Chaining Sequence/Structure Seeds for Computing RNA Similarity

Laetitia Bourgeade, Cedric Chauve, Julien Allali

Chaining Sequence/Structure Seeds for Computing RNA Similarity

Laetitia Bourgeade [1], Cédric Chauve [1,3], and Julien Allali [1,2]

[1]LaBRI, Université de Bordeaux, 341 cours de la Libération, Talence, France
[2]Institut Polytechnique de Bordeaux, 146 Rue Léo Saignat, Talence, France
[3]Department of Mathematics, Simon Fraser University, Burnaby (BC), Canada
{laetitia.bourgeade,julien.allali}@labri.fr{cedric.chauve}@sfu.ca

Abstract. We describe a new method to compare a query RNA with a static set of target RNAs. Our method is based on (i) a static indexing of the sequence/structure seeds of the target RNAs, (ii) searching the target RNAs by detecting seeds of the query present in the target, chaining these seeds in promising candidate homologs, then (iii) completing the alignment using an anchor-based exact alignment algorithm. We apply our method on the benchmark Bralibase2.1 and compare its accuracy and efficiency with the exact method *LocaRNA* and its recent seeds-based speed-up *ExpLoc-P*. Our pipeline *RNA-unchained* greatly improves computation time of *LocaRNA* and is comparable to the one of *ExpLoc-P*, while improving the overall accuracy of the final alignments.

1 Introduction

A major advance in molecular biology of the last decade has been the discovery that RNA molecules, especially non-coding RNAs (ncRNAs), are involved in many cellular processes such as the regulation of gene expression, splicing, signaling, ... [16]. This is well illustrated by the growth of the Rfam database [7], whose content went from 15,255 RNAs, in 2002 (date of its creation) to 6,125,803 RNAs, in 2012 (last release). Moreover, several recent studies of the RNA structurome at the whole genome level have lead to the discovery of new families of ncRNAs and to a better understanding of the role of RNAs in the cell [20, 11, 18].

The general problem of annotating, classifying or clustering of RNA sequences is thus an important problem in computational biology, that relies on solving efficiently and accurately the following computational question: given an RNA query Q and a set of RNA sequence targets D, what are the members of D whose similarity with Q is large enough to indicate a potential relationship, either evolutionary and/or functional? For RNA genes, due to the importance of the structure in terms of biological function, it is natural to consider both the sequence and secondary structure when comparing genes. Most RNA comparison methods can be classified in two families: (i) tools requiring the knowledge of an RNA secondary structure, such as *RNAforester* [9] or *Gardenia* [4] to cite only two (see [3] for a thorough evaluation of such methods), and (ii) tools taking

only RNA sequences as input and using covariance models or base pairing probabilities such as *LocaRNA* [20] or *Infernal* [13]. The first family of approaches relies on the classical notions of edit distance and alignment. RNAs are modeled using tree-like structures and algorithms look either for a set of edit operations of optimal score that transforms the first RNA into the second one, or for an alignment maximizing the similarity. A cubic time complexity is the current reference for pairwise RNA structure comparison (see for example [21]), underlining the issue of using such approach directly when a large number of pairwise comparison is required. The second family of RNA pairwise comparison methods works directly on RNA sequences. The current reference method *LocaRNA* aligns RNA sequences based on the pairing probabilities for each sequence, computed from the partition function of the ensemble of all possible foldings into secondary structures, under the assumption of a free-energy based Boltzmann distribution on this ensemble and has a quartic time complexity. Approaches have been introduced to speed-up the alignment, at the expense of guaranteed optimality, such as *ExpaRNA-P*/*Exploc-P* [14]; these methods rely on conserved sequence/structure motifs, called Exact Pattern Matches (EPM), that can be detected in quadratic time and are provided to *LocaRNA* as alignment constraints, thus breaking the alignment computation into smaller independent problems and reducing the overall computation time. Finally, a last set of tools aims at solving the classification problem, that asks to assign a given query RNA sequence to a set of predefined families, such as the Rfam. The Rfam classification engine *Infernal* starts by computing covariance models for families based on RNA sequences that are known to belong to them. Again, despite recent improvements this approach remains time consuming, which has motivated the development of filters such as *RNA sifter* [10], based on the abstract shape approach (see [15]).

