence, that is, an interaction edge between every possible interaction, but restrict interaction edges to those having base pair probabilities larger than a threshold $p_{\min}$. For our experiments we resorted to a value of 0.003, similar as in PMCOMP. The impact of different $p_{\min}$ values is twofold: First, the lower the value is, the higher the structure scores are. Secondly, a high $p_{\min}$ value leads to a sparser structure graph.

For the scoring of the edges, LARA provides two different schemes: First, a scoring system based on base pair probability matrices (BPP scoring in short) that rescales the scores in spirit of PMCOMP. More precisely, given the probability p_{ij} that nucleotide i and j pair, the actual score s_{ij} for the structural interaction between i and j is given by

$$s_{ij} = \lg\left(\frac{p_{ij}}{p_{\min}}\right)$$

where lg is the natural logarithm. For the sequence scoring, we take the entries from the RIBOSUM matrices [47] as the actual sequence scores (that is the scores for pairs of nucleotides) and multiply them by a user-specific adjustment factor τ. The default value for τ is 0.05, leading to a small sequence score contribution to the overall score. If one knows, however, that sequence is equally or more important than the structure (e.g., in case of riboswitches), one simply has to increase the value of τ.

The second scheme employs the RIBOSUM scoring matrices both for sequence and structure scoring: these matrices are based on given alignments of ribosomal RNAs from which log-odds scores were derived. They provide both sequence and structure scores, without rescaling the scores.

The second crucial LARA parameter is the number of iterations: the more iterations LARA computes, the more often the penalty terms are adapted (yielding better alignments). As one can see in Fig. 12 the number of iterations influences the quality of the computed alignment while the running time increases linearly with the number of iterations. In our experiments we set the number of iterations to 500.

The scoring of gap edges follows the scheme of *affine gap costs* with an gap open and extension penalty of -6 and -2, respectively.

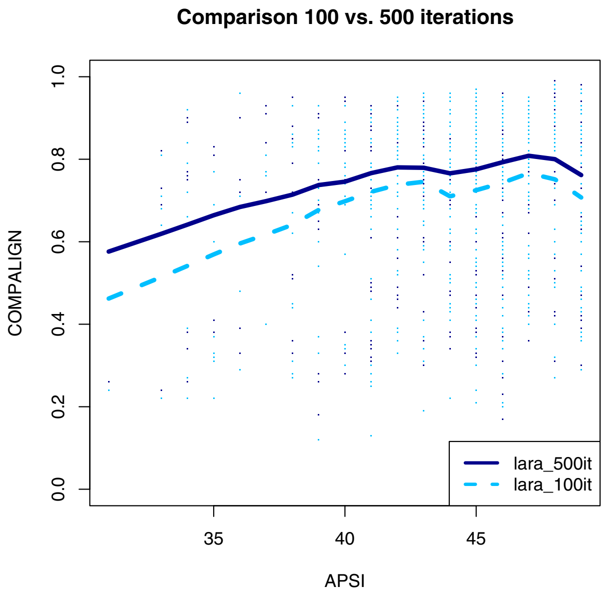


Figure 12
Different number of iterations. Comparison of all *k*10 instances of low homology between LARA running 100 or 500 iterations. Each dot correponds to one problem instance, the thick lines were computed using Lowess regression. The x-axis gives the average pairwise sequence identity (APSI). The y-axis codes the Compalign score.

Score vs. alignment accuracy

We were interested to what extent the accuracy of our alignments correlates with the actual BPP score that we computed. Since the score depends on the length of the input sequences, we normalized the score with respect to the number of paired bases in the *minimum free energy* structure. Note that we did not use the actual structure, but the number of base pairs in the structure to get a rough estimate of how many pairings we expect in the structure. Then, let $\hat{p}$ and n be the average score and the number of base pairs in the MFE structure, then the *base-pair normalized score* is given by $\hat{p}/n$. The left side of Fig. 13 shows the results for all 189 *k*10 instances with an average pairwise sequence identity less than 50%. The great majority of instances behaves as expected: the higher the bp-score is, the better is the corresponding Compalign score: There is, however, a group of 10 outliers (represented by the red boxes). Although they have a high bp-score (greater than 10.0), the alignment accuracy is bad: it turned out that these 10 instances are all SECIS-elements, indicating that the BPP scoring scheme is not appropriate for this group.

