doi:10.1006/jmbi.2001.5351 available online at http://www.idealibrary.com on IDEAL® J. Mol. Biol. (2002) 317, 191–203

JMB

Dynalign: An Algorithm for Finding the Secondary Structure Common to Two RNA Sequences

David H. Mathews and Douglas H. Turner*

Department of Chemistry University of Rochester, RC Box 270216, Rochester, NY 14627-0216, USA

*Corresponding author

With the rapid increase in the size of the genome sequence database, computational analysis of RNA will become increasingly important in revealing structure-function relationships and potential drug targets. RNA secondary structure prediction for a single sequence is 73 % accurate on average for a large database of known secondary structures. This level of accuracy provides a good starting point for determining a secondary structure either by comparative sequence analysis or by the interpretation of experimental studies. Dynalign is a new computer algorithm that improves the accuracy of structure prediction by combining free energy minimization and comparative sequence analysis to find a low free energy structure common to two sequences without requiring any sequence identity. It uses a dynamic programming construct suggested by Sankoff. Dynalign, however, restricts the maximum distance, *M*, allowed between aligned nucleotides in the two sequences. This makes the calculation tractable because the complexity is simplified to $O(M^3N^3)$, where N is the length of the shorter sequence.

The accuracy of Dynalign was tested with sets of 13 tRNAs, seven 5 S rRNAs, and two R2 3′ UTR sequences. On average, Dynalign predicted 86.1 % of known base-pairs in the tRNAs, as compared to 59.7 % for free energy minimization alone. For the 5 S rRNAs, the average accuracy improves from 47.8 % to 86.4 %. The secondary structure of the R2 3′ UTR from *Drosophila takahashii* is poorly predicted by standard free energy minimization. With Dynalign, however, the structure predicted in tandem with the sequence from *Drosophila melanogaster* nearly matches the structure determined by comparative sequence analysis.

Keywords: RNA secondary structure; free energy minimization; comparative sequence analysis

Introduction

The rapidly expanding databases of genome sequences provide a foundation for rapidly generating new databases of RNA secondary structures. These secondary structures are important for understanding structure-function relationships and choosing drug targets. Comparative sequence analysis is the gold standard for determination of RNA secondary structure in the absence of a structure solved by X-ray crystallography.[1] Structures of large RNAs solved by X-ray crystallography have largely verified the base-pairs predicted by comparative sequence analysis.[2-8] While only a small number of RNA structures have been determined by crystallography, many classes of RNAs have secondary structures determined by comparative sequence analysis. These include the small subunit rRNA,[9] large subunit rRNA,[10] 5 S rRNA,[11] group I intron,[12] group II intron,[13] RNAase P RNA,[14] SRP RNA,[15] tRNA,[16] telomerase RNA,[17,18] and tmRNA.[19]

Comparative sequence analysis requires an alignment of a large number of sequences with identical function. When only one sequence is available, the secondary structure can be predicted on the basis of free energy minimization with an accuracy of roughly 73 % on average for sequences of less than 700 nucleotides.[20,21] Several algorithms are available for free energy minimization of RNA secondary structure.[20–25]

Algorithms have also been developed to combine free energy minimization with comparative sequence analysis.[26–32] The advantages of these

E-mail address of the corresponding author: turner@chem.rochester.edu

0022-2836/02/020191–13 $35.00/0

programs are the improved accuracy of secondary structure prediction and automation of the laborious process of comparative sequence analysis.

Many of the algorithms that employ free energy minimization as a tool for comparative sequence analysis require a fixed sequence alignment as input.[26–28,31,32] Alignments determined by sequence matching, however, are complicated by compensating base changes and the fact that most RNAs are composed of only four different nucleotides. The fixed alignment can be flawed and so may restrict the algorithms' ability to find a conserved structure.

Algorithms that use free energy minimization to find a conserved structure without assuming a fixed alignment are more robust,[30,33] although they are generally more time consuming. Notredame *et al.*[29] wrote a program that uses a genetic algorithm to find the structure of a sequence given a second, related sequence with known structure. Chen *et al.*[30] developed a genetic algorithm that finds a conserved structure for a set of sequences without requiring a known structure.

Eddy and Durbin[34] developed an approach to automate comparative sequence analysis that is not based on free energy minimization. They developed a covariance model that takes a set of unaligned RNA sequences and determines a sequence alignment and consensus structure with multiple rounds of refinement.[34]

Sankoff[35] proposed that a dynamic programming algorithm could simultaneously solve the sequence alignment and folding problems for multiple sequences. Gorodkin *et al.*[33] wrote the first practical algorithm of this type, FOLDALIGN, by utilizing three simplifications to speed the calculation. Firstly, the dynamic programming calculation is limited to predicting the structures for two sequences at a time. Secondly, the algorithm optimizes the number of base-pairs in the structures, rather than the free energies. Thirdly, multibranch loops are not allowed.

Here, a dynamic programming algorithm, called Dynalign, is presented that aligns two sequences and finds a common structure, including multibranch loops. Dynalign is based on the dynamic programming solution proposed by Sankoff[35] and uses nearest-neighbor rules for predicting the free energies of secondary structures.[20,36,37] When tested with tRNA, 5 S rRNA, and R2 3′ UTR RNAs, Dynalign improves the accuracy of secondary structure prediction relative to prediction for a single sequence by free energy minimization.

Results

Algorithm

Dynalign is a dynamic programming algorithm that takes two sequences as input and then outputs a sequence alignment and a common structure for the two sequences. The sequence alignment indicates the nucleotides aligned in paired regions, but does not align exactly those nucleotides in unpaired regions. For the common structure, base-pairs are allowed only if both sequences can accommodate a canonical pair at the same position in the alignment. Dynalign minimizes the total free energy of the system, $\Delta G^\circ_{\text{total}}$, where:

$$\Delta G^\circ_{\text{total}} = \Delta G^\circ_{\text{sequence 1}} + \Delta G^\circ_{\text{sequence 2}} + (\Delta G^\circ_{\text{gap}})\ (\text{number of gaps}) \quad (1)$$

Gaps are locations in a sequence alignment for which a nucleotide in one sequence has no analogous nucleotide in the second sequence. $\Delta G^\circ_{\text{sequence 1}}$ and $\Delta G^\circ_{\text{sequence 2}}$ are the conformational free energies for sequences 1 and 2, respectively, calculated with a nearest-neighbor approximation.[20,36–38] $\Delta G^\circ_{\text{gap}}$ is an empirical factor that penalizes each gap nucleotide in the alignment with a value that scales the penalty to relate to the conformational free energies. This definition of $\Delta G^\circ_{\text{total}}$ does not depend on matching nucleotides in the sequence alignment and, therefore, no sequence identity is required for accurate sequence alignment and structure prediction.

The Dynalign algorithm is the practical application of the method suggested by Sankoff[35] and is the synthesis of the dynamic programming algorithms for sequence alignment[39,40] and RNA secondary structure prediction.[41] To make the calculation tractable, the problem is restricted from the original formulation[35] in two ways. Firstly, the common structure and alignment is found for only two sequences. Secondly, it is assumed that the position of aligned nucleotides will be within a certain maximum distance, called M. This makes intuitive sense because, for example, an alignment of the first nucleotide of one sequence to the last nucleotide of the other sequence is not worth considering. M is defined by the user and must be tailored to the system being studied.

The original algorithm by Sankoff[35] included a scoring function that accounted for sequence matching in the alignment. Dynalign, however, does not explicitly score the sequence identity and, therefore, it does not depend upon sequence homology to find the common structure. This is an advantage because compensating base changes in comparative sequence analysis are proof of helices, but confound any automation of sequence alignment based on nucleotide matching. Sequence is implicitly considered in the free energy nearest-neighbor parameters.[20,36] For example, hairpin loops of four unpaired nucleotides are given enhanced stability if they have certain sequences that occur frequently in known secondary structures. Therefore, the nearest-neighbor parameters are likely to conserve such tetraloops.

By design, base-pairs in the structure predicted for one sequence correspond one to one with base-pairs in the second sequence, but one exception

can be allowed. Single inserted base-pairs can be included in either structure if that base-pair is between two conserved base-pairs. In this case, the gap penalty is applied twice as this inserted pair is opposite two gaps in the sequence alignment. This feature can be toggled on or off by the user.

