Exact Pattern Matching for RNA Structure Ensembles

Christina Schmiedl[1,⋆], Mathias Möhl[1,⋆], Steffen Heyne[1,⋆], Mika Amit[2], Gad M. Landau[2,3], Sebastian Will[1,4,⋆⋆], and Rolf Backofen[1,5,⋆⋆]

[1] Bioinformatics, Institute of Computer Science, Albert-Ludwigs-Universität, Freiburg, Germany, {will,backofen}@informatik.uni-freiburg.de
[2] Department of Computer Science, University of Haifa, Haifa, Israel
[3] Department of Computer Science and Engineering, NYU-Poly, Brooklyn NY, USA
[4] CSAIL and Mathematics Department, MIT, Cambridge MA, USA
[5] Center for Biological Signaling Studies (BIOSS), Albert-Ludwigs-Universität, Freiburg, Germany

Abstract. ExpaRNA's core algorithm computes, for two fixed RNA structures, a maximal non-overlapping set of maximal exact matchings. We introduce an algorithm ExpaRNA-P that solves the lifted problem of finding such sets of exact matchings in entire Boltzmann-distributed structure ensembles of two RNAs. Due to a novel kind of structural sparsification, the new algorithm maintains the time and space complexity of the algorithm for fixed input structures. Furthermore, we generalized the chaining algorithm of ExpaRNA in order to compute a compatible subset of ExpaRNA-P's exact matchings. We show that ExpaRNA-P outperforms ExpaRNA in BRAliBase 2.1 benchmarks, where we pass the chained exact matchings as anchor constraints to the RNA alignment tool LocARNA. Compared to LocARNA, this novel approach shows similar accuracy but is six times faster.

1 Introduction

Genome-wide transcriptomics [1–3] provides evidence for massive transcription in eukaryotic genomes, going as far as suggesting that most of both genomic strands of human might be transcribed [4]. Most of these transcripts do not code for proteins; furthermore, it has become clear that the majority of them perform primarily regulatory functions [5]. Thus, these RNAs play a crucial role in the living cell. However, their functional annotation is strongly lagging behind; reliable automated annotation pipelines exist only for subclasses of ncRNAs such as tRNAs, microRNAs or snoRNAs [6].

Regulatory RNAs are often structural, and their secondary structures are then usually well-conserved due to their functional importance. This fact is used by a priori RNA-gene finders like QRNA [7], RNAz [8], and Evofold [9],

⋆ Joint first authors
⋆⋆ Joint corresponding authors

which detect conserved RNA-structures in whole genome alignments. More recently, a strategy towards the automatic annotation of non-coding RNAs has emerged, which identifies RNAs with similar sequence and common secondary structure [10–12] on a genomic scale. This can be used to determine remote members of RNA-*families* as defined in the Rfam-database, or to determine new RNA-*classes* of structural, and hence likely functionally similar, ncRNAs by using clustering approaches. In Rfam, the term RNA-*clan* has been introduced for a collection of RNA-families that share similar structure but little sequence conservation. Prominent examples of RNA-*classes* are microRNAs or snoRNAs.

Albeit this approach is appealing, a wide-spread, or even automated, application of these methods has been hindered by the huge complexity of the underlying sequence/structure alignment approach for detecting similarity in both sequence *and* structure. The first practical approaches for multiple structural alignment, such as `RNAforester` [13] and `MARNA` [14], depend on predicted or known secondary structures. In practice, however, these approaches are limited by the low accuracy of non-comparative structure prediction. Sankoff's algorithm [15] provides a general solution to the problem of simultaneously computing an alignment and the common secondary structure of two aligned sequences. In its full form, the problem requires $O(n^6)$ CPU time and $O(n^4)$ memory (for RNA sequences of length n). This complexity is already limiting for most practical problems such as routinely scanning remote members of RNA-*families*. For detecting novel RNA-*classes* in the plethora of newly discovered RNA-transcripts, this complexity becomes plainly prohibitive, since this task requires clustering based on quadratically many all-against-all pairwise RNA comparisons.

For that reason, many variants of the Sankoff algorithm with different optimizations have been introduced. `FoldAlign` [16] and `dynalign` [17] implement a full energy model for RNA that is evaluated during the alignment computation. In contrast, `PMcomp` [18] and LocARNA [10] use a lightweight energy model, which assignes energies to single base pairs. This simplification reduces the computational cost significantly. They achieve their accuracy by precomputing the energy contributions of base pairs from their probabilities in a full-featured energy model [19]. Whereas the approaches [20, 16, 17, 21] have to compensate their computational demands by strong, often sequence-based, heuristics, LocARNA [10] takes advantage of structural sparsity in the RNA structure ensembles to reduce its complexity to $O(n^4)$ time and $O(n^2)$ space. This successful approach is consequently found in other Sankoff-like methods [22–24].

We introduce a strategy that reduces the computational demands further, but differs fundamentally from heuristic improvements, like [24], that restrict the search space based on sequence alignments. It computes the sequence-structure-conserved elements that form highly probable local substructures in the RNA structure ensemble of both input RNAs; subsequently, these elements are used as anchor constraints in a full sequence-structure alignment by LocARNA. In [25], we have proposed a similar strategy, which computes conserved elements in pairs of fixed RNA secondary structures, based on an algorithm with quadratic time and space complexity [26]. Albeit this approach reduces the overall computa-

tion time significantly, it faces similar problems as the first generation of RNA aligment methods [13, 14], due to the use of a single predicted input structure for each sequence. Since predicting minimum free energy (MFE) structures from single-sequences is unreliable, this strategy fails frequently and causes severe misalignments.

Overcoming the problems of the previous approach, the novel algorithm for determining exact sequence-structure patterns is based on probabilities in the RNA structure ensembles. We point out that a straight-forward extension of the fixed input structure algorithm to RNA structure ensembles, would result in a complexity of $O(n^4)$ time and $O(n^2)$ space. This complexity is as high as the one of LocARNA, which would nullify the benefits of exact matching.

Thus, our main technical contribution is to solve the ensemble based problem in quadratic time and space; the advancement is comparable to the leap from first generation RNA alignment to efficient Sankoff-style alignment. For this achievement, we introduce a method of sparsification that uses the ensemble properties of the input sequences. Previous sparsification approaches reduced the number of computations required for each entry [27–31] or the number of matrices to be considered [10, 22]. In addition, we identify sparse regions of each matrix *a priori* such that, in total, only quadratically many entries remain; each of these entries is calculated in constant time. The *a priori* identification of sparse regions is based on the joint probability that a sequence position occurs as part of a particular loop. Since the sum of these probabilities is bound by one, we can control the complexity on a global scale by setting a probability threshold. As a further benefit over sparsification methods that filter non-optimal solutions [27–31], our sparsification allows us to enumerate suboptimal solutions.

