Bioinformatics Advance Access published November 15, 2007

# MASTR: Multiple alignment and structure prediction of non-coding RNAs using simulated annealing

Stinus Lindgreen[a,*], Paul P. Gardner[a b,†] and Anders Krogh[a]

[a]Bioinformatics Centre, Department of Molecular Biology, University of Copenhagen, Denmark
[b]Molecular Evolution Group, Department of Molecular Biology, University of Copenhagen, Denmark

Associate Editor: Prof. Dmitrij Frishman

## ABSTRACT

**Motivation:** As more non–coding RNAs are discovered, the importance of methods for RNA analysis increases. Since the structure of ncRNA is intimately tied to the function of the molecule, programs for RNA structure prediction are necessary tools in this growing field of research. Furthermore, it is known that RNA structure is often evolutionarily more conserved than sequence. However, few existing methods are capable of simultaneously considering multiple sequence alignment and structure prediction.

**Results:** We present a novel solution to the problem of simultaneous structure prediction and multiple alignment of RNA sequences. Using Markov chain Monte Carlo in a simulated annealing framework, the algorithm *MASTR* (*M*ultiple *A*lignment of *ST*ructural *R*NAs) iteratively improves both sequence alignment and structure prediction for a set of RNA sequences. This is done by minimizing a combined cost function that considers sequence conservation, covariation and basepairing probabilities. The results show that the method is very competitive to similar programs available today, both in terms of accuracy and computational efficiency.

**Availability:** Source code available from http://mastr.binf.ku.dk/

**Contact:** stinus@binf.ku.dk

## 1 INTRODUCTION

In recent years, the amount of evidence that RNAs play a much more active role in the cell than previously thought has grown dramatically. The view has now shifted away from the assumption that non-coding RNAs (ncRNA) merely helped in the protein synthesis (e.g. tRNA, rRNA), and today a wide variety of catalytically active RNAs or ribozymes have been characterized. It has also become clear that ncRNA is a very diverse group of molecules both in terms of function and structure.

RNA molecules have been found to play important and diverse roles [1, 2, 27]: The recently discovered family of microRNAs (miRNA) is involved in gene expression and cell specialization, vault RNAs seem to be involved in multi-drug resistance important for the treatment of cancer, and small nuclear RNAs (snRNA) are key players in the splicing of pre-mRNA. A large number of other ncRNA families have yet to be functionally characterized.

It has also become clear that these non-protein coding genes vary greatly in size, ranging from microRNAs of approximately 20 nucleotides to more than 10,000 nucleotides in RNAs involved in eukaryotic gene silencing [19], and also that they are transcribed in different ways: Some reside in introns of protein coding genes, while others are large transcripts that include introns and the possibility of alternative splicing although they lack the open reading frame of a protein coding gene [27].

Experimental studies show that a huge fraction of the human genome is transcribed [3], and computational studies show evidence that thousands of structurally conserved RNAs can be found in the human genome [31, 40]. There is therefore little doubt that RNAs are biologically very important, and the structural analysis of RNA sequences is a field of growing interest. Through evolution, the sequences of related RNAs can diverge although the structure remains conserved. Pure sequence comparison methods therefore fail when applied to ncRNAs that have diverged too much [7].

It is ultimately the tertiary structure that determines the function of the molecule, and advances are being made in this field [4, 34]. However, in the case of RNA it is often sufficient to determine the secondary structure. The reason is that the formation of secondary structure is fast, and the basepairing interactions are strong. The secondary structure, therefore, contributes the major part of the folding energetics, forming a stable scaffold for the formation of tertiary interactions [30].

There exist methods to fold a single RNA sequence either by maximizing basepairing interactions [29], or by minimizing the free energy of the structure (mfold [44], RNAfold [16]). Another approach is to use an existing sequence alignment and predict a consensus structure based on this. In RNAalifold [18], this has been pursued using a combination of free energy and covariation. In Pfold [21, 22], a stochastic context-free grammar (SCFG) is used to predict a common structure from a multiple alignment.

If pseudoknots are disregarded, an RNA structure can be represented as a tree. Since comparison of strings can be extended to trees [36, 43], alignments could be based on the structures directly. In RNAforester [15] the input is a set of RNA sequences with (possibly predicted) secondary structures, and the problem thus becomes a forest alignment problem. The program performs either local or global alignment of the structures, and the output is an alignment and predicted consensus structure based on the structural similarities. MARNA [35] is another heuristic method, where a multiple structural alignment is inferred from all pairwise alignments of secondary structures.

Due to the tight relationship between sequence and structure, the solution to the sequence alignment problem and the structure prediction are interdependent. Whether aligning without considering the structure, or folding without considering sequence alignment, information is ignored. Ideally, one should therefore perform the sequence alignment and structure prediction in parallel. In 1985,

[*] to whom correspondence should be addressed

[†] present address: Wellcome Trust Sanger Institute, Cambridge, UK

Sankoff presented an exact solution to this $\mathcal{NP}$–hard problem [33], but the exponential running time of $\mathcal{O}(L^{3N})$ and memory usage of $\mathcal{O}(L^{2N})$ makes it intractable even for problems of moderate size.

Various implementations of the Sankoff–algorithm exist. Foldalign [9, 13] and Dynalign [25] are both limited to local pairwise alignment. In PMcomp/PMmulti [17], the optimal alignment of two basepairing probability matrices is found instead of aligning the RNA sequences *per se*. A multiple alignment can be built using progressive alignment of the basepairing probability matrices. A similar progressive approach is used in FoldalignM [39]. LocARNA [41] is a local alignment tool similar to but more efficient than PMcomp, and this program can also be used to do progressive multiple alignment and structure prediction. In RNAcast [32] the consensus structure problem is dealt with in a different way: By using abstract shapes [8], where the structures can be regarded without all details but only using the layout of the structure, the search space is reduced. RNAcast predicts the best common shape for all the sequences and, for each sequence, the energetically best structure.