Dynalign improves the accuracy of secondary structure prediction

To test the hypothesis that RNA secondary structure prediction can be improved by finding the lowest free energy structure common to two sequences, three types of RNA with known secondary structures were studied: 13 tRNAs,[16] seven 5 S rRNAs,[11] and two R2 3′ UTRs.[42]

For the 13 tRNAs, there are 156 possible pairwise combinations. To examine the effect of ΔG°_{gap}, each of these combinations was tested with a range of gap penalties as large as 2.0 kcal mol^{-1} (gap nt)$^{-1}$ and single base-pair inserts allowed (Figure 1). The accuracy of tRNA secondary structure prediction by Dynalign is improved by suppressing the insertion of gaps and therefore the accuracy increases with increasing ΔG°_{gap} up to 1.9 kcal mol^{-1} (gap nt)$^{-1}$ and then levels off (Figure 1).

Sequences of tRNA were chosen both with (RL0503, RL1141, RL6371, RS0380, and RS1141) and without (RD0260, RD1140, RD2640, RD4800, RE2140, RE6781, and RF6320) inserts in the variable region. Figure 1 shows the average accuracy of secondary structure prediction by Dynalign for cases in which the two RNAs are similar in length (both either with or without insert) and when they are different in length (one with and one without insert). The average accuracy of prediction is significantly better when the two RNA sequences are of similar length. This is likely due to the fact that the variable region, if present, forms a fifth helix. An inserted helix in one sequence presents a more difficult problem for free energy minimization.

For all pairwise tRNA combinations and ΔG°_{gap} of 2.0 kcal/mol, the average accuracy of structure prediction by Dynalign is 86.1 % when single base-pair inserts are forbidden. The average accuracy by free energy minimization alone of each individual strand is only 59.7 %. Table 1 shows the accuracy of prediction for each tRNA sequence paired with each other sequence. The first column gives the accuracy of secondary structure prediction by free energy minimization of the sequence alone;[20] these sequences were chosen so that those with structures predicted poorly by free energy minimization[20] are over-represented. When two sequences with well predicted secondary structures are considered in tandem by Dynalign, such as RD1140 and RE2140, the accuracy remains 100 % for each sequence. When one sequence is well predicted and the second is poorly predicted, such as RD1140 and RF6340, the Dynalign result generally provides better accuracy. Most importantly, in many cases, when two poorly predicted tRNAs are considered in tandem, the Dynalign result is an improvement in accuracy for both. Therefore, the comparative sequence analysis component of the algorithm is able to improve the accuracy of structure prediction for cases when the standard free energy minimization algorithm is inadequate.

To illustrate the improvement in tRNA secondary structure prediction by including comparative sequence analysis, Figure 2 shows results for

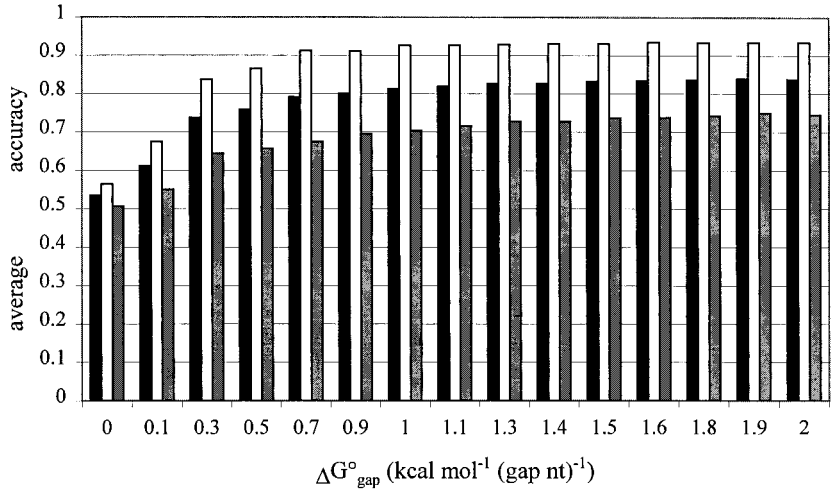


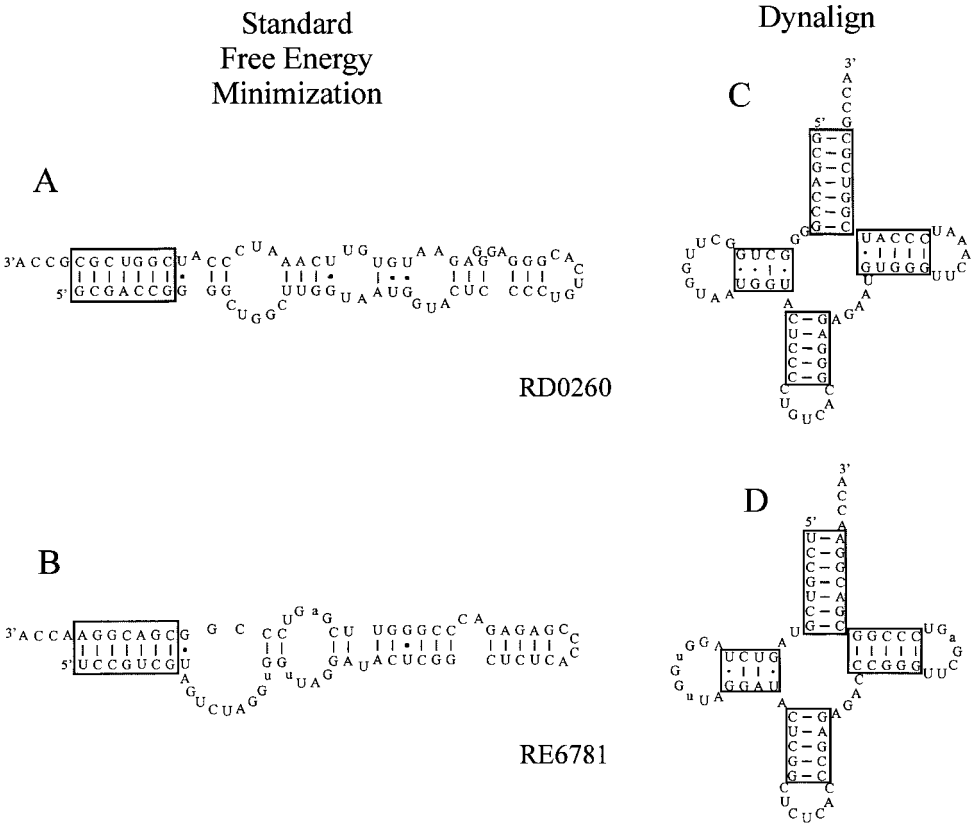
Figure 1. The average accuracy of secondary structure prediction of the tRNA database by Dynalign as a function of ΔG°_{gap}. The filled bars are the average for all pairwise tRNA combinations. The white bars are the average for all combinations of tRNAs either with or without an insert in the variable region and the gray bars are the average for combinations of tRNAs when one had an insert and the other did not have an insert in the variable region. The calculations were with $M = 15$ and single base-pair inserts were allowed. Modified nucleotides that cannot canonically pair are prohibited to base-pair.[20] The average accuracy for all the tRNAs folded individually by free energy minimization is 59.7 %.

RD0260 and RE6781. As indicated by Table 1, the accuracy of prediction for each alone is 33.3 %,[20] but the accuracy of prediction by Dynalign is 100 % for the pairwise combination.

For the 5 S rRNA sequences, the optimization of ΔG°_{gap} was repeated. Figure 3 shows the average accuracy for all 42 pairwise combinations with ΔG°_{gap} from 0.0 to 1.6 kcal mol^{-1} (gap nt)$^{-1}$. The optimal accuracy is with ΔG°_{gap} of 0.4 kcal mol^{-1} (gap nt)$^{-1}$. This suggests that the optimal value of ΔG°_{gap} will depend on the sequence studied. The smaller gap penalty for 5 S rRNA as compared to tRNA is explained by the difference in the sequence alignments determined by comparative sequence analysis. The 5 S sequence alignment requires gaps in both sequences in order to keep the base-paired nucleotides in register. The tRNA alignment instead requires a gap inserted in one sequence at most and therefore the number of gaps is generally the minimum possible, i.e. the difference in sequence lengths. Therefore, a large gap penalty improves tRNA structure prediction accuracy because it suppresses spurious gaps, whereas a smaller gap penalty maximizes 5 S rRNA prediction accuracy because more gaps are required.

Table 2 shows the accuracy of structure prediction by Dynalign for each pairwise combination of 5 S rRNA sequences and of each sequence predicted by using free energy minimization alone.[20] With a gap penalty of 0.4 kcal mol^{-1} (gap nt)$^{-1}$, the average accuracy for this set of 5 S rRNA sequences is improved from 47.8 % to 86.4 % by using Dynalign to take advantage of the comparative sequence data.

