De-novo identification of homologous RNA secondary structure domains using base-pairing probabilities

Stefan E Seemann[1,2], Martin A Smith[1,3], John S Mattick[1,3]
[1]*Garvan Institute of Medical Research, 384 Victoria Street, Sydney 2010, Australia*
[2]*University of Copenhagen, Groennegaardsvej 3, Frederiksberg, Denmark*
[3]*St Vincents Clinical School, UNSW, Sydney 2010, Australia*
seemann@rth.dk

Keywords: RNA secondary structure, Basepair probability, Structure-based alignment

Abstract: Non-protein coding RNAs (ncRNAs) are the prevalent transcriptional product of higher eukaryote genomes. Their varied biological functions are governed by both their sequence composition and their higher-order structural conformation. The uncertainty of secondary structure prediction algorithms for single RNA sequences in conjunction with the limited diversity of well-characterised RNA structures have restricted the identification and annotation of novel functional ncRNA domains. Here, we present a unified computational methodology for the identification of common RNA secondary structures from a set of sequences, requiring little to no user intervention while being fully customisable. We compare the performance of several state of the art tools for pairwise secondary structure alignment with `DotAligner`, a novel algorithm we developed that considers the ensemble of sub-optimal RNA base pairings between two RNA sequences simultaneously. Through hierarchical clustering and bootstrapping analysis, our method identifies statistically significant clusters of homologous, structured RNA domains with no limitations on the sequence composition of the input. We successfully identify known RNA secondary structures mixed in with randomised controls, as well as novel structured domains from various previously published transcriptomic datasets.

1 INTRODUCTION

The structure of RNA molecules is an essential functional criteria of many non-coding RNAs (ncRNAs), such as the stem-loop of microRNAs and the double stem-loop RNA motifs of the HOTAIR long ncRNA (**?**). NcRNAs can be divided in RNA families of similar inherent functionality, structures, or composition. The largest collection of RNA families is the Rfam database with 2,208 families in its version 11.0 (**?**). However, high-throughput sequencing continuously uncovers novel non-coding RNA transcripts and genome-wide RNA structure predictions have revealed hundreds of thousands putative conserved RNA secondary structures. We hypothesize that the RNA secondary structure is the scaffold for intermolecular interactions of many ncRNA-driven regulatory pathways. Protein binding domains of RNA molecules may evolve totally independent from sequence and, instead, may be solely determined by structure. It has been shown that if the sequence similarity falls below 60% sequence comparison will not find anymore domain similarities that are based on structure (**?**). In addition, competing structures and suboptimal structures may support or even drive the functionality of an RNA domain. Hence, methods are needed that find structural similarity independent from sequence conservation and freed from one single optimal RNA secondary structure.

For clustering of RNA domains a dissimilarity measurement of all pairs of query structures is needed. The dissimilarity is described through a pairwise weighted string alignment with arbitrary pairwise dependencies (for base pairings). The Needleman-Wunsch (2) algorithm solves the maximum weight string alignment problem by dynamic programming in $O(N^2)$ by preserving the sequence order and maximizing the similarity. The consideration of pairs of nucleotides in each sequence that form intra-molecular interactions extends the problem to pairwise dependencies among positions in each string. This problem variant is MAX-SNP-hard. However, the problem can be attacked by intelligent heuristics that avoid the examination of all possible aligning states.

Simultaneous alignment and folding (**?**) is the ac-

knowledged gold standard to predict the consensus structure and alignment of a set of related RNA sequences. Because the Sankoff algorithm is practically not applicable, the pre-calculation of the structure ensemble of each sequence, *e.g.* basepair probabilities in thermodynamically equilibrated RNA structure ensembles (**?**), is used by different methods to speed up the calculation of structure-based alignments. The programs `pmcomp` for pairwise and `pmmulti` for multiple alignments (**?**), as well as `LocaRNA` (**?**) score the alignment based on the notion of a common secondary structure. Despite of the usage of the basepair probability matrices these methods extract the maximum-weight common secondary structure but do not explicitely consider suboptimal structures in the alignment. The pairwise alignment of basepair probability matrices (dot plots) has been first introduced by `CARNA` (**?**; **?**). `CARNA` finds iteratively better alignments with an effective constraint programming technique using a branch and bound scheme (propagator).

