The Application of Stochastic Context-Free Grammars to Folding, Aligning and Modeling Homologous RNA Sequences

Yasubumi Sakakibara*†, Michael Brown†, Richard Hughey‡, I. Saira Mian§, Kimmen Sjölander†, Rebecca C. Underwood†, David Haussler†

† Computer and Information Sciences ‡ Computer Engineering § Sinsheimer Laboratories
University of California, Santa Cruz, CA 95064, USA
Email: haussler@cse.ucsc.edu

UCSC-CRL-94-14
November 1, 1993

Keywords: Stochastic Context-Free Grammar, RNA, Transfer RNA, Multiple Sequence Alignments, Secondary Structure.

Abstract

Stochastic context-free grammars (SCFGs) are applied to the problems of folding, aligning and modeling families of homologous RNA sequences. SCFGs capture the sequences' common primary and secondary structure and generalize the hidden Markov models (HMMs) used in related work on protein and DNA. The novel aspect of this work is that SCFG parameters are learned automatically from unaligned, unfolded training sequences. A generalization of the HMM forward-backward algorithm is introduced to do this. The new algorithm, Tree-Grammar EM, based on tree grammars and faster than the previously proposed SCFG inside-outside training algorithm, produced a model that we tested on the transfer RNA (tRNA) family. Results show that after having been trained on as few as 20 tRNA sequences from only two tRNA subfamilies (mitochondrial and cytoplasmic), the model can discern general tRNA from similar-length RNA sequences of other kinds, can find secondary structure of new tRNA sequences, and can produce multiple alignments of large sets of tRNA sequences. Our results suggest potential improvements in the alignments of the D- and T-domains in some mitochdondrial tRNAs that cannot be fitted into the canonical secondary structure.

1 Introduction

Both computer science and molecular biology are evolving rapidly as disciplines, and predicting the structure of macromolecules by theoretical or experimental means remains a challenging problem. Efforts to sequence the genomes of organisms (Sanger *et al.*, 1982; Daniels *et al.*, 1983; Sanger *et al.*, 1977; Sanger *et al.*, 1978; Dunn & Studier, 1981; Dunn & Studier, 1983; Baer *et al.*, 1984; Daniels

*Current address: ISIS, Fujitsu Labs Ltd., 140, Miyamoto, Numazu, Shizuoka 410-03, Japan

et al., 1992; Plunkett 3d *et al.*, 1993; Ogasawara, 1993; Oliver *et al.*, 1992; Sulston *et al.*, 1992; Merriam *et al.*, 1991; Joint NIH/DOE Mouse Working Group, 1993; Olson, 1993; Okada & Shimura, 1993; von Montagu *et al.*, 1992; Minobe, 1993) and organelles (Hiratsuka *et al.*, 1989; Hallick *et al.*, 1993; Crozier & C., 1993; Oda *et al.*, 1992; Tzeng *et al.*, 1992; Cummings *et al.*, 1990; Cantatore *et al.*, 1989; Gadaleta *et al.*, 1989; Sutcliffe, 1979) have heightened awareness of the use of computers in data acquisition, management and analysis. The increasing numbers of DNA, RNA and protein sequences yielded by these projects (Courteau, 1991) highlight a growing need for developing new approaches in computational biology such as hidden Markov models (HMMs) (Lander & Green, 1987; Churchill, 1989; Rabiner, 1989; Haussler *et al.*, 1993; Krogh *et al.*, 1994; Baldi *et al.*, 1993; Cardon & Stormo, 1992) and other approaches (Hunter *et al.*, 1993). In addition to the accelerated discovery of sequences related by a natural phylogeny, the generation of "artificial" phylogenies by experimental design for proteins (reviewed in Arnold's paper (Arnold, 1993)) and RNA (reviewed in an article by Burke and Berzal-Herranz (Burke & Berzal-Herranz, 1993)) serves only to exacerbate the problem of growth in sequence data. Hence, determining common or consensus patterns among a family of sequences, producing a multiple sequence alignment, discriminating members of the family from non-members and discovering new members of the family will continue to be some of the most important and fundamental tasks in mathematical analysis and comparison of macromolecular sequences (Dahlberg & Abelson, 1989; Doolittle, 1990). In this paper, we apply *stochastic context-free grammars* (SCFGs) to the problems of statistical modeling, multiple alignment, discrimination and prediction of the secondary structure of RNA families. This approach is highly related to our work on modeling protein families and domains with HMMs (Haussler *et al.*, 1993; Krogh *et al.*, 1994).

In RNA, the nucleotides adenine (`A`), cytosine (`C`), guanine (`G`) and uracil (`U`) interact in specific ways to form characteristic secondary-structure motifs such as helices, loops and bulges (Saenger, 1984; Wyatt *et al.*, 1989). Further folding and hydrogen-bonding interactions between remote regions orient these secondary-structure elements with respect to each other to form the functional system. Higher-order interactions with other proteins or nucleic acids may also occur. In general, however, the folding of an RNA chain into a functional molecule is largely governed by the formation of intramolecular `A-U` and `G-C` Watson-Crick pairs as well as `G-U` and, more rarely, `G-A` base pairs. Such base pairs constitute the so-called *biological palindromes* in the genome.

Comparative analyses of two or more protein or nucleic-acid sequences have been used widely in detection and evaluation of biological similarities and evolutionary relationships. Several methods for producing these multiple sequence alignments have been developed, most based on dynamic programming techniques (for example, see works by Waterman (Waterman, 1989)). However, when RNA sequences are to be aligned, both the primary *and* secondary structure need to be considered since generation of a multiple sequence alignment and analysis of folding are mutually dependent exercises. Elucidation of common folding patterns among two or more sequences may indicate the pertinent regions to be aligned and vice versa (Sankoff, 1985).

Currently, there are two principal methods for predicting secondary structure of RNA, or which nucleotides are base-paired. Phylogenetic analysis of homologous RNA molecules (Fox & Woese, 1975; Woese *et al.*, 1983) ascertains structural features that are conserved during evolution. It is based on the premise that functionally equivalent RNA molecules are also structurally equivalent and relies on alignment and subsequent folding of many sequences into similar secondary structures (see review papers (James *et al.*, 1989; Woese *et al.*, 1983)). Comparative methods have been used

to infer the structure of tRNA (Levitt, 1969; Holley *et al.*, 1965; Madison *et al.*, 1966; Zachau *et al.*, 1966; RajBhandary *et al.*, 1966), 5S RNA (Fox & Woese, 1975), 16S ribosomal RNA (rRNA) (Woese *et al.*, 1980; Stiegler *et al.*, 1980; Zwieb *et al.*, 1981), 23S rRNA (Noller *et al.*, 1991; Glotz *et al.*, 1981; Branlant *et al.*, 1981), group I introns (Michel & Westhof, 1990; Michel *et al.*, 1990), group II introns (Michel *et al.*, 1989), RNAse P RNA (Brown *et al.*, 1991; Tranguch & Engelke, 1993), small nuclear RNAs (Guthrie & Patterson, 1988), 7S RNA (signal recognition particle RNA) (Zwieb, 1989), telomerase RNA (Romero & Blackburn, 1991), MRP RNA (Schmitt *et al.*, 1993) and TAR RNA of human and simian immunodeficiency viruses (Berkhout, 1992). The original procedure of Noller and Woese (Noller & Woese, 1981) detected compensatory base changes in putative helical elements: contiguous antiparallel arrangement of `A-U`, `G-C` and `G-U` pairings. Positions that covaried were assumed to be base-paired. This procedure was subsequently formalized into an explicit computer algorithm (Waterman *et al.*, 1984; Waterman, 1988) that stores *all* "interesting" patterns, a potential problem as the number of patterns increases. The algorithm of Sankoff (Sankoff, 1985) for simultaneously aligning and folding sequences is generally impractical in terms of time and space for large numbers of long sequences. Given an alignment of homologous RNA sequences, heuristic methods have been proposed to predict a common secondary structure (Han & Kim, 1993; Chiu & Kolodziejczak, 1991; Chan *et al.*, 1991). However, there remains no reliable or automatic way of inferring an optimal consensus secondary structure even if the related sequences are already aligned. Because considerable manual intervention is still required to identify potential helices that maintain base complementarity, automation and development of more rigorous comparative analysis protocols are under continual development (Gutell *et al.*, 1992; Lapedes, 1992; Klinger & Brutlag, 1993; Waterman, 1989; Winker *et al.*, 1990).

The second technique for predicting RNA secondary structure employs thermodynamics to compare the free energy changes predicted for formation of possible secondary structure and relies on finding the structure with the lowest free energy (Tinoco Jr. *et al.*, 1971; Turner *et al.*, 1988; Gouy, 1987). Such energy minimization depends on thermodynamic parameters and computer algorithms to evaluate the optimal and suboptimal free-energy folding of an RNA species (see review papers (Jaeger *et al.*, 1990; Zuker & Sankoff, 1984)). To obtain a common folding pattern for a set of related molecules, Zuker has suggested predicting a folding for each sequence separately using these algorithms and then searching for a common structure (Zuker, 1989). Limitations of this method are partially due to the uncertainty in the underlying energy model, and the technique may be overly sensitive to point mutations. Attempts are being made to combine both phylogenetic and energetic approaches (Le & Zuker, 1991).

Using methods different from those described above, several groups have enumerated schemes or programs to search for patterns in proteins or nucleic acid sequences (Staden, 1990; Lathrop *et al.*, 1987; Sibbald & Argos, 1990; Abarbanel *et al.*, 1984; Saurin & Marlière, 1987; Gautheret *et al.*, 1990; Cohen *et al.*, 1986; Presnell & Cohen, 1993). String pattern-matching programs based on the UNIX `grep` function, developed in unpublished work by S. R. Eddy (Schneider *et al.*, 1992) and others (Macke *et al.*, 1993), search for secondary structure elements in a sequence database. If there is prior knowledge about sequence and structural aspects of an RNA family, this can be employed to create a *descriptor* (discriminating pattern) for the family which can then be used for database searching or generating an alignment for the family. This has been demonstrated most clearly for tRNA (Fichant & Burks, 1991; Staden, 1980; Marvel, 1986), where approximate string matching (locating all occurrences of substrings that are within a given similarity neighborhood of

an exact match to the pattern) proved to be important.

Our method of multiple alignment and folding differs markedly from the conventional techniques because it builds a statistical model *during* rather than *after* the process of alignment and folding. Such an approach has been applied successfully to modeling protein families with HMMs (Haussler *et al.*, 1993; Krogh *et al.*, 1994).

Though in principle HMMs could be used for RNA, we strongly suspect that the more general statistical models described here are required. Since base-pairing interactions, most notably A-U, G-C and G-U, play such a dominant role in determining RNA structure and function, any statistical method that does not consider this will eventually encounter insurmountable problems. The problem is that if two alignment positions are base-paired in the typical RNA, then the bases occurring there will be highly correlated, whereas the standard HMM approach will treat them as having independent distributions.

In this paper, we describe a means to generalize HMMs to model most of the interactions seen in RNA using formal language theory. As in the elegant work of Searls (Searls, 1992), we view the strings of characters representing pieces of DNA, RNA and protein as sentences derived from a formal grammar. The simplest kind of grammar is a *regular* grammar, in which strings are derived from productions (rewriting rules) of the forms $S \rightarrow aS$ and $S \rightarrow a$, where S is a *nonterminal symbol*, which does not appear in the final string, and a is a *terminal symbol*, which appears as a letter in the final string. Searls has shown base pairing in RNA can be described by a *context-free grammar* (CFG), a more powerful class of formal grammars than the regular grammar (see Section 2.1). CFGs are often used to define the syntax of programming languages. A CFG is similar to a regular grammar but permits a greater variety of productions, such as those of the forms $S \rightarrow SS$ and $S \rightarrow aSa$. As described by Searls, it is precisely these additional types of productions that are needed to describe the base-pairing structure in RNA.[1] In particular, the productions of the forms $S \rightarrow \mathtt{A}\ S\ \mathtt{U}$, $S \rightarrow \mathtt{U}\ S\ \mathtt{A}$, $S \rightarrow \mathtt{G}\ S\ \mathtt{C}$ and $S \rightarrow \mathtt{C}\ S\ \mathtt{G}$ describe the structure in RNA due to Watson-Crick base pairing. Using productions of this type, a CFG can specify the language of biological palindromes.

Searls' original work (Searls, 1992) argues the benefits of using CFGs as models for RNA folding, but does not discuss stochastic grammars or methods for creating the grammar from training sequences. One purpose of this paper is to provide an effective method for building a stochastic context-free grammar (SCFG) to model a family of RNA sequences. Some analogues of stochastic grammars and training methods do appear in Searls' most recent work in the form of costs and other trainable parameters used during parsing (Searls, 1993a; Searls, 1993b; Searls & Dong, 1993), but we believe that our integrated probabilistic framework may prove to be a simpler and more effective approach.

If we specify a probability for each production in a grammar, we obtain a *stochastic* grammar. A stochastic grammar assigns a probability to each string it derives. Stochastic regular grammars are equivalent to HMMs and suggest an interesting generalization from HMMs to SCFGs (Baker, 1979). In this paper, we pursue a stochastic model of the family of transfer RNAs (tRNAs) by using a SCFG that is similar to our protein HMMs (Krogh *et al.*, 1994) but incorporates base-pairing

[1] CFGs can not describe all RNA structure, but we believe they can account for enough to make useful models. In particular, CFGs cannot account for pseudoknots, structures generated when a single-stranded loop region base pairs with a complementary sequence outside the loop (ten Dam *et al.*, 1992; Wyatt *et al.*, 1989; Pleij, 1990). Similarly, base triples involving three positions, as well as interactions in parallel (versus the more usual anti-parallel) are not currently modeled.

information. (A representative tRNA sequence, a yeast tRNA, is shown in Figure 1.) We build a SCFG forming a statistical model of tRNA sequences in much the same way that we constructed an HMM representing a statistical model of a protein family. We use this model to discriminate tRNA from other RNAs of similar length, and to obtain a multiple alignment in the same manner as for proteins. We also use the model to fold unfolded tRNA sequences, determining the base pairing that defines their secondary structure.

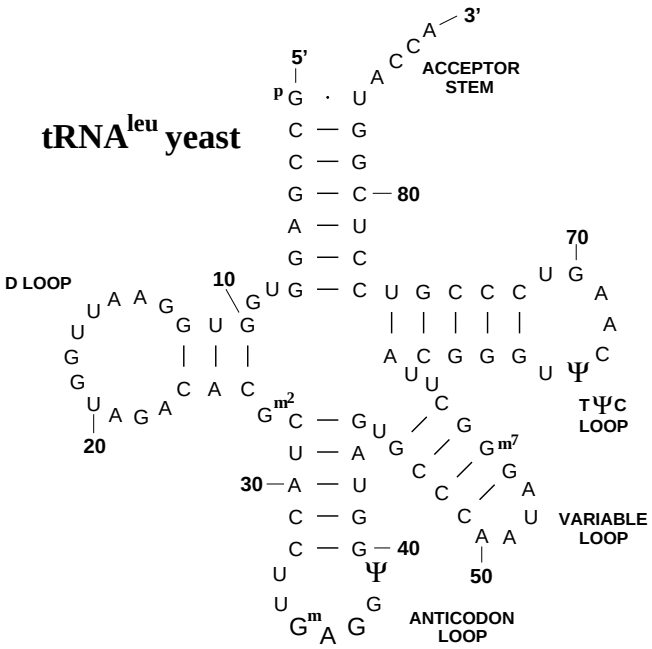


Figure 1: A typical tRNA sequence contains four main domains: acceptor, D-, anticodon and $T\Psi C$- or T. Watson-Crick base pairings are indicated as dashes and G-U as a dot. The superscripted positions indicate modified bases. The D- or T-domains are sometimes deleted in mitochondrial tRNAs, while the variable loop may be of different lengths.

