BIOINFORMATICS

Vol. 15 no. 6 1999
Pages 446–454

RNA secondary structure prediction using stochastic context-free grammars and evolutionary history

B. Knudsen and J. Hein

Department of Genetics and Ecology, The Institute of Biological Sciences, University of Aarhus, Building 550, Ny Munkegade, 8000 Aarhus C, Denmark

Received on December 22, 1998; revised and accepted on February 22, 1999

Abstract

Motivation: *Many computerized methods for RNA secondary structure prediction have been developed. Few of these methods, however, employ an evolutionary model, thus relevant information is often left out from the structure determination. This paper introduces a method which incorporates evolutionary history into RNA secondary structure prediction. The method reported here is based on stochastic context-free grammars (SCFGs) to give a prior probability distribution of structures.*

Results: *The phylogenetic tree relating the sequences can be found by maximum likelihood (ML) estimation from the model introduced here. The tree is shown to reveal information about the structure, due to mutation patterns. The inclusion of a prior distribution of RNA structures ensures good structure predictions even for a small number of related sequences. Prediction is carried out using maximum a posteriori estimation (MAP) estimation in a Bayesian approach. For small sequence sets, the method performs very well compared to current automated methods.*

Contact: *bk@imf.au.dk*

Introduction

Computerized methods have been used for RNA secondary structure prediction for a number of years (e.g. Nussinov *et al*., 1978; Zuker and Stiegler, 1981). During the last 10 years, further methods have been developed (e.g. Zuker, 1989; Eddy and Durbin, 1994; Sakakibara *et al*., 1994; Cary and Stormo, 1995; Tabaska *et al*., 1998). Some methods use single sequences, which take advantage of prior information on RNA structures, usually through energy functions, e.g. Zuker (1989). No knowledge concerning related sequences is used, so these methods are not ideal when estimating structures of sequences with known homologs.

Covariance methods (Eddy and Durbin, 1994) and profile stochastic context-free grammars (SCFGs) (Sakakibara *et al*., 1994), on the other hand, do use information from more than one sequence, but do not explicitly take phylogeny into account, and do not use a prior probability distribution of structures. Maximum weighted matching methods (Cary and Stormo, 1995; Tabaska *et al*., 1998) share these characteristics.

The method introduced here uses prior knowledge about RNA structure in making a maximum a posteriori (MAP) estimation of the secondary structure. This is performed on an alignment of sequences assumed to have identical secondary structures, i.e. the alignment is assumed to be a structural alignment. The method takes the phylogenetic tree of the sequences into account, including branch lengths, using a model of mutation processes in RNA. Furthermore, the tree can be estimated by a maximum likelihood (ML) method.

The idea for this work originates in work by Goldman *et al*. (1996), who developed a method for predicting protein secondary structure using hidden Markov models (HMMs) and including phylogenetic information. This method uses 20×20 rate matrices for amino acid replacements. Three matrices are employed: one for α-helices, one for β-sheets and one for coils (the rest). These matrices are estimated from sequences of known structure. An HMM with three states, corresponding to the structure types, models the structures along sequences. This HMM is then used in conjunction with the rate matrices to find the ML estimate of the tree relating sequences in an alignment and to predict their secondary structures. The method described here is an extension of this model to RNA secondary structure.

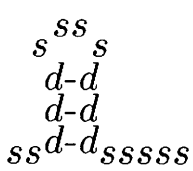
Secondary structures in RNA are not local, like in proteins, thus it is necessary to use a more complex model than an HMM for modelling these. SCFGs, which are used here, can describe some long-range interactions, including most of the ones in RNA secondary structure. SCFGs are unable to model crossing interactions, thus pseudoknots cannot be predicted by this method.

Algorithms

The input for this analysis is an alignment of RNA sequences, while the output is a single common structure for the sequences. The model consists of two distinct parts: the SCFG and the evolutionary model.

a) $S \rightarrow LS \rightarrow LLLLLLLS \rightarrow LLLLLLLL$
$\rightarrow ssLssssss \rightarrow ssdFdssssss$
$\rightarrow ssdddFddddssssss$
$\rightarrow ssdddLSddddssssss$
$\rightarrow ssdddLLLLddddssssss$
$\rightarrow ssddddssssddddssssss$

b)



c) $F \rightarrow dFd \rightarrow ddFdd \rightarrow ddLSdd$
$\rightarrow ddLLdd \rightarrow ddLsdd \rightarrow dddFdsdd$

Fig. 1. Production of RNA structures by the grammar. (**a**) The rules being used, starting from *S*. (**b**) The corresponding structure. (**c**) The production of a stem with a bulge.

The grammar

A grammar consists of a set of variables, some terminal and some non-terminal. Specifically, a starting non-terminal called *S* is contained in a grammar. The non-terminals are rewritten according to a set of production rules, which (for an SCFG) specifies a single non-terminal and a string of variables, that it should be changed to. Successive production rules are applied until only a string of terminals are left (Chomsky, 1959; Lari and Young, 1990).

