Fast and Accurate Structural RNA Alignment by Progressive Lagrangian Optimization*

Markus Bauer [a,b], Gunnar W. Klau [c], Knut Reinert [a]

a) Institute of Computer Science, Free University of Berlin, Germany, b) International Max Planck Research School on Computational Biology and Scientific Computing, c) Institute of Mathematics, Free University of Berlin, Germany

Abstract. During the last few years new functionalities of RNA have been discovered, renewing the need for computational tools for their analysis. To this respect, multiple sequence alignment is an essential step in finding structurally conserved regions in related RNA sequences. In contrast to proteins, many classes of functionally related RNA molecules show a rather weak sequence conservation but instead a fairly well conserved secondary structure. Hence, any method that relates RNA sequences in form of multiple alignments should take structural features into account, which has been verified in recent studies.
Progress has been made in developing new structural alignment algorithms, however, current methods are computationally costly or do not have the desired accuracy to make them an everyday tool. In this paper we present a fast, practical, and accurate method for computing multiple, structural RNA alignments. The method is based on combining a new pairwise structural alignment method with the popular program T-Coffee. Our pairwise method is based on an integer linear programming (ILP) formulation resulting from a graph-theoretic reformulation of the structural alignment problem. We find provably optimal or near-optimal solutions of the ILP with a Lagrangian approach. Tests on a recently published benchmark set show that our Lagrangian approach outperforms current programs in quality and in the length of the sequences it can align.

1 Introduction

Recently, it has become clear that RNA molecules perform additional functions that were previously thought of being carried out by proteins. Many more of these *functional RNAs* have yet to be discovered. Computing multiple alignments to detect structural features is usually the first step in analyzing sequences of biomolecules. Unfortunately, and unlike proteins, many functional classes of RNA show little sequence conservation, but rather a conserved secondary structure which is formed by folding in space and forming hydrogen bonds between its bases. Among such RNAs are tRNA, rRNA, snoRNAs, and SRP RNA [11].

* Supported by the DFG Research Center Matheon "Mathematics for key technologies" in Berlin.

Hence, algorithms to compute multiple alignments ought to take not only the sequence, but also the secondary structure into account. Washietl and Hofacker [26] support this consideration by showing that sequence based alignments are significantly worse than sequence-structure based alignments if their pairwise sequence identity sinks below $\approx 60\%$. This observation is confirmed by Gardner and coworkers [8] in a paper that also benchmarks numerous multiple alignment programs.

Thus, the problem of producing RNA alignments that find a common structure has become the bottleneck in the computational study of functional RNAs. To date, the available tools for computing structural alignments are often incapable of handling reasonable input sizes or produce alignments of low quality. With this work we present a *multiple* RNA sequence-structure alignment tool that computes fast and accurate alignments. Our method uses a new pairwise structural alignment algorithm based on Lagrangian relaxation in combination with the progressive alignment tool T-Coffee.

Previous Work. The computational problem of considering sequence and structure of an RNA molecule simultaneously was first addressed by Sankoff [23] who proposed a dynamic programming algorithm that aligns a set of RNA sequences while at the same time predicting their common fold. The running time of this algorithm is $O(n^{3m})$ where m is the number of sequences. Algorithms similar in spirit were proposed later for the problem of comparing one RNA sequence to one or more of known structure. Corpet and Michot [5] align simultaneously a sequence with a number of other, already aligned, sequences using both primary and secondary structure. Their dynamic programming algorithm requires $O(n^5)$ running time and $O(n^4)$ space (n is the length of the sequences) and thus can handle only short sequences. Current implementations modify Sankoff's algorithm by imposing limits on the size or shape of substructures (*e.g.*, Dynalign [20, 19], Foldalign [15], PMcomp [11], Stemloc [13, 12], or work by Gorodkin *et al.* [9]).

