

# Constraint-based strategy for pairwise RNA secondary structure prediction

Olivier Perriquet, Pedro Barahona

CENTRIA - Centre for Artificial Intelligence - Dep. de Informática,  
FCT/UNL - Quinta da Torre 2829-516 CAPARICA - Portugal  
Tel. (+351) 21 294 8536 Fax. (+351) 21 294 8541  
olivier@perriquet.net, pb@di.fct.unl.pt

**Abstract.** RNA secondary structure prediction depends on context. When only a few (sometimes putative) RNA homologs are available, one of the most famous approach is based on a set of recursions proposed by Sankoff in 1985. Although this modus operandi insures an algorithmically optimal result, the main drawback lies in its prohibitive time and space complexities. A series of heuristics were developed to face that difficulty and turn the recursions usable. In front of the inescapable intricacy of the question when handling the full thermodynamic model, we come back in the present paper to a biologically simplified model that helps focusing on the algorithmic issues we want to overcome. We expose our ongoing developments by using the constraints framework which we believe is a powerful paradigm for heuristic design. We give evidence that the main heuristics proposed by others (structural and alignment banding, multi-loop restriction) can be refined in order to produce a substantial gain both in time computation and space requirements. A beta implementation of our approach, that we named ARNICA, exemplify that gain on a sample set that remains unaffordable to other methods. The sources and sample tests of ARNICA are available at <http://centria.di.fct.unl.pt/~op/arnica.tar.gz>

## 1 Introduction

The interest for RNA secondary structure prediction in the last decades may be driven by the combination of two reasons, a biological one and a theoretical one. From the biological viewpoint, RNA secondary structure prediction is a challenging problem to help bridging the gap between the different levels of description of the structure [CMR08,SYKB07]. The discovery of new families of non-coding RNA (ncRNA) demands adapted tools to predict or at least provide information about their structure, and the programs that compute secondary structure naturally organize themselves along several noticeable directions that implicitly depend on their context of use. When the family of sequences under consideration is large enough and the sequences not too divergent, a good multiple alignment is obtained by sequence alignment methods ([OH98] assess the reliability of such methods for rRNA) or by semi-automated methods. The structure

prediction programs based on covariation analysis [ED94,KH03] can then handle the alignment and they have proven to be very accurate in guessing the structure in that context. With a small family of poorly conserved RNA sequences, a good starting alignment can hardly be constructed, resulting usually in the inapplicability of the methods based on pre-alignment. In that second context it makes sense to search for the alignment and the structure at the same time. Sankoff [San85] pioneered the field by exhibiting a set of recursions to optimally compute the best structural alignment of two RNA sequences when the structure is not known a priori. These recursions can be straightforwardly extended to  $N$  sequences and can also handle the energy parameters traditionally used for energy minimization [MSZT99]. Although the time and space complexities for two sequences remains polynomial (resp.  $O(n^6)$  and  $O(n^4)$  if their lengths are the same order of magnitude  $n$ ), the algorithm is not applicable without heuristic adaptations. From this more abstract point of view, RNA structural alignment becomes a challenging problem in terms of algorithmic issues. Diverse heuristic ideas were applied in able to turn the recursions usable. The first implementation of the Sankoff recursions was DYNALIGN [MT02,HSM07] that reduced the complexity to  $O(M^2n^2)$  in space and  $O(M^3n^3)$  in time, where  $M$  is a (tunable) hard constant which bounds the shift allowed between the two sequences. [HLSG05,HTG07] investigated further (FOLDALIGN) by combining other different restrictions, namely:

- **alignment banding**: like in DYNALIGN the maximal shift between the sequences is bounded by a constant  $\delta$
- **structural banding**: the default behavior for the program is to perform a local alignment and the maximal size for a common motif is bounded by  $\lambda$
- **multi-loop restriction**: the structure bifurcativity is limited in multi-loops

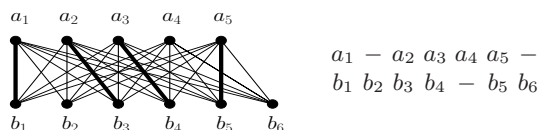
The resulting space and time complexities in FOLDALIGN become  $O(n^2\lambda\delta)$  and  $O(n^2\lambda^2\delta^2)$ . The Vienna RNA Package also proposes an alternative attempt PM-Comp [HBS04] which is implemented as part of the RNAfold program and makes use of the McCaskill algorithm to compute the probabilities of base pairings for a single sequence [McC90]. The authors of FOLDALIGN then revisited the idea by integrating their banding heuristic to PMComp [THG07].

