Bioinformatics, 2015, 1–8
doi: 10.1093/bioinformatics/btv185
Advance Access Publication Date: 2 April 2015
Original Paper

OXFORD

Sequence analysis

SPARSE: quadratic time simultaneous alignment and folding of RNAs without sequence-based heuristics

Sebastian Will[1,2,†], Christina Otto[1,†], Milad Miladi[1,†], Mathias Möhl[1] and Rolf Backofen[1,3,4,5,*]

[1]Bioinformatics, Department of Computer Science, University of Freiburg, Freiburg, Germany, [2]Bioinformatics, Department of Computer Science, University of Leipzig, Leipzig, Germany, [3]Centre for Biological Systems Analysis (ZBSA), University of Freiburg, Freiburg, Germany, [4]Centre for Non-coding RNA in Technology and Health, University of Copenhagen, Copenhagen, Denmark and [5]Centre for Biological Signalling Studies (BIOSS), University of Freiburg, Freiburg, Germany

*To whom correspondence should be addressed.
[†]The authors wish it to be known that, in their opinion, the first three authors should be regarded as Joint First Authors.
Associate Editor: John Hancock

Received on November 7, 2014; revised on March 9, 2015; accepted on March 25, 2015

Abstract

Motivation: RNA-Seq experiments have revealed a multitude of novel ncRNAs. The gold standard for their analysis based on simultaneous alignment and folding suffers from extreme time complexity of $O(n^6)$. Subsequently, numerous faster 'Sankoff-style' approaches have been suggested. Commonly, the performance of such methods relies on sequence-based heuristics that restrict the search space to optimal or near-optimal sequence alignments; however, the accuracy of sequence-based methods breaks down for RNAs with sequence identities below 60%. Alignment approaches like LocARNA that do not require sequence-based heuristics, have been limited to high complexity ($\geq$ quartic time).

Results: Breaking this barrier, we introduce the novel Sankoff-style algorithm 'sparsified prediction and alignment of RNAs based on their structure ensembles (SPARSE)', which runs in *quadratic time* without sequence-based heuristics. To achieve this low complexity, on par with sequence alignment algorithms, SPARSE features strong sparsification based on structural properties of the RNA ensembles. Following PMcomp, SPARSE gains further speed-up from lightweight energy computation. Although all existing lightweight Sankoff-style methods restrict Sankoff's original model by disallowing loop deletions and insertions, SPARSE transfers the Sankoff algorithm to the lightweight energy model completely for the first time. Compared with LocARNA, SPARSE achieves similar alignment and better folding quality in significantly less time (speedup: 3.7). At similar run-time, it aligns low sequence identity instances substantially more accurate than RAF, which uses sequence-based heuristics.

Availability and implementation: SPARSE is freely available at http://www.bioinf.uni-freiburg.de/Software/SPARSE.

Contact: backofen@informatik.uni-freiburg.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The majority of transcripts are non-coding RNAs (ncRNAs), which unlike mRNAs do not code for proteins. ncRNAs are associated with a large range of important cellular functions; furthermore, there is increasing evidence of pervasive transcription (e.g. Jacquier, 2009; Clark *et al.*, 2011). Particularly, up to 450 000 ncRNAs have been predicted in the human genome alone (Rederstorff *et al.*, 2010).

Analyzing the huge amount of RNA sequences poses major challenges for bioinformatics; particularly, since the sequence of related ncRNAs is often conserved only weakly, while the RNAs can still share a strongly conserved consensus structure. Therefore, taking sequence and structure similarity into account is indispensable for ncRNA analysis (Will *et al.*, 2007; Torarinsson *et al.*, 2007; Shi *et al.*, 2009; Tseng *et al.*, 2009; Saito *et al.*, 2011; Parker *et al.*, 2011). However, accurate methods for this purpose have extreme computational costs.

The gold standard for RNA alignment has been introduced by Sankoff (1985). Because structure prediction and alignment of RNAs depend on each other, Sankoff's approach solves the alignment and folding problem simultaneously. For two RNA sequences, it finds two energetically favorable structures of the same shape together with a good alignment that reflects the similarity of the structures. For this purpose, Sankoff composes its objective function from a sequence alignment score and the free energies of the two structures. Because of the extreme $O(n^6)$ time complexity of this algorithm, numerous Sankoff-like strategies have been developed aiming to speed up Sankoff's task, while preserving accuracy as much as possible.

A major class of such methods utilizes information from sequence-based alignment to reduce the search space for the computationally much more expensive alignment and folding algorithm. This idea was introduced in Holmes (2005), and later refined by Dowell and Eddy (2006) and Harmanci *et al.* (2007). The latter restricts the alignment space to an envelope around the base matches, whose sequence alignment probabilities exceed a fixed cutoff. Generally, such approaches have to cope with the well-known fact that sequence alignment fails for sequence identities below 60% (Gardner *et al.*, 2005). Consequently, sequence alignments can provide hints at the optimal structure-based alignment, but are potentially far-off. Moreover, even with such improvements Sankoff-like methods remain too expensive for large analysis tasks like clustering putative ncRNAs in large datasets, e.g. the entire human transcriptome or even meta-genomics data.

PMcomp (Hofacker *et al.*, 2004) suggested a fundamentally different route to faster RNA alignment. It introduced a new Sankoff-like scoring model that enables lightweight computation. For this purpose, it employs a base pair-based energy model instead of the original loop-based energy model. Furthermore, PMcomp simplifies Sankoff's model by predicting only a single consensus structure. Tools like LocARNA (Will *et al.*, 2007), FoldAlignM (Torarinsson *et al.*, 2007) and LocARNA-P (Will *et al.*, 2012) build on the PMcomp model, but additionally sparsify the folding spaces of both RNAs, resulting in $O(n^4)$ time complexity. CARNA (Sorescu *et al.*, 2012) extends the PMcomp model to pseudoknot structures. RAF (Do *et al.*, 2008) combined the ideas of Harmanci *et al.* (2007) and Hofacker *et al.* (2004), resulting in a lightweight Sankoff-variant with sequence-based speed up.

1.1 Contributions

In this work, we introduce the novel lightweight Sankoff-style approach SPARSE (sparsified prediction and alignment of RNAs based on their structure ensembles) with quadratic time complexity. Achieving the same complexity as sequence alignment with SPARSE is a breakthrough for RNA analysis, because this profound performance gain works without sequence-based heuristics. On the contrary, it is purely based on information from the RNAs' structural ensembles. Consequently, the technique is applicable without compromising the alignment quality in the low sequence identity zone.

Going beyond LocARNA, which sparsifies based on base pair probabilities, SPARSE additionally sparsifies based on conditional probabilities of bases and base pairs within loops (Otto *et al.*, 2014). The latter results in a quadratic time improvement over LocARNA, which already improved by a quadratic factor over Sankoff's algorithm; this results in the quadratic time complexity of SPARSE, a *quartic speedup* over the original Sankoff algorithm.