In RNA Sampler [42] stems are the core building blocks. For each sequence, a list of all possible stems consisting of consecutive $A\bullet U$, $G\bullet C$ and $G\bullet U$ pairs is generated. A pairwise alignment is found by aligning all stems from one sequence with all stems from another, and the loop regions are aligned using ClustalW [37]. Since bulges are not allowed in stems, the alignment process can be done efficiently by sliding one stem along the other. Such a block of aligned stems has a conservation score including both nucleotide alignment probabilities and basepairing probabilities. From the set of blocks, a common structure is found by sampling, and the probability of a block being chosen depends on the conservation score. The probability matrices are then updated based on the sampled structures, and the process is iterated until convergence. This process has been extended to multiple sequences by considering all pairs in a set of multiple RNA sequences.

SimulFold [28] is a fully probabilistic model using Bayesian Markov chain Monte Carlo. The program takes as input a set of unaligned sequences $D$ and samples both multiple alignment $A$, secondary structure $S$ and a phylogenetic tree $T$ from the joint posterior probability $P(S, A, T|D)$. This very comprehensive program came out very recently, but although it has some methodology in common with MASTR (e.g. sampling based on the likelihood of the solution) it does so in a very different way, which also shows in the computational complexity of the program.

Since the exact solution to the problem is too time and memory consuming to be pursued, all the methods above are simplified in one way or another. Furthermore, it has been suggested that the optimal minimum free energy structure is not necessarily a good solution to the consensus structure problem [5]. We therefore pursue a heuristic sampling approach where the structure and sequence alignment can be optimized in parallel. In our approach a cost function (or energy) is defined as a sum of three terms: an alignment term, a structure term, and a covariance term. This cost function is minimized using simulated annealing to obtain the combined alignment *and* structure with minimum cost – the best solution according to the cost function. This optimization is carried out by changing the structure on the basepair level or by moving gaps around in the sequence alignment. The change is then judged by the change in the cost function and either accepted or rejected. The procedure is implemented in the program called *MASTR* (*M*ultiple *A*lignment of *ST*ructural *R*NAs).

## 2 METHODS

### Defining the cost function

To find a solution to the problem of simultaneous multiple alignment and structure prediction, we define a cost function which will be minimized in order to search for the optimal solution. A solution should minimize a combined cost function $cost(A, \mathcal{S})$ which incorporates both the sequence alignment $A$ and the predicted consensus structure $\mathcal{S}$. The different parameters used in the program (e.g. scaling parameters and thresholds) have been set using grid optimization. A small number of low identity RNA datasets have been used to optimize the parameters by changing the settings slightly and reevaluating the results. It should be noted that the datasets used for optimizing the parameters are not the same as in the test, and that the datasets do not cover all the families used in the comparison.

*Calculating alignment cost.* There exist many ways of determining the cost of a multiple alignment: Sum of Pairs using a substitution matrix and minimization of the entropy of the alignment are two well–known examples [6], and using a phylogenetic tree to sum the pairwise alignments inferred by the edges has also been pursued [14].

During the development of the algorithm, various sequence cost functions were examined. Sum of Pairs and different entropy based measures were tested using both single nucleotide and dinucleotide domains. We selected the best performing cost function, which proved to be a log-likelihood cost function inspired by Hidden Markov models (HMMs) over single nucleotides.

The cost function is fully probabilistic in its treatment of both gaps and nucleotides. We assume independence between the sites in the alignment. When calculating the cost, we have an alignment of length $L$ consisting of $N$ sequences. Let $x_j^i$ denote the $j$'th character in sequence $i$, and let $P(x_j^i)$ denote the probability of seeing character $x_j^i$ at this specific position. Assuming the sites are independent, the probability of the multiple alignment $A$ becomes:

$$P(A) = \prod_{j=1}^{L}\prod_{i=1}^{N} P(x_j^i)$$

The individual character probabilities need to be determined and gaps have to be taken into account. If $x_j^i$ is a gap we have two cases: Let $P_{GO}$ denote the gap open probability, i.e. the probability of having a gap at position $j$ given that position $j-1$ contained a nucleotide. Similarly, $P_{GE}$ is the gap extension probability used when both position $j$ and $j-1$ contain a gap. Both of these probabilities can be estimated from known structural alignments. In the program, they are set to $P_{GO} = 0.5$ and $P_{GE} = 0.74$.

If $x_j^i$ is a nucleotide from the alphabet $\Sigma = \{A, C, G, U\}$, the probability $P(x_j^i)$ is calculated based on the nucleotides that comprise the column. Additionally, the probability is dependent on the preceding character. If $x_{j-1}^i = -$, we have a gap closing, and the probability is multiplied by $(1 - P_{GE})$. Similarly, if $x_{j-1}^i \in \Sigma$, the probability is multiplied by $(1 - P_{GO})$. Let $c_j(a)$ be the number of occurrences of nucleotide $a \in \Sigma$ at position $j$ in the alignment. The probabilities are given as:

$$P(x_j^i) = \begin{cases} P_{GO} & x_j^i = -, x_{j-1}^i \in \Sigma \\ P_{GE} & x_j^i = -, x_{j-1}^i = - \\ (1 - P_{GO})\dfrac{c_j(a)}{\sum_{b\in\Sigma} c_j(b)} & x_j^i = a \in \Sigma, x_{j-1}^i \in \Sigma \\ (1 - P_{GE})\dfrac{c_j(a)}{\sum_{b\in\Sigma} c_j(b)} & x_j^i = a \in \Sigma, x_{j-1}^i = - \end{cases}$$

A simple pseudo-count function is used where $c_j(a)$ is incremented by 1 for each $a \in \Sigma$ and $1 \leq j \leq L$. An IUPAC ambiguity character is exchanged with one of the nucleotides it symbolizes with equal probability. For instance, if an $N$ occurs in a sequence, it is replaced by any one of the four nucleotides with 25% chance each. Similarly, an $S$ will be exhanged with either a $C$ or a $G$ with a 50% chance each.This exchange is done once in the beginning of the program. Having these probabilities, $P(A)$ can be calculated. The cost function used is the negative log-likelihood based on the alignment probability:

$$Q(A) = -\log_2\left(P(A)\right) \qquad (1)$$

*Calculating structure cost.* The cost of the structure is defined as the sum of the cost of the individual basepairs. Let $\mathcal{S}$ be the structure consisting of basepairs $(i, j)$:

$$cost(\mathcal{S}) = \sum_{(i,j)\in\mathcal{S}} cost(i, j)$$

There are two ways to score the structure: by the free energy of single sequences and by covariation. In the present work, we use the two measures that proved best at predicting true basepairs in our previous study [23]: The McCaskill basepair probabilities [26], called $P(bp_{i,j})$, and a novel version of the covariation measure used in RNAalifold [18] extended to include stacking of basepairs, called $C(bp_{i,j})$.