Beside of `LocaRNA` and a method based on directed acyclic graph kernels (**?**), the alignment-free approach `ClustGraph` (**?**) has been used to cluster RNA structure in common domains. Here, we propose an alternative heuristic for the pairwise weighted string alignment with arbitrary pairwise dependencies that can deliver dissimilarity scores of dot plots in time close to an Needleman-Wunsch alignment which makes the approach applicable for clustering of large numbers of putative RNA domains.

2 IMPLEMENTATION

As described in (**?**) the weight W of alignment A of two arc-annotated sequences (S_a, P_a) and (S_b, P_b) is defined by

$$\begin{aligned} W(A) &= \sigma(A) + \tau(A) + \gamma(A) \\ &= \sum_{(i,i')\in A} \sigma(i,i') + \sum_{\substack{(i,j)\in P_a,\\ (i',j')\in P_b,\\ (i,i')\in A,\\ (j,j')\in A}} \tau(i,j,i',j') + \gamma \times N \end{aligned} \tag{1}$$

where S is a sequence and P is a base pairing probability matrix, $\sigma(i,i')$ is the similarity of sequence positions $S_a[i]$ and $S_b[i']$, $\tau(i,j,i',j')$ is the similarity of arcs $(i,j) \in P_a$ and $(i',j') \in P_b$, and γ is the gap cost associated with each sequence position that is not matched ($N = |S_a| + |S_b| - 2|A|$). The alignment problem finds the maximal $W(A)$. As its solution is MAX-SNP-hard, in praxis heuristics are used to find near-optimal solutions. Here, we present the program `DotAligner` which solves the related problem of aligning two basepair probability matrices (dot plots). `DotAligner` employs the heuristics alignment-envelope, which imposes constraints to sub-optimal string alignments, and fold-envelope, which imposes constraints to pre-calculated base pairing probabilities, to built pairwise sequence-structure alignments. We make use of the observation that large samples from the ensemble of stochastic sequence alignments contain the correct structure-based alignment with significant probability even though the optimal sequence alignment deviates significantly from the structural alignment (**?**). A major criteria for the implementation was a fast running time to make `DotAligner` applicable for RNA structure clustering of large data sets. The alignment procedure consists of two steps:

1. pairwise probabilistic string alignments,
2. stochastic backtracking of string alignments and combined weight of corresponding dot plot alignments.

In the following we describe the alignment procedure and its weight functions implemented in `DotAligner`.

2.1 Pairwise probabilistic string alignments

In step 1 the computation of the partition function over all canonical pairwise string alignments is adapted from probA (**?**). The probability of an alignment A in the ensemble of all alignments $Z(T)$ is

$$Pr(A;T) = \frac{1}{Z(T)} exp(\beta W(A)), \tag{2}$$

where $\beta = 1/T$. The parameter T is analogous to the temperature in the thermodynamic interpretation of the alignment problem and determines the relative importance of the optimal string alignment. If $T = 1$ then we recover the 'true' probability, if $T \to 0$ then $Pr(A;0) = 0$ for all alignments with a score $W(A)$ less then the score of the optimal string alignment, and if $T \to \infty$ then all alignments have the same $Pr(A,\infty) = 1/Z(\infty)$. Hence, T controls the search space of suboptimal alignments for step 2. The algorithm runs in $O(N^3)$ for calculating the partition function. The weight function $W(A)$ of the probA implementation is changed to explore the ensemble of dot plot alignments. We reduce the sequence-structure alignment problem to a two-dimensional problem similar to the metric introduced in StrAL (**?**). Hence, step 1 considers only the similarity σ and the gap cost γ described in equation **??**:

$$W_{\text{Step1}}(A) = \sigma(A) + \gamma(A) \tag{3}$$

The similarity $\sigma(i,i')$ for matched sequence positions $S_a[i]$ and $S_b[i']$ takes into account sequence similarity M_{Seq} and the similarity in their unpaired probabilities $\Delta\omega(i,i')$ weighted by the parameter θ:

$$\sigma(i,i') = \theta \times M_{Seq}^{(i,i')} + (1-\theta) \times \Delta\omega(i,i') \tag{4}$$