Concretely, SPARSE sparsifies the novel Sankoff-style algorithm prediction and alignment of RNAs based on their structure ensembles (PARSE), which—like PMcomp—supports lightweight computation. Going beyond PMcomp, it predicts two potentially different structures for the two RNA sequences, supporting insertions and deletions of loops. The dependencies between these structures and the alignment are exactly the same as in the original problem formulation of Sankoff. Thus, the increased flexibility of PARSE over the PMcomp-like previous RNA alignment approaches is an important contribution by itself. We show that the novel problem can be efficiently solved by dynamic programming. Moreover, we present a model and algorithm for affine gap costs that distinguishes insertions and deletions of single bases and entire loops.

Based on SPARSE, we furthermore develop a fast *multiple* RNA alignment approach. In our benchmarks, compared with LocARNA, the much faster SPARSE computes alignments of similar quality with improved folding quality. Compared with RAF (Do *et al.*, 2008), SPARSE is similarly fast (speedup over LocARNA of 4.2 versus 3.7) and has the same time complexity, but provides superior alignment quality in the hard case of low sequence identity.

2 Methods

Figure 1 illustrates the subsequent fundamental preliminaries.

RNAs

An *RNA sequence* A is a string over the alphabet $\{\mathcal{A}, C, G, U\}$ with *length* $|A|$. We denote the base at the i-th position of A by $A[i]$; the substring from position i to j, by $A[ij]$; such substrings of RNA sequences are subsequently called *subsequences*. A *base pair a of A* is a pair $(a^{\mathrm{L}}, a^{\mathrm{R}})$ $(1 \leq a^{\mathrm{L}} < a^{\mathrm{R}} \leq |A|)$. A *non-crossing RNA structure S*

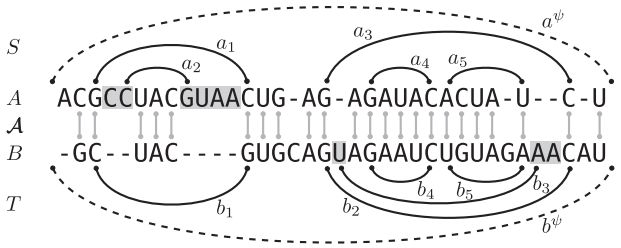

Fig. 1. Example *alignment* $\mathcal{A}$ of two *RNA sequences* A and B together with (*non-crossing*) *structures* $S = \{a_1, a_2, a_3, a_4, a_5\}$ and $T = \{b_1, b_2, b_3, b_4, b_5\}$. We highlight the positions in the *loop closed by* a_1 and in the *loop closed by* b_2. The base pair a_1 is the *parent* of the highlighted positions in A and of a_2. The base pair a_1 closes a 2-*loop*; a_2, a 1-*loop* and a_3, a *multiloop*. The latter is a 3-*loop*, since a_3 has two *inner* base pairs (a_4 and a_5.) Note that the *structure alignment triple* $(\mathcal{A}, S, T)$ covers the *external* base pairs a_1, a_3, b_1 and b_2; as well as the inner base pairs of the two multiloops. Finally, $(\mathcal{A}, S, T)$ *deletes the entire* 2-*loop of* a_1 and *inserts the entire* 2-*loop of* b_2

of A, in the following called *structure*, is a set of base pairs, where each two different base pairs (i, j) and (i', j') of S do not share any end, i.e. i, j, i', and j' are pairwise different, and base pairs of S do not cross, i.e. there are no $(i,j),(i',j') \in S$ with $i < i' < j < j'$. For reasons of simplicity, we introduce a pseudo base pair $a_\psi := (0, |\mathcal{A}| + 1)$, which formally closes the external loop of A. Although a_ψ does not satisfy $1 \leq a_\psi^{\mathrm{L}} < a_\psi^{\mathrm{R}} \leq |\mathcal{A}|$, it is otherwise handled like a base pair of A. The position k of A is *paired* according to S, iff there exists k' such that $(k,k') \in S$ or $(k',k) \in S$. Otherwise k is *unpaired*.

Loops

For any position k of A, we define the *parent of k in S*, written $\mathrm{parent}_S(k)$, as the $(i,j) \in S \cup \{a_\psi\}$ with $i < k < j$ such that there does not exist any $(i',j') \in S$ with $i < i' < k < j' < j$. Analogously, the *parent of a base pair a in S*, written $\mathrm{parent}_S(a)$, is the parent of a^{L} (or, equivalently, the parent of a^{R}). A position k in A or base pair $(k,j) \in S$ is *in the loop closed by a* iff $\mathrm{parent}_S(k) = a$, *external* according to S iff $\mathrm{parent}_S(k) = a_\psi$, otherwise *internal*. Furthermore, for a base pair $a \in S$, the *loop closed by a* consists of the positions $\mathrm{loop}_S(a) := \{k | k \in [1..|\mathcal{A}|], \mathrm{parent}_S(k) = a\}$. In a structure S, a *k-loop* is a loop with closing base pair a and $k-1$ *inner base pairs* $a' \in S$, where $\mathrm{parent}_S(a') = a$. A *multiloop* is a k-loop where $k > 2$.

Alignments

An *alignment $\mathcal{A}$ of RNA sequences A and B* consists of a set of edges written as pairs (i, k), where i is a position in $\mathcal{A}$, and k a position in B. Alignment edges do not *cross*, i.e. for all $(i,j),(i',j') \in \mathcal{A} : (i < i' \Rightarrow j < j'$ and $i = i' \Leftrightarrow j = j')$. Usually, we consider alignments $\mathcal{A}$ together with structures S of A and T of B, forming the triple $(S, T, \mathcal{A})$.

The position i of A is called *deleted by* $\mathcal{A}$, iff there is no k of B s.t. $(i,k) \in \mathcal{A}$; k of B is *inserted by* $\mathcal{A}$, iff there is no i of A s.t. $(i,k) \in \mathcal{A}$. Positions that are neither deleted nor inserted by $\mathcal{A}$ are *covered by* $\mathcal{A}$. Analogously, we call a base pair $a \in S$ *covered by* $(S, T, \mathcal{A})$, iff there is a base pair $b \in T$, s.t. $(a^{\mathrm{L}}, b^{\mathrm{L}}),(a^{\mathrm{R}}, b^{\mathrm{R}}) \in \mathcal{A}$; symmetrically, we define *covered by* $(S, T, \mathcal{A})$ for base pairs $b \in T$. A base pair (i, j) is called *inserted* (*deleted*), iff i and j are *inserted* (*deleted*), respectively. *$\mathcal{A}$ matches two positions i and k to each other*, iff $(i,k) \in \mathcal{A}$. Two base pairs are *matched to each other by* $(S, T, \mathcal{A})$, iff $\mathcal{A}$ matches their left and right ends to each other.

For 2-loops, we define the notion of insertion or deletion of entire loops. $(S, T, \mathcal{A})$ deletes (inserts) the entire 2-loop closed by the base pair $a \in S$ ($b \in T$), iff it deletes (inserts) all bases in $\mathrm{loop}_S(a)$ ($\mathrm{loop}_T(b)$), respectively.

