The language of RNA: a formal grammar that includes pseudoknots

Elena Rivas and Sean R. Eddy*

Department of Genetics, Washington University, St. Louis, MO 63110, USA

Abstract

Motivation: *In a previous paper, we presented a polynomial time dynamic programming algorithm for predicting optimal RNA secondary structure including pseudoknots. However, a formal grammatical representation for RNA secondary structure with pseudoknots was still lacking.*

Results: *Here we show a one-to-one correspondence between that algorithm and a formal transformational grammar. This grammar class encompasses the context-free grammars and goes beyond to generate pseudoknotted structures. The pseudoknot grammar avoids the use of general context-sensitive rules by introducing a small number of auxiliary symbols used to reorder the strings generated by an otherwise context-free grammar. This formal representation of the residue correlations in RNA structure is important because it means we can build full probabilistic models of RNA secondary structure, including pseudoknots, and use them to optimally parse sequences in polynomial time.*

Contact: *eddy@genetics.wustl.edu*

Introduction

Biological sequence analysis algorithms treat DNA, RNA, and protein molecules as strings of nucleotide or amino acid residues. Most biological sequence analysis algorithms further assume *uncorrelated* strings of residues, in which the identity of a residue at one position has no effect on the identity of another residue. This second assumption breaks down most dramatically in RNA sequence analysis. RNA secondary structure significantly constrains RNA sequence, because it produces strong long-distance pairwise correlations between Watson–Crick base pairs. RNA secondary structure prediction algorithms (Zuker and Stiegler, 1981) and RNA sequence/structure alignment algorithms (e.g. recognizing homologues of a known RNA structure in a sequence database) (Eddy and Durbin, 1994; Sakakibara *et al.*, 1994; Lefebvre, 1996) are applications which must adopt models that deal with long-distance pairwise correlations between residues.

Computational linguistics is a rich source of ideas for how to model strings with correlated symbols (Searls, 1992). A central concept is the Chomsky hierarchy of formal transformational grammars (Chomsky, 1959). The assumptions of most biological sequence algorithms correspond to those of the *regular grammars*, the lowest level in the Chomsky hierarchy. In order of increasing power to describe higher-order correlations, the other levels of the Chomsky hierarchy are the *context-free*, the *context-sensitive*, and the *unrestricted* grammars. Polynomial time algorithms are available to optimally parse (e.g. align, or recognize) strings generated by the regular grammars or the context-free grammars. In contrast, parsing context-sensitive languages is NP-complete, and there is no parser for unrestricted languages that is guaranteed to halt. There is thus a line in the sand somewhere between the context-free and the context-sensitive languages; beyond it, efficient, optimal general parsing algorithms are not available.

From a computational linguistics standpoint, the language of RNA is dominated by *nested* pairwise correlations (as in RNA stem-loops, Figure 1) which are easily generated by the context-free grammars (Searls, 1992). Stochastic context-free grammars have explicitly been used to create probabilistic models that describe RNA secondary structure (Sakakibara *et al.*, 1994; Eddy and Durbin, 1994). The popular MFOLD RNA secondary structure prediction algorithm (Zuker and Stiegler, 1981) seeks minimal energy structures using thermodynamic parameters, rather than seeking maximum likelihood structures using probabilities, but the dynamic programming algorithm that MFOLD uses is essentially identical to the CYK algorithm used to parse sequences with stochastic context-free grammars.

However, the language of RNA also includes important non-nested (crossed) correlations known as pseudoknots (Figure 1). RNA pseudoknots are functionally important in several known RNAs (ten Dam *et al.*, 1992). For example, by comparative analysis, RNA pseudoknots are conserved in ribosomal RNAs (Zimmerman and Dahlberg, 1996), the catalytic core of group I introns, and RNase P RNAs (Cech, 1993). Pseudoknotted structures have been
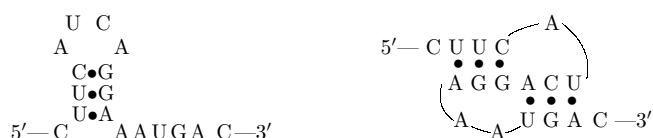