In preliminary work (Sakakibara *et al.*, 1993), we derived a SCFG's probabilities (parameters) directly from an existing alignment of tRNA sequences to see how well the SCFG could model an RNA family. Here, we design an algorithm that deduces the parameters automatically from a set of *unaligned* primary sequences with a novel generalization of the *forward-backward algorithm* commonly used to train HMMs. Our algorithm, henceforth called Tree-Grammar EM, is based on tree grammars, and is more efficient than the *inside-outside algorithm* (Lari & Young, 1990), a computationally expensive generalization of the forward-backward algorithm developed to train SCFGs

(Baker, 1979). We use Tree-Grammar EM to derive, from different training sets of tRNA sequences, several *trained grammars*: `MixedTRNA500`, `ZeroTrain`, `MT100`, `MT10CY10` and `RandomTRNA618`. Our training and testing sequences were taken from the 1993 compilation of aligned tRNA sequences (Steinberg *et al.*, 1993b) maintained by EMBL Data Library (we converted all modifed bases to their unmodified forms). We refer to the alignments in this compilation as TRUSTED alignments.

For the preliminary trained grammar, `MixedTRNA500`, we chose 500 unfolded and unaligned sequences at random from 1477 tRNA sequences which can be fitted into a canonical tRNA structure (Figure 1). We withheld the remaining 977 sequences in order to test the trained grammar on data not used in training. For the remaining four grammars, we omitted duplicate sequences and sequences containing unusual characters from the tRNA compilation. The remaining 1222 tRNA sequences were then split into six groups—archaea, cytoplasm, mitochondria, cyanelles and chloroplasts, viruses and eubacteria. The four grammars were trained on subsets of these groups and tested on all remaining tRNAs, as well as on 2016 fragments of RNA taken from non-tRNA features in the NewGenBank and GenBank databases (we call these *non-tRNA sequences*). The grammar `MT10CY10` was trained with just 10 randomly selected mitochondrial sequences and 10 randomly selected cytoplasmic tRNA sequences; `MT100` was trained with 100 randomly selected mitochondrial tRNAs; and `RandomTRNA618` was trained with 618 tRNA from various families. The grammar `ZeroTrain` is a control that has no training, only prior probabilities as described in Section 2.6.

We assess each grammar's ability to perform three tasks: to discriminate tRNA sequences from non-tRNA sequences, to produce multiple alignments and to ascertain the secondary structure of new sequences. The results show that all the grammars except `ZeroTrain` can perfectly discriminate the nonmitochondrial tRNA sequences from the non-tRNA sequences. Some tRNAs have secondary structures that cannot be fitted into the canonical structure shown in Figure 1. These sequences, whose alignments differ from the conventional alignment, are treated separately in the publicly available tRNA database (Steinberg *et al.*, 1993b) and we refer to these as Part III tRNAs. Belonging to this group are tRNAs from mitochdondria of parasitic worms lacking the T- or D-domain, mammalian mitochondria lacking the D-domain, mitochondria of mollusc, insect and echinoderm with extended anticodon and T-stems, single cell organisms and fungi and *Trypanosoma brucei*.

Our trained grammars are able to discriminate regular mitochondrial tRNA from non-tRNA quite well. However, only 50% of the Part III tRNAs can be reliably distinguished from non-tRNAs even by our most heavily trained grammars. Here "reliably distinguished" means having a score that is more than 5 standard deviations from that of a typical non-tRNA of the same length, as described in Section 3.4. The majority of the sequences that could not be discriminated are parastic worm and mammalian mitochondrial tRNAs lacking the D-domain. In addition, these sequences cannot be aligned in the same manner as TRUSTED but inspection of their alignments indicates that a revision around the T-domain would create a T-stem with a greater number of Watson-Crick base pairs than in TRUSTED. However, PART III mitochondrial sequences lacking the T-domain *can* be both discriminated from non-tRNAs and their alignment is the same as TRUSTED.

We also compare the alignments and secondary structures predicted by our grammars to the TRUSTED alignments. For each tRNA sequence, we compute the percentage of base pairs present in the secondary structure from the TRUSTED alignment that are also present in the secondary structure predicted by the grammar. We find that all three trained grammars have approximately

98–99% base-pair agreement with both trusted alignments for all but the Part III sequences. As mentioned earlier, there are examples of plausible alternative alignments for some of these mitochondrial sequences.

We recently discovered that Sean Eddy and Richard Durbin have independently done work closely related to ours, obtaining comparable results (Eddy & Durbin, 1994). It appears that our basic grammar training algorithm, which is quite different from theirs, may be somewhat faster, and that our custom-designed grammars and greater emphasis on learned, as opposed to constructed, Bayesian prior probability densities (Brown *et al.*, 1993b) may allow us to train with fewer training sequences. However, they have developed an exciting new technique to learn the structure of the grammar itself from unaligned training sequences, rather than just learn the probabilities of the productions and rely on prior information to specify the structure of the grammar (as we do). Both investigations serve to demonstrate that SCFGs are a powerful tool for RNA sequence analysis. Such tools will become increasingly important as *in vitro* evolution and selection techniques produce greater numbers of "novel" RNA families (Burke & Berzal-Herranz, 1993; Bartel & Szostak, 1993; Ellington & Szostak, 1992; Lehman & Joyce, 1993; Beaudry & Joyce, 1992; Tuerk & Gold, 1990; Schneider *et al.*, 1992; Brenner & Lerner, 1992).

2 Methods

2.1 Context-free grammars for RNA

A grammar is principally a set of productions (rewrite rules) that is used to generate a set of strings, a *language*. The productions are applied iteratively to generate a string, a process called *derivation*. For example, application of the productions in Figure 2 could generate the RNA sequence CAUCAGGGAAGAUCUCUUG by the following derivation:

Beginning with the start symbol S_0, any production with S_0 left of the arrow can be chosen to have its right side replace S_0. If the production $S_0 \rightarrow S_1$ is selected (in this case, this is the only production available), then the symbol S_1 replaces S_0. This derivation step is written $S_0 \Rightarrow S_1$, where the double arrow signifies application of a production. Next, if the production $S_1 \rightarrow \mathtt{C}\ S_2\ \mathtt{G}$ is selected, the derivation step is $S_1 \Rightarrow \mathtt{C}\ S_2\ \mathtt{G}$. Continuing with similar derivation steps, each time choosing a nonterminal symbol and replacing it with the right-hand side of an appropriate production, we obtain the following derivation terminating with the desired sequence:

$$
\begin{aligned}
S_0 &\Rightarrow S_1 \Rightarrow \mathtt{C}S_2\mathtt{G} \Rightarrow \mathtt{CA}S_3\mathtt{UG} \Rightarrow \mathtt{CA}S_4S_9\mathtt{UG} \\
&\Rightarrow \mathtt{CAU}S_5\mathtt{A}S_9\mathtt{UG} \Rightarrow \mathtt{CAUC}S_6\mathtt{GA}S_9\mathtt{UG} \\
&\Rightarrow \mathtt{CAUCA}S_7\mathtt{GA}S_9\mathtt{UG} \Rightarrow \mathtt{CAUCAG}S_8\mathtt{GA}S_9\mathtt{UG} \\
&\Rightarrow \mathtt{CAUCAGGGA}S_9\mathtt{UG} \Rightarrow \mathtt{CAUCAGGGAA}S_{10}\mathtt{UUG} \\
&\Rightarrow \mathtt{CAUCAGGGAAG}S_{11}\mathtt{CUUG} \\
&\Rightarrow \mathtt{CAUCAGGGAAGA}S_{12}\mathtt{UCUUG} \\
&\Rightarrow \mathtt{CAUCAGGGAAGAU}S_{13}\mathtt{UCUUG} \\
&\Rightarrow \mathtt{CAUCAGGGAAGAUCUCUUG}.
\end{aligned}
$$

A derivation can be arranged in a tree structure called a *parse tree* (Figure 3, left). A parse tree represents the syntactic structure of a sequence produced by a grammar. For an RNA sequence, this syntactic structure corresponds to the physical secondary structure (Figure 3, right).

$$
\begin{array}{rlll}
P = \{ & S_0 \rightarrow S_1, & S_7 \rightarrow \mathtt{G}\ S_8, & \\
& S_1 \rightarrow \mathtt{C}\ S_2\ \mathtt{G}, & S_8 \rightarrow \mathtt{G}, & \\
& S_1 \rightarrow \mathtt{A}\ S_2\ \mathtt{U}, & S_8 \rightarrow \mathtt{U}, & \\
& S_2 \rightarrow \mathtt{A}\ S_3\ \mathtt{U}, & S_9 \rightarrow \mathtt{A}\ S_{10}\ \mathtt{U}, & \\
& S_3 \rightarrow S_4\ \ S_9, & S_{10} \rightarrow \mathtt{C}\ S_{10}\ \mathtt{G}, & \\
& S_4 \rightarrow \mathtt{U}\ S_5\ \mathtt{A}, & S_{10} \rightarrow \mathtt{G}\ S_{11}\ \mathtt{C}, & \\
& S_5 \rightarrow \mathtt{C}\ S_6\ \mathtt{G}, & S_{11} \rightarrow \mathtt{A}\ S_{12}\ \mathtt{U}, & \\
& S_6 \rightarrow \mathtt{A}\ S_7, & S_{12} \rightarrow \mathtt{U}\ S_{13}, & \\
& S_7 \rightarrow \mathtt{U}\ S_7, & S_{13} \rightarrow \mathtt{C} & \}
\end{array}
$$

Figure 2: This set of productions P generates RNA sequences with a certain restricted structure. $S_0, S_1, \ldots, S_{13}$ are nonterminals; `A`, `U`, `G` and `C` are terminals representing the four nucleotides.

Formally, a context-free grammar G consists of a set of nonterminal symbols N, a terminals alphabet Σ, a set of productions P, and the start symbol S_0. For a nonempty set of symbols X, let X^* denote the set of all finite strings of symbols in X. Every CFG production has the form $S \rightarrow \alpha$ where $S \in N$ and $\alpha \in (N \cup \Sigma)^*$, thus the left-hand side consists of one nonterminal and there is no restriction on the number or placement of nonterminals and terminals on the right-hand side. The production $S \rightarrow \alpha$ means that the nonterminal S can be replaced by the string α. If $S \rightarrow \alpha$ is a production in P, then for any strings γ and δ in $(N \cup \Sigma)^*$, we define $\gamma S \delta \Rightarrow \gamma \alpha \delta$ and we say that $\gamma S \delta$ *directly derives* $\gamma \alpha \delta$ in G. We say the string β can be *derived* from α, denoted $\alpha \stackrel{*}{\Rightarrow} \beta$, if there exists a sequence of direct derivations $\alpha_0 \Rightarrow \alpha_1$, $\alpha_1 \Rightarrow \alpha_2, \ldots, \alpha_{n-1} \Rightarrow \alpha_n$ such that $\alpha_0 = \alpha$, $\alpha_n = \beta$, $\alpha_i \in (N \cup \Sigma)^*$, and $n \geq 0$. Such a sequence is called a *derivation*. Thus, a derivation corresponds to an order of productions applied to generate a string. The grammar generates the language $\{w \in \Sigma^* \mid S_0 \stackrel{*}{\Rightarrow} w\}$, the set of all terminal strings w that can be derived from the grammar.

Our work in modeling RNA uses productions of the following forms: $S \rightarrow SS$, $S \rightarrow aSa$, $S \rightarrow aS$, $S \rightarrow S$ and $S \rightarrow a$, where S is a nonterminal and a is a terminal. $S \rightarrow aSa$ productions describe the base pairings in RNA; $S \rightarrow aS$ and $S \rightarrow a$ describe unpaired bases; $S \rightarrow SS$ describe branched secondary structures and $S \rightarrow S$ (called *skip productions*) are used in the context of multiple alignments.

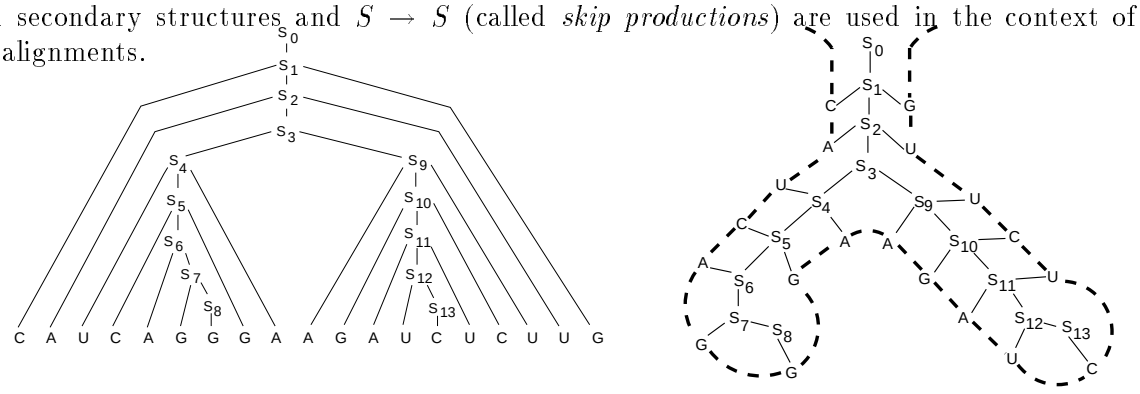


Figure 3: For the RNA sequence `CAUCAGGGAAGAUCUCUUG`, the grammar whose productions are given in Figure 2 yields this parse tree (left), which reflects a specific secondary structure (right).

As in our protein HMM (Krogh *et al.*, 1994), we distinguish two different types of nonterminals: *match* and *insert*. The match nonterminals in a grammar correspond to important structural positions in an RNA or columns in a multiple alignment. Insert nonterminals generate nucleotides in the same way, but have different distributions. These are used to insert extra nucleotides between important (match) positions. Skip productions are used to skip a match nonterminal, so that no nucleotide appears at that position in a multiple alignment (equivalent to deletions).

2.2 Stochastic context-free grammars

In a SCFG, every production for a nonterminal S has an associated probability value such that a probability distribution exists over the set of productions for S. (Any production with the nonterminal S on the left side is called "a production for S.") We denote the associated probability for a production $S \rightarrow \alpha$ by $\mathcal{P}(S \rightarrow \alpha)$.

A stochastic context-free grammar G generates sequences and assigns a probability to each generated sequence, and hence defines a probability distribution on the set of sequences. The probability of a derivation (parse tree) can be calculated as the product of the probabilities of the production instances applied to produce the derivation. The probability of a sequence s is the sum of probabilities over all possible derivations that G could use to generate s, written as follows:

$$\begin{aligned} \text{Prob}(s \mid G) &= \sum_{\substack{\text{all derivations} \\ \text{(or parse trees) } d}} \text{Prob}(S_0 \stackrel{d}{\Rightarrow} s \mid G) \\ &= \sum_{\alpha_1, \ \cdots \ ,\alpha_n} \text{Prob}(S_0 \Rightarrow \alpha_1 \mid G) \cdot \ \text{Prob}(\alpha_1 \Rightarrow \alpha_2 \mid G) \cdot \ \cdots \ \cdot \ \text{Prob}(\alpha_n \Rightarrow s \mid G) \end{aligned}$$

Efficiently computing $\text{Prob}(s \mid G)$ presents a problem because the number of possible parse trees for s is exponential in the length of the sequence. However, a dynamic programming technique analogous to the Cocke-Younger-Kasami or Early parsing methods (Aho & Ullman, 1972) for non-stochastic CFGs can complete this task in polynomial time (specifically, in time proportional to the cube of the length of sequence s). We define the negative logarithm of the probability of a sequence given by the grammar G, $-\log(\text{Prob}(s \mid G))$, as the *negative log likelihood (NLL) score* of the sequence. The NLL score quantifies how well the sequence s fits the grammar—how likely it is that the grammar with its production probabilities could produce the sequence s.

Since CFGs are generally ambiguous in that the grammar gives more than one parse tree for a sequence, and alternative parse trees reflect alternative secondary structures (foldings), a grammar often gives several possible secondary structures for one RNA sequence. An advantage of a SCFG is that it can provide the most likely parse tree from this set of possibilities. If the productions are carefully chosen and the probabilities are carefully designed, the correct secondary structure will appear as the most likely parse tree among the alternatives produced by the grammar G. As discussed in Section 3.2, the most likely parse trees given by the tRNA-trained grammar give exactly the accepted secondary structures for most of the tRNA sequences we test.

