

Sequence analysis

Multiple structural alignment and clustering of RNA sequences

Elfar Torarinsson^{1,2}, Jakob H. Havgaard¹ and Jan Gorodkin^{1,*}¹Division of Genetics and Bioinformatics, IBHV and Center for Bioinformatics and ²Department of Natural Sciences, Faculty of Life Sciences, University of Copenhagen, 1870 Frederiksberg C, Denmark

Received on November 30, 2006; revised on January 16, 2007; accepted on February 6, 2007

Advance Access publication February 26, 2007

Associate Editor: Dmitrij Frishman

ABSTRACT

Motivation: An apparent paradox in computational RNA structure prediction is that many methods, in advance, require a multiple alignment of a set of related sequences, when searching for a common structure between them. However, such a multiple alignment is hard to obtain even for few sequences with low sequence similarity without simultaneously folding and aligning them. Furthermore, it is of interest to conduct a multiple alignment of RNA sequence candidates found from searching as few as two genomic sequences.

Results: Here, based on the PMcomp program, we present a global multiple alignment program, FOLDALIGNM, which performs especially well on few sequences with low sequence similarity, and is comparable in performance with state of the art programs in general. In addition, it can cluster sequences based on sequence and structure similarity and output a multiple alignment for each cluster. Furthermore, preliminary results with local datasets indicate that the program is useful for post processing FOLDALIGN pairwise scans.

Availability: The program FOLDALIGNM is implemented in JAVA and is, along with some accompanying PERL scripts, available at <http://foldalign.ku.dk/>

Contact: gorodkin@genome.ku.dk

1 INTRODUCTION

Traditionally, most RNA molecules have been regarded as merely an intermediate on the DNA → mRNA → protein pathway. The main exceptions were transfer and ribosomal RNAs, both involved in the process of translation. However, since the late 1990s the view on RNA has changed dramatically with the discovery of a steadily increasing number of untranslated RNAs. These untranslated RNAs are of great importance in all three domains of life (Eddy, 2001; Hüttenhoffer *et al.*, 2002; Mattick, 2001).

Obviously, the structure is of importance to understand the function of the RNA and whereas single sequence folding methods (Ding, 2004; Hofacker, 1994; Zucker, 2003) provide some information about the structure, it can most reliably be detected from a set of related sequences (if available) (Westhof and Michel, 1994; Westhof *et al.*, 1996). For example, 97–98% of all base pairs Gutell and co-workers had found by hand

(Gutell *et al.*, 2002) are indeed present in the 3D structure of the ribosome (Ban *et al.*, 2000). Whereas an expert in a specific RNA by hand can determine the base pairs through tedious study of compensating changes in a large set of sequences, the number of sequences might, in general, be few and they might also have low sequence similarity. Hence, with the increasing number of novel RNAs the need for automated structure prediction became clear and a number of methods were introduced. These include the first FOLDALIGN version (Gorodkin *et al.*, 1997), as well as methods such as Dynalign (Uzilov *et al.*, 2006), Pfold (Knudsen and Hein, 1999), RNAalifold (Hofacker *et al.*, 2002), COVE (Eddy and Durbin, 1994), Stemloc (Holmes, 2005) and RNacast (Reeder and Giegerich, 2005) to mention a few examples.

Until recently some of the most reliable methods to computationally find non-coding RNAs (ncRNAs) were still limited to simple sequence similarity searches with known ncRNAs or scanning with stochastic context-free grammars (SCFG) for the already known ncRNAs. This is implemented in programs such as COVE, its successor, INFERNAL (Eddy, 2002) and RSEARCH which allows for searching a sequence database for homologs of a given, structured, RNA molecule (Klein and Eddy, 2003). Speed improvements on INFERNAL have been implemented in RAVENNA, which utilizes hidden Markov models (HMMs) for filtering, speeding the scanning considerably (Weinberg and Ruzzo, 2004a,b). However, these methods cannot in general identify new families of ncRNAs. Two recent efficient approaches include EvoFold (Pedersen *et al.*, 2006) and RNAz (Washietl *et al.*, 2005) which are efficient enough to be applied to genome-wide scans and can detect new conserved structural RNAs, but only in pre-aligned sequences. Furthermore, EvoFold and RNAz are sensitive to alignment errors and can only perform a scan using predefined fixed window sizes. A more recent very promising program is CMfinder (Yao *et al.*, 2006) which unlike EvoFold and RNAz is independent of alignments and is local.

Recent implementation of the Sankoff algorithm (Sankoff, 1985), FOLDALIGN, provides an effective approach for searching two sequences for common local structure motifs (Havgaard *et al.*, 2005). FOLDALIGN uses two types of parameters to calculate scores, energy and sequence similarity. FOLDALIGN is particularly effective in low sequence similarity ranges and was recently used to scan human-mouse pairs, unalignable in primary sequence, for conserved structures, resulting in thousands of significant candidates (Torarinsson *et al.*, 2006).