Fig. 1. Left: a stem-loop composed of three nested interactions. Right: A simple pseudoknot constructed from the previous nested stem-loop by introducing additional correlations between nucleotides inside the hairpin loop and nucleotides outside the original stem-loop.

observed by structure determination of the $3'$ end of plant viral RNAs (Kolk *et al.*, 1998), where pseudoknots are apparently used to mimic tRNA structure, and for two ribozymes of the hepatitis delta virus (Ferre-D'Amare *et al.*, 1998). *In vitro* RNA evolution (SELEX) experiments have yielded families of RNA structures which appear to share a common pseudoknotted structure, such as RNA ligands selected to bind HIV-1 reverse transcriptase (Tuerk *et al.*, 1992; Burke *et al.*, 1996).

Pseudoknots (non-nested correlations) are more difficult to model than palindromic (nested) correlations. Pseudoknots cannot be generated using a single context-free grammar, nor can they be predicted by MFOLD. The next category of languages in the Chomsky hierarchy, the context-sensitive languages, are unattractive because for them parsing is an NP-complete problem. Some attempts to model the language of RNA pseudoknots involve using intersections of context-free grammars (Brown and Wilson, 1996; Lefebvre, 1996). However, the intersecting grammar approach is a heuristic, and guaranteed optimality of the resulting parse is unfortunately sacrificed. Similarly, most 'conventional' approaches (i.e. not based on ideas borrowed from linguistics) to the prediction of pseudoknotted RNA structures are also heuristic (Abrahams *et al.*, 1990; Gultyaev *et al.*, 1995; van Batenburg *et al.*, 1995).

It is now clear that pseudoknotted RNA structures can be parsed using optimal polynomial time parsing algorithms. Stormo's group (Cary and Stormo, 1995; Tabaska *et al.*, 1998) has showed that polynomial-time maximum weighted matching algorithms can deal with pseudoknotted RNA structures. Subsequently, we described a polynomial-time dynamic programming algorithm for minimum energy RNA secondary structure prediction (Rivas and Eddy, 1999), which mirrors the essential features of the MFOLD algorithm while extending it to pseudoknotted structures. However, in that paper, we only developed the recognition algorithm, and did not develop the formal grammar that corresponded to it.

A formal pseudoknot grammar, in stochastic form, would allow us to develop full probabilistic models of pseudoknotted RNA structures. This means we could deal with pseudoknots not only in the problem of single-sequence structure prediction by minimum energy (Rivas and Eddy, 1999), but also in the essentially statistical problems of consensus secondary structure prediction by comparative analysis, or of structural homology recognition in database searches (Eddy and Durbin, 1994).

In this paper we describe a formal transformational grammar that corresponds to the pseudoknot folding algorithm described in (Rivas and Eddy, 1999). The approach we take is to develop a grammar formalism that extends and includes the context-free grammars. Comparable approaches in the computational linguistics literature include the mildly context-sensitive grammars (Joshi *et al.*, 1991), string variable grammars (Searls, 1993), indexed grammars (Aho, 1968), or cut grammars (Searls, 1999). The key feature of our grammar is the use of special nonterminal symbols which dictate specific rearrangements of substrings in a derivation, an idea that has long been used in problems concerning natural language processing (Gazdar *et al.*, 1985).

The paper is organized as follows. First, we specify the components of this class of grammars. Then we give a simple example grammar that deals with the classical example of a *copy language*. This language requires more power than the one provided by context-free grammars. We then give a generalized form of the grammar class, and point out the specific approximations that lead to the particular grammar instances that generate the copy language and the RNA folding including pseudoknots. The recognition algorithm for these two particular grammars works in polynomial time. Finally, we give the specific grammar that corresponds to the parsing algorithm in Rivas and Eddy (1999) for single-sequence RNA secondary structure prediction including pseudoknots. This grammar closely captures the main features of the accepted thermodynamic model for RNA secondary structure (Turner *et al.*, 1987; Zuker and Stiegler, 1981), extended to include pseudoknotted structures.

Crossed-interaction grammar

Components of the grammar

A grammar G that includes crossing interactions (pseudoknots) can be defined by the sextuple

$$G = \{V, T, S, I, P, R\}. \qquad (1)$$

- V is the finite set of nonterminal symbols.