To evaluate the practical benefits of these algorithmic innovations, we devise a novel pipeline ExpLoc-P for sequence-structure alignment. In its first stage, it enumerates suboptimal exact matchings of local sequence-structure patterns due to the introduced algorithm ExpaRNA-P. In the second stage, the suboptimal matchings are chained to select an optimal subset of compatible matchings that can simultaneously occur in an alignment of RNAs. Finally, these matchings are utilized as anchor constraints in a subsequent Sankoff-style alignment by LocARNA. In benchmarks on BRAliBase 2.1, ExpLoc-P's accuracy is comparable to the unconstrained LocARNA, although it is six times faster.

2 Preliminaries

A RNA sequence A is a string over the alphabet $\{A, C, G, U\}$, the base at the i-th position of A is denoted by A_i, the subsequence from position i to j by $A_{i..j}$ and the length of it by $|A|$. A structure of A is a set P_A of base pairs $p = (i, j)$ such that $1 \leq i < j \leq |A|$, where A_i and A_j form a complementary Watson-Crick base pair (A-U or C-G) or a non-standard base pair G-U. We denote the left end i of p by p^L and the right end j by p^R. For a single structure, we also assume that each sequence position is involved in at most one base pair (for all

$(i,j),(i',j') \in P_A$: $(i = i' \Leftrightarrow j = j')$ and $i \neq j'$) and base pairs do not cross (there are no $(i,j),(i',j') \in P_A$ with $i < i' < j < j'$).

Since non-crossing RNA structures correspond to trees, we define, for any position k of A, the *parent of* k as the $(i,j) \in P_A$ with $i < k < j$ such that there does not exist any $(i',j') \in P_A$ with $i < i' < k < j' < j$. Analogously, the *parent of a base pair* (i,j) is the parent of i (which is also the parent of j). Intuitively, if a base or base pair has a parent (i,j), it is located in the loop closed by (i,j). For external positions k that are not included in any loop, we define the parent to be an additional imaginary base pair $(0, |A|+1)$ covering the entire sequence.

3 Exact Pattern Matchings in RNA Structure Ensembles

In this section, we formalize the problem that our algorithm ExpaRNA-P solves. We fix sequences A and B.

Definition 1 (EPM). *An Exact Pattern Matching (EPM) is a tuple* $(\mathcal{M}, \mathcal{S})$ *with* $\mathcal{M} \subseteq \{(i \sim k) | i \in \{1,\dots,|A|\}, k \in \{1,\dots,|B|\}\}$ *and* $\mathcal{S} \subseteq \{(ij \sim kl) | (i,j) \in \{1,\dots,|A|\}^2, i < j, (k,l) \in \{1,\dots,|B|\}^2, k < l\}$ *such that*

- *for all* $(i \sim k) \in \mathcal{M} : A_i = B_k$
- *for all* $(i \sim k), (j \sim l) \in \mathcal{M} : (i < j \Rightarrow k < l \wedge i = j \Leftrightarrow k = l)$
- $(ij \sim kl) \in \mathcal{S} \Rightarrow \{(i \sim k), (j \sim l)\} \subseteq \mathcal{M}$
- *the structure* $\{(i,j) | (ij \sim kl) \in \mathcal{S}\}$ *is non-crossing (with the previous condition this implies* $\{(k,l) | (ij \sim kl) \in \mathcal{S}\}$ *is non-crossing).*
- *the matching is connected on the sequence or structure level, i.e. the graph* $(\mathcal{M}, E)$ *with* $E = \{(i \sim k, j \sim l) | (i = j+1$ *and* $k = l+1)$ *or* $(ij \sim kl) \in \mathcal{S}\}$ *is (weakly) connected.*

EPMs are exact matches that are not necessarily contiguous subsequences, but structure-local in the sense of [32, 33]. We define the set of structure matches as $\mathcal{M}|_{\mathcal{S}} := \{(i \sim k), (j \sim l) | (ij \sim kl) \in \mathcal{S}\}$. Note that the correspondence between nested RNA structures and trees naturally generalizes to EPMs. Therefore, we define the parent of some element of $\mathcal{M} \cup \mathcal{S}$ as

$$\text{parent}_{\mathcal{S}}(i \sim k) = \underset{(i'j' \sim k'l') \in \mathcal{S} \cup \{(0|A|+1 \sim 0|B|+1)\}, i' \leq i \leq j'}{\text{argmin}} |j' - i'| \tag{1}$$

$$\text{parent}_{\mathcal{S}}(ij \sim kl) = \underset{(i'j' \sim k'l') \in \mathcal{S} \cup \{(0|A|+1 \sim 0|B|+1)\}, i' < i < j < j'}{\text{argmin}} |j' - i'| \tag{2}$$

Note that every matched element that is not enclosed by matched base pairs has the *pseudo-parent* $(0|A|+1 \sim 0|B|+1)$ which is best understood as additional match of pseudo-base pairs outside of the two sequences. Also note that for $\text{parent}_{\mathcal{S}}(ij \sim kl) \in \mathcal{S}$ $\text{parent}_{\mathcal{S}}(ij \sim kl) \neq \text{parent}_{\mathcal{S}}(i \sim k) = \text{parent}_{\mathcal{S}}(j \sim l) = (ij \sim kl)$.

Since we only want to match structures that are probable in the ensemble of the given sequences, we define the notion of significant EPMs. Considering only significant EPMs is crucial for both the quality of the results and the complexity of the algorithm. To define significant EPMs we consider the following probabilities over the Boltzmann ensemble of structures.

- $\Pr\{(i,j)|X\}$ denotes the probability, that a structure in the ensemble of $X \in \{A, B\}$ contains the base pair (i, j),
- $\Pr^{\text{loop}}_{(i,j)}(k|X)$ denotes for $i < k < j$ and $X \in \{A, B\}$ the joint probability that the structure of X contains the base pair (i, j) and the unpaired base k such that (i, j) is the parent of k.
- $\Pr^{\text{loop}}_{(i,j)}((i', j')|X)$ denotes for $i < i' < j' < j$ and $X \in \{A, B\}$ the joint probability that the structure of X contains the base pairs (i, j) and (i', j') and that (i, j) is the parent of (i', j').

