Small Subunit Ribosomal RNA Modeling Using Stochastic Context-Free Grammars

Michael P.S. Brown *
HNC
5935 Cornerstone Court West
San Diego, CA 92121–3278
mpsb@hnc.com

Abstract

We introduce a model based on stochastic context-free grammars (SCFGs) that can construct small subunit ribosomal RNA (SSU rRNA) multiple alignments. The method takes into account both primary sequence and secondary structure basepairing interactions. We show that this method produces multiple alignments of quality close to hand edited ones and outperforms several other methods. We also introduce a method of SCFG constraints that dramatically reduces the required computer resources needed to effectively use SCFGs on large problems such as SSU rRNA. Without such constraints, the required computer resources are infeasible for most computers. This work has applications to fields such as phylogenetic tree construction.

Keywords: Ribosomal RNA, Multiple Alignment, Stochastic Context-Free Grammar, HMM, Constraints

Introduction

The ribosome is one of the most complicated and interesting machines in the natural world. The ribosome translates the information contained in messenger RNA (mRNA) transcribed from DNA into a string of amino acids that constitute biologically active proteins. Without the ribosome, life as we know it would not exist. The complexity of this machine is enormous and elucidation of its structure and functioning has been the center of much research. In addition to this complexity is the problem of characterizing its evolutionary history. Because of its fundamental importance to life on Earth, this molecule has been studied extensively to find evolutionary relationships between different organisms and to help construct the phylogenetic tree of life. Constructing the tree of life has mainly concentrated on characterizing the RNA component of the ribosome. This work presents a new method of characterizing small subunit ribosomal RNA (SSU rRNA) by automatically producing multiple alignments of it that take into account both primary and secondary structure using stochastic context-free grammars (SCFGs). Stochastic context-free grammars are a fully probabilistic sequence model that are an extension of hidden Markov models. We compare the SCFG method to several other methods including a thermodynamic RNA folding program (MFOLD), a primary sequence aligner (ClustalW), and a hidden Markov model based primary sequence aligner (SAM). We show that the SCFG method outperforms the other methods and produces multiple alignments of quality close to hand-edited ones.

The Ribosome and Small Subunit Ribosomal RNA

The ribosome consists of several RNA and protein components (Watson *et al.* 1987). The small subunit ribosomal RNA has length of approximately 1500 bases and forms a complicated secondary structure that has largely been determined using comparative sequence analysis (Gutell 1984; 1993; Gutell *et al.* 1992; Woese & Gutell 1989; Woese *et al.* 1983). The secondary structure consists of approximately 50 helical stalks. About two thirds of the bases in SSU rRNA are involved in basepair interactions. See Figure 1 for the secondary structure of SSU rRNA for E.coli (Maidak *et al.* 1997). The high resolution three dimensional crystal structure of the ribosome is nearing completion (Clemons Jr. *et al.* 1999; Ban *et al.* 1999; Cate *et al.* 1999).

The ribosome is fundamentally important to life because all living things rely on it to produce protein from the mRNA transcripts of the genome. The ubiquity of the ribosome in living things has been exploited to construct phylogenetic trees containing widely different life forms from bacteria to humans by using information from the RNA components of the ribosome. There are several reasons why ribosomal RNA is informative for this purpose (Woese 1987; Olsen & Woese 1993). Ribosomal RNA serves as a good molecular chronometer. Its function is highly conserved. It occurs in nearly all organisms. Different positions in the sequence change at different rates allowing

*This work was supported under a PMMB Burroughs Wellcome fellowship at the University of California Santa Cruz.

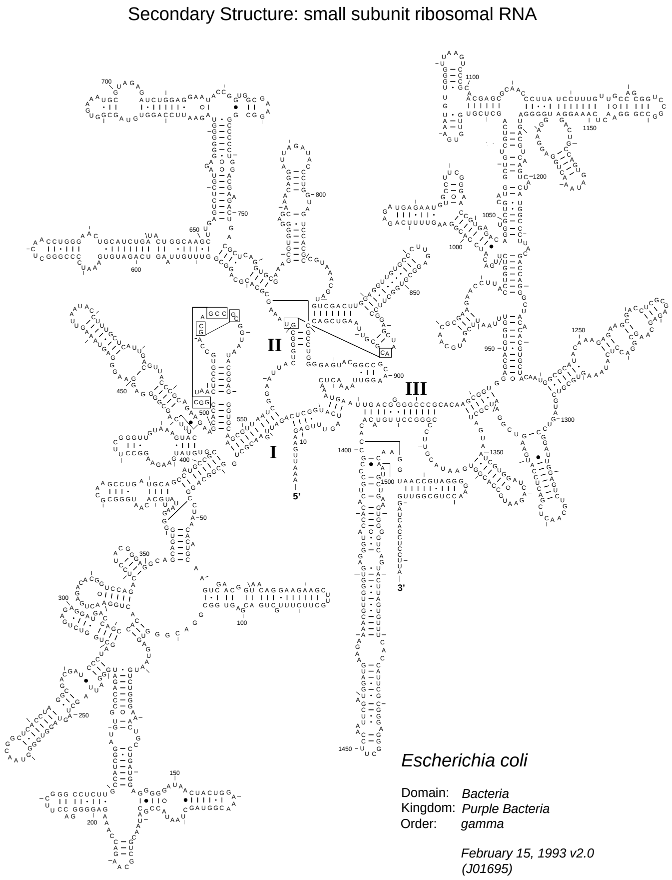


Figure 1: The secondary structure of SSU rRNA for E.coli. The primary sequence starts at the 5' end near the center of the image and ends at the 3' end to the right and down from the 5' end. Lines represent basepair interactions. Image from the Ribosomal Database Project (originally created by Dr. Robin Gutell) (Maidak *et al.* 1997).

a wide dynamic range of time periods to be measured. Its size is large and consists of several domains that contain a high degree of information. It is easily and directly sequenced using reverse transcriptase reactions. Although there are several debates questioning the reliability of rRNA to represent phylogenetic relationships among organisms (Olsen & Woese 1993) especially in the case of bacteria in which lateral gene transfer is common (Doolittle 1999), the analysis of rRNA has had a tremendous impact on the field of phylogenetic tree construction.