- T is the finite set of terminal symbols (the alphabet). T^*, the set of all strings over T, includes the empty string ϵ, and an additional empty string, $\wedge$, the 'hole' string, which also has length zero.

- S is the start nonterminal ($S \in V$).

- I is the finite set of extra nonterminal symbols. These symbols, together with the hole string (which acts as a marker), will differentiate this grammar from a conventional context-free grammar.

- P is the finite set of productions. The I symbols can only appear in the right-hand side of these productions.

- R is the finite set of rearrangement rules. The rearrangements are applied after the productions P, when all the conventional nonterminals have been eliminated. The extra symbols I are then used to rearrange (according to R) the strings generated by the productions P.

A simple example: copy-language grammar

A simple example of a language that includes crossing interactions is a copy language. A copy language for the alphabet $T = \{a, b\}$ consists of a set of patterns (such as *ab*, or *aba*, etc.) that are duplicated. The apparently simple string *abab* in which both *a*s and both *b*s are correlated is an example of a crossed interaction.

A grammar of the type described in (1) that generates the copy language

$$L = \{\epsilon, w \wedge w \mid w \in (a, b)^*\} \qquad (2)$$

requires only two nonterminals $V = \{W, W_H\}$, of which W is the start nonterminal. To take care of the crossed interactions we require three extra symbols

$$I = \{(,), \times\}. \qquad (3)$$

The productions P are given by

$$W \longrightarrow WW \mid (W_H \times W_H) \mid \epsilon \qquad (4)$$
$$W_H \longrightarrow a \wedge a \mid b \wedge b \mid (W_H \times W_H) \mid \wedge. \qquad (5)$$

The vertical bars indicate disjunction. ϵ is the usual empty string. We also introduce a second empty string $\wedge$ (which we call the 'hole' string). The hole string also has length zero but it is different from ϵ because it determines the possible point of insertion of another string. Note that the I symbols (3) can only appear in the right-hand side of the P productions (4) and (5).

In this example we use only one rearrangement R,

$$(m_1 \wedge m_1' \times m_2 \wedge m_2') \quad \underset{R}{\longrightarrow} \quad m_1 m_2 \wedge m_1' m_2', \qquad (6)$$

for any string of terminals m_1, m_1', m_2, and m_2' in the alphabet T. The parentheses indicate that a rearrangement has to be performed once the strings of terminals have been generated. That is, rearrangements take place only after all conventional nonterminals have been eliminated.

Rearrangements in inner parentheses are performed first. Once all rearrangements have been performed, and the extra symbols I have been eliminated, the $\wedge$ hole string becomes just equivalent to the empty string ϵ.

There is a simple diagrammatic representation associated with this grammar. None of the productions for W in (4) includes the $\wedge$ string, and in the derivations from W the final step has to produce a ϵ string. Therefore, the nonterminal W is diagrammatically represented by a no-hole diagram (or no-gap matrix). On the other hand, some of the productions for W_H (5) include the $\wedge$ string, and in the derivations from W_H the final step has to include a $\wedge$ string. Therefore the nonterminal W_H is diagrammatically represented by a diagram with a hole (or gap matrix). The extra symbol $\times$ represents the crossing necessary to put together two W_H gap matrices. The diagrammatic representation of productions in (4) and (5) in which we observe how the rearrangements take place is as follows:
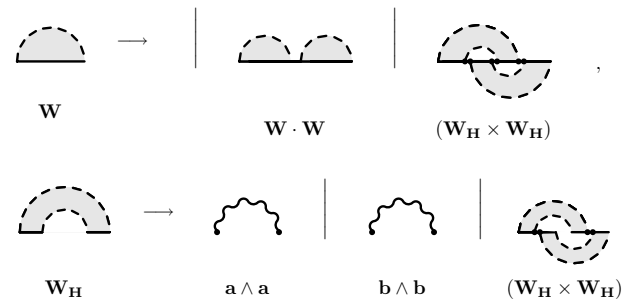
Adopting the conventions introduced in Rivas and Eddy (1999), a wavy line represents the coordinated emission of two terminals; a discontinuous line indicates no emission. (In RNA language, the wavy line represents the emission of two paired nucleotides.)