2.1 Lightweight Sankoff-style alignment

Given two RNA sequences A and B (of respective sizes n and m), Sankoff's problem of simultaneous alignment and folding (Sankoff, 1985) asks for an alignment and RNA structures S and T for both sequences that simultaneously optimize a score of the form 'energy of S + energy of T + sequence edit distance' in a loop-based energy model (Mathews *et al.*, 1999). Importantly, the alignment $\mathcal{A}$ and the structures S and T are not independent of each other: Sankoff requires that all external base pairs and interior base pairs of multiloops of both RNAs are covered by $(S, T, \mathcal{A})$. Moreover, Sankoff (1985) requires that any k-loop ($k > 2$) is '*aligned with a single k-loop in the other structure*', whereas 2-loops can be flexibly '*inserted or deleted in toto*' to align stems of different length. Due to these conditions, aligned multiloops are of the same degree, which preserves the shape [called *branching configuration* in Sankoff (1985)] of the RNA structures.

2.1.1 PMcomp—a lightweight and simplified Sankoff variant

PMcomp (Hofacker *et al.*, 2004) transfers Sankoff's idea to a *lightweight* energy model based on base pairs, which allows much faster computation. However, PMcomp simplifies the problem even more by introducing a one-to-one dependency between the predicted structures for S and T. In consequence, PMcomp predicts only a single consensus structure, whereas Sankoff much more flexibly predicts two *compatible* structures of A and B. We are going to show that PMcomp's second simplification (namely, to predict only the consensus structure) is not required for fast computation and even more has adverse effects.

In the simplified energy model of PMcomp, the energy of a structure is the sum of energy-like weights Ψ_{ij} of the single base pairs in each structure. Because PMcomp defines Ψ_{ij} as log odds of the base pair probability p_{ij} (McCaskill, 1990), the model effectively multiplies single base pair probabilities. Here, PMcomp follows the general idea to simplify probability calculations by assuming independence. Otherwise PMcomp, like the original Sankoff algorithm, optimizes a score composed of the sequence similarity and energies.

We rephrase the *alignment and folding score* of PMcomp, which is assigned to an alignment $\mathcal{A}$ and RNA structures S and T, as

$$\mathrm{score}(S,T,\mathcal{A}) := \sum_{(i,j)\in S} \Psi^{\mathcal{A}}_{ij} + \sum_{(k,l)\in T} \Psi^{B}_{kl} + \sum_{(i,k)\in\mathcal{A}} \sigma(i,k) + N_{\mathrm{indel}}\gamma, \quad (1)$$

where σ is the base similarity, γ is the gap penalty ($\gamma \leq 0$) and N_{indel} is the number of insertion and deletions in $\mathcal{A}$.

Due to its second simplification of Sankoff, PMcomp maximizes this score only over the $(S, T, \mathcal{A})$ that cover all base pairs in S and T. This is a strong restriction compared with the more expressive Sankoff algorithm, which requires only that the interior base pairs of multiloops and the external base pairs are covered by $(S, T, \mathcal{A})$. Thus, while Sankoff allows flexibly aligning stems of different lengths due to insertion and deletion of 2-loops, PMcomp cannot handle loop deletions and insertions at all.

2.1.2 PARSE—lightweight and flexible folding and alignment

In the novel algorithm PARSE, we overcome the limitations of PMcomp, maximizing the score of Equation (1) with the full flexibility of Sankoff's specification in terms of dependencies between alignment and predicted structures. Here, we paraphrase Sankoff's constraints using our notation and terminology.

DEFINITION 1 (Sankoff's Dependencies Between Alignment and Structures). Let A and B be sequences; S of A and T of B, structures; and $\mathcal{A}$, an alignment of A and B. $(S, T, \mathcal{A})$ is a *structure alignment triple of A and B* (satisfying Sankoff's dependencies), iff for each base pair $a \in$ S and $b \in T$

1. $(S, T, \mathcal{A})$ *covers a or deletes the entire loop of* $\mathrm{parent}_S(a)$,
2. $(S, T, \mathcal{A})$ *covers b or inserts the entire loop of* $\mathrm{parent}_T(b)$;

in (1) *and* (2), *the deleted or inserted loops have to be* 2*-loops.*

Figure 1 shows a structure alignment triple. The definition makes explicit that base pairs are inserted or deleted only together with their entire 2-loop. This enables predicting stems of different length and align them to each other. At the same time, the two predicted structures cannot differ arbitrarily, but must have the same shape.

2.1.3 Realistic gap cost

Because PARSE supports insertion and deletion (indel) of entire loops, which—unlike base indels—correspond to elongation or shortening of stems in the RNA structure, it is reasonable to distinguish the different evolutionary events of base indel and loop indel in our scoring function. Technically, we introduce two different gap penalties γ_{base} for gaps due to base indels and γ_{loop} for gaps due to loop indels. To produce biologically relevant alignments, we extend the above score to support affine gap costs. The different nature of loop indels and base indels suggests two different gap opening penalties $\beta_{\text{loop}} \leq 0$ and $\beta_{\text{base}} \leq 0$. Thus, in addition to the regular gap opening costs for base indels, we penalize the change of structure by loop indels with a specific loop gap opening cost.

3 Algorithms

In general, the structures S and T are selected from respective sets of possible base pairs P and Q of sizes N and M. For example, P and Q could consist of all canonical base pairs, but subsequently we are going to sparsify those sets. In our context, we generally assume that P and Q are sparse subsets of all possible base pairs. For example, in LocARNA (Will *et al.*, 2007), P and Q consist of—in the sequence length—only linearly many base pairs due to filtering by a fixed threshold probability.

3.1 A sparsification perspective on PMcomp

Originally, PMcomp was presented (Hofacker *et al.*, 2004) without sparsification in mind, defining dynamic programming matrices indexed by the ends of the aligned subsequences $\mathcal{A}[\text{i..}j]$ and $B[k..\text{l}]$. To make the correspondence to base pairs in P and Q explicit, we define matrix entries $\hat{D}(a,b)$ that represent the scores of matching the two base pairs a and b and aligning the two enclosed subsequences. Each $\hat{D}(a,b)$ is computed from entries $\hat{M}^{ab}(i,k)$ that store the best score of any two structures and alignment of subsequences $\mathcal{A}[a^{\text{L}}+1..i]$ and $B[b^{\text{L}}+1..k]$, where all base pairs in S and T are covered by $(S,T,\mathcal{A})$; subsequence scores are defined in the appendix.