The basis of the model is a simple SCFG with the following production rules:

$$S \rightarrow LS \mid L$$
$$F \rightarrow dFd \mid LS$$
$$L \rightarrow s \mid dFd$$

Here, *s* symbolizes a base in a single string and *d*s symbolize bases that pair up in a stem. The non-terminal *S* produces loops and *F* produces stems, while *L* decides whether a specific loop position should be a single base or the start of a new stem. An illustration is given in Figure 1. Stems can have any length, whereas loops have lengths of at least two positions, due to the fact that *F* produces *LS* instead of just *S*. Here, positions should be understood in the broad interpretation that the start of a new stem is also a position. This means that the two-position loops can either be two bases, one base and a new stem (a bulge), or two new stems (a bifurcation).

The probabilities of the production rules determine the prior distribution of secondary structures, in that each structure has a certain probability given by the SCFG. The SCFG production probabilities are estimated from a training set of folded RNAs.

Most literature on SCFGs assumes the grammar to be in Chomsky normal form for the algorithms to be used (Chomsky, 1959; Baker, 1979; Lari and Young, 1990). The algorithms are easily adapted to other forms, which has been done here. The number of computations needed for solving problems with these algorithms is proportional to the cube of the sequence length.

Probabilities of columns

First, we look at the columns of the alignment one at a time. Denote the number of sequences in the alignment by n, which gives the height of the columns. The probability of a column of non-pairing bases is assumed to be independent of the other columns, given the tree relating the sequences. Likewise, the probability of two pairing columns is assumed to be independent of any other columns, again given the tree.

Let $p = (p_A, p_U, p_G, p_C)$ be the distribution of bases in loop regions of RNA sequences. Furthermore, for $X \neq Y$, let r_{XY} denote the rate of mutation from base X to base Y, while r_{XX} is the negative of the rate by which base X mutates to other bases. The rate matrix can be written as:

$$R = \begin{bmatrix} r_{AA} & r_{AC} & r_{AG} & r_{AU} \\ r_{CA} & r_{CC} & r_{CG} & r_{CU} \\ r_{GA} & r_{GC} & r_{GG} & r_{GU} \\ r_{UA} & r_{UC} & r_{UG} & r_{UU} \end{bmatrix}$$

The rates are assumed to satisfy:

$$p_X r_{XY} = p_Y r_{YX} \text{ for } X \neq Y$$

which means that the overall flow from base X to Y equals the flow from base Y to X, i.e. reversibility of mutations. The r_{XX} values are calculated as

$$r_{XX} = -\sum_{Y \neq X} r_{XY}$$

This is known as a general reversible model (Tavaré, 1986). Given a tree, including branch lengths, the column probabilities are calculated using post-order traversal as described by Felsenstein (1981).

For base pairs, the probability of the two columns, given that they form a pair, is calculated using a similar rate matrix, except that base pairs are used instead of single bases. We thus have a distribution of the 16 base pairs and a 16×16 rate matrix, i.e. a general reversible model for base pairs. The reason for including all 16 base pairs is to make it possible to model rare non-standard base pairs. Parameterized rate matrices for base pairs were described by Muse (1994) and Schöniger and von Haeseler (1994). This way of looking at base pairs means that any

base pair change, like AU to GC, is regarded as a single mutation. Even very closely related sequences show these 'double' mutations. Pairs of the rRNAs described below, with sequence identity of 98% or more, were analysed (again as described below). This showed that the base pair mutations between them consisted of 22% 'double' mutations, justifying using a full 16 × 16 matrix for the mutation model. This makes it possible to exploit the differences in base distribution and mutation patterns between loops and stems to obtain good structure predictions.

If a gap is present in one of the sequences, it is handled by treating it as an unknown base, according to the overall base distribution in the model.

Probability of an alignment

Now the entire alignment is taken into consideration. The columns are numbered C_1, C_2, …, C_l, where l denotes the total length of the alignment. The input data, D, are then given as the ordered set of columns: $D = (C_1, C2, \ldots, C_l)$. By M, denote the model including the mutational model and the SCFG. Assuming that the tree is known and the model given, the probability of the alignment can be found. This is done by summing over all possible secondary structures, σ:

$$\begin{aligned} P(D|T,M) &= \sum_{\sigma} P(D,\sigma|T,M) \\ &= \sum_{\sigma} P(D|\sigma,T,M)P(\sigma|T,M) \\ &= \sum_{\sigma} P(D|\sigma,T,M)P(\sigma|M) \end{aligned}$$

The last equality stems from the fact that the secondary structure only is dependent on the tree through the data. The terms $P(\sigma|M)$ are probabilities of secondary structures, given the model. These are the prior probabilities from the grammar previously described.

The terms $P(D|\sigma, T, M)$, i.e. the alignment probabilities, given the secondary structure and the tree, are products of the column probabilities. This results from the assumption that columns which do not pair are independent:

$$\begin{aligned} P(D|\sigma,TM) &= P(C_1 \cdot \cdot \cdot C_n|\sigma,T,M) \\ &= \prod_{s} P(C_s|\sigma,T,M) \prod_{\mathrm{d}} P(C_d C_{d^c}|\sigma,T,M) \end{aligned}$$

The product over s is over the columns of single bases, while the product over d is over left columns of pairs, while the d^cs are the corresponding right columns of the pairs.