The improved accuracy of 5 S rRNA structure prediction is illustrated in Figure 4 for sequences from *Arthrobacter globiformis* and *Halofera volcanii*. Free energy minimization alone poorly predicts the




```
RD0260 GCGACCGGGGCUGGCUUGGUAAUGGUACUCCCCUGUCACGGGAGAGAAUGUGGGUUCAAAUCCCAUCGGUCGCGCCA
RE6781 UCCGUCGUAGUCUAGGuGGuUAGGAUACUCGGCUCUCACCCGAGAGAC-CCGGGUUCGaGUCCCGGCGACGGAACCA
       ^^^^^^^  ^^^^          ^^^^ ^^^^^       ^^^^^      ^^^^^       ^^^^^^^^^^^^
```

Figure 2. Prediction of secondary structure and sequence alignment for RD0260 and RE6781. Structures A and B are the secondary structures predicted for RD0260 and RE6781, respectively, by free energy minimization.[20] Structures C and D are the structures predicted by Dynalign for RD0260 and RE6781, respectively, when they are considered in tandem. The boxed helices are those helices established by comparative sequence analysis.[16] Lowercase nucleotides are modified nucleotides that cannot canonically pair and are forced single-stranded.[11,16,20] On the bottom is the sequence alignment predicted by Dynalign. Base-paired nucleotides are indicated by carets. The ΔG°_{37} for RD0260 and RE6781 are −24.5 and −24.4 kcal/mol, respectively. The single gap cost 2.0 kcal/mol for the final total score of −46.9 kcal/mol.

Table 1. The percent accuracy of tRNA sequences folded with Dynalign

	Single sequence	RD0260	RD0500	RD1140	RD2640	RD4800	RE2140	RE6781	RF6320	RL0503	RL1141	RL6371	RS0380	RS1141
RD0260	33.3	-	95.2	100.0	100.0	57.1	100.0	100.0	100.0	95.2	71.4	95.2	100	71.4
RD0500	61.9	95.2	-	95.2	95.2	57.1	95.2	95.2	61.9	85.7	85.7	57.1	85.7	57.1
RD1140	100.0	100.0	95.2	-	100.0	100.0	100.0	100.0	100.0	95.2	95.2	95.2	100	71.4
RD2640	42.9	100.0	95.2	100.0	-	100.0	100.0	100.0	100.0	95.2	95.2	95.2	100	95.2
RD4800	42.9	57.1	52.4	100.0	100.0	-	100.0	100.0	100.0	57.1	47.6	95.2	66.7	0.0[a]
RE2140	100.0	100.0	95.2	100.0	100.0	100.0	-	100.0	100.0	95.2	57.1	95.2	100	47.6
RE6781	33.3	100.0	95.2	100.0	100.0	100.0	100.0	-	100.0	95.2	47.6	95.2	100	47.6
RF6320	0.0	95.5	59.1	95.5	95.5	95.4	95.5	95.5	-	95.5	68.2	95.5	95.5	54.5
RL0503	57.1	95.2	76.2	95.2	95.2	47.6	95.2	95.2	100.0	-	95.2	95.2	95.2	95.2
RL1141	50.0	75.0	75.0	100.0	100.0	35.0	75.0	50.0	75.0	100	-	95	100	85
RL6371	95.2	95.2	57.1	95.2	71.4	71.4	71.4	95.2	100.0	95.2	90.5	-	95.2	95.2
RS0380	100.0	100.0	81.0	100.0	100.0	42.9	100.0	100.0	100.0	95.2	95.2	95.2	-	95.2
RS1141	60.0	75.0	55.0	75.0	75.0	0.0[a]	30.0	30.0	55.0	100	85	100	100	-

The accuracy of each sequence folded alone is scored in the first column. Each other column gives the accuracy of secondary structure prediction by Dynalign of the sequence indicated by the row when paired with the sequence indicated at the top of the column. The calculations were performed with $M = 15$, $\Delta G^{\circ}_{gap} = 2.0$ kcal mol^{-1} (gap nt)$^{-1}$, and no single base-pair inserts. tRNA nomenclature is from Sprinzl *et al.*[16] and accuracy is scored using the method of Mathews *et al.*[20] Only the four main helices shown by Sprinzl *et al.*[16] are scored.

[a] The accuracy of the structure found for RD4800 paired with RS1141 is 0 % for each sequence. When single base-pair inserts are allowed, the accuracy is instead 52.4 % for RD4800 and 40.0 % for RS1141.

Table 2. The percent accuracy of 5 S rRNA sequences folded with Dynalign

	Single sequence	*Haloferax volcanii*	*Arthrob. globiformis*	*Mycobacterium phlei*	*Sporolactobacillus plantarum*	*Porphyromonas gingivalis*	*Microbotrylum violaceum*	*Phleogena faginea*
Haloferax volcanii	29.0	-	93.5	93.5	48.4	83.9	93.5	93.5
Arthrob. globiformis	37.5	87.5	-	87.5	87.5	87.5	93.8	93.8
Mycobacterium phlei	100.0	90.3	90.3	-	100.0	87.1	100.0	100.0
Sporolactobacillus plantarum	25.8	41.9	80.6	90.3	-	74.2	90.3	83.9
Porphyromonas gingivalis	0.0	88.0	96.0	96.0	80.0	-	64.0	48.0
Microbotrylum violaceum	100.0	87.9	87.9	100.0	90.9	81.8	-	87.9
Phleogena faginea	42.4	90.9	97.0	97.0	81.8	90.9	90.9	-

The accuracy of each sequence folded alone is scored in the first column. Each other column gives the accuracy of secondary structure prediction by Dynalign of the sequence indicated by the row when paired with the sequence indicated at the top of the column. The calculations were performed with $M = 15$, $\Delta G^{\circ}_{gap} = 0.4$ kcal mol^{-1} (gap nt)$^{-1}$, and single base-pair inserts were allowed.

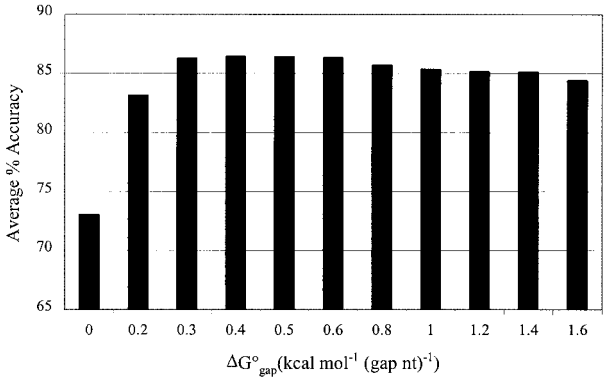


Figure 3. The average accuracy of secondary structure prediction for 5 S rRNA as a function of ΔG°_{gap}. The average accuracy for all the 5 S rRNAs folded individually by free energy minimization is 47.8 %.

secondary structure of each sequence (Figure 4(a)). The accuracy of structure prediction is improved drastically by the simultaneous folding by Dynalign (Figure 4(b) and Table 2).

Dynalign was also tested with two longer sequences, the 3′ UTR sequences of R2 elements from *Drosophila takahashii* and *Drosophila melanogaster*.[43,44] These RNA sequences have a secondary structure established by a limited comparative sequence analysis and supported by chemical mapping.[42] They are roughly 225 nucleotides long (see Table 3) and have many more gaps in the alignment as compared to tRNA and 5 S rRNA sequences.

The secondary structure of the R2 3′ UTR of *D. takahashii* is poorly predicted by free energy minimization (Figure 5(a)).[20] When paired with the R2 3′ UTR sequence from *D. melanogaster*, however, the secondary structure predicted by Dynalign for *D. takahashii* is nearly that predicted by comparative sequence analysis (Figure 5(b)).[42] Only two base-pairs from the comparative sequence analysis are absent in the structure predicted by Dynalign when the calculation is performed with single base-pair inserts prohibited and the ΔG°_{gap} of 0.4 kcal mol^{-1} (gap $nt)^{-1}$ determined by optimization of the set of 5 S rRNA sequences. When single base-pair inserts are allowed, the Dynalign-predicted *D. takahashii* secondary structure is also missing the two 5′ most helices from the comparative analysis structure.