The PMcomp algorithm is defined by recurrences for all $a \in P$, $b \in Q$, i $(a^{\text{L}} < i < a^{\text{R}})$, and k $(b^{\text{L}} < k < b^{\text{R}})$ (Recall that we simplified the score; still, the original recursions are easily obtained by adding the base pair match contribution $\tau(a,b)$ to $\hat{D}(a,b)$.):

$$\hat{M}^{ab}(i,k) = \max \begin{cases} \hat{M}^{ab}(i-1,k-1) + \sigma(i,k) \\ \hat{M}^{ab}(i,k-1) + \gamma \\ \hat{M}^{ab}(i-1,k) + \gamma \\ \max\limits_{\substack{a_1 \in P, b_1 \in Q \\ \text{s.t. } a_1^{\text{R}} = i,\ b_1^{\text{R}} = k}} \hat{M}^{ab}(a_1^{\text{L}}-1, b_1^{\text{L}}-1) + \hat{D}(a_1,b_1) \end{cases}$$

$$\hat{D}(a,b) = \hat{M}^{a,b}(a^{\text{R}}-1, b^{\text{R}}-1) + \Psi_a^{\mathcal{A}} + \Psi_b^{B} + \sigma(a^{\text{L}},b^{\text{L}}) + \sigma(a^{\text{R}},b^{\text{R}}).$$

The score of the best pair of consensus structure and alignment of A and B is $\hat{M}^{a_\psi b_\psi}(n,m)$; recall that a_ψ and b_ψ denote the pseudo-base pairs covering the entire sequences A and B. The alignment and structures themselves are obtained by traceback.

Notably, these recursions can be evaluated in $O(nm+NM)$ space—i.e. space depends on the respective lengths of A and B and sizes N and M of the sets P and Q of considered base pairs. This space complexity is realized in the same way as in LocARNA: at each time, only one $\hat{M}$ matrix needs to be represented in space, since one matrix $\hat{M}^{a,b}$ recurses only to itself and $\hat{D}$, but does not depend on other $\hat{M}$ matrices. Even the traceback does not require to store all $\hat{M}$ matrices, because recomputing the matrices on the trace is comparably inexpensive.

The time complexity is dominated by computing the $\hat{M}$ matrices. Evaluating a single matrix $\hat{M}$ takes $O(nm+NM)$ time. Because, the straightforward evaluation of PMcomp's recursions computes NM such matrices, this results in $O(nmNM+N^2M^2)$. Assuming a linear number of base pairs in P and Q (as it holds for LocARNA), this yields $O(n^2m^2)$ time.

However, this evaluation strategy would consider certain subsequences repeatedly, namely as prefixes of different loops, albeit their scores are identical: by definition, the matrices $\hat{M}^{a_1b_1}$ and $\hat{M}^{a_2b_2}$ share common entries, if their base pairs share the same left ends, i.e. $a_1^{\text{L}} = a_2^{\text{L}}$ and $b_1^{\text{L}} = b_2^{\text{L}}$; thus, LocARNA combines the computation of such matrices. Although (assuming N and M are linearly bound) this does not change the complexity, it substantially speeds up the computation in practice.

3.2 The PARSE core algorithm

Alignment and folding with the original structure and alignment dependencies of Sankoff requires substantially different recursions. However, we keep the presentation as uniform as possible to our presentation of the PMcomp algorithm. Most obviously, the deletion and insertion of loops requires additional matrices (I_{A}^{ab} and I_{B}^{ab}). More subtly, but centrally, we change the definition of matrix entries for pairs of base pairs. Where the PMcomp algorithm defines $\hat{D}(a,b)$ as the score of a consensus structure and an alignment matching a and b, PARSE requires optimum scores of structure alignment triples of the subsequences between (and excluding) the ends of a and b without assuming the match of those base pairs; these scores are stored in entries $D(a,b)$. We recursively define entries for all $a \in P$, $b \in Q$, i $(a^{\text{L}} < i < a^{\text{R}})$ and k $(b^{\text{L}} < k < b^{\text{R}})$; Figure 2 visualizes these recursions.

$$D(a,b) = \max \begin{cases} M^{ab}(a^{\text{R}}-1, b^{\text{R}}-1) \\ I_{\text{A}}^{ab}(a^{\text{R}}-1) \\ I_{\text{B}}^{ab}(b^{\text{R}}-1) \end{cases}$$

$$M^{ab}(i,k) = \max \begin{cases} M^{ab}(i-1,k-1) + \sigma(i,k) \\ M^{ab}(i,k-1) + \gamma \\ M^{ab}(i-1,k) + \gamma \\ \max\limits_{\substack{a_1 \in P, b_1 \in Q \\ a_1^{\text{R}} = i, b_1^{\text{R}} = k}} \begin{pmatrix} M^{ab}(a_1^{\text{L}}-1, b_1^{\text{L}}-1) \\ + D(a_1,b_1) + \Psi_{a_1}^{\mathcal{A}} + \Psi_{b_1}^{B} \\ + \sigma(a^{\text{L}},b^{\text{L}}) + \sigma(a^{\text{R}},b^{\text{R}}) \end{pmatrix} \end{cases}$$

$$I_{\text{A}}^{ab}(i) = \max \begin{cases} I_{\text{A}}^{ab}(i-1) + \gamma \\ \max\limits_{a_1 \in P, a_1^{\text{R}} = i} (a_1^{\text{L}} - a^{\text{L}} + 1)\gamma + D(a_1,b) + \Psi_{a_1}^{\mathcal{A}} \end{cases}$$

$$I_{\text{B}}^{ab}(k) = \max \begin{cases} I_{\text{B}}^{ab}(k-1) + \gamma \\ \max\limits_{b_1 \in Q, b_1^{\text{R}} = k} (b_1^{\text{L}} - b^{\text{L}} + 1)\gamma + D(a,b_1) + \Psi_{b_1}^{B} \end{cases}$$

The $I_{\text{A}}^{ab}(i)$ [resp. $I_{\text{B}}^{ab}(i)$] matrix correspond to the new case of a loop insertion into A (resp. B). The additional cases and matrices do

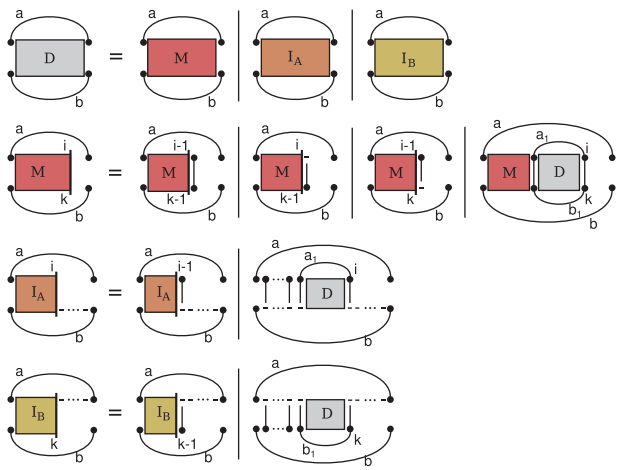

Fig. 2. Recursions of the novel lightweight alignment algorithm PARSE

not add to the time or space complexity over the PMcomp algorithm; in particular, the space and time for the I_A and I_B matrices is dominated by the M matrices. Therefore, analogous arguments let us conclude that, assuming LocARNA-style ensemble-based sparsification, the algorithm runs (like LocARNA) in quadratic space and quartic time.