The sum can be calculated using a dynamical programming approach (Baker, 1979), by extending the view of the grammar described above to include productions of columns as follows. When an s is used in a production rule, it corresponds to a column in the alignment of sequences. Such a column has a probability, given the tree, which is multiplied to the production probability each time an s is produced. Likewise, probabilities for rules producing base pairs, like $F \rightarrow dFd$, are multiplied to the probability of the two columns, given that they form a pair. This makes the grammar equivalent to a grammar that generates columns in alignments instead of just secondary structure, meaning that for a two-sequence alignment, the production rule $L \rightarrow s$ covers the following rules:

$$L \rightarrow \begin{bmatrix} X \\ Y \end{bmatrix} \text{ for } X, Y \in \{\mathrm{A}, \mathrm{U}, \mathrm{G}, \mathrm{C}\}$$

with $\begin{bmatrix} X \\ Y \end{bmatrix}$ denoting a column with the base X in the first sequence and the base Y in the second sequence. Thus, for n aligned sequences a rule like $L \rightarrow s$ covers 4^n rules, while a rule like $F \rightarrow dFd$ covers 4^{2n} rules (some being unlikely, with rare base pairings).

The full model

If the phylogenetic tree relating the sequences is not given, it must be estimated from the model. For a given tree, T, $P(D|T, M)$ can be calculated as above. The ML estimate of the tree, given the model, can then be obtained by:

$$T^{ML} = \underset{T}{\operatorname{argmax}} \; P(D|T,M)$$

which can be found by using numerical optimization: given a tree topology, the branch lengths can be obtained by maximizing the probability of the alignment, $P(D|\text{topology}, M)$. This is a $2n - 3$ dimensional search for a maximum, which can be done using standard methods (e.g. Press *et al.*, 1992). Estimating tree topology can, for example, be done by an exhaustive search, a branch and bound method or a heuristic method (Swofford *et al.*, 1996). The choice will be highly dependent upon the number of sequences in the alignment, considering the fast rate of growth in the number of trees with respect to the number of sequences. The maximum likelihood estimate of the tree is used in the MAP estimation of the structure. It would be better to integrate over all trees during the structure determination, but the above described approach is simpler.

The alignment of sequences is the data to be used in the secondary structure estimation. To perform a MAP estimation, we need to maximize $P(\sigma|D, T, M)$, which means to find the most likely secondary structure, given what we know. Using Bayes theorem, while conditioning on T and M, we obtain:

$$\begin{aligned} P(\sigma|D,\ T,\ M) &= \frac{P(D|\sigma,T,M)P(\sigma|T,M)}{P(D|T,M)} \\ &= \frac{P(D|\sigma,T,M)P(\sigma|M)}{P(D|T,M)} \end{aligned}$$

$P(\sigma|M)$ is the prior distribution of structures given by the SCFG. $P(D|T, M)$ is independent of the structure, and thus constant over all structures. The MAP estimate of the structure is then given by:

$$\sigma^{MAP} = \underset{\sigma}{\text{argmax}} \; P(D|\sigma, T^{ML}, M)P(\sigma|M)$$

which is found using the CYK algorithm (Durbin *et al.*, 1998) on the extended grammar, producing alignments.

From the posterior secondary structure prediction, various questions regarding the structure can be answered, including the most probable overall secondary structure (MAP estimate), the certainty of the prediction in each position and probabilities of the pairing of specific bases.

Implementation

The model was estimated in a number of steps:

1. A suitable database of sequences with known structures was made.
2. Single base and base pair frequencies were estimated.
3. Mutation rates were estimated.
4. The grammar parameters were estimated.

The database

The database used for estimating this model should represent RNA secondary structures in general, because it is attempted here to model RNA structures as a whole. For this reason, the database should be composed of various types of RNA. tRNAs and large subunit ribosomal RNAs (LSU rRNAs) were chosen. These are publically available and have well-established structures. The database made here consists of RNA sequences along with their entire secondary structures.

The tRNAs are from the database by Sprinzl *et al.* (1998). Part one of this database contains 2146 aligned tRNA gene sequences with corresponding RNA structures. This database was reduced by removing sequences with unknown bases and the like. Furthermore, interior loops, having one unpaired base on each side, were changed into stems [e.g. structures like '(.(.....).)' were changed to '(((.....)))']. [Parenthesis notation is used for describing the structures in this article. Matching parentheses (or later, brackets) denote positions that form a pair.] This pairs the non-standard pairs that the structures imply, which are assumed to bond (this is sometimes true, sometimes not). Allowing for non-standard base pairs gives the algorithm more robustness towards sequencing and alignment errors. Before this operation, the database only contained AU, GC and GU base pairs. The revised database had 1968 tRNA sequences with corresponding secondary structures.

The LSU rRNAs are from a database by De Rijk *et al.* (1998), which contains 709 sequences. A reduction was performed as above, resulting in 305 remaining sequences. This database contains a number of non-standard base pairs.

The training was carried out with a weighting of the sequences to represent the two RNA families equally.

Frequencies

The single base frequencies were estimated from counts of the bases in the single base positions of the sequences. Overall base frequencies were also determined. Base pair frequencies were estimated by counting base pairs. Interestingly, tRNAs show more GC than CG base pairs, meaning that in GC/CG base pairs the G tend to be nearer the 5′ end of the RNA than the C. This might have to do with functional constraints on evolution. As this model aims to be general, unique characteristics of the training sequences should not be modelled. Therefore, to obtain equal frequencies of *XY* and *YX* base pairs, each occurrence of an *XY* base pair also counted as a *YX* base pair (the rarely occurring pairs of identical bases were counted twice).