Bafna *et al.* [1] gave an algorithm that simultaneously aligns the primary and secondary structure of two sequences that runs in time $O(n^4)$ which still does not make it applicable to instances of realistic size. Common motifs among several sequences are searched by Waterman [27]. Eddy and Durbin [7] describe probabilistic 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. Hofacker *et al.* [11] give a different structural alignment approach: instead of folding and aligning sequences simultaneously, they present a dynamic programming approach to align the corresponding *base pair probability matrices*, computed by McCaskill's partition function algorithm [21]. Their approach takes time $O(n^6)$ and space $O(n^4)$, but can be reduced by solving a banded version of the problem to $O(n^4)$ time and $O(n^3)$ space complexity.

The base pair probabilities can be directly used to weight edges in the structural alignment graph introduced in Lenhof *et al.* [18] where the authors presented a branch-and-cut algorithm for structurally aligning two RNA sequences. The underlying graph-theoretical formulation is flexible and allows for pseudoknots. Previous work on contact map overlap in the area of proteomics by Caprara and Lancia [4] and for the two-sequence case of the structural alignment problem by Bauer and Klau [2] indicates, however, that Lagrangian relaxation is better suited to obtain provably optimal or near-optimal solutions to the corresponding integer linear programming (ILP) formulation than a direct branch-and-cut approach in terms of running time. Bauer, Klau, and Reinert extend these ideas to multiple sequences [3]. Currently, however, the approach is applicable only to few sequences and small instance sizes.

Contribution. Our goal is to devise a fast method to compute high-quality, multiple structural alignments for a large number of possibly long RNA sequences.

Our key idea is to use the program T-Coffee [22], a successful multiple sequence alignment program that conducts a progressive alignment similar to ClustalW [25] but additionally incorporates *local* alignment information in form of so called *libraries*. This idea is not new by itself. Siebert and Backofen [24] already employ it in their program MARNA. The difference lies in the way the pairwise alignments are computed.

We use the implementation of Bauer and Klau [2] (Lara) and improve it in several ways, such that the obtained pairwise, structural alignments are very accurate, while Siebert and Backofen structurally align a set of sequences using the edit operations proposed in [16].

We will show that our implementation T-Lara consistently outperforms MARNA on a published benchmark set [8]. In addition, T-Lara is better or competitive to other, more costly structural alignment programs and can handle much longer sequences while maintaining a running time of only a couple of minutes.

2 Lagrangian Structural Alignment of Two Sequences

We have described the theoretical framework of the Lagrangian approach to structural sequence alignment elsewhere (see [2, 3]). We therefore provide only a short summary of the basic approach and focus on practical improvements of the approach such as the incorporation of affine gap costs and a more sophisticated selection of candidate edges.

2.1 Terminology and Basic Approach

Let S be a sequence $s_1, \ldots, s_n$ of length n over the alphabet $\Sigma = \{A, C, G, U\}$. A paired base (i, j) is called an *interaction* if (i, j) forms a Watson-Crick-pair. The set P of interactions is called the *annotation* of sequence S. Two interactions are said to be in *conflict*, if they share one base; they form a *pseudoknot* if they cross each other. A pair (S, P) is called an *annotated sequence*. Note that a

structure where no pair of interactions is in conflict with each other forms a valid secondary structure of an RNA sequence, possibly with pseudoknots.

We are given two annotated sequences (S_1, P_1) and (S_2, P_2) and model the input as a graph $G = (V, L \cup I)$. The set V denotes the vertices of the graph, in this case the bases of the sequences. The set L contains *alignment edges* between vertices of the two input sequences (for sake of better distinction called *lines*) whereas the set I codes the two annotations by means of *interaction edges* between vertices of the same sequence. A subset $\mathcal{L} \subset L$ corresponds to an *alignment* of the two sequences if $\mathcal{L}$ does not contain crossing lines, since those correspond to ordering conflicts of the letters in the sequences. Two interaction edges $(i_1, i_2) \in P_i$ and $(j_1, j_2) \in P_j$ are said to be *realized* by an alignment $\mathcal{L}$ if and only if $\mathcal{L}$ contains the alignment edges $l = (i_1, j_1)$ and $m = (i_2, j_2)$. The pair (l, m) is called an *interaction match*. Note that we define (l, m) as an ordered tuple, that is, (l, m) is distinct from (m, l). Figure 1 illustrates the above definitions by means of an example.