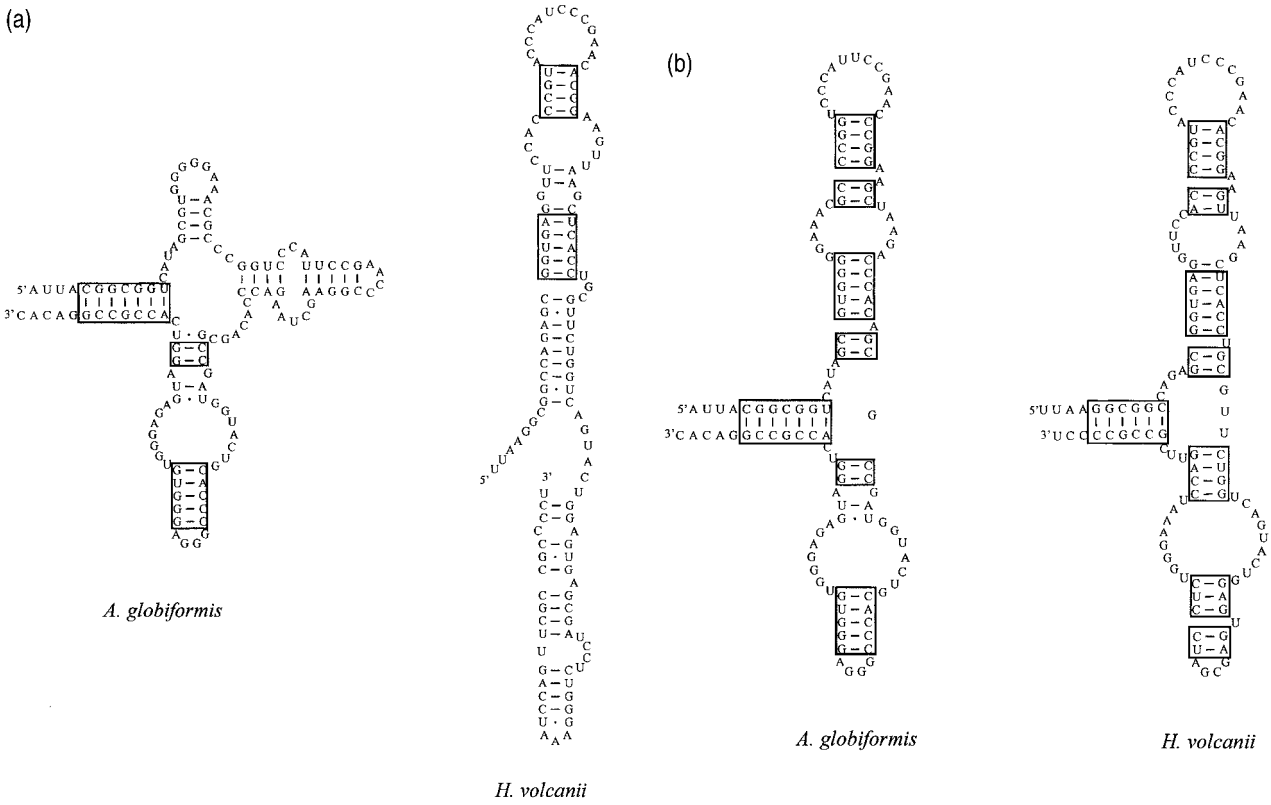


Figure 4. The secondary structures predicted for the 5 S rRNA sequences from *A. globiformis* and *H. volcanii*. (a) The secondary structures predicted by standard free energy minimization.[20] Neither structure has the three-way multibranch loop that is a hallmark of the 5 S rRNA secondary structure.[11] Boxed base-pairs are those found in the structure determined by comparative sequence analysis.[11] (b) The secondary structures predicted by Dynalign for both sequences considered simultaneously. The structures show the correct overall conformation about the central multibranch loop. The ΔG°_{37} for *A. globiformis* and *H. volcanii* are −44.4 and −34.2 kcal/mol, respectively. There are eleven gaps in the alignment for a total gap penalty of 4.4 kcal/mol for the final total score of −74.2 kcal/mol.

Table 3. Time and memory requirement (RAM) for Dynalign calculation on three sets of sequences

System	Sequence 1	Sequence 2	Length 1	Length 2	*M*	CPU time	RAM
tRNA	RD0260	RE6781	77	76	15	34 min 8 s	19 MB
5 S rRNA	*Haloferax volcanii*	*Arthrob. globiformis*	122	121	15	142 min 36 s	40 MB
R2 3′ UTR	*D. takahashii*	*D. melanogaster*	218	234	24	42 hr 45 min	252 MB

The calculations were performed on a personal computer running Red Hat Linux 7.0 with Dynalign compiled using GNU C++. The machine uses a Pentium III 600 MHz processor and 512 MB of RAM.

Speed of calculation

The Dynalign algorithm is $O(M^3N^3)$ where N is the length of the shorter sequence. There is no reason to believe that the maximum insertion size, M, will scale proportionally with the length of the sequence, N. Therefore, a doubling in sequence length requires an eightfold increase in calculation time for the same M. The memory use scales proportionally to M^2N^2 so that a doubling in sequence length requires a fourfold increase in storage. Table 3 lists the time and storage requirements for three systems with N ranging from 76 to 218.

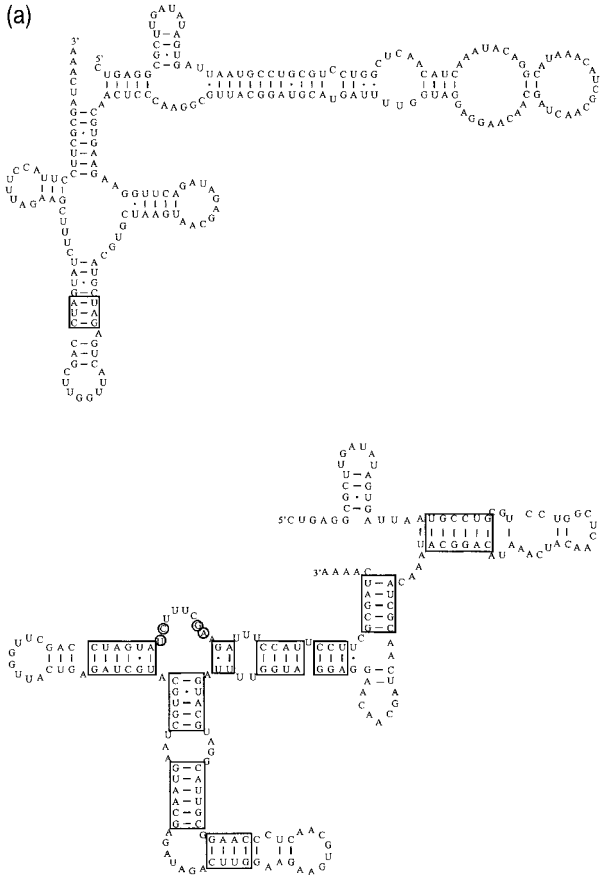


Figure 5. The secondary structure predicted for the R2 3′ UTR from *Drosophila takahashii*. (a) The secondary structure predicted by standard free energy minimization.[20] Boxed base-pairs are common to the secondary structure determined by comparative sequence analysis.[42] (b) The secondary structure predicted by Dynalign for *D. takahashii* determined in concert with the *D. melanogaster* R2 3′ UTR. Circled nucleotides indicate two positions for which base-pairs determined by comparative analysis[42] are not found by Dynalign. The calculation was performed with $M = 24$, gap penalty = 0.4 kcal mol^{-1} (gap nt)$^{-1}$, and no single base-pair inserts allowed.

Discussion

Determining RNA secondary structure is important for revealing structure-function relationships and designing oligonucleotides for antisense applications and gene chip arrays by identifying targetable regions and suggesting possible confounding structures.[45–48] The Dynalign algorithm takes advantage of both free energy minimization and comparative sequence analysis to predict RNA secondary structure. It can improve the accuracy of secondary structure prediction compared to standard free energy minimization methods that consider the structure of only one sequence.[20,22–24,49] Dynalign is directly applicable to many problems, such as determining the secondary structures of RNAs found by *in vitro* evolution. These are often short in length. RNA secondary structures predicted by Dynalign can also be a starting point for further refinement on the basis of chemical[50] and enzymatic mapping[51] or extensive comparative sequence analysis.[1] The explosion in genomic sequences should provide many other opportunities for sequence comparison.

Strengths and limitations of Dynalign

Dynalign uses the dynamic programming algorithm solution to comparative sequence analysis guided by conformational free energy minimization, as first suggested by Sankoff.[35] This solution has both strengths and weaknesses compared to other approaches. It requires no sequence homology because the identity of aligned nucleotides is not part of the scoring function and thus Dynalign is not confounded by compensating base changes. It simultaneously determines the common secondary structure and sequence alignment and therefore does not require a prior sequence alignment like many other approaches.[26–28,31,32] For two sequences, given gap penalty, and set of thermodynamic parameters,[20,36] the algorithm guarantees the optimal solution, unlike heuristic approaches. Dynalign, however, is limited in the length of the

sequences for which a structure can be determined because of the $O(M^3N^3)$ computational complexity. To keep the calculation tractable, Dynalign does not predict pseudoknots and is limited to comparisons of two sequences per calculation.