Furthermore, we assumed that there should a correlation between the actual performance of our algorithm and, again, the quality of our alignments: Remember that each Lagrange iteration results in a new valid solution and a new upper bound for the problem instance. Dividing the value of the highest lower bound by the value of the lowest upper bound gives an *optimality ratio*, i.e., a measure of how close the best solution is to an optimal one. Assuming an inverse correlation between the gap between lower and upper bound and the quality of the alignment, we again took all *k*10 BRALIBASE instances of low pairwise sequence identity and computed the arithmetic mean of the optimality ratios of all pairwise alignments. The right side of Fig. 13 shows the plot for all 189 *k*10 instances with a sequence similarity lower than 50%. Most of the instances behave as expected: the higher the average optimality ratio is, the closer is the computed alignment to the reference alignment (and vice versa). There is, however, a group of 19 instances that behave differently (marked as red boxes in Fig. 13): Although their average optimality ratio is high (> 0.7), the corresponding Compalign value is rather low compared to instances of a similar average optimality ratio. A closer inspection revealed that all instances of the upper left corner (that is instances having a Compalign value lower than 0.65 and an average optimality ratio of greater than 0.7, represented by red boxes in Fig. 13) comprises almost all instances of either bacterial SRP RNAs or SECIS elements (just one SRP RNA instance is not among the 19 instances). We therefore increased the number of iterations for one SECIS instance to see whether this would positively influence the quality of the alignment. By setting the number of iterations to 500, 1000, and 2000 we got average optimality ratios of 0.83, 0.85, and 0.87, by simultaneously yielding Compalign values of 0.39, 0.38, and 0.36, respectively. Obviously, the better the computed alignments in terms of the optimality ratio are, the worse they got with respect to the reference alignment.

Consequently, for the outlier instances described above, we changed the scoring from BPP to RIBOSUM scores. Figure 14 shows the change in terms of the Compalign score and optimality ratio for the 19 outlier instances: 16 instances had better Compalign scores by using the RIBO-

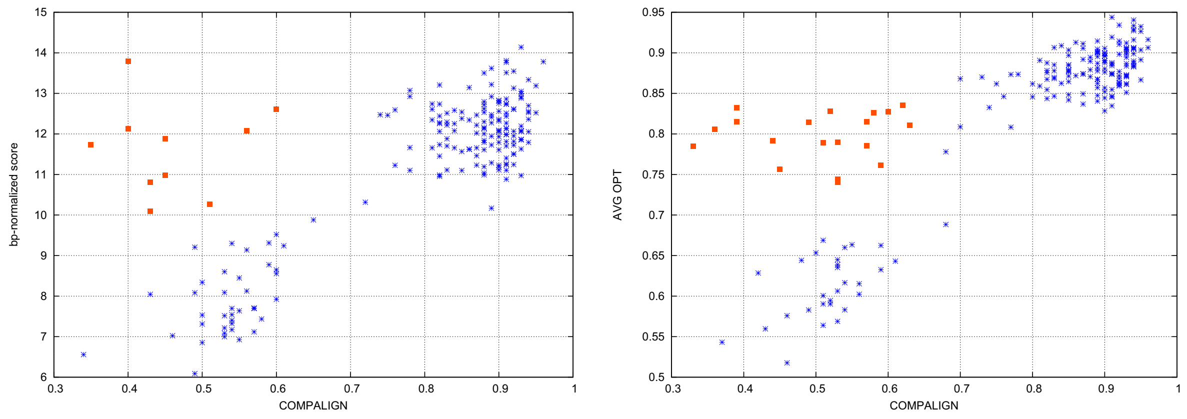


Figure 13
Score vs. alignment accuracy. All 189 BRALIBASE *k*10 instances of low pairwise sequence identity where each cross or box corresponds to one instance: The x-axis gives the Compalign score. The y-axis codes either a structure-normalized score (left side), or the optimality ratio (right side). The red boxes mark the outliers.

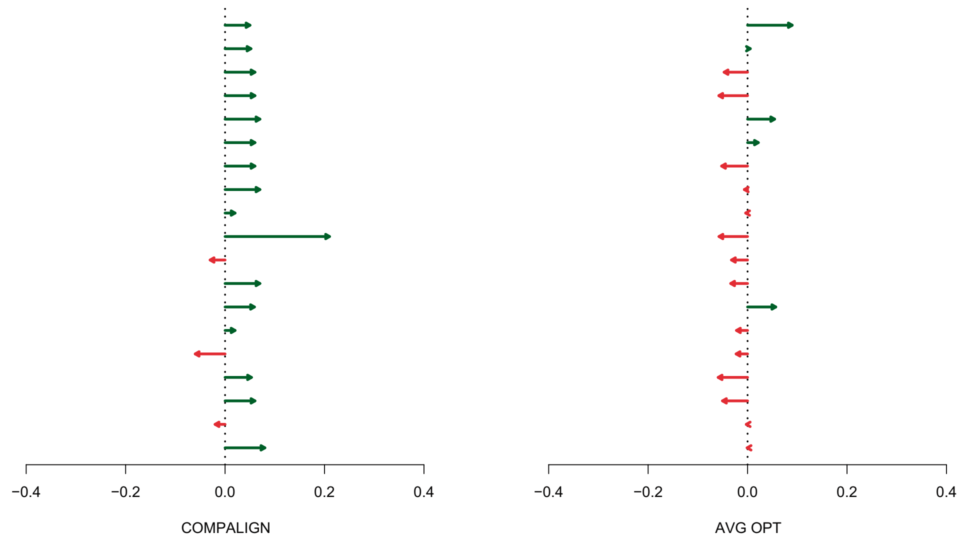


Figure 14
Changing the scoring from BPP to RIBOSUM. Change of the Compalign score and optimality ratio after changing the scoring from BPP to RIBOSUM matrices for the 19 outlier instances.