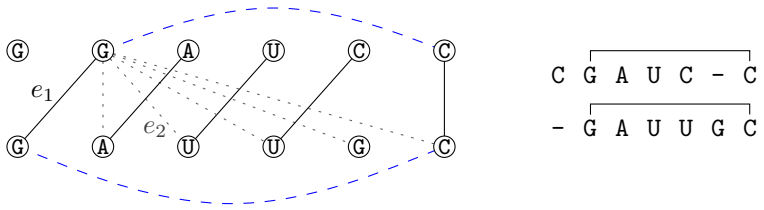

Fig. 1. Graph-theoretic concept of alignment. The right side shows a structural alignment of two annotated sequences, the left side the corresponding graph G. Solid lines represent alignment edges in $\mathcal{L}$, dotted lines represent additional candidate edges from L (only a subset shown). Replacing, *e.g.*, $e_1 \in \mathcal{L}$, by e_2 creates a crossing. Lines $\mathcal{L}$ realize two interaction matches (remember that interaction matches are ordered tuples).

We assign positive weights w_l and w_{ij} to each line l and each interaction match (i, j), respectively, that represent the benefit of realizing the line or the match. The weights are given, for example, by mutation score matrices or—in the case of interaction matches—by the number of hydrogen bonds between the bases or by the base pair probabilities.

The structural alignment problem now corresponds to finding a maximally weighted subset of lines and interaction edges in the input graph such that no lines cross each other, each interaction match is realized, and no vertex is incident to more than one interaction edge. We define binary variables x_e for each edge e and y_{lm} for each interaction match (l, m) and rewrite the problem as the following integer linear program:

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

$$\text{s.t.} \sum_{l \in I} x_l \leq 1 \qquad \forall \text{ sets of crossing lines } I \tag{2}$$

$$y_{lm} = y_{ml} \qquad \forall\, l, m \in L \tag{3}$$

$$\sum_{m \in L} y_{lm} \leq x_l \qquad \forall\, l \in L \tag{4}$$

$$0 \leq x \leq 1, \quad 0 \leq y \leq 1 \qquad \text{integer} \tag{5}$$

We have shown in [2] that dropping constraints (3) leads to a much easier problem, namely a classical primary sequence alignment problem that can be solved in polynomial time. We follow the iterative Lagrangian optimization method and move the complicating constraints into the objective function with a penalty term for their violation, resulting in the *relaxed problem*. An iteration consists of solving an instance of the relaxed problem and adapting the penalty terms. As a by-product we obtain a feasible solution in each iteration by interpreting the solution of the relaxed problem as an input graph for a maximum weighted matching problem.

2.2 Practical Improvements

We have implemented various modifications of the basic approach described in the preceding section in order to increase its applicability to practical RNA data.

The basic approach does not consider gap costs and alignments computed with an early version of our implementation suffered from this drawback. We have therefore replaced the recurrence relation in the standard dynamic programming algorithm for classical primary sequence alignment by a version that takes into account affine gap scores (see, *e.g.*, [10]). We have also modified the backtracking in the dynamic programming matrix in order to account for a different treatment of gaps occurring at the beginning or the end of the sequence.

We achieved a speed-up compared to the basic approach by providing another way we select the candidate edges. Note that only the complete bipartite graph models all possible alignments of two sequences. In practice, this is computationally too expensive, and we resort to a heuristic selection of promising candidate edges:

Instead of computing a conventional sequence alignment with affine gap costs and subsequently inserting all alignment edges realized by any suboptimal alignment scoring better than a fixed threshold s below the optimal score (as used in [2]), we provide a *sliding window technique*—as described in [17]—that adjusts the suboptimality threshold s according to the local quality of the alignment. More precisely, for every nucleotide we compute a *confidence value* evaluating the quality of the local alignment within a certain window. In regions of the sequence where the quality of the conventional sequence alignment appears to

be very good, none or only a small number of suboptimal alignment edges are considered. In alignment regions that show little sequence conservation, more alignment edges are generated.