We can compute the most likely parse tree efficiently using a variant of the above procedure for calculating $\text{Prob}(s \mid G)$. To obtain the most likely parse tree for the sequence s, we calculate

$$\max_{\text{parse trees } d} \text{Prob}(S_0 \stackrel{d}{\Rightarrow} s \mid G).$$

The dynamic-programming procedure to do this resembles the Viterbi algorithm for HMMs (Rabiner, 1989). We also use this procedure to obtain multiple alignments: the grammar aligns each sequence by finding the most likely parse tree, after which the mutual alignment of the sequences among themselves is determined.

2.3 Estimating SCFGs from sequences

All parameters in the SCFG (the productions in the grammar as well as the production probabilities) could in principle be chosen using an existing alignment of RNA sequences. Results using this approach were reported in our previous work (Sakakibara *et al.*, 1993) and in recent work of Eddy and Durbin (Eddy & Durbin, 1994). However, as is also discussed in those papers, it is possible to estimate many aspects of the grammar directly from unaligned tRNA training sequences. Eddy and Durbin report results in which nearly all aspects of the grammar are determined solely from the training sequences (Eddy & Durbin, 1994). In contrast, we make more use of prior information about the structure of tRNA to design an appropriate initial grammar, and then use training sequences only to refine our estimates of the probabilities of the productions used in this grammar.

2.3.1 The Tree-Grammar EM training algorithm

To estimate the SCFG parameters from unaligned training tRNA sequences, we introduce Tree-Grammar EM, a new method for training SCFGs that is a generalization of the forward-backward algorithm commonly used to train HMMs. Tree-Grammar EM is more efficient than the inside-outside algorithm, which was previously proposed to train SCFGs.

The inside-outside algorithm (Lari & Young, 1990; Baker, 1979) is an Estimation Maximization (EM) algorithm that calculates maximum likelihood estimates of a SCFG's parameters based on training data. However, it requires the grammar to be in Chomsky normal form, which is possible but inconvenient for modeling RNA (and requires more nonterminals). Further, it takes time at least proportional to n^3, whereas the forward-backward procedure for HMMs takes time proportional to n^2, where n is the length of the typical training sequence. There are also many local minima in which the method can get caught, and this presents a problem when the initial grammar is not highly constrained.

To avoid such problems, we have developed a method to obtain a SCFG for an RNA family with an inner loop that takes only time n^2 per training sequence, and hence may be practical on RNA sequences somewhat longer than tRNA. Tree-Grammar EM requires folded RNA as training examples, rather than unfolded ones. Thus, some tentative "base pairs" in each training sequence have to be identified before the inner loop of the algorithm can begin iteratively reestimating the grammar parameters. When actual or trusted base-pair information is not available, base pairs themselves are estimated in the outer loop of our algorithm, as described in Section 2.5.

The Tree-Grammar EM procedure is based on the theory of stochastic tree grammars. Tree grammars are used to derive labeled trees instead of strings. Labeled trees can be used to represent the secondary structure of RNA easily (Shapiro & Zhang, 1990) (see Figure 3). When working with a tree grammar for RNA, one is explicitly working with both the primary sequence and the secondary structure of each molecule. Since these are given explicitly in each training molecule, Tree-Grammar EM does not have to (implicitly) sum over *all* possible interpretations of the secondary structure of the training examples when reestimating the grammar parameters, as

the inside-outside method must do. The Tree-Grammar EM algorithm iteratively finds the best parse for each molecule in the training set and then readjusts the production probabilities to maximize the probability of these parses. The new algorithm also tends to converge faster because each training example is much more informative (Sakakibara *et al.*, 1993).

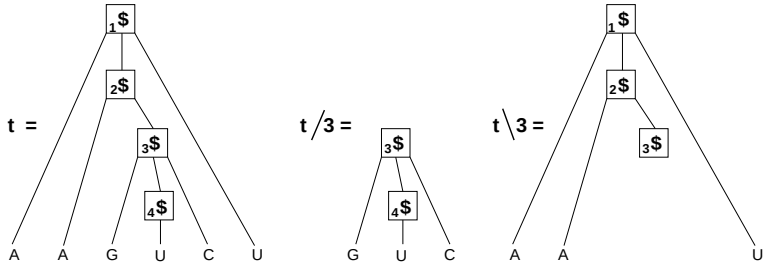


Figure 4: The folded RNA sequence `(AA(GUC)U)` can be represented as a tree t (left), which can be broken into two parts such as $t/3$ (middle) and $t\backslash 3$ (right). The root, ${}_1\$$, and the internal node, ${}_3\$$, represent `A-U` and `G-C` base pairs, respectively.

To avoid unnecessary complexity, we describe this new algorithm in terms of CFGs instead of tree grammars (Thatcher & Wright, 1968; Sakakibara, 1992). A tree is a rooted, directed, connected acyclic finite graph in which the direct successors of any node are linearly ordered from left to right. The predecessor of a node is called the *parent*; the successor, a *child*; and a child of the parent, a *sibling*. A folded RNA sequence can be represented by a labeled tree t as follows. Each leaf node is labeled by one of four nucleotides $\{\mathtt{A}, \mathtt{U}, \mathtt{G}, \mathtt{C}\}$ and all internal nodes are labeled by one special symbol, say \$. The sequence of nucleotides labeled at leaf nodes traced from left to right exactly constitutes the RNA sequence, and the structure of the tree represents its folding structure. See Figure 4 for an example of a tree representation of the folded RNA sequence `(AA(GUC)U)`. We assume all internal nodes in t are numbered from 1 to T (the number of internal nodes) in some order. For an internal node n ($1 \leq n \leq T$), let t/n denote the subtree of t with root n (Figure 4, center) and let $t\backslash n$ denote the tree obtained by removing a subtree t/n from t (Figure 4, right).

The probability of any folded sequence t given by a SCFG $G = (N, \Sigma, P, S_0)$ is efficiently calculated using a dynamic programming technique, as is done with the forward algorithm in HMMs. A labeled tree t representing a folded RNA sequence has the shape of a parse tree, so to parse the folded RNA, the grammar G needs only to assign nonterminals to each internal node according to the productions. Let the quantity $in_n(S)$ define the probability of the subtree t/n given that the nonterminal S is assigned to node n and given grammar G, for all nonterminals S and all nodes n such that $1 \leq n \leq T$. We can calculate $in_n(S)$ inductively as follows:

1. Initialization: $in_n(\mathtt{X}) = 1$, for all leaf nodes n and all terminals `X` (each nucleotide).

   This extension of $in_n(S)$ is for the convenience of the inductive calculation of $in_n(S)$.

2. Induction:

$$in_m(S) = \sum_{\substack{Y_1, \ldots, Y_k \\ \in (N \cup \Sigma)}} in_{n_1}(Y_1) \cdots in_{n_k}(Y_k) \cdot \mathcal{P}(S \rightarrow Y_1 \cdots Y_k),$$

   for all nonterminals S, all internal nodes m and all m's children nodes $n_1, \ldots, n_k$.

3. Termination: For the root node n and the start symbol S_0,

$$\text{Prob}(t \mid G) \;=\; in_n(S_0). \tag{1}$$

This calculation enables us to estimate the new parameters of a SCFG in time proportional to the square of the number of nonterminals in the grammar multiplied by the total size of all the folded training sequences. We need one more quantity, $out_n(S)$, which defines the probability of $t \backslash n$ given that the nonterminal S is assigned to node n and given grammar G, which we obtain similarly.

1. Initialization: For the root node n,

$$out_n(S) \;=\; \begin{cases} 1 & \text{for } S = S_0 \text{ (start symbol)}, \\ 0 & \text{otherwise.} \end{cases}$$

2. Induction:

$$out_m(S) = \sum_{\substack{Y_1,\ldots,Y_k \\ \in (N \cup \Sigma), \\ S' \in N}} in_{n_1}(Y_1) \;\cdots\; in_{n_k}(Y_k) \cdot\; \mathcal{P}(S' \rightarrow Y_1 \;\cdots\; S \;\cdots\; Y_k) \cdot\; out_l(S'),$$

for all nonterminals S, all internal nodes l and m such that l is the parent of m, and all nodes $n_1, \ldots, n_k$ are m's siblings. (There is no termination step given in this case because the calculation of $\text{Prob}(t \mid G)$ is given in the termination step for $in_n(S)$.)

Given a set of folded training sequences $t(1), \ldots, t(n)$, we can determine how well a grammar fits the sequences by calculating the probability that the grammar generates them. This probability is simply a product of terms of the form given by (1), i.e.,

$$\text{Prob}(\text{sequences} \mid G) = \prod_{j=1}^{n} \text{Prob}(t(j) \mid G), \tag{2}$$

where each term $\text{Prob}(t(j) \mid \text{G})$ is calculated as in Equation (1). The goal is to obtain a high value for this probability, called the *likelihood* of the grammar. The *maximum likelihood* (ML) method of model estimation finds the model that maximizes the likelihood (2). There is no known way to directly and efficiently calculate the best model (the one that maximizes the likelihood) without the possibility of getting caught in suboptimal solutions during the search. However, the general EM method, given an arbitrary starting point, finds a local maximum by iteratively reestimating the model such that the likelihood increases in each iteration, and often produces a solution that is acceptable if not optimal. This method is often used in statistics. Here we present a version of the EM method to estimate the parameters of a SCFG from folded training RNA sequences. The inner loop of our Tree-Grammar EM algorithm proceeds as follows:

1. An initial grammar is created by assigning values to the production probability $\mathcal{P}(S \rightarrow Y_1 \;\cdots\; Y_k)$ for all S and all $Y_1, \ldots, Y_k$, where S is a nonterminal and Y_i $(1 \leq i \leq k)$ is a nonterminal or terminal. If some constraints or features present in the folded sequences are known, these are encoded in the initial grammar. The current grammar is set to this initial grammar.

2. Using the current grammar, the values $in_n(S)$ and $out_n(S)$ for each nonterminal S and each node n for each folded training sequence are calculated in order to get a new estimate of each production probability, $\hat{\mathcal{P}}(S \rightarrow Y_1 \ \cdots \ Y_k)$=

$$\frac{\sum_{\text{sequences } t} \left(\sum_{\text{nodes } m} out_m(S) \cdot \mathcal{P}(S \rightarrow Y_1 \ \cdots \ Y_k) \ \cdot \ in_{n_1}(Y_1) \ \cdots \ in_{n_k}(Y_k) \ / \ \text{Prob}(t \mid G) \right)}{\text{norm}},$$

where G is the old grammar and "norm" is the appropriate normalizing constant such that $\sum_{Y_1,\ldots,Y_k} \hat{\mathcal{P}}(S \rightarrow Y_1 \ \cdots \ Y_k) = 1$.

3. A new current grammar is created by replacing $\mathcal{P}(S \rightarrow Y_1 \cdots Y_k)$ with the reestimated probability $\hat{\mathcal{P}}(S \rightarrow Y_1 \cdots Y_k)$.
4. Steps 2 and 3 are repeated until the parameters of the current grammar change only insignificantly.

2.4 Overfitting and regularization

A grammar with too many free parameters cannot be estimated well from a relatively small set of training sequences. Attempts to estimate such a grammar will encounter the problem of *overfitting*, in which the grammar fits the training sequences well, but poorly fits related (test) sequences not included in the training set. One solution is to control the effective number of free parameters by *regularization*. We regularize our grammars by taking a Bayesian approach to the parameter estimation problem, similar to the approach we took in modeling proteins with HMMs (Krogh *et al.*, 1994; Brown *et al.*, 1993a).

Before we began the training of our grammars, we constructed a prior probability density for each of the important sets of parameters in our stochastic grammars. The form of this prior density is that of a Dirichlet distribution (Santner & Duffy, 1989). There were two important types of productions in our CFGs for which we had to estimate probabilities: productions of the form $S \rightarrow aSb$ which generate base pairs (these come in groups of 16, one for each of the 16 possibilities for terminals $a, b \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$), and productions of the form $S \rightarrow aS$ which generate nucleotides in loop regions (these come in groups of four, one for each terminal $a \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$). For the base-pairing productions, we employed sources of prior information about which productions are most likely. For instance, the Watson-Crick pairs are much more frequently observed than other base pairs. In order to calculate more precise prior information about base-pair probabilities, we used a large alignment of 16S rRNA sequences (Larsen *et al.*, 1993), to obtain the 16 parameters of a Dirichlet density over possible base-paired position distributions. We similarly used the alignment to calculate a four-parameter Dirichlet prior for nucleotide distributions in loop region positions. Further details of this method are presented elsewhere (Brown *et al.*, 1993a). We then used these parameters as a regularizer, adding them as "pseudocounts" during each reestimation step of Tree-Grammar EM (Figure 5). This means that at each iteration we compute mean posterior estimates of the parameters of the model rather than maximum likelihood estimates.

The probability distributions for other types of productions of the grammars were also regularized in a Bayesian manner analogous to that in our previous HMM work (Krogh *et al.*, 1994). These include chain rules of the form $S \rightarrow S$, branch productions $S \rightarrow SS$ and productions of the form $S \rightarrow aS$ that are used to insert extra nucleotides into the loop regions to adjust the loop length.

		3′			
		C	G	U	A
5′	C	0.134879	3.403940	0.162931	0.176532
	G	1.718997	0.246768	0.533199	0.219045
	U	0.152039	0.784135	0.249152	2.615720
	A	0.135167	0.192695	1.590683	0.160097

C	G	U	A
0.21	0.18	0.20	0.26

Figure 5: Helix (top) and loop (bottom) pseudocounts are added to actual observed frequencies to reflect prior information. These counts are based upon estimated Dirichlet distributions for helix regions and loop regions. The matrix is asymmetric because the distributions differ with the base ordering in a base pair (ex., 5′ C paired with 3′ G has higher probability than 5′ G paired with 3′ C).

The latter we regularized with very large uniform pseudocounts over the four possible nucleotides so that their probability distributions would be fixed at uniform values rather than estimated from the training data, again as in our previous HMM work (Krogh *et al.*, 1994). This further reduced the number of parameters to be estimated, helping to avoid overfitting.

2.5 Using the new Tree-Grammar EM algorithm

As mentioned above, since Tree-Grammar EM uses folded rather than unfolded RNA for training examples, approximate "base pairs" in each training sequence must be identified before the EM iteration begins. If only unfolded training sequences are available, we iteratively estimate the folding of the training sequences as follows:

1. Design a rough initial grammar that may only represent a portion of the base-pairing interactions and parse the unfolded RNA training sequences to obtain a set of partially folded RNA sequences.

2. Estimate a new SCFG using the partially folded sequences and the inner loop of Tree-Grammar EM. Further productions might be added to the grammar at this stage, although we have not yet experimented with this possibility.

3. Use the trained grammar to obtain more accurately folded training sequences.

4. Repeat Steps 2 and 3 until the folding stabilizes.

2.6 The initial grammar

Represented pictorially in Figure 6 and textually in Figure 7 is the high-level description for the initial grammar we designed for our tRNA experiments (we call it the *meta-grammar*). This meta-grammar is based on tRNA structure previously described (Steinberg *et al.*, 1993b). The meta-grammar text has strings such as "d-arm" and "anti-codon" (we call these *meta-nonterminals*) corresponding to tRNA structures illustrated in Figure 6. Each of these meta-nonterminals has a

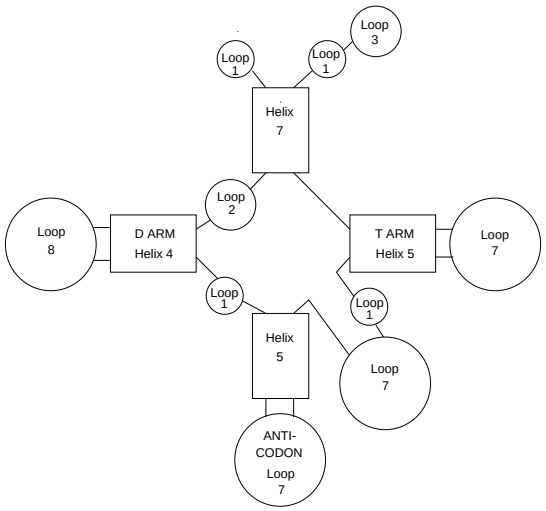


Figure 6: This shows graphically the high-level abstracted description for a desired initial grammar for tRNA. The same description is shown in text form in Figure 7.

further set of productions associated with it (not shown). We have written a program that automatically generates actual productions given only the meta-grammar, greatly simplifying grammar specification.