SUM scoring, whereas the optimality ratio decreased in the majority of instances.

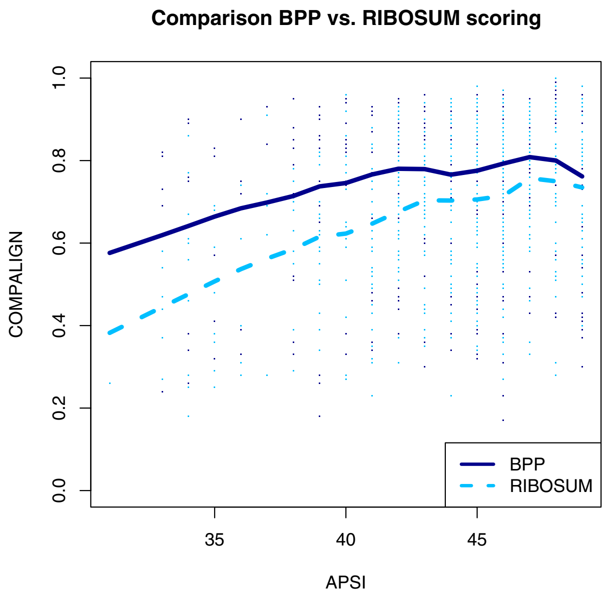


Figure 15
BPP vs. RIBOSUM scoring. Comparison between base pair probability (BPP) and RIBOSUM scoring. The x-axis gives the average pairwise sequence identity (APSI). The y-axis codes the Compalign score.

In general, however, our experiments showed that RIBOSUM scoring is not superior to BPP scoring (at least for the BRALIBASE benchmark and LARA): Figure 15 shows a comparison of all low homology $k5$ instances using either base pair probability matrices or RIBOSUM scoring, and it is obvious that base pair probability scoring yields better results on these input instances.

3.5 Comparison with other programs

As described in Sect. 3.2 we used two different scores to assess the quality of the computed alignments: the Compalign (the degree of similarity between the test alignment to a given reference alignment) and the SCI score (the degree of structural conservation induced by the test alignment).

FOLDALIGNM performs an alignment and clustering of the input sequences at the same time: in some instances, FOLDALIGNM splits the input sequences into two clusters. Since the scores that we use depend on the number of input sequences, we dropped those FOLDALIGNM alignments that did not contain all sequences in the final alignment: This leads to 43, 30, 11, 15, 19, and 6 instances that we did not consider in case of $k2$, $k3$, $k5$, $k7$, $k10$, and $k15$ instances.

In Fig. 16 we show the results of our experiments broken down to the different input classes (either $k2$, $k3$, $k5$, $k7$, $k10$, or $k15$). These graphics have the average pairwise sequence identity and the Compalign score as their x- and y-axis, respectively. The reference alignments therefore