Although these heuristics allow the application of the algorithm of Sankoff on natural RNA sequences, some recalcitrant contexts still remain where the method stays inapplicable. When the sequences under consideration show a large difference of length, poor conservation at the primary level or important structural variations, all Sankoff-based method either do not apply or – if they do – the banding heuristics prevent the algorithm from finding the correct structure, while the memory and time consumption explodes. One of the main drawbacks of this kind of heuristics is their global nature: they cannot take advantage of local similarities in the sequences. When performing a global structural alignment, the shift restriction  $\delta$  should be at least the difference of length between the sequences. FOLDALIGN bypasses the later difficulty by doing local alignment as a default behavior. In this paper we prove that the banding can be local, which results in a huge gain, both in memory and time consumption. We

also propose a strategy to guarantee optimality in our framework even if we are using a heuristic. The strategy is presented in terms of constraints, as we believe indeed that the constraint paradigm (already in use for other kind of methods, like CONSAN [DE06] or STEMLOC [Hol05]) is helpful and relevant to model the forthcoming improvements of our method. We observe that the recursions of Sankoff are a natural combination of the two kinds of recursions it is supposed to extend: sequence alignment and secondary structure prediction. Likewise, a structural alignment may be modeled as a subgraph of a combination of two graphs - an alignment graph and a folding graph. Imposing structural constraints on those graphs may lead to efficient heuristic design. The authors of FOLDALIGN already proposed a combination of constraints on each of these graphs, namely alignment banding and structural banding ; the next section proposes a framework that allows for the integration of more accurate constraints. Experimental results are given in the last section.

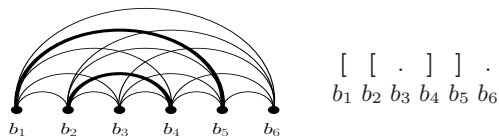
## 2 Applying constraints on the recursions of Sankoff

*Alignments* – An alignment of two sequences may be seen as a subgraph of the full bipartite graph where the nodes are the respective positions in each sequence. Such a subgraph (see Figure 1) is said to be an alignment if the following conditions hold: (1) the arity of each node is at most one, and (2) there is no crossing edge (we assume the nodes are placed in the sequence order). Moreover, we impose the extra restriction of maximality (3) no edge can be added without breaking one of the two former requirements. If the graph were not maximal in the sense of (3), it would mean that somewhere a deletion followed by an insertion would be preferred to a substitution. When the edges and remaining free bases are weighted according to the usual scoring schemes for two sequences, with linear or affine gap penalties [NW70,EGG88,EGGI92b,EGGI92a] such an alignment is automatically disqualified, since an appropriate shift of the corresponding bases would result immediately in an alignment of better score.



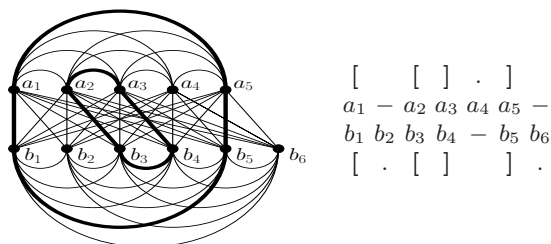
**Fig. 1.** An alignment can be seen as a subgraph of the full bipartite graph, we highlighted here the subgraph corresponding to a sample alignment

*Secondary structure* – Once again we consider the sequence as a list of nodes drawn in increasing order. Following the usual definition, a secondary structure is a subgraph of the full graph on these nodes with no crossing edge when drawn in a half plane (a so called outerplanar graph) and for which the node arities are at most one.



**Fig. 2.** A secondary structure and the corresponding subgraph

In our framework, a structural alignment of two sequences can be represented as a combination of three graphs: an alignment graph and two structure graphs, like in Figure 3. In the following, we will call **structural envelope** any graph that contains the structural alignment we are searching for. In practice, the structural envelope is not the full graph. For instance the steric constraints of the molecule imply that the minimal size for a loop should be three bases, consequently we can already remove all the edges that do not fulfill that constraint. In the next section, we investigate how the efficiency of the algorithm can be improved when extra constraints are imposed on the structural envelope.