In the present work, we address the general problem of the one-against-all RNA pairwise comparison, where a given query RNA Q is compared to unstructured set D of target RNAs. We introduce a new method, *RNA-unchained*, aimed at computing efficiently high quality alignments between Q and the members of D. Our method is based on a classical principle in sequence comparison following four steps: seed indexing for the target set, seeds look-up in the index for the given query, seeds chaining between the query and the targets sharing common seeds, and finally exact anchor-based alignments. To evaluate *RNA-unchained*, we followed the approach of [14] and used the benchmark BRAliBase2.1, which is composed of set of reference pairwise alignments between ncRNA sequences. We measured the accuracy of the alignments obtained by *RNA-unchained*, using the Sum of Pair Scores statistics (SPS) [17]. We observe that we obtain alignments of quality comparable or better than *LocaRNA*, and consistently better than *ExpLoc-P*, in a time comparable to the time taken by *ExpLoc-P*.

2 Methods

The pipeline we describe takes as input a set D of RNA sequences and an RNA sequence query Q, and aims at computing quickly candidate sequences of D that

are similar to Q, together with alignments between Q and these candidates. Our pipeline applies to the case where secondary structures are provided or not. It is composed of two elements: a static preprocessing stage for D and, for a given query Q, a dynamic search in D for RNA similar to Q.

Preprocessing D. This stage is static, *i.e.* is performed once for all. It consists in folding the sequences of D, each into one or several candidate RNA secondary structures, followed by extracting and indexing a set of seeds, defined as sequence/structure motifs of a given length.

Querying D. For a given query Q, its sequence is first folded into an RNA secondary structure and all the seeds it contains are generated. Then the index of seeds from D is searched to identify candidate sequences in D sharing motifs with Q. Next for each candidate from D, an optimal set of seeds that is compatible with the secondary structures of both Q and the candidate is extracted using a fast seeds chaining algorithm, and these seeds are used as *anchors* for a constrained alignment between the candidate and Q.

We now describe the details of our pipeline, starting with the modeling of RNA secondary structures and of seeds. Next we detail how to index these seeds and look-up for existing seeds. Finally we describe the seeds chaining and anchor-based alignments stages.

2.1 Modeling RNA Secondary Structures and Seeds

RNA sequence and secondary structure. An RNA is a molecule composed of four nucleic acids usually symbolized by the alphabet $\{A, C, G, U\}$. Pairs of bases in an RNA molecule can form hydrogen links, thus generating a spatial folding of the molecule forming its secondary structure. Here we consider pseudoknot-free RNA secondary structures, *i.e.* we assume that each base is involved in at most one base pair and that base pairs define a crossing-free planar structure. An RNA secondary structure can be encoded by an *arc-annotated sequence* (aa-sequence for short) [6], and we rely on this modeling to describe our method. An aa-sequence $A = (S, P)$ representing a pseudoknot-free RNA structure is composed of a sequence S of length $|A| = n$ on the alphabet $\{A, C, G, U\}$, representing the RNA primary structure (sequence) and of a well-parenthesized sequence P of length n on the alphabet $\{., (,)\}$, representing the paired bases defining the secondary structure. For a sequence S, we denote by $S[i]$ the $i + 1^{\text{th}}$ symbol of S and by $S[i, j] = S[i]S[i + 1] \dots S[j]$ the factor of S of length $j - i + 1$ starting at position i, for any $i \leq j$ in $\{0, \dots, n - 1\}$. Similar definitions hold for P.

Seeds. Seeds, defined as sequence/structure motifs, are used for two purposes in our pipeline. They are first aimed at detecting quickly candidates RNAs from D that share enough seeds with Q. In a second time, an optimal set of seeds that are compatible with the secondary structures of Q and the candidate is computed, and serves as anchors for the final alignment. So the definition of seeds should (1) allow a fast look-up in the indexing structure and (2) satisfy some compactness condition that makes them compatible with the chaining algorithm we use [2].