3 Extension to Multiple Sequences

We have shown how to extend the formulation (1)-(5) and the Lagrangian relaxation technique to the multiple sequence case in [3]. Here, we follow a different approach, since the inherent computational complexity of the multiple structural sequence alignment problem impedes the use of exact methods for instances with many sequences. We wish to remark that we are following two different lines of research: on the one hand, we investigate the structure of truly optimal multiple alignments and aim at solving instances of three or four sequences to provable optimality. On the other hand, we wish to provide a fast and practical—although possibly suboptimal—tool based on the good results of the pairwise algorithm. For this reason, we decided to integrate our pairwise algorithm into a progressive alignment framework.

3.1 Progressive Alignment with T-Coffee

T-Coffee uses a progressive alignment approach similar to the one of ClustalW [25]. Progressive methods build multiple alignments from pairwise alignments. The pairwise distances are usually used to compute a guide tree which in turn is used to determine the order in which the sequences are aligned to the evolving multiple alignment.

Progressive approaches usually 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. A nice feature about the T-Coffee implementation is, that the user can supply such local alignment information. While the default local library is computed with Lalign [14], an alignment algorithm based on primary sequence, we compute the local library using Lara [2] thereby effectively providing structural information to T-Coffee.

4 Computational Results

4.1 Materials and Methods

We took a subset of data from the recently published BRaliBase dataset[1] [8] and used the *structure conservation index* SCI as a score to compare the results from different programs.

The SCI value compares the minimum free energies of the single sequences in an alignment with a "consensus energy" imposed by the alignment, which

[1] BRaliBase is freely available from `http://www.binf.ku.dk/users/pgardner/bralibase/`

Table 1. Average SCI scores computed over a test set of 242 instances with different programs.

Program	Av. SCI
`clustal`	0.6076
`MUSCLE`	0.6069
`T-Coffee`	0.5972
`T-Lara`	0.71

Table 2. Average T-Lara SCI scores for the different groups of test instances.

Group (# of instances)	Av. SCI
5S rRNA (39)	0.84
U5 spliceosomal (101)	0.60
Group II introns (72)	0.73
tRNA (30)	0.84

is computed by incorporating covariation terms into a free energy minimization computation. More technically, let $\hat{E}$ be the consensus energy value of the alignment and E_n be the mean of all MFE (*minimum free energy*) values of n sequences, respectively. Then the SCI is defined as

$$\text{SCI} = \frac{\hat{E}}{E_n}$$

An SCI close to zero indicates that there is no conserved structure within the alignment, whereas SCI > 1 exhibits a perfectly conserved structure, additionally supported by compensatory mutations. Therefore, the SCI assesses in particular the structural quality of an alignment.

As a first test, we took all instances with low homology (that is with sequence identity $< 55\%$) of the first dataset that was used by Gardner *et al.* in [8]: we computed a structural alignment of all 242 instances, with one instance being a set of either five Group II introns, 5S rRNA, tRNA or U5 spliceosomal RNA sequences. The entire computation took 345.93 minutes on an AMD Opteron server running at 2Ghz. Table 1 shows the average SCI scores of the three best-scoring sequence-based programs on the low homology data. It should be noted that the alignment program (`clustalW`) computing the best SCI score of the first dataset reached an average SCI score of only 0.6076. Table 2 gives a more detailed view of T-Lara's performance on the different subgroups.

The big gap between T-Lara and the other programs is easily explained by the fact that due to the extensive computational demands of structure alignment programs, Gardner and colleagues only used sequence based approaches for the first dataset, a limitation that T-Lara removes. In case of sequences with low sequence identity (say below 50%), structure alignment programs compute significantly better alignments in terms of conserving structural motifs.