In the special case where $(i, j) = (0, |A| + 1)$ we define $\Pr^{\text{loop}}_{(0,|A|+1)}((i', j')|X) := \Pr\{(i', j')|X\}$ and $\Pr^{\text{loop}}_{(0,|A|+1)}(k|X)$ as the probability that base k of X is unpaired, i.e. $1 - \sum_{j<i} \Pr\{(j, i)|X\} - \sum_{i<j} \Pr\{(i, j)|X\}$.[6] In Sec. 4 we show how to compute the probabilities efficiently.

For significant EPMs we introduce three different thresholds θ_1, θ_2 and θ_3. We require that all matched base pairs have a probability of at least θ_1 and that the probabilities of all matched unpaired bases and matched base pairs to occur as part of the loop of their respective parent is at least θ_2 and θ_3, respectively.

Definition 2 (significant EPM). *Given the thresholds θ_1, θ_2, θ_3, an EPM is significant iff*

- *for all* $(ij \sim kl) \in \mathcal{S}$*:* $\Pr\{(i, j)|A\} \geq \theta_1$ *and* $\Pr\{(k, l)|B\} \geq \theta_1$
- *for all* $(i \sim k) \in \mathcal{M} \setminus \mathcal{M}|_{\mathcal{S}}$ *with* $(i'j' \sim k'l') = \text{parent}_{\mathcal{S}}(i \sim k)$*:*
  $\Pr^{loop}_{(i',j')}(i|A) \geq \theta_2$ *and* $\Pr^{loop}_{(k',l')}(k|B) \geq \theta_2$
- *for all* $(ij \sim kl) \in \mathcal{S}$ *with* $(i'j' \sim k'l') = \text{parent}_{\mathcal{S}}(ij \sim kl)$*:*
  $\Pr^{loop}_{(i',j')}((i, j)|A) \geq \theta_3$ *and* $\Pr^{loop}_{(k',l')}((k, l)|B) \geq \theta_3$

The score of an EPM $(\mathcal{M}, \mathcal{S})$ consists of a score $\sigma(i, k)$ for each pair of matched unpaired bases and $\tau(i, j, k, l)$ for each pair of matched base pairs:

$$\text{score}(\mathcal{M}, \mathcal{S}) = \sum_{(i\sim k)\in\mathcal{M}\setminus\mathcal{M}|_{\mathcal{S}}} \sigma(i, k) + \sum_{(ij\sim kl)\in\mathcal{S}} \tau(i, j, k, l) \tag{3}$$

The algorithm described in this paper determines all significant maximally extended EPMs up to a certain score threshold, where maximally extended is defined as follows.

Definition 3 (maximally extended EPMs). *An EPMs $(\mathcal{M}, \mathcal{S})$ is maximally extended, if there does not exist any $(\mathcal{M}', \mathcal{S}')$ with $\mathcal{M} \subset \mathcal{M}'$, $\mathcal{S} \subseteq \mathcal{S}'$ and such that for all $(i \sim k) \in \mathcal{M}$ $\text{parent}_{\mathcal{S}}(i \sim k) = \text{parent}_{\mathcal{S}'}(i \sim k)$.*

As shown in Fig. 1, the last condition of this definition is required to ensure that we consider EPMs with different structures as being different. Due to this

[6] Note that these probabilities include the cases where (i, j) or k are covered by some base pair. This is reasonable as the EPMs are structurally local; thus, they can be enclosed by other structure or be external.

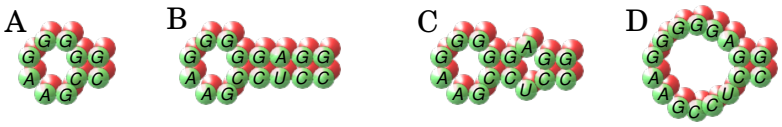

Fig. 1. EPM A is not maximally extended if there exists a larger EPM like B or C. EPMs B, C, and D can all be maximally extended simultaneously since in each case some base matches have different parents.

definition, the set of maximally extended EPMs does not contain proper substructures, such as Fig. 1A depicts a proper substructure of the EPM of Fig. 1B, but contains structural variants of the same set of matched positions. We select a relevant subset of the structural variants in the answer set of our algorithm by considering only significant EPMs.

4 The Algorithm ExpaRNA-P

Precomputing Likely Loops In a preprocessing step, we compute, separately for each sequence, the probabilities required to determine the significant EPMs. Hence, in clustering scenarios, for example, where all pairs from a set of sequences need to be matched, this preprocessing needs to be done only once for each sequence and not for all quadratically many pairs. To simplify notation, we show how to compute the probabilities for A, the computation for B is identical. While the base pair probabilities $\Pr\{(i,j)|A\}$ are computed by McCaskill's algorithm [19], we extend this algorithm to compute the probabilities $\Pr^{\text{loop}}_{(i,j)}(k|A)$ and $\Pr^{\text{loop}}_{(i,j)}((i',j')|A)$. For this purpose, we utilize the matrices Q_{ij}, Q^b_{ij}, Q^m_{ij}, and Q^{m1}_{ij} of McCaskill's algorithm (details can be found in [19]). For $1 \leq i \leq j \leq |S|$, the entries of these matrices represent the sum over the Boltzmann weights of the following set of structures of $A_{i..j}$

- Q_{ij}: all structures of $A_{i..j}$
- Q^b_{ij}: all structures P of $A_{i..j}$ with $(i,j) \in P$
- Q^m_{ij}: all non-empty structures of $A_{i..j}$ scored as part of a multiloop
- Q^{m1}_{ij}: all structures P of $A_{i..j}$, scored as part of a multiple loop, such that for some k holds $(i,k) \in P$ and for all $(i',j') \in P$ holds $i \leq i' < j' \leq k$.

Intuitively Q^{m1}_{ij} counts the Boltzmann weights of all structures that are part of a multiloop and have exactly one outermost base pair, starting at position i. In addition to the classical McCaskill matrices, we compute a matrix $Q^{m2}_{ij} = \sum_{i<k<j-1} Q^m_{ik} Q^{m1}_{k+1\,j}$, representing parts of a multiloop with at least two outermost base pairs.