The use of ribosomal RNA phylogenetic analyses has lead to fundamental insights in the the nature of life on Earth including the incorporation of a new domain into the tree of life, *Archaea* (Woese & Fox 1977; Woese, Kandler, & Wheelis 1990). The ability to classify organisms based on genotypic rather than phenotypic features has opened the doors to research on bacterial phylogenies (Woese 1987; Gray, Burger, & Lang 1999). Ribosomal RNAs are routinely used to construct phylogenies of different organisms and help explain functional characteristics (Zardoya *et al.* 1995). These phylogenies can be important in medicine, since several antibiotics interfere directly with ribosomal RNA (Fourmy *et al.* 1996).

Fundamental to all rRNA phylogenetic analysis is the construction of a multiple alignment of the sequences (Olsen & Woese 1993; Woese 1987; Feng & Doolittle 1987; Morrison & Ellis 1997). Most phylogenetic analysis involves examining these multiple alignments to produce a tree relating the organisms. The estimation of the phylogenetic tree and the multiple alignment can be tackled simultaneously (Durbin *et al.* 1998; Knudsen & Hein 1999) but this problem is difficult and is not explored in our modeling. This work concentrates on constructing multiple alignments of ribosomal RNA given a simple set of sequences. Readers interested in the phylogenetic analysis of multiple alignments might refer to other works (Durbin *et al.* 1998; Gulko 1995; Felsenstein 1981).

Multiple Alignment of Ribosomal RNA Using Stochastic Context-Free Grammars

Previous attempts at aligning ribosomal RNA have relied on both automated methods and human hand fine-tuning. Indeed, recent papers have lamented that a fully automated, accurate alignment of rRNA sequences remains a difficult problem (O'Brien, Notredame, & Higgins 1998). Some of the difficulties stem from the fact that ribosomal RNAs are large, a fact that also makes them good sources of information for phylogenetic inference. Another more fundamental difficulty arises because ribosomal RNAs form complex secondary structures. A correct alignment of rRNA must not only take into account primary sequence conservation but also information present in the secondary structure. Because most automated sequence alignment programs are unable to account for secondary structure, this poses a fundamental problem that has largely been solved by hand-tuning the alignment to make sure it agrees with the secondary structure. We propose stochastic context-free grammars (SCFGs) as a solution to both problems (Sakakibara *et al.* 1994; Eddy & Durbin 1994). SCFGs solve the problem of aligning most secondary structure by forming a probabilistic model that takes into account both primary and secondary structure information. We also introduce a new probabilistically well founded technique that constrains the computation of SCFGs, allowing complex models with thousands of states to analyze biological sequences such as SSU rRNA with thousands of bases in a reasonable amount of time. Without these algorithmic techniques, simple implementations of the SCFG techniques would lead to impractical required computer resources.

Stochastic context-free grammars are an extension of hidden Markov models and have their basis in formal language theory and probability. SCFGs can be used to model secondary structure interactions of biological molecules. Like HMMs, SCFGs are probabilistic generative models. SCFGs use a grammar to generate a string by applying a series of string rewrite rules or productions. The sequence of productions used can be interpreted as representing different biological structures such as basepairs. See Figure 2 for an illustration of an SCFG grammar applied to RNA structure. SCFGs can be thought of as models that capture the information in a multiple alignment in which certain columns are considered to be basepaired and therefore have an informative joint distribution. SCFGs can be used to align new sequences of rRNA including prediction of all basepairs, bulges, and loop regions of a new sequence of SSU rRNA, discriminate between rRNA and non-rRNA, and create multiple alignments of rRNA (Sakakibara *et al.* 1994; Eddy & Durbin 1994; Brown 1999). The use of SCFGs usually begins with a structural alignment of an RNA family. This alignment is accompanied by a list of basepair columns or columns that should be modeled with a joint distribution because they are interacting. From this information a SCFG can be automatically estimated and used. Other methods such as the `FOLDALIGN` program also align on primary and secondary structure but do not scale well to problems the size of SSU rRNA (Gorodkin, Heyer, & Stormo 1997).

Technically, the SCFG is unable to correctly represent pseudoknots because the abstracted language of pseudoknots is not well nested and therefore not context-free. However, most of the basepairing structure of SSU rRNA is well nested and can be modeled with a SCFG. For the purposes of this work, it is thought that the computational cost outweighs the gain of modeling pseudoknot interactions in SSU rRNA and therefore pseudoknot modeling is not done (Brown & Wilson 1995; Rivas & Eddy 1999; Chen, Le, & Maizel 1992).

SCFG Constraints

Although computation using SCFGs is efficient in the sense that it is a polynomial-time algorithm, using SCFGs for large models and large database searches becomes infeasible because of the computational demands that are required. SCFG algorithm running times are known to be $O(ML^3)$ and space requirements are $O(ML^2)$ for the grammars used in this work where M is the number of nonterminals and L is the length of the string. These complexities derive from the dynamic program used to compute SCFG probabilities. The dynamic program table stores for every nonterminal, the probability that the nonterminal derives the subword indexed from i to j where $1 \leq i \leq j \leq L$. This gives the $O(ML^2)$ dynamic program table size, not considering the space to hold the grammar itself. The time required to compute this table is $O(ML^3)$ because each table entry can take $O(L)$ time to compute. Productions of