McCaskill showed how to calculate the partition function over all possible secondary structures of an RNA sequence. The basepair probabilities are found using the weighted Boltzman ensemble favoring more stable structures. We use RNAfold, which is part of the Vienna package [16], to calculate the probability matrices. Since gaps are added to the sequences as part of the alignment this has to be taken into account when indexing the matrices: The partition function is calculated once for each ungapped sequence $s = 1, \ldots, N$ before the optimization starts, and the results are stored in individual probability matrices $M^s$. These matrices do not change throughout the algorithm. For a basepair $(i, j)$ in the alignment, we need to correct the indices to be able to find the probability for that particular basepair. Let these gap-corrected indices be denoted $(i^s, j^s)$, where $i^s = i - M^s$ and $M^s$ is the number of gaps preceding position $i$ in sequence $s$, and similarly for index $j$. The probability for the basepair in sequence $s$ is then found as $M^s(i^s, j^s)$. If either $i$ or $j$ is a gap in sequence $s$, $M^s(i^s, j^s) = 0$. A basepair $(i, j)$ in the alignment is scored by the mean probability:

$$P(bp_{i,j}) = \frac{1}{N}\sum_{s=1}^{N} M^s(i^s, j^s)$$

To transform this into a cost function for the basepairs, the negative logarithm of the mean probability is taken and a threshold is introduced. The threshold reflects the background probability $P_{null}$ of random basepairs found in the probability matrices:

$$cost_P(i, j) = -\log_2\left(P(bp_{i,j})\right) + \log_2\left(P_{null}\right) \qquad (2)$$

A background probability of $P_{null} = 0.25$ is used based on parameter optimization (data not shown).

Through evolution, related RNA sequences can mutate which leads to different sequences of nucleotides while the same core secondary structure is retained. When a mutation happens at a position that is involved in a basepair, selection favours mutations at the other position that maintain the structure and molecular function. This is known as compensatory mutations. Thus, structure is often more conserved than sequence, and this signal can be measured by a covariation score.

In [23] we analyzed various measures of covariation. We refer to this paper for details, but here the chosen cost function will be briefly explained. The RNAalifold measure uses a matrix $\Pi^\alpha$ for each sequence $\alpha$, where $\Pi^\alpha_{i,j} = 1$ if sequence $\alpha$ can form a basepair between position $i$ and $j$, and $\Pi^\alpha_{i,j} = 0$ otherwise. The function $\delta(x_i^\alpha x_j^\alpha, x_i^\beta x_j^\beta)$ measures the Hamming distance between two aligned pairs at positions $i$ and $j$ in sequences $\alpha$ and $\beta$. The goal is to measure the fraction of consistently aligned pairs. A penalty term, $q_{i,j}$, measures the fraction of sequences with inconsistent pairs in the alignment. The covariation is then found as:

$$B_{i,j} = \left(\frac{1}{\binom{N}{2}}\sum_{\alpha<\beta}\delta(x_i^\alpha x_j^\alpha, x_i^\beta x_j^\beta)\Pi^\alpha_{i,j}\Pi^\beta_{i,j}\right) - q_{i,j}$$

To add stacking information, the two basepairs enclosing the pair in question are also considered, but more weight is put on the actual pair:

$$C(bp_{i,j}) = \frac{B_{i-1,j+1} + 2\cdot B_{i,j} + B_{i+1,j-1}}{4}$$

To turn this into a cost function, the same approach is used as for the partition function above. The covariation score is negated and a threshold value added:

$$cost_C(i, j) = -C(bp_{i,j}) + \phi \qquad (3)$$

A threshold of $\phi = 0.25$ is used. Using the two cost functions $cost_P(i, j)$ and $cost_C(i, j)$ (Eqs. 2 and 3, respectively), the predicted structure can be evaluated and a move either accepted or rejected based on Eq. 4 below.

*Combined cost.* Since we simultaneously optimize both sequence alignment and structure prediction, the cost function is a combination of three terms: The log-likelihood cost in Eq. 1, the basepair probability cost in Eq. 2, and the covariation cost in Eq. 3. The cost of the secondary structure is given as a sum over all basepairs in the structure $\mathcal{S}$:

$$cost(A, \mathcal{S}) = Q(A) + \sum_{(i,j)\in\mathcal{S}}\left(\alpha\cdot cost_P(i, j) + \beta\cdot cost_C(i, j)\right)$$

The parameters $\alpha$ and $\beta$ are parameters used to balance the contributions from the different terms in the combined cost. As default, they are set to $\alpha = 1.5$ and $\beta = 0.6$, which are obtained from an initial grid search parameter optimization (data not shown).

## Optimizing the solution

Simulated annealing [20] is an optimization technique inspired by the physical process of annealing, which describes the slow cooling of material to form a crystal structure. The idea is that the positions of the individual atoms can be described as a probability distribution dependent on the temperature of the system: At high temperatures the atoms have a high energy and therefore move around, but as the temperature is lowered, the system becomes more stable. The goal is to form crystals with few defects, and the most stable crystal structure is the one with the lowest free energy. If the temperature of the system is decreased too fast, the crystal structure becomes brittle since the system becomes stuck in a local energy minimum. If the temperature is decreased slowly, the local energy minima can be avoided due to the thermal fluctuations, and the structure becomes more ordered and stable, and the minimum free energy conformation may be reached.