Prospects for extensions

Currently, secondary structure prediction by Dynalign on a standard personal computer is roughly limited to sequences shorter than 300 nucleotides. The rapid increase in the computational power of computers and the access to powerful computers provided by internet connections suggest that Dynalign calculations on longer sequences will be possible in the near future. Furthermore, the calculation lends itself to parallelization[52] so that multiprocessor computers or computer clusters could be used to tackle more computationally difficult problems with a parallelized version of Dynalign.

The program could be extended to find the structure common to more than two sequences. For example, the calculation performed for three sequences would be $O(M^6N^3)$ and memory use would be proportional to M^4N^2. Applied to three tRNA sequences, the calculation would currently take approximately 80 days and require 4 GB of RAM on a PIII 600 MHz machine. Thus more powerful computers will be necessary to allow inclusion of more than two sequences.

Alternatively, the program could be integrated into another system that determines the alignment for multiple sequences.[33] Some of the popular multiple sequence alignment programs build a multiple sequence alignment from many pairwise sequence alignments first determined by a dynamic programming algorithm.[53,54] Dynalign could be used to generate the pairwise alignments.

Dynalign does not currently predict pseudoknotted base-pairs. This allows the algorithm to be $O(M^3N^3)$, similar to other dynamic programming algorithms for secondary structure prediction.[41,55,56] Dynalign could be constructed to predict pseudoknots using a method such as that introduced by Rivas and Eddy.[25] In the best case, a second parameter, L, could be introduced to limit the distance over which pseudoknotted base-pairs could form in a sequence. This would provide a Dynalign algorithm with $O(M^6L^4N^2)$ and storage requirements of $M^4L^2N^2$. This is currently impractical for most sequences of interest, however.

Dynalign finds the lowest free energy solution for a structure common to two sequences. The algorithm could also be extended to produce suboptimal structures and alignments using methods similar to that used for prediction of suboptimal secondary structures[55] and for suboptimal sequence alignment.[57] This requires nearly a doubling in the storage requirement and calculation time, however, and so was not included in this study.

Comparison to genetic algorithm approach

Genetic algorithms[30] are simulations and therefore do not guarantee convergence to the optimal solution of conserved structure or even to the same structure for each simulation run. In contrast, the dynamic programming algorithm, Dynalign, is a calculation that guarantees the optimal solution given the framework of the problem. Furthermore, a genetic algorithm potentially requires a large amount of time to reach convergence. An advantage of the genetic algorithm is that it lends itself to considering more than two sequences per simulation. Dynalign can consider only two sequences at a time because of computational complexity.

These two approaches to solving the same problem are complementary. For example, a synthesis of both approaches may be helpful for comparative sequence analysis of many sequences. Dynalign could be used to construct a sequence alignment through many pairwise comparisons. This alignment could then be used as a starting point for further refinement by the genetic algorithm.

Availability

Dynalign is available in two forms by free download from the Turner laboratory website: http://rna.chem.rochester.edu. One form is the ANSI C++ code suitable for compilation on any platform. Dynalign has also been incorporated into the user-friendly RNAstructure package for secondary structure prediction in the Microsoft Windows environment.

Materials and Methods

Dynalign algorithm

Dynalign is a four-dimensional dynamic programming algorithm and as such the calculation is divided into two steps. The fill step calculates three arrays of free energies, $W(i,j,k,l)$, $V(i,j,k,l)$, and $W5(i,k)$. $W(i,j,k,l)$ is the sum of the minimum free energies for nucleotide fragments i to j from the first sequence and k to l from the second sequence with i aligned to k and j aligned to l plus any gap penalties for interior nucleotides in the sequence alignment. $V(i,j,k,l)$ is defined the same as $W(i,j,k,l)$, except that i and j are base-paired and k and l are base-paired. This is illustrated in Figure 6. $W5(i,k)$ is the sum of free energies of nucleotide fragments from 1 to i in the first sequence and 1 to k in the second sequence. $W5(N_1,N_2)$ is the lowest free energy sum for a structure common to both sequences where N_1 is the length of one sequence and N_2 is the length of the other sequence. These arrays are filled without explicitly determining the structural conformations that satisfy those minimal free energies. The traceback step utilizes the information in the arrays to find the structure common to the two sequences that has the lowest free energy sum.

The structure calculation is simplified computationally by restricting the distances between aligned nucleotides to within a parameter, M. This means that, for $W(i,j,k,l)$ and $V(i,j,k,l)$, $i - M \leqslant k \leqslant i + M$ and $j - M \leqslant l \leqslant j + M$. Furthermore, free energies are rounded to the nearest

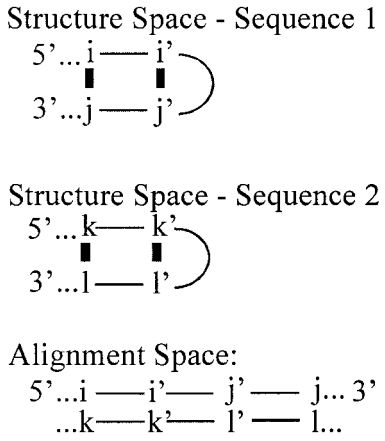


Figure 6. Illustration of conformations considered in the calculation of $V(i,j,k,l)$. $V(i,j,k,l)$ contains the optimal value of the sum of free energies of nucleotide fragment i to j in sequence 1 and fragment k to l in sequence 2. Base-pairs are required between nucleotides i and j and also between k and l as illustrated in the structure space of each sequence. Furthermore, i is aligned to k and j to l as illustrated in the alignment space. The term V_2 is the optimal free energy of the search through structures that contain interior base-pairs i', j', k', and l'. The positions of these nucleotides are indicated for both structure and alignment spaces.

tenth of a kcal/mol and are multiplied by ten for storage in the arrays. This allows integer math and therefore $V(i,j,k,l)$, $W(i,j,k,l)$, and $W5(i,k)$ are arrays of short integers. $V(i,j,k,l)$ and $W(i,j,k,l)$ are of size $(N)(N-1)(2M+1)^2$ and $W5(i,k)$ is of size $(N)(2M+1)^2$ where N is the length of the shorter of the two sequences.

$V(i,j,k,l)$ and $W(i,j,k,l)$ are filled by considering every 5-mer (the minimum sequence length that allows unimolecular secondary structure without base-pair distortion), 6-mer, 7-mer, and so forth up to a length of N. If $i-j$ or $k-l$ is a non-canonical base-pair, then $V(i,j,k,l)$ is set to a large positive free energy. If both $i-j$ and $k-l$ can form canonical base-pairs, i.e. A-U, G-C, or G-U, then $V(i,j,k,l)$ is the minimum of three terms, V_1, V_2, and V_3. V_1 considers hairpin loops closed by base-pairs $i-j$ and $k-l$:

$$V_1 = \Delta G^\circ_{\text{hairpin}}(i,j) + \Delta G^\circ_{\text{hairpin}}(k,l) + (\Delta G^\circ_{\text{gap}})|j-i-l+k|$$

V_2 is the lowest sum of free energies for a helix extension, bulge loop, or internal loop in the common structure.

V_2 requires a search through parameters i', j', k', and l' so that:

$$V_2 = \min[V(i',j',k',l') + \Delta G^\circ_{\text{motif 1}} + \Delta G^\circ_{\text{motif 2}}]$$

for: $i < i' < j' < j$ and $k < k' < l' < l$. Furthermore, $i' - M \leqslant k' \leqslant i' + M$ and $j' - M \leqslant l' \leqslant j' + M$. The search is restricted to the region of: $i < i' \leqslant i + S$, $j - S \leqslant j' < j$, $k < k' \leqslant k + S$, $l - S \leqslant l' < l$. For the tests of the algorithm in this work, $S = 20$. This is a commonly used technique to limit the depth of internal loop searching. $\Delta G^\circ_{\text{motif 1}}$ is the free energy of the motif closed by pairs $i-j$ and $i'-j'$ and $\Delta G^\circ_{\text{motif 2}}$ is similarly the energy for the motif in sequence 2. For $i' = i+1$ and $j' = j-1$, this is a continuation of a canonical helix. For either $i' = i+1$ or $j' = j-1$, but not both, this motif is a bulge loop. Otherwise, this is an internal loop.