*To whom correspondence should be addressed.

To further make it possible to collect data from pairwise scans of multiple organisms we here present an approach which can conduct clustering followed by global multiple structural alignments of the candidates found from the pairwise scans. The approach is based on the principles of PMcomp and PMmulti (Hofacker *et al.*, 2004), but it deviates in a number of ways described below. In particular it can use pair probability score matrices based on FOLDALIGN pairwise scans.

2 METHODS

Multiple alignments were in the early version of FOLDALIGN found through a greedy approach (Gorodkin *et al.*, 1997). However, a greedy approach with the current more extensive FOLDALIGN version is not suitable. The current pairwise version of FOLDALIGN employs a scoring scheme in which a joint score, composed of folding energy, substitution of entire base pairs and single nucleotides, is maximized (Havgaard *et al.*, 2005). This complicates implementation of a greedy approach in the same manner and is also expected to be time-consuming. In contrast the PMcomp and PMmulti algorithms promise good speed assuming an efficient implementation. Whereas PMmulti lacks the ability to conduct local multiple structural RNA alignments, it is here implemented as a possible layer on top of a pairwise FOLDALIGN scan. We do this by calculating the base-pair probability matrices using FOLDALIGN (see end of Section 2.1) and not McCaskill's algorithm (McCaskill, 1990), as implemented in PMcomp (Hofacker *et al.*, 2004). PMcomp is currently available as a Perl implementation and hence is relatively slow. Therefore, we re-implemented PMcomp in JAVA with several improvements. This re-implementation, which we named FOLDALIGNM, can align both McCaskill's and FOLDALIGN generated probability matrices. In addition, it can cluster similar candidates and perform alignments to a given structure and sequence.

2.1 PMcomp improvements

The PMcomp algorithm is in general implemented as described by Hofacker *et al.* (2004) but with several improvements. Briefly, PMcomp aligns two sequences A and B by aligning their base-pair probability matrices P^A and P^B , which are computed by the McCaskill's algorithm (McCaskill, 1990). Only base-pair probabilities above a given cutoff, which we note p^* , are considered. In general PMcomp looks for an alignment of the sequences A and B with N_{gap} insertions or deletions together with a consensus secondary structure Z such that

$$\sum_{(ij;kl) \in Z} \left[\Psi_{ij}^A + \Psi_{kl}^B + \tau(A_i, A_j; B_k, B_l) \right] + \gamma N_{\text{gap}} + \sum_{i \in A, k \in B \notin Z} \sigma(A_i, B_k) \rightarrow \max, \quad (1)$$

where Ψ_{ij}^A is the weight of pair (i, j) from sequence A , $\gamma < 0$ is the gap penalty and the scores represented by σ and τ describe the substitution of unpaired bases and base pairs, respectively (Hofacker *et al.*, 2004).

Here, we have added the FOLDALIGN δ parameter, which is the maximum allowed length difference in an alignment between two subsequences. In general, this gives complexities of $O(L^2\delta^2)$ in time and memory, where L is the length of the sequence, reducing the PMcomp complexities of $O(L^4)$. Furthermore, we also include the following memory and time improvements presented in the updated version of FOLDALIGN for pairwise (local) alignments (Havgaard *et al.*, submitted for publication). The dynamic programming matrix is filled in two steps as described in details by Havgaard *et al.* (submitted for publication). Briefly, in the first step only the multibranch points are located, which are the positions where we join two substructures. The multibranch

points are used to divide the structure into unbranched segments. This is done by using a short-term matrix (S in Equation (2)) and a long-term matrix (S' in Equation (2)), where the short-term matrix only has to remember the last position we came from so we can do this using $O(L\delta^2)$ memory. The long-term matrix only contains the significant base pairs and the branch points. Because each base can take part in at most $1/p^*$ many base pairs, there are only $O(L)$ significant base pairs in each sequence, thus the long-term matrix requires only $O(L\delta)$ memory (and not $O(L^2)$ because we only allow δ length difference between any subsequences). In the second step, we treat all segments, as given by our multibranch point list, independently using only $O(M^2\delta^2)$ memory and CPU time for each segment, where M is the length of the segment. In addition, also following Havgaard *et al.* (submitted for publication), we prune away low scoring cells saving CPU time (~ 20 – 60% depending on the sequences). Pruning is simply done by excluding sub-alignments from the matrix when the score is smaller than a given threshold.