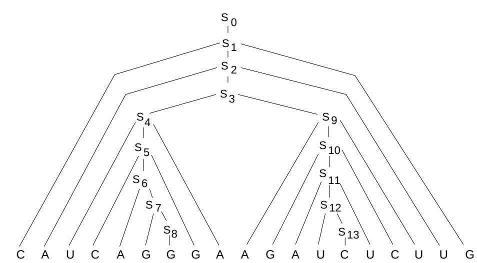
a. Productions

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

b. Derivation

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

c. Parse tree



d. Secondary Structure

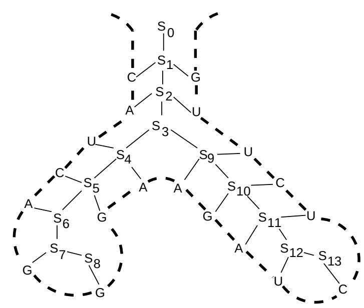


Figure 2: A simple context-free grammar which may be used to derive a set of RNA molecules including the specific example illustrated here, `CAUCAGGGAAGAUCUCUUG`. *a.* A set of productions P which generates RNA sequences with a certain restricted structure. S_0 (start symbol), $S_1, \ldots, S_{13}$ are nonterminals; `A, U, G` and `C` are terminals representing the four nucleotides. *b.* Application of the productions P could generate the given sequence by the derivation indicated. For example, if the production $S_1 \rightarrow CS_2G$ is selected, the string CS_2G replaces S_1 and the derivation step is written $S_1 \Rightarrow CS_2G$. *c.* The derivation in *b* may be arranged in a tree structure called a *parse* or *derivation tree*. *d.* The physical secondary structure of the RNA sequence is a reflection of the parse tree (or syntactic structure).

the form ($S \rightarrow TU$) require all order L splitpoints to be considered. This time complexity assumes that the number of productions associated with any nonterminal is constant.

Modeling even fairly short structures, such as t-RNA with an average length of 76 bases, has high computational costs when performing large database searches. It has been estimated that it would take at least nine years of computer time to find t-RNAs in the human genome using SCFGs without performance optimizations (Lowe & Eddy 1997). Parsing larger structures such as small subunit ribosomal RNA with length of approximately 2000 bases with a fully parameterized stochastic context-free grammar requires a conservative estimate of 16 billion bytes of memory simply to hold the dynamic programming tables. This is not feasible for today's computers.

Previous approaches to solve this problem were to preprocess the data with a simple filter. The program tRNAscan-SE uses this method when it searches for t-RNAs (Lowe & Eddy 1997). It should be noted that in this method the preprocess does not influence the SCFG's work. It simply presents likely strings to the SCFG but does not change the computation done by the SCFG on this string. This method works for small structures like t-RNA in which the preprocess simply eliminates much of the database and allows only likely sections to be analyzed by the SCFG. It fails for larger structures like SSU rRNA, because the computational requirements for analyzing even a single SSU rRNA structure are impractically large.

Parsing a string of length L given a grammar can be made more efficient through the use of constraints. SCFG grammars are usually highly ambiguous in a formal language sense and constraints limit the ambiguity by disallowing certain derivations that conflict with the given constraints. This additional constraint knowledge allows the parsing problem to be carried out on subproblems of length m that are smaller than the original L length problem. Constraints fall naturally in the context-free grammar framework because they provide a hierarchical decomposition of the pattern they are describing.

A SCFG constraint is defined to be 5-tuple: (S, a, b, c, d) where S is a nonterminal and (a, b, c, d) are indices in which $1 \leq a \leq b \leq L$ and $1 \leq c \leq d \leq L$ where L is the length of the string being parsed. The constraint is interpreted to mean that if the nonterminal S is used in the most likely parse of the sequence then S must derive a substring (i, j) in which $a \leq i \leq b$ and $c \leq j \leq d$. The zero-knowledge constraint, $(S, 1, L, 1, L)$, enforces that S can derive any substring beginning and ending anywhere in the string of length L. The four indices allow uncertainty in the range that the nonterminal derives.

Constraints can be implemented in a SCFG framework by modifying the dynamic programming algorithm that computes the probability of a nonterminal deriving the subword i to j to not explore terms that involve nonterminals deriving subwords that are not allowed by their constraints. All derivations that violate the constraints are given probability zero and are not explored. For example, if the dynamic program starts to compute the probability that nonterminal S derives the subword 10 to 20 and the constraint $(N, 2, 8, 22, 30)$ exists then the dynamic program can fill in zero probability and not do any further work because 10 is not between 2 and 8 and 20 is not between 22 and 30. The algorithmic issues associated with this approach including the propagation of constraint information between nonterminals that are hierarchically related through the SCFG are discussed in detail elsewhere (Brown 1999).

Constraints can be generated automatically. The technique used to generate constraints in this work relies on hidden Markov model (HMM) posterior decodings. The motivation for using HMMs comes from their ability to model primary sequence structure in a way identical to SCFGs. While HMMs are unable to model pairwise interactions, they are able model primary structure. Most molecules, even RNA molecules with a large number of secondary structure interactions, have some primary conservation of structure. Small subunit ribosomal RNA, for example, has regions of highly conserved primary structure across practically all divisions of life that are easily aligned using primary sequence only (O'Brien, Notredame, & Higgins 1998). These highly conserved regions are, with little doubt, linked to structurally important blocks of SSU rRNA tertiary structure. HMMs are able to model these primary regions very well. Therefore, one can argue that if an HMM model of SSU rRNA maps a base of a sequence to an internal HMM model state with very high probability then that mapping can be believed with some confidence. This confidence can be quantified through HMM posterior decoding. Our constraint generation procedure uses an HMM approximation to the SCFG. A new string is posterior decoded using the HMM to identify high confidence base-to-HMM-state mappings. These mappings are then translated to constraints on SCFG nonterminals using a reference string.