Simulated annealing works in analogy to this. In order to escape from local minima of a cost or energy function, steps towards worse states (i.e. higher cost) should be taken often in the beginning (at high temperature) and occasionally later at lower temperatures. This is done in a Monte Carlo simulation with an artificial temperature parameter that has absolutely no physical meaning. The probability of acceptance depends on the change in cost (huge increases should be accordingly improbable) as well as on the number of iterations (since the system is closer to the "stable" optimum towards the end). Given an infinite amount of time, it can be shown that simulated annealing will approach the optimal solution to any finite problem [11]. Simulated annealing can be used to minimize any cost function, and has for instance been used for multiple alignment [24].

Simulated annealing depends on an artificial temperature $T$ that decreases over time. Initially the temperature should be high enough to give an "unstable system" – in this case an alignment prone to changes. As more and more changes are sampled, the temperature decreases to "stabilize" the system. Normally the temperature decreases exponentially [24], although there is no theoretical reason for this. If the new cost is lower than the previous, the change is always accepted. If the change increases the cost, the chance of acceptance $P$ depends on the old cost $c_{OLD}$, the new (larger) cost $c_{NEW}$ and the temperature $T$:

$$P = \exp\left(-\frac{c_{NEW} - c_{OLD}}{T}\right) \qquad (4)$$

This is known as the *Metropolis–Hastings* algorithm [12, 20]. Using this, the possible states are sampled based on the cost of the current state. Since a state only depends on the previous state, this generates a Markov chain. In combination, MCMC using simulated annealing can be used to sample the solution space of multiple alignments and RNA structures. Changes can be made by moving the gaps in the alignment and by adding or removing basepairs in the structure, and the move is either rejected or accepted based on the change in cost.

The initial alignment is built by adding gaps at random to all sequences until they have equal length. By default, the length of the initial alignment is $1.06 \cdot L_{max}$ where $L_{max}$ is the length of the longest sequence. The moves through the solution space can either affect the sequence alignment or the structure. Since it makes little sense to try and deduce a common structure from randomly aligned sequences, the first iterations are purely sequence moves. As the alignment becomes more stable, we start doing a combination of sequence and structure moves.

The total number of iterations performed depends both on the length of the longest sequence, since that affects the number of structure moves needed, and on the the size of the alignment, since that affects the number of sequence moves needed. The alignment size is measured as the total number of nucleotides in the dataset, $N_{total}$. The dependencies are denoted $N_{dep}$ and $L_{dep}$, respectively, and the number of iterations is found as:

$$I = N_{dep} \cdot N_{total} + L_{dep} \cdot L_{max}$$

We use $N_{dep} = 1000$ and $L_{dep} = 1700$ as default. After initially only performing sequence moves, a mixture of alignment and structure altering moves are performed. The structure moves are initiated either after a fixed fraction of the total number of iterations or, as per default, after $N_{dep} {\cdot} N_{total}$ iterations. The remaining iterations are a mix of sequence and structure moves. The ratio between the two is set by a parameter $R$. Per default, $R = 0.75$ of the last iterations are structure moves.

Initially all moves are accepted (i.e. a temperature of infinity is used) and the first $0.1\%$ of the iterations are used to determine a good starting temperature. These results are used to estimate the standard deviation $\sigma$ of the cost distribution. By deciding on the desired initial probability of acceptance $P_0$ the temperature $T_0$ can be determined:

$$T_0 = -\frac{\sigma}{\log_2(P_0)}$$

We use $P_0 = 0.99$ as default. The scaling of the temperature has to make sure that we end up close to $T = 0$. An exponential scaling is used:

$$T_i = T_{i-1} \cdot \tau \text{ where } 0 < \tau < 1$$

Wanting the final temperature to be $T_{final} = 10^{-5}$, this yields:

$$\tau = \exp\left(\frac{\log_2\left(\frac{10^{-5}}{T_0}\right)}{I}\right)$$

*Sequence moves.* The moves aimed at changing the sequence alignment do this by moving gaps in the sequences. They can of course be moved without altering the order of the nucleotides. Three different types of moves are implemented which in combination ensures that the alignment can be reduced, extended and altered locally.

- **Gap block move:** Local changes are facilitated through *gap blocks*. A gap block is a subsequence consisting of 1 or more consecutive gaps in 1 or more aligned sequences. To make this move, a random gap in a random sequence is picked. Then the gap block is extended vertically with probability 0.85 through the other sequences containing a gap at that position. Afterwards, the gap block is extended horizontally to both sides with probability 0.85 if all the chosen sequences contain a gap there. Finally, the gap block is moved to a randomly chosen new position in the alignment. The procedure is illustrated in Fig. 1 and constitutes $85\%$ of the sequence moves.
- **Gap insertion:** Insertion of gaps has to insert the same number of gaps in all sequences. One could insert the gaps at random positions in all sequences, but that would greatly affect the cost of the alignment. Instead, the gaps are inserted in either end of the alignment. From these positions the gaps can diffuse into the alignment as needed. These moves constitute $10\%$ of the sequence moves.
- **Gap deletion:** Removing gaps is done by locating gap columns, i.e. columns in the alignment containing a gap in all of the sequences. Using a well–defined cost function, superfluous gaps are placed in the same columns of the alignment. These moves constitute $5\%$ of the sequence moves.

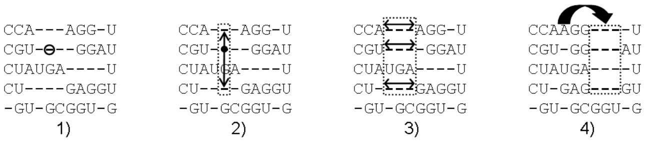

**Fig. 1.** Illustration of the gap block moves used in the sampling approach. 1) Choose a gap at random in a sequence, 2) Extend vertically, 3) Extend horizontally, 4) Move to new position in alignment.