In Figure 6, BRANCH meta-nonterminals are depicted approximately as lines connecting helices and loops. Each of the remaining meta-nonterminals is either of type LOOP or of type HELIX, and each has an associated length, given by a numeric parameter. For a meta-nonterminal LOOP(n), the grammar generating program creates a subgrammar that is equivalent to an HMM model with n match states as described in our previous work on proteins (Krogh *et al.*, 1994), except that the four-letter alphabet of nucleic acids replaces the twenty-letter alphabet of amino acids. Thus, this subgrammar derives strings with no base pairs that typically have length n, and the distributions of the nucleotides in these strings are defined by the probabilities of the productions for n match nonterminals. Longer or shorter strings can be derived using special nonterminals and productions that allow position-specific insertions and deletions.

For a meta-nonterminal HELIX(n), the grammar generating program creates a subgrammar consisting of n nonterminals, each of which has 16 productions that derive possible base pairs for its position in the helix, each nonterminal having its own probability distribution over these 16 possible productions. These probability distributions, like those above for the match nonterminals in the loops, are initially defined using the Dirichlet priors (Section 2.4). In addition, further nonterminals and productions are added to allow deletions of base pairs, enabling length variations in the helix. Currently this program does not generate special insertion productions to allow for bulges as in the grammars of Durbin and Eddy (Eddy & Durbin, 1994), but it is straightforward to add this capability. There is sometimes a tradeoff between more complex grammars that may better model the data, and simpler grammars that are faster to train and do not overfit the training data. However, in this case it appears that both approaches work well.

Finally, all the subgrammars for the various structures in the model are combined according to the high-level specification to produce the complete initial grammar. At this point additional special treatment of nonterminals involved in branch productions of the form $S \rightarrow SS$ can be included. In particular, we specify that certain branch productions may also, with some probability, omit

```
S                2BranchR      init-ins      rest1
init-ins         Loop          1
rest1            2BranchRNec   molecule      end
end              2BranchL      end1          end3
end1             Loop          1
end3             Loop          3
molecule         Helix         7             rest2
rest2            2BranchR      btw12         rest3
btw12            Loop          2
rest3            2BranchR      d-arm         rest4
d-arm            Helix         4             d-arm-loop
d-arm-loop       Loop          8
rest4            2BranchR      btw23         rest5
btw23            Loop          1
rest5            2BranchRNec   anti-codon    rest6
anti-codon       Helix         5             anti-codon-loop
anti-codon-loop  Loop          7
rest6            2BranchR      variable      rest7
variable         Loop          7
rest7            2BranchL      nothing       t-arm
t-arm            Helix         5             t-arm-loop
t-arm-loop       Loop          7
nothing          Loop          1
```

Figure 7: This meta-grammar was used to generate the productions and probabilities for an initial grammar to model tRNA. The loop and helix descriptions (ex., `Loop 3`) are referred to in small capitals in the text (ex., LOOP(3)).

one of the nonterminals on the right-hand side. This allows the grammar to derive tRNAs that are missing either the D-arm or the T-arm. In general, any substructure in the grammar can be specified to be absent with some probability. These probability values are initialized to default prior values and then reestimated during training on actual sequences, as are all the parameters of the grammar.

3 Experimental results

As described in the previous section, we used Tree-Grammar EM to deduce three trained grammars from training sets of unfolded and unaligned tRNA sequences (Figure 8). A primary training phase was performed to determine the reliability and utility of our Tree-Grammar EM algorithm, and generated `MixedTRNA500` (Sakakibara *et al.*, 1994). This grammar was trained on 500 randomly chosen tRNA from TRUSTED and incorporated only rudimentary knowledge about RNA secondary structure into its initial production probabilities.[2] Nonetheless, it was able to discriminate perfectly between previously unseen complete[3] tRNA and non-tRNA sequences. The experiments discussed

[2]The `MixedTRNA500` grammar was rudimentary in that the initial grammar had simply a uniform distribution over each set of same-type productions (where the types are $S \rightarrow a$, $S \rightarrow a\ S\ a$ and so forth), but with Watson-Crick base pairs weighted twice as heavily.

[3]tRNA sequences that we term Part III, namely those with missing arms, were not included.

here focused on honing the initial grammar used by Tree-Grammar EM—determining which productions and nonterminals should be included and what their initial probabilities should be. Using this refined initial grammar before training (Step 1 of Section 2.5), Tree-Grammar EM produced the remaining three trained grammars: `MT10CY10`, `MT100` and `RandomTRNA618`. `ZeroTrain` is also "trained" in the sense that it embodies pseudocount information as detailed in Section 2.6.

The Tree-Grammar EM algorithm was used to refine the initial grammar with varying numbers of training sequences (Figure 8). The run time for training was around 30 CPU minutes for 100 training sequences on a Sun Sparcstation 10/30 for a single step through the inner loop of Tree-Grammar EM. Finding the best parse for each sequence given a partially trained grammar required 2–3 CPU seconds for a typical tRNA sequence on a DEC AXP 3000/400 running OSF/1. During the training process, only the probabilities of the productions were reestimated and no nonterminals or productions were added or deleted, unlike "model surgery" in our HMM work (Krogh *et al.*, 1994).

3.1 Data

The experiments for generating the trained grammars used data from two sources:

1. We obtained our tRNA training sets from EMBL Data Library's tRNA database maintained by Mathias Sprinzl and co-workers (Steinberg *et al.*, 1993b). In particular, we obtained 1477 aligned and folded sequences for training and testing. We refer to these as TRUSTED alignments and most of these sequences can be fitted into a canonical tRNA structure (Figure 1). The compilation includes tRNAs from virus, archaea, eubacteria, cyanelle, chloroplast, cytoplasm and mitochondria. We changed several specific symbols used for representing modified bases to the usual `A`, `C`, `G` and `U` symbols. We omitted duplicate primary sequences and sequences containing unusual characters to obtain 1222 unique sequences, each between 51 and 93 bases long (Figure 8). Included in this set were 58 tRNAs with unusual secondary structure which are called *Part III* tRNAs in the database. This group includes tRNAs from mitochondria of parasitic worms lacking the T- or D-domain, mammalian mitochondria lacking the D-domain, mitochondria of mollusc, insect and echinoderm with extended anticodon and T-stems, single cell organisms and fungi and *Trypanosoma brucei*. More recently, Steinberg and co-workers (Steinberg *et al.*, 1993a) have updated and revised this tRNA database (we refer to this version as TRUSTEDNEW); these alignments were provided to us by Robert Cedergren.

2. From the National Center for Biotechnology Information's (NCBI) NewGenBank database (version 75.0+, dated 18 February 1993) and GenBank database (version 75.0, dated 10 February 1993), we generated about 2020 non-tRNA test sequences by cutting non-tRNA features—including mRNA, rRNA, and CDS—into tRNA-sized lengths. In particular, we created 20 non-tRNA sequences for each sequence length between 20 to 120 bases.[4]

3.2 Multiple alignments and secondary structure

From a grammar it is possible to obtain a multiple alignment of all sequences. The grammar can produce the most likely parse tree for the sequences to be aligned, yielding an alignment of all the nucleotides that align to the match nonterminals in the grammar. Between match nonterminals

[4]The actual size of the final data set was 2016 because we discarded four anomalous tRNAs that appeared in the set of 2020 non-tRNAs through unusual labeling in GenBank. We discovered these when the trained grammars "misclassified" them in discrimination experiments.

Data Set	Type of tRNA	Total	Number of Sequences ZT	MT10CY10	MT100	R618
ARCHAE	archaea	103	0	0	0	50
CY	cytoplasm	230	0	10	0	100
CYANELCHLORO	cyanelle and chloroplast	184	0	0	0	100
EUBACT	eubacteria	201	0	0	0	100
VIRUS	viruses	24	0	0	0	10
MT	mitochondria	422	0	10	100	200
PART III	Part III	58	0	0	0	58
	Totals	1222	0	20	100	618

Figure 8: We organized the tRNA sequences (Steinberg *et al.*, 1993b) into seven groups and then used randomly chosen subsets of these groups to train and test our three trained grammars. The `ZeroTrain` grammar (abbreviated `ZT`) was trained on no tRNA sequences, but was invested with prior information about tRNA (Section 2.6). `RandomTRNA618` (abbreviated `R618`) was trained on the most tRNA sequences—about half the total sequences per group and all of the Part III.

there can be insertions of varying lengths, but by inserting enough spaces in all the sequences to accommodate the longest insertion, an alignment is obtained.

Once the `RandomTRNA618` grammar was completed, a multiple alignment was produced for the entire set of 1222 tRNA sequences. The TRUSTED alignment agrees substantially with the trained grammar's predicted alignment. Boundaries of helices and loops are the same; the major difference between the two alignments is the extra arm, which is highly variable in its length and sequence. Figure 9 shows the TRUSTED alignment of selected tRNA sequences with the alignment predicted by the trained grammar `RandomTRNA618` for the same sequences.

To assess the accuracy of the four trained grammars' predicted foldings, for each set of sequences, we counted the fraction of base pairs specified by the TRUSTED alignment that matched in our grammars' predicted multiple alignments. These counts are tabulated by group (rows) and grammar (columns) in Figure 11. In the sequence sets ARCHAE and VIRUS, every one of the three trained grammars captures all the base pairing present in TRUSTED. In the case of CY, CYANELCHLORO, EUBACT and MT, the agreement between TRUSTED and grammar-predicted base pairings is extremely good, but for PART III it is considerably poorer. We examined in detail all cases where the fraction of base pairs specified by TRUSTED that matched in our predicted alignment was less than 100% for `MT10CY10`, `MT100` and `RandomTRNA618`. The results are summarized in Figure 17. (Six mammalian mitochdondrial serine tRNAs/tDNA sequences with anticodon `UGA/TGA` (sequences denoted with `$`) were included in PART III in TRUSTED but were reclassified as MT in TRUSTEDNEW.) One EUBACT, two CYANELCHLORO, 12 MT and 30 PART III sequences were so "misaligned" by all three grammars (sequences with ... in the three columns under `Align`). It can be seen that disagreements are not distributed globally across the entire length of the sequence, but are confined to specific helices (note the large number of `<==>` "helices"). In some sequences, the misalignment merely reflects differences in location of a gap between TRUSTED and grammar

```
           [     ]  <     D-domain    > <   Anticodon   >< Extra ><    T-domain   >[     ]
           (((((((  ((((           )))) (((((  ===  )))))         (((((       )))))))))))
1 DC0380  -GCCAAGGTGGCAGAGTTCGGCCTAACGCGGCGGCCTGCAGAGCCGCTC----ATCGCCGGTTCAAATCCGGCCCTTGGCT---
2 DA6281  -GGGCGTGTGGCGTAGTC-GGT--AGCGCGCTCCCTTAGCATGGGAGAG----GTCTCCGGTTCGATTCCGGACTCGTCCA---
3 DE2180  --GCCCCATCGTCTAGA--GGCCTAGGACACCTCCCTTTCACGGAGGCG----A-CGGGGATTCGAATTCCCCTGGGGGTA---
4 DC2440  -GGCGGCATAGCCAAGC--GGT--AAGGCCGTGGATTGCAAATCCTCTA----TTCCCCAGTTCAAATCTGGGTGCCGCCT---
5 DK1141  -GTCTGATTAGCGCAACT-GGC--AGAGCAACTGACTCTTAATCAGTGG----GTTGTGGGTTCGATTCCCACATCAGGCACCA
6 DA0260  -GGGCGAATAGTGTCAGC-GGG--AGCACACCAGACTTGCAATCTGGTA----G-GGAGGGTTCGAGTCCCTCTTTGTCCACCA
7 DA3880  -GGGGCTATAGTTTAACT-GGT--AAAACGGCGATTTTGCATATCGTTA----T-TTCAGGATCGAGTCCTGATAACTCCA---
8 DH4640  -AGCTTTGTAGTTTATGTG-----AAAATGCTTGTTTGTGATATGAGTGAAAT--------------------TGGAGCTT---

           (((((((  ((((            )))) (((((  ===  )))))        (((((       ))))))))))))
1 DC0380  -GCCAAGGUGGCAG.AGUUcGGccUAACGCGGCGGCCUGCAGAGCCGCUC---AUCGCCGGUUCAAAUCCGGCCCUUGGCU---
2 DA6281  -GGGCGUGUGGCGU.AGUC.GG..UAGCGCGCUCCCUUAGCAUGGGAGAGG---UCUCCGGUUCGAUUCCGGACUCGUCCA---
3 DE2180  -GCCCC-AUCGUCU.AGAG.GCc.UAGGACACCUCCCUUUCACGGAGGCG----ACGGGGAUUCGAAUUCCCCU-GGGGGU--A
4 DC2440  -GGCGGCAUAGCCA.AGC-.GG..UAAGGCCGUGGAUUGCAAAUCCUCUA---UUCCCCAGUUCAAAUCUGGGUGCCGCCU---
5 DK1141  -GUCUGAUUAGCGC.AACU.GG..CAGAGCAACUGACUCUUAAUCAGUGGG---UUGUGGGUUCGAUUCCCACAUCAGGCACCA
6 DA0260  -GGGCGAAUAGUGUcAGCG.GG..-AGCACACCAGACUUGCAAUCUGGUA----GGGAGGGUUCGAGUCCCUCUUUGUCCACCA
7 DA3880  -GGGGCUAUAGUUU.AACU.GG..UAAAACGGCGAUUUUGCAUAUCGUUA----UUUCAGGAUCGAGUCCUGAUAACUCCA---
8 DH4640  -AGCUUUGUAGUUU.A--U.GU..GAAAAUGCUUGUUUGUGAUAUGAGUGA--AAU-----------------UGGAGCUU---
```

Figure 9: Shown are two sets of alignments of several representative tRNAs identified by their database code. The top set is from TRUSTED (Steinberg *et al.*, 1993b); the bottom set was produced by trained grammar `RandomTRNA618`. Parentheses indicate which columns (positions) form base pairs (`===` locates the anticodon). "`[`" and "`]`" denote the 5′ and 3′ sides of the acceptor helix, respectively. For `RandomTRNA618`, capital letters correspond to nucleotides aligned to the match nonterminals of the grammar, lowercase to insertions, `-` to deletions by skip prodution and `.` to fill characters required for insertions. The sequences are from the seven groups in Figure 8: 1. ARCHAE (*Halobacterium cutirubrum*), 2. CY (*Saccharomyces cerevisiae*), 3. CYANELCHLORO (*Cyanophora paradoxa*), 4. CYANELCHLORO (*Chlamydomonas reinhardtii*), 5. EUBACT (*Mycoplasma capricolum*), 6. VIRUS (phage T5), 7. MT (*Aspergillus nidulans*) and 8. PART III (*Ascaris suum*).

alignments in one or both sides of a helix. Other instances are examples of alternative, but equally plausible, base-pairing schemes in the various helices (indicated by `<::>`). However, there are cases where the grammar-generated alignments suggest (small) improvements over the TRUSTED alignments, principally in the base pairing of the D- or T-helices. A selection of such sequences is shown in Figure 10. A notable example are the PART III class of mammalian mitochondrial tRNAs lacking the D-domain and mollusc, insect and echinoderm mitochondrial tRNAs with extended anticodons and T-stems. Here, readjustment of residues in the 5′ side of the T-helix and flanking unpaired residues would create a T-stem with a greater number of Watson-Crick base pairs than in TRUSTED. It should be noted that in both the mammalian and parastic worm mitochondrial PART III sequences that lack the D-domain, the 5′ side of the D-stem is absent in the TRUSTED alignments. Interestingly, these tRNAs lacking the D-domain are the only sets that can neither be "aligned" in the same manner as TRUSTED nor discriminated from non-tRNAs (see Section 3.4).