Affine gap cost

We add the affine gap costs of the previous section without increasing the complexity; in parallel to distinguishing opening costs for base and loop gaps, we apply different gap penalties. First, similar to the algorithm of Gotoh (1982), we introduce matrices with entries $E^{ab}(i,k)$ and $F^{ab}(i,k)$; these contain best scores of structure alignment triples like the entries $M^{ab}(i,k)$, however, constrain the alignments to delete $\mathcal{A}[i]$ and insert $B[k]$, respectively. Thus, we define

$$E^{ab}(i,k) = \max \begin{cases} E^{ab}(i-1,k) + \gamma_{\text{base}} \\ M^{ab}(i-1,k) + \beta_{\text{base}} + \gamma_{\text{base}} \end{cases}$$

for deletion, define $F^{ab}(i,k)$ analogously for insertion, and replace the deletion and insertion cases of the M-recursion of PARSE by $E^{ab}(i,k)$ and $F^{ab}(i,k)$, respectively. Second, we extend the recursions for I_A and I_B. We show the case of I_A, since I_B is analogous. It suffices to make explicit the case of recursing to $I_A^{a_1 b}(a_1^R - 1)$, which extends an already open gap, and add loop gap opening penalty β_{loop} in the general case of recursing to $D(a_1, b)$.

$$I_A^{ab}(i) = \max \begin{cases} I_A^{ab}(i-1) + \gamma_{\text{loop}} \\ \max\limits_{a_1 \in P, a_1^R = i} \begin{pmatrix} (a_1^L - a^L + 1)\gamma_{\text{loop}} \\ + I_A^{a_1 b}(a_1^R - 1) + \Psi_{a_1}^{\mathcal{A}} \end{pmatrix} \\ \max\limits_{a_1 \in P, a_1^R = i} \begin{pmatrix} (a_1^L - a^L + 1)\gamma_{\text{loop}} \\ + D(a_1, b) + \Psi_{a_1}^{\mathcal{A}} + \beta_{\text{loop}} \end{pmatrix} \end{cases}$$

Notably, these extensions do not change the time or space complexity.

3.3 SPARSE—folding and alignment of RNA with ensemble-based sparsification

This section describes the sparsification of PARSE, resulting in SPARSE, which achieves quadratic time complexity. Instead of filling the whole matrix M^{ab} for each pair of base pairs a, b, we are going to skip matrix cells that do not contribute to probable solutions. To define the probable structures and alignments, and thus determine the 'relevant' entries of the matrix, we define several probabilities of structure elements in the structure ensemble of a sequence A. For defining these probabilities, we assume that the structures of an RNA sequence A are Boltzmann-distributed in the structure ensemble, where RNA energy is given by a loop-based energy model.

- $\Pr[(i,j)|\mathcal{A}] = \sum_{(i,j)\ni S} \Pr[S|\mathcal{A}]$ denotes the probability that a structure in the ensemble of A contains the base pair (i, j).
- $\Pr[k \in \text{loop}(i,j)|\mathcal{A}]$, denotes the joint probability that for a structure S in the ensemble of A, (i, j) is a base pair of S or the pseudo base pair *and* k is unpaired in the loop closed by (i, j).
- $\Pr[(i',j') \in \text{loop}(i,j)\mathcal{A}]$, $i < i' < j' < j$, denotes the joint probability that for a structure S in the ensemble of A, (i, j) is base pair of S or the pseudo base pair *and* the ends of (i', j') are in the loop closed by (i, j).

Note that we explicitly include the case that (i, j) is the pseudo base pair a_ψ, which closes the external loop. The base pair probabilities are the immediate outcome of McCaskill's original algorithm (McCaskill, 1990), whereas we calculate the joint probabilities by an extension of this algorithm with the same computational complexity (Otto *et al.*, 2014). Importantly for the complexity of our final algorithm, all these probabilities can be precomputed since they depend only on the single sequences.

3.3.1 Sparse structure and alignment space

The key to sparsifying PARSE, is to optimize only over a subset of solutions, namely probable structures and alignments defined applying fixed probability thresholds θ_1, θ_2 and θ_3 to the above probabilities.

DEFINITION 2 (Sparse Structure and Alignment Space). Assume fixed thresholds $\theta_1, \theta_2, \theta_3 \in [0, 1]$ and sequences A and B. The structure alignment triple $(S, T, \mathcal{A})$ of A and B is contained in the *sparse structure and alignment space* of A and B and, for this reason, called *sparse* iff

(C1)	$\Pr[a\|\mathcal{A}] \geq \theta_1$ *and* $\Pr[b\|B] \geq \theta_1$	*for all base pairs* $a \in S$ *and* $b \in T$
(C2)	$\Pr[i \in \text{loop}(\hat{a})\mathcal{A}] \geq \theta_2$ *and* $\Pr[k \in \text{loop}(\hat{b})B] \geq \theta_2$	*for all* $(i,k) \in \mathcal{A}$, *i unpaired and internal in S, k unpaired and internal in T, where* $\hat{a}$ *denotes the parent of i in S and* $\hat{b}$*, the parent of k in T*
(C3)	$\Pr[a \in \text{loop}(\hat{a})\mathcal{A}] \geq \theta_3$ *and* $\Pr[b \in \text{loop}(\hat{b})B] \geq \theta_3$	*for all internal base pairs a in S and b in T, where* $\hat{a}$ *denotes the parent of a in S;* $\hat{b}$*, the parent of b in T.*

Note that in those definitions, explicitly we do not restrict elements in the external loop by Conditions (C2) and (C3), since this does not improve the final complexity further (see Complexity Analysis), but allows more flexibility. An alternative, which we have chosen in our implementation, is filtering the external bases and base pairs by their probabilities to be external (i.e. treating them as elements of the external loops $\mathcal{A}_\psi$ and B_ψ).

Finding the best structure alignment triple in the sparse space is a form of *constrained evaluation* of the PARSE recursions (see previous section). We modify the recursions such that only cases are considered that extend a sub-solution in a way that satisfies the conditions of

Definition 2. First of all, only matrices for base pairs a and b satisfying Condition (C1) are considered. This restricts the cases of the recursions for M, I_A and I_B that predict base pairs (in S, T or both.) Furthermore, Condition (C2) constrains the base match case of the M-recursion; and Condition (C3), its base pair match case.

Constraining the recursion cases based on these probabilities is possible only because, during the evaluation, one has determined the base pairs closing the loop containing the considered bases and base pairs in the single structures. In contrast, previous PMcomp-like algorithms, e.g. LocARNA (Will *et al.*, 2007), RAF (Do *et al.*, 2008) and FoldAlignM (Torarinsson *et al.*, 2007), keep track of only the consensus structure (i.e. the pairs of base pairs matched to each other). Thus, one cannot apply the appropriate joint probabilities with respect to closing base pairs in the single structures; consequently, the developed techniques are not applicable. Figure 3A illustrates that constraining according to loops in the consensus structure would discard relevant structure alignment triples. In the figure, the base pair a_3 is highly unlikely contained in the loop closed by a_1 (due to stacking, it is much more likely in the loop closed by a_2.) Consequently, even moderate thresholds would disallow its alignment to b_2.