Given those matrices, we compute $\Pr^{\text{loop}}_{(i,j)}(k|A)$ as

$$\Pr^{\text{loop}}_{(i,j)}(k|A) = \Pr\{(i,j)|A\} \frac{H + I + M}{Q^b_{ij}}, \text{ where} \tag{4}$$

$$H = \exp(-\beta F_1(i,j)) \tag{5}$$

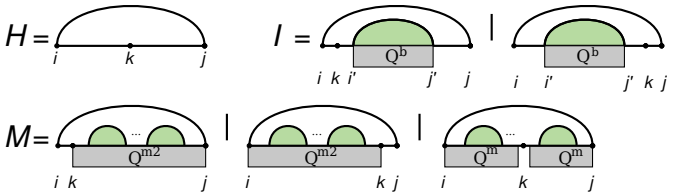

Fig. 2. Recursions of the partition functions for computing the probability that a position k occurs inside a hairpin loop (H), interior loop (I), or multiloop (M) closed by a base pair (i,j).

$$I = \sum_{\substack{i',j' \\ i<k<i'<j'<j}} \exp(-\beta F_2(i,j,i',j'))Q^b_{i'j'} + \sum_{\substack{i',j' \\ i<i'<j'<k<j}} \exp(-\beta F_2(i,j,i',j'))Q^b_{i'j'} \quad (6)$$

$$M = Q^{m2}_{k+1j-1}\exp(-\beta(a+(k-i)c)) + Q^{m2}_{i+1k-1}\exp(-\beta(a+(j-k)c)) \quad (7)$$
$$+ Q^m_{i+1k-1}Q^m_{k+1j}\exp(-\beta(a+c))$$

The formulas are visualized in Fig. 2. H, I, and M represent the cases where k is contained in a hairpin, interior loop, and multiloop, respectively. The constants a, c, k, and T and the energy functions F_1 and F_2 are defined as in McCaskill [19], where $\beta := (kT)^{-1}$. The three sums in the computation of M cover the cases where k is in the leftmost, the rightmost, and any other unpaired region of the loop, respectively. Note that Q^{m2} is required to ensure, without increasing complexity, that M considers only multiloops (with at least two inner base pairs).

In a similar way we compute $\Pr^{\text{loop}}_{(i,j)}((i',j')|A)$. All these joint probabilities are computed within the same asymptotic complexity as the McCaskill algorithm.

Computing the Significant EPMs The algorithm computes table entries $D((ij),(kl))$, which store the best EPM enclosed by each base pair match $(ij \sim kl)$. More precisely, $D((ij),(kl))$ has entries for each $(i,j) \in P_A$ and $(k,l) \in P_B$ with $\Pr\{(i,j)|A\} \geq \theta_1$ and $\Pr\{(k,l)|B\} \geq \theta_1$ and $D((ij),(kl))$ denotes the maximum score of a significant EPM $(\mathcal{M},\mathcal{S})$ of $A_{i..j}$ and $B_{k..l}$ with $(ij \sim kl) \in \mathcal{S}$. The entries of D are computed in increasing order with respect to their size such that during the computation of some $D((ij),(kl))$ any $D((i'j'),(k'l'))$ with $i < i' < j' < j$ and $k < k' < l' < l$ is already computed. For the computation of each $D((ij),(kl))$ we compute matrices $L^{ijkl}(j',l')$, $G_A^{ijkl}(j',l')$, $G_{AB}^{ijkl}(j',l')$, and $LR^{ijkl}(j',l')$. These matrices contain entries for (j',l') with $i < j' < j$, $k < l' < l$. In Sec. 4, we argue that the matrices are sparse.

Intuitively, we use the matrices L, G_A, G_{AB}, and LR to compute a matching of the loops below (i,j) and (k,l) by matching bases and closed substructures from left to right. We start matching from the left using L which represents the part of the matching that is connected to the left ends i, k of the base pairs. Then at some point we are allowed to introduce a gap in both sequences using matrices G_A and G_{AB} and then start matching the part that is connected to the right ends j, l of the base pairs using matrix LR.

The matrices are computed according to the recursions visualized in Fig. 3. The base cases are $L^{ijkl}(i,k) = 0$, $L^{ijkl}(j',k) = -\infty$ for all $j' > i$, $L^{ijkl}(i,l') = -\infty$ for all $l' > k$, $LR^{ijkl}(i,l') = G_A^{ijkl}(i,l') = G_{AB}^{ijkl}(i,l') = 0$ for all $l' > k$ and $LR^{ijkl}(j',k) = G_A^{ijkl}(j',k) = G_{AB}^{ijkl}(j',k) = 0$ for all $j' > i$. Intuitively, the recursion for L always matches the last positions j' and l' or assigns $-\infty$ if they don't match. Left of this match of the last positions can either be a matched unpaired position of the loops (second case) or a match of two base pairs (third case). The recursion for LR is analogous to L except that it considers the additional case that the gap has just been ended at positions $j'-1$ and $l'-1$. The gap itself, computed in G_A and G_{AB}, simply allows to skip once an arbitrary number of positions in both sequences when going from the left matched part to the right matched part. To avoid ambiguity, the recursion enforces to first skip the positions in A (using G_A) and after that the positions of B (using G_{AB}). This is necessary for the suboptimal traceback which would otherwise enumerate the same solutions more than once. Also note that in the computation of $D((ij),(kl))$ not only LR^{ijkl} but also L^{ijkl} is considered. Here, LR^{ijkl} represents the situations where the best EPM contains a gap, and L^{ijkl} the situation where the best matching has no gap, i.e. the parts matched at the left and right ends are connected.

After the matrix D has been computed, a final matrix F is computed where for $0 \leq j' \leq |A|$ and $0 \leq l' \leq |B|$ each $F(j',l')$ denotes the maximum score of a significant EPM of $A_{1..j'}$ and $B_{1..l'}$ which ends at (j',l') (i.e. with $(j \sim l) \in \mathcal{M}$). The base cases are $F(j',0) = F(0,l') = 0$ for all j', l'. The recursion for F (Fig. 3) is almost identical to the recursion for L, except for the first case, which is 0 instead of $-\infty$, since the EPMs in F are (similar to local sequence alignments) allowed to start at any point. Also, since the base pairs of F are external (i.e. not enclosed by some other base pair of the EPM), the check for the second and third condition of significant EPMs (Def. 2) are discarded.

The suboptimal maximally extended EPMs are obtained by doing standard suboptimal tracebacks enumerating all EPMs up to a given score threshold. Since the recursions are all unambiguous (i.e. the cases do not overlap) no EPM is enumerated more than once. To enumerate only maximally extended EPMs, we start tracebacks only from entries $F(j',l')$ for which $A_{j'+1} \neq B_{l'+1}$.