The obtained frequencies are shown in Table 1, which shows that the overall base frequencies are approximately equal. In stems, there are a significant majority of GC/CG base pairs, which probably has to do with the high binding energy associated with this pair.

Table 1. Base frequencies, showing nearly equal overall distribution of bases, with a slight underrepresentation of Cs. Stems have high GC/CG base pair frequencies, while loops have low content of Cs and Gs. The lowest row shows the distribution of bases between loops and stems

Stem		Loop		Overall	
AU/UA	35.6%	A	36.4%	A	26.8%
GC/CG	53.4%	C	15.1%	C	21.4%
UG/GU	9.8%	G	21.2%	G	26.7%
Other	1.2%	U	27.3%	U	25.1%
Total:	52.6%	Total:	47.4%		

Mutation rates

For estimating mutation rates, a number of sequences from the above-described database were paired. All possible ordered pairs were made of sequences having at least 85% identical base sequences. The 85% limit makes it reasonable to assume that only single mutations, in the sense of the mutation mechanisms described above, have occurred between the sequences. The single base positions in these sequence pairs were examined and all differences between the sequences counted. Thus, if a given position had base X in one sequence and base Y in the other, the counters c_{XY} and c_{YX} were incremented. Columns, in the pairs, having a gap, were not used. For a given pair, P, define t_P as the time between

sequences, N_P as the number of columns in the two-sequence alignment and P_s as the probability of a base being in a single base position. Because of the single mutation assumption, we have for $X \neq Y$:

$$E(c_{XY}) \approx \sum_P P_s N_p (p_x t_p r_{XY} + P_Y t_p r_{YX})$$
$$\Rightarrow E(c_{XY}) \approx 2 \sum_P P_s N_P t_P p_x r_{XY}$$
$$\Rightarrow r_{XY} \approx \frac{c_{XY}}{2 p_X P_s \sum_P t_P N_P} = K \frac{c_{XY}}{p_X P_s}$$

where the sums over P are over all pairs. K is a constant that is independent of X and Y, implying that $r_{XY} \propto c_{XY}/(p_X P_s)$ for all bases X and Y with $X \neq Y$. To ensure equal weighting of information from different sequences, the count from pairs having the same first sequence was divided by the number of pairs having this first sequence. This should decrease the variance of the estimates and only affect the constant K, ensuring that we still have $r_{XY} \propto c_{XY}/(p_X P_s)$. The rates were normalized so that the total rate of mutations in single base positions was one, making the rate matrix uniquely determined. In the article by Goldman *et al.* (1996), the rates were found in a similar way, except that the constant K was divided out. This meant that they had to estimate amino acid frequencies from the rate matrix. In this work, it is viewed as essential to have the best possible estimates of base frequencies, thus the rate matrix is estimated using these.

Pairs were counted using symmetry, both in position and time. The counts were dealt with in a similar fashion as the single-base counters. The normalization was performed relative to the single base rates, which shows that the mutation rate, considered on a single base level, for stem regions is 0.90 times the rate for single bases.

The mutation rates for single bases are shown in Table 2. Variations between mutation rates are observed. It is obvious that transitions (A–G and T–C mutations in DNA) are more frequent than transversions (the rest), which agrees with established knowledge (e.g. Gojobori *et al.*, 1982).

Table 2. The entries, r_{XY}, for the loop rate matrix. Transitions are more frequent than transversions

$X\backslash Y$	A	C	G	U
A	–0.75	0.16	0.32	0.26
C	0.40	–1.57	0.24	0.93
G	0.55	0.17	–0.96	0.24
U	0.35	0.51	0.19	–1.05

Mutation rates for the most frequent base pairs are shown in Table 3. This table shows that the pair mutations requiring only a single base change are the most frequent, while mutations requiring two transversions are very rare. This is what should be expected. Table 4 shows the mutation rates for single bases in stem regions. This table shows that the transition/transversion ratio is higher in stem regions than in loop regions. This is because single transversions disrupt pairing, while transitions may conserve pairing (e.g. both A and G can pair with U).

Table 3. Some of the entries for the stem rate matrix. Only rates between the six most frequent base pairs are included

$X\backslash Y$	AU	UA	GC	CG	UG	GU
AU	–1.16	0.18	0.50	0.12	0.02	0.27
UA	0.18	–1.16	0.12	0.50	0.27	0.02
GC	0.33	0.08	–0.82	0.13	0.02	0.23
CG	0.08	0.33	0.13	–0.82	0.23	0.02
UG	0.08	1.00	0.10	1.26	–2.56	0.04
GU	1.00	0.08	1.26	0.10	0.04	–2.56

Table 4. Marginal rate matrix for stems. This matrix is similar to the above matrix for loops, except that this one was estimated from stem regions. Notice the high transition/transversion ratio relative to loops

$X\backslash Y$	A	C	G	U
A	–1.15	0.13	0.79	0.23
C	0.09	–0.84	0.16	0.59
G	0.45	0.13	–0.70	0.11
U	0.18	0.70	0.16	–1.03

Grammar parameters

The production probabilities of the grammar reflect the way secondary structures behave: loop lengths, stem lengths, bifurcations, etc. For estimating these probabilities, secondary structures from the database were used. This estimation can be done using the inside–outside algorithm (an expectation maximization procedure) on this training set of secondary structures (Baker, 1979; Lari and Young, 1990). In the case of the simple grammar described here, the number of times each rule is used is uniquely determined by the training set, meaning that only one iteration had to be performed. Furthermore, the counting was performed in a simple way, which made it possible to analyse the long LSU rRNA sequences. The production probabilities obtained were the following:

$$S \rightarrow LS\ (86.9\%) \mid L\ (13.1\%)$$
$$F \rightarrow dFd\ (78.8\%) \mid LS\ (21.2\%)$$
$$L \rightarrow s\ (89.5\%) \mid dFd\ (10.5\%)$$

Probabilities are written in parentheses.