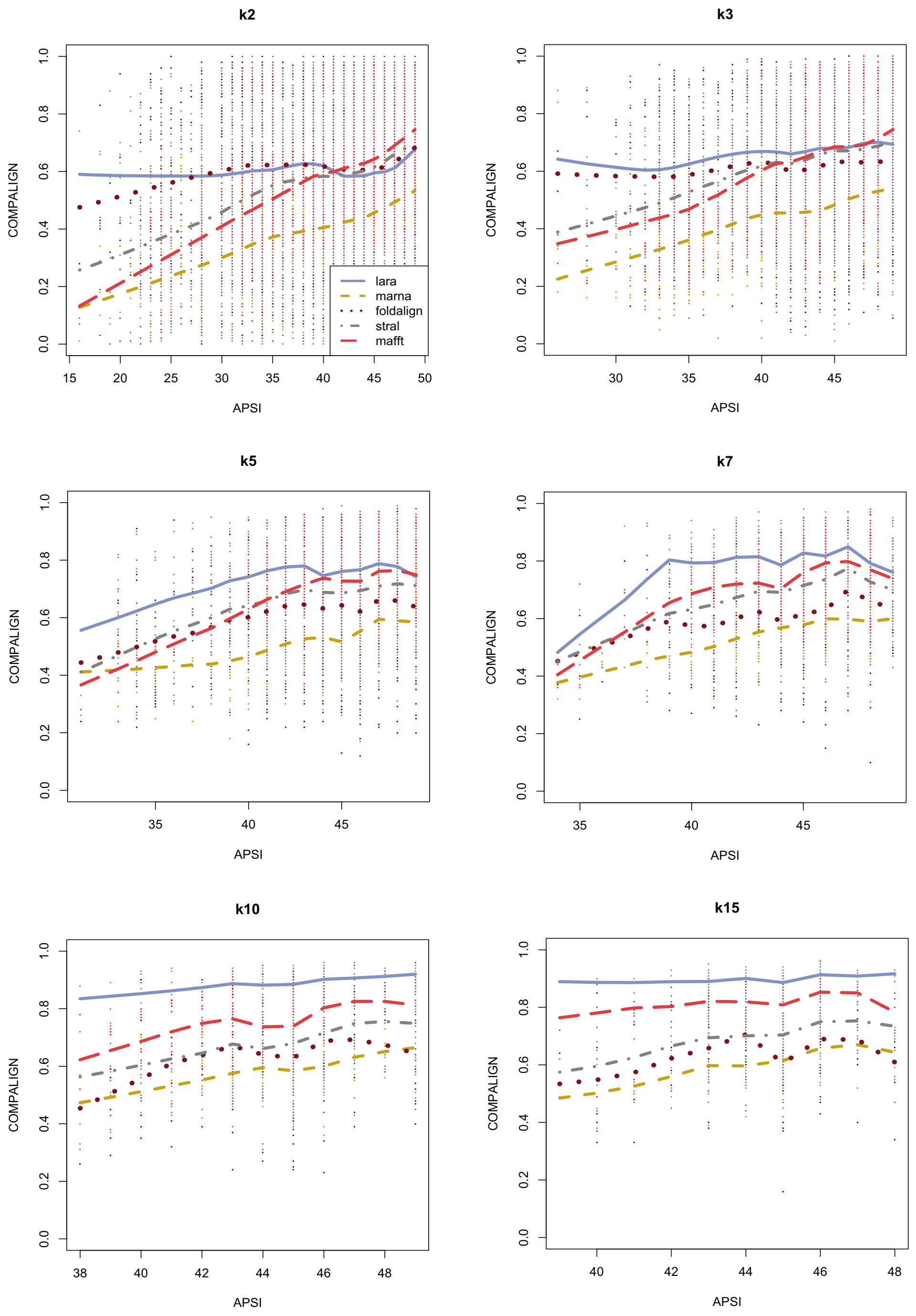


Figure 16

Compalign results on low homology instances. Results on all low homology instances containing 2 (upper left), 3 (upper right), 5 (middle left), 7 (middle right), 10 (lower left), and 15 (lower right) instances from the BRALIBASE benchmark set. The *x*- and *y*-axes give the average pairwise sequence identity (APSI) and the Compalign score. The legend from the upper left plot applies to the other plots as well. Mind that the different APSI-ranges in the six plots are a result of the BRALIBASE benchmark set: there are, for example, no *k*15 instances in BRALIBASE below 38%.