With this grammar we can generate the toy 'knotted' sequence
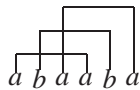

in the following way

$$
\begin{aligned}
W &\Longrightarrow (W_H \times W_H) \\
&\overset{*}{\Longrightarrow} (a \wedge a \times b \wedge b) \qquad (7) \\
&\underset{R}{\Longrightarrow} ab \wedge ab.
\end{aligned}
$$

When we apply a production $A \rightarrow \alpha$, we call that a derivation and it is represented by a double arrow. Similarly a single derivation for a rearrangement gets represented by $\underset{R}{\Longrightarrow}$. (Steps that include a multiple (but finite) number of derivations are represented by $\overset{*}{\Longrightarrow}$ and $\underset{R}{\overset{*}{\Longrightarrow}}$ respectively.)

Schematically, the rearrangement derivation can be visualized as

$$\left(\overline{a \wedge a} \quad \times \quad \overline{b \wedge b} \right) \quad \underset{R}{\Longrightarrow} \quad \overline{a \, b \, \wedge \, a \, b} \; .$$
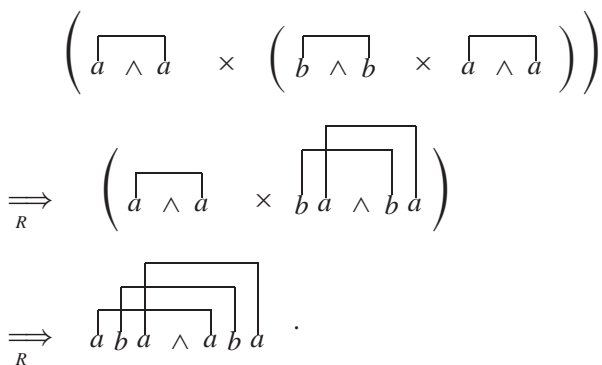
The use of parentheses allows us to create more complicated pseudoknotted structures. For example, the sequence

$$a \, b \, a \, a \, b \, a$$

can be parsed with the previous grammar in the following way:

$$
\begin{aligned}
W & \Longrightarrow (W_H \times W_H) \\
& \Longrightarrow (W_H \times (W_H \times W_H)) \\
& \overset{*}{\Longrightarrow} (a \wedge a \times (b \wedge b \times a \wedge a)) \\
& \underset{R}{\Longrightarrow} (a \wedge a \times ba \wedge ba) \\
& \underset{R}{\Longrightarrow} aba \wedge aba .
\end{aligned}
\tag{8}
$$

Schematically the two rearrangement derivations correspond to

$$\left(\overline{a \, \wedge \, a} \quad \times \quad \left(\overline{b \, \wedge \, b} \quad \times \quad \overline{a \, \wedge \, a} \right) \right)$$

$$\underset{R}{\Longrightarrow} \quad \left(\overline{a \, \wedge \, a} \quad \times \quad \overline{b \, a \, \wedge \, b \, a} \right)$$

$$\underset{R}{\Longrightarrow} \quad \overline{a \, b \, a \, \wedge \, a \, b \, a} \; .$$

From the diagrammatic representation of the grammar, it is an easy observation that the recognition algorithm has a worse-case complexity of $\mathcal{O}(n^6)$ in time and $\mathcal{O}(n^4)$ in space, for a string of length n.

General formalism

In the general formalism for a crossed-interaction grammar given an alphabet $T = \{a_1, \cdots, a_n\}$, we define T^* the set of all strings over T as

$$T^* = \{\epsilon, \wedge, a_1, \ldots, a_1 a_1, a_1 a_2, \ldots, a_1 a_1 a_1, a_1 a_1 a_2, \ldots\}.$$

Where ϵ is the empty string, and $\wedge$ is the 'hole' string which also has length zero but it is different from ϵ because it acts as a marker of the possible splice point for the insertion of another string. Nonterminals that do not include the $\wedge$ string in their derivation are the 'non-gapped nonterminals'. Nonterminals that include the $\wedge$ string in their derivation are the gapped nonterminals.