The alignments produced by the three trained grammars were also compared to those in the revised and updated alignments TRUSTEDNEW. In these cases as well, the predicted alignments were nearly identical to the trusted alignments. Base-pair counts also were very similar to those

```
             [     ]            <    D-domain     >     <     Anticodon domain     >           <     T-domain     > [     ]
Base pairing (((((((            ((((           ))))     (((((                 )))))           (((((          ))))) )))))))
 1 Trusted  {GGGCUAU}---------ua{GCUC}agcgguaga{gcgc}--g{CGCCC}---------cugauaa{GGGCG}-----agguc{UCUGG}-uucaaau{CCAGG}{AUAGCCC}a---
   MT100**  {GGGCUAU}---------ua{GCUC}--agcggua{gagc}gcg{CGCCC}---------cugauaa{GGGCG}-----agguc{UCUGG}-uucaaau{CCAGG}{AUAGCCC}a---
 2 Trusted  {GCCCCUA}---------ua{GUUG}---aaacac{aacc}--a{AGAGC}---------uuuucac{GCUCU}-----uaagu{UUGAG}-uuaaaau{CUCAA}{UAGGAGC}u---
   MT100**  {GCCCCUA}---------ua{GUUG}----aaaca{caac}-ca{AGAGC}---------uuuucac{GCUCU}-----uaagu{UUGAG}-uuaaaau{CUCAA}{UAGGAGC}u---
 3 Trusted  {GUUUCAU}---------ga{GUAU}-----agca{GUAC}--a{UUCGG}---------cuuccaa{CCGAA}-----agguu{uuugu}-aaacaac{CAAAA}{AUGAAAU}a---
   MT100**  {GUUUCAU}---------ga{GUAU}-----agca{GUAC}--a{UUCGG}---------cuuccaa{CCGAA}------aggu{uuuug}uaaacaac{CAAAA}{AUGAAAU}a---
 4 Trusted  {aggacgu}---------ua{aaua}---gauaag{CUAU}--g{CCUAG}---------uuacggu{CUGGG}---aagagag{-----}--------{-----}{ucgucuu}u---
   MT100*   {aggacgu}--------uaa{auag}----auaag{CUAU}--g{CCUAG}---------uuacggu{CUGGG}---aagagag{-----}--------{-----}{ucgucuu}u---
   MT10CY10 {ag-gacg}-------uuaa{auag}----auaag{CUAU}--g{CCUAG}---------uuacggu{CUGGG}--aagagagu{-----}--------{-----}{cguc-uu}u---
 5 Trusted  {aacgagu}----------u{caua}---------{--aa}--g{CAAGU}---------cuucuaa{AUUUG}------uucu{-agg-}--uuaaau{--ccu}{gcucguu}u---
   MT100*   {aacgagu}----ucauaaa{----}---------{----}--g{CAAGU}---------cuucuaa{AUUUG}------uucu{--agg}--uuaaau{ccu--}{gcucguu}u---
   RND618   {aacga-g}---uucauaaa{----}---------{----}--g{CAAGU}---------cuucuaa{AUUUG}-------uuc{-uagg}--uuaaau{ccug-}{c-ucguu}u---
 6 Trusted  {AAGAAAG}-----------{----}---------{auug}--c{AAGAA}---------cugcuaa{UUCAU}---gcuucca{ug-uu}--uaaaaa{CAUGG}{CUUUCUU}a---
   MT100**  {AAGAAAG}-------auug{----}---------{----}--c{AAGAA}---------cugcuaa{UUCAU}------gcuu{ccaug}uuuaaaaa{CAUGG}{CUUUCUU}a---
 7 Trusted  {GAGAAAG}-----------{----}---------{cuca}--c{aagaa}---------cugcuaa{cucau}---gccccca{ug-uc}--uaacaa{CAUGG}{CUUUCUC}acca
   MT100*   {GAGAAAG}-------cuca{----}---------{----}--c{aagaa}---------cugcuaa{cucau}------gccc{ccaug}ucuaacaa{CAUGG}{CUUUCUC}acca
   RND618   {GAGAAAG}---------cuc{----}---------{----}--a{caaga}--------acugcuaac{ucaug}--------ccc{ccaug}ucuaacaa{CAUGG}{CUUUCUC}acca
 8 Trusted  {-aaaucu}-----------{auu-}----gguuu{accu}---{UAGUC}---------cugcuaa{GUCUA}---aaggcuu{gcggu}-ucaaucc{cguug}{aguuuuc}----
   MT100**  {aaaucua}---------uu{ggu-}-------uu{-acc}--u{UAGUC}---------cugcuaa{GUCUA}----aaggcu{ugcgg}-uucaauc{ccguu}{gaguuuu}c---
 9 Trusted  {GAAAUAU}-----------{guu-}-----gauc{-aag}---{AAAAG}---------cugcuaa{CUUUU}----ucuuua{auggu}-uuaauuc{cauua}{uauuucu}-cca
   MT100    {GAAAUAU}----------g{-uug}-------au{caa-}--g{AAAAG}---------cugcuaa{CUUUU}-----ucuuu{aaugg}-uuuaauu{ccauu}{auauuuc}ucca
   MT10CY10*{GAAAUAU}----------g{uug-}-------au{-caa}--g{AAAAG}---------cugcuaa{CUUUU}-----ucuuu{aaugg}-uuuaauu{ccauu}{auauuuc}ucca
10 Trusted  {GAAAAAG}----------u{caug}---gaggcc{augg}--g{GUUGG}---------cuugaaa{CCAGC}------uuug{GGGGG}-uucgauu{CCUUC}{CUUUUUU}g---
   MT100**  {GAAAAAG}---------uc{augg}------agg{ccau}ggg{GUUGG}---------cuugaaa{CCAGC}------uuug{GGGGG}-uucgauu{CCUUC}{CUUUUUU}g---
11 Trusted  {AAAAUUA}---------ua{uauu}---uucuag{uuug}--a{ucgaa}---------aaugcuu{uucga}uuugaaaauu{uaaau}-uaaauuu{AAGUU}{UAAUUUU}c---
   MT100*   {AAAAUUA}---------ua{uauu}uucuaguuu{gauc}--g{aaaau}gcuuuucgauuugaaa{auuua}------aauu{aaauu}-------u{AAGUU}{UAAUUUU}c---
   MT10CY10 {AAAAUUA}uauauuuucua{guuu}---gaucga{aaau}--g{cuuuu}---------cgauuug{aaaau}---uuaaauu{aaauu}-------u{AAGUU}{UAAUUUU}c---
```

Figure 10: This figure lists alignments of sequences selected from Figure 17. TRUSTED (Steinberg *et al.*, 1993a) is shown first, followed by alignments produced by `MT100`, `MT10CY10` and `RandomTRNA618`. An asterix indicates sequences where two grammars produced identical alignments. Nucleotides in curly braces correspond to those present in one side of a helix; when the TRUSTED and grammar alignments differ, they are shown in lowercase. Unpaired nucleotides are depicted in lowercase and are *not* shown aligned in this figure. The sequences are 1. DI2620, 2. DE5080, 3. DG5000, 4. DR4640, 5. DS4680, 6. DS5321, 7. RS5880, 8. DS5041, 9. RS4800, 10. DS5880 and 11. DA3681.

Sequence Set	ZeroTrain	MT10CY10	MT100	RandomTRNA618
ARCHAE	94.87%	100.00%	100.00%	100.00%
CY	98.28%	99.76%	99.89%	99.87%
CYANELCHLORO	96.22%	99.64%	99.64%	99.79%
EUBACT	99.69%	99.86%	99.86%	99.86%
VIRUS	96.83%	100.00%	100.00%	100.00%
MT	89.19%	98.33%	98.91%	98.93%
PART III	55.98%	81.10%	83.21%	83.00%

Figure 11: Shown for each tRNA class and each grammar are the fraction of base pairs specified by the TRUSTED alignment that matched in our grammars' predicted multiple alignments. For comparison, the first column shows statistics for the pre-training initial grammar ZeroTrain.

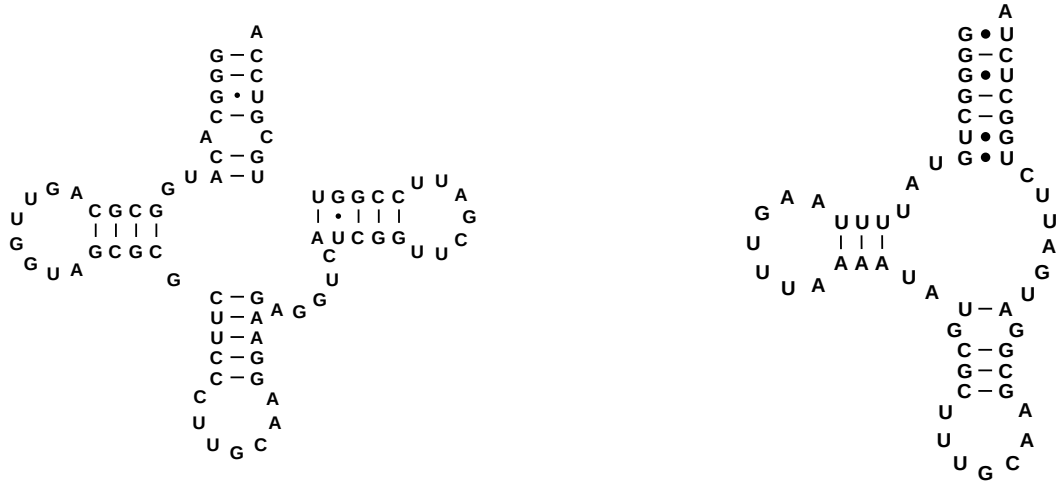


Figure 12: After training on 618 randomly chosen tRNA examples (Figure 8), the RandomTRNA618 grammar produced these foldings for two unaligned and unfolded tRNA sequences: a CY (left) and a Part III (right). XRNA generated these diagrams.

reported for TRUSTED in Figure 11.

3.3 Displaying folded RNA sequences

XRNA is an X Windows-based program for editing and display of RNA primary, secondary and tertiary structure (Weiser *et al.*, 1993). Using simple filters, we were able to transform the secondary structure predicted by our trained grammars into XRNA format. Figure 12 shows two foldings predicted by the RandomTRNA618 grammar: a CY tRNA and a Part III tRNA.

3.4 Discriminating tRNAs from non-tRNAs

As described in Section 2.2, we calculate a NLL score for each test sequence and use it to measure how well the sequence fits the grammar. This raw NLL score depends too much on the test sequence's length to be used directly to decide whether a sequence belongs to the family modeled by the grammar. We normalize the raw scores by calculating the difference between the NLL score of a sequence and the average NLL score of a typical non-tRNA sequence of the same length measured in standard deviations. This number is called the *Z score* for the sequence (Krogh *et al.*, 1994). We then choose a Z-score cutoff, and sequences with Z scores above the cutoff are classified as tRNAs. While we cannot prove that our normalized scores actually exhibit Gaussian tails for non-tRNAs, this kind of Gaussian approximation has worked well previously (Krogh *et al.*, 1994).

To test the ability of our grammars to discriminate tRNA from other RNA sequences of similar length, for each of our trained grammars, we computed the Z score of every sequence in our tRNA database and every sequence in our set of 2016 non-tRNAs. Although the highest Z score of any non-tRNA is never much greater than 4, we do not consider a tRNA sequence to be succesfully discriminated from the non-tRNAs unless its Z score is greater than 5. For each grammar, Figure 13 shows the number of tRNAs in each family that are succesfully discriminated from the non-tRNAs using this criterion. Figures 14, 15, and 16 are histograms of the Z scores for selected grammars.

The results show that training on as few as 20 sequences yields a dramatic improvement in discrimination over what is acheived with an untrained grammar. Note in particular how the Z-scores histogram for non-MT, non-PART III tRNAs "slides" from left to right when the grammar producing the most likely parses is `ZeroTrain` versus when it is `MT10CY10` (Figure 14); whereas the `ZeroTrain` grammar shows poor discrimination between non-MT and non-PART III tRNA from non-tRNA, the `MT10CY10` already shows perfect discrimination.

The `MT10CY10` grammar also does well in the more difficult task of discriminating mitochondrial tRNA from non-tRNA. Setting aside the Part III sequences, `MT10CY10` is able to discriminate 399 out of 422 mitochondrial sequences from non-tRNA, performing nearly as well as the grammars trained on many more tRNA sequences, `MT100` and `RandomTRNA618` (Figures 13 and 15). However, good discrimination of the Part III sequences from non-tRNA sequences is not acheived by any of the grammars, even the `RandomTRNA618` grammar, which is trained on these sequences. It can be seen from the histograms in Figure 16 that this training improves discrimination of some Part III sequences, but half of these sequences still have Z scores below 5. Figure 17 tabulates all Part III tRNA sequences and all other tRNA sequences that were scored below the Z-score cutoff of 5 by some trained grammar or that were incorrectly aligned (with respect to the TRUSTED alignment) by all three trained grammars.

A total of 29 PART III tRNAs could not be discriminated from non-tRNA sequences by either `MT100`, `MT10CY10` or `RandomTRNA618` (8 of these have a Z score between 4 and 5 in at least one grammar). Interestingly, 19 of the 29 sequences could neither be discriminated nor "aligned" by all three grammars in the same manner as TRUSTED (Section 3.2 and see Figure10). All but three of these sequences are mammalian and parasitic tRNAs that lack the D-domain. However, the grammars are able to discriminate PART III tRNAs lacking the T-domain. Overall, the trained grammars are able to generalize well in that they require few training examples to perform discrimination. As can be seen from the case of PART III tRNAs, however, a grammar clearly gains discriminative power from being trained on a large and varied sequence set.

	Number of Sequences with Z Scores											
	Above 5 Standard Dev.				Between 4 and 5 Std. Dev.				Below 4 Standard Dev.			
Test Set	ZT	MT10	MT100	R618	ZT	MT10	MT100	R618	ZT	MT10	MTh	R618
ARCHAE	66	103	103	103	19	0	0	0	18	0	0	0
CY	135	230	230	230	53	0	0	0	42	0	0	0
CYANELCHLORO	61	184	184	184	52	0	0	0	71	0	0	0
EUBACT	160	201	201	201	30	0	0	0	11	0	0	0
VIRUS	16	24	24	24	4	0	0	0	4	0	0	0
MT (train)	N/A	10	99	193	N/A	0	1	6	N/A	0	0	1
MT (test)	64	389	313	218	89	10	7	3	269	13	2	1
PART III	0	9	7	29	1	15	14	8	57	34	37	21
NON-TRNA	0	0	0	0	0	0	1	1	2016	2016	2015	2015
Totals	502	1150	1161	1182	248	25	23	18	2488	2063	2054	2038

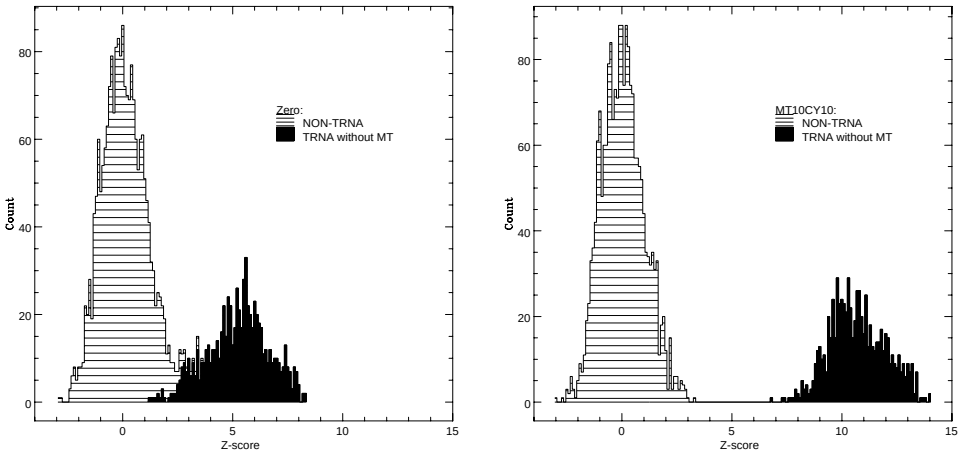
	Z Scores for Boundary Sequences			
	ZeroTrain	MT10CY10	MT100	RandomTRNA618
Highest NON-TRNA	3.954	3.341	4.018	4.080
Lowest non-MT non-Part III tRNA	1.220	6.791	6.211	8.759
Group of the lowest tRNA	CYANELCHLORO	CYANELCHLORO	CY	CY

Figure 13: The top table shows how each grammar partitions the 3238 total sequences (1222 tRNA and 2016 non-tRNA) based on their Z scores. The columns correspond to the four grammars (`ZeroTrain` is abbreviated as `ZT`, `MT10CY10` as `MT10` and `RandomTRNA618` as `R618`) such that the sum of each grammar's three "Totals" entries is 3238. The first grouping of four columns indicates the number of tRNAs correctly discriminated from non-tRNA by each grammar. (Because all three grammars perfectly discriminated all nonmitochondrial tRNA sequences, only the results for mitochondrial tRNA sequences are partitioned into separate discrimination results for the training and test sets.) The bottom table shows the Z scores for the highest-scoring non-tRNA sequence and lowest-scoring tRNA sequence (excluding the mitochondrial and Part III tRNA sequences), listing the group to which the lowest-scoring tRNA belongs, for each grammar.