Let $S_{i,j;k,l}$ be the score of the best match of the subsequences $A[i \dots j]$ and $B[k \dots l]$. With this definition, we can obtain the following dynamic programming recursions

$$S_{i,j;k,l} = \max \begin{cases} S_{i+1,j;k,l} + \gamma, \\ S_{i,j;k+1,l} + \gamma, \\ S_{i,j-1;k,l} + \gamma, \\ S_{i,j;k,l-1} + \gamma, \\ S_{i+1,j;k+1,l} + \sigma(A_i, B_k), \\ S_{i,j-1;k,l-1} + \sigma(A_j, B_l), \\ S_{i+1,j-1;k+1,l-1} + \Psi_{ij}^A + \Psi_{kl}^B + \tau(A_i, A_j; B_k, B_l) \\ \max_{\substack{i < m < j \\ k < n < l}} \{ S_{i,m;k,n} + S'_{m+1,j;n+1,l} \} \end{cases} \quad (2)$$

with the initialization $S_{L_A, L_A; L_B, L_B} = 0$ where L_A and L_B are the lengths of sequence A and B , respectively. The first four terms account for gaps in one of the two sequences. The next two terms describe the extension of both subsequences with an unpaired position. The next term describes a base-pair match in both subsequences and, finally, the last term describes the joining of two alignments to get a multibranch structure. This is only performed when the right part has base pairs between $(m+1$ and $j)$ and $(n+1$ and $l)$, kept in S' , and the left part of the structure is limited to the cases where i is base paired and k is base paired (Havgaard *et al.*, 2005). The last term is not needed in step two where we only align unbranched segments; because we know that they should be unbranched from step one where we locate all the multibranch points. This recursion differs from that of PMcomp in two ways, first of all we can move in both directions on both sequences and secondly PMcomp joins together the last two terms because they calculate everything in one step and not in two as we do. The maximum allowed length difference, δ , gives the following restrictions

$$\begin{aligned} |i - k| &\leq 2\delta, \\ |(j - i) - (l - k)| &\leq \delta \end{aligned} \quad (3)$$

PMmulti constructs a progressive multiple alignment using PMcomp (Hofacker *et al.*, 2004). PMmulti use the weighted pair-group clustering method to produce a guide tree with similarity scores. For a multiple alignment of N RNAs, the similarity scores are computed using $N(N-1)/2$ cheap string-like alignments which only cost ($O(L^2)$) in time (Bonhoeffer *et al.*, 1993). The more expensive PMcomp algorithm is restricted to the $N-1$ progressive alignments along the guide tree, i.e. first it aligns the highest scoring pair, calculates a consensus base-pair probability matrix for these and aligns the next sequence to this consensus matrix, then makes a new consensus matrix for these three sequences and so on. Hence, the program always compares two probability matrices. FOLDALIGNM also contains the same progressive multiple-alignment approach except when using FOLDALIGN base probability matrices, where we simply use the pairwise FOLDALIGN scores which are already available to produce the guide tree.

In PMcomp, the consensus base-pair probability matrix is defined, for a pair of columns pq in the alignment of A and B , as

$$P_{p,q}^{AB} = \begin{cases} \sqrt{P_{i_p j_q}^A P_{k_p l_q}^B} & \text{for matches} \\ 0 & \text{for gaps} \end{cases} \quad (4)$$

where i_p and j_q are the positions corresponding to p and q in sub-alignment A respectively. k_p and l_q are defined analogously for sequence B . This means that whenever one sub-alignment contains a gap or has a very low base-pair probability, then the structural information from the other sub-alignment is lost. This tends to remove most base pairs when aligning many sequences. In order to avoid this problem in PMcomp we add a step to redefine the final alignment. In this step we compute the most frequently occurring base pairs in all the progressive alignments, make a new base pair probability matrix with these base pairs and assign the probability of 1.0 to each base pair, and align each sequence once again to this consensus matrix. More specifically we count, for every column in the alignment, the number of sequences in the alignment that base pair at this column. If more than Forty percent of the sequences base pair at these columns, they are added to a new consensus probability matrix with probability 1. Here we are not defining a strict consensus structure but a looser list of frequently occurring base pairs that we align again to each sequence. Forty percent was chosen after testing several different parameters but it can also be given as an argument to the program. This tends to improve on the alignment, especially when dealing with many sequences.

To generate FOLDALIGN base-pair probability matrices we compute all $N(N-1)/2$ pairwise alignments. For each sequence the frequency of each base pair is simply the observed frequency, i.e. when aligning 11 sequences and a given base pair in sequence A was predicted in five of all 10 possible pairwise alignments, the probability of this base pair in sequence A would then be 50%.

2.2 Clustering

FOLDALIGNM can also cluster the candidates that are similar in structure and sequence based on all-against-all pairwise FOLDALIGN scores. As an initial step we used a simple clustering procedure described in the following, somewhat similar to simple single-linkage clustering. In the clustering, we start by sorting the pairwise FOLDALIGN alignments based on their scores. Then we add the sequences in the highest scoring alignment to the first cluster and go through the list of descending scores and if the score is above a given cutoff (an option to the program), do the following:

- (1) If neither sequence exists in a cluster, make a new cluster and add them to it.
- (2) If one of the sequences exists in a cluster add the other one to that clustering.
- (3) If both sequences are already in the same cluster do nothing.
- (4) If the sequences are in different clusters, note the score of the alignment and add this score to the score between these two clusters.