The mapping from base x to HMM state H is made using posterior decoding of the HMM. The posterior decoding for a string x and HMM θ gives the probability that the state deriving the ith base, π_i, is k,

$$P(\pi_i = k | x, \theta)$$

This value is computed efficiently an HMM. The maximum posterior decoding is

$$\hat{\pi}_i = \text{argmax}_k P(\pi_i = k | x, \theta)$$

This maps each base, i, to its most likely posterior state, k. We then rely on the fact that a high posterior mapping implies high confidence that the mapping is correct.

The translation from HMM state to SCFG nonterminal is accomplished using a reference string. A Viterbi path is computed for the reference string under the

SCFG. The reference string is a sequence with a well-known parse, such as *E.coli* SSU rRNA. A maximum posterior decoding is computed for the reference string under an HMM approximation to the SCFG. If base x of the reference string is derived by nonterminal T in the SCFG Viterbi parse and by state H in the HMM with a high enough posterior probability then a mapping is made from HMM state H to SCFG nonterminal T. This technique maps HMM states used in the maximum posterior decoding to SCFG nonterminals used in the reference Viterbi parse.

In biological modeling, constraints allow additional knowledge to be incorporated into the parse. If, for example, a basepair is known to form between bases 9 and 25 at a certain position in a helix then a constraint can be formulated for the nonterminal that outputs that basepair: $(helixNT, 9, 9, 25, 25)$. There is no uncertainty for this nonterminal, $helixNT$, given this constraint because the constraint specifies the begin and end ranges as having length exactly equal to one.

We tested the constraint generation technique using hidden Markov model posterior decoding by parsing 169 SSU rRNA sequences with an estimated SCFG and HMM using the methods described in this section. We compared the constrained dynamic programming table size to dynamic programming table size required without the constraints. The average size of the constrained tables is $0.000575(\pm 0.000199)$ the size of the unconstrained tables. Reductions of this size allow previously impractical problems to be computed efficiently.

Experimental Methods Overview

We gauge the ability of several algorithms to produce alignments of small subunit ribosomal RNA (SSU rRNA). We examine stochastic context-free grammars(RNACAD), hidden Markov models(SAM), a primary sequence aligner(ClustalW), and an energy minimization technique(MFOLD). Each method is started with a small training set of phylogenetically diverse sequences and allowed to learn. Some methods require no training and therefore do not benefit from this learning phase. We than take an independent set of test sequences and align them using each method producing test alignments. We score each test alignment using a performance metric described below and report the results. Note that the energy minimization technique, MFOLD, does not produce an alignment as the other methods do. However, our performance metric allows comparison of MFOLD to the other methods.

Methods

Small Subunit Ribosomal RNA Data

There exist several sites maintaining alignments of small subunit ribosomal RNA (SSU rRNA) from thousands of species (Maidak *et al.* 1997; de Peer *et al.* 1997). The data for this experiment was obtained from the Ribosomal Database Project (RDP) (Maidak *et al.* 1997). The RDP has multiple alignments, phylogenetic trees, and secondary structures for ribosomal RNA. The multiple alignments and phylogenetic trees are available in a variety of human and machine readable forms. The secondary structures however are available only as postscript diagrams that show the secondary structure in graphical form and are not machine readable. The multiple alignments have no explicit identification of basepairing.

Tt.maritim	Tmc.roseum	D.radiodur	Bac.fragil	
Sap.grandi	Chl.limico	Pir.staley	Clm.psitta	
Nost.muscr	Syn.6301	Olst.lut_C	Zea_mays_C	
Spi.haloph	Lpn.illini	Ric.prowaz	Nis.gonor1	
Ps.aerugi3	E.coli	Dsv.desulf	Myx.xanthu	Hlb.pylor6
Cam.jejun5	Fus.nuclea	Stm.ambofa	Arb.globif	
Bif.bifidu	Eub.barker	B.subtilis	L.casei	Eco.faecal
Acp.laidla	C.ramosum	M.capricol	C.pasteuri	

Table 1: The 34 bacterial sequences in the training set. The listing left to right and top to bottom is phylogenetically ordered. These sequences are identified in the RDP Bacterial representative set as spanning the phylogenetic tree.

Our training set consists of 34 bacterial SSU rRNA sequences. See Table 1 for a list of the training sequences. The training set is a subset of the RDP set of Prokaryotic representative sequences. The RDP representative set consists of 80 sequences that form a representative sample across the tree of life. We removed from the representative set all sequences containing more than 1% dot characters. This is necessary because the dot character in the RDP multiple alignments indicates zero or more unknown bases and is hard to model because its definition is too loose. For the remaining sequences we changed all dots to the indel character, '-', if the alignment column contains only indels and dots and to the wildcard character, 'N', otherwise. This left 34 sequences with no dots as our training set.

In addition to the training sequences, we include the secondary structure for the training sequence *E. coli* in our training data. The secondary structure is given as a list of pairs of bases that are thought to be involved in basepairing interactions. This list was obtained by examining the RDP secondary structure postscript diagram for *E.coli*. We remove any pseudoknotted interactions from the secondary structure specification because they are not representable by the SCFG. We infer basepairing interactions for sequences other than *E.coli* using the multiple alignment. If *E.coli* has a pair of basepairing bases located in two multiple alignment positions in the RDP alignment then all other sequences that have bases in those two columns are inferred to be basepairing as well. The stochastic context-free grammar algorithms are able to exploit the information contained in the pairwise interactions specified by the secondary structure. Therefore if there is additional information in the pairwise interactions that is not present in the primary sequence, the stochastic context-free grammar algorithms will be able to use this information and provide superior performance.

Our test set contains a total of 169 SSU rRNA sequences: 119 Bacterial and 50 Archaea sequences. This set is derived from the RDP Prokaryotic Multiple Alignment datafile and consists of all sequences in the database that do not contain the dot character. As mentioned previously, the dot character has a loose definition indicating zero or more unknown bases and is too noisy to be modeled accurately.