Definition *(*(l,d)-centered-seed *(short name* (l,d)*-cs) Let* $A = (S,P)$ *be an aa-sequence of length* n *. Let* d *and* l *be two integers such that* $2d \leq l$*. For a given* i *in* $\{0,\ldots,n-l\}$*, the* (l,d)-cs *of* A *in position* i*, denoted by* cs_i*, is the pair* (s,p) *defined by* $p = P[i, i+l-1]$ *and* $s = S[i+d, i+l-d-1]$*.*

Note that s is a sequence of length $l-2d$ and p a sequence of length l, so a (l,d)-cs is not an aa-sequence as both sequences do not have the same length (see Fig. 1). It follows immediately that the maximal number of distinct (l,d)-cs is $3^l 4^{l-2d}$. Furthermore, such seeds can be seen as spaced seeds [5] with no structural mismatch and possible nucleotide mismatches in a prefix and a suffix of length d of the seed. Next, we define the notion of seed common to two aa-sequences, that we call a *hit*.

Definition *(*hit*) Let* $A_1 = (S_1, P_1)$ *and* $A_2 = (S_2, P_2)$ *be two aa-sequences. Let* d *and* l *be two integers such that* $2d \leq l$*. A hit is an* (l,d)-cs *common to* A_1 *and* A_2*, i.e. a pair* (i,j) *of integers such that:*

- $0 \leq i \leq |A_1| - l - 1$ *and* $0 \leq j \leq |A_2| - l - 1$,
- $P_1[i, i+l-1] = P_2[j, j+l-1]$,
- $S_1[i+d, i+l-d-1] = S_2[j+d, j+l-d-1]$.

The score of a hit S between two aa-sequences S_1 and S_2, composed of a conserved (l,d)-cs located at positions i in S_1 and j in S_2 is defined by:

$$score(S) = \sum_{k=0}^{l-1} f(S_1[i+k], S_2[j+k])$$
$$where f(a,a) = 1 \; and \; f(a,b) = 0 \; if \; a \neq b$$

It follows that the score s of S satisfies $l - 2d \leq s \leq l$. For example on Fig. 2 the score of $\{"AA";"((..))"\}$ is 3.

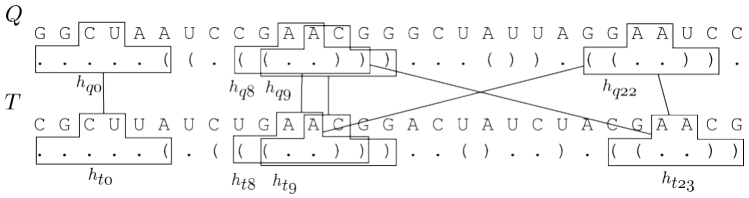

Fig. 1. Example of *hits* between two RNAs Q and T. We can notice that h_{q8} / h_{q9} and h_{t8} / h_{t9} are overlapping hits and h_{q8} / h_{q23} and h_{t8} / h_{t22} are crossing hits. So there is 6 hits $\{(h_{q0};h_{t0}),(h_{q8};h_{t8}),(h_{q8};h_{t23}),(h_{q22};h_{t8}),(h_{q9};h_{t9}),(h_{q22};h_{t23})\}$.

2.2 Seed indexing and hits lookup

The first key element in the method we present consists in indexing in a hash table all (l,d)-cs present in the RNA target set D of interest, for given parameters

l and d. We denote by $\mathcal{I}_l^d$ this index. Comparing Q with the RNAs from D starts by searching in $\mathcal{I}_l^d$ the RNAs of D having seeds present in Q.

Indexing seeds. Given an aa-sequence $A = (S, P)$ of size n and cs parameters l and d, all $k = n - l + 1$ seeds of A are indexed in $\mathcal{I}_l^d$. To do this all computed $(l, d)cs$ are converted in integers as follows: the cs encoding for a $(l, d)cs$ on A at i^{th} position is defined by

$$\begin{aligned} S_{Value}(A, i, l, d) &= 4^{l-2d} \times \sum_{j=i}^{i+l-1} (encode(P_j) \times 3^{i+l-1-j}) \\ &+ \sum_{j=i+d}^{i+l-d-1} (encode(S_j) \times 4^{i+l-d-1-j}) \\ &with\ \ encode: \ A = 0;\ C = 1;\ G = 2;\ U = 3;\ . = 0;\ (= 1;\ ) = 2 \end{aligned}$$

Given an integer x, $\mathcal{I}_l^d[x]$ will contain all occurrences of the cs which S_{Value} is x, that is :

$$\mathcal{I}_l^d[x] = \{(A, i) \mid S_{Value}(A, i, l, d) = x\}$$

For example, with the aa-sequences of Fig. 1, the integer associated to the (l, d)-cs {"AA","((..))"} is 5312, and $\mathcal{I}_6^2[5312] = \{\{(Q, 8), (Q, 22)\}, \{(T, 8), (T, 23)\}\}$.