Lemma 1. *A maximally extended EPM $(\mathcal{M},\mathcal{S})$ of $A_{1..j'}$ and $B_{1..l'}$ with $(j' \sim l') \in \mathcal{M}$ is also a maximally extended EPM of A and B, iff $A_{j'+1} \neq B_{l'+1}$.*

Proof. Obviously, if $A_{j'+1} = B_{l'+1}$ the larger EPM $(\mathcal{M} \cup \{(j'+1 \sim l'+1)\}, \mathcal{S})$ satisfies the condition of $(\mathcal{M}',\mathcal{S}')$ in Def. 3 and hence $(\mathcal{M},\mathcal{S})$ is not maximally extended. On the other hand, if $A_{j'+1} \neq B_{l'+1}$ a larger EPM $(\mathcal{M}',\mathcal{S}')$ exists only if there exist some $(ij \sim kl) \in \mathcal{S}'$ with $i \leq j' < j$ and $k \leq l' < l$. Then the condition $\text{parent}_{\mathcal{S}}(j' \sim l') = \text{parent}_{\mathcal{S}'}(j' \sim l')$ of Def. 3 is not satisfied.

Similarly, we need to ensure that the suboptimal traceback only enumerates EPMs which are maximally extended at the gaps created by the G_A and G_{AB} matrices. For this purpose, whenever we trace through these matrices, we record the length of the created gap in both sequences and only consider traces through

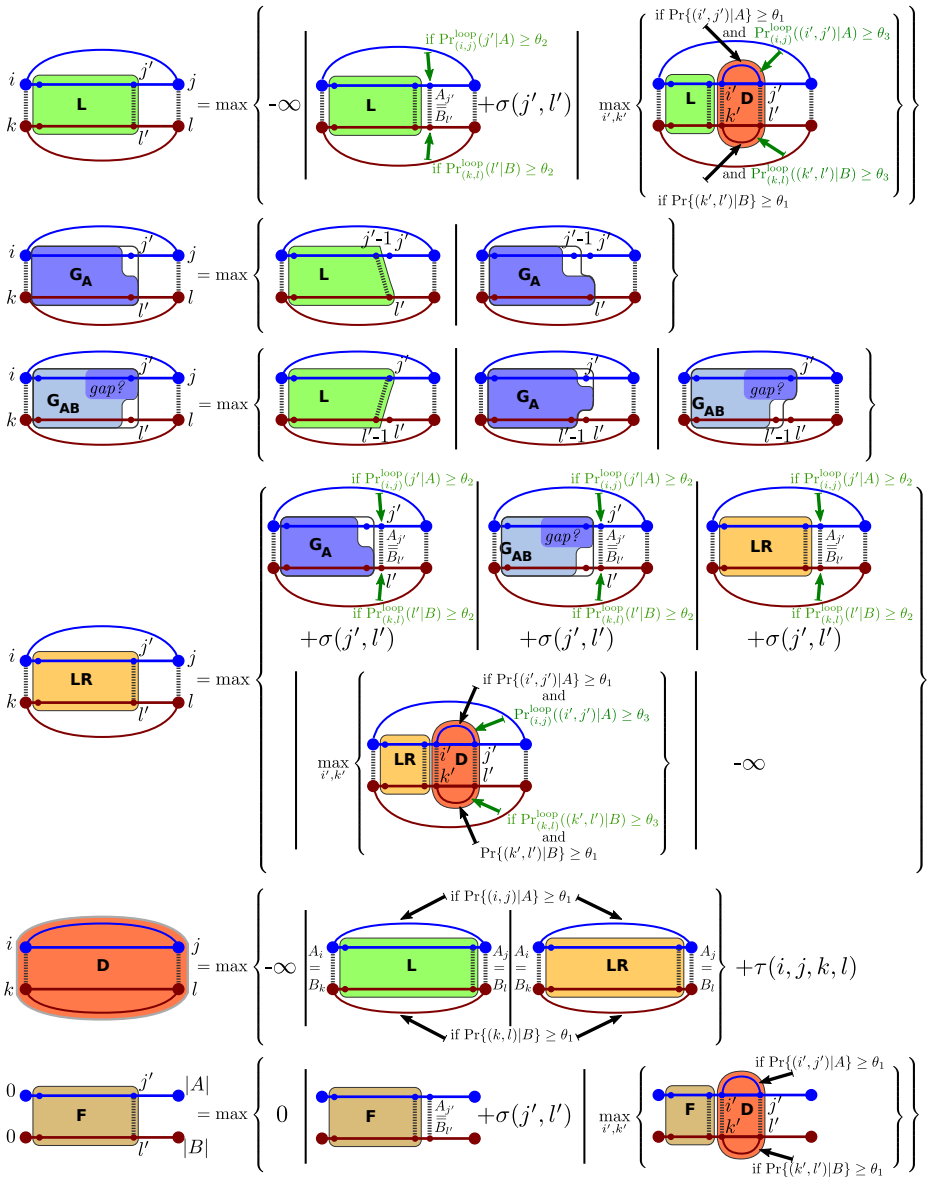

Fig. 3. Visualization of the recursions to compute the matrix entries $L^{ijkl}(j',l')$, $G_A^{ijkl}(j',l')$, $G_{AB}^{ijkl}(j',l')$, $LR^{ijkl}(j',l')$, $D((ij),(kl))$, and $F(j',l')$.

gaps that either start and end at positions that do not match or have a length of 0 in one of the two sequences.