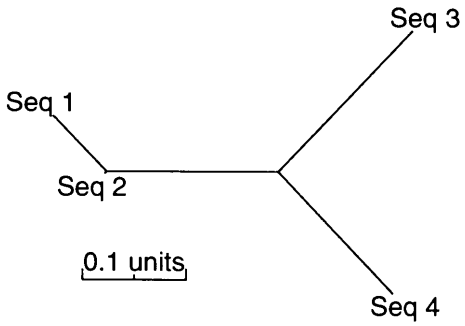


Fig. 2. The phylogenetic tree relating the four analysed sequences, as calculated using the ML estimation described above. The length units correspond to the rate matrices of the model.

Results

The test sequences

To test the method described here, four representative bacterial RNase P RNA sequences were chosen from the database by Brown (1998) and analysed:

Sequence 1	*Klebsiella pneumoniae*
Sequence 2	*Serratia marcescens*
Sequence 3	*Pseudomonas fluorescens*
Sequence 4	*Thiobacillus ferrooxidans*

The structures and alignment of the sequences are known. The sequences have lengths ranging from 344–383 bases, while their structural alignment has a total of 385 columns. The pairwise sequence identities range from 65–92%. The relationships between the sequences are shown in Figure 2, while the alignment is shown in Figure 3. The pseudoknot denoted by square brackets from positions 68–76 and 368–361 could be written using parentheses in these positions and square brackets in position 18–12 and 370–364. This is a stem of seven positions, while the other pseudoknot has four pairs. This means that a structure prediction of this type will have at least 22 positions wrongly predicted in each sequence.

Using related sequences

A number of predictions were made from the four RNase P RNA sequences by the method described above. The accuracy of a prediction is here defined as the total number of non-gap positions in each sequence having the correct assignment, divided by the total number of non-gap positions. A base pair is only considered correct if both base positions are correct. The alignments used were the structural alignments from the database by Brown (1998). Firstly, all sequences were analysed one by one, then all six pairs of sequences were used, then all four triples, and finally all the sequences were used. The results in the top of Table 5 show very significant improvement of prediction accuracy when sequences are added, especially going from one to two sequences. This exemplifies the large potential of methods using several sequences and their phylogeny, in making RNA secondary structure predictions. The pseudoknot stems, denoted by brackets in Figure 3, are invariant throughout these four sequences, which makes them hard to predict when using mutational patterns.

Table 5. Structural alignment, no phylogeny

	Structural alignment			
No. of sequences	1	2	3	4
Min result	41.2%	65.2%	73.9%	79.2%
Max result	57.7%	82.1%	79.6%	79.2%
Average	48.3%	73.6%	77.8%	79.2%
	CLUSTAL W alignment			
No. of sequences	1	2	3	4
Min result	41.2%	54.9%	60.1%	73.8%
Max result	57.7%	69.1%	76.9%	73.8%
Average	48.3%	64.4%	68.5%	73.8%
	Structural alignment, no phylogeny			
No. of sequences	1	2	3	4
Min result	41.2%	59.9%	67.7%	76.2%
Max result	57.7%	76.6%	76.6%	76.2%
Average	48.3%	68.9%	72.2%	76.2%

Table 6. What happens when a limit of certainty is imposed on the results. Each row shows how many positions have a certainty above a given limit and how many of these are correctly predicted. There is a high correlation between the accuracy of prediction and the certainty that the model predicts

Limit	No. of positions	Correct positions	Accuracy
0%	1459	1156	79.2%
50%	1314	1146	87.2%
70%	1150	1064	92.5%
80%	1068	1014	94.9%
90%	932	890	95.5%
95%	825	799	96.8%

In many situations, the structural alignment is not available. Therefore, it is necessary to assess the results using an alignment algorithm. For this, the same analyses as above were made, but each subset of the four sequences was aligned using CLUSTAL W (Thompson *et al.*, 1994). The results of this are shown in the middle of Table 5 with the column of one sequence identical to the earlier analysis. This gave lower accuracies than using structural alignments, which is not surprising. The rise in accuracy, when using more sequences, now arises both from better alignments and more data. Good results are still obtained using four sequences.

Neglecting phylogeny

If the phylogeny of the sequences is not taken into account, some information is lost and poorer prediction results. Such results are shown at the bottom of Table 5. This table was made like the top of Table 5, but using long branch lengths to simulate independent sequences. For two and three sequences, the phylogenetic information improves the result by ~5%. This shows that the tree conveys information about the structure. Results are compared in Figure 4.

Weight of results

The algorithm allows for a calculation of the probability that each position is correctly predicted. This is done using the inside and outside variables. It can give the user of the method an impression of how certain the predictions are, assuming that the model is correct. This can be considered as an equivalent to the partition function for energy calculations (McCaskill, 1990).