Tested Modeling Algorithms

Stochastic Context-Free Grammar, RNACAD The RNACAD system was used to analyze the SSU rRNA data. The system was presented with the training set and the list of basepairs. From this data, system automatically generated an estimated grammar that had approximately 9,000 states and 50,000 parameters. This includes 554 modeled loop positions and 478 modeled basepair positions. This estimated grammar can be used in alignment and database tasks. In order to use this grammar on the test set, an HMM approximation to the SCFG was needed in order to generate constraints. The SAM HMM modeling system was used(Hughey & Krogh 1996). The SAM system was given the multiple alignment and generated an estimated HMM using the `modelfromalign` program. The SCFG was then used to parse the test sequences with constraints supplied by the HMM. Problems of this size would not run without the HMM supplied constraints because their size is too large. The parses of the test set sequences were scanned to identify bases that were produced by helix productions and therefore were predicted to be basepairing.

Hidden Markov Model, SAM The SAM system models primary sequence structure using hidden Markov models(HMMs)(Hughey & Krogh 1996). An HMM probabilistically encodes the information of a multiple alignment in its state structure but does not model any interactions between multiple alignment columns. HMMs model loop positions exactly as SCFGs do. However, HMMs are unable to model pairwise interactions and therefore basepaired positions are modeled as two independent loop positions.

An HMM was trained on the RDP multiple alignment using `modelfromalign`. The length of the HMM superstates was 1515. This HMM was then used to align the test set using `align2model`. Only default parameters were used. A list of basepairs was derived from this multiple alignment using the secondary structure of *E.coli* and inferring basepairing interactions for other sequences in the multiple alignment. SAM v2.2.1 was used.

Profile Alignment, ClustalW The ClustalW method models primary structure and uses profile-based progressive multiple alignments to construct a full multiple alignment from a set of sequences (Thompson, Higgins, & Gibson 1994). This is done by first constructing a distance matrix on all pairs of sequences using a pairwise dynamic programming alignment, clustering the distances to construct a guide tree, and using this tree to progressively align sequences starting with the most similar sequences and continuing with sequence-sequence, sequence-profile, and profile-profile alignments until all sequences are aligned. This is one of the most widely used sequence alignment programs.

ClustalW(v 1.7) was used to align the test set using the Slow/Accurate multiple alignment option. A list of basepairs was derived from this multiple alignment using the secondary structure of *E.coli* and inferring basepairing interactions for other sequences in the multiple alignment.

Energy Minimization, MFOLD MFOLD relies on thermodynamic free energy parameters to optimally fold an RNA sequence taking into consideration basepairing interactions and stacking(Zuker, Mathews, & Turner 1999; Mathews *et al.* 1999). The parameters are thermodynamically derived and are not specific to the particular RNA that is being folded. This is one of the most widely used programs for predicting RNA secondary structure.

The test sequences were submitted to the MFOLD server and a list of basepairs was derived from resulting foldings. The web-based version of MFOLD 2.3 was used.

The Performance Metric

In order to gauge the ability of the algorithms to produce good alignments of small subunit ribosomal RNA (SSU rRNA) we use a stringent performance metric based on basepairing interactions. Performance on the test set for each method is reported by three numbers: the number of correctly identified basepairs (true positives), the number of missed basepairs (false negatives), and the number of falsely reported basepairs (false positives). All numbers are reported with respect to the basepairing interactions as reported or implied by our data source, the Ribosomal Database II Project (RDP).

This metric is an indication of good alignment because more than half of the bases in SSU rRNA are involved in basepairs. If an alignment program were to shift any of the bases involved in either the 3' or 5' side of a basepair, an error would result. Therefore a method can get no errors only if it identifies every base in every basepair exactly. The metric could be made more stringent by forcing all loop positions to align also but because some loop regions are thought to be "inserts", no alignment in these regions is expected. Thus penalizing for an incorrect alignment in these regions was thought to be too harsh.

In order to compute the performance metric, a list of correct basepairs must be generated from the RDP alignment. This is done by taking the secondary structure basepair list for *E.coli*. For each *E.coli* basepair, find the corresponding columns in the multiple alignment that contain that E.coli basepair. If the sequence, S, has bases i and j in those multiple alignment columns then a basepair (i, j) is reported for sequence S. Note that we assume that the RDP alignment is correct. If the RDP alignment contains any errors then an erroneous basepair might be reported and any method that corrects the error will be penalized by the performance metric because it disagrees with the RDP.

The stochastic context-free grammar system, RNACAD, and the energy minimization algorithm, MFOLD, both make basepair predictions directly. The hidden Markov model system, SAM, and the primary sequence aligner, ClustalW, produce multiple alignments that are processed as described above to produce basepair lists.

Results and Discussion

The test set contains 169 sequences and 76,167 basepairs as identified by the RDP multiple alignment and the secondary structure of *E.coli*. Methods are tested to see how many of the 76,167 basepairs are correctly identified.

Primary Alignment, ClustalW

False Positives	False Negatives	True Positives
5623	7283 (9.6%)	68884 (90.4%)

ClustalW is a primary sequence aligner and is able to exploit the well-conserved regions of SSU rRNA to produce a multiple alignment. However ClustalW does slightly worse than the HMM method which also is a primary sequence aligner. It seems as though the statistical information captured by the HMM is important in the alignment of SSU rRNA. ClustalW's method of using pairwise alignment, using these alignments to construct a guide tree, and then using this tree to construct a full alignment might have done well, especially because sequences are weighted to account for biased representation, but its performance is worse than an HMM.

Hidden Markov Models, SAM

False Positives	False Negatives	True Positives
5161	5632 (7.4%)	70535 (92.6%)

The HMM is unable to represent basepairing interactions but still does well. The HMM is able to model all loop positions and well conserved basepairing positions are modeled as two independent positions very well. This primary probabilistic modeling captures much of Bacterial SSU rRNA information as evidenced by its good performance.