$M_{Seq}^{(i,i')}$ is 1 if sequence positions $S_a[i]$ and $S_b[i']$ match and else 0. The similarity of unpaired probabilities is defined as

$$\Delta\omega(i,i') = \begin{cases} 0 & \text{if } \omega(i) == 0 \\ & \text{and } \omega(i') == 0 \\ 1 - |\omega(i) - \omega(i')| & \text{else} \end{cases} \tag{5}$$

so that $\Delta\omega = (0,1)$. Alternatively a statistical substitution model R_{Seq} replaces the sequence similarity and is multiplied with the ζ weighted sum of $\Delta\omega$ and the similarity in ratios of upstream pairing probability $\Delta\omega^{up}$:

$$\begin{aligned} \sigma(i,i') = R_{Seq}^{(i,i')} \times \zeta \times \Delta\omega(i,i') + \\ R_{Seq}^{(i,i')} \times (1-\zeta) \times \Delta\omega^{up}(i,i') \end{aligned} \tag{6}$$

R_{Seq} is a 4×4 matrix of probabilities for observing a given substitution relative to background nucleotide frequencies. We use the log-odd scores L from the RIBOSUM85-60 matrix introduced in (**?**) which are transformed to probabilities R_{Seq} by $2^{L(i,i')}/(1+2^{L(i,i')})$. The ratio of upstream pairing probability ω^{up} is defined as

$$\omega^{up}(i) = \sum_{k=1}^{i-1} \psi(k,i) / \sum_{k=1}^{|S|} \psi(k,i) \tag{7}$$

where $i \in S$, $|S|$ is the length of sequence S, and $\psi(k,i)$ is the pairing probability of sequence positions $S[k]$ and $S[i]$. The downstream pairing probability is implicitly considered in the weight function through the usage of unpaired probability and upstream pairing probability. The gap term in equation **??** is replaced with affine gap costs:

$$\gamma(A) = l \times g_o + (N-l) \times g_{ext} \tag{8}$$

where l is the number of initiation gaps, N is the number of all gaps, g_o is the penalty for opening a gap and g_{ext} is the penalty for gap extensions. Start and end gaps are considered as free.

2.2 Stochastic backtracking and combined weight of dot plot alignments

In step 2 a properly weighted sample of stochastic pairwise string alignments in the alignment ensemble is examined for their sequence-structure similarity. The stochastic backtracking is adapted from probA (**?**) for selecting s suboptimal string alignments A_s. The combined weight W_{Step2} is a variant of equation **??** to explore the similarity of the corresponding dot plot alignments:

$$W_{\text{Step2}}(A_s) = \kappa \times \frac{W_{\text{Step1}}(A_s)}{|A_s|} + (1-\kappa) \times \frac{\tau(A_s)}{|\text{Match}_{A_s}|^2} \tag{9}$$

where the parameter κ weights for each alignment A_s between the sequence-based similarity $W_{\text{Step1}}(A_s)$ normalized by alignment length $|A_s|$ and dot plot similarity $\tau(A_s)$ normalized by the number of aligned bases $|\text{Match}_{A_s}|$ in alignment A_s. Similar to equation **??** the dot plot similarity τ sums the parameter θ weighted similarity of aligned basepairs M_{paired} and the similarity in their pairing probabilities $\Delta\psi$:

$$\tau(i,j,i',j') = \theta \times M_{paired}^{(i,j,i',j')} + (1-\theta) \times \Delta\psi(i,j,i',j') \tag{10}$$