Go through the clusters and check if there exists a score between clusters (step 4), i.e. check if there are any high scoring pairs whose sequences belong to different clusters. If so, check if the average score between the sequences in these clusters is higher than a given cutoff, which depends directly on the first cutoff mentioned above. If so merge the clusters otherwise do not merge the clusters. The choice of cutoff is crucial for the clustering procedure. By choosing a high cutoff, one gets small well-defined clusters with higher specificity whereas a low cutoff would result in large clusters with higher sensitivity. This simple clustering approach worked very well (see Results section) so we did not implement a more advanced clustering scheme.

2.3 Performance measures

The accuracy of all predictions is computed at the base-pair level relative to the Rfam base-pair assignments, ignoring non-canonical base pairs. Let P_t be the correctly predicted base pairs (assuming that the Rfam assignment is correct), P_f the falsely predicted base pairs and N_f the true base pairs that are not predicted. The sensitivity is defined as $P_t/(P_t + N_f)$, positive predictive value (PPV) as $P_t/(P_t + P_f)$ (also referred to as specificity in some articles) and overall prediction accuracy as their geometric mean. For good sensitivity and PPV, the latter metric approximates Matthews correlation coefficient (MCC) (Gorodkin et al., 2001). We calculate the prediction accuracy for each sequence and report the average for the whole family as the average of every sequence belonging to that family. As we want to compare our results to that of CMfinder (Yao et al., 2006), which uses this performance measure, we also use this measure.

For evaluating the cluster prediction, we calculate MCC (Matthews, 1975).

$$MCC = \frac{P_t N_t - P_f N_f}{\sqrt{(P_t + P_f)(P_t + N_f)(N_t + P_f)(N_t + N_f)}} \quad (5)$$

2.4 Data

Five different datasets were used to test FOLDALIGNM. For a general evaluation of performance on global alignments, we used the CMfinder datasets from Yao et al. (2006). These consist of 19 different families from the Rfam seed alignments (Griffiths-Jones et al., 2003) and contain 9–71 sequences per family with average sequence similarities ranging from 43 to 81% and average lengths from 30 to 216. We excluded the family, Enterotoxin OriR, because all the sequences miss half of the motif in the evaluation made by Yao et al. (2006). Cobalamin was also excluded because it has 71 sequences with extreme differences in lengths, which due to memory limitations in our method is not feasible. We refer to this dataset as the *CMfinder dataset*.

We also made a more comprehensive dataset for three different families from the Rfam seed alignments (Griffiths-Jones et al., 2003). This dataset includes three families where each family includes 20 subsets with 2–10 sequences and 20–39, 40–59, 60–79 and 80–99% sequence similarity, i.e. there are 20 subsets with two sequences that each has 20–40% sequence similarity to each other, etc. This gives 720 subsets for each family. We used the families tRNA, bacterial SRP and eukaryotic and archaeal SRP, because they have well-defined structures (Griffiths-Jones et al., 2003; Rosenblad et al., 2003) and are the only families where it is possible to make such a comprehensive dataset from the Rfam seed sequences. In fact we could only make a complete dataset for tRNA. The other families were complete only for two to five sequences and it was in some cases not possible to generate 20 different sets with six to ten sequences for some specific sequence similarities. Still the dataset was adequate in providing good understanding about the different strengths of the algorithms. We refer to this set as the *global comprehensive dataset*.

To test the clustering function the same 19 families as Yao et al. used were reused. We constructed 10 different datasets where we randomly chose between five and nine sequences, from each of the 19 families, put them all together and ran our clustering procedure on this dataset. This is our *global clustering set*.

For the local clustering we prepared two datasets, a local pairwise set, to test clustering on top of FOLDALIGN local pairwise scans, and a local multiple set, to test multiple local clustering using all-against-all FOLDALIGN. The *local pairwise* dataset contained 14 tRNA, 13 THI, two U1 and two Purine sequence pairs from the FOLDALIGN dataset (Havgaard et al., 2005). Each motif lies within its 500 long genomic context and each pair has less than 40% sequence similarity and is energetically indistinguishable from its genomic context. For the *local multiple dataset* we used, from the 500 long sequences, eight tRNAs,

seven UIs and five Purines, in addition we shuffled the nucleotides for two randomly chosen sequences from each of these families and added those to our set. The shuffling preserves the dinucleotide frequencies (Altschul and Erickson, 1985; Workman and Krogh, 1999).

3 RESULTS

The performance of FOLDALIGNM was evaluated on the five different datasets described in Materials and methods section. These are (i) the CMfinder dataset, (ii) our global comprehensive dataset, (iii) global clustering dataset, (iv) a local pairwise set and (v) a local multiple set.