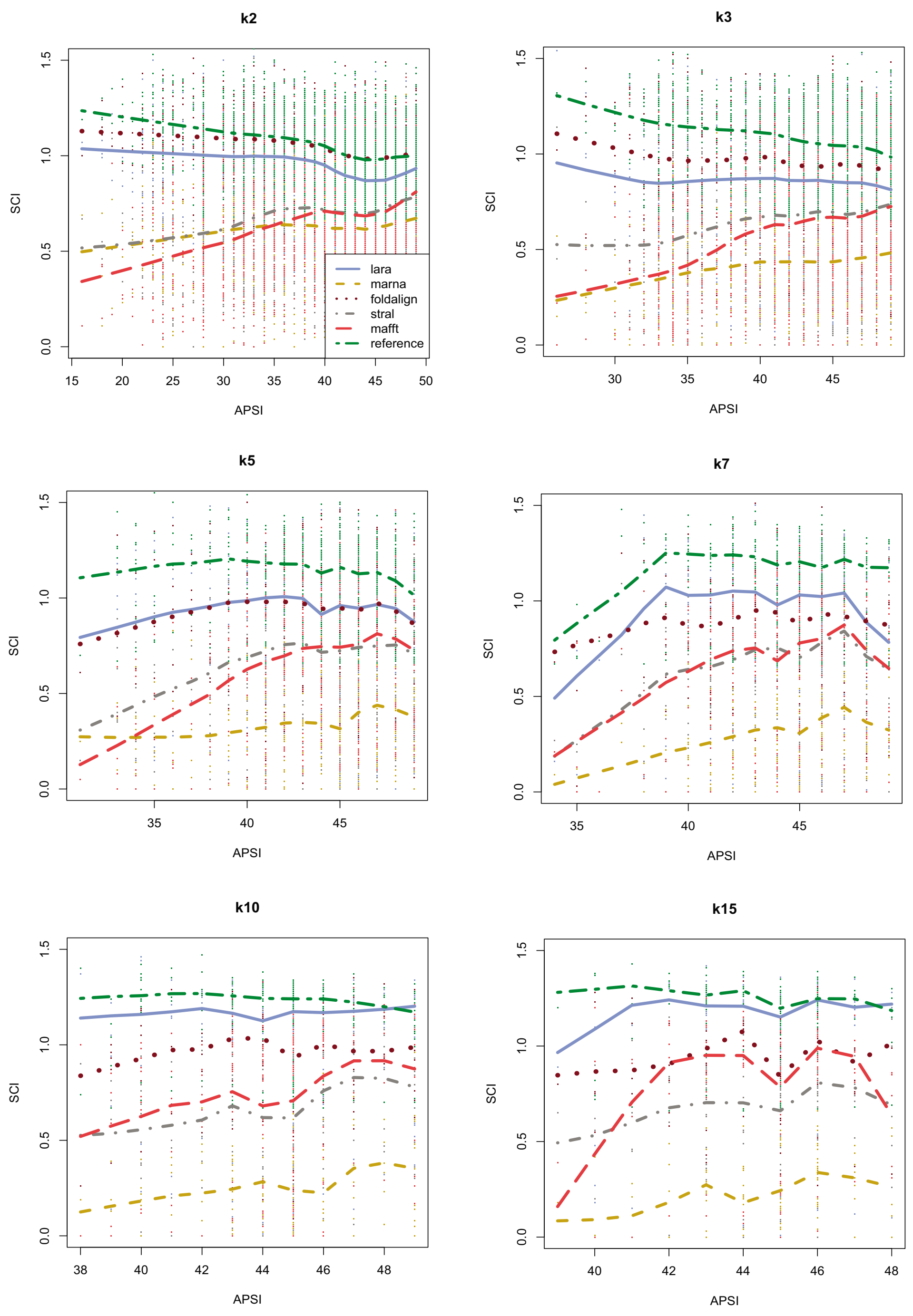


Figure 17
SCI results on low homology instances. Results on all low homology instances containing 2 (upper left), 3 (upper right), 5 (middle left), 7 (middle right), 10 (lower left), and 15 (lower right) instances from the BRALIBASE benchmark set. The *y*-axis gives the SCI score. The legend from the upper left plot applies to the other plots as well. Mind that the different APSI-ranges in the six plots are a result of the BRALIBASE benchmark set: there are, for example, no *k*15 instances in BRALIBASE below 38%.

correspond to horizontal lines at a Compalign score of 1.0.

We have made several observations: First of all, in the pairwise case (i.e., the $k2$ instances) LARA has a similar performance as the Sankoff variant FOLDALIGNM up to a sequence identity of ≈ 42%. For the range of ≈ 42 – 50% all programs (even sequence-based MAFFT) have comparable performance (except for MARNA). With an increasing number of input sequences per instance, especially for the $k10$ and $k15$ sequences, the results change tremendously: LARA outperforms the other programs, yielding average Compalign scores of ≈ 90%, whereas the other structure-based alignment programs have scores around ≈ 55 – 75%. This is quite remarkable, especially considering that FOLDALIGNM and LARA show a similar performance in the pairwise case: FOLDALIGNM, however, computes multiple alignments in a progressive fashion, whereas LARA computes *all* pairwise alignments and leaves it to T-COFFEE to compute an alignment that is highly consistent with all pairwise alignments. With an increasing number of input sequences, the consistency-based approach generates better alignments than the progressive methods (at least in the case of our experimental setup).

Another astonishing observation is the performance of MAFFT, a purely sequence-based program: the $k2$ and $k3$ instances show a comparable performance for instances above ≈ 42%, which is already surprising. With a growing number of input instances, the performance of MAFFT becomes even better: in case of 15 input instances, the program yields – on average – the second best results (behind LARA), outperforming even FOLDALIGNM and STRAL, which incorporate structural information. It has to be investigated whether the creation of the benchmark set has to be revisited, because these plots clearly contradict the hypothesis that sequence-based programs yields significantly worse results for input instances of a pairwise sequence identity below 50%.

In Fig. 17 we show the results with respect to the SCI score (remember that the SCI is a measure for the structural conservation of an alignment). The general trend is the same as in Fig. 16. In the pairwise case, the LARA curve has the same shape as the reference curve, but shifted to the bottom by about 0.1. FOLDALIGNM yields the best approximation to the reference line, having almost the same performance for instances with an APSI greater than 30%. With an increasing number of input sequences, the situation changes: from $k5$ on LARA generates the best approximation to the reference line, with FOLDALIGNM being the second best program. Taking a look at the various result plots puts the extraordinary performance of MAFFT into perspective regarding the $k10$ and $k15$ input sets.

Table 2: Failed instances. Unsolved instances within a time limit of 20 minutes.

Program	*k*2	*k*3	*k*5	*k*7	*k*10	*k*15
LARA	0	0	0	0	0	0
FOLDALIGNM	0	0	0	0	0	0
STRAL	0	0	0	0	0	0
MARNA	0	49	23	17	12	6
MAFFT	0	0	0	0	0	0