Energy Minimization, MFOLD

False Positives	False Negatives	True Positives
7700	11284 (65.7%)	5883 (34.3%)

MFOLD does not do very well. In nature, SSU rRNA folds interacting with many other protein and RNA components. Perhaps simple thermodynamic parameters cannot capture all the necessary information for correct folding of complexes of RNA and other biological molecules.

Previous results show that the accuracy of MFOLD to be somewhat better than the results reported here with about a 50% true positive rate (Konings & Gutell 1995). This might be explained by the smaller test set of about 41 non-Eukaryote sequences used in the other work. This smaller set might contain more easily predicted sequences than the 169 sequences examined here.

Stochastic Context-Free Grammars, RNACAD

False Positives	False Negatives	True Positives
1926	2073 (2.7%)	74094 (97.3%)

The SCFG does very well, the best in these tests. The next section examines some of the errors in more detail.

Analysis of RNACAD Errors When we examine the false negative and false positive errors made by RNACAD with respect to the RDP alignment we find that the errors group together. See Figure 3 for a graphical representation of these errors. Some errors occur in structure that is different for Archaea (Woese 1987). Because our training set was entirely Bacteria and our test set contained Archaea, errors have occurred because the RDP alignment for Archaea was different from Bacteria. Because the RNACAD system is trying to model all sequences under *one* universal secondary structure, it tries to place the Archaea structure under the Bacteria structure in several regions and consequently these predictions are counted as errors when compared to the "correct" RDP alignment. Table 2 shows how errors are distributed among different nonterminals. In essence different nonterminals correspond to different columns (or pairs of columns for basepair positions) in a multiple alignment. Table 2 and Figure 3 show that most regions in the multiple alignment do not have any errors but a few regions disagree with the RDP alignment.

Examining the errors as grouped by sequence we find that most sequences have few errors. See Table 4 for a grouping of errors by sequence. See Table 3 for a graphical representation of how the errors are distributed phylogenetically given the phylogenetic grouping of sequences maintained by the RDP.

Interestingly the sequence Prth.zop_C with the worst prediction with 101 errors is close in the phylogenetic listing to Ochs.nea_C with only 1 error. Prth.zop_C is in SKELETONEMA SUBGROUP while Ochs.nea_C is in OCHROSPHAERA SUBGROUP both under BACTERIA, CYANOBACTERIA AND CHLOROPLASTS, CHLOROPLASTS AND CYANELLES, OTHER CHLOROPLASTS. The length of Prth.zop_C is 1515 and the length of Ochs.nea_C is 1483. While the two sequences have similar lengths, the sequence Ochs.nea_C is much more similar to E.coli than Prth.zop_C is. The sequence Prth.zop_C might

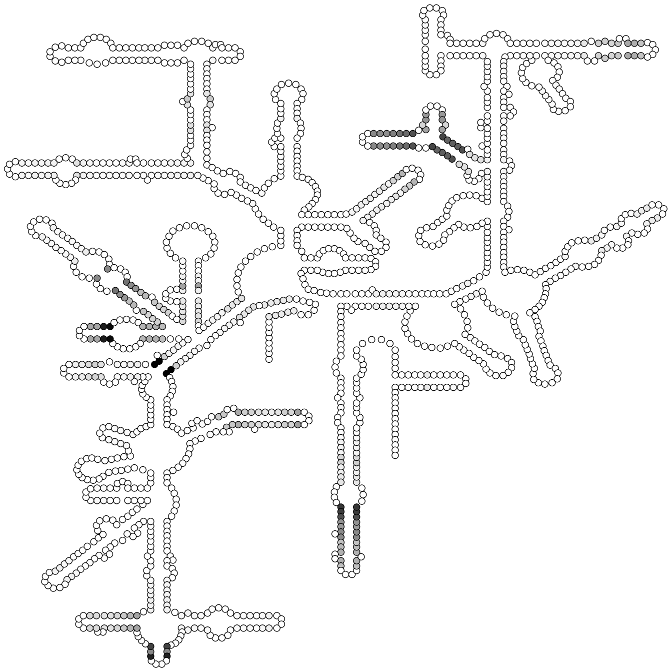


Figure 3: Number of basepair errors on the SSU rRNA test set made by the RNACAD system for all 169 test sequences. The graphic represents the E.coli secondary structure for SSU rRNA, one circle for each base. The darkness of each circle represents the number of errors that were made for that basepair nonterminal. The number of errors is the number of false positives plus the number of false negatives. Black represents the maximum number of errors (100) and white represents no errors. It is clear the errors do not occur randomly but are grouped together.

Errors range >	and <	Number of Nonterminals
-1	1	186
0	11	194
10	21	37
20	31	17
30	41	14
40	51	12
50	61	3
60	71	6
70	81	4
80	91	2
90	101	3
-1	101	478

Table 2: RNACAD basepair errors grouped by nonterminal. The table gives for a range of number of errors, the number of nonterminals having a number of errors in that range. Number of errors is the number of false positives plus the number of false negatives for that nonterminal. Most nonterminals have no or a small number of errors. The total number of helix producing nonterminals is 478.