3.1 Global alignment

For our general test we used a global dataset, generated and tested by Yao *et al.* (2006) on their own algorithm, CMfinder and several others. Using the fast version of our program, which uses McCaskill's base-pairing matrices like PMcomp does, we compared our performance to that of CMfinder and their runs with Pfold (Knudsen and Hein, 1999), and RNAalifold (Hofacker *et al.*, 2002), on these 17 families. In addition we also ran STRAL (Dalli *et al.*, 2006) with RNAalifold, Stemloc (Holmes, 2005) and RNACast (Reeder and Giegerich, 2005) using the default parameters. STRAL and RNAalifold performed as expected slightly better than using Clustalw and RNAalifold with average MCC of 0.65.

FOLDALIGNM and CMfinder showed comparable performances, which again perform significantly better than the other approaches (see Table 1). FOLDALIGNM outputs two different alignments, one similar to that of PMmulti and one where we used our redefining method to redefine the alignments (see Material and methods sections). We calculated the performance of FOLDALIGNM, using McCaskill's base-pair probability matrices, for both alignments. The average for

our redefined alignments was 0.79 whereas the average was 0.73 for the original alignments, so we always used the redefined one. For this dataset FOLDALIGNM uses <2 min to output the results for most of these families, although the longer families and those with large differences in sequence length can take a few hours to run.

Programs like RNACast and Stemloc are seemingly not performing too well, but if one only considers the families where they actually produce a prediction, they perform quite well with an average MCC of 0.80 and 0.76, respectively. RNACast is very fast so it can be nice as a starting point since it will usually perform well if it finds a consensus shape for the different input sequences. Playing around with different parameters would probably result in better performances, especially for those two algorithms.

In order to gain better insight into the different properties of FOLDALIGNM and CMfinder, we constructed and ran tests on a global comprehensive dataset. This dataset contains subsets with different number of sequences and different sequence similarity ranges.

FOLDALIGN is particularly effective on pairs with low sequence similarity. This is also the case for FOLDALIGNM which outperforms CMfinder in the 20–59% sequence identity range (Table 2 and Fig. 1). The performance is similar in the 60–79% sequence identity range and CMfinder performs better in the 80–99% similarity range. FOLDALIGNM has a tendency to perform better on few sequences which is not surprising in that CMfinder is designed with more sequences in mind where it is very effective.

3.2 Global clustering

When using all-against-all FOLDALIGN pairwise alignments to make the base-pair probability matrix, one also gets pairwise

Table 1. Performance on 17 Rfam families from the CMfinder dataset

Family	Number of sequences	FOLD ALIGNM	CM finder	Clustal/Pfold	Clustal/Alifold	Stem loc	RNA cast
Entero CRE	56	0.75	0.77	0.95	0.71	0.40	0.82
Histone 3	63	1	1	1	1	1	1
IRE	29	0.83	0.92	0.67	0.64	0.88	0.71
Intron gpII	75	0.73	0.71	0.78	0.76	0	0
Lysine	48	0.60	0.77	0.59	0.22	0	0.73
Purine	29	0.89	0.90	0.76	0	0.76	0
RFN	47	0.73	0.73	0.72	0.77	0	0
Rhino CRE	12	0.82	0.96	0.66	0.77	0.80	0.72
SECIS	63	0.73	0.68	0	0	0	0.65
S box	64	0.73	0.81	0.77	0.72	0	0
Tymo tRNA-like	22	0.74	0.77	0.62	0.73	0.78	0
ctRNA pGAl	17	0.94	0.89	0.86	0.92	0.83	0.95
glmS	14	0.72	0.74	0.62	0.49	0	0
let-7	9	0.83	0.79	0.84	0.80	0.75	0.82
lin-4	9	0.78	0.76	0.72	0.73	0.82	0.78
mir-10	11	0.81	0.85	0.75	0.85	0.79	0.78
s2m	23	0.79	0.68	1	0.64	0.55	0
AVG	35	0.79	0.81	0.72	0.63	0.49	0.47

The numbers are approximated Matthew's correlation coefficients as described in Methods section. FOLDALIGNM is run with McCaskill-generated base-pair probability matrices.

Table 2. Comprehensive four-family dataset for different pairwise sequence similarity ranges

	20–39%	40–59%	60–79%	80–99%
tRNA				
CMfinder	0.40	0.85	0.91	0.79
FOLDALIGNM–McCaskill	0.77	0.83	0.84	0.71
FOLDALIGNM–FOLDALIGN	0.92	0.96	0.95	0.73
SRP-euk-arch				
CMfinder	0.39	0.62	0.74	0.73
FOLDALIGNM–McCaskill	0.56	0.70	0.71	0.68
FOLDALIGNM–FOLDALIGN	0.49	0.74	0.65	0.53
SRP-bacterial				
CMfinder	0.54	0.64	0.79	0.83
FOLDALIGNM–McCaskill	0.73	0.75	0.79	0.79
FOLDALIGNM–FOLDALIGN	0.75	0.79	0.81	0.81

The numbers are approximated Matthew’s correlation coefficients as described in Methods section. FOLDALIGNM–McCaskill version uses McCaskill’s base-pair probability matrix whereas FOLDALIGNM–FOLDALIGN uses a FOLDALIGN generated matrix. These numbers represent the averages for the datasets containing 2–10 sequences, where each sequence in every set have the noted sequence similarity to every other sequence in its set.