3.3.2 Sparsified evaluation

Constraining the evaluation allows us to represent and fill the matrices only partially for optimizing in the sparse structure and alignment space. Concretely, we do not represent all matrix entries that can be derived only via insertion and deletion cases, since we calculate them from smaller represented entries by adding appropriate gap cost (Fig. 3B).

Definition 3 (Represented entries). *A position i in A is* represented for a base pair $a \in P$ ($a^{\mathrm{L}} < i < a^{\mathrm{R}}$), *iff* $\Pr[i \in \mathrm{loop}(a)|\mathcal{A}] \geq \theta_2$ *or there exists some i'* ($a^{\mathrm{L}} < i' < i$): $(i', i) \in P$ *and* $\Pr[(i', i) \in \mathrm{loop}(a)|\mathcal{A}] \geq \theta_3$. *The position* $i^* := \max\{i' \leq i | i' \text{ is represented for } a\}$ *is called* predecessor of i for a. *The entry* $I_{\mathrm{A}}^{ab}(i)$ *is* represented *iff i is represented for a*; $I_{\mathrm{B}}^{ab}(k)$, *iff k is represented for b; and* $M^{ab}(i, k)$, *iff both conditions hold.*

Lemma 1 (Value of unrepresented entries). Restricting the optimization to the sparse alignment and structure space, an unrepresented entry $M^{ab}(i, k)$ has the value $M^{ab}(i^*, k^*) + (i - i^*)\gamma + (k - k^*)\gamma$, where i^* is the predecessor of i for a and k^* is the predecessor of k for b.

Intuitively, Lemma 1 holds, since the unrepresented entries in a matrix M^{ab} correspond to alignments of subsequences $\mathcal{A}[a^{\mathrm{L}} + 1..i]$ and $B[b^{\mathrm{L}} + 1..k]$ that must end in a gap (because the base match (i, k) and the match of base pairs with right ends i and k are disallowed). Moreover, in such an alignment the entire subsequences $\mathcal{A}[i^* + 1..i]$ and $\mathcal{A}[k^* + 1..\mathrm{k}]$ cannot be aligned. The formal proof of Lemma 1 is given in the appendix.

3.3.3 Complexity analysis

Let us prepare the analysis of SPARSE by first deriving the time complexity of PMcomp, where we apply the weak ensemble-based sparsification of LocARNA, i.e. P and Q contain only base pairs satisfying Condition (C1) of Definition 2. Then, for each position i of A, the number of base pairs $a \in P$ where $a^{\mathrm{L}} = i$ is constantly bounded by $1/\theta_1$; analogously, this holds for Q (Will *et al.*, 2007). Consequently, evaluating each single entry takes constant time; it remains to count the computed entries. Define the number of times each respective position i (or k) in sequence A (or B) is considered in combination with some base of the second sequence in the entire computation. Denote this number of times by $\#_{\mathcal{A}}(i)$. Then, we count

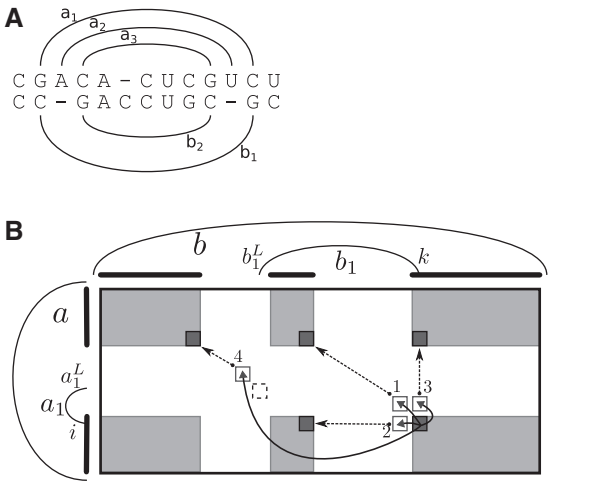

Fig. 3. (A) Example alignment. Due to stacking effects, the probability of a base pair a_3 in the loop closed by a_2 (loop of single structure) is much higher than its probability in the loop closed by a_1 [loop of the consensus structure $\{(a_1, b_1), (a_3, b_2)\}$]. (**B**) Computing represented entries in the sparsified algorithm. We show the matrix M^{ab}; the rounded bars left and on top of the matrix symbolize the represented positions for a and b; the gray areas contain the represented entries for M^{ab}. In our example, the entry $M^{ab}(i, k)$ recurses (solid arrows) to unrepresented entries $M^{ab}(i-1, k-1)$, $M^{ab}(i, k-1)$, $M^{ab}(i-1, k)$ and $M^{ab}(a_1^{\mathrm{L}} - 1, b_1^{\mathrm{L}} - 1)$ (white boxes); the latter via matching base pairs; their left ends correspond to the dashed box at $(a_1^{\mathrm{L}}, b_1^{\mathrm{L}})$. The numbers 1-4 at the arrow heads refer to the respective recursion case. The unrepresented entries are computed from represented entries (dashed arrows to black boxes), each in constant time

the computed entries by $\sum_{i \in [1..n], k \in [1..m]} \#_{\mathcal{A}}(i)\#_B(k)$. Because, $\#_{\mathcal{A}}(i) \in O(n)$ and $\#_B(k) \in O(m)$, we have re-derived the time complexity $O(n^2m^2)$.

Theorem 1 SPARSE optimizes the folding and alignment score in the sparse structure and alignment space in $O(nm)$ time and space.

PROOF: Analogous to the above analysis of PMcomp, we bound the time complexity in the same way from numbers $\#_{\mathcal{A}}^{\mathrm{sp}}(i)$ and $\#_B^{\mathrm{sp}}(k)$, where $\#_{\mathcal{A}}^{\mathrm{sp}}(i)$ is the number of base pairs a in P such that i is represented for a. The sum over all such base pairs a of the terms $\min(\Pr[i \in \mathrm{loop}(a)|\mathcal{A}], \Pr[(i', i) \in \mathrm{loop}(a)\mathcal{A}])$ is smaller or qual 1, since this is a sum of probabilities of disjoint events. Due to the Conditions (C2) and (C3), each term is at least $\min(\theta_2, \theta_3)$, which bounds the number of such base pairs a, i.e. $\#_{\mathcal{A}}^{\mathrm{sp}}(i)$, by $1/\min(\theta_2, \theta_3)$ in $O(1)$. Finally, the complexity is bounded by $\sum_{i \in [1..n], k \in [1..m]} \#_{\mathcal{A}}^{\mathrm{sp}}(i)\#_B^{\mathrm{sp}}(k) \in O(nm)$ for computing all entries $D(a, b)$ and filling $M^{a_\psi b_\psi}$ in $O(nm)$.