Inserting all cs of an RNA is done in linear time using a sliding window of length l. From a practical point of view, this index can easily be modified by adding the seeds of a new RNA sequence and it can compute and use simultaneously indexes for different values of l and d.

Index look-up. Given a query Q, the search for aligning it with the target RNAs from D starts by computing all the seeds of Q, for a given pair (l, d) of parameters, then searching $\mathcal{I}_l^d$ for all RNAs from D that have seeds present in Q. For a given RNA T from D, let $\mathcal{LU}_l^d$ be the set of all $(l, d) - cs$ common to both Q and T, *i.e* hits:

$$\mathcal{LU}_l^d(Q, T) = \{(i, j) \mid S_{Value}(Q, i, l, d) = S_{Value}(T, j, l, d)\}$$

For example, using the same aa-sequences as in Fig. 1, $\mathcal{LU}_6^2(Q, T) = \{(0, 0),$ $(8, 8), (8, 23), (22, 8), (9, 9), (22, 23)\}$. This phase is done using a standard hash-table look-up using the integer associated to each seed as key. The time required to compute the $\mathcal{LU}_l^d$ is linear in the size of the query and the overall number of hits. *RNA-unchained* offers the option to reduce the set of candidates based on the number of hits.

Hits/seeds optimization. Preliminary experiments showed that hits that do not contain both structural signal were more likely to be false positive. So in order to obtain more stringent seeds, the hits in $\mathcal{LU}_l^d(Q, T)$ can be filtered to keep only the ones with two types of RNA structural symbols structure (right base

of base pair and left base of base pair) which correspond to seeds that comprise the *hairpin-loops:* ( ) and *stems junction:*) (motifs, that are known to be well preserved and important structural patterns to detect secondary structure similarity [1].

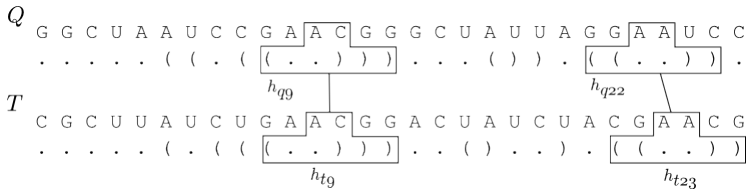

Fig. 2. Compared to Fig. 1 3 hits are lost because of structural composition, crossing and overlapping. So this example show the final hits between Q and T.

2.3 Chaining Algorithm and Anchors

The core of our approach to align Q with the target set D is to first compare Q and a member T of D using solely their hits. To do so, we use a recent efficient algorithm for chaining seeds developed in [2], followed by a stage where gaps between seeds are completed using the *LocaRNA* algorithm. The first steps consists in extending seeds defining hits to account for base-pairing given by the considered RNA secondary structures.

For example on Fig. 3, the last parenthesis of h_{q9} leads to the extension with the opening parenthesis of Q at position 6.

Note that the corresponding seeds might not be contiguous along the sequences (as illustrated in Fig. 3). It follows that they do not satisfy the definition of EPM; however they satisfy the definition of seeds introduced in [2]. Given a set of k extended hits for two RNAs Q and T, an *anchor* is a subset of hits such that, first the corresponding seeds are non-overlapping in both Q and T, second the seeds in both RNAs are compatible in terms of secondary structure (see *chain* definition in [2]). The score of an anchor C is the sum of the scores of the seeds defining the hits it contains, and the chaining score between Q and T is the maximum score of an anchor, taken among all anchors between Q and T (called an *optimal anchor*). The algorithm we use computes an optimal anchor in time $O(k^2 \log k)$, where k is the number of hits. In our running example, an anchor is composed of the hits $\{\{(Q, 9), (T, 9)\}, \{(Q, 22), (T, 23)\}\}$ (see Fig. 2). At the end of the chaining stage, we thus have, for each RNA T from D a set of hits between Q and T that forms an optimal anchor $\mathcal{A}(Q, T)$. We call *gaps* the segments of the RNAs Q and T that are not involved in the anchor.