*Structure moves.* Structural moves either add or delete basepairs. The structure is forced to contain only non–crossing basepairs (i.e. prediction of pseudoknots is not yet supported), and a minimum loop length of 3 nucelotides is ensured. Using the three simple moves described below, the structure can be built, extended and reduced.

- **Adding a basepair:** A new basepair is added by choosing a nucleotide pair $(i, j)$ at random and adding it to the structure if it does not violate the constraints. These moves constitute $70\%$ of the structure moves.
- **Extending a stem:** A stem is extended by choosing a basepair $(i, j)$ already in the structure. The stem that includes basepair $(i, j)$ is then extended by adding a new basepair to it, with a 50% chance of extending the stem either internally or externally. These moves constitute $20\%$ of the structure moves.
- **Deleting a basepair:** Deleting a basepair is done by choosing a pair $(i, j)$ in the structure and removing it. This cannot lead to any new violations of the constraints. These moves constitute $10\%$ of the structure moves.

## Datasets

Since consensus structures were not available from BRaliBase II [7] at the time, we sampled alignments with consensus structures in much the same way as in the BRaliBase study. MASTR relies on the partition function to calculate basepair probability matrices, so we have chosen to use only short (appr. 70 - 250 nucleotides) sequences in the test. There are known problems when using the partition function to calculate basepair probability matrices for long sequences. Hence, the program will not perform well on long sequences until this has been dealt with.

Datasets were generated from large, trusted seed alignments obtained from Rfam [10]. The 5 families chosen are tRNA, 5S ribosomal RNA (5S rRNA), U5 spliceosomal RNA (U5), Hepatitis C virus internal ribosome entry site (IRES) and TPP riboswitch THI element (TPP).

From each RNA family, a number of datasets were sampled. Each dataset has an average pairwise identity within a specified 10% interval. These intervals go from 30% to 100% with 5% overlap, i.e. $30\% - 40\%, 35\% - 45\%, 40\% - 50\% \ldots$ This sampling procedure gave 14 datasets from tRNA, 5S rRNA and U5, 8 datasets from TPP, and 2 datasets from IRES. In total, 52 datasets were sampled and details on average pairwise identity, number of sequences and average sequence lengths can be seen in Table 1. The datasets can be obtained from http://mastr.binf.ku.dk/.

## 3 RESULTS

The program MASTR is implemented in C++ and tested against the programs FoldalignM, LocARNA and RNA Sampler. RNAcast is used to produce input to RNAforester, and Clustal alignments were used as input to RNAalifold. All programs were used with their default settings except for RNAforester where the clustering cutoff had to be changed in order to produce complete alignments of all sequences. FoldalignM does not predict a single consensus structure but returns a structure for each sequence in the final alignment. Therefore, we define the consensus structure to be the basepairs that are predicted for all sequences. The other programs all predict a single multiple alignment and consensus structure. To evaluate the predicted structures, Matthew's Correlation Coefficient (*MCC*) is used:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Let $TP$ be the number of truly predicted basepairs, $FP$ be the number of predicted basepairs not in the reference structure, and $FN$ be the number of basepairs in the reference structure not predicted by the program. $TN$ is defined here as the number of possible basepairing interactions in a sequence that are not predicted and not in the reference structure, i.e. pairs of nucleotide $xy$ that are at least 4 nucleotides apart, and where $xy \in \{AU, UA, CG, GC, UG, GU\}$.

To evaluate the quality of the alignment, the Sum of Pairs score (*SPS*) [38] is used. SPS is a sensitivity–like measure that compares the predicted alignment to a reference. For each pair of aligned sequences, the number of aligned positions that are present in both the prediction and the reference alignment is counted. The total number of correctly aligned positions is then compared to the total number of aligned non–gap pairs present in the reference alignment. This yields a number between 0 and 1 where 1 is perfect correspondence between prediction and reference.

In Fig. 2 the performance of the programs are compared in terms of structure quality (plot a), alignment quality (plot b) and running time (plot c) as a function of the average pairwise identity of the datasets (%ID). The plots are averages over the different RNA families used for each %ID point.

The test shows that MASTR can predict consensus structures of a quality comparable to other existing methods. On the lowest identity datasets MASTR is outperformed by RNA Sampler, but after appr. 45% ID the structure predictions of MASTR are on average better than or comparable to the best programs tested. MASTR is consistently better than or comparable to all other methods regarding alignment quality. As it can be seen, MASTR is clearly faster than FoldalignM by up to an order of magnitude, while LocARNA is even faster. Clustal + RNAalifold is of course by far the fastest method used. RNAforester has a running time comparable to LocARNA but produces consistently worse alignments – probably due to the fact that all sequences had to be included in the same alignment for comparison. The structure predictions are comparable to FoldalignM.

The dip in the quality of the structure predictions that is visible for all methods in the highest identity range could be explained by the lack of covariation in these datasets. Most of the methods rely on some signal from compensatory mutations, and this signal diminishes as the sequences become too similar. RNA Sampler does not depend on a covariation term which could explain why the dip is less prominent here. Likewise, FoldalignM shows an almost monotone increase in the predictions as a function of identity.

Since RNA Sampler seems to be the best of the other methods tested here, a more detailed comparison of MASTR and RNA Sampler can be seen in Table 1. In total, 52 datasets are used. The running time is on the same scale for the two programs, although MASTR is in general slightly faster. The structure predictions are better than or equal to RNA Sampler in 28 cases (54%), and the alignments are better than or equal to RNA Sampler in 43 cases (83%). The differences seem to depend both on the RNA family and on the level of identity.

## 4 DISCUSSION

We have developed a new algorithm for simultaneous alignment and structure prediction of multiple non-coding RNA sequences. As shown above, *MASTR* is highly competitive both in terms of structure prediction quality, sequence alignment, and running time. The program can also handle larger datasets than e.g. RNA Sampler or FoldalignM.