Taking the analysis of the structural alignment of all four sequences with a phylogenetic tree as an example, results from choosing only to believe regions of high certainty are shown in Table 6. This shows that discarding, for example, positions having a certainty of <70% means that 309 positions are discarded, of which only 92 were correctly predicted. This results in an accuracy of prediction for the remaining positions of 92.5%. This, of course, does not improve overall accuracy, but shows that badly predicted areas can be pointed out. Other methods, perhaps experimental, can then be used for these areas.

Comparison with other methods

To give an impression of the performance of this method relative to other methods, some comparisons have been made. The folding program, MFOLD Version 3.0 (Web server URL: http://www.ibc.wustl.edu/~zuker/rna/form1.cgi), by Zuker (1989) and Walter *et al.* (1994), using energy minimization was used for folding the four sequences one by one. Standard parameters were used, resulting in predictions ranging from 36 to 68%, with an average of 51% (see Table 7). This is comparable to the above-described method applied to single sequences, but does not suggest that this method is always as good as Zuker's for single sequences. The energy minimization method has more parameters than the above-described model, in the case of one sequence, where evolution does not come into consideration. This gives Zuker's method a potential for better results. Varying the parameters for the method might improve results; furthermore, results will be different for different families of RNA.

The method of maximum weighted matching was used on the four sequences, with the structural alignment. The scoring schemes used here are the ones described by Tabaska *et al.* (1998). Both helix-plot and mutual information were in-

```
     1                                                                                                                  100
1:   GAAGCUGACC AGACAGUCGC CGCUUCGUCG UCGUCCUCCU UCGGGGGGAG ACGGGCGGAG GGGAGGAAAG UCCGGGCUCC AUAGGGCAAG GUGCCAGGUA
2:   GGAGUUGACC AGACAGUCGC CGCUUCAUUG CCGUCCUC-U UCG-GGGGAG ACAGAUGGAG GGGAGGAAAG UCCGGGCUCC AUAGGGCAGG GUGCCAGGUA
3:   AGAGUCGAUU GGACAGUCGC UGCCCUCUAU ---------G AAA------- -AUUAGGGGG GGGAGGAAAG UCCGGGCUCC AUAGGGCGAA GUGCCAGGUA
4:   GGAGUGGGCC AGGCGACCGC CGCGGA---- ---------G CAA------- ------UCCG GGGAGGAAAG UCCGGGCUCC AUAGGGCAAG GCGCCGGUUA
s:   (((((((((( (((((((.( (.(((((((( {(.((((((. ..)))))))) }))).)))))) .)).....[[[ .[[[[[[((( ...[[[[.{( ((((((((..
p:   (((((((((( .(((((((((( ..(((((((( (((((((((. ..)))))))) )))).)))))) ....(((... )))..((((( ........(( ((((((((..

     101                                                                                                                200
1:   ACGCCUGGGG GGUGUCACGA CCCACGACCA GUGCAACAGA GAGCAAACCG CCGA-UGGCC CGCGCAAGCG GGAUCA-GGU AAGGGUGAAA GGGUGCGGUA
2:   ACGCCUGGGA GGC-GCAA-G CCUACGACUA GUGCAACAGA GAGCAAACCG CCGA-UGGCC CGCGCAAGCG GGAUCA-GGU AAGGGUGAAA GGGUGCGGUA
3:   AUGCCUGGGG GGC-GUGA-G CCUACGGAAA GUGCCACAGA AAAUA-ACCG CCUAAGCAC- ---UUCG--- -G-UGCCGGU AAGGGUGAAA AGGUGCGGUA
4:   ACGGCCGGGG GGC-GUGA-G CCUACGGAAA GUGCCACAGA AAAUAUACCG CCAA-GCGC- ---GUAA--- -G-CGC-GGU AAGGGUGAAA AGGUGCGGUA
s:   ...))))))( (((......) )))(.((... ((........ .......((( ((...((((( (((....))) )).))).))) ..)).....  .((((((...
p:   ...))))).( (((......) ))).....  .......... ......(((( .....((((( (((....))) )).)))))))  ..((((.... .((((((...

     201                                                                                                                300
1:   AGAGCGCACC GCGCGGCUGG UAACAGUCCG CGGCACGGUA AACUCCACCC GGAGCAAGGC CAAAUAGGGG UUCAUAAGGU ACGGCCCGUA CUGAACCCGG
2:   AGAGCGCACC GCGCGGCUGG UAACAGUUCG UGGCACGGUA AACUCCACCC GGAGCAAGGC CAAAUAGGGG UUCACAUGGU ACGGCCCGUA CUGAACCCGG
3:   AGAGCGCACC GCACGACUGG CAACAGUUCG UGGCUAGGUA AACCCCACUU GGAGCAAGAC CAAAUAGGGU UCCA--AGGC GUGGCCCGCG CUGGAACCGG
4:   AGAGCGCACC GCAUUUCCGG UAACGG-AAA UGGCAGGGAA AACCCCGCCU GGAGCAAGAC CAAAUAGGCG UGCGA-UACC GUGGCCCGCG GUGCACGCGG
s:   ....))))))  .(((((((((. ...))).))) )).........  .)))))))))} ))))...((( (......((( (((....((( ((.]]]]))))  )))))))))..
p:   ....))))))  .(((((((((. ...))).))) )).........  .)))))))))  ))))...(( (......((( ((((....(( (((...)))) )))))))))..

     301                                                                                                  385
1:   GUAGGCUGCU UGAGCCAGUG AGCGAUUGCU GGCCUAGAUG AAUGACUGUC CACGACAGAA CCCGGCUUAU CGGUCAGUUU CACCU
2:   GUAGGCUGCU UGAGCCAGUG AGCGAUUGCU GGCCUAGAGG AAUGACUGUC CACGACAGAA CCCGGCUUAU CGGUCAACUC CCUC-
3:   GUAGGUUGCU AAAGAUGUCC AGUGAUGGCC AUCGUAGACG AAUGACUGUU CAAGACAGAA CCCGGCUUAU AGAUCGACUC UCCAC
4:   GUAGGUUGCU GGAGCCUGUG CGUAAGUGCA GGCCUAGAGG AAUGGUCGUC CACGACAGAA CCCGGCUUAU CGGCCCACUC CAAUU
s:   ...))))...  ...((((.{{ (....)))}) )))..... ...)))))))  .........  ]]]]]]]].)  .)))))))))  ).....
p:   ...))).)).  ...(((((((  (....)))))  )))..... ...)))))))  .........  .........  .)))))))))  ).....
```