2.4 Anchor extension

Prior to aligning the gaps of Q and T with an exact but more costly algorithm, we perform a phase of seeds extension aimed at reducing the gap size and compensating the fact that initial seeds are of bounded length. First, each hit of the

anchor between the two RNAs Q and T is extended on both sides based on exact sequence similarity. As an example, on Fig. 3 the hit (h_{q22}, h_{t23}) is extended to the left by two nucleotides.

Next, still prior to the gaps alignment, we fill the remaining gaps using an adapted Longest Common Subsequence (LCS) considering triplet of letters (see Fig. 3 for an example) to avoid irrelevant constraints. This improvement has meaning only if the cover of the anchor is large enough. So we fix a threshold of $4l$ bases matches.

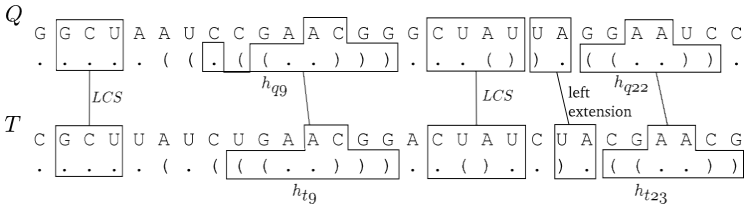

Fig. 3. The matches between Q and T are left extended only for h_{q22} and h_{t23} and gaps are filled thanks to LCS computation.

2.5 Anchor-constrained alignment

Finally, for each candidate homolog T, the gaps defined by the anchor between Q and T are aligned using the exact algorithm *LocaRNA* where the anchor is provided as a set of constraints (see Fig. 4).

Fig. 4. The anchor of our example, seen as sequence constraints.

3 Results

In order to assess the ability of *RNA-unchained* to provide accurate alignments, we applied on the set of reference RNA alignment provided by the benchmark Bralibase2.1 [19], composed of 8,976 pairwise alignments, classified into 36 families.

We analyzed this benchmark with *RNA-unchained* using several sets of options, *LocaRNA* used as a reference exact alignment tool and *ExpLoc-P*, a seed-based speed-up of *LocaRNA*. In order to reproduce the results on Bralibase2.1

shown in [14], we obtained from the authors of *ExpLoc-P* the corresponding code and parameters, and we ran *ExpLoc-P* with these optimized parameters.

In addition to these existing methods, we ran *RNA-unchained* with the following default parameters: (1) for each RNA sequence, its MFE secondary structure was obtained using RNAfold [12, 22], (2) the parameters l and d for seeds were chosen to be $l = 9$ and $d = 1$, after exploring a wide range of possible values for these parameters (see discussion below). We denote this default *RNA-unchained* version $91MFE$. In order to assess the impact of adding stringency criterion to the seed selection process, as well as the impact of the anchor extensions methods described in the 2 section, we ran *RNA-unchained* with additional options. (1) $91r2$: only $(9, 1)$cs containing two-types parenthesis are conserved. (2) $91r2fb91epcLCS36$: $91r2$ with seeds optimization, *i.e.* if there is no $91r2$ seed consider $91MFE$ seeds, and anchor optimization. So all together, we show the results of two reference programs (*LocaRNA*, *ExpLocPOpt*) and three versions of *RNA-unchained* ($91MFE$, $91r2$ and $91r2fb91epcLCS36$).

To compare the obtained alignments with the reference alignments from Bralibase2.1, we use the SPS statistics (Fig. 5). Given a reference alignment r of length l_r and a computed alignment e of length l_e, the SPS is defined by the ratio SP^e/l_r where SP^e is the number of pairs (i, j) where position i and j of the aligned sequences form a match in both r and e. In addition, we also consider the coverage in percent of the input RNA by the anchors, defined as the ratio between the number of bases belonging to anchors by the length of the RNAs (Fig. 6). This statistic is important to evaluate the impact of anchors, both in terms of computation time and of accuracy, as a high coverage by wrong hits will mechanically result in a low SPS, while a very low coverage by high confidence hits might not result in a significant gain of computation time. Finally, we display our result according to the similarity between the pairs of compared RNAs, where the similarity value $Sim(Q, T)$ between a query Q and a target T is defined, from the reference alignments, as follows:

$$Sim(Q, T) = \frac{\sum_{i=0}^{l-1} f(Q'[i], T'[i])}{|Q'|}$$

where Q', T' are the aligned sequence derived from Q, T and $f(a, a) = 1$ and $f(a, b) = 0$ if $a \neq b$. Both Fig. 5 and 6 present the number of alignments per similarity level (right scale).