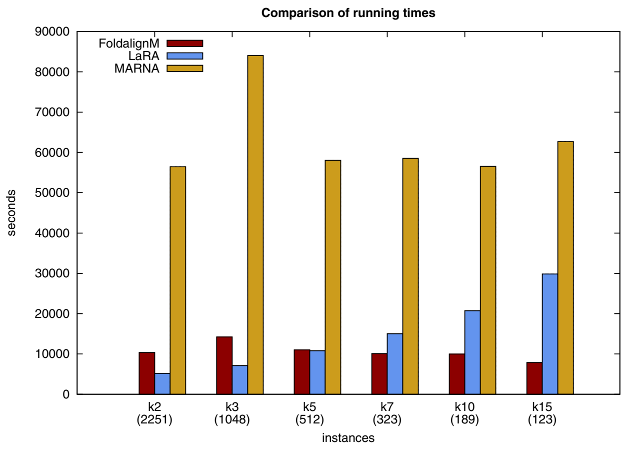


Figure 18
Running times of sequence-structure alignment programs. The plot shows a comparison of the running times between the structural programs tested. With an increasing number of input sequences, a progressive alignment strategy pays off compared to the computation of all pairwise alignments. The *x*-axis lists the different input instances, either containing 2, 3, 5, 7, 10, or 15 sequences (denoted by *k*2, *k*3, *k*5, *k*7, *k*10, and *k*15, repetively). The numbers in brackets denote the number of instances per input class. The y-axis gives the number gives the computation time in seconds.

Comparison of running times

We compared the programs tested on the same computing server with an Intel Xeon CPU running at 3.2 GHz, 3.5 GB RAM, and Linux kernel version 2.6.16. It turned out that memory requirement was not an issue, but the computation time instead: especially MARNA scales in $O(n^4)$, which makes the alignment of longer sequences (for example the SRP instances of BRALIBASE) rather time-consuming. This, however, is not the case with LARA and FOLDALIGNM, since these two programs have running times in $O(n^2)$. To evaluate the time consumption within reasonable time, we therefore set a time limit of 20 minutes per instance: If the computation was not finished within 20 minutes, the process was killed and we took 20 minutes as the actual running time. In Table 2 we list the number of instances that the corresponding program was not able to align within 20 minutes.

We were especially interested in how the running times of the programs that use structure information scaled with respect to the number of the input sequences: FOLDALIGNM is a progressive approach which computes (n - 1) pairwise alignments given n input sequences. MARNA and LARA, however, compute all $\frac{n(n-1)}{2}$ pairwise alignments. Figure 18 shows the execution time of all five programs on all $k2$, $k3$, $k5$, $k7$, $k10$, and $k15$ instances. As one can see, with an increasing number of input sequences, a progressive alignment strategy pays off compared to the computation of all pairwise alignments.

4 Conclusion

We have presented a novel method for computing high-quality pairwise structural RNA alignments. We approach the original problem using a flexible graph-based model, which naturally deals with pseudoknots.

We find solutions in our model by means of an integer linear programming formulation and the Lagrangian relaxation technique. For the multiple case, we compute all-against-all pairwise solutions and pass this information to T-COFFEE, a progressive alignment algorithm.

Our extensive computational experiments on a large set of benchmark alignments show that LARA, the implementation of our algorithm, is competitive with state-of-the art tools and outperforms alternative approaches with an increasing number of input sequences. The difference to other programs gets larger the more sequences that have to be aligned. In this context, we also find the performance of MAFFT, a purely sequence-based program, remarkable. MAFFT comes closer to manually curated reference alignments than all other structure-specific tools besides LARA for alignments of more than ten sequences.

Our plans for the future include a local version of our alignment algorithm. Furthermore, we are currently implementing an exact branch-and-bound framework around the Lagrangian approach and will develop a stem-based variant of LARA. Furthermore, the openness to pseudoknots is the main advantage of LARA over alternative approaches, and we plan to adapt our method to produce high-quality alignments of pseudoknotted structures.

Availability and requirements

LARA (Lagrangian relaxed alignments) is part of the C++ library LiSA and is freely available for academic purposes from http://www.planet-lisa.net. The binary runs under the Linux operating system.

All alignments that we computed and the scripts for generating the plots are also available from http://www.planet-lisa.net/.

Authors' contributions

MB, GWK, and KR developed the integer linear program model. MB coded the program and carried out the computational experiments. MB and GWK drafted the manuscript, GWK and KR coordinated the research. All authors read and approved the final manuscript.

Additional material

Additional file 1

This is a compressed tar file containing all alignments used in the experimental study.
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-8-271-S1.gz]

Acknowledgements

This work has been partly supported by the DFG grant KL 1390/2-1. MB is additionally supported by the International Max Planck Research School for Computational Biology and Scientific Computing. The authors thank the anonymous referees for valuable comments.