V_3 is the lowest sum of free energies for a multibranch loop closed by pairs $i-j$ and $k-l$. V_3 requires a search through parameters i' and k', with $i < i' < j$ and $k < k' < l$, and is the minimum of 16 terms:

$$V_3 = \min[V_{3\text{-}1}, V_{3\text{-}2}, V_{3\text{-}3}V_{3\text{-}4}, V_{3\text{-}5}, V_{3\text{-}6}, V_{3\text{-}7}, V_{3\text{-}8}, V_{3\text{-}9}, V_{3\text{-}10}, V_{3\text{-}11}, V_{3\text{-}12}, V_{3\text{-}13}, V_{3\text{-}14}, V_{3\text{-}15}, V_{3\text{-}16}]$$

$$V_{3\text{-}1} = W(i+1,i',k+1,k') + W(i'+1,j-1,k'+1,l-1) + 2\Delta G^\circ_{\text{MBL closure}} + 2\Delta G^\circ_{\text{helix terminating in MBL loop}}$$

$$V_{3\text{-}2} = W(i+1,i',k+1,k') + W(i'+1,j-1,k'+1,l-2) + 2\Delta G^\circ_{\text{dangle l–1}} + \Delta G^\circ_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^\circ_{\text{MBL closure}} + 2\Delta G^\circ_{\text{helix terminating in MBL loop}} + (\Delta G^\circ_{\text{gap}})$$

$$V_{3\text{-}3} = W(i+1,i',k+2,k') + W(i'+1,j-1,k'+1,l-1) + \Delta G^\circ_{\text{dangle k+1}} + \Delta G^\circ_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^\circ_{\text{MBL closure}} + 2\Delta G^\circ_{\text{helix terminating in MBL loop}} + (\Delta G^\circ_{\text{gap}})$$

$$V_{3\text{-}4} = W(i+1,i',k+2,k') + W(i'+1,j-1,k'+1,l-2) + \Delta G^\circ_{\text{dangle k+1}} + \Delta G^\circ_{\text{dangle l–1}} + 2\Delta G^\circ_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^\circ_{\text{MBL closure}} + 2\Delta G^\circ_{\text{helix terminating in MBL loop}} + 2(\Delta G^\circ_{\text{gap}})$$

$$V_{3\text{-}5} = W(i+1,i',k+1,k') + W(i'+1,j-2,k'+1,l-1) + \Delta G^\circ_{\text{dangle j–1}} + \Delta G^\circ_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^\circ_{\text{MBL closure}} + 2\Delta G^\circ_{\text{helix terminating in MBL loop}} + (\Delta G^\circ_{\text{gap}})$$

$$V_{3\text{-}6} = W(i+1,i',k+1,k') + W(i'+1,j-2,k'+1,l-2) + \Delta G^\circ_{\text{dangle j–1}} + \Delta G^\circ_{\text{dangle l–1}} + 2\Delta G^\circ_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^\circ_{\text{MBL closure}} + 2\Delta G^\circ_{\text{helix terminating in MBL loop}}$$

$$V_{3\text{-}7} = W(i+1,i',k+2,k') + W(i'+1,j-2,k'+1,l-1) + \Delta G^\circ_{\text{dangle j–1}} + \Delta G^\circ_{\text{dangle k+1}} + 2\Delta G^\circ_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^\circ_{\text{MBL closure}} + 2\Delta G^\circ_{\text{helix terminating in MBL loop}} + 2(\Delta G^\circ_{\text{gap}})$$

$$V_{3\text{-}8} = W(i+1,i',k+2,k') + W(i'+1,j-2,k'+1,l-2) + \Delta G^{\circ}_{\text{dangle j}-1} + \Delta G^{\circ}_{\text{dangle k}+1} + \Delta G^{\circ}_{\text{dangle l}-1} + 3\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}} + (\Delta G^{\circ}_{\text{gap}})$$

$$V_{3\text{-}9} = W(i+2,i',k+1,k') + W(i'+1,j-1,k'+1,l-1) + \Delta G^{\circ}_{\text{dangle i}+1} + \Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}} + (\Delta G^{\circ}_{\text{gap}})$$

$$V_{3\text{-}10} = W(i+2,i',k+1,k') + W(i'+1,j-1,k'+1,l-2) + \Delta G^{\circ}_{\text{dangle i}+1} + \Delta G^{\circ}_{\text{dangle l}-1} + 2\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}} + 2(\Delta G^{\circ}_{\text{gap}})$$

$$V_{3\text{-}11} = W(i+2,i',k+2,k') + W(i'+1,j-1,k'+1,l-1) + \Delta G^{\circ}_{\text{dangle i}+1} + \Delta G^{\circ}_{\text{dangle k}+1} + 2\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}}$$

$$V_{3\text{-}12} = W(i+2,i',k+2,k') + W(i'+1,j-1,k'+1,l-2) + \Delta G^{\circ}_{\text{dangle j}-1} + \Delta G^{\circ}_{\text{dangle k}+1} + \Delta G^{\circ}_{\text{dangle l}-1} + 3\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}} + (\Delta G^{\circ}_{\text{gap}})$$

$$V_{3\text{-}13} = W(i+2,i',k+1,k') + W(i'+1,j-2,k'+1,l-1) + \Delta G^{\circ}_{\text{dangle i}+1} + \Delta G^{\circ}_{\text{dangle j}-1} + 2\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}} + 2(\Delta G^{\circ}_{\text{gap}})$$

$$V_{3\text{-}14} = W(i+2,i',k+1,k') + W(i'+1,j-2,k'+1,l-2) + \Delta G^{\circ}_{\text{dangle i}+1} + \Delta G^{\circ}_{\text{dangle j}-1} + \Delta G^{\circ}_{\text{dangle l}-1} + 3\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}} + (\Delta G^{\circ}_{\text{gap}})$$

$$V_{3\text{-}15} = W(i+2,i',k+2,k') + W(i'+1,j-2,k'+1,l-1) + \Delta G^{\circ}_{\text{dangle i}+1} + \Delta G^{\circ}_{\text{dangle j}-1} + \Delta G^{\circ}_{\text{dangle k}+1} + 3\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}} + (\Delta G^{\circ}_{\text{gap}})$$

$$V_{3\text{-}16} = W(i+2,i',k+2,k') + W(i'+1,j-2,k'+1,l-2) + \Delta G^{\circ}_{\text{dangle i}+1} + \Delta G^{\circ}_{\text{dangle j}-1} + \Delta G^{\circ}_{\text{dangle k}+1} + \Delta G^{\circ}_{\text{dangle l}-1} + 4\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}} + 2\Delta G^{\circ}_{\text{MBL closure}} + 2\Delta G^{\circ}_{\text{helix terminating in MBL loop}}$$

The 16 cases account for all combinations of whether or not $i+1$ and $j-1$ are dangling ends on the pair $i-j$ and whether $k+1$ or $l-1$ are dangling ends on the pair $k-l$.

$W(i,j,k,l)$ is the minimum of three terms, W_1, W_2, and W_3 with:

$$W_1 = W(a,b,c,d) + (x)(\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}}) + (y)(\Delta G^{\circ}_{\text{gap}})$$

$$W_2 = V(a,b,c,d) + (x)(\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}}) + 2\Delta G^{\circ}_{\text{MBL closure}} + (y)(\Delta G^{\circ}_{\text{gap}})$$

$$W_3 = \min[W(i,i',k,k') + W(i'+1,j,k'+1,l)]$$

W_1 is the case of adding unpaired nucleotides to a multibranch loop. Similarly to V_3, there are 16 combinations of adding nucleotides and so a is either i or $i+1$, b is either j or $j-1$, c is either k or $k+1$, and d is either l or $l-1$. The $\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}}$ is multiplied by the number of nucleotides added, x, and the gap penalty is multiplied by y, the number of nucleotides that are added in one sequence, but not added in the second sequence. For example, if a is $i+1$, b is $j-1$, c is k, and d is $l-1$, then the number of nucleotides being added to the loop is 3, with a single gap penalty because there is an insert at $i+1$.

W_2 accounts for helix termini. The terms a, b, c, d, x, and y are defined similarly to W_1. In this case, the variation in values for a, b, c, and d allows for dangling ends at helix termini and the favorable stability for each dangling end is added to the term.

W_3 accounts for bifurcations in the structure. This term is necessary for considering multibranch loops with more than three branching helices. A search is conducted through $i < i' < j$ and $k < k' < l$. The cost of the search is limited to $i' - M \leqslant k' \leqslant i' + M$.