We also show the computation time of the different methods that we considered. Note that the difference between the running time of *LocaRNA* and *ExpLocPOpt* is not as important as shown in [14], but the computation time of *ExpLocPOpt* is comparable between our experiments and [14]. A first point we can notice is that all methods perform well and with relatively similar behaviour, but for reference alignment between pairs of RNAs that exhibit a similarity in the range [0.6,0.8] (*i.e.* 60%-80%), where *RNA-unchained* used with stringent grains composed of at least two kinds of structural elements, performs better, and even obtains better results than *LocaRNA*, although *LocaRNA* is guaranteed to obtain alignment with better alignment scores. This shows that adding high quality

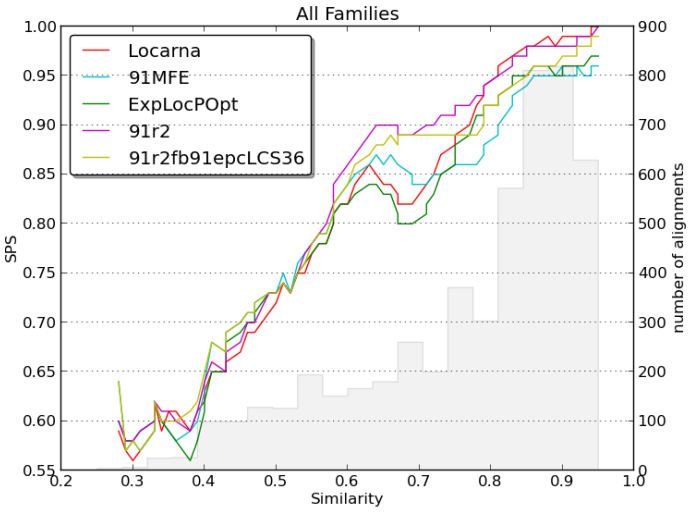

Fig. 5. SPS value for all 5 methods applied to the Bralibase2.1 benchmark.

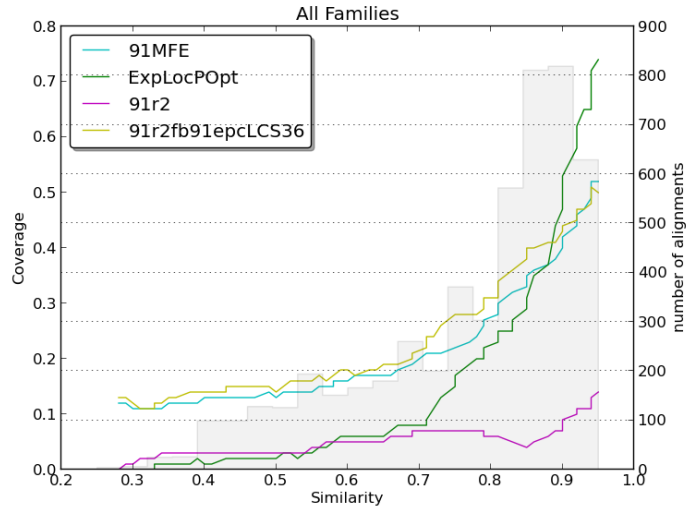

Fig. 6. Coverage of RNA sequences by anchors in percent for all 4 seeds-based methods (*LocaRNA* alone does not use seeds) applied to the Bralibase2.1 benchmark.

anchors, even if they cover a smaller part of the considered RNA sequences (see Fig. 6, $91r2$), can improve significantly the alignment quality. However, being too stringent in defining hits results in a coverage that is relatively low, which

Computation time	Hits/chaining	Gaps alignment	Total
LocaRNA	0	9,022	9,022
ExpLocPOpt	1,492	6,070	7,562
$91MFE$	3,386	4,563	7,949
$91r2$	3,157	6,242	9,399
$91r2fb91epcLCS36$	3,283	4,510	7,793

Table 1. Computation times (in seconds). The time required to build the index is not included but takes less than 1 minute. Experiments were performed on a server with double Intel Xeon 3.3GHz processor. The seeds indexing, hits look-up and chaining are implemented in Java.