References

1. Lagos-Quintana M, Rauhut R, Lendeckel W, Tuschl T: **Identification of novel genes coding for small expressed RNAs.** *Science* 2001, **294(5543):**853-8.
2. Lau NC, Lim LP, Weinstein EG, Bartel DP: **An abundant class of tiny RNAs with probable regulatory roles in *Caenorhabditis elegans*.** *Science* 2001, **294(5543):**858-62.
3. Samarsky DA, Fournier MJ: **A comprehensive database for the small nucleolar RNAs from Saccharomyces cerevisiae.** *Nucleic Acids Res* 1999, **27:**161-164.
4. Gorodkin J, Knudsen B, Zwieb C, Samuelsson T: **SRPDB (Signal Recognition Particle Database).** *Nucleic Acids Res* 2001, **29:**169-170.
5. Kim VN: **Small RNAs just got bigger: Piwi-interacting RNAs (piRNAs) in mammalian testes.** *Genes Dev* 2006, **20(15):**1993-1997.
6. Mattick JS: **The functional genomics of noncoding RNA.** *Science* 2005, **309(5740):**1527-1528.
7. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic Local Alignment Search Tool.** *J Mol Biol* 1990, **215:**403-410.
8. Smith TF, Waterman MS: **Identification of Common Molecular Subsequences.** *J Mol Biol* 1981, **147:**195-197.
9. Zhang S, Haas B, Eskin E, Bafna V: **Searching Genomes for Noncoding RNA Using FastR.** *IEEE/ACM Trans Comput Biol Bioinform* 2005, **2(4):**366-379.
10. Wolf M, Achtziger M, Schultz J, Dandekar T, Müller T: **Homology modeling revealed more than 20,000 rRNA internal transcribed spacer 2 (ITS2) secondary structures.** *RNA* 2005, **11(11):**1616-1623.