$W5(i,k)$ is the minimum of four terms, $W5_1$, $W5_2$, $W5_3$, and $W5_4$, where:

$$W5_1 = W5(i-1,k) + \Delta G^{\circ}_{\text{gap}}$$

$$W5_2 = W5(i,k-1) + \Delta G^{\circ}_{\text{gap}}$$

$$W5_3 = W5(i-1,k-1) + \Delta G^{\circ}_{\text{gap}}$$

$$W5_4 = V(i',c,k',d) + W5(a,b) + (x)(\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}}) + 2\Delta G^{\circ}_{\text{MBL closure}} + (y)(\Delta G^{\circ}_{\text{gap}}) + \Delta G^{\circ}(\text{dangling ends})$$

where $i' < i$, $k' < k$, $a = i' - 1$ or $i' - 2$, $b = k' - 1$ or $k' - 2$, $c = i$ or $i-1$, and $d = k$ or $k-1$. As with W_2 and V_3, there are 16 possible combinations to allow for dangling ends on the helices closed by $V(i',c,k',d)$. $W5(0,x)$ and $W5(x,0)$ are predefined as $(x)(\Delta G^{\circ}_{\text{gap}})$. By counting gap penalties for gaps in the sequence alignment that are 5′

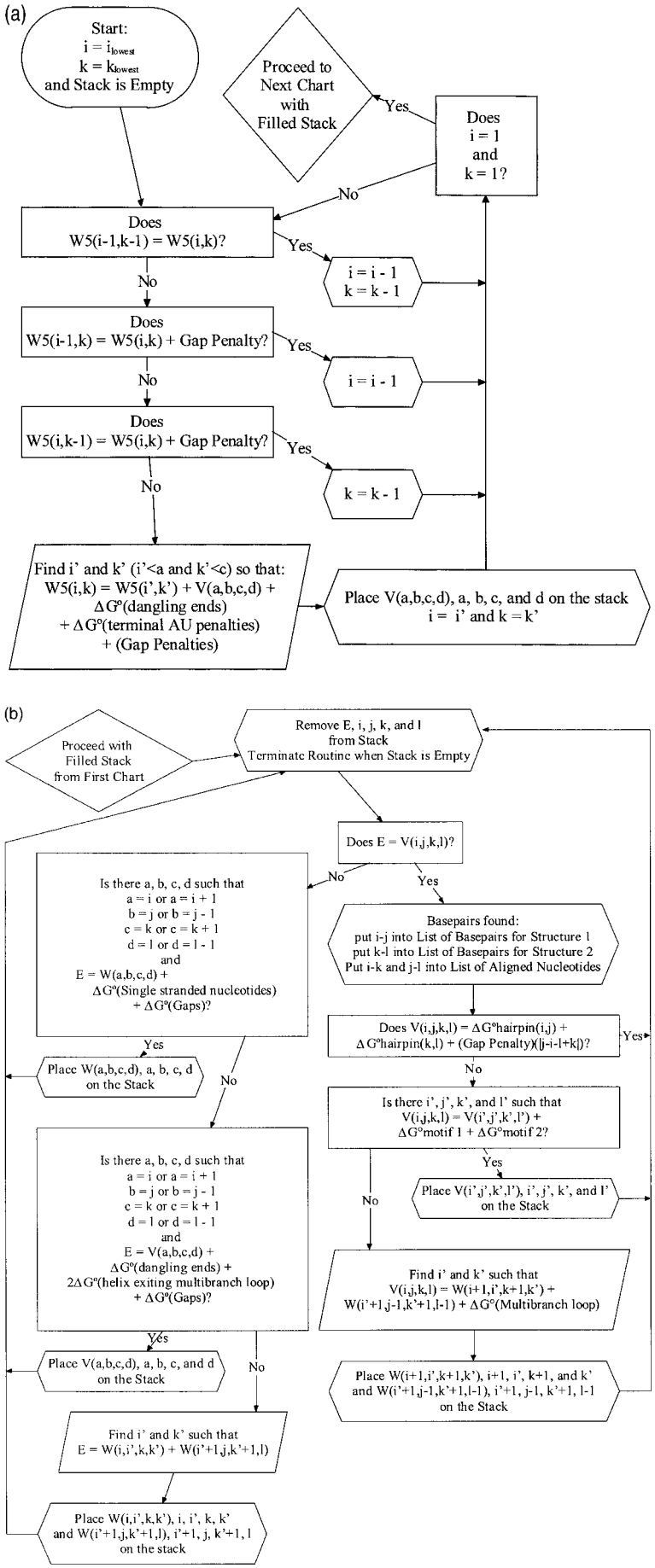


Figure 7. The traceback scheme flowchart. The algorithm starts in the upper left of (a) with i_{lowest} and k_{lowest} determined such that $W5(i_{lowest},k_{lowest})$ is the minimum value of $W5$. Chart (a) finds the exterior base-pairs (those base-pairs that close a domain) in the lowest free energy structure. Chart (b) then traces the base-pairs interior to the base-pairs found in (a). The stack contains E, i, j, k, and l. E is the sum of the free energies of the structure fragments from nucleotides i to j and k to l.

or 3′ to any base-pair, the algorithm is calculating the global alignment. A local alignment scheme could have been used by modifying the calculation of $W5(i,k)$.[58]

With $W(i,j,k,l)$, $V(i,j,k,l)$, and $W5(i,k)$ defined as above, a traceback scheme as illustrated in Figure 7 is used to find the structure that satisfies the optimal solution. This is the faster of the two steps of the calculation.

Nearest-neighbor parameters

The nearest-neighbor parameters for conformational free energy are derived from Mathews *et al.*[20] and Xia *et al.*[36] as reviewed by Turner[37] with the exception of multibranch loop initiation parameters, i.e. $\Delta G^{\circ}_{\text{unpaired nucleotide in MBL loop}}$, $\Delta G^{\circ}_{\text{MBL closure}}$, $\Delta G^{\circ}_{\text{helix terminating in MBL loop}}$, which are from earlier studies.[23,38] The optimized multibranch loop parameters utilized by Mathews *et al.*[20] are divided into a set used by a dynamic programming algorithm to generate a library of energetically reasonable structures and a set used to recalculate the free energies of these structures with the efn2 algorithm.[20,59] Because Dynalign does not have an efn2 step, a single set of parameters is required.

Acknowledgments

This work was supported by NIH grant GM22939. D.H.M. is a trainee in the medical scientist training program, NIH grant 5T32 GM07356

References

1. Pace, N. R., Thomas, B. C. & Woese, C. R. (1999). Probing RNA structure, function, and history by comparative analysis. In *The RNA World* (Gesterland, R. F., Cech, T. R. & Atkins, J. F., eds), 2nd edit., pp. 113-141, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York.
2. Cate, J. H., Gooding, A. R., Podell, E., Zhou, K., Golden, B. L., Kundrot, C. E. *et al.* (1996). Crystal structure of a group I ribozyme domain: principles of RNA packing. *Science,* **273**, 1678-1685.
3. Ban, N., Nissen, P., Hansen, J., Moore, P. B. & Steitz, T. A. (2000). The complete atomic structure of the large ribosomal subunit at 2.4 Å resolution. *Science,* **289**, 905-920.
4. Wimberly, B. T., Brodersen, D. E., Clemons, W. M., Morgan-Warren, R. J., Carter, A. P., Vonrhein, C. *et al.* (2000). Structure of the 30S ribosomal subunit. *Nature,* **407**, 327-329.
5. Schluenzen, F. A. T., Zarivach, R., Harms, J., Gluehmann, M., Janell, D., Bashan, A. *et al.* (2000). Structure of functionally activated small ribosomal subunit at 3.3 angstroms resolution. *Cell,* **102**, 615-623.
6. Kim, S. H., Suddath, F. L., Quiqley, G. J., McPherson, A., Sussman, J. L., Wang, A. H. J. *et al.* (1974). Three dimensional tertiary structure of yeast phenylalanine transfer RNA. *Science,* **185**, 435-440.
7. Robertus, J. D., Ladner, J. E., Finch, J. T., Rhodes, D., Brown, R. S., Clark, B. F. C. & Klug, A. (1974). Structure of yeast phenylalanine tRNA at 3 Å resolution. *Nature,* **250**, 546-551.
8. Yusupov, M. M., Yusupova, G. Z., Baucom, A., Lieberman, K., Earnest, T. N., Cate, J. H. D. & Noller, H. F. (2001). Crystal structure of the ribosome at 5.5 Å resolution. *Science,* **292**, 883-896.
9. Gutell, R. R. (1994). Collection of small subunit (16 S- and 16 S-like) ribosomal RNA structures. *Nucl. Acids Res.* **22**, 3502-3507.
10. Schnare, M. N., Damberger, S. H., Gray, M. W. & Gutell, R. R. (1996). Comprehensive comparison of structural characteristics in Eukaryotic cytoplasmic large subunit (23 S-like) ribosomal RNA. *J. Mol. Biol.* **256**, 701-719.
11. Szymanski, M., Specht, T., Barciszewska, M. Z., Barciszewski, J. & Erdmann, V. A. (1998). 5 S rRNA data bank. *Nucl. Acids Res.* **26**, 156-159.
12. Damberger, S. H. & Gutell, R. R. (1994). A comparative database of group I intron structures. *Nucl. Acids Res.* **22**, 3508-3510.
13. Michel, F., Umesono, K. & Ozeki, H. (1989). Comparative and functional anatomy of group II catalytic introns - a review. *Gene,* **82**, 5-30.
14. Brown, J. W. (1998). The ribonuclease P database. *Nucl. Acids Res.* **26**, 351-352.
15. Larsen, N., Samuelsson, T. & Zwieb, C. (1998). The signal recognition particle database (SRPDB). *Nucl. Acids Res.* **26**, 177-178.
16. Sprinzl, M., Horn, C., Brown, M., Ioudovitch, A. & Steinberg, S. (1998). Compilation of tRNA sequences and sequences of tRNA genes. *Nucl. Acids Res.* **26**, 148-153.
17. Chen, J. L., Blasco, M. A. & Greider, C. W. (2000). Secondary structure of vertebrate telomerase RNA. *Cell,* **100**, 503-514.
18. Romero, D. P. & Blackburn, E. H. (1991). A conserved secondary structure for telomerase RNA. *Cell,* **67**, 343-353.
19. Zwieb, C. & Wower, J. (2000). tmRDB (tmRNA database). *Nucl. Acids Res.* **28**, 169-170.
20. Mathews, D. H., Sabina, J., Zuker, M. & Turner, D. H. (1999). Expanded sequence dependence of thermodynamic parameters provides improved prediction of RNA secondary structure. *J. Mol. Biol.* **288**, 911-940.
21. Mathews, D. H., Diamond, J. M. & Turner, D. H. (2000). The application of thermodynamics to the modeling of RNA secondary structure. In *Thermodynamics in Biology* (Di Cera, E., ed.), pp. 177-201, Oxford University Press, Oxford.
22. Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M. & Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.* **125**, 167-168.
23. Gultyaev, A. P., van Batenburg, F. H. D. & Pleij, C. W. A. (1995). The computer simulation of RNA folding pathways using a genetic algorithm. *J. Mol. Biol.* **250**, 37-51.
24. Ding, Y. & Lawrence, C. E. (1999). A Bayesian statistical algorithm for RNA secondary structure prediction. *Comput. Chem.* **23**, 387-400.
25. Rivas, E. & Eddy, S. R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.* **285**, 2053-2068.
26. Juan, V. & Wilson, C. (1999). RNA Secondary structure prediction based on free energy and phylogenetic analysis. *J. Mol. Biol.* **289**, 935-947.
27. Lück, R., Steger, G. & Riesner, D. (1996). Thermodynamic prediction of conserved secondary structure: application to the RRE element of HIV, the tRNA-like element of CMV and the mRNA of prion protein. *J. Mol. Biol.* **258**, 813-826.