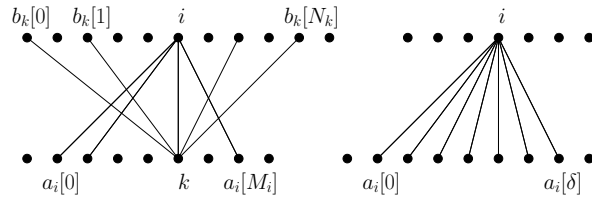
**Fig. 3.** Example of a structural alignment seen as a subgraph of its structural envelope. The structural envelope holds no constraint here.

*The recursions of Sankoff* – The recursion set used in RNA prediction methods based on free energy minimization [JTZ89,ZMT99,Zuk03,Hof03] is a more complex version of the Nussinov formulas [NJ80]. We express here the recursions given by Sankoff in the simplified model of Nussinov and we discuss later how far they can be extended to a more sophisticated model with no loss in complexity. While, for Nussinov, two indices are needed to represent any segment of the sequence under consideration, in the case of Sankoff, four indices are required to represent any pair of segment. We note  $seq_0 = seq_0[1..n]$  and  $seq_1 = seq_1[1..m]$  the two sequences in use. The Sankoff recursions call for the computation of a four-dimensional matrix where a cell  $\mathbf{e}[i, j, k, l]$  stores the score of the best structural alignment of the pair of segments  $seq_0[i..j]$  and  $seq_1[k..l]$ .

$$\mathbf{e}[i, j, k, l] := \text{MAX} \begin{cases} \mathbf{e}[i, j-1, k, l-1] + \text{align\_base}(seq_0[j], seq_1[l]) \\ \mathbf{e}[i, j, k, l-1] + \text{gap\_penalty}() \\ \mathbf{e}[i, j-1, k, l] + \text{gap\_penalty}() \\ \mathbf{e}[i, x-1, k, y-1] + \mathbf{e}[x+1, j-1, y+1, l-1] + \\ \quad \text{match\_pair}(seq_0[x], seq_0[j], seq_1[y], seq_1[l]) \\ \quad [i \leq x \leq j] \quad [k \leq y \leq l] \end{cases}$$

The value  $\mathbf{e}[1, n, 1, m]$  gives the best score for a structural alignment in that model. Any of the corresponding best alignments can be retrieved in linear time by tracing back into the dynamic programming matrix if the corresponding pairings that maximize the score were stored during the search, which would require only  $O(n^2)$  space. In that form, the Sankoff recursions may be considered, in a certain sense, a combination of the recursions for sequence alignment [SW81] and the recursions for secondary structure prediction of Nussinov. The Sankoff algorithm can integrate any Boolean restraints imposed (by the user or by automated methods) on the structural envelope by simply assigning an infinite score to the forbidden matches. What we show is that these constraints can be used to reduce both memory and time consumption.

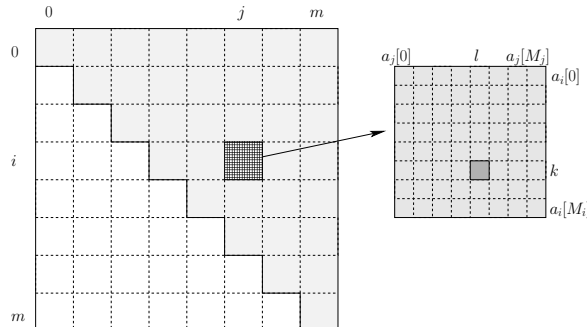
*Constraining the alignment* – Two bases, belonging to each sequence (or their index in the respective sequence) are said to be **alignable** if they are allowed to participate in the final alignment. We call the resulting subgraph the **alignment envelope**, as termed by Eddy and Holmes [Hol05,DE06] (Dowell and Eddy use a series of highly reliable single anchor points that they call *pins* to restrain the alignment. In our case, we do not adopt the same perspective: rather than anchoring the sequence alignment, we simply restrict it by constraints). In Figure 4, we show such an alignment envelope where we only represented the edges for one index in each sequence. For a given index  $i$  of arity  $M_i$  in the first sequence, we note  $a_i = a_i[0 .. M_i] = \{ a_i[0], \dots, a_i[M_i] \}$  the list of alignable bases, and for  $k$  in the second sequence, of arity  $N_k$ , we note  $b_k$  the list of its alignable bases in the first sequence. The sequence banding heuristic used in DYNALIGN and FOLDALIGN gives for any index a list of contiguous bases of fixed length  $\delta$  where the value for  $\delta$  is a global constant. In our case, alignable bases in a list do not need to be contiguous in the sequence.