where $M_{paired}^{(i,j,i',j')}$ is 1 if $S_a[i]$ and $S_a[j]$ as well as $S_b[i']$ and $S_b[j']$ form canonical basepairs (G-C, C-G, A-U, U-A, G-U or U-G) and else 0. The similarity in pairing probabilities $\Delta\psi$ is then calculate by

$$\Delta\psi(i,j,i',j') = \begin{cases} 0 \text{ if } \psi(i,j) == 0 \text{ and } \psi(i',j') == 0 \\ 1 - |\psi(i,j) - \psi(i',j')| \text{ else} \end{cases} \tag{11}$$

Similar to M_{Seq} in equation **??** the basepair similarity matrix M_{paired} can be replaced by a statistical substitution model R_{paired} which describes the probability for observing a given basepair substitution relative to background nucleotide frequencies:

$$\tau(i,j,i',j') = R_{paired}^{(i,j,i',j')} \times \Delta\psi(i,j,i',j') \tag{12}$$

Again the log-odd scores L from the RIBOSUM85-60 matrix (**?**) are transformed to probabilities R_{paired}.

For both sequences S_a and S_b, the pairing probability matrices P_a and P_b are computed in advance using McCaskill's algorithm, implemented in RNAfold or RNAplfold. The robustness of the alignment is improved by applying log-odds scores ψ of having a specific base pairing against the null model of a random pairing (**?**):

$$\psi(i,j) = max\left(0, log\frac{P(i,j)}{p_0} / log\frac{1}{p_0}\right) \quad (13)$$

where p_0 is the expected probability for a pairing to occur at random. The term $log\frac{1}{p_0}$ is a normalization factor that transforms the scores to a maximum of 1. $P == 1$ results in $\psi = 1$, $P > p_0$ results in $\psi > 0$, and $P \leq p_0$ results in $\psi = 0$. This transformation gives weaker similarities if low basepair probabilities are compared, but stronger similarities for high basepair probabilities. Unpaired probabilities are handled in a similar way by

$$\omega(i) = max\left(0, log\frac{1-\sum_k P(i,k)}{p_0} / log\frac{1}{p_0}\right) \quad (14)$$

where p_0 is the expected probability for an unpaired base to occur at random.

3 RESULTS

The accuracy of the proposed algorithm is assessed using the specificity (SP) and the sensitivity (SN), which are defined as follows:

$$SP = \frac{TN}{TN+FP}, \quad SN = \frac{TP}{TP+FN} \quad (15)$$

where TP is the number of correctly predicted positives, FP is the number of incorrectly predicted positives, TN is the number of correctly predicted negatives, and FN is the number of incorrectly predicted negatives. Furthermore, the area under the receiver operating characteristic (ROC) curve was used to optimize the different combinations of parameters. The ROC curve plots the true positive rates (SN) as a function of the false positive rates (1 - SP) for varying parameters.

As benchmark data set we selected 300 sequences of 10 H/ACA-box snoRNA families from Rfam version 11.0 seed alignments with average pairwise sequence identity (APSI) $< 90\%$ and sequence lengths of $>$ 130bp and $<$ 140bp: *SNORA1*, *SNORA13*, *SNORA14*, *SNORA15*, *SNORA16*, *SNORA17*, *SNORA18*, *SNORA19*, *SNORA2*, *SNORA22*. We chose only sequences of similar length because step 1 of `DotAligner` performs global alignments.