4 Discussion

The method we have proposed represents a significant new direction in computational biosequence analysis. SCFGs provide a flexible and highly effective statistical method for solving a number of RNA sequence analysis problems including discrimination, multiple alignment and prediction of secondary structures. In addition, the grammar itself may be a valuable tool for representing an RNA family or domain. The present work demonstrates the usefulness of SCFGs with tRNA sequences and could prove useful in maintaining, updating and revising compilations of their alignments. For example, our results suggest potential improvements in the aligments of the D- and T-domains in mitochondrial tRNAs from parastic worms and mammals that lack the D-domain, and mollusc, insect and echinoderm tRNAs with extended T-stems. Further classes of RNA sequences potentially appropriate to model using this method include group I introns (Michel &

Figure 14: These histograms show discrimination of 2016 non-tRNA sequences from 742 tRNA sequences (excluding MT and PART III sequences) for the untrained `ZeroTrain` (left) and trained `MT10CY10` (right) grammars. `MT10CY10` is trained on only 20 sequences.



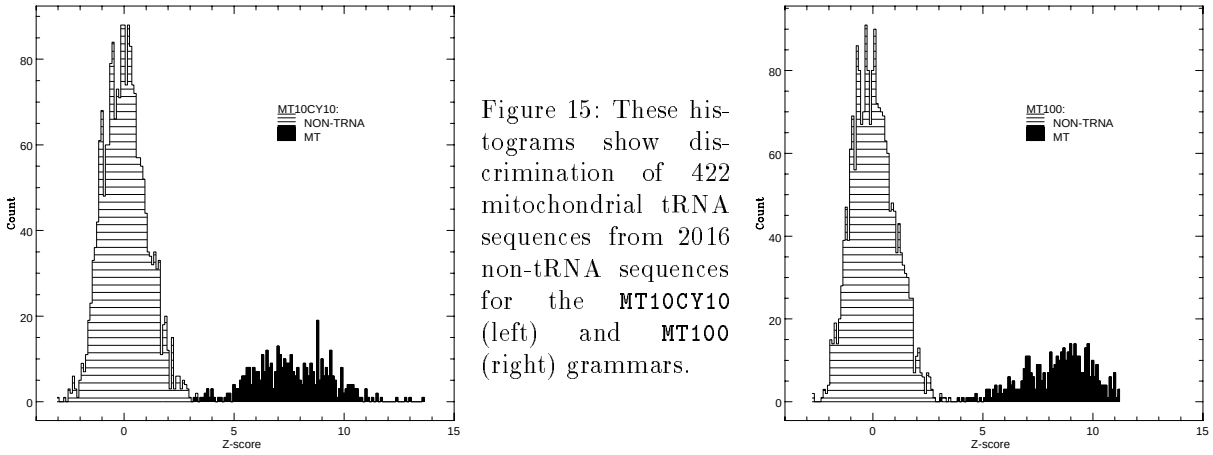


Figure 15: These histograms show discrimination of 422 mitochondrial tRNA sequences from 2016 non-tRNA sequences for the `MT10CY10` (left) and `MT100` (right) grammars.

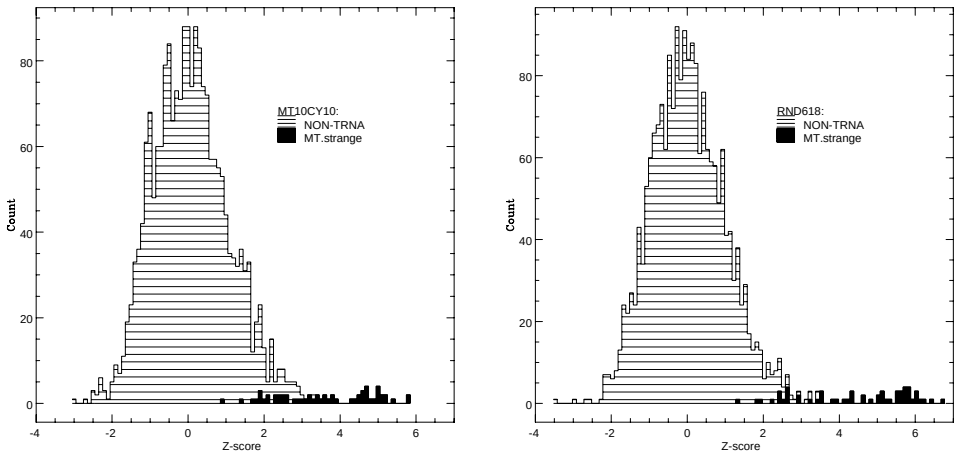


Figure 16: These histograms show discrimination of 58 Part III tRNA sequences from 2016 non-tRNA sequences for the `MT10CY10` grammar (left) and the `RandomTRNA618` grammar (right). Part III sequences are labeled `MT.strange` in the diagrams.

```
        Disc.  Align.   Grammar
                        MT10CY10          MT100             RandomTRNA618     Anticodon & Organism
EUBACT (Eubacteria)
 DL1141  # # #  . . .    <::><+=><==><==>  <::><+=><==><==>  <::><+=><==><==>  TAG MYCOPLASMA CAPRIC.


CYANELCHLORO (Chloroplast)
 DA2620  # # #  . . .    <::><==><==><==>  <=:><==><==><==>  <::><==><==><==>  TGC COLEOCHAETE ORBIC.
 DI2620* # # #  . . .    <==><=+><==><==>  <==><=+><==><==>  <-->><=+><==><==>  GAT COLEOCHAETE ORBIC.

MT (Animal mitochondria)
 DC5080  . % %  A A A                                                          GCA STRONGYLOCEN.PURP.
 DC5100  . % %  A A A                                                          GCA GADUS MORHUA
 DC5120  . . %  A A A                                                          GCA XENOPUS LAEVIS
 DC5160  % # #  A A A                                                          GCA RANA CATESBEIANA
 DD5000  . % .  A A A                                                          GTC ASTERINA PECTINI.
 DE5080* # # #  . . .    <==><=+><==><==>  <==><=+><==><==>  <==><=+><==><==>  TTC STRONGYLOCEN.PURP.
 DG5000* # # #  . . .    <==><==><==><+=>  <==><==><==><+=>  <==><==><==><+=>  TCC ASTERINA PECTINI.
 DG5020  . % .  . . .    <==><--><==><==>  <==><::><==><==>  <=-><--><==><::>  TCC ASTERIAS FORBESII
 DH4880  . # #  A A A                                                          GTG DROSOPHILA YAKUBA
 DH5440  % # #  A . .                                                          GTG MACACA FUSCATA
 DK5280  . # %  A A A                                                          TTT RAT
 DK5281  % # #  A A A                                                          TTT RAT
 DK5320  % # #  A . .                                                          TTT MOUSE
 DL5081  # # #  . . .    <::><==><==><::>  <::><==><==><::>  <::><==><==><::>  TAA STRONGYLOCEN.PURP.
 DN5320  # # #  . . .    <==><::><==><==>  <==><==><==><==>  <==><::><==><==>  GTT MOUSE
 DP5360  % # #  A A A                                                          TGG BOVINE
 DQ5080  # # #  . . .    <::><==><==><==>  <::><==><==><::>  <::><==><==><==>  TTG STRONGYLOCEN.PURP.
 DR4880  . % %  A A A                                                          TCG DROSOPHILA YAKUBA
 DT4980  % # #  . . .    <-->><::><-->><++>  <==><::><-->><++>  <-->><::><==><++>  TGT PISASTER OCHRACEUS
 DT5880  % # #  A A A                                                          TGT HUMAN
 DV4980  % # #  A A A                                                          TAC PISASTER OCHRACEUS
 DV5040  # # #  . . .    <==><=:><==><==>  <==><=:><==><==>  <==><=:><==><==>  TAC PARACENTROTUS LIV.
 DV5080  # # #  . . .    <==><=:><==><==>  <==><=:><==><==>  <==><=:><==><==>  TAC STRONGYLOCEN.PURP.
 DW5020  % # #  . . .    <==><--><+=><==>  <==><==><+=><==>  <==><::><+=><==>  TCA ASTERIAS FORBESII
 DW5280  . # %  A A A                                                          TCA RAT
 DW5281  . % %  A A A                                                          TCA RAT
 DW5320  . # %  A . A                                                          TCA MOUSE

MT (Single cell or fungal mitochondria)
 DC3920  # # #  . . .    <==><++><-->><::>  <==><++><==><::>  <==><--><==><::>  GCA NEUROSPORA CRASSA
 DS3960  # # #  . . .    <==><--><==><==>  <==><--><==><==>  <==><--><==><==>  GCT PODOSPORA ANSERINA
 DX3720  . . %  A A .                                                          CAT PARAMECIUM PRIM.
 DX3800  . % #  A . .                                                          CAT TETRAHYMENA PYRIF.
 DX3840  % % #  A A A                                                          CAT TETRAHYMENA THERM.
```

Figure 17: This table shows all Part III sequences and all other sequences that either were below the Z-scores cutoff of 5 for some grammar, or were "incorrectly" aligned by all three grammars. The three columns each under `Disc.`, `Align.` and `Grammar` are ordered `MT10CY10`, `MT100` and `RandomTRNA618` (left to right). The first column lists the identifier; the last two columns list the anticodon and source organism. The three `Disc.` columns indicate which grammars correctly discriminated the sequence from non-tRNA sequences (Z score > 5, `#`), which grammars did not discriminate the sequence from non-tRNA (Z score < 5, "."), and grammars for which the Z-score was between 4 and 5 (`%`). The three `Align.` columns show which grammars aligned the sequence the same as the EMBL trusted alignment TRUSTED(`A`), and which grammars did not (continued)

```
        Disc.  Align.   Grammar
                        MT10CY10          MT100             RandomTRNA618     Anticodon & Organism
PART III (Parasitic worm mitochondrial tRNAs lacking the T-domain)
 DA4640  . . #  A A A                                                          TGC ASCARIS SUUM
 DC4640  . . %  A A A                                                          GCA ASCARIS SUUM
 DC4680  % % #  A A A                                                          GCA CAENORHABDI.ELEG.
 DD4640  # % #  A A A                                                          GTC ASCARIS SUUM
 DD4680  # # #  A A A                                                          GTC CAENORHABDI.ELEG.
 DE4640  . . #  A A A                                                          TTC ASCARIS SUUM
 DF4640  . . .  A A A                                                          GAA ASCARIS SUUM
 DG4640  % % #  A A A                                                          TCC ASCARIS SUUM
 DG4680  # # #  A A A                                                          TCC CAENORHABDI.ELEG.
 DH4640  % . %  A A A                                                          GTG ASCARIS SUUM
 DH4680  . . #  A A A                                                          GTG CAENORHABDI.ELEG.
 DI4640  # % #  A A A                                                          GAT ASCARIS SUUM
 DK4640  % % #  A A A                                                          TTT ASCARIS SUUM
 DK4680  # # #  A A A                                                          TTT CAENORHABDI. ELEG.
 DL4640  % # #  A A A                                                          TAG ASCARIS SUUM
 DL4641  . . #  A A A                                                          TAA ASCARIS SUUM
 DL4680  % # #  A A A                                                          TAG CAENORHABDI. ELEG.
 DL4681  % . #  A A A                                                          TAA CAENORHADBI. ELEG.
 DW4640  % . #  A . .                                                          TGG ASCARIS SUUM
 DW4680  # % #  A A A                                                          GTT CAENORHABDI. ELEG.
 DP4640  % % #  A A A                                                          TGG ASCARIS SUUM
 DQ4640  . . #  A A A                                                          TTG ASCARIS SUUM
 DR4640* . . .  . . .    <::><++><==><==>  <==><++><==><==>  <==><++><==><==>  ACG ASCARIS SUUM
 DR4680  . . %  A . .                                                          ACG CAENORHABDI. ELEG.
 DT4640  . . %  A A A                                                          TGT ASCARIS SUUM
 DT4680  # % #  A A A                                                          TGT CAENORHABDI.ELEG.
 DV4640  % . #  A A A                                                          TAC ASCARIS SUUM
 DW4640  % % #  A A A                                                          TCA ASCARIS SUUM
 DW4680  % % #  A A A                                                          TCA CAENORHABDI. ELEG.
 DX4640  % % #  A A A                                                          CAT ASCARIS SUUM
 DX4680  % % #  A A A                                                          CAT CAENORHABDI.ELEG.
 DY4640  # % #  A A A                                                          GTA ASCARIS SUUM
```

Figure 17: align (".") the sequence the same. The three `Grammar` columns indicate how the alignments produced by all three grammars differed from TRUSTED. The $5'$ and $3'$ sides of each of the four helices in a typical tRNA are represented as a pair of symbols enclosed in angled brackets `<` and `>`. These "helices" are ordered acceptor arm, D-arm, anticodon arm and T-arm, respectively (left to right). For each, we codify the difference between each grammar's predicted alignment and the trusted alignment as follows: `-` means the predicted alignment is worse than TRUSTED; `=`, identical to TRUSTED; `:`, equivalent to TRUSTED (shift of a gap, etc.) (continued)

Westhof, 1990; Michel *et al.*, 1990), group II introns (Michel *et al.*, 1989), RNAse P RNA (Brown *et al.*, 1991; Tranguch & Engelke, 1993), small nuclear RNAs (Guthrie & Patterson, 1988) and 7S RNA (signal recognition particle RNA) (Zwieb, 1989).