Although we have not used it in the above tests, it is also possible to add structural constraints if some knowledge is available about one of the sequences (e.g. known basepairs, knowledge about upstream or downstream interactions, or knowledge about non-basepaired positions). Additionally, already aligned sequences can be used as input with or without a consensus structure.

As the testing of the program showed, MASTR does not have top performance on very dissimilar sequences. In this range, one would assume that covariance is important, and it is therefore interesting that RNA sampler, which does not use covariance, is better. One possible explanation for this is that covariation in itself is not enough to deduce structure from alignment. Covariation is only an indicator of conserved basepairs, but it is not sufficient to predict pairing columns (this corresponds well with our previous study [23]). MASTR builds up the structure in small steps, which might make it vulnerable to erronously high covariation, whereas RNA Sampler makes sure that the alignment is structurally sound by fixing whole stems. MASTR therefore needs to have a relatively stable (and correct) alignment before predicting structure. This could explain the relatively poor performance on low identity datasets, and this should be explored further.

In future work, we would like to make a local version of the program. This would be ideal for dealing with long sequences where there are known problems with the standard basepair probability matrices. Since MASTR does not have the same limitations towards crossing basepairs as pure energy based methods, an extension to include the prediction of pseudoknots will also be pursued.

One of the advantages of MASTR is that the optimization is decoupled from the cost function, which makes it very easy to change the latter. It also makes it reasonably straight-forward to add phylogenetic prediction to the program, which would be similar to the goal of SimulFold [28], but MASTR functions in a very different way. We would also like to make it possible to input a set of related and already aligned sequences together with the set of unaligned sequences. Thus, new sequences can be aligned to reference alignments in a structurally sound manner.

**Table 1.** Detailed comparison of the predictions by MASTR and RNA Sampler

| | | | | MASTR | | RNA Sampler | |
|---|---|---|---|---|---|---|---|
| Family | ID | Seqs | Length | MCC | SPS | MCC | SPS |
| tRNA | 35 | 11 | 74 | 0.72 | 0.59 | **0.90** | **0.68** |
| | 40 | 11 | 72 | 0.89 | **0.84** | **0.97** | 0.79 |
| | 45 | 12 | 71 | 0.89 | *0.88* | **0.94** | *0.88* |
| | 50 | 12 | 73 | **0.99** | **0.97** | 0.85 | 0.95 |
| | 44 | 12 | 73 | **1.00** | **0.98** | 0.98 | 0.91 |
| | 60 | 11 | 73 | *1.00* | *0.99* | *1.00* | *0.99* |
| | 65 | 11 | 73 | **1.00** | *0.99* | 0.86 | *0.99* |
| | 68 | 7 | 73 | 0.89 | **1.00** | **0.99** | 0.99 |
| | 75 | 5 | 72 | *1.00* | **0.98** | *1.00* | 0.97 |
| | 80 | 6 | 72 | *1.00* | *0.99* | *1.00* | *0.99* |
| | 84 | 6 | 69 | **0.14** | *0.98* | -0.31 | *0.98* |
| | 90 | 7 | 72 | 0.83 | **1.00** | **0.98** | 0.98 |
| | 95 | 8 | 73 | *0.97* | *1.00* | *0.97* | *1.00* |
| | 97 | 6 | 73 | 0.34 | *1.00* | **0.48** | *1.00* |
| 5S rRNA | 36 | 5 | 110 | 0.45 | **0.62** | **0.51** | 0.60 |
| | 42 | 6 | 111 | **0.66** | **0.74** | 0.62 | 0.59 |
| | 45 | 8 | 116 | **0.54** | **0.76** | 0.41 | 0.59 |
| | 50 | 9 | 113 | **0.59** | **0.74** | 0.52 | 0.73 |
| | 55 | 10 | 113 | **0.56** | *0.91* | 0.40 | *0.91* |
| | 60 | 11 | 118 | **0.89** | **0.96** | 0.75 | 0.90 |
| | 65 | 11 | 117 | **0.83** | *0.97* | 0.81 | *0.97* |
| | 69 | 10 | 116 | **0.52** | **0.95** | 0.50 | 0.92 |
| | 75 | 9 | 115 | **0.58** | **0.96** | 0.53 | 0.95 |
| | 80 | 11 | 118 | **0.93** | **1.00** | 0.85 | 0.98 |
| | 85 | 15 | 117 | 0.11 | **1.00** | **0.13** | 0.99 |
| | 89 | 15 | 117 | -0.04 | 0.97 | **0.01** | **0.98** |
| | 95 | 20 | 117 | **0.20** | *1.00* | 0.12 | *1.00* |
| | 97 | 13 | 117 | **0.38** | **1.00** | 0.10 | 0.98 |
| U5 | 36 | 5 | 122 | 0.40 | 0.33 | **0.44** | **0.38** |
| | 41 | 6 | 123 | 0.48 | 0.37 | **0.49** | **0.42** |
| | 44 | 9 | 120 | 0.46 | 0.45 | **0.64** | **0.49** |
| | 51 | 8 | 120 | 0.50 | 0.51 | **0.63** | **0.56** |
| | 55 | 10 | 118 | 0.67 | 0.61 | **0.72** | **0.64** |
| | 60 | 9 | 115 | 0.71 | **0.68** | **0.82** | 0.65 |
| | 65 | 9 | 114 | **0.78** | **0.76** | 0.52 | 0.68 |
| | 70 | 10 | 115 | 0.87 | *0.81* | **0.93** | *0.81* |
| | 75 | 8 | 116 | 0.90 | 0.82 | **0.93** | **0.84** |
| | 80 | 8 | 119 | **0.50** | **0.83** | 0.24 | 0.76 |
| | 86 | 7 | 116 | *0.96* | **0.96** | *0.96* | 0.94 |
| | 90 | 9 | 115 | **0.95** | **0.98** | 0.92 | 0.95 |
| | 95 | 17 | 115 | **1.00** | **1.00** | 0.98 | 0.99 |
| | 97 | 7 | 116 | **0.96** | *1.00* | 0.95 | *1.00* |
| TPP | 35 | 8 | 122 | 0.33 | **0.42** | **0.45** | 0.40 |
| | 41 | 9 | 104 | 0.37 | **0.65** | **0.71** | 0.51 |
| | 45 | 11 | 105 | 0.43 | **0.75** | **0.61** | 0.64 |
| | 50 | 11 | 103 | 0.55 | *0.72* | **0.65** | *0.72* |
| | 55 | 11 | 107 | 0.70 | **0.86** | **0.74** | 0.78 |
| | 58 | 7 | 102 | 0.63 | **0.91** | **0.69** | 0.81 |
| | 61 | 4 | 101 | **0.66** | **0.97** | 0.59 | 0.85 |
| | 70 | 4 | 90 | **0.45** | **0.86** | 0.31 | 0.85 |
| ires | 69 | 2 | 249 | 0.40 | 0.59 | **0.46** | **0.86** |
| | 73 | 4 | 250 | **0.54** | **0.91** | 0.45 | 0.86 |