28. Lück, R., Gräf, S. & Steger, G. (1999). ConStruct: a tool for thermodynamic controlled prediction of conserved secondary structure. *Nucl. Acids Res.* **27**, 4208-4217.
29. Notredame, C., O'Brien, E. A. & Higgins, D. G. (1997). RAGA: RNA sequence alignment by genetic algorithm. *Nucl. Acids Res.* **25**, 4570-4580.
30. Chen, J., Le, S. & Maizel, J. V. (2000). Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucl. Acids Res.* **28**, 991-999.
31. Corpet, F. & Michot, B. (1994). RNAlign program - alignment of RNA sequences using both primary and secondary structures. *Comput. Appl. Biosci.* **10**, 389-399.
32. Hofacker, I. L., Fekete, M., Flamm, C., Huynen, M. A., Rauscher, S., Stolorz, P. E. & Stadler, P. F. (1998). Automatic detection of conserved RNA structure elements in complete RNA virus genomes. *Nucl. Acids Res.* **26**, 3825-3836.
33. Gorodkin, J., Heyer, L. J. & Stormo, G. D. (1997). Finding the most significant common sequence and structure in a set of RNA sequences. *Nucl. Acids Res.* **25**, 3724-3732.
34. Eddy, S. R. & Durbin, R. (1994). RNA sequence analysis using covariance models. *Nucl. Acids Res.* **22**, 2079-2088.
35. Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *Siam J. Appl. Math.* **45**, 810-825.
36. Xia, T., SantaLucia, J., Jr, Burkard, M. E., Kierzek, R., Schroeder, S. J. & Jiao, X., *et al*. (1998). Parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick pairs. *Biochemistry,* **37**, 14719-14735.
37. Turner, D. H. (2000). Conformational changes. In *Nucleic Acids* (Bloomfield, V. A., Crothers, D. M. & Tinoco, I., Jr, eds), pp. 259-334, University Science Books, Sausalito, CA.
38. Mathews, D. H., Andre, T. C., Kim, J., Turner, D. H. & Zuker, M. (1998). An updated recursive algorithm for RNA secondary structure prediction with improved thermodynamic parameters. In *Molecular Modeling of Nucleic Acids* (Leontis, N. B. & SantaLucia, J., Jr, ed.), pp. 246-257, American Chemical Society, Washington, DC.
39. Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443-453.
40. Sellers, P. H. (1974). On the theory and computation of evolutionary distances. *SIAM J. Appl. Math.* **26**, 787-793.
41. Zuker, M. & Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucl. Acids Res.* **9**, 133-148.
42. Mathews, D. H., Banerjee, A. R., Luan, D. D., Eickbush, T. H. & Turner, D. H. (1997). Secondary structure model of the RNA recognized by the reverse transcriptase from the R2 retrotransposable element. *RNA,* **3**, 1-16.
43. Jakubczak, J. L., Burke, W. D. & Eickbush, T. H. (1991). Retrotransposable elements R1 and R2 interrupt the rRNA genes of most insects. *Proc. Natl Acad. Sci. USA,* **88**, 3295-3299.
44. Eickbush, T. H. (1994). Origin and evolutionary relationships of retroelements. In *The Evolutionary Biology of Viruses* (Morse, S. S., ed.), Raven Press, New York.
45. Mathews, D. H., Burkard, M. E., Freier, S. M., Wyatt, J. R. & Turner, D. H. (1999). Predicting oligonucleotide affinity to nucleic acid targets. *RNA,* **5**, 1458-1469.
46. Rychlik, W. & Rhoads, R. E. (1989). A computer program for choosing oligonucleotides for filter hybridization, sequencing and *in vitro* amplification of DNA. *Nucl. Acids Res.* **17**, 8543-8551.
47. Walton, S. P., Stephanopoulos, G. N., Yarmush, M. L. & Roth, C. M. (1999). Prediction of antisense oligonucleotide binding affinity to a structured RNA target. *Biotechnol. Bioeng.* **65**, 1-9.
48. Schena, M. (1999). *DNA Microarrays. A Practical Approach. The Practical Approach Series* (Hames, B. D., ed.), Oxford University Press, New York.
49. Zuker, M., Mathews, D. H. & Turner, D. H. (1999). Algorithms and thermodynamics for RNA secondary structure prediction: a practical guide. In *RNA Biochemistry and Biotechnology* (Barciszewski, J. & Clark, B. F. C., eds), pp. 11-43, Kluwer Academic Publishers, Boston.
50. Ehresmann, C., Baudin, F., Mougel, M., Romby, P., Ebel, J. & Ehresmann, B. (1987). Probing the structure of RNAs in solution. *Nucl. Acids Res.* **15**, 9109-9128.
51. Knapp, G. (1989). Enzymatic approaches to probing RNA secondary and tertiary structure. *Methods Enzymol.* **180**, 192-212.
52. Shapiro, B. A., Chen, J., Busse, T., Navetta, J., Kasprzak, W. & Maizel, J. V. (1995). Optimization and performance analysis of a massively parallel dynamic programming algorithm for RNA secondary structure prediction. *Int. J. Supercomput. Appl.* **9**, 29-39.
53. Thompson, J. D., Higgins, D. G. & Gibson, T. J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucl. Acids Res.* **22**, 4673-4680.
54. Feng, D. & Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.* **25**, 351-360.
55. Zuker, M. (1989). On finding all suboptimal foldings of an RNA molecule. *Science,* **244**, 48-52.
56. McCaskill, J. S. (1990). The equilibrium partition function and base-pair probabilities for RNA secondary structure. *Biopolymers,* **29**, 1105-1119.
57. Zuker, M. (1991). Suboptimal sequence alignment in molecular biology. Alignment with error analysis. *J. Mol. Biol.* **221**, 403-420.
58. Smith, T. F. & Waterman, M. S. (1981). Comparison of bio-sequences. *Advan. Appl. Math.* **2**, 482-489.
59. Walter, A. E., Turner, D. H., Kim, J., Lyttle, M. H., Müller, P., Mathews, D. H. & Zuker, M. (1994). Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proc. Natl Acad. Sci. USA,* **91**, 9218-9222.

Edited by I. Tinoco

(Received 25 June 2001; received in revised form 20 November 2001; accepted 20 November 2001)