The main difficulties in applying this work to other families of RNA will be the development of appropriate initial grammars and the computational cost of parsing longer sequences. The latter problem can only be solved by the development of fundamentally different parsing methods, perhaps relying more on branch-and-bound methods (Lathrop & Smith, 1994) or heuristics. It is currently not clear which approach will be best. The former problem might be solved by the development of effective methods for learning the grammar itself from training sequences. The work of Eddy and Durbin is an important step in this direction (Eddy & Durbin, 1994). Their method relies on correlations between columns in a multiple alignment (Gutell *et al.*, 1992; Lapedes,

```
        Disc.  Align.   Grammar
                        MT10CY10          MT100             RandomTRNA618     Anticodon & Organism
PART III (Parasitic worm mitochondrial tRNAs lacking the D-domain)
 DS4640  . . . . . .    <==><::><==><==>  <==><::><==><::>  <--><::><==><++>  TCT ASCARIS SUUM
 DS4680* . . . . . .    <==><::><==><::>  <==><::><==><::>  <--><::><==><++>  TCT CAENORHABDI. ELEG.
 DS4681  . . . . . .    <==><++><--><-->  <==><--><==><::>  <==><--><==><-->  TGA CAENORHABDI. ELEG.

PART III (Mammalian mitochondrial tRNAs (anticodon GCU) lacking the D-domain)
 DS5321* . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><==><+=>  GCT MOUSE
 DS5440  . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><==><+=>  GCT MACACA FUSCATA
 DS5480  . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><==><+=>  GCT MACACA MULATTA
 DS5520  . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><==><+=>  GCT MACACA FASCICULA.
 DS5560  . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><==><+=>  GCT MACACA SYLVANUS
 DS5600  . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><--><+=>  GCT SAIMIRI SCIUREUS
 DS5640  . . . . . .    <==><::><==><++>  <--><::><==><+=>  <==><::><--><++>  GCT TARSIUS SYRICHTA
 DS5720  . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><==><+=>  GCT CHIMPANZEE
 DS5760  . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><==><+=>  GCT GIBBON
 DS5800  . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><==><+=>  GCT GORILLA
 DS5840  . . . . . .    <==><::><::><+=>  <==><::><::><+=>  <==><::><::><+=>  GCT ORANG UTAN
 RS5240  . . . . . .    <==><::><::><+=>  <==><::><==><+=>  <==><::><==><+=>  GCU HAMSTER
 RS5880* . . . . . .    <==><::><==><+=>  <==><::><==><+=>  <==><::><::><+=>  GCU HUMAN

PART III (Mollusc, insect and echinoderm mitochondrial tRNAs with extended anticodon and T-stems)
 DS4800  . % %  . . .   <==><==><--><++>  <--><++><==><++>  <==><==><==><++>   GCT AEDES ALBOPICTUS
 DS4880  . . %  . . .   <==><::><==><++>  <::><::><==><++>  <::><--><==><++>   GCT DROSOPHILA YAKUBA
 DS5001  . . . . . .    <==><--><==><++>  <==><::><==><++>  <--><--><==><++>   GCT ASTERINA PECTINI.
 DS5041* . . %  . . .   <::><++><==><++>  <::><++><==><++>  <::><++><==><++>   GCT PARACENTROTUS LIV.
 DS5081  . . #  . . .   <::><::><==><++>  <::><::><==><++>  <::><::><==><++>   GCT STRONGYLOCEN.PURP.
 RS4800* # # #  . . .   <=+><::><==><==>  <=+><::><==><++>  <=+><::><==><++>   GCU AEDES ALBOPICTUS

PART III (Mammalian mitochondrial serine tRNAs/tDNA sequences with Anticodon UGA/TGA)
$DS5280  # # #  . . .   <==><+=><==><==>  <==><+=><==><==>  <==><+=><==><==>   TGA RAT
$DS5282  # # #  . . .   <==><+=><==><==>  <==><+=><==><==>  <==><+=><==><==>   TGA RAT
$DS5360  # # #  . . .   <?=><+=><==><==>  <?=><+=><==><==>  <?=><+=><==><==>   TGA BOVINE
$DS5880* # # #  . . .   <==><++><==><==>  <==><++><==><==>  <==><++><==><==>   TGA HUMAN

PART III (Sequences for which the secondary structure is especially unusual or is not established)
 DA3680  . . . . . .    <==><++><::><++>  <==><==><=:><++>  <==><--><--><-->   TGC TRYPANOSOMA BRUCEI
 DA3681* . . %  . . .   <==><++><::><+=>  <==><++><--><+=>  <==><++><--><+=>   TGC TRYPANOSOMA BRUCEI
 DS5100  . . .  A A A                                                         GCT GADUS MORHUA
 DF4720  % # #  . . .   <==><==><==><++>  <==><==><==><++>  <==><==><==><-->   GAA ARTEMIA SP.
```

Figure 17: and +, improvement over TRUSTED. An asterisk * indicates a sequence for which the predicted and trusted alignments are shown in Figure 9. $ preceding a sequence identifier indicates sequences that were included in PART III of TRUSTED, but in PART I in TRUSTEDNEW.

1992; Klinger & Brutlag, 1993; Waterman, 1989; Winker *et al.*, 1990; Sankoff, 1985; Waterman, 1988) to discover the essential base-pairing structure in an RNA family. Another approach would be to use a method like that proposed by Waterman (Waterman, 1989) to find helices in a rough initial multiple alignment, use these helices to design a simple initial grammar in a semi-automated fashion using our high-level RNA grammar specification language, then use the grammar to obtain a better multiple alignment, and iterate this process until a suitable result is obtained. We are currently exploring this approach.

Another important direction for further research is the development of stochastic grammars for tRNA and other RNA families that can be used to search databases for these structures at the DNA level. In order to do this, the grammar must be modified to allow for the possibility of introns in the sequence, and the parsing method must be modified so that it can efficiently search for RNAs

that are embedded within larger sequences. Durbin and Eddy have done the latter modifications in their tRNA experiments and report good results in searching the GenBank structural RNA database and 2.2 Mb of *C. elegans* genomic sequence for tRNAs, even without using special intron models. In our earlier work (Sakakibara *et al.*, 1994), we reported some very preliminary results on modifying tRNA grammars to accommodate introns. We are currently planning to do further work in this direction. We see no insurmountable obstacles in developing effective stochastic grammar-based search methods, but predict that the main practical problem will be dealing with the long computation time required by the present methods.

Finally, there is the question of what further generalizations of hidden Markov models, beyond SCFGs, might be useful. The key advantage of our method over the HMM method is that it allows us to explicitly deal with the secondary structure of the RNA sequence. By extending stochastic models of strings to stochastic models of trees, we can model the base-pairing interactions of the molecule, which determine its secondary structure. This progression is similar to the path taken by the late King Sun Fu and colleagues in their development of the field of syntactic pattern recognition (Fu, 1982). Modeling pseudoknots and higher-order structure would require still more general methods. One possibility would be to consider *stochastic graph grammars* (see the introductory survey by Engelfriet and Rozenberg (Engelfriet & Rozenberg, 1991)) in hopes of obtaining a more general model of the interactions present in the molecule beyond the primary structure. If a stochastic graph grammar framework could be developed that included both an efficient method of finding the most probable folding of the molecule given the grammar and an efficient EM method for estimating the grammar's parameters from folded examples, then extensions of our approach to more challenging problems, including RNA tertiary structure determination and protein folding, would be possible. This is perhaps the most interesting direction for future research suggested by the results of this paper.

Acknowledgments

We thank Anders Krogh, Harry Noller and Bryn Weiser for discussions and assistance, and Michael Waterman and David Searls for discussions. We also thank Sergey Steinberg, Daniel Gautheret, Robert Cedergren, and Mathias Sprinzl for providing us with their unpublished alignments of tRNA and tRNA gene sequences (Steinberg *et al.*, 1993a).

This work was supported by NSF grants CDA-9115268 and IRI-9123692 and NIH grant number GM17129. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

Readers may obtain this paper and its data and multiple alignments via anonymous ftp from `ftp.cse.ucsc.edu` in `/pub/rna`.

References

Abarbanel, R. M., Wieneke, P. R., Mansfield, E., Jaffe, D. A., & Brutlag, D. L. (1984). Rapid searches for complex patterns in biological molecules. *Nucleic Acids Research,* **12** (1), 263–280.

Aho, A. V. & Ullman, J. D. (1972). *The Theory of Parsing, Translation and Compiling, Vol. I: Parsing.* Englewood Cliffs, N.J.: Prentice Hall.

Arnold, F. H. (1993). Engineering proteins for nonnatural environments. *Faseb Journal,* **7**, 744–749.

Baer, R., Bankier, A. T., Biggin, M. D., Deininger, P. L., Farrell, P. J., Gibson, T. J., Hatfull, G., Hudson, G. S., Satchwell, S. C., & Seguin, C. (1984). DNA sequence and expression of the B95-8 Epstein-Barr virus genome. *Nature,* **310**, 207–211.

Baker, J. K. (1979). Trainable grammars for speech recognition. *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America,* pp. 547–550.

Baldi, P., Chauvin, Y., Hunkapiller, T., & McClure, M. A. (1993). Hidden Markov models in molecular biology: new algorithms and applications. In: *Advances in Neural Information Processing Systems 5*, (Hanson, Cowan, & Giles, eds) pp. 747–754, San Mateo, CA: Morgan Kauffmann Publishers.

Bartel, D. P. & Szostak, J. W. (1993). Isolation of new ribozymes from a large pool of random sequences. *Science,* **261**, 1411–1418.

Beaudry, A. A. & Joyce, G. F. (1992). Directed evolution of an RNA enzyme. *Science,* **257**, 635–641.

Berkhout, B. (1992). Structural features in TAR RNA of human and simian immunodeficiency viruses: a phylogenetic analysis. *Nucleic Acids Research,* **20**, 27–31.

Branlant, C., Krol, A., Machatt, M. A., Pouyet, J., Ebel, J. P., Edwards, K., & Kossel, H. (1981). Primary and secondary structures of *Escherichia coli* MRE 600 23S ribosomal RNA. Comparison with models of secondary structure for maize chloroplast 23S rRNA and for large portions of mouse and human 16S mitochondrial rRNAs. *Nucleic Acids Research,* **9**, 4303–4324.

Brenner, S. & Lerner, R. A. (1992). Encoded combinatorial chemistry. *Proceedings of the National Academy of Sciences of the United States of America,* **89**, 5381–5383.

Brown, J. W., Haas, E. S., James, B. D., Hunt, D. A., Liu, J. S., & Pace, N. R. (1991). Phylogenetic analysis and evolution of RNase P RNA in proteobacteria. *Journal of Bacteriology,* **173**, 3855–3863.

Brown, M. P., Hughey, R., Krogh, A., Mian, I. S., Sjölander, K., & Haussler, D. (1993a). Using Dirichlet mixture priors to derive hidden Markov models for protein families. In: *Proc. of First Int. Conf. on Intelligent Systems for Molecular Biology*, (Hunter, L., Searls, D., & Shavlik, J., eds) pp. 47–55, Menlo Park, CA: AAAI/MIT Press.

Brown, M. P., Hughey, R., Krogh, A., Mian, I. S., Sjölander, K., & Haussler, D. (1993b). Dirichlet mixture priors for HMMs. In preparation.

Burke, J. M. & Berzal-Herranz, A. (1993). *In vitro* selection and evolution of RNA: applications for catalytic RNA, molecular recognition, and drug discovery. *Faseb Journal,* **7**, 106–112.

Cantatore, P., Roberti, M., Rainaldi, G., Gadaleta, M. N., & Saccone, C. (1989). The complete nucleotide sequence, gene organization, and genetic code of the mitochondrial genome of *Paracentrotus lividus*. *Journal of Biological Chemistry,* **264**, 10965–10975.

Cardon, L. R. & Stormo, G. D. (1992). Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *Journal of Molecular Biology,* **223**, 159–170.

Chan, L., Zuker, M., & Jacobson, A. B. (1991). A computer method for finding common base paired helices in aligned sequences: application to the analysis of random sequences. *Nucleic Acids Research,* **19**, 353–358.

Chiu, D. K. & Kolodziejczak, T. (1991). Inferring consensus structure from nucleic acid sequences. *Computer Applications in the Biosciences,* **7**, 347–352.

Churchill, G. A. (1989). Stochastic models for heterogeneous DNA sequences. *Bull Math Biol,* **51**, 79–94.

Cohen, F. E., Abarbanel, R. M., Kuntz, I. D., & Fletterick, R. J. (1986). Turn prediction in proteins using a pattern-matching approach. *Biochemistry,* **25**, 266–276.

Courteau, J. (1991). Genome databases. *Science,* **254**, 201–207.

Crozier, R. H. & C., C. Y. (1993). The mitochondrial genome of the honeybee *Apis mellifera*: complete sequence and genome organization. *Genetics,* **133**, 97–117.

Cummings, D. J., McNally, K. L., Domenico, J. M., & Matsuura, E. T. (1990). The complete DNA sequence of the mitochondrial genome of *Podospora anserina*. *Current Genetics,* **17**, 375–402.

Dahlberg, J. E. & Abelson, J. N., eds (1989). *RNA processing. Part A. General Methods* , volume 180 of *Methods in Enzymology*. New York: Academic Press.

Daniels, D. L., Plunkett 3d, G., Burland, V., & Blattner, F. R. (1992). Analysis of the *Escherichia coli* genome. DNA sequence of the region from 84.5 to 86.5 minutes. *Science,* **257**, 771–778.

Daniels, D. L., Sanger, F., & Coulson, A. R. (1983). Features of bacteriophage lambda: analysis of the complete nucleotide sequence. *Cold Spring Harbor Symposia on Quantitative Biology,* **47**, 1009–1024.

Doolittle, R. F., ed (1990). *Molecular Evolution*, volume 183 of *Methods in Enzymology*. New York: Academic Press.

Dunn, J. J. & Studier, F. W. (1981). Nucleotide sequence from the genetic left end of bacteriophage T7 DNA to the beginning of gene 4. *Journal of Molecular Biology,* **148**, 303–330.

Dunn, J. J. & Studier, F. W. (1983). Complete nucleotide sequence of bacteriophage T7 DNA and the locations of T7 genetic elements. *Journal of Molecular Biology,* **166**, 477–535.

Eddy, S. R. & Durbin, R. (1994). RNA sequence analysis using covariance models. Submitted to *Nucleic Acids Research*.

Ellington, A. D. & Szostak, J. W. (1992). Selection in vitro of single-stranded DNA molecules that fold into specific ligand-binding structures. *Nature,* **355**, 850–852.

Engelfriet, J. & Rozenberg, G. (1991). Graph grammars based on node rewriting: An introduction to NLC graph grammars. In: *Lecture Notes in Computer Science*, (Ehrig, E., Kreowski, H., & Rozenberg, G., eds) volume 532 pp. 12–23. Springer-Verlag.

Fichant, G. A. & Burks, C. (1991). Identifying potential tRNA genes in genomic DNA sequences. *Journal of Molecular Biology,* **220**, 659–671.

Fox, G. E. & Woese, C. R. (1975). 5S RNA secondary structure. *Nature,* **256**, 505–507.

Fu, K. S. (1982). *Syntactic pattern recognition and applications.* Englewood Cliffs, NJ: Prentice-Hall.

Gadaleta, G., Pepe, G., De Candia, G., Quagliariello, C., Sbisa, E., & Saccone, C. (1989). The complete nucleotide sequence of the *Rattus norvegicus* mitochondrial genome: cryptic signals revealed by comparative analysis between vertebrates. *Journal of Molecular Evolution,* **28**, 497–516.

Gautheret, D., Major, F., & Cedergren, R. (1990). Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for tRNA. *Computer Applications in the Biosciences,* **6** (4), 325–331.

Glotz, C., Zwieb, C., & Brimacombe, R. (1981). Secondary structure of the large subunit ribosomal RNA from *Escherichia coli*, *Zea mays* chloroplast, and human and mouse mitochondrial ribosomes. *Nucleic Acids Research,* **9**, 3287–3306.

Gouy, M. (1987). Secondary structure prediction of RNA. In: *Nucleic acid and protein sequence analysis, a practical approach*, (Bishop, M. J. & Rawlings, C. R., eds) pp. 259–284. IRL Press Oxford, England.

Gutell, R. R., Power, A., Hertz, G. Z., Putz, E. J., & Stormo, G. D. (1992). Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucleic Acids Research,* **20**, 5785–5795.

Guthrie, C. & Patterson, B. (1988). Spliceosomal snRNAs. *Annual Review of Genetics,* **22**, 387–419.

Hallick, R. B., Hong, L., Drager, R. G., Favreau, M. R., Monfort, A., Orsat, B., Spielmann, A., & Stutz, E. (1993). Complete sequence of *Euglena gracilis* chloroplast DNA. *Nucleic Acids Research,* **21**, 3537–3544.

Han, K. & Kim, H.-J. (1993). Prediction of common folding structures of homologous RNAs. *Nucleic Acids Research,* **21**, 1251–1257.

Haussler, D., Krogh, A., Mian, I. S., & Sjölander, K. (1993). Protein modeling using hidden Markov models: Analysis of globins. In: *Proceedings of the Hawaii International Conference on System Sciences* volume 1 pp. 792–802, Los Alamitos, CA: IEEE Computer Society Press.