- The set of productions P have the general form

$$P = \{A \to \alpha \mid A \in V, \quad \alpha \in (V(IV)^* \cup T)^*\}. \tag{9}$$

Derivations are similar to those of context-free grammars except that extra symbols I may appear interspersed in between nonterminals V in the right-hand side of the productions. We introduce the notation $IV = \{i\alpha \mid \alpha \in V \quad \text{and} \quad i \in I\}$.

- The rearrangements R are of the general form,

$$R = \{(\gamma) \underset{R}{\to} m \mid \gamma \in (T \cup I)^*, \quad m \in T^*\}. \tag{10}$$

The rearrangements are applied after the productions P, when all the conventional nonterminals have been eliminated. In the rearrangement rules the extra symbols I are used to reorder the strings generated by the productions (9). In the absence of extra symbols I, the $\wedge$ empty string becomes equivalent to the ϵ empty string.

- The language generated by this grammar $L(G)$ is

$$L(G) = \{m \mid m \in T^* \quad \text{such that} \quad S \overset{*}{\Rightarrow} \gamma \underset{R}{\overset{*}{\Rightarrow}} m,$$

$$\text{for} \quad \gamma \in (T \cup I)^*\}. \tag{11}$$

We say that $\alpha_1 \overset{*}{\Rightarrow} \alpha_2$ if α_2 can be derived from α_1 in a finite number of derivations. Therefore, a string belongs to $L(G)$ if it consists only of terminals (extra symbols are not allowed), and it can be generated from the start nonterminal S.

In the particular case $I = \emptyset$, then $\alpha \in (V \cup T)^*$ in (9). The rearrangement rules then turn into an identity, the wedge string becomes equivalent to the empty string, and the grammar becomes a standard context-free grammar.

In general, for unrestricted transitions $A \to \alpha$ with $\alpha \in (V(IV)^* \cup T)^*$ parsing using this grammar is non-trivial and probably an NP-complete problem. The key to make the grammar tractable is to realize that

$$(V(IV)^* \cup T)^* = \cup_{n=0}^{\infty}(V(IV)^n \cup T)^*. \tag{12}$$

If we truncate this infinite sum by using only the term of lowest order ($n = 0$) we have again a context-free grammar ($A \to \alpha, \quad \alpha \in (V \cup T)^*$). For $n > 0$ we deviate from a context-free grammar. The copy language grammar presented in the previous section corresponds to truncating the infinite sum keeping only the first two terms ($A \to \alpha, \quad \alpha \in (V \cup T)^* \cup (VIV \cup T)^*$) (with a specific set of

extra symbols $I = \{(,), \times\}$). For RNA pseudoknots, a specific grammar adhering to the same approximation is quite adequate to describe a large number of pseudoknot structures. This restriction is equivalent to the one introduced in Rivas and Eddy (1999) in the diagrammatic representation, and corresponds to not allowing more than two gap matrices to come together at a given time. For the specific case of the RNA pseudoknot grammar (below) we have developed a recognition algorithm (Rivas and Eddy, 1999) that works in polynomial time.

RNA pseudoknot grammar

The grammar to describe RNA pseudoknots has the following specifications:

- The grammar has the following nonterminals,

$$V = \{W, W_B, V^{ab}, W_H, V_H^{abcd}, IS1, IS2\}. \quad (13)$$

W, W_B, and V^{ab} are non-gapped nonterminals. W_H and V_H^{abcd} are the gapped nonterminals, for $a, b, c, d \in T$. $IS1$ and $IS2$ are the nonterminals that create hairpin loops and loops closed by two interactions (stems, bulges and internal loops) respectively.

- The alphabet is the RNA alphabet

$$T_{\text{RNA}} = \{a, c, g, u\}. \quad (14)$$

- There are six extra symbols

$$I = \{(,), \times, \times_L, \times_R, \supset\}. \quad (15)$$

These special nonterminals are specific for pseudoknots and model the different types of knotted bifurcations present in RNA.
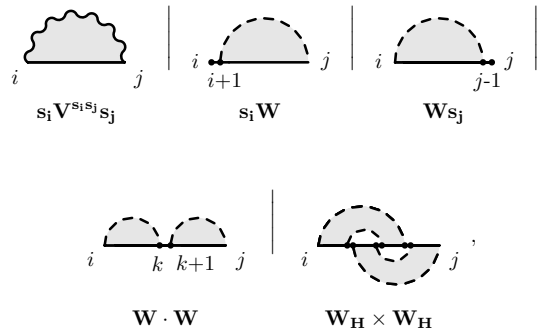
- Production rules.