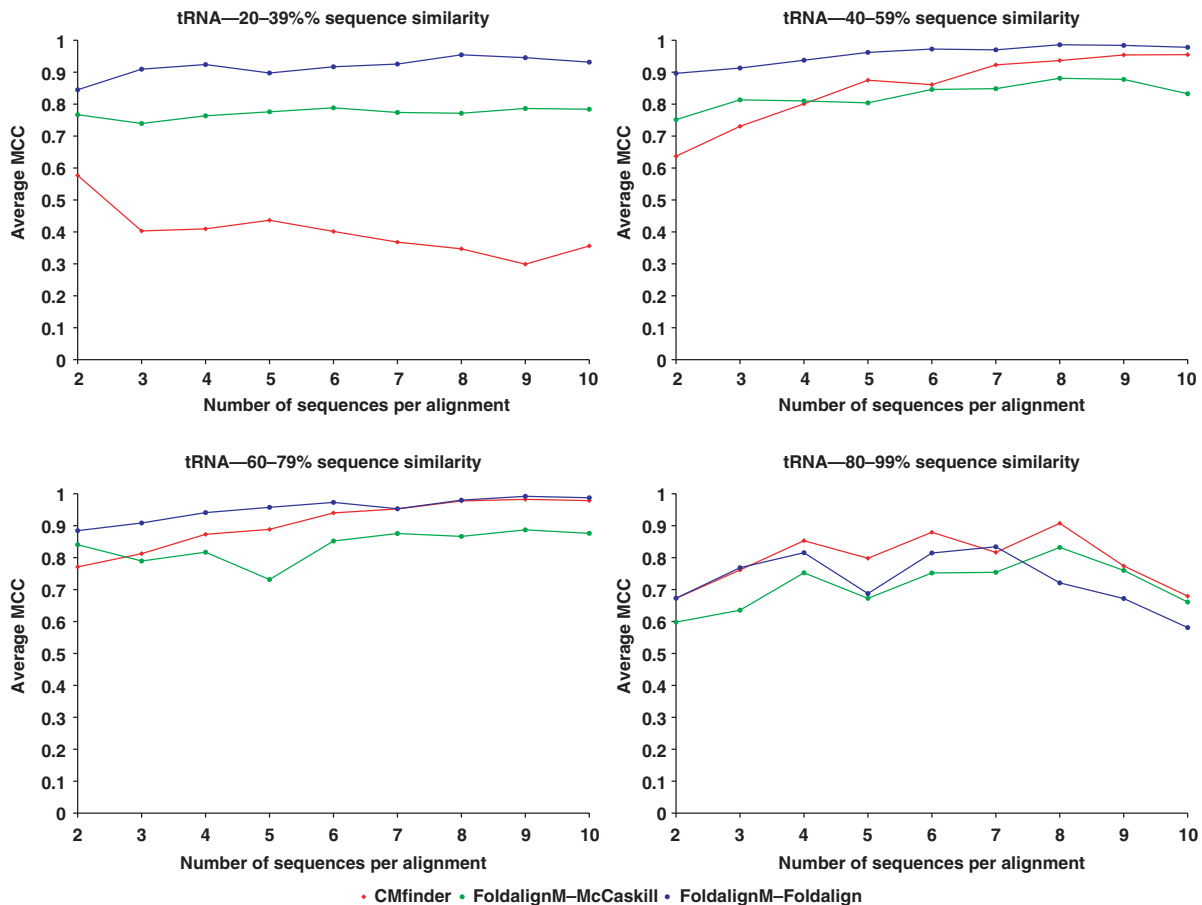


Fig. 1. The average MCC performance of CMfinder, FOLDALIGNM–McCaskill and FOLDALIGNM–FOLDALIGN on different sequence similarity sets of tRNA.

scores between all the sequences. These scores can be used to cluster similar sequences together and then we generate a multiple alignment for each cluster. To test this, we again used the 19 family dataset from Yao *et al.* (2005). We constructed

10 different datasets where we randomly chose between 5 and 9 sequences from each of the 19 families and ran our clustering procedure on this dataset. We calculated Matthew’s correlation coefficient by considering every possible pair

of sequences, counting true and false, positives and negatives as follows: a true positive is a pair of sequences in the same family and same cluster, a false positive is a pair of sequences not in the same family but in the same cluster, a true negative is a pair of sequences not in the same family and not in the same cluster and finally a false negative is a pair of sequences in the same family but not in the same cluster. The average MCC for these 10 datasets is 0.78. The following results were averaged for the 10 datasets. From every dataset containing 135 sequences, 24 different clusters were predicted. Nineteen sequences do not belong to any cluster. Seven clusters were perfectly predicted, containing all the sequences in the respective family. Five clusters were correct but were missing one or more sequences from the family. Six families were split into two (generally) or more clusters, and one cluster contained sequences from two or more families. Usually a single sequence from one miRNA family joined another miRNA family cluster.