3.3.4 Relaxing the problem for further speedup

So far, we have considered finding the best sparse structure alignment triple. In practice, it is usually sufficient to find some (not necessarily sparse) triple that is at least as good as any sparse triple. For solving the relaxed problem, analogously to the optimization of PMcomp, we combine the computation of all matrices $M^{a_1 b_1}, \ldots, M^{a_\ell b_\ell}$, where $a_q^{\mathrm{L}} = i$ and $b_q^{\mathrm{L}} = k$ for $1 \leq q \leq \ell$ and positions i and k. For all these matrices, we consider an entry iff it is a represented entry of any of the matrices. Consequently, each considered entry would have been considered by the first algorithm at least once, but possibly several times for different matrices $M^{a_q b_q}$. Although, thus, we do not increase the complexity, we save

computation time in the latter case. This algorithm searches completely through all sparse structure alignment triples. However, it can not guarantee that the triple is sparse. For example, a solution could match a base that is not represented for a predicted base pair a_1 ($a_1 \in S$), but is represented for an (unpredicted) base pair a_2 ($a_2 \notin S$) that shares the left end with a_1 ($a_1^{\mathrm{L}} = a_2^{\mathrm{L}}$).

3.4 Multiple alignment

To construct multiple alignments, we suggest a progressive alignment pipeline based on the pairwise SPARSE algorithm. There, we apply SPARSE to construct the guide tree and to align profiles in each progressive step. Unlike PMmulti, the multiple alignment extension to PMcomp (Hofacker *et al.*, 2004), which otherwise applies a similar strategy, we apply RNAalifold (Hofacker *et al.*, 2002; Bernhart *et al.*, 2008) to compute 'profile' consensus dot plots in the progressive phase.

For aligning k sequences with maximal length n, we start by computing pairwise alignments between all pairs of the input sequences. Because the required ensemble probabilities depend only on the single sequences, they are precomputed for each sequence. Thus, all pairwise alignments are performed in $O(kn^3 + k^2n^2)$ time. Then, the pipeline constructs a guide tree in $O(k^2)$ time by UPGMA (Gronau and Moran, 2007). In each step of the progressive alignment, RNAalifold computes all required ensemble probabilities in $O(n^3)$ and SPARSE calculates an alignment in $O(n^2)$ (We extended RNAalifold to compute the additional joint probabilities without changing the complexity.). Thus, the progressive alignment takes $O(kn^3 + kn^2)$ time, resulting in total time $O(kn^3 + k^2n^2)$. In comparison, the corresponding pipeline for LocARNA requires $O(kn^3 + kn^4)$ time.

4 Results

4.1 Speedup and alignment quality

We evaluated our implementation of SPARSE on the Bralibase 2.1 (Wilm *et al.*, 2006) benchmark sets k2 and k3, which consist of pairwise and three-way alignments, respectively. We compared SPARSE to LocARNA and RAF; For set k2, which contains only short and medium-sized RNAs of ~110 nt average length, SPARSE and RAF achieve similar speed ups over LocARNA (Table 1) (For SPARSE, we used these values: $\theta_1 = 1\text{e-}3$, $\theta_2 = 5\text{e-}5$, $\theta_3 = 1\text{e-}4$, $\beta_{base} = -900$, $\gamma_{base} = -3500$, $\beta_{loop} = -900$ and $\gamma_{loop} = -350$; for LocARNA and RAF, we used default settings.). For each alignment tool, Figure 4 shows the dependency of alignment quality on sequence identity across k2; for each benchmark instance, the quality is measured as similarity to the Rfam-derived reference alignment reported as sum-of-pairs score (SPS) by `compalignp` (Wilm *et al.*, 2006). The dependencies are estimated by non-parametric regression (lowess; Cleveland, 1981). This resulting curve visualizes the approximate average SPS at each sequence identity. To furthermore visualize the distribution of the SPS values, we iterated the lowess method both on the elements above and below of the main lowess curve. LocARNA and SPARSE show qualitatively similar performance; across the entire range of sequence identities, we observe a largely constant quality offset, where both tools maintain a high alignment quality even for low sequence identities. In contrast, the quality of RAF alignment drops dramatically when sequence similarity decreases; we conjecture that this is a consequence of the strong sequence-based heuristics in RAF. Our benchmarks on k3 (Supplementary Fig. S1) suggest that this behavior extends to multiple alignment. Although all RNA comparison tools introduce some form of time-quality trade-off, remarkably, in terms of alignment quality SPARSE and RAF behave strongly different at very similar run times.

Table 1. Total run-time and speed up of pairwise alignments due to sparsification across Bralibase 2.1 set k2

Tool	Total time (s)	Mean time (s)	Speedup (versus LocARNA)
LocARNA	13400	1.49	1.0
SPARSE	3600	0.40	3.7
RAF	3200	0.36	4.2

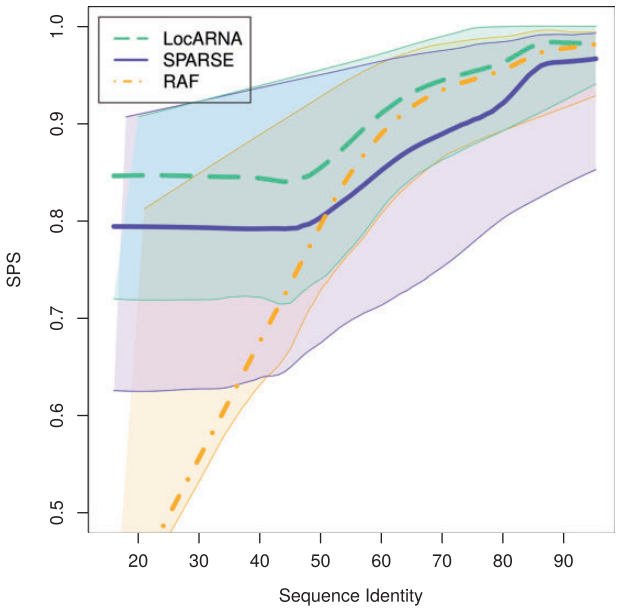

Fig. 4. Alignment quality (measured by SPS) at different sequence identities for pairwise alignments (Bralibase 2.1 set k2). The curves are lowess curves (Cleveland, 1981) through data points for each benchmark instance. The thin lines visualize the distribution of scores by estimating the respective instance averages above and below of the main lowess curve

4.2 The SPARSE model improves folding

Moreover, we studied the effect of the novelties in the SPARSE alignment model. Recall that only SPARSE maintains the full flexibility of Sankoff's approach in the lightweight model, while all previous methods restrict Sankoff's model by disallowing loop insertions and deletions. Furthermore, the sparsification of SPARSE is expected to affect structure prediction (cf. Fig. 3A.) Comparing our implementations of SPARSE and LocARNA enables isolating these effects, since by design these implementations behave as similar as possible otherwise. The quality of each predicted structure is measured as Matthews Correlation Coefficient (MCC; Matthews, 1975) relative to the Rfam-derived reference structure. For each sequence of the benchmark, we derive reference structures from Rfam, by constrained folding of the associated Rfam consensus structure of the sequence. We compute MCC values for predictions by SPARSE and LocARNA across k2. Figure 5 compares the structure prediction quality by SPARSE and LocARNA. Visualized by the non-overlapping notches, there is strong evidence for improvements (Chambers *et al.*, 1983) across all sequence identity ranges covered by the benchmark set. For example, these effects are illustrated well by RNAs of the family gcvT (Appendix).