The nonterminal W is the start nonterminal. [The nonterminal W_B has similar productions to W (just substitute W_B for W in the following recursions), but it appears only in enclosed structures after a pair has already been emitted (i.e. multiloops).] The production rules for W are (excluding coaxials or danglings):
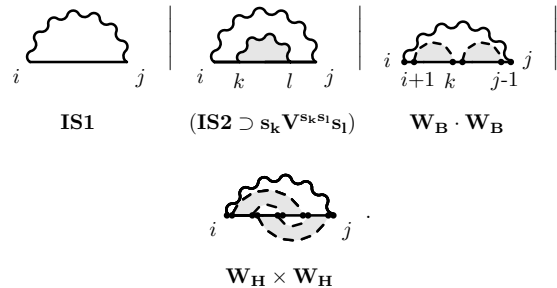

$$\quad (16)$$

where $s_i \in T$ stands for the terminal (nucleotide) emitted in position i.

V^{ab} is the nonterminal we are at after a paired emission $a, b \in T$ has occurred. The production rules for nonterminal V^{ab} are,
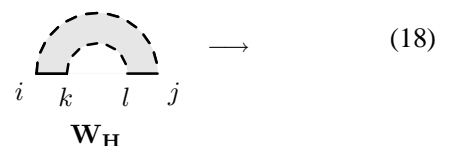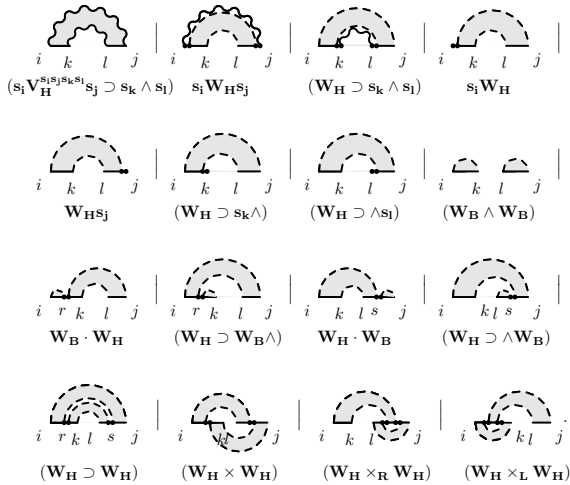

$$\quad (17)$$

W_H and V_H^{abcd} are nonterminals specific for pseudoknots and correspond to the gap matrices whx and vhx described in Rivas and Eddy (1999). W_H is the nonterminal that introduces pseudoknots, and its recursion is given by
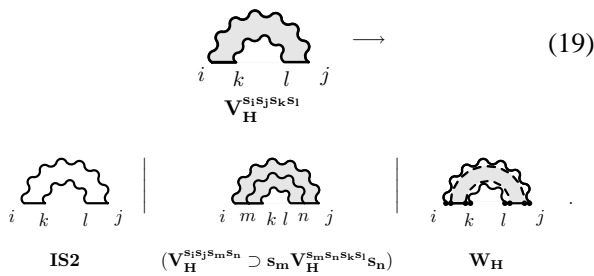

$$\quad (18)$$

$(s_i V_H^{s_i s_j s_k s_l} s_j \supset s_k \wedge s_l)$ | $s_i W_H s_j$ | $(W_H \supset s_k \wedge s_l)$ | $s_i W_H$

$W_H s_j$ | $(W_H \supset s_k \wedge)$ | $(W_H \supset \wedge s_l)$ | $(W_B \wedge W_B)$

$W_B \cdot W_H$ | $(W_H \supset W_B \wedge)$ | $W_H \cdot W_B$ | $(W_H \supset \wedge W_B)$

$(W_H \supset W_H)$ | $(W_H \times W_H)$ | $(W_H \times_R W_H)$ | $(W_H \times_L W_H)$