Sparsification We need to compute matrices L^{ijkl}, G_A^{ijkl}, G_{AB}^{ijkl}, and LR^{ijkl} only for i, j, k, l with $\Pr\{(i,j)|A\} \geq \theta_1$ and $\Pr\{(k,l)|B\} \geq \theta_1$. For each of these matrices, we further reduce the number of entries as follows. We call each j' a *candidate* of (i,j) if $\Pr^{\text{loop}}_{(i,j)}(j'|A) \geq \theta_2$ or if for some i' $\Pr^{\text{loop}}_{(i,j)}((i',j')|A) \geq \theta_3$. Analogously, l' is a candidate of (k,l) if $\Pr^{\text{loop}}_{(k,l)}(l'|B) \geq \theta_2$ or if for some k' $\Pr^{\text{loop}}_{(k,l)}((k',l')|B) \geq \theta_3$. Note that if j' or l' is no candidate, the recursion directly implies that $L^{ijkl}(j',l') = LR^{ijkl}(j',l') = -\infty$ and hence we neither have to explicitly compute nor to store these entries. This allows to skip the corresponding entries $G_A^{ijkl}(j',l')$ and $G_{AB}^{ijkl}(j',l')$, because for $L^{ijkl}(j',l') = -\infty$ their value is identical to their respective neighboring entry. In total, this optimization allows to skip in L^{ijkl}, G_A^{ijkl}, G_{AB}^{ijkl}, and LR^{ijkl} each complete row or column whose index is no candidate. Since we can compute (in a preprocessing step and for each sequence separately) a mapping from sequence positions to candidate positions, the recursion can be implemented on matrices that only contain the candidate rows and columns. In the following complexity analysis, we show that this optimization reduces the, across all matrices, $O(|A|^3|B|^3)$ entries to only $O(|A||B|)$ remaining entries.

Complexity Analysis

Lemma 2. *For a fixed j', there are only $O(1)$ base pairs (i,j), such that j' is a candidate of (i,j) (and analogously for l' and (k,l) in sequence B).*

Proof. We fix some j' and denote by $p_{j'}(i,j)$ the probability that a structure of A contains the base pair (i,j) and j' occurs as an unpaired base or right end of a base pair in the loop closed by the base pair (i,j): $p_{j'}(i,j) := \Pr^{\text{loop}}_{(i,j)}(j'|A) + \sum_{i<i'<j'} \Pr^{\text{loop}}_{(i,j)}((i',j')|A)$. If j' is a candidate, it follows $p_{j'}(i,j) \geq \theta^* := \min\{\theta_2, \theta_3\}$, since then either $\Pr^{\text{loop}}_{(i,j)}(j'|A) \geq \theta_2$ or $\Pr^{\text{loop}}_{(i,j)}((i',j')|A) \geq \theta_3$ for some i'. Note that for different (i,j) the events of probabilities $p_{j'}(i,j)$ are disjoint, since in any structure j' can occur in just one loop. Therefore $\sum_{i,j} p_{j'}(i,j) \leq 1$. Hence there are at most $\frac{1}{\theta^*} \in O(1)$ base pairs (i,j) for which $p_{j'}(i,j) \geq \theta^*$ and only for those j' can be a candidate.

Note that for this lemma it is crucial to consider the probabilities within the single loops and not only general base pair and unpaired probabilities. Considering these probabilities is the key insight of this new way of sparsification.

Theorem 1. *There are only $O(n^2)$ entries $L^{ijkl}(j',l')$, $G_A^{ijkl}(j',l')$, $G_{AB}^{ijkl}(j',l')$, and $LR^{ijkl}(j',l')$ such that j' is a candidate of (i,j) and l' is a candidate of (k,l).*

Proof. Due to Lem. 2 there are $O(n)$ many combinations i, j, j'. Analogously there are $O(n)$ combinations k, l, l' and therefore $O(n^2)$ combinations i, j, k, l, j', l' satisfying the conditions.

Corollary 1. *The time and space complexity of computing all entries* $L^{ijkl}(j',l')$, $G_A^{ijkl}(j',l')$, $G_{AB}^{ijkl}(j',l')$, *and* $LR^{ijkl}(j',l')$, D *and* F *is* $O(n^2)$.

Consequently, the preprocessing step, namely computing the base pair probabilities using McCaskill's algorithm is the dominating factor in the complexity.

Chaining Furthermore, we implemented a chaining algorithm that selects from the computed suboptimal EPMs a non-crossing and non-overlapping subset that can be extended to an alignment. It generalizes the chaining of ExpaRNA [25] to cope with more than one EPM ending at the same position. The algorithm recursively fills the gaps of all EPMs with other EPMs. For each of the gaps a matrix of size $O(|A||B|)$ is computed (for details see [25]). At each of its entries all EPMs are considered that end at this position. Since each EPM ends at exactly one position, the complexity is $O(H \cdot (|A||B| + E))$, where E is the number of input EPMs and H the total number of their gaps.

If we guarantee that E is $O(|A||B|)$, i.e. there is only a constant number of EPMs ending at each position, the complexity of the chaining is $O(H|A||B|)$ (as in ExpaRNA). Whereas the suboptimal traceback does not *guarantee* $E \in O(|A||B|)$, we also evaluated a heuristic strategy that satisfies the assumption by considering only the best EPM ending at each position.

5 Evaluation

We implemented ExpaRNA-P together with the chaining algorithm in C++. Furthermore we implemented two versions of the traceback: the suboptimal traceback and a heuristic version that, for each match $i \sim j$, considers only the optimal EPM ending at that match. We instantiated the scoring of ExpaRNA-P (see Eq. 3) by $\sigma(i,k) = 1$ and $\tau(i,j,k,l) = 5(\Pr\{(i,j)|A\} + \Pr\{(k,l)|B\}) + 2$. In addition to the presented scoring, we add a reward of $5\Pr\{(i,j)|(i+1,j-1)|X\}$ ($X \in \{A,B\}$) for each stacking in the EPM. In the suboptimal traceback, we enumerate EPMs that have a score of at least 90 and a score difference of less than 20 to the optimal EPM. Furthermore, we set $\theta_1 = \theta_2 = 0.01$ and $\theta_3 = 0$.

In order to assess the performance of ExpaRNA-P in comparison to other alignment tools, we designed the following pipeline: In a first step we compute the significant EPMs with ExpaRNA-P and use the chaining algorithm to extract from these EPMs an optimal non-overlapping and non-crossing subset. Then we compute a sequence structure alignment that includes all matches of the chained EPMs. For this purpose, we apply LocARNA using the EPMs as anchor constraints [25]. This is faster than computing an unconstrained alignment since each anchor reduces the alignment space. We refer to this pipeline, i.e. the combination of ExpaRNA-P and LocARNA, as ExpLoc-P.