11. Hofacker IL, Fekete M, Stadler PF: **Secondary Structure Prediction for Aligned RNA Sequences.** *J Mol Biol* 2002, **319:**1059-1066.
12. Gardner P, Wilm A, Washietl S: **A benchmark of multiple sequence alignment programs upon structural RNAs.** *Nucl Acids Res* 2005, **33(8):**2433-2439.
13. Bauer M, Klau GW: **Structural Alignment of Two RNA Sequences with Lagrangian Relaxation.** In *Proc ISAAC'04, Volume 3341 of LNCS* Springer-Verlag; 2004:113-125.
14. Bauer M, Klau GW, Reinert K: **Multiple Structural RNA Alignment with Lagrangian Relaxation.** *Proc WABI'05, Volume 3692 of LNBI* 2005:303-314.
15. Bauer M, Klau GW, Reinert K: **Fast and Accurate Structural RNA Alignment by Progressive Lagrangian Relaxation.** *Proc CompLife'05, Volume 3695 of LNBI* 2005:217-228.
16. Bauer M, Klau GW, Reinert K: **An Exact Mathematical Programming Approach to Multiple RNA Sequence-Structure Alignment.** *Tech Rep TR-B-07-07* 2007 [http://www.inf.fu-berlin.de/inst/pubs]. Dept. of Mathematics and Computer Science, Free University Berlin [Submitted to **Algorithmic Operations Research**]
17. Freyhult EK, Bollback JP, Gardner PP: **Exploring genomic dark matter: A critical assessment of the performance of homology search methods on noncoding RNA.** *Genome Research* 2007, **17:**117-125.
18. McCaskill JS: **The Equilibrium Partition Function and Base Pair Binding Probabilities for RNA Secondary Structure.** *Biopolymers* 1990, **29:**1105-1119.
19. Zhang K, Shasha D: **Simple fast algorithms for the editing distance between trees and related problems.** *SIAM J Comput* 1989, **18(6):**1245-1262.
20. Jiang T, Wang J, Zhang K: **Alignment of Trees – An Alternative to Tree Edit.** *Theor Comput Sci* 1995, **143:**137-148.
21. Höchsmann M, Töller T, Giegerich R, Kurtz S: **Local Similarity in RNA Secondary Structures.** *Proc IEEE Comput Soc Bioinform Conf* 2003, **2:**159-168.
22. Bafna V, Muthukrishnan S, Ravi R: **Computing similarity between RNA strings.** In *Proc of CPM'95, no. 937 in LNCS* Springer; 1995:1-16.
23. Eddy SR: **A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure.** *BMC Bioinformatics* 2002, **3:**18.
24. Dalli D, Wilm A, Mainz I, Steger G: **STRAL: progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time.** *Bioinformatics* 2006, **22(13):**1593-1599.
25. Jiang T, Lin GH, Ma B, Zhang K: **A general edit distance between RNA structures.** *J Comput Biol* 2002, **9(2):**371-388.
26. Siebert S, Backofen R: **MARNA: Multiple alignment and consensus structure prediction of RNAs based on sequence structure comparisons.** *Bioinformatics* 2005, **21(16):**3352-3359.
27. Sankoff D: **Simultaneous solution of the RNA folding, alignment, and proto-sequence problems.** *SIAM J Appl Math* 1985, **45:**810-825.
28. Mathews DH, Turner DH: **Dynalign: An Algorithm for Finding Secondary Structures Common to Two RNA Sequences.** *J Mol Biol* 2002, **317:**191-203.
29. Mathews D: **Predicting a set of minimal free energy RNA secondary structures common to two sequences.** *Bioinformatics* 2005, **21:**2246-2253.
30. Hull Havgaard J, Lyngsø R, Stormo G, Gorodkin J: **Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%.** *Bioinformatics* 2005, **21:**1815-1824.
31. Hofacker IL, Bernhart SHF, Stadler PF: **Alignment of RNABase Pairing Probability Matrices.** *Bioinformatics* 2004, **20:**2222-2227.
32. Torarinsson E, Havgaard JH, Gorodkin J: **Multiple structural alignment and clustering of RNA sequences.** *Bioinformatics* 2007, **23(8):**926-932.
33. Will S, Reiche K, Hofacker IL, Stadler PF, Backofen R: **Inferring Noncoding RNA Families and Classes by Means of Genome-Scale Structure-Based Clustering.** *PLoS Comput Biol* 2007, **3(4**e65 [http://dx.doi.org/10.1371/journal.pcbi.0030065].
34. Eddy SP, Durbin R: **RNA sequence analysis using covariance models.** *Nucleic Acids Res* 1994, **22(11):**2079-2088.
35. Holmes I: **A probabilistic model for the evolution of RNA structure.** *BMC Bioinformatics* 2004, **5:**166.
36. Holmes I: **Accelerated probabilistic inference of RNAstructure evolution.** *BMC Bioinformatics* 2004, **5:**73.
37. Dowell R, Eddy S: **Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints.** *BMC Bioinformatics* 2006, **7:**400.
38. Sato K, Sakakibara Y: **RNA secondary structural alignment with conditional random fields.** *Bioinformatics* 2005, **21(suppl 2):**237-242.
39. Sakakibara Y: **Pair hidden Markov models on tree structures.** *Bioinformatics* 2003, **19:**i232-240.
40. Kececioglu J: **The maximum weight trace problem inmultiple sequence alignment.** *Proc CPM'93, Volume 684 of LNCS* 1993:106-119.
41. Lenhof HP, Reinert K, Vingron M: **A Polyhedral Approach to RNA Sequence Structure Alignment.** *J Comput Biol* 1998, **5(3):**517-530.
42. Caprara A, Lancia G: **Structural Alignment of Large-Size Proteins via Lagrangian Relaxation.** In *Proc of RECOMB'02* ACM Press; 2002:100-108.
43. Bauer M, Klau GW: **Structural Alignment of Two RNA Sequences with Lagrangian Relaxation.** In *Proc of ISAAC'04, no 3341 in LNCS* Springer; 2004:113-123.
44. Althaus E, Caprara A, Lenhof HP, Reinert K: **A Branch-and-Cut Algorithm for Multiple Sequence Alignment.** *Mathematical Programming* 2006, **105(2–3):**387-425.
45. Staple DW, Butcher SE: **Pseudoknots: RNA Structures with Diverse Functions.** *PLoS Biology* 2005, **3(6):**e213.
46. Dost B, Han B, Zhang S, Bafna V: **Structural Alignment of Pseudoknotted RNA.** *Proceedings of RECOMB* 2006:143-158.
47. Klein R, Eddy SR: **RSEARCH: Finding homologs of single structured RNA sequences.** *BMC Bioinformatics* 2003, **4:**44.
48. Dowell RD, Eddy SR: **Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction.** *BMC Bioinformatics* 2004, **5:**71.
49. Goldman D, Papadimitriou CH, Istrail S: **Algorithmic Aspects of Protein Structure Similarity.** *Proc FOCS'99* 1999:512-522.
50. Notredame C, Higgins DG, Heringa J: **T-Coffee: A novel method for fast and accurate multiple sequence alignment.** *Journal of Molecular Biology* 2000.
51. Wilm A, Mainz I, Steger G: **An enhanced RNA alignment benchmark for sequence alignment programs.** *Algorithms for Molecular Biology* 2006, **1:**19.
52. **BRAliBase 2.1** [http://www.biophys.uni-duesseldorf.de/bralibase/]
53. Griffiths-Jones S, Moxon S, Marshall M, Khanna A, Eddy SR, Bateman A: **Rfam: annotating non-coding RNAs in complete genomes.** *Nucl Acids Res* 2005, **33:**D121-124.
54. Washietl S, Hofacker I, Lukasser M, Hüttenhofer A, Stadler P: **Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome.** *Nature Biotechnology* 2005, **23(11):**1383-1390.
55. **MARNA** [http://biwww2.informatik.uni-freiburg.de/Software/MARNA/index.html]
56. **StrAl Webservice** [http://www.biophys.uni-duesseldorf.de/stral/]
57. **Foldalign** [http://foldalign.ku.dk/software/index.html]
58. Katoh K, Kuma Ki, Toh H, Miyata T: **MAFFT version 5: improvement in accuracy of multiple sequence alignment.** *Nucl Acids Res* 2005, **33(2):**511-518.
59. **MAFFT – a multiple sequence alignment program** [http://align.bmr.kyushu-u.ac.jp/mafft/software/source.html]
60. Thompson JD, Higgins DG, Gibson TJ: **CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.** *Nucl Acids Res* 1994, **22(22):**4673-4680.
61. Shapiro BA, Zhang K: **Comparing Multiple RNA Secondary Structures Using Tree Comparisons.** *CABIOS* 1990, **6:**309-318.
62. Evans P: **Finding Common Subsequences with Arcs and Pseudoknots.** In *Proc of CPM'99, no 1645 in LNCS* Springer; 1999:270-280.