For comparing structure alignment programs, Gardner *et al.* chose a subset of tRNA instances consisting of only two tRNA sequences (some programs tested in

Table 3. SCI scores of `clustalW`, `MARNA` and `T-Lara` on SRP sequences.

SeqID	Instance	`clustalW`	`MARNA`	`T-Lara`
0.49	aln38	0.55	0.55	0.66
0.50	aln58	0.86	0.68	1.00
0.50	aln27	0.54	0.26	0.58
0.51	aln16	0.54	0.22	0.62
0.52	aln11	0.48	0.22	0.54
0.53	aln6	0.62	0.44	0.66
0.53	aln7	0.63	0.56	0.70
0.54	aln20	0.66	0.55	0.78
0.54	aln28	0.62	0.35	0.69
0.54	aln5	0.63	0.35	0.73
0.60	aln21	0.49	0.45	0.54

that survey are only capable of computing pairwise structural alignments). Since our approach can handle multiple sequences, we augmented this dataset and calculated all tRNA instances (consisting of five sequences) from the first dataset and compared them to `pmmulti`—a banded variant of Sankoff's approach—and `clustalW`. Over a set of 97 instances of five tRNA sequences (`pmmulti` failed on one instance) the average SCI score of `clustalW`—one of the best sequence-based alignment programs from the first dataset—is 0.82, whereas `pmmulti` and T-Lara reach average SCI scores of 1.043 and 1.029 at a running time of 101.91 and 75.16 minutes, respectively. `pmmulti` and T-Lara have almost the same score, but it has to be noted that due to the extensive computational costs the exact approach `pmmulti` can only be applied to short sequences (say at most 150 nucleotides), whereas T-Lara can handle sequences of several hundred nucleotides (a length where structure alignment programs based on dynamic programming must fail).

To illustrate our ability to handle long sequences, we took 11 instances of three SRP RNA sequences from BRaliBase and compared the alignments computed by T-Lara to those of `clustalW` and `MARNA` (a structure alignment program that is also capable of dealing with long sequences).

Table 3 shows the computed SCI scores of `clustalW`, `MARNA`, and T-Lara, respectively. We were able to calculate only such a small number of instances, since `MARNA` can be accessed only by a web interface which makes the evaluation tedious. For the instances computed, however, the table shows that T-Lara clearly outperforms `clustalW` and `MARNA` in terms of conserving structural elements. Furthermore, computing the alignments of the 11 instances takes just 34.5 minutes in total.

5 Discussion

In this paper we presented the new multiple structural alignment program T-Lara. Our experiments show that T-Lara computes structural alignments comparable or better than those computed by variants of Sankoff's algorithm. Our approach, however, can also be applied to longer sequences (*e.g.*, 16S rRNA

sequences of length ≈ 1600 nucleotides) since we do not suffer from the restrictive demands in terms of CPU time and memory imposed by Sankoff's dynamic programming algorithm.

Additionally, our algorithm does not restrict the secondary structure of a given sequence in any way (*i.e.*, the approach allows arbitrary pseudoknots). Therefore, we plan to integrate more accurate base pair probabilities based on pseudoknot energy parameters (like for example [6]).

In the future we will extend our Lagrangian approach with our own progressive code (similar in spirit to `pmmulti`), and incorporating better scoring matrices (*e.g.*, `RIBOSUM` matrices) should additionally enhance the quality of the alignments.

Furthermore, a web service providing access to our algorithm is currently developed. A public-domain version of the program will follow in the next weeks.

Acknowledgments. The authors thank Veronika Gamper for implementing Lara's T-Coffee library support and the gap score modifications.