We did a benchmark test on the k2 dataset of BRAliBase 2.1 which contains only pairwise alignments [34, 35]. To measure the quality of the calculated alignment in comparison to the reference alignment, we utilized the compalign score which refers to a sum-of-pairs score (SPS) introduced in this specific form with

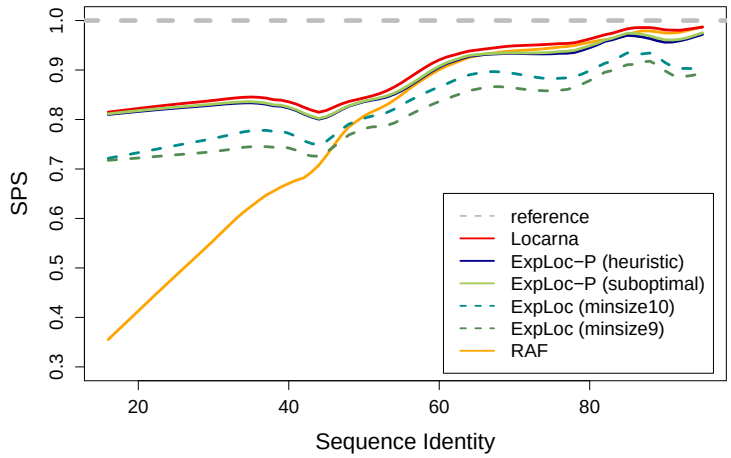

Fig. 4. Alignment quality vs. sequence identity on the k2 dataset of BRAliBase 2.1

BRAliBase 2.1 [34]. Besides the quality of the results, we also compared the runtime of the different methods. We compared our new approach ExpLoc-P with three other approaches: LocARNA without any anchor constraints, ExpLoc [25], and RAF [24]. ExpLoc is similar to our new ExpLoc-P except that it identifies EPMs in the MFE structure (using ExpaRNA). RAF is the currently fastest Sankoff-style sequence structure alignment approach due to its heuristic filtering based on sequence alignments. Fig. 4 shows the compalign score with respect to the sequence identity on the k2 dataset of BRAliBase 2.1. LocARNA achieves the best results at the expense of the highest computation time. Tab. 1 lists the speedups of the other approaches compared to LocARNA. Our novel combined approach ExpLoc-P achieves with both the heuristic and the subobtimal traceback almost the same quality as LocARNA but is 6 and 4.9 times faster, respectively. The best alignment quality that could be obtained with ExpLoc in [25] was achieved with parameter minsize = 10. Even for this optimal setting the quality of the result is significantly lower than the one for LocARNA alone and ExpLoc-P. Additionally, the speedup for this setting is only 4.4 which is also less than both speedups for ExpLoc-P. With minsize = 9, the speedup of ExpLoc is comparable to ExpLoc-P but the quality declines much more. RAF achieves the best speedup of 15.6 but the drawback of the sequence alignment based heuristic filtering which causes this speedup is clearly visible: For sequence similarities below 50% the quality drops tremendously. This indicates that RAF is only successful on instances where sequence information alone is sufficient to get already reasonable alignments. In summary this means that our novel tool ExpLoc-P finds the best tradeoff between alignment quality and speedup and is robust regarding the alignment quality for the whole range of sequence identities.

To analyze the quality of ExpLoc-P further, we investigated whether the compalign scores of ExpLoc-P and unconstraint LocARNA do correlate well. We

Table 1. Runtime comparison of the different approaches. The speedup factor is measured relative to the speed of LocARNA. The runtime is the total runtime for computing the entire benchmark dataset on a single Opteron 2356 processor (2.3 GHz). For ExpLoc-P the first value in brackets is the time for computing and chaining the EPMs and the second one the runtime for the subsequent LocARNA alignments.

	LocARNA	ExpLoc-P (heuristic)	ExpLoc-P (suboptimal)	ExpLoc (minsize 10)	ExpLoc (minsize 8)	RAF
speedup	1	6.0	4.9	4.4	5.4	15.6
total time	14.3h	2.4h (0.4h+2h)	2.9h (0.4h+2.5h)	3.2h	2.6h	0.9h

found a high correlation of 0.85. This indicates that the six-times faster ExpLoc-P pipeline can replace LocARNA in clustering approaches such as [10–12].

Notably, ExpLoc-P significantly outperforms LocARNA for a prominent cluster of the RNA family IRES_HCV. In contrast to most other RNA families, this cluster shows only local conservation. This suggests that ExpLoc-P can improve cluster-based approaches for genome-wide prediction of structural RNA-families, where typically the boundaries of ncRNAs are loosely defined.

6 Conclusion

We introduced the algorithm (ExpaRNA-P) to identify exact pattern matches (EPMs) in RNA structure ensembles. Using a novel sparsification technique, the complexity of the algorithm can be reduced to quadratic complexity, which is the same as the complexity of the algorithm for the corresponding, simpler case of fixed RNA structures (ExpaRNA). Our evaluation demonstrates that EPMs from structure ensembles outperform EPMs from fixed structures when utilized as anchor constraints for structure alignments in our new pipeline ExpLoc-P.

As future work, we will investigate relaxations of the notion of exact patterns to further improve the results. In particular, the same recursions can be used to detect patterns that allow mismatches of base pairs or unpaired bases. Furthermore, EPM based anchor constraints could be used to improve other alignment tools, like RAF. While for LocARNA the constraints yield a considerable speed-up, in RAF they could improve the quality which is poor for low sequence similarity. The score of the chained EPMs could also be used as a distance measure for clustering approaches. This would speed up the clustering process since the expensive computation of full structure alignments can be avoided.

Acknowledgement

This work was partially supported by the German Research Foundation (BA 2168/3-1, MO 2402/1-1, and WI 3628/1-1) and by the German Federal Ministry of Education and Research (S.H., BMBF grant 0313921 FRISYS to R.B.).