**Fig. 4.** On the left are represented the lists  $a_i$  and  $b_k$  of alignable bases for some indices  $i$  and  $k$ . We note  $M_i$  the length of  $a_i$  and  $N_k$  the length of  $b_k$ . The figure on the right shows the list of contiguous indices for an arbitrary index  $i$  when using a sequence banding heuristic with a fixed global constant  $\delta$ , like in DYNALIGN and FOLDALIGN.

If a linear gap penalty is used and a double indel is assumed to be always worse than a substitution, only a reduced 4D-matrix needs to be allocated (Figure 5). Like for the banding heuristic, only the indices corresponding to the admissible pairs of segments need to be allocated but in our case, we are far more flexible on the possible indices, as discussed in the last paragraph. We prove that when a recursive call falls out of that allocated matrix during the com-

putation process, the best structural alignment between any segments  $seq_0[i..j]$  and  $seq_1[k..l]$  can be retrieved in constant time from the allocated part of the 4D-matrix. The demonstration of that result may not be valid for more sophisticated scoring schemes and would also have to be adapted for more than two sequences, where the second assumption about the preference of a substitution over a double indel is not always true in an optimal multiple alignment.



**Fig. 5.** The 4-dimensional matrix used for the computation of the best structural alignment can be represented as a bi-dimensional matrix, each cell being a bi-dimensional matrix. The value in the cell  $[i, j, x, y]$  stores the score for the best structural alignment between the segments  $seq_0[i..j]$  and  $seq_1[a_i[x]..a_i[y]]$ .

To prove that the score  $\mathbf{e}[i, j, k, l]$  can be retrieved in constant time from the reduced matrix of Figure 5, let's assume that during the computation  $\mathbf{e}[i, j, k, l]$  falls out of the matrix. This means that either  $i$  and  $k$  are not alignable and/or  $j$  and  $l$  are not alignable. We detail the right hand case, the other one is symmetric. Given two non-empty segments  $[i..j]$  and  $[k..l]$ , belonging to each sequence, if  $j$  and  $l$  are not alignable then either  $j$  align with some base in  $[k..l]$ , or  $l$  with some base in  $[i..j]$ . Otherwise  $j$  and  $l$  would both create an indel and the optimal alignment of the two segments would then result in a double indel which was previously assumed to be more expensive than a substitution. Let's assume first that  $l$  aligns with a base in  $[i..j]$ . Consequently  $[i..j] \cap b_l \neq \emptyset$ . Let's call  $\beta_l = \max([i..j] \cap b_l)$ . Every base of the segment  $[\beta_{l+1}..j]$  has no alignable partner in  $[k..l]$  or it would contradict the maximality of  $\beta_l$ . The remaining sequence  $seq_0[\beta_{l+1}..j]$  has no other possibility than being deleted, resulting in a  $(j - \beta_l)$  long gap in the alignment. This gives:

$$\mathbf{e}[i, j, k, l] = \mathbf{e}[i, \beta_l, k, l] + (j - \beta_l) * \mathbf{gap\_penalty}$$

If  $l$  does not align, then  $j$  must align with some base in  $[k..l]$  and we pose  $\alpha_j = \max([k..l] \cap a_j)$ . In that case we must have:

$$\mathbf{e}[i, j, k, l] = \mathbf{e}[i, j, k, \alpha_j] + (l - \alpha_j) * \mathbf{gap\_penalty}$$

Note that  $\alpha_j$  and  $\beta_l$  depend only on three indices and can be precomputed in cubic time so that the former retrieval can then be managed in constant time

when the routines are calling an index out of the matrix. The left hand case with indices  $i$  and  $k$  is symmetric. As the indices are not dependent, a single test on each of the four indices is enough to guarantee to fall back in the allocated part.