3.1 Parameter optimization

The weight function of `DotAligner` includes the following parameters whose settings are discussed in this section:

1. κ is weight of sequence (string) similarity compared to structure (dot plot) similarity
2. usage of similarity matrices M_{seq} and M_{paired} or statistical substitution matrices R_{seq} and R_{paired}
3. if similarity matrices M_{seq} and M_{paired} are used then θ is weight of sequence/basepair similarity compared to similarity of unpaired/pairing probability
4. if substitution matrix R_{seq} is used then ζ is weight of unpaired probability compared to ratio of upstream pairing probability
5. g_o and g_{ext} are the gap-open and gap-extension penalties respectively
6. s is number of examined suboptimal alignment
7. T is a measure of the relative importance of the optimal string alignment
8. p_0 is minimal considered unpaired/pairing probability and set to 0.0005

Parameter optimization by using Bralibase 2.1? Parameters 1 to 7 have to be optimized!!! Compare with reference alignments by the SPS measure introduced in Bralibase 2.1 and SCI measure (see CARNA paper).

The ROC curve in Figure **??** shows the lowest γ as most sensitive (SN $= 0.61$) and the highest γ as most specific (SP $= 1.0$) for correctly clustering the selected Rfam families. In the following we choose $\gamma = 4$, whereas the optimal gap cost lies somewhere between 3 and 4.

3.2 Benchmarking methodology

The reliability of our pairwise structure alignment algorithm at clustering homologous RNA structures was tested on a curated database of RNA structure families (cf RFAM). This enables both qualitative and quantitative performance evaluation using a gold-standard reference. We compared `DotAligner` to other RNA structure alignment and clustering tools using the following framework:

1. Generate dissimilarity matrix dM_A from $\frac{n(n-1)}{2}$ pairwise structure comparisons with each algorithm
2. Hierarchical clustering of RNA secondary structures and significance testing with `pvclust` (Suzuki R and Shimodaira H. Bioinformatics 2006).
3. Generate dissimilarity matrix dM_R from scoring metric of (1.) from curated RFAM alignments (constrained alignment).

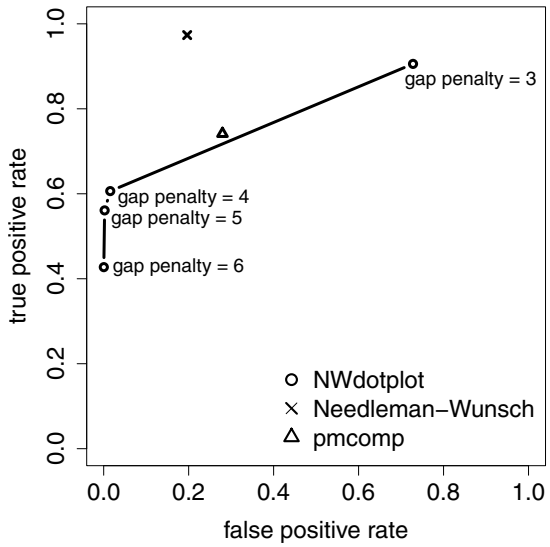

Figure 1: Performance comparison of hierarchical cluster analysis: Degree of agreement between the 10 tested Rfam families and the automated clustering based on distance scores from `DotAligner` with different gap penalties, Needleman-Wunsch algorithm and `pmcomp`.

4. Calculate the correlation coefficient between dM_A and dM_R using the Mantel correlation statistic (the cross-product between the standardised distances).

3.3 Benchmark data generation

Martins benchmark for different APSIs. Compare Rfam families with significant clusters generated by `pvclust`.

Benchmarking was performed on both complete RNA structures (global alignment) and randomly selected subsequences (local alignment) for various RFAM families, as described below.

xx RNA families were manually selected from the seed alignments of RFAM 11 (REF). *How should we limit the mean pairwise identity? All structures must be within a given range and perform several independent comparisons, i.e. one per SeqID range? Then compare the individual SeqID ranges to a sample of variable SeqIDs (without selection)?*

We employed the BuildRfamBenchmark JAVA program from (Smith M et al. NAR 2013) to generate the sample alignments for the RFAM entries listed in TableXX. The tRNA sample includes special tRNAs, like ser-tRNA with a 5th hairpin to see how the latter gets clustered by the algorithms.