V_H^{abcd} is the nonterminal we are in after emitting two pairs $a, b \in T$ and $c, d \in T$ in a pseudoknot. The recursion for nonterminal V_H^{abcd} is

$$\longrightarrow \qquad (19)$$

$$V_H^{s_i s_j s_k s_l}$$

$$\mathbf{IS2} \quad | \quad (V_H^{s_i s_j s_m s_n} \supset s_m V_H^{s_m s_n s_k s_l} s_n) \quad | \quad W_H .$$

Moreover, the RNA language requires nonterminals to take care of length distributions for hairpin loops (IS1s), and internal loops (bulges, stems, and internal loops that we collect under the name of IS2s). For the nonterminals required to take care of loop-length distributions, we have for hairpin loops,

$$IS1 \longrightarrow \epsilon \mid s_1 \mid s_1 s_2 \mid \ldots \mid s_1 \ldots s_{\text{maxloop}}. \quad (20)$$

For stems, bulges and internal loops,

$$IS2 \longrightarrow \wedge \mid s_1 \cdots s_k \wedge \mid \wedge s_1 \cdots s_k \mid$$
$$s_1 \cdots s_{i-1} \wedge s_i \cdots s_k, \quad (21)$$

with $1 \leq i \leq k$ and $1 \leq k \leq \text{maxloop}$.

- The rearrangement rules are

$$(m_1 \wedge m_1' \times m_2 \wedge m_2') \xrightarrow{R} m_1 m_2 \wedge m_1' m_2',$$
$$(m_1 \wedge m_1' \times_L m_2 \wedge m_2') \xrightarrow{R} m_2 m_1 m_2' \wedge m_1',$$
$$(m_1 \wedge m_1' \times_R m_2 \wedge m_2') \xrightarrow{R} m_1 \wedge m_2 m_1' m_2', \quad (22)$$
$$(m_1 \wedge m_1' \supset m_2 \wedge m_2') \xrightarrow{R} m_1 m_2 \wedge m_2' m_1',$$

for any sequence $m_1, m_1', m_2,$ and $m_2' \in T^*$.

A recognition algorithm for the grammar presented here (with some additional features such as danglings and coaxials) has been implemented in Rivas and Eddy (1999) as a non-stochastic model using thermodynamic parameters (Turner *et al.*, 1987) to describe the different transitions. The dynamic programming algorithm has a time complexity of $\mathcal{O}(L^6)$ and a storage complexity of $\mathcal{O}(L^4)$, for a RNA sequence of length L. The algorithm is able to predict a large number of the RNA pseudoknots found—either confirmed or proposed—in the literature (Rivas and Eddy, 1999).

This RNA pseudoknot grammar also permits us to define a probabilistic model similar to those that have been developed for context-free grammars (Eddy and Durbin, 1994). In this way we can implement a full probabilistic model for RNA folding with pseudoknots. The pseudoknot grammar described here is also ambiguous (in the formal sense) and, therefore, it is able to capture alternative secondary structures when the parser is implemented as an Inside algorithm (Searls, 1999).

Conclusion

Previously, we described a polynomial-time dynamic programming algorithm for optimal RNA structure prediction including pseudoknots. The RNA pseudoknot algorithm was implemented in a non-stochastic form using free energies to estimate the energetically optimal folding. This recognition algorithm is more complicated than a context-free parser, but a formal grammar associated with the algorithm had not yet been developed. In this paper we have specified such a grammar for RNA pseudoknots. We have introduced a general grammar class (for which we do not specify a parser), and a specific grammar instance that corresponds to our RNA pseudoknot parsing algorithm. The pseudoknot grammar has an equivalent diagrammatic representation in terms of Feynman diagrams that provides a systematic means for its implementation. With a formal pseudoknot grammar in hand, it will be straightforward to produce a stochastic version of the recognition algorithm that would allow us to include pseudoknots in statistical problems such as homology recognition and consensus secondary structure prediction.

References

Aho,A.V. (1968) Indexed grammars—an extension of context-free grammars. *J. ACM*, **15**, 647–671.

Abrahams,J.P., van der Berg,M., van Batenburg,E. and Pleij,C.W.A. (1990) Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucl. Acids Res.*, **18**, 3035–3044.