*Implementation* – We provide and discuss an implementation (named ARNICA) based on a rather empirical scoring scheme for sequence alignment and a reduced energy model that partially takes into account the stabilizing effect of base pair stacking in stems, aimed at exemplifying the gain both in space and time consumption when running on live sequences. ARNICA proceeds in three steps, detailed below:

- **1. Setting alignment constraints** - we build the graph of alignable bases by computing all the alignments within a user-specified distance of the optimum alignment value
- **2. Setting structural constraints** - we compute all the pairing probabilities for each sequence and filter with a threshold
- **3. Computing common folding and alignment** - we compute the optimal structural alignment with the recursions of Sankoff

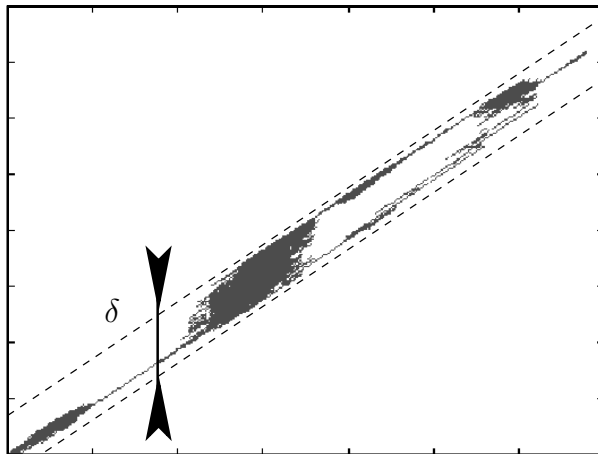
**1. Setting alignment constraints** – This first phase uses a variant of the standard alignment with affine gap penalty (`open_cost = -80`, `elongation_cost = -30`) reminiscent of the algorithm of Waterman [Wat83] that gives for each pair of positions  $(i, j)$  the score of the best alignment when the bases at position  $i$  in the first sequence and  $j$  in the second one are imposed to be aligned. The variant of the algorithm has the same algorithmic complexities than for standard alignment methods. Then we build the adjacency matrix of the alignment envelope by simply applying a threshold (`thd`): a pair  $i, j$  will be allowed to align if there exists at least one alignment passing by this coordinate with a global score exceeding the best alignment score by less than `thd`.

**2. Setting structural constraints** – In the second phase, the pairing probabilities are computed for each sequence with the McCaskill algorithm [McC90] and the graph of possible pairings is filtered: two bases for which the pairing probability is less than 1% are not allowed to pair. This has no consequence on the computation space/time but simply increases the quality of the solutions found by discarding spurious base pairs.

**3. Computing common folding and alignment** – The optimal folding and alignment is computed with the recursions of Sankoff. The alignment part of the score uses a linear gap penalty scheme and the structural part a probability-based score. The combination of two scores of different nature (an alignment score and a structure score, here based on probabilities) is a difficult issue and a full problem in itself, related to the paradox of structural alignment, not discussed in the present paper. The model in use takes into account only indirectly (via the algorithm of McCaskill) the stabilizing and destabilizing effects of stacking. Integrating the full thermodynamic model would provide results more accurate biologically but also call for non-straightforward developments and adaptations. The folding computed by ARNICA is optimal with regard to

our model. The next section demonstrate that the method already appears competitive despite the simplicity of the model and that it clearly outperforms other programs based on the same recursions when the data presents uneasy features.

*Discussion on complexity* – The size of the 4-dimensional matrix to be allocated in the last step obviously increases with the alignment threshold chosen in the first step. If the threshold chosen is infinite, then the alignment envelope is the full bipartite graph and there is no gain over the complexities of the Sankoff recursions. In practice however, our framework is quite effective and far more efficient than any alignment banding heuristic, which is no more than a peculiar case of it. The algorithmic complexities for a Sankoff-based pairwise secondary structure alignment with alignment banding  $\delta$  are  $O(\delta^2 n^2)$  in space and  $O(\delta^3 n^3)$  in time (we suppose that  $m \sim n$ ). They can be reformulated as  $O(\alpha^2)$  and  $O(\alpha^3)$ , where  $\alpha \sim \delta n$  is the size of the diagonal band corresponding to the alignment envelope induced by the banding heuristic. More precisely,  $\alpha$  is the size of the *true* zone in the adjacency matrix of the alignment envelope. As shown on Figure 6, this zone is an exact diagonal band in the banding heuristic, whereas it can be a tunable zone in our framework.



**Fig. 6.** Suboptimal alignments of two RNase P RNA (D.desulfuricans vs A.eutrophus) showing alternative alignment paths resulting from large zones of deletion. To reach any of these suboptimal alignments with a banding heuristic, the value chosen for the allowed shift  $\delta$  has to encompass all the possible paths.