= ARCHAEA	51.88
== EURYARCHAEOTA	48.02
=== METHANOCOCCALES	37.00
=== METHANOBACTERIALES	54.40
=== METHANOMICROBACTERIA_AND_RELATIVES	46.26
=== THERMOCOCCALES	68.00
=== METHANOPYRALES	42.00
== CRENARCHAEOTA	69.00
=== CRENARCHAEOTA_GROUP_I	70.88
=== CRENARCHAEOTA_GROUP_II	54.00
= BACTERIA	11.97
== THERMOTOGALES	16.00
=== GEOTOGA_GROUP	28.00
=== THERMOTOGA_GROUP	4.00
== GREEN_NON-SULFUR_BACTERIA_AND_RELATIVES	7.50
=== CHLOROFLEXUS_SUBDIVISION	15.00
=== DEINOCOCCUS-THERMUS_SUBDIVISION	0.00
== FLEXIBACTER-CYTOPHAGA-BACTEROIDES_PHYLUM	5.00
=== SUBDIVISION_I	5.00
== GREEN_SULFUR_BACTERIA	14.00
== PLANCTOMYCES_AND_RELATIVES	3.67
=== PLANCTOMYCES_SUBDIVISION	4.00
=== CHLAMYDIA_SUBDIVISION	3.50
== CYANOBACTERIA_AND_CHLOROPLASTS	20.03
=== CHLOROPLASTS_AND_CYANELLES	20.03
== SPIROCHETES_AND_RELATIVES	11.00
=== LEPTOSPIRA_SUBDIVISION	11.00
== PURPLE_BACTERIA	9.85
=== ALPHA_SUBDIVISION	23.67
=== BETA_SUBDIVISION	3.67
=== GAMMA_SUBDIVISION	3.44
=== DELTA_SUBDIVISION	13.50
=== EPSILON_SUBDIVISION	6.33
== FUSOBACTERIA_AND_RELATIVES	14.00
== GRAM_POSITIVE_PHYLUM	5.45
=== HIGH_G+C_SUBDIVISION	5.67
=== C.LITUSEBURENSE_GROUP	19.00
=== BACILLUS-LACTOBACILLUS-STREPTOCOCCUS_SUBDIVISION	1.87
=== MYCOPLASMA_AND_RELATIVES	9.50
=== CLOSTRIDIUM_AND_RELATIVES	5.25

Table 3: Number of RNACAD errors grouped on the RDP linearly arranged phylogenetic tree. The characters "=" represent level of the tree. The number after the group name is the average number of basepair errors for all sequences contained in that phylogenetic group. The length of the bars on the right represent the average number of errors for that group. The smallest non-zero average error is 1.87 and the largest is 70.88.

have more basepair prediction errors because it is more distantly related to the set of sequences used to estimate the SCFG which includes E.coli. Even though the RDP places both Prth.zop_C and Ochs.nea_C under the subgroup OTHER CHLOROPLASTS, the sequences have mutated to be very different from each other. The subgroup OTHER CHLOROPLASTS, it seems, contains distantly related organisms.

Conclusions and Future Work

Probabilistic models of small subunit ribosomal RNA using stochastic context-free grammars produce superior multiple alignments in relation to several other methods in terms of basepair prediction and alignment. The quality of the SCFG alignment is close to the quality of the hand-edited alignment. The method runs in a reasonable amount time using a system of constraints. This is significant because simple implementations of the method would lead to impossible required computer resources.

A web server that uses this method is located at http://www.cse.ucsc.edu/research/compbio/ssurrna.html. This web server allows you to enter a sequence and predict its secondary structure as a bacterial small subunit ribosomal RNA. You can then generate

Errors range >	and <	Number of Sequences
-1	11	82
10	21	13
20	31	19
30	41	8
40	51	21
50	61	10
60	71	10
70	81	2
80	91	4
90	101	0
100	102	1
-1	102	169

Table 4: RNACAD basepair errors grouped by sequence. The table gives for a range of number of errors, the number of sequences having a number of errors in that range. Number of errors is the number of false positives plus the number of false negatives for that nonterminal. The total number of sequences is 169.

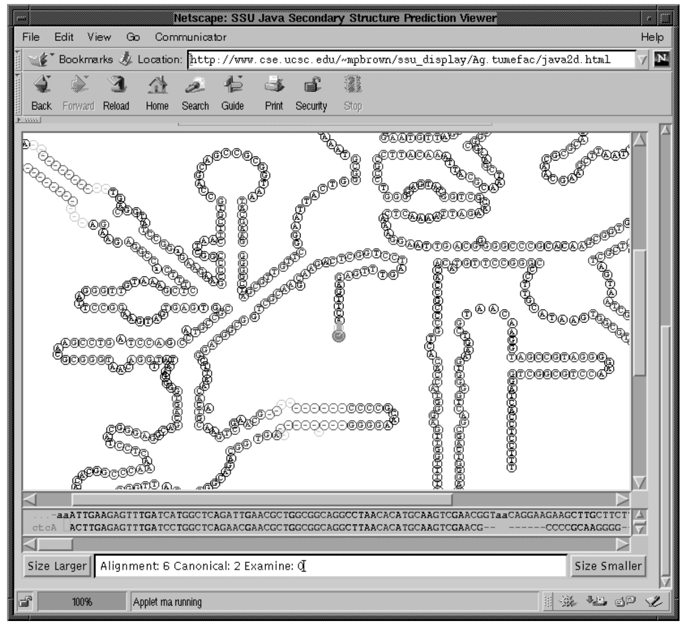


Figure 4: The web-based SSU rRNA RNACAD system located at http://www.cse.ucsc.edu/research/compbio/ssurrna.html. Shown is the java secondary structure viewer. The sequence Ag.tumefac is being examined. Ag.tumefac's secondary structure is shown in relation to E.coli's secondary structure. The alignment of Ag.tumefac and E.coli is also shown.

multiple alignments, report a list of basepairs, and interact with the secondary structure using a Java graphical user interface. See Figure 4 for a snapshot of this tool. Check the website for information and any future software updates.

Because of the method's ability to produce high quality multiple alignments of ribosomal RNA, it has direct application in areas such as phylogenetic tree reconstruction (Olsen & Woese 1993). This could be important in areas such as organism diversity in natural environments (Pace 1997) or identification of disease or infection (Riley *et al.* 1998).