Hiratsuka, J., Shimada, H., Whittier, R., Ishibashi, T., Sakamoto, M., Mori, M., Kondo, C., Honji, Y., Sun, C.-R., Meng, B.-Y., Li, Y.-Q., Kanno, A., Nishizawa, Y., Hirai, A., Shinozaki, K.,

& Sugiura, M. (1989). The complete sequence of the rice (*Oryza sativa*) chloroplast genome: intermolecular recombination between distinct tRNA genes accounts for a major plastid DNA inversion during the evolution of the cereals. *Molecular and General Genetics,* **217**, 185–194.

Holley, R. W., Apgar, J., Everett, G. A., Madison, J. T., , Marquisee, M., Merrill, S. H., Penswick, J. R., & Zamir, A. (1965). Structure of a ribonucleic acid. *Science,* **147**, 1462–1465.

Hunter, L., Searls, D., & Shavlik, J., eds (1993). *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology.* Menlo Park, CA: AAAI/MIT Press.

Jaeger, J. A., Turner, D. H., & Zuker, M. (1990). Predicting optimal and suboptimal secondary structure for RNA. *Methods in Enzymology,* **183**, 281–306.

James, B. D., Olsen, G. J., & Pace, N. R. (1989). Phylogenetic comparative analysis of RNA secondary structure. *Methods in Enzymology,* **180**, 227–239.

Joint NIH/DOE Mouse Working Group (1993). A plan for the mouse genome project. *Mammalian Genome,* **4** (6), 293–300.

Klinger, T. & Brutlag, D. (1993). Detection of correlations in tRNA sequences with structural implications. In: *First International Conference on Intelligent Systems for Molecular Biology*, (Hunter, L., Searls, D., & Shavlik, J., eds) , Menlo Park: AAAI Press.

Krogh, A., Brown, M., Mian, I. S., Sjölander, K., & Haussler, D. (1994). Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology,* **235**, 1501–1531.

Lander, E. S. & Green, P. (1987). Construction of multilocus genetic linkage maps in humans. *Proceedings of the National Academy of Sciences of the United States of America,* **84**, 2363–2367.

Lapedes, A. (1992). Private communication.

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

Larsen, N., Olsen, G. J., Maidak, B. L., McCaughey, M. J., Overbeek, R., Macke, T. J., Marsh, T. L., & Woese, C. R. (1993). The ribosomal database project. *Nucleic Acids Research,* **21**, 3021–3023.

Lathrop, R. H. & Smith, T. F. (1994). A branch-and-bound algorithm for optimal protein threading with pairwise (contact potential) amino acid interactions. In: *Proceedings of the 27th Hawaii International Conference on System Sciences* , Los Alamitos, CA: IEEE Computer Society Press.

Lathrop, R. H., Webster, T. A., & Smith, T. F. (1987). Ariadne: pattern-directed inference and hierarchical abstraction in protein structure recognition. *Communications of the ACM,* **30** (11), 909–921.

Le, S. Y. & Zuker, M. (1991). Predicting common foldings of homologous RNAs. *Journal of Biomolecular Structure and Dynamics,* **8**, 1027–1044.

Lehman, N. & Joyce, G. F. (1993). Evolution *in vitro* of an RNA enzyme with altered metal dependence. *Nature,* **361**, 182–185.

Levitt, M. (1969). Detailed molecular model for transfer ribonucleic acid. *Nature,* **224**, 759–763.

Macke, T., Mian, I. S., Cohen, P., & Noller, H. F. (1993). stgrep: a language for performing symbolic computation on RNA sequences and secondary structure. In preparation.

Madison, J. T., Everett, G. A., & Kung, H. K. (1966). On the nucleotide sequence of yeast tyrosine transfer RNA. *Cold Spring Harbor Symposium on Quantitative Biology,* **31**, 409–416.

Marvel, C. C. (1986). A program for the identification of tRNA-like structure in DNA data. *Nucleic Acids Research,* pp. 431–435.

Merriam, J., Ashburner, M., Hartl, D. L., & Kafatos, F. C. (1991). Toward cloning and mapping the genome of *Drosophila. Science,* **254**, 221–225.

Michel, F., Ellington, A. D., Couture, S., & Szostak, J. W. (1990). Phylogenetic and genetic evidence for base-triples in the catalytic domain of group I introns. *Nature,* **347**, 578–580.

Michel, F., Umesono, K., & Ozeki, H. (1989). Comparative and functional anatomy of group II catalytic introns–a review. *Gene,* **82**, 5–30.

Michel, F. & Westhof, E. (1990). Modelling of the three-dimensional architecture of group I catalytic introns based on comparative sequence analysis. *Journal of Molecular Biology,* **216**, 585–610.

Minobe, Y. (1993). Analysis of rice genome. *Tanpakushitsu Kakusan Koso. Protein, Nucleic Acid, Enzyme,* **38**, 704–712.

Noller, H. F., Kop, J., Wheaton, V., Brosius, J., Gutell, R. R., Kopylov, A. M., Dohme, F., Herr, W., Stahl, D. A., Gupta, R., & Woese, C. R. (1991). Secondary structure model for 23S ribosomal RNA. *Nucleic Acids Research,* **9**, 6167–6189.

Noller, H. F. & Woese, C. R. (1981). Secondary structure of 16S ribosomal RNA. *Science,* **212**, 403–411.

Oda, K., Yamato, K., Ohta, E., Nakamura, Y., Takemura, M., Nozato, N., Kohchi, T., Ogura, Y., Kanegae, T., Akashi, K., & Ohyama, K. (1992). Gene organization deduced from the complete sequence of liverwort *Marchantia polymorpha* mitochondrial DNA. *Journal of Molecular Biology,* **223**, 1–7.

Ogasawara, N. (1993). Sequencing project of *Bacillus subtilis* genome. *Tanpakushitsu Kakusan Koso. Protein, Nucleic Acid, Enzyme,* **38**, 669–676.

Okada, K. & Shimura, Y. (1993). Genome studies of *Arabidopsis thaliana. Tanpakushitsu Kakusan Koso. Protein, Nucleic Acid, Enzyme,* **38**, 713–719.

Oliver, S. G., van der Aart, Q. J., Agostoni-Carbone, M. L., Aigle, M., Alberghina, L., Alexandraki, D., Antoine, G., Anwar, R., Ballesta, J. P., & Benit, P. (1992). The complete DNA sequence of yeast chromosome III. *Nature,* **357**, 38–46.

Olson, M. V. (1993). The human genome project. *Proceedings of the National Academy of Sciences of the U.S.A.* **90**, 4338–4344.

Pleij, C. W. A. (1990). Pseudoknots: a new motif in the RNA game. *Trends in Biochemical Sciences,* **15**, 143–147.

Plunkett 3d, G., Burland, V., Daniels, D. L., & Blattner, F. R. (1993). Analysis of the *Escherichia coli* genome. III. DNA sequence of the region from 87.2 to 89.2 minutes. *Nucleic Acids Research,* **21**, 3391–3398.

Presnell, S. R. & Cohen, F. E. (1993). Artificial neural networks for pattern recognition in biochemical sequences. *Annual Review of Biophysics and Biomolecular Structure,* **22**, 283–298.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE,* **77** (2), 257–286.

RajBhandary, U. L., Stuart, A., Faulkner, R. D., Chang, S. H., & Khorana, H. (1966). Nucleotide sequence studies of yeast phenylalanine sRNA. *Cold Spring Harbor Symposium on Quantitative Biology,* **31**, 425–434.

Romero, D. P. & Blackburn, E. H. (1991). A conserved secondary structure for telomerase RNA. *Cell,* **67**, 343–353.

Saenger, W. (1984). *Principles of nucleic acid structure.* Springer Advanced Texts in Chemistry. New York: Springer-Verlag.

Sakakibara, Y. (1992). Efficient learning of context-free grammars from positive structural examples. *Information and Computation,* **97**, 23–60.

Sakakibara, Y., Brown, M., Mian, I. S., Underwood, R., & Haussler, D. (1994). Stochastic context-free grammars for modeling RNA. In: *Proceedings of the Hawaii International Conference on System Sciences* , Los Alamitos, CA: IEEE Computer Society Press.

Sakakibara, Y., Brown, M., Underwood, R., Mian, I. S., & Haussler, D. (1993). Stochastic context-free grammars for modeling RNA. Technical Report UCSC-CRL-93-16 UC Santa Cruz Computer and Information Sciences Dept., Santa Cruz, CA 95064.

Sanger, F., Air, G. M., Barrell, B. G., Brown, N. L., Coulson, A. R., Fiddes, C. A., Hutchison, C. A., Slocombe, P. M., & Smith, M. (1977). Nucleotide sequence of bacteriophage ϕX174 DNA. *Nature,* **265**, 687–695.

Sanger, F., Coulson, A. R., Friedmann, T., Air, G. M., Barrell, B. G., Brown, N. L., Fiddes, J. C., Hutchison 3d, C. A., Slocombe, P. M., & Smith, M. (1978). The nucleotide sequence of bacteriophage ϕX174. *Journal of Molecular Biology,* **125**, 225–246.

Sanger, F., Coulson, A. R., Hong, G. F., Hill, D. F., & Petersen, G. B. (1982). Nucleotide sequence of bacteriophage lambda DNA. *Journal of Molecular Biology,* **162**, 729–773.

Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.* **45**, 810–825.

Santner, T. J. & Duffy, D. E. (1989). *The Statistical Analysis of Discrete Data.* New York: Springer Verlag.

Saurin, W. & Marlière, P. (1987). Matching relational patterns in nucleic acid sequences. *Computer Applications in the Biosciences,* **3** (2), 115–120.

Schmitt, M. E., Bennett, J. L., Dairaghi, D. J., & Clayton, D. A. (1993). Secondary structure of RNase MRP RNA as predicted by phylogenetic comparison. *Faseb Journal,* **7**, 208–213.

Schneider, D., Tuerk, C., & Gold, L. (1992). Selection of high affinity RNA ligands to the bacteriophage R17 coat protein. *Journal of Molecular Biology,* **228**, 862–869.

Searls, D. B. (1992). The linguistics of DNA. *American Scientist,* **80**, 579–591.

Searls, D. B. (1993a). The computational linguistics of biological sequences. In: *Artificial Intelligence and Molecular Biology* chapter 2, pp. 47–120. AAAI Press.

Searls, D. B. (1993b). String variable grammar: a logic grammar formalism for DNA sequences. Unpublished.

Searls, D. B. & Dong, S. (1993). A syntactic pattern recognition system for DNA sequences. In: *Proc. 2nd Int. Conf. on Bioinformatics, Supercomputing and complex genome analysis* , : World Scientific. In press.

Shapiro, B. A. & Zhang, K. (1990). Comparing multiple RNA secondary structures using tree comparisons. *CABIOS,* **6** (4), 309–318.

Sibbald, P. R. & Argos, P. (1990). Scrutineer: a computer program that flexibly seeks and describes motifs and profiles in protein sequence databases. *Computer Applications in the Biosciences,* **6** (3), 279–288.

Staden, R. (1980). A computer program to search for tRNA genes. *Nucleic Acids Research,* pp. 817–825.

Staden, R. (1990). Searching for patterns in protein and nucleic acid sequences. In: (Doolittle, 1990) pp. 193–211.

Steinberg, S., Gautheret, D., Cedergren, R., & Sprinzl, M. (1993a). tRNA database and access system (tDAS). In: *Transfer RNA*, (Soll, D. & RajBhandary, U. L., eds). American Society of Microbiology Washington DC.

Steinberg, S., Misch, A., & Sprinzl, M. (1993b). Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Research,* **21** (13), 3011–3015.

Stiegler, R., Carbon, P., Zuker, M., Ebel, J. P., & Ehresmann, C. (1980). Secondary and topographic structure of ribosomal RNA 16S of *escherichia coli. Academy Sciences (Paris) Series D,* **291**, 937–940.

Sulston, J., Du, Z., Thomas, K., Wilson, R., Hillier, L., Staden, R., Halloran, N., P., G., Thierry-Mieg, J., Qiu, L., Dear, S., Coulson, A., Craxton, M., Durbin, R., Berks, M., Metzstein, M., Hawkins, T., Ainscough, R., & Waterston, R. (1992). The *C. elegans* genome sequencing project: a beginning. *Nature,* **356**, 37–41.

Sutcliffe, J. G. (1979). Complete nucleotide sequence of the *Escherichia coli* plasmid pBR322. *Cold Spring Harbor Symposia on Quantitative Biology,* **43**, 77–90.

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

Thatcher, J. W. & Wright, J. B. (1968). Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory,* **2**, 57–81.

Tinoco Jr., I., Uhlenbeck, O. C., & Levine, M. D. (1971). Estimation of secondary structure in ribonucleic acids. *Nature,* **230**, 363–367.

Tranguch, A. J. & Engelke, D. R. (1993). Comparative structural analysis of nuclear RNase P RNAs from yeast. *Journal of Biological Chemistry,* **268**, 14045–1455.

Tuerk, C. & Gold, L. (1990). Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science,* **249**, 505–510.

Turner, D. H., Sugimoto, N., & Freier, S. M. (1988). RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry,* **17**, 167–192.

Tzeng, C. S., Hui, C. F., Shen, S. C., & Huang, P. C. (1992). The complete nucleotide sequence of the *Crossostoma lacustre* mitochondrial genome: conservation and variations among vertebrates. *Nucleic Acids Research,* **20**, 4853–4858.

von Montagu, M., Dean, C., Flavell, R., Goodman, H., Koornneef, M., Meyerowitz, E., Peacock, J., Shimura, Y., & Somerville, C. (1992). The multinational coordinated *Arabidopsis thaliana* genome research project. Progress Report: year two. Technical Report NSF 90-80 National Science Foundation Washington DC.

Waterman, M. S. (1988). Computer analysis of nucleic acid sequences. *Methods in Enzymology,* **164**, 765–792.

Waterman, M. S. (1989). Consensus methods for folding single-stranded nucleic acids. In: *Mathematical Methods for DNA Sequences*, (Waterman, M. S., ed) chapter 8. CRC Press.

Waterman, M. S., Arratia, R., & Galas, D. J. (1984). Pattern recognition in several sequences: consensus and alignment. *Bulletin of Mathematical Biology,* **46**, 515–527.

Weiser, B., Gutell, R., & Noller, H. F. (1993). XRNA: An X Windows environment RNA editing/display package. Unpublished manuscript.

Winker, S., Overbeek, R., Woese, C., Olsen, G., & Pfluger, N. (1990). Structure detection through automated covariance search. *Computer Applications in the Biosciences,* **6**, 365–371.

Woese, C. R., Gutell, R. R., Gupta, R., & Noller, H. F. (1983). Detailed analysis of the higher-order structure of 16S-like ribosomal ribonucleic acids. *Microbiology Reviews,* **47** (4), 621–669.

Woese, C. R., Magrum, L. J., Gupta, R., Siegel, R. B., Stahl, D. A., Kop, J., Crawford, N., Brosius, J., Gutell, R., Hogan, J. J., & Noller, H. F. (1980). Secondary structure model for bacterial 16S ribosomal RNA: phylogenetic, enzymatic and chemical evidence. *Nucleic Acids Research,* **8**, 2275–2293.

Wyatt, J. R., Puglisi, J. D., & Tinoco Jr., I. (1989). RNA folding: pseudoknots, loops and bulges. *BioEssays,* **11** (4), 100–106.

Zachau, H. G., Dütting, D., Feldmann, H., Melchers, F., & Karau, W. (1966). Serine specific transfer ribonucleic acids. XIV. Comparison of nucleotide sequences and secondary structure models. *Cold Spring Harbor Symposium on Quantitative Biology,* **31**, 417–424.

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

Zuker, M. & Sankoff, D. (1984). RNA secondary structures and their prediction. *Bull. Math. Biol.* **46**, 591–621.

Zwieb, C. (1989). Structure and function of signal recognition particle RNA. *Progress in Nucleic Acid Research and Molecular Biology,* **37**, 207–234.

Zwieb, C., Glotz, C., & Brimacombe, R. (1981). Secondary structure comparisons between small subunit ribosomal RNA molecules from six different species. *Nucleic Acids Research,* **9**, 3621–3640.