### 3 First experimental results

As our main focus is the difficult context of few poorly conserved RNA sequences, we have selected 7 RNase P RNA of the alpha subdivision taken in the database of Brown [Bro99], showing deep variations in structure, which make them difficult candidates for all Sankoff-based methods, as mentioned by [GG04]. The

sequences are around 400 bases long with an average identity rate of 64%. A wider performance assessment will be proceeded along with the next improvements on ARNICA. In this paper, we only compared to FOLDALIGN for simplicity, as it is one of the prominent Sankoff-based structural alignment method. The program FOLDALIGN does not allocate the whole needed memory at once: in the comparisons, we always display the maximum amount of memory in use by the program during the computation. To keep in reasonable space and time limits, and given that a memory allocation of several hundreds of Mb would usually result in hours of computation, we stopped the computation whenever the estimated needed resources for memory were above 300Mb. All the tests were run on an IBM thinkpad T40 (pentium 1.5GHz - RAM 512Mb). Table 1 gives the average performances of ARNICA with different values for the threshold, compared to FOLDALIGN ; the results for a medium threshold of 100 are detailed in Table 2. For FOLDALIGN (version 2.1.0), we use the default parameters and the default option of performing a local alignment, as the global option can seldom be chosen, the difference of length being too important between the sequences. However, we indicate in Table 2 the percentage of sequence covered by the structural alignment predicted. When this coverage is close to 100%, the difference between local and global is weak and comparing the performances is meaningful.

	option	specificity	sensitivity	time (sec)	space (Mb)
FOLDALIGN	local	56.2%	40.5%	1107	142.1
ARNICA	thd 0	73.6%	43.3%	6	16.5
	thd 10	74.1%	45.9%	7	16.7
	thd 30	75.7%	51.0%	10	17.5
	thd 50	76.2%	55.0%	20	19.1
	thd 80	75.7%	58.3%	77	23.7
	thd 100	74.7%	58.7%	148	29.0
	thd 150	73.9%	60.5%	536	46.5
	thd 200	73.2%	61.0%	1079	64.4

**Table 1.** Average performance of ARNICA and FOLDALIGN on a set of RNase P (alpha subdivision) ; specificity = number of true predicted pairings / total number of predicted pairings ; sensitivity = number of true predicted pairings / number of pairings in the known structure

On this data set, FOLDALIGN seems to be limited by its heuristic restrictions whereas ARNICA performs better from any point of view (specificity, sensitivity, time and memory usage) remaining fast and low memory consuming. The average gain in computational time and memory with a threshold 100 is by a factor 7 and 5, and the correctness of ARNICA is neighboring 75%. This globally stable behavior despite the divergence of structure is a promising advantage for the integration of a more complete model.

## 4 Conclusions

This paper proposes a refinement of the heuristics commonly used by the Sankoff-based pairwise secondary structure RNA prediction methods. The strategies for heuristic design are conceptualized and refined using the constraints paradigm in a simpler model, that actually constitutes the core of these methods, for which we proved that our framework is valid. Our ongoing developments aim at incorporating a more complete thermodynamic model while refining even further the method by allowing dynamic restraints on the graphs, the promising behavior of ARNICA on poorly conserved structures being its major asset.

## References

- [Bro99] J.W. Brown. The Ribonuclease P database. *NAR*, 27(314), 1999. <http://www.mbio.ncsu.edu/RNaseP/>.
- [CMR08] Emidio Capriotti and Marc A. Marti-Renom. Computational RNA structure prediction. *Current Bioinformatics*, 3(1):32–45, January 2008.
- [DE06] Robin D. Dowell and Sean R. Eddy. Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinformatics*, 7:400+, September 2006.
- [ED94] S.R. Eddy and R. Durbin. RNA sequence analysis using covariance models. *NAR*, 22:2079–2088, 1994.
- [EGG88] D. Eppstein, Z. Galil, and R. Giancarlo. Speeding up dynamic programming. In *Proceedings of the 29th IEEE Annual Symposium on Foundations of Computer Science*, pages 488–496, White Plains, NY, 1988. IEEE Computer Society Press.
- [EGGI92a] David Eppstein, Zvi Galil, Raffaele Giancarlo, and Giuseppe F. Italiano. Sparse dynamic programming I: linear cost functions. *J. ACM*, 39(3):519–545, 1992.
- [EGGI92b] David Eppstein, Zvi Galil, Raffaele Giancarlo, and Giuseppe F. Italiano. Sparse dynamic programming II: convex and concave cost functions. *J. ACM*, 39(3):546–567, 1992.
- [GG04] P. P. Gardner and R. Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5(1), September 2004.
- [HBS04] I. L. Hofacker, S. H. Bernhart, and P. F. Stadler. Alignment of RNA base pairing probability matrices. *Bioinformatics*, 20(14):2222–2227, September 2004.
- [HLSG05] Jakob Hull Havgaard, Rune B. Lyngsø, Gary D. Stormo, and Jan Gorodkin. Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. *Bioinformatics*, 21(9):1815–1824, 2005.
- [Hof03] I. L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Res*, 31(13):3429–3431, July 2003.
- [Hol05] I. Holmes. Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*, 6(1), 2005.
- [HSM07] Arif O. Harmanci, Gaurav Sharma, and David H. Mathews. Efficient pairwise RNA structure prediction using probabilistic alignment constraints in dynalign. *BMC Bioinformatics*, 8, April 2007.