Brown,M. and Wilson,C. (1996) RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. *Pacific Symposium on Biocomputing 1996*.

Burke,D.H., Scates,L., Andrews,K. and Gold,L. (1996) Bent pseudoknots and novel RNA inhibitors of type 1 human immunodeficiency virus (HIV-1) reverse transcriptase. *J. Mol. Biol.*, **264**, 650–666.

Cary,R.B. and Stormo,G.D. (1995) Graph-theoretic approach to RNA modeling using comparative data. In Rawlings,C. *et al.*, (eds), *ISMB-95*. AAAI Press, Menlo Park, CA, pp. 75–80.

Cech,T.R. (1993) Structure and mechanism of the large catalytic RNAs: Group I and group II introns and ribonuclease P. In Gesteland,R.F. and Atkins,J.F. (eds), *The RNA World.* Cold Spring Harbor Press, New York, pp. 239–270.

Chomsky,D. (1959) On certain formal properties of grammars. *Inform. Cont.*, **2**, 137–176.

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

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

Ferré-D'Amaré,A.R., Zhou,K. and Doudna,J.A. (1998) Crystal structure of a hepatitis delta virus ribozyme. *Nature*, **395**, 567–574.

Gazdar,G., Klein,E., Pullum,G.K. and Sag,I.A. (1985) *Generalized Phase Structure Grammar*. Harvard University Press, Cambridge, MA.

Gultyaev,A.P., van Batenburg,F.H. and Pleij,C.W.A. (1995) The computer simulation of RNA folding pathways using a genetic algorithm. *J. Mol. Biol.*, **250**, 37–51.

Joshi,A.K., Vijay-Shanker,K. and Weir,D. (1991) The convergence of mildly context-sensitive grammar formalisms. In Sells,P., Shieber,S.M. and Wasow,T. (eds), *Foundational Issues in Natural Language Processing.* MIT Press, Cambridge, MA, pp. 31–81.

Kolk,M.H., van der Graff,M., Wijmenga,S.S., Pleij,C.W.A., Heus,H.A. and Hilbers,C.W. (1998) NMR structure of a classical pseudoknot: interplay of single- and double-stranded RNA. *Science*, **280**, 434–438.

Lefebvre,F. (1996) A grammar-based unification of several alignment and folding algorithms. In Rawlings,C. *et al.*, (eds), *Procceedings, Fourth International Conference on Intelligent Systems for Molecular Biology.* AAAI Press, Menlo Park, pp. 143–154.

Rivas,E. and Eddy,S.R. (1999) A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, **285**, 2053–2068.

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. *Nucl. Acids Res.*, **22**, 5112–5120.

Searls,D.B. (1992) The linguistics of DNA. *Am. Sci.*, **20**, 579–591.

Searls,D.B. (1993) String variable grammar: a logic grammar formalism for the biological language of DNA. *J. Logic Program.*, **12**, 1–30.

Searls,D.B. (1999) Formal language theory and biological macromolecules. *Ser. Discrete Math. Theor. Comp. Sci.*, **47**, 117–140.

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.

ten Dam,E., Pleij,K. and Draper,D. (1992) Structural and functional aspects of RNA pseudoknots. *Biochemistry*, **31**, 11665–11676.

Tuerk,C., MacDougal,S. and Gold,L. (1992) RNA pseudoknots that inhibit human immunodeficiency virus type 1 reverse transcriptase. *Proc. Natl Acad. Sci. USA*, **89**, 6988–6992.

Turner,D., Sugimoto,N., Jaeger,J., Longfellow,C., Freier,S. and Kierzek,R. (1987) Improved parameters for prediction of RNA structure. *Cold Spring Harbor Symp. Quant. Biol.*, **52**, 123–133.

van Batenburg,F.H.D., Gultyaev,A.P. and Pleij,C.W.A. (1995) An APL-programmed genetic algorithm for the prediction of RNA secondary structure. *J. Theor. Biol.*, **174**, 269–280.

Zimmerman,R.A. and Dahlberg,A.E. (1996) *Ribosomal RNA*. CRC Press, Boca Raton, FL.

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