has for consequence that many alignments are constraint-free and rely purely on *LocaRNA*. Note however, that the low covering hits selected with this method still results in a significant accuracy improvement over *LocaRNA* in the range [0.6,0.8]. On the opposite, $91MFE$ has a good coverage meaning that *LocaRNA* takes great advantage on alignment computation as its computation time is divided by two. However, the accuracy of the alignments can be significantly lower than with the other methods, as some hits with low structural information are false positive that can be selected by the chaining, thus misleading *LocaRNA* in the final anchor-based exact alignment phase. Finally, the two optimizations we propose, the seed optimization (see paragraph 2.2) and the LCS-based anchor extension (see section 2.4 improves significantly the coverage by the anchor (see method $91r2fb91LCS36$) without impacting too much the alignments accuracy. This is reflected by the SPS values for $91r2fb91LCS36$ that are accuracy results while the coverage is the best one achieved by our versions. As a result, the gain in terms of alignment time compared to *LocaRNA* is maximal (divided by two).

Regarding seeds parameters, various values of l and d from $(5, 1)$ to $(10, 2)$ were tested before settling for the combination $l = 9$ and $d = 1$. The experiments we carried to explore these parameters (results not shown) indicated clearly that seeds with a small conserved sequence lead to many false positive seeds, while large conserved structures, especially with the additional requirement of two types of structural elements, lead to a very low coverage, and so a higher computational time in the gap alignment phase. In general, these experiments show a relatively consistent pattern of correlated increasing coverage / decreased SPS.

4 Discussion

Summary. The main contribution we presented in this paper is a complete pipeline for the one-against-all RNA pairwise comparison, based on the notions of seeds, seeds index and seeds chaining. The key points are: a seed model describing both primary and secondary structure elements and a fast (sub-cubic)

seeds chaining algorithm. The ability to index quickly and retrieve efficiently common seeds between a query and a set of targets is an important point of our method, that scales well as the main memory consumption of the index is determined by the number of keys in the hash table, which depends only on the seeds parameters (l, d), as the data itself grows linearly with the cumulated size of the target RNAs. Our experiments using the benchmark Bralibase2.1 show clearly that *RNA-unchained* obtains results that are more accurate than current state-of-the art methods, with comparable computation times (and probably better computation times in the C++ next release).

Seeds model and parameters. The seed model we introduce differ significantly from the *ExpaRNA* model while EPMs are designed as connected subgraphs of the RNA secondary structure. It is interesting to notice that our seeds model (with the seeds and anchors extensions) provide a coverage of the RNA sequences that is comparable to the one obtained with EPMs (Supplementary Fig. **??**). This shows that both models probably are able to capture important conserved structural features. However, we can notice that in the similarity range 60%-80%, where *RNA-unchained* outperforms *ExpLoc-P* and *LocaRNA* in terms of SPS, we observe a significant difference. More generally, our work and the line of work centered on EPM suggest that the general seeds chaining approach deserves to be explored, both in terms of seed models and chaining algorithms. In particular, unlike sequence seeds, that have been deeply studied, formal studies of RNA seeds, including statistical aspects, are lacking.

Secondary structure. A major difference between our approach and the *LocaRNA/ ExpLoc-P* lies in the way the secondary structure of RNA is accounted for. We explored several intermediate approaches, based on sampling RNA secondary structures using *RNAsubopt* and *RNAshapes* or based on keeping alignment with best score when using several suboptimal structures, but found that the MFE still provided the best accuracy results while minimizing the computation time (results not shown). This apparent concordance between two very different approaches suggests again that the notion of RNA structural seeds still deserves to be further studied.

Gap filling and chaining. An important aspect of *RNA-unchained* concerns the use of *LocaRNA* for the anchor-based gaps alignment. As gaps are segments where conserved structural motifs are absent, this allows to reduce the impact of the choice of the MFE, that is used only to detect seeds and compute the anchor, and is likely to be one of the reasons that explains the concordance between both approaches. However, this part of our pipeline is still the most costly in terms of computation time. As suggested by the results obtained with our LCS anchor-extension, a hierarchical/iterative approach, that would consider less conserved sequence and/or structure motifs detected within a given gap, and thus help again to reduce the segments on which an exact alignment algorithm is used, could be an efficient approach. Taking a somewhat extreme point of view, one could even ask if, in applications where exact alignments are not needed,

the approach described above, limited to the computation of extended anchors, possibly completed by a quick way to evaluate the similarity between short gaps in RNA sequences, would not be sufficient.