- [HTG07] Jakob H. Havgaard, Elfar Torarinsson, and Jan Gorodkin. Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix. *PLoS Computational Biology*, 3(10):e193+, October 2007.
- [JTZ89] J.A. Jaeger, D.H. Turner, and M. Zuker. Improved predictions of secondary structures for RNA. *PNAS*, 86:7706–7710, October 1989.
- [KH03] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res*, 31(13):3423–3428, July 2003.
- [McC90] J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.
- [MSZT99] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *JMB*, 288:911–940, 1999.
- [MT02] D.H. Mathews and D.H. Turner. Dynalign: An algorithm for finding the secondary structure common to two RNA sequences. *JMB*, -:in press, 2002.
- [NJ80] R. Nussinov and A.B. Jacobson. Fast algorithm for predicting the secondary structure of single stranded RNA. *PNAS*, 77:6309–6313, 1980.
- [NW70] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453, March 1970.
- [OH98] E.A. O’Brien and D.G. Higgins. Empirical estimation of the reliability of ribosomal RNA alignments. *Bioinformatics*, 14(10):830–838, 1998.
- [San85] D. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, 45(5):810–825, 1985.
- [SW81] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *JMB*, 147:195–197, 1981.
- [SYKB07] B. A. Shapiro, Y. G. Yingling, W. Kasprzak, and E. Bindewald. Bridging the gap in RNA structure prediction. *Curr Opin Struct Biol*, 17(2):157–165, April 2007.
- [THG07] Elfar Torarinsson, Jakob H H. Havgaard, and Jan Gorodkin. Multiple structural alignment and clustering of RNA sequences. *Bioinformatics*, February 2007.
- [Wat83] M. S. Watermann. Sequence alignments in the neighborhood of the optimum with general application to dynamic programming. *Applied Mathematical Sciences*, 80:3123–3124, May 1983.
- [ZMT99] M. Zuker, D.H. Mathews, and D.H. Turner. Algorithms and thermodynamics for RNA secondary structure prediction. A practical guide. *RNA Biochemistry and Biotechnology*, 1999.
- [Zuk03] M. Zuker. MFOLD web server for nucleic acid folding and hybridization prediction. *NAR*, 31(13):1–10, 2003.