For each dataset, the average pairwise identity (ID) is shown along with the number of sequences (Seqs) and the average sequence length (Length). The results for MASTR and RNA Sampler are detailed (MCC: Structure quality, SPS: Alignment quality). For each dataset, the best results are highlighted with **bold**, and identical results with *italics*.

## ACKNOWLEDGEMENT

SL acknowledges support from the Novo Scolarship Program. The work was supported by the Carlsberg Foundation and the Danish National Research Foundation. The authors thank two anonymous referees for their helpful comments; especially one reviewer for interesting comments on covariation.

## REFERENCES

[1] Athanasius F. Bompfünewerer Consortium, R. Backofen, S. H. Bernhart, C. Flamm, C. Fried, G. Fritzsch, J. Hackermuller, J. Hertel, I. L. Hofacker, K. Missal, A. Mosig, S. J. Prohaska, D. Rose, P. F. Stadler, A. Tanzer, S. Washietl, and S. Will. RNAs everywhere: genome-wide annotation of structured RNAs. *J. Exp. Zoolog. (Mol. Dev. Evol.)*, 308(1):1–25, 2007.

[2] A. F. Bompfünewerer, C. Flamm, C. Fried, G. Fritzsch, I. L. Hofacker, J. Lehmann, K. Missal, A. Mosig, B. Müller, S. J. Prohaska, B. M. Stadler, P. F. Stadler, A. Tanzer, S. Washietl, and C. Witwer. Evolutionary patterns of non-coding RNAs. *Theory in Biosciences*, 123(4):301–369, 2005.

[3] J. Cheng, P. Kapranov, J. Drenkow, S. Dike, S. Brubaker, S. Patel, J. Long, D. Stern, H. Tammana, G. Helt, V. Sementchenko, A. Piccolboni, S. Bekiranov, D. Bailey, M. Ganesh, S. Ghosh, I. Bell, D. Gerhard, and T. Gingeras. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science*, 308:1149–1154, 2005.

[4] R. Das and D. Baker. Automated de novo prediction of native-like RNA tertiary structures. *PNAS*, 104(37):14664–14669, 2007.

[5] Y. Ding and C. E. Lawrence. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Research*, 31(24):7280–7301, 2003.

[6] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[7] P. Gardner, A. Wilm, and S. Washietl. A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Research*, 33(8):2433–2439, 2005.

[8] R. Giegerich, B. Voß, and M. Rehmsmeier. Abstract shapes of RNA. *Nucleic Acids Research*, 32(16):4843–4851, 2004.

[9] J. Gorodkin, L. J. Heyer, and G. D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Research*, 25(18):3724–3732, 1997.

[10] S. Griffiths-Jones, S. Moxon, M. Marshall, A. Khanna, S. R. Eddy, and A. Bateman. Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Research*, 33:D121–D124, 2005.

[11] O. Häggström. *Finite Markov chains and algorithmic applications*. Cambridge University Press, 2002.

[12] W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[13] J. H. Havgaard, R. B. Lyngsø, G. D. Stormo, and J. Gorodkin. Pairwise local structure alignment of RNA sequences with sequence similarity less than 40%. *Bioinformatics*, 21(9):1815–1824, 205.

[14] J. Hein. A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences when the phylogeny is given. *Molecular Biological Evolution*, 6(6):649–668, 1989.

[15] M. Höchsmann, T. Töller, R. Giegerich, and S. Kurtz. Local similarity of RNA secondary structures. In *Proc of the IEEE Bioinformatics Conference*, pages 159–168, 2003.

[16] I. Hofacker. Vienna RNA secondary structure server,. *Nucleic Acids Research*, 31(13):3429–3431, 2003.

[17] I. Hofacker, S. Bernhart, and P. Stadler. Alignment of RNA base pairing probability matrices. *Bioinformatics*, 20(14):2222–2227, 2004.

[18] I. L. Hofacker, M. Fekete, and P. F. Stadler. Secondary structure prediction for aligned RNA sequences. *Journal of Molecular Biology*, 319(5):1059–1066, 2002.

[19] F. Jossinet, T. E. Ludwig, and E. Westhof. RNA structure: bioinformatic analysis. *Current Opinion in Microbiology*, 10(3):279–285., 2007.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[21] B. Knudsen and J. Hein. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, 15(6):446–454, 1999.

[22] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, 31(13):3423–3428, 2003.

[23]S. Lindgreen, P. P. Gardner, and A. Krogh. Measuring covariation in RNA alignments: Physical realism improves information measures. *Bioinformatics*, 22(24):2988–2995, 2006.

[24]A. V. Lukashin, J. Engelbrecht, and S. Brunak. Multiple alignment using simulated annealing: branch point definition in human mRNA splicing. *Nucleic Acids Research*, 20(10):2511–2516, 1992.

[25]D. H. Mathews and D. H. Turner. Dynalign: An algorithm for finding the secondary structure common to two RNA sequences. *Journal of Molecular Biology*, 317(2):191–203, 2002.

[26]J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990.