References

1. The FANTOM Consortium: The transcriptional landscape of the mammalian genome. Science **309**(5740) (2005) 1559–63
2. Cheng, J., Kapranov, P., Drenkow, J., Dike, S., Brubaker, S., Patel, S., Long, J., Stern, D., Tammana, H., Helt, G., Sementchenko, V., Piccolboni, A., Bekiranov, S., Bailey, D.K., Ganesh, M., Ghosh, S., Bell, I., Gerhard, D.S., Gingeras, T.R.: Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. Science **308** (2005) 1149–1154
3. Bertone, P., Stoc, V., Royce, T.E., Rozowsky, J.S., Urban, A.E., Zhu, X., Rinn, J.L., Tongprasit, W., Samanta, M., Weissman, S., Gerstein, M., Snyder, M.: Global identification of human transcribed sequences with genome tiling arrays. Science **306** (2004) 2242–2246
4. Kapranov, P., Willingham, A.T., Gingeras, T.R.: Genome-wide transcription and the implications for genomic organization. Nat Rev Genet **8**(6) (2007) 413–23
5. Mattick, J.S., Taft, R.J., Faulkner, G.J.: A global view of genomic information - moving beyond the gene and the master regulator. Trends in Genetics (2009)
6. Consortium, A.F.B., Backofen, R., Bernhart, S.H., Flamm, C., Fried, C., Fritzsch, G., Hackermuller, J., Hertel, J., Hofacker, I.L., Missal, K., Mosig, A., Prohaska, S.J., Rose, D., Stadler, P.F., Tanzer, A., Washietl, S., Will, S.: RNAs everywhere: genome-wide annotation of structured RNAs. J Exp Zoolog B Mol Dev Evol **308**(1) (2007) 1–25
7. Rivas, E., Eddy, S.R.: Noncoding RNA gene detection using comparative sequence analysis. BMC Bioinformatics **2**(1) (2001) 8
8. Washietl, S., Hofacker, I.L.: Identifying structural noncoding RNAs using RNAz. Curr Protoc Bioinformatics **Chapter 12** (2007) Unit 12.7
9. Pedersen, J.S., Bejerano, G., Siepel, A., Rosenbloom, K., Lindblad-Toh, K., Lander, E.S., Kent, J., Miller, W., Haussler, D.: Identification and Classification of Conserved RNA Secondary Structures in the Human Genome. PLoS Comput Biol **2**(4) (2006) e33
10. Will, S., Reiche, K., Hofacker, I.L., Stadler, P.F., Backofen, R.: Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. PLOS Computational Biology **3**(4) (2007) e65
11. Kaczkowski, B., Torarinsson, E., Reiche, K., Havgaard, J.H., Stadler, P.F., Gorodkin, J.: Structural profiles of human miRNA families from pairwise clustering. Bioinformatics **25**(3) (2009) 291–4
12. Parker, B.J., Moltke, I., Roth, A., Washietl, S., Wen, J., Kellis, M., Breaker, R., Pedersen, J.S.: New families of human regulatory RNA structures identified by comparative analysis of vertebrate genomes. Genome Res (2011)
13. Höchsmann, M., Töller, T., Giegerich, R., Kurtz, S.: Local similarity in RNA secondary structures. In: Proceedings of Computational Systems Bioinformatics (CSB 2003). Volume 2., IEEE Computer Society (2003) 159–168
14. Siebert, S., Backofen, R.: MARNA: multiple alignment and consensus structure prediction of RNAs based on sequence structure comparisons. Bioinformatics **21**(16) (2005) 3352–9
15. Sankoff, D.: Simultaneous solution of the RNA folding, alignment and protosequence problems. SIAM J. Appl. Math. **45**(5) (1985) 810–825
16. Havgaard, J.H., Lyngso, R.B., Stormo, G.D., Gorodkin, J.: Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. Bioinformatics **21**(9) (2005) 1815–24

17. Mathews, D.H., Turner, D.H.: Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. Journal of Molecular Biology **317**(2) (2002) 191–203
18. Hofacker, I.L., Bernhart, S.H., Stadler, P.F.: Alignment of RNA base pairing probability matrices. Bioinformatics **20**(14) (2004) 2222–7
19. McCaskill, J.S.: The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers **29**(6-7) (1990) 1105–19
20. Gorodkin, J., Heyer, L., Stormo, G.: Finding the most significant common sequence and structure motifs in a set of RNA sequences. Nucleic Acids Res **25**(18) (1997) 3724–32
21. Bradley, R.K., Pachter, L., Holmes, I.: Specific alignment of structured RNA: stochastic grammars and sequence annealing. Bioinformatics **24**(23) (2008) 2677–83
22. Torarinsson, E., Havgaard, J.H., Gorodkin, J.: Multiple structural alignment and clustering of RNA sequences. Bioinformatics **23**(8) (2007) 926–32
23. Bauer, M., Klau, G.W., Reinert, K.: Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. BMC Bioinformatics **8** (2007) 271
24. Do, C.B., Foo, C.S., Batzoglou, S.: A max-margin model for efficient simultaneous alignment and folding of RNA sequences. Bioinformatics **24**(13) (2008) i68–76
25. Heyne, S., Will, S., Beckstette, M., Backofen, R.: Lightweight comparison of RNAs based on exact sequence-structure matches. Bioinformatics **25**(16) (2009) 2095–2102
26. Backofen, R., Siebert, S.: Fast detection of common sequence structure patterns in RNAs. Journal of Discrete Algorithms **5**(2) (2007) 212–228
27. Wexler, Y., Zilberstein, C., Ziv-Ukelson, M.: A study of accessible motifs and RNA folding complexity. Journal of Computational Biology **14**(6) (2007) 856–72
28. Havgaard, J.H., Torarinsson, E., Gorodkin, J.: Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. PLoS Comput Biol **3**(10) (2007) 1896–908
29. Ziv-Ukelson, M., Gat-Viks, I., Wexler, Y., Shamir, R.: A faster algorithm for RNA co-folding. In Crandall, K.A., Lagergren, J., eds.: WABI 2008. Volume 5251 of Lecture Notes in Computer Science., Springer (2008) 174–185
30. Backofen, R., Tsur, D., Zakov, S., Ziv-Ukelson, M.: Sparse RNA folding: Time and space efficient algorithms. In Kucherov, G., Ukkonen, E., eds.: Proc. 20th Symp. Combinatorial Pattern Matching. Volume 5577 of LNCS., Springer (2009) 249–262
31. Salari, R., Möhl, M., Will, S., Sahinalp, S., Backofen, R.: Time and space efficient RNA-RNA interaction prediction via sparse folding. In Berger, B., ed.: Proc. of RECOMB 2010. Volume 6044 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2010) 473–490
32. Backofen, R., Will, S.: Local sequence-structure motifs in RNA. Journal of Bioinformatics and Computational Biology (JBCB) **2**(4) (2004) 681–698
33. Otto, W., Will, S., Backofen, R.: Structure local multiple alignment of RNA. In: Proceedings of German Conference on Bioinformatics (GCB'2008). Volume P-136 of Lecture Notes in Informatics (LNI)., Gesellschaft für Informatik (GI) (2008) 178–188
34. Wilm, A., Mainz, I., Steger, G.: An enhanced RNA alignment benchmark for sequence alignment programs. Algorithms Mol Biol **1** (2006) 19
35. Gardner, P.P., Wilm, A., Washietl, S.: A benchmark of multiple sequence alignment programs upon structural RNAs. Nucleic Acids Research **33**(8) (2005) 2433–9