To test the robustness of the clustering we added the shuffled sequences, maintaining dinucleotide frequencies (Altschul and Erickson 1985; Workman and Krogh 1999), of two families to one of our datasets and ran it again. The results for this dataset did not change and all the shuffled sequences were predicted to be outside the clusters.

3.3 Semi-local clustering

Since FOLDALIGN performs local pairwise alignments, we can further conduct semi-local clustering. Even though it is beyond the scope of this article to do local clustering for multiple sequences, we could still perform some putative studies on the possibilities for doing local clustering and alignments using FOLDALIGN input. For this we used 14 tRNA, 13 THI, two U1 and two Purine sequence pairs. We extracted the best hit from every pairwise scan, added the 30 flanking genomic nucleotides to both ends, and ran local all-against-all pairwise FOLDALIGN alignments. Clustering and aligning the results from this resulted in three good clusters, one with all four U1s, one with all four Purines and finally one with 16 of the 28 tRNAs. Both the localization and the structure prediction were very good with an average MCC of 0.81 for the three clusters. FOLDALIGN does not perform very well on THI and therefore these did not cluster well together.

These results indicate that we can cluster sequences with well-defined structures from a pairwise scan. We also made a dataset to test basic local multiple alignment and clustering, using a dataset with eight tRNAs, seven U1s, five Purines and six shuffled sequences. Running all-against-all local pairwise FOLDALIGN followed by clustering and multiple alignment on this dataset, resulted in three perfect clusters with good localization and good structure prediction, averaging a MCC of 0.84 for the three clusters.

Despite significant improvements to the latest version of FOLDALIGN (Havgaard *et al.*, submitted for publication) it is still time-consuming to run larger datasets pairwise all-against-all with FOLDALIGN. This makes the use of FOLDALIGN generated base-pair probability matrices more attractive when you already have the FOLDALIGN output available.

4 CONCLUSION

We have, based on the PMcomp program, presented a global multiple alignment algorithm that considers both sequence and structure. It has comparable performance to other recent programs and is especially effective on sequences with low sequence similarities. In addition to this our program, FOLDALIGNM, is also capable of clustering similar sequences together and providing a multiple alignment for each cluster. Further, the current version should already be useful in clustering results from a scan such as that performed by Torarinsson *et al.* (2006). Perspectives for FOLDALIGNM include implementing a more advanced clustering approach like, for example, as described by Ding *et al.* (2005), using hierarchical clustering (Johnson, 1967) and then estimating the number of clusters with the CH index (Calinski and Harabasz, 1974) which was assessed to be the best in a comprehensive study (Milligan and Cooper, 1985). Although our simple clustering works well, it could be interesting to compare to state-of-the-art clustering algorithms. Furthermore we would like to implement a local version. Our preliminary results, where we localize the hits using all-against-all FOLDALIGN, are very promising, and the performance is likely to increase with a local version of FOLDALIGNM. Also when clustering large datasets it is not necessary to run all-against-all FOLDALIGN as we do now, since many sequences could be pre-filtered based on features such as length and sequence and structure similarity, before running the more expensive all-against-all FOLDALIGN. Our algorithm is sensitive to very large differences between the lengths of the sequences and is quite memory intensive on long sequences. When this work was being finished, we realized that another re-implementation of PMComp, LocaRNA (Will *et al.*, submitted for publication), deals very elegantly with this problem and uses only $O(L^2)$ memory.

ACKNOWLEDGEMENTS

This work was supported by Danish Research Council for production and technology and the Danish Center for Scientific Computation. We thank Ivo L. Hofacker, Peter F. Stadler, Paul P. Gardner and Stinus Lindgreen for comments on this manuscript.

Conflict of Interest: none declared.

REFERENCES

- Altschul,S.F. and Erikson,B.W. (1985) Significance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleotide and codon usage. *Mol. Biol. Evol.*, **2**, 526–538.
- Ban,N. *et al.* (2000) The complete atomic structure of the large ribosomal subunit at 2.4 Å resolution. *Science*, **289**, 905–920.
- Bonhoeffer,S. *et al.* (1993) RNA multistructure landscapes. *Eur. Biophys. J.*, **22**, 13–24.
- Calinski,R.B. and Harabasz,J. (1974) A dendrite method for cluster analysis. *Comm. Stat.*, **3**, 1–27.
- Dalli,D. *et al.* (2006) . StrAl: progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time. *Bioinformatics*, **22**, 1593–1599.
- Ding,Y. *et al.* (2004) Sfold web server for statistical folding and rational design of nucleic acids. *Nucleic Acids Res.*, **32**, W135–W141.