RFAM ID	RNA class	average length
	5s rRNA	
	SRP	
	tRNA	
	HaCa snoRNA	
	pre-miRNA	

3.4 Complete RFAM sequences

Global alignment. More emphasis on quantitative clustering, accuracy, and correlation with control.

	SeqId 10 . . . 55		
	SP	SN	Time [s]
`DotAligner`	84.1	64.8	7.2
`CARNA`	?	?	?
`LocaRNA`	96.9	54.0	?
`FOLDALIGN`	88.6	73.7	34.8
`pmcomp`	97.9	35.7	289.9
`Needleman-Wunsch`	92.6	54.6	0.002
	SeqId 56 . . . 95		
	SP	SN	Time [s]
`DotAligner`	100	86.4	7.1
`CARNA`	?	?	?
`LocaRNA`	?	?	?
`FOLDALIGN`	97.2	79.7	37.5
`pmcomp`	100	64.5	338.4
`Needleman-Wunsch`	91.8	90.2	0.002

We compare `DotAligner` with sequence alignments (in-house implementation of Needleman-Wunsch algorithm with the blastn parameters match = 2, mismatch = -3 and gap penalty = 5 which are optimized for sequence identity of 90%) and the structure alignment tools `pmcomp` (using default parameters or larger values for parameter *-D* if the length difference of two sequences is > 5 bp), `CARNA`, and `LocaRNA`. Figure **??** shows that the sequence aligner (SP = 0.80, SN =0.97) performs very well on our benchmark set with a very high sensitivity which is most likely due to the fact that the input sequences have some degree of sequence information. `pmcomp` (SP = 0.72, SN = 0.74) performed with a medium sensitivity and specificity. With `DotAligner` we are able to find very well defined clusters (SP = 0.99), however, at the cost of sensitivity (SN = 0.61), see Figure **??**.

3.5 Fragmented RFAM sequences

Local alignment, simulating genomic screens. More emphasis on qualitative clustering

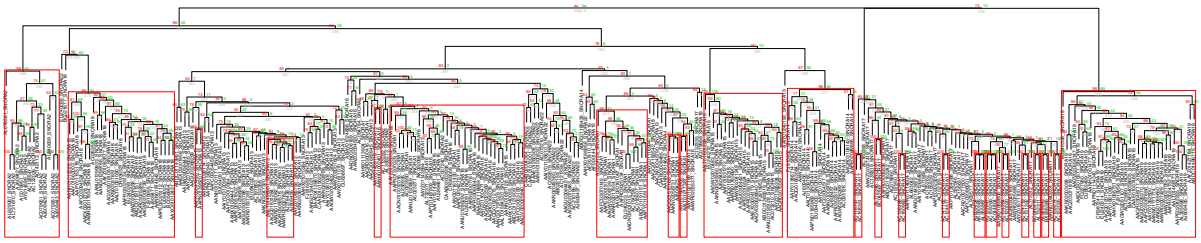

Figure 2: Automated hierarchical clustering of 300 sequences from 10 H/ACA snoRNA families. The dissimilarity matrix was calculated through `DotAligner` with gap penalty 4. The clustering was conducted by the R-package `pvclust` with multiscale bootstrap resampling with number of bootstrap 1000. We define clusters (red rectangles) as Approximately Unbiased (AU) p-values > 0.95 rejecting the hypothesis that "the cluster does not exist" with significance level 0.05.

3.6 A unified RNA structure clustering pipeline

We implemented a user-friendly pipeline that automates all steps required for the detection of homologous RNA secondary structure motifs from a set of user- provided sequences. The pipeline is implemented in BASH programming language and is designed for execution on a high performance computing server (currently, only SGE is supported). This enables non-specialists to complete such an analysis with minimal bioinformatics knowledge, while facilitating parameter modification and customization for advanced users.