References

1. V. Bafna, S. Muthukrishnan, and R. Ravi. Computing similarity between RNA strings. In Z. Galil and E. Ukkonen, editors, *Proc. of the 6th Annual Symp. on Combinatorial Pattern Matching*, number 937 in Lecture Notes in Computer Science, pages 1–16. Springer, 1995.
2. M. Bauer and G. W. Klau. Structural Alignment of Two RNA Sequences with Lagrangian Relaxation. In *Proc. Symp. on Algorithms and Computation (ISAAC)*, number 3341 in Lecture Notes in Computer Science, pages 113–123. Springer, 2004.
3. M. Bauer, G. W. Klau, and K. Reinert. Multiple structural RNA alignment with Lagrangian relaxation. Submitted.
4. A. Caprara and G. Lancia. Structural Alignment of Large-Size Proteins via Lagrangian Relaxation. In *Proceedings of the Sixth Annual International Conference on Computational Biology*, pages 100–108. ACM Press, 2002.
5. F. Corpet and B. Michot. RNAlign program: alignment of RNA sequences using both primary and secondary structures. *CABIOS*, 10(4):389–399, 1994.
6. R. Dirks and N. Pierce. An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *Journal of Computational Chemistry*, 25:1295–1304, 2004.
7. S. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Research*, 22(11):2079–2088, 1994.
8. P. P. Gardner, A. Wilm, and S. Washietl. A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Research*, 22(8):2433–2439, 2005.
9. J. Gorodkin, L. J. Heyer, and G. D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucl. Acids Res.*, 25:3724–3732, 1997.
10. O. Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, pages 705–708, 1982.
11. I. L. Hofacker, S. H. F. Bernhart, and P. F. Stadler. Alignment of RNA base pairing probability matrices. *Bioinformatics*, 20:2222–2227, 2004.

12. I. Holmes. Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*, 5:73, 2004.
13. I. Holmes. A probabilistic model for the evolution of RNA structure. *BMC Bioinformatics*, 5:166, 2004.
14. X. Huang and W. Miller. A time efficient, linear space local similarity algorithm. *Adv. Appl. Math.*, 12:337–357, 1991.
15. J. Hull Havgaard, R. Lyngso, G. Stormo, and J. Gorodkin. Pairwise local structural alignment of RNA sequences with sequence similarity less then 40%. *Bioinformatics*, page in press, 2005.
16. T. Jiang, G.-H. Lin, B. Ma, and K. Zhang. A general edit distance between RNA structures. *J. of Computational Biology*, 9:371–388, 2002.
17. J. Kececioglu, H.-P. Lenhof, K. Mehlhorn, P. Mutzel, K. Reinert, and M. Vingron. A polyhedral approach to sequence alignment problems. *Discrete Applied Mathematics*, 104:143–186, 2000.
18. H.-P. Lenhof, K. Reinert, and M. Vingron. A Polyhedral Approach to RNA Sequence Structure Alignment. *Journal of Comp. Biology*, 5(3):517–530, 1998.
19. D. Mathews. Predicting a set of minimal free energy RNA secondary structures common to two sequences. *Bioinformatics*, page in press, 2005.
20. D. H. Mathews and D. H. Turner. Dynalign: An algorithm for finding secondary structures common to two RNA sequences. *J. Mol. Biol.*, 317:191–203, 2002.
21. J. S. McCaskill. The Equilibrium Partition Function and Base Pair Binding Probabilities for RNA Secondary Structure. *Biopolymers*, 29:1105–1119, 1990.
22. C. Notredame, D. G. Higgins, and J. Heringa. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 2000.
23. D. Sankoff. Simultaneous solution of the RNA folding, alignment, and protosequence problems. *SIAM J. Appl. Math.*, 45:810–825, 1985.
24. S. Siebert and R. Backofen. Marna: A server for multiple alignment of RNAs. In *GCB 2003*, October 2003.
25. J. D. Thompson, D. G. Higgins, and T. J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucl. Acids Res.*, 22:4673–4680, 1994.
26. S. Washietl and I. L. Hofacker. Consensus folding of aligned sequences as a new measure for the detection of functional rnas by comparative genomics. *Journal of Molecular Biology*, 2004.
27. M. Waterman. Consensus methods for folding single-stranded nucleic adds. *Mathematical Methods for DNA Sequences*, pages 185–224, 1989.