References

1. J. Allali and M F Sagot (2008) A multiple layer model to compare RNA secondary structures, *Software Practice and Experience*, **38**, 775-792.
2. J Allali, C Chauve, P Ferraro and A L Gaillard (2011) Efficient chaining of seeds in ordered trees , *J Discrete Algorithms*, **14**, 107-118.
3. J Allali, Y Aubenton-Carafa, C Chauve *and al.* (2012) Benchmarking RNA secondary structure comparison algorithms, *Advances in Bioinformatics*, **2012**, 893048.
4. G Blin, A Denise, S Dulucq, C Herrbach and H Touzet (2010) Alignment of RNA structures, *IEEE/ACM Trans Comput Biol Bioinformatics*, **7**, 309-322.
5. D G Brown (2008) Bioinformatics Algorithms: Techniques and Applications, chapter A survey of seeding for sequence alignment, *Wiley-Interscience*.
6. P A Evans (1999) Finding Common Subsequences with Arcs and Pseudoknots, *Combinatorial Pattern Matching*, *Springer*, 270-280.
7. S Griffiths-Jones, A Bateman, M Marshall, A Khanna and S R Eddy (2003) Rfam: an RNA family database, *Nucleic Acids Res*, **31**, 439-441.
8. S Heyne, W Sebastian,M Beckstette and R Backofen (2009) Lightweight comparison of RNAs based on exact sequence–structure matches, *Bioinformatics*, **25**, 2095-2102.
9. M Hochsmann, T Toller, R Giegerich and S Kurtz (2003) Local similarity in RNA secondary structures, *Computational Systems Biology*, *ACM*, 159-168.
10. S Janssen, J Reeder and R Giegerich (2008) Shape based indexing for faster search of RNA family databases, *BMC Bioinformatics*, **9**, 131.
11. M Kertesz, Y Wan, E Mazor, JL Rinn, RC Nutter *and al.* (2010) Genome-wide measurement of RNA secondary structure in yeast, *Nature*, **467**, 103-107.
12. R Lorenz, S H Bernhart, C H zu Siederdissen, H Tafer, C Flamm, P F Stadler and I L Hofacker (2011) ViennaRNA Package 2.0, *Algorithms Mol Biol*, **6**, 26.
13. E P Nawrocki and S R Eddy (2013) Infernal 1.1: 100-fold faster RNA homology searches, *Bioinformatics*, **29**, 2933-2935.
14. C Schmiedl, M Möhl, S Heyne, M Amit, G M Landau, S Will *and al.* (2012) Exact pattern matching for RNA structure ensembles, *RCMB*, *Springer*, 245-260.
15. P Steffen, B Vo, M Rehmsmeier, J Reeder *and al.* (2006) RNAshapes: an integrated RNA analysis package based on abstract shapes, *Bioinformatics*, **22**, 500-503.
16. C R Thomas, D Bennett, B Jasny, K Kelner, L Miller, P Szuromi, D Voss, P Kiberstis, S Parks *and al.* (1992) The molecule of the year, *Science*, **258**, 1861.
17. J D Thompson, F Plewniak, O Poch (1999) A comprehensive comparison of multiple sequence alignment programs, *Nucl Acid Res*, **27**, 2682-2690.
18. Y Wan, M Kertesz, RC Spitale, E Segal and HY Chang (2011) Understanding the transcriptome through RNA structure, *Nat Rev Genet*, **12**, 641-655.
19. A Wilm, I Mainz and G Steger (2006) An enhanced RNA alignment benchmark for sequence alignment programs, *Algorithms Mol Biol*, **1**, 19.
20. S Will, K Reiche *and al.* (2007) Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering, *PLOS Comput Biol*, **3**, e65.
21. C Zhong and S Zhang (2013) Efficient alignment of RNA secondary structures using sparse dynamic programming, *BMC Bioinformatics*, **14**, 269 .
22. M Zuker, P Stiegler (1981) Optimal computer folding of large RNA sequences using thermodynamic and auxiliary information, *Nucl Acid Res*, **9**, 133-148.