- Ding, Y. et al. (2005) RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. *RNA*, **11**, 1157–1166.
- Eddy, S.R. and Durbin, R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res.*, **22**, 2079–2088.
- Eddy, S.R. (2001) Non-coding RNA genes and the modern RNA world. *Nat. Rev. Genet.*, **2**, 919–929.
- Eddy, S.R. (2002) A memory efficient dynamic programming algorithm for optimal structural alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics*, **3**, 18.
- Gorodkin, J. et al. (1997) Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, **25**, 3724–3732.
- Gorodkin, J. et al. (2001) Discovering common stem-loop motifs in unaligned RNA sequences. *Nucleic Acids Res.*, **29**, 2135–2144.
- Griffiths-Jones, S. et al. (2003) Rfam: an RNA family database. *Nucleic Acids Res.*, **31**, 439–441.
- Gutell, R. et al. (2002) The accuracy of ribosomal RNA comparative structure models. *Curr. Opin. Struct. Biol.*, **12**, 301–310.
- Havgaard, J.H. et al. (2005) Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. *Bioinformatics*, **21**, 1815–1824.
- Hofacker, I.L. (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188.
- Hofacker, I.L. et al. (2002) Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, **319**, 1059–1066.
- Hofacker, I.L. et al. (2004) Alignment of RNA base pairing probability matrices. *Bioinformatics*, **14**, 2222–2227.
- Holmes, I. (2005) Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*, **6**, 73.
- Huttenhoffer, A. et al. (2002) RNomics: identification and function of small, non-messenger RNAs. *Curr. Opin. Chem. Biol.*, **6**, 835–843.
- Johnson, S.C. (1967) Hierarchical clustering schemes. *Psychometrika*, **2**, 241–254.
- Klein, R.J. and Eddy, S.R. (2003) RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics*, **4**, 44.
- Knudsen, B. and Hein, J. (1999) RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, **15**, 446–454.
- Matthews, B.W. (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochem. Biophys. Acta*, **405**, 442–451.
- Mattick, J.S. (2001) Non-coding RNAs: the architects of eukaryotic complexity. *EMBO Rep.*, **2**, 986–991.
- McCaskill, J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.
- Milligan, G.W. and Cooper, M.C. (1985) An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, **50**, 159–179.
- Pedersen, J.S. et al. (2006) Identification and classification of conserved RNA secondary structures in the human genome. *PLoS Comput. Biol.*, **2**, e33.
- Reeder, J. and Giegerich, R. (2005) Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction. *Bioinformatics*, **21**, 3516–3523.
- Rosenblad, M.A. et al. (2003) SRPDB: signal recognition particle database. *Nucleic Acids Res.*, **31**, 363–364.
- Sankoff, D. (1985) Simultaneous solution of the RNA folding, alignment and proteosequence problems. *SIAM. J. Appl. Math.*, **45**, 810–825.
- Torarinsson, E. et al. (2006) Thousand of corresponding human and mouse genomic regions unalignable in primary sequence contain common RNA structure. *Genome Res.*, **16**, 885–889.
- Uzilov, A.V. et al. (2006) Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change. *BMC Bioinformatics*, **7**, 173.
- Washietl, S. et al. (2005) Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. USA*, **102**, 2454–2459.
- Weinberg, Z. and Ruzzo, W.L. (2004a) Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. *Bioinformatics*, **20** (Suppl. 1), 342.
- Weinberg, Z. and Ruzzo, W.L. (2004b) Faster genome annotation of non-coding rna families without loss of accuracy. In *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology (RECOMB)*. ACM Press, pp. 243–251.
- Westhof, E. and Michel, F. (1994) In Nagai, K. and Mattaj, I.W. (eds.) *RNA-Protein Interactions*. Oxford University Press, Oxford, UK, pp. 26–51.
- Westhof, E. et al. (1996) In Bihop, M. J. and Rawlings, C. J. (eds.) *DNA-Protein Sequence Analysis*. Oxford University Press, Oxford, UK, pp. 255–278.
- Workman, C. and Krogh, A. (1999) No evidence that mRNA have lower folding free energies than random sequences with the same dinucleotide distribution. *Nucleic Acids Res.*, **27**, 4816–4822.
- Yao, Z. et al. (2006) Cmfnder—a covariance model based RNA motif finding algorithm. *Bioinformatics*, **22**, 445–452.
- Zuker, M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.*, **31**, 3406–3415.