Acknowldgements

Most of this work was done at the University of California at Santa Cruz as a PhD student in David Haussler's group and was supported with a PMMB Burroughs Wellcome fellowship.

References

Ban, N.; Nissen, P.; Hansen, J.; Capel, M.; Moore, P.; and Steitz, T. 1999. Placement of protein and rna structures into a 5 angstrom resolution map of the 50s ribosomal subunit. *Nature* 400:841–847.

Brown, M. P. S., and Wilson, C. 1995. Rna pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. In Hunter, L., and Klein, T., eds., *Pacific Symposium on Biocomputing*, 109–125.

Brown, M. P. 1999. *RNA Modeling Using Stochastic Context-Free Grammars*. Ph.D. Dissertation, University of California, Santa Cruz.

Cate, J.; Yusupov, M. M.; Yusupova, G. Z.; Earnest, T. N.; and Noller, H. F. 1999. X-ray crystal structures of 70s ribosome functional complexes. *Science* 285(5436):2095–2104.

Chen, J.-H.; Le, S.-Y.; and Maizel, J. V. 1992. A procedure for rna pseudoknot prediction. *CABIOS* 8:243–248.

Clemons Jr., W.; May, J.; Wimberyly, B.; McCutcheon, J.; Capel, M.; and Ramakrishnan, V. 1999. Structure of a bacterial 30s ribosomal subunit at 5.5 Angstom resolution. *Nature* 400:833–840.

de Peer, Y. V.; Jansen, J.; Rijk, P. D.; and Wachter, R. D. 1997. The antwerp ribosomal rna database. *NAR* 25:111–116.

Doolittle, W. 1999. Phylogenetic classification and the universal tree. *Science* 284:2124–2128.

Durbin, R.; Eddy, S.; Krogh, A.; and Mitchison, G. 1998. *Biological sequence analysis : probabalistic models of proteins and nucleic acids*. Cambridge University Press.

Eddy, S. R., and Durbin, R. 1994. RNA sequence analysis using covariance models. *NAR* 22:2079–2088.

Felsenstein, J. 1981. Evolutionary trees from dna sequences. *Journal of Molecular Evolution* 17:368–376.

Feng, D. F., and Doolittle, R. F. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution* 25:351–360.

Fourmy, D.; Recht, M. I.; Blanchard, S. C.; and Puglisi, J. D. 1996. Structure of the a site of escherichia coli 16s ribosomal rna complexed with an aminoglycoside antibiotic. *Science* 274(5291):1367–1371.

Gorodkin, J.; Heyer, J.; and Stormo, G. 1997. Finding common sequence and structure motifs in a set of rna sequences. In *Proceedings of ISMB-97*.

Gray, M.; Burger, G.; and Lang, B. 1999. Mitochondrial evolution. *Science* 283:1476–1481.

Gulko, B. 1995. Using phylogenetic markov trees to detect conserved structure in rna multiple alignments. Master's thesis, UC Santa Cruz.

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

Gutell, R. 1984. *Comparative structural analysis of 16s ribosomal rna*. Ph.D. Dissertation, UC Santa Cruz.

Gutell. 1993. Comparitive analysis of rna. In Gestel, and Atkins., eds., *The RNA World*. Addison-Wesley.

Hughey, R., and Krogh, A. 1996. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS* 12(2):95–107. Information on obtaining SAM is available at `http://www.cse.ucsc.edu/research/compbio/sam.html`.

Knudsen, B., and Hein, J. 1999. Rna secondary structure prediction using stochastic context-free grammars and evolutionary history. *bioinformatics* 15(6):446–454.

Konings, D. A. M., and Gutell, R. R. 1995. A comparison of thermodynamic foldings with comparatively derived structures of 16s and 16s-like rrnas. *RNA* 1:559–574.

Lowe, T., and Eddy, S. 1997. trnascan-se: a program for improved detection of transfer rna genes in genomic sequence. *NAR* 25:955–964.

Maidak, B.; Olsen, G.; Larsen, N.; Overbeek, R.; McCaughey, M.; and Woese, C. 1997. The ribosomal data project (rdp). *NAR* 25:443–453.

Mathews, D.; Sabina, J.; Zuker, M.; and Turner, D. 1999. Expanded sequence dependence of thermodynamic parameters provides robust prediction of rna secondary structure. *JMB* 288:911–940.

Morrison, D., and Ellis, J. 1997. Effects of nucleotide sequence alignment of phylogeny estimation. *Mol. Biol. Evol.* 14(4):428–441.

O'Brien, E.; Notredame, C.; and Higgins, D. 1998. Optimization of ribosomal rna profile alignments. *Bioinformatics* 14(4):332–341.

Olsen, G., and Woese, C. 1993. Ribosmal rna: a key to phylogeny. *FASEB* 7:113–123.

Pace, N. 1997. A molecular view of microbial diversity and the biosphere. *Science* 276:734–740.

Riley, D.; Berger, R.; Miner, D.; and Krieger, J. 1998. Diverse and related 16s rrna-encoding dna sequences in prostate tissues of men with chronic prostatitis. *JOURNAL OF CLINICAL MICROBIOLOGY* 36(6):1646–1652.

Rivas, E., and Eddy, S. R. 1999. A dynamic programming algorithm for rna structure prediction including pseudoknots. *JMB* 285(5):2053–2068.

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

Thompson, J. D.; Higgins, D. G.; and Gibson, T. J. 1994. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties, and weight matrix choice. *NAR* 22(22):4673–4680.

Watson, J.; Hopkins, N.; Roberts, J.; Steitz, J.; and Weiner, A. 1987. *Molecular Biology of the Gene*. Benjamin/Cummings.

Woese, C., and Fox, G. 1977. Phylogenetic structure of the prokaryotic domain. *PNAS* 74:5088–5090.

Woese, C., and Gutell, R. 1989. Evidence for several higher order