Although SPARSE improved prediction quality and speed over LocARNA, these results suggest even more general conclusions. That is, the improvements can be directly ascribed to the single differences of the tools: sparsification and model flexibility.

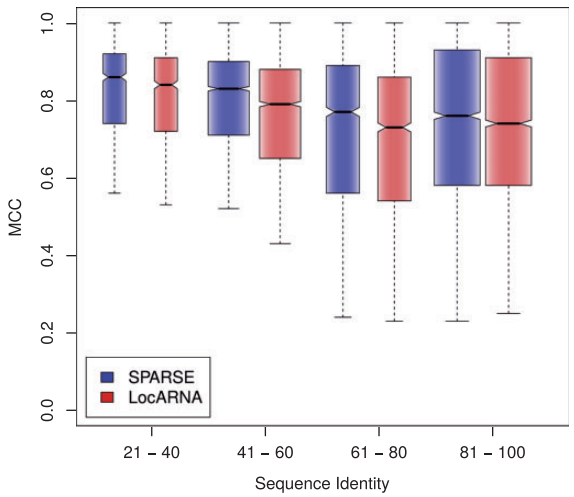

Fig. 5. Structure prediction quality measured by MCC within different ranges of average pairwise sequence identity (APSI) shown as boxplots. (Bralibase 2.1 set k2) whiskers are extended up to one interquartile range from the boxes

5 Discussion

We presented a novel method for simultaneous alignment and folding of RNAs. The relevance of this method is 2-fold. First, we developed the first *full-featured* lightweight variant of the Sankoff model. This fundamentally improves over the lightweight model of PMcomp. Because this model drastically lowered the computational burden of simultaneous alignment and folding, it has been adopted by many successful RNA alignment approaches. However, all of these methods lack the full flexibility of the Sankoff model; for the first time, SPARSE combines this flexibility with lightweight computation. Second, we present a novel method to speed up Sankoff-style alignment that is purely based on the structure ensemble of RNAs; in particular, it does not have to resort to sequence-based heuristics, which could compromise the alignment quality. We showed that by sparsification based on ensemble probabilities of unpaired bases and base pairs in specific loops, simultaneous alignment and folding requires only quadratic time.

Performing Bralibase 2.1 benchmarks, we demonstrated that, also in practice, the method provides a profound speed up. At similar speed as one of the fastest known simultaneous alignment and folding tools RAF, SPARSE maintains high alignment and folding quality for the 'twilight' zone of RNAs with low sequence identity, which are particularly hard to align. Finally, these benchmarks (and concrete examples, see Appendix) suggest that the added expressivity of the novel lightweight model improves the folding accuracy.

Acknowledgements

We would like to thank Niklas Meinzer for implementing a benchmark tool for our evaluation and the anonymous reviewers for their help to improve this article.

Funding

This work was partially supported by the German Research Foundation (DFG grants BA 2168/3-3, BA 2168/4-3 SPP 1395 InKoMBio, MO 2402/1-1) and German Federal Ministry of Education and Research (BMBF grant 031 6165A e:Bio RNAsys) to R.B.

Conflict of Interest: none declared.

References

Bernhart,S.H. *et al.* (2008) RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, **9**, 474.

Chambers,J.M. *et al.* (1983) *Graphical Methods for Data Analysis*. Wadsworth, Belmont, CA.

Clark,M.B. *et al.* (2011) The reality of pervasive transcription. *PLoS Biol*, **9**, e1000625; discussion e1001102.

Cleveland,W.S. (1981) Lowess: a program for smoothing scatterplots by robust locally weighted regression. *Am. Stat.*, **35**, 54.

Do,C.B. *et al.* (2008) A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics*, **24**, i68–i76.

Dowell,R.D. and Eddy,S.R. (2006) Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinformatics*, **7**, 400.

Gardner,P.P. *et al.* (2005) A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Res.*, **33**, 2433–2439.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

Gronau,I. and Moran,S. (2007) Optimal implementations of UPGMA and other common clustering algorithms. *Inf. Process. Lett.*, **104**, 205–210.

Harmanci,A.O. *et al.* (2007) Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign. *BMC Bioinformatics*, **8**, 130.

Hofacker,I.L. *et al.* (2002) Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, **319**, 1059–1066.

Hofacker,I.L. *et al.* (2004) Alignment of RNA base pairing probability matrices. *Bioinformatics*, **20**, 2222–2227.

Holmes,I. (2005) Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*, **6**, 73.

Jacquier,A. (2009) The complex eukaryotic transcriptome: unexpected pervasive transcription and novel small RNAs. *Nat. Rev. Genet.*, **10**, 833–844.

Mathews,D. *et al.* (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.

Matthews,B. (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochem. Biophys. Acta*, **405**, 442–451.

McCaskill,J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

Otto,C. *et al.* (2014) ExpaRNA-P: simultaneous exact pattern matching and folding of RNAs. *BMC Bioinformatics*, **15**, 6602.

Parker,B.J. *et al.* (2011) New families of human regulatory RNA structures identified by comparative analysis of vertebrate genomes. *Genome Res.* **21**, 1929–1943.

Rederstorff,M. *et al.* (2010) RNPomics: defining the ncRNA transcriptome by cDNA library generation from ribonucleo-protein particles. *Nucleic Acids Res.*, **38**, e113.

Saito,Y. *et al.* (2011) Fast and accurate clustering of noncoding RNAs using ensembles of sequence alignments and secondary structures. *BMC Bioinformatics*, **12** (Suppl 1), S48.

Sankoff,D. (1985) Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, **45**, 810–825.

Shi,Y. *et al.* (2009) Metatranscriptomics reveals unique microbial small RNAs in the ocean's water column. *Nature*, **459**, 266–269.

Sorescu,D.A. *et al.* (2012) CARNA—alignment of RNA structure ensembles. *Nucleic Acids Res.*, **40**, W49–W53. DAS, MMö, and MMa contributed equally to this work.

Torarinsson,E. *et al.* (2007) Multiple structural alignment and clustering of RNA sequences. *Bioinformatics*, **23**, 926–932.

Tseng,H.-H. *et al.* (2009) Finding non-coding RNAs through genome-scale clustering. *J. Bioinform. Comput. Biol.*, **7**, 373–388.

Will,S. *et al.* (2007) Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, **3**, e65.

Will,S. *et al.* (2012) LocARNA-P: accurate boundary prediction and improved detection of structural RNAs. *RNA*, **18**, 900–914.

Wilm,A. *et al.* (2006) An enhanced RNA alignment benchmark for sequence alignment programs. *Algorithms Mol. Biol.*, **1**, 19.