[27]I. M. Meyer. A practical guide to the art of RNA gene prediction. *Briefings in Bioinformatics*, 2007.

[28]I. M. Meyer and I. Miklos. SimulFold: Simultaneously inferring RNA structures including pseudoknots, alignments and trees using a Bayesian MCMC framework. *PLoS Computational Biology*, 3(8):e149, 2007.

[29]R. Nussinov, G. Pieczenik, J. Griggs, and D. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35(1):68–82, 1978.

[30]B. Onoa and I. Tinoco Jr. RNA folding and unfolding. *Current Opinions in Structural Biology*, 14(3):374–379, 2004.

[31]J. Pedersen, G. Bejerano, A. Siepel, K. Rosenbloom, K. Lindblad-Toh, E. Lander, J. Kent, W. Miller, and D. Haussler. Identification and classification of conserved RNA secondary structures in the human genome. *PLoS Comput Biol.*, 2(4):e33, 2006.

[32]J. Reeder and R. Giegerich. Consensus shapes: An alternative to the Sankoff algorithm for RNA consensus structure prediction. *Bioinformatics*, 21(17):3516–3523, 2005.

[33]D. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825, 1985.

[34]B. A. Shapiro, Y. G. Yingling, W. Kasprzak, and E. Bindewald. Bridging the gap in RNA structure prediction. *Current Opinion in Structural Biology*, 17(2):157–165, 2007.

[35]S. Siebert and R. Backofen. MARNA: multiple alignment and consensus structure prediction of RNAs based on sequence structure comparisons. *Bioinformatics*, 21(16):3352–3359, 2005.

[36]K. Tai. The tree-to-tree correction problem. *Journal of the ACM*, 26:422–433, 1979.

[37]J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.

[38]J. D. Thompson, F. Plewniak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, 27(13):2682–2690, 1999.

[39]E. Torarinsson, J. Havgaard, and J. Gorodkin. Multiple structural alignment and clustering of RNA sequences. *Bioinformatics*, 23(8):926–932, 2007.

[40]S. Washietl, I. L. Hofacker, M. Lükasser, A. Huttenhofer, and P. F. Stadler. Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. *Nature Biotechnology*, 23(11):1383–1390, 2005.

[41]S. Will, K. Reiche, I. Hofacker, P. Stadler, and R. Backofen. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, 3(4):e65, 2007.

[42]X. Xu, Y. Ji, and G. Stormo. RNA Sampler: A new sampling based algorithm for common RNA secondary structure prediction and structural alignment. *Bioinformatics*, 2007.

[43]K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing*, 18(6):1245–1262, 1989.

[44]M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981.

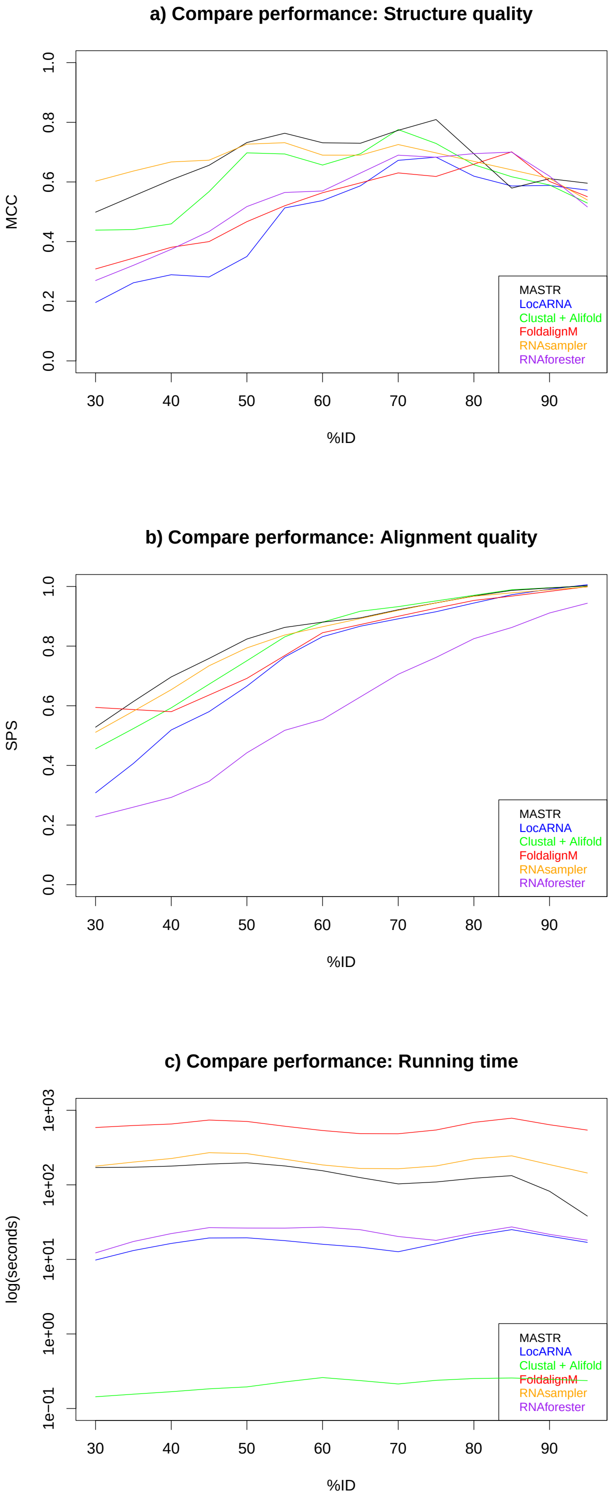

**Fig. 2.** The performance of the tested programs in terms of structure prediction (a), alignment quality (b) and running time (c) as a function of average pairwise identity ($\%ID$). Each plot shows the performance as the average over the RNA families used. Note that plot c is on a logarithmic scale. Black: MASTR, Blue: LocARNA, Green: Clustal+RNAalifold, Red: FoldalignM, Orange: RNA Sampler, Purple: RNAcast+RNAforester.