In summary, the pipeline performs the following tasks on a fasta file input:

1. Generates base-pairing probability matrices for each sequence with RNAfold's parition function algorithm
2. Performs all-vs-all pairwise alignment in parallel with DotAligner (and/or CARNA, locarna, ...)
3. Generate (dis)similarity matrix from pairwise alignment scores
4. Perform hierarchical clustering and bootstrap significance testing with `pvclust` (Suzuki R and Shimodaira H. Bioinformatics 2006).
5. Extract the sequences and associated guide trees for significant clusters
6. Render a consensus secondary structure motif using the multiple structure alignment tool mlocarna

3.6.1 On consensus hierarchical clustering

We are investigating the practicality and efficiency of a consensus hierarchical clustering approach, where the (dis)similarity matrices of different pairwise structure alignment algorithms are concurrently employed for cluster analysis. **This is cutting-edge stuff and Luis will report back soon.**

3.6.2 On multiple structure alignment and 2D motif rendering

Generating a multiple structural alignment at the end of the pipeline is an important but tricky step. Right now, we are using mlocarna for this, which to my knowledge is the only tool that can produce such output without too much fuss. However, there is a substantial concern that arises from its use: `DotAligner`and mlocarna use fundamentally different alignment algorithms. This caveat is somewhat resolved by enforcing mlocarna to use the guide tree produced from the all-vs-all (dis)similarity matrix from `DotAligner`. Mlocarna will still align the sequences based on their consensus structure, therefore some additional benchmarking may be required. N.B., we can dictate which pairwise aligner (or probabilistic aligner) to be used by mlocarna its execution parameters, although mlocarna may ignore this when a guide-tree is provided—thus employing locarna to produce intermediate alignments even when only 2 sequences are involved. **CHECK THIS WITH SEBASTIAN WILL**

Some more specific points to consider:

- –threads=X seemingly doesn't affect mlocarna performance. Is this only implemented for pairwise comparisons? Generating the intermediary alignments uses one CPU. Perhaps –cpu=X will work?
- RNAPLfold is used for longer input sequences, right? Is this because it overcomes sequence length discrepancies? Should we enfore a size limit on the input sequences (either trim or extend the input to XX nucleotides divergence)?
- Try option `--pw-aligner path/to/DotAligner` and see if it will give more reliable consensus structure
- Test whether the speed limitation of iterative refinement (`--iterations=XX`) will be compensated by better quality alignments

- Will this cause (m)locarna to use the entire dot plots for the alignment? Test the effect of the following parameters `--probabilistic --consistency-transformation --it-reliable-structure=XX`.

4 DISCUSSION

The application of `DotAligner` is a fastly calculated similarity score between two probability matrices to enable their subsequent clustering. Similar to `CARNA`, which also does not garanty the optimal solution, `DotAligner` is not deterministic.

We plan to integrate the proposed method in a pipeline that screens regions of interest for structured RNA domains in a collection of RNA molecules. The so far presented approach finds only global alignments. This strategy is applicable for input sequences of similar lengths or if one sequence is considerably shorter (due to the usage of free end gaps). However, local alignment is favorable if both input sequences are long. Despite of the partition function version of the local alignment problem is available, its application dramatically increases the search space and, thus, the running time. As alternative, a possible screening pipeline may comprise window based thermodynamic folding, *e.g.* by `RNAplfold` (**?**), and filter regions of high intra-molecular binding probabilities in a pre-processing step, *e.g.* by using `RNAlocal` (**?**), followed by the presented alignment tool `DotAligner`. SHOULD WE INCLUDE THIS ANALYSIS IN THIS PAPER AND IF YES WHERE? The pre-selection of local structural potential may improve the boundaries of common structured RNA domain.

`DotAligner` can also be extended for multiple alignments, similar to the strategy implemented in `pmmulti` (**?**), and the generation of phylogenetic trees. This may replace or support the hierarchical clustering approach used here. In addition, both may serve as input for RNA secondary structure predictors, such as `PETfold` (**?**) unifying thermodynamic and evolutionary information.

ACKNOWLEDGEMENTS

I thank the Carlsberg foundation for my travel grant. Skål!
MAS is funded in part by a Cancer Council NSW project grant and