Fig. 3. The alignment of the four RNase P RNA sequences. The predicted structure, using all four sequences, is denoted p. The structure from the database is denoted s, with square brackets denoting parts of pseudoknots. The square brackets used here match the structure description in the database. The curly brackets denote positions where the structure differs: the sequences that have a non-standard pair in these positions have loop regions or bulges, the rest have pairs.

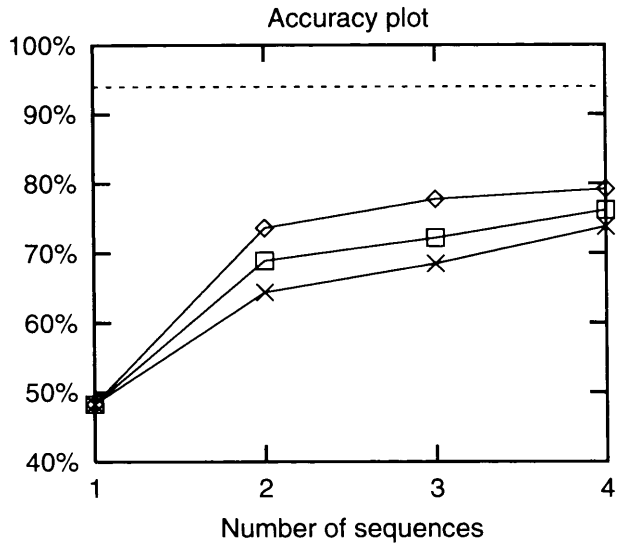


Fig. 4. A comparison of results with and without phylogeny. Diamonds (◇) denote the curve for predictions with phylogeny, while boxes (□) denote the one without. Crosses (×) denote results using CLUSTAL W alignments and phylogeny estimation. The dotted line at 94% represents the maximum possible prediction accuracy with regard to the pseudoknots.

corporated, giving a maximum of 60% accuracy. The covariance method, COVE Version 2.4.4., by Eddy and Durbin (1994), was also tried on the sequences, with lower accuracy. These methods were developed for larger numbers of sequences, and should not be expected to give optimal results using only four sequences. This shows the significance of the method described here in situations where only a few sequences are known.

Table 7. Accuracy table, showing comparisons of single sequence predictions using the method described in this paper and MFOLD Version 3.0, by Zuker (1989) and Walter *et al.* (1994). Predictions of secondary structures were made on single sequences, which is the only possibility using MFOLD. The average results are comparable

Sequence	SCFG method	MFOLD
Seq 1	57.7%	67.1%
Seq 2	48.2%	54.0%
Seq 3	41.2%	35.6%
Seq 4	46.2%	50.3%
Average	48.3%	51.7%

Conclusion

The limitations of this method include:

- Inability to predict pseudoknots.
- Loop and stem lengths are assumed to be geometrically distributed.
- A good alignment is needed.
- The dynamical programming algorithms are relatively slow. [They have a time complexity of $O(N^3)$ with respect to the length of the alignment.]

The problem with pseudoknots is shared by many algorithms (e.g. Zuker, 1989; Eddy and Durbin, 1994), although some algorithms can predict pseudoknots (e.g. Tabaska *et al.*, 1998).

The problem relating to the length distributions has to do with the nature of the specific SCFG used here, and can be solved by making different non-terminals producing stems or loops of various lengths. Special non-terminals describing small bulges will probably improve results. This introduces some extra computation time, but can definitely be carried out.

The problem of the alignment is not easily solved, because making an alignment without knowing the structure is unlikely to produce a structural alignment. It might be possible to realign sequences once a structure prediction has been made. This approach would probably be prone to local maxima in the likelihood function for alignments. One possible way of avoiding this would be to use Gibbs sampling in a Markov chain Monte Carlo method, sampling from alignments and summing over structures (Gilks *et al.*, 1996).