				ARNICA						FOLDALIGN						
sequence 1	sequence 2	id	$\delta l$	spec1	sens1	spec2	sens2	time	mem	spec1	sens1	spec2	sens2	time	mem	cov
C.crescentus	A.tumefaciens	68%	4	<b>70%</b>	<b>69%</b>	<b>74%</b>	<b>73%</b>	<b>46</b>	<b>22.7</b>	69%	66%	65%	62%	1428	170.5	99%
R.capsulatus	A.tumefaciens	67%	3	<b>83%</b>	<b>81%</b>	<b>82%</b>	<b>84%</b>	<b>31</b>	<b>21.4</b>	59%	54%	74%	71%	1647	181.4	99%
R.capsulatus	C.crescentus	67%	7	<b>77%</b>	<b>58%</b>	<b>73%</b>	<b>58%</b>	<b>55</b>	<b>23.6</b>	49%	33%	45%	32%	1090	169.9	71%
R.palustris	A.tumefaciens	75%	77	<b>77%</b>	<b>59%</b>	<b>77%</b>	<b>78%</b>	<b>87</b>	<b>30.1</b>	36%	17%	42%	26%	1329	200.6	58%
R.palustris	C.crescentus	65%	81	<b>83%</b>	<b>57%</b>	<b>77%</b>	<b>71%</b>	<b>114</b>	<b>32.1</b>	19%	12%	27%	23%	1840	226.5	87%
R.palustris	R.capsulatus	67%	74	<b>82%</b>	<b>61%</b>	<b>78%</b>	<b>74%</b>	<b>287</b>	<b>42.4</b>	24%	16%	27%	23%	2326	262.7	91%
R.prowazekii	A.tumefaciens	60%	17	<b>75%</b>	<b>50%</b>	<b>74%</b>	<b>48%</b>	<b>37</b>	<b>21.3</b>	70%	59%	66%	53%	516	79.0	87%
R.prowazekii	C.crescentus	56%	13	<b>84%</b>	<b>52%</b>	<b>80%</b>	<b>49%</b>	<b>66</b>	<b>24.1</b>	65%	52%	61%	48%	423	76.1	84%
R.prowazekii	R.capsulatus	56%	20	<b>80%</b>	<b>47%</b>	<b>83%</b>	<b>45%</b>	<b>39</b>	<b>21.5</b>	61%	51%	61%	48%	585	75.4	86%
R.prowazekii	R.palustris	60%	94	<b>78%</b>	<b>53%</b>	<b>81%</b>	<b>41%</b>	<b>161</b>	<b>31.6</b>	35%	28%	40%	24%	554	83.9	76%
R.rubrum	A.tumefaciens	74%	27	<b>70%</b>	<b>57%</b>	<b>71%</b>	<b>65%</b>	<b>43</b>	<b>23.2</b>	46%	43%	62%	65%	1991	208.6	98%
R.rubrum	C.crescentus	69%	31	<b>60%</b>	<b>51%</b>	<b>64%</b>	<b>61%</b>	<b>28</b>	<b>21.3</b>	62%	57%	63%	64%	2030	214.4	97%
R.rubrum	R.capsulatus	68%	24	<b>77%</b>	<b>63%</b>	<b>81%</b>	<b>71%</b>	<b>46</b>	<b>24.0</b>	68%	63%	75%	74%	2253	252.2	96%
R.rubrum	R.palustris	74%	50	<b>80%</b>	<b>68%</b>	<b>83%</b>	<b>60%</b>	<b>167</b>	<b>34.5</b>	34%	30%	43%	32%	2960	299.5	91%
R.rubrum	R.prowazekii	55%	44	<b>39%</b>	<b>26%</b>	<b>59%</b>	<b>45%</b>	<b>728</b>	<b>57.7</b>	66%	47%	77%	64%	588	94.4	80%
Wolbachia-sp	A.tumefaciens	60%	54	<b>67%</b>	<b>65%</b>	<b>72%</b>	<b>59%</b>	<b>71</b>	<b>23.2</b>	69%	35%	69%	29%	277	65.8	51%
Wolbachia-sp	C.crescentus	56%	50	<b>72%</b>	<b>49%</b>	<b>76%</b>	<b>44%</b>	<b>95</b>	<b>25.3</b>	56%	36%	61%	33%	309	66.0	64%
Wolbachia-sp	R.capsulatus	59%	57	<b>77%</b>	<b>73%</b>	<b>78%</b>	<b>60%</b>	<b>87</b>	<b>24.8</b>	81%	39%	81%	31%	265	62.9	48%
Wolbachia-sp	R.palustris	58%	131	<b>81%</b>	<b>78%</b>	<b>85%</b>	<b>53%</b>	<b>744</b>	<b>54.5</b>	33%	10%	57%	11%	274	69.2	28%
Wolbachia-sp	R.prowazekii	67%	37	<b>61%</b>	<b>42%</b>	<b>70%</b>	<b>42%</b>	<b>77</b>	<b>23.3</b>	70%	39%	79%	34%	195	55.0	58%
Wolbachia-sp	R.rubrum	60%	81	<b>79%</b>	<b>76%</b>	<b>68%</b>	<b>49%</b>	<b>97</b>	<b>25.6</b>	62%	42%	51%	26%	376	79.0	61%
average				<b>74%</b>	<b>59%</b>	<b>76%</b>	<b>59%</b>	<b>148</b>	<b>29.0</b>	54%	39%	58%	42%	1107	142.1	

**Table 2.** Performance of ARNICA and FOLDALIGN on a set of RNase P (alpha subdivision). Each sequence is folded together with each other. We display their percentage of identity and their difference in length (their average length is around 400 bases) [spec1 and spec2 (specificity for each of the two sequences) - sens1 and sens2 (sensitivity) - time is in seconds - mem is in Mb - cov is the percentage of sequence covered by the local alignment given by FOLDALIGN (ARNICA is global)]