An alignment method which simultaneously folds and aligns a set of RNA sequences to find common structural elements locally has been implemented by Gorodkin *et al.* (1997). The algorithm has a computational complexity of $O(N^4)$, relative to the sequence length. The method has proven useful for relatively short sequences, and an alignment produced by such a method would be a good starting point for SCFG methods (Gorodkin *et al.*, 1997), including the one described here.

Profile SCFGs and covariance models predict secondary structure at the same time as making alignments, but seem to need a large number of sequences (Eddy and Durbin, 1994; Sakakibara *et al.*, 1994). Further work in making algorithms for simultaneous RNA folding and alignment will probably show up in the future because of the importance of solving this problem.

Further improvements to the method introduced here include modelling base stacking, which is not very difficult. It consists of conditioning the probability of two pairing columns on the neighbouring columns. Thus, in estimating the model, neighbour base pairs should be counted to indicate the conditional distributions of base pairs. This information could then be used in the calculations to give improved results.

Finally, it would be interesting to look into the evolutionary model proposed here. Statistical tests of its ability to describe RNA evolution would be enlightening. It would also

be useful to reduce the number of parameters for the rate matrices, especially the base pair rate matrix, e.g. as done by Muse (1994).

It is the hope of the authors that this method can be made available to the public via the World Wide Web.

Acknowledgements

We would like to thank Jan Gorodkin, Carsten Wiuf and the anonymous reviewers for critically reviewing the manuscript and suggesting improvements. Furthermore, J.H. acknowledges generous support by the Newton Institute for Mathematical Sciences in Cambridge, UK.

References

Baker,J.K. (1979) Trainable grammars for speech recognition. In Klatt,D.H. and Wolf,J.J. (eds), *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*. Acoustal Society of America, New York.

Brown,J.W. (1998) The ribonuclease P database. *Nucleic Acids Res*., **26**, 351–352.

Cary,R.B. and Stormo,G.D. (1995) Graph-theoretic approach to RNA modeling using comparative data. In Rawlings,C., Clark,D., Altman,R., Hunter,L., Lengauer,T. and Wodak,S. (eds), *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology.*

Chomsky,N. (1959) On certain formal properties of grammars. *Info. Control*, **2**, 137–167.

De Rijk,P., Caers,A., Van de Peer,Y. and De Wachter,R. (1998) Database on the structure of large ribosomal subunit RNA. *Nucleic Acids Res*., **26**, 183–186.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.

Eddy,S.R. and Durbin,R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res*., **22**, 2079–2088.

Felsenstein,J. (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol*., **17**, 368–376.

Gilks,W.R., Richardson,S. and Spiegelhalter,D.J. (1996) *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.

Gojobori,T., Wen-Hsiung,L. and Graur,D. (1982) Patterns of nucleotide substitution in pseudogenes and functional genes. *J. Mol. Evol*., **18**, 360–369.

Goldman,N., Thorne,J.L. and Jones,D.T. (1996) Using evolutionary trees in protein secondary structure prediction and other comparative sequence analyses. *J. Mol. Biol*., **263**, 196–208.

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

Lari,K. and Young,S.J. (1990) The estimation of stochastic context-free grammars using the inside-outside algorithm. *Comput. Speech Lang*., **4**, 35–56.

McCaskill,J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

Muse,S.V. (1994) Evolutionary analyses of DNA sequences subject to constraints on secondary structure. *Genetics*, **139**, 1429–1439.

Nussinov,R., Pieczenik,G., Griggs,J.R. and Kleitman,D.J. (1978) Algorithms for loop matchings. *SIAM J. Appl. Math*., **35**, 68–82.

Press,W.H., Teukolsky,S.A., Vetterling,W.T. and Flannery,B.P. (1992) *Numerical Recipes in C*, 2nd edn. Cambridge University Press, Cambridge.

Sakakibara,Y., Brown,M., Hughey,R., Mian,I.S., Sjölander,K., Underwood,R.C. and Haussler,D. (1994) Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Res*., **22**, 5112–5120.

Schöniger,M. and von Haeseler,A. (1994) A stochastic model for the evolution of autocorrelated DNA sequences. *Mol. Phylogenet. Evol*., **3**, 240–247.

Sprinzl,M., Horn,C., Brown,M., Ioudovitch,A. and Steinberg,S. (1998) Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res*., **26**, 148–153.

Swofford,D.L., Olsen,G.J., Waddell,P.J. and Hillis,D.M. (1996) Phylogenetic inference. In Hillis,D.M. and Moritz,C. (eds), *Molecular Systematics*, 2nd edn. Sinauer Associates, pp. 407–514.

Tabaska,J.E., Cary,R.B., Gabow,H.N. and Stormo,G.D. (1998) An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics*, **14**, 691–699.

Tavaré,S. (1986) Some probabilistic and statistical problems in the analysis of DNA sequences. In *Lectures on Mathematics in the Life Sciences*. American Mathematical Society, Vol. 17, pp. 57–86.

Thompson,J.D., Higgins,D.G. and 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. *Nucleic Acids Res*., **22**, 4673–4680.

Walter,A.E., Turner,D.H., Kim,J., Lyttle,M.H., Mueller,P., Mathews,D.H. and 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.

Zuker,M. (1989) On finding all suboptimal foldings of an RNA molecule. *Science*, **244**, 48–52.

Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamic and auxiliary information. *Nucleic Acids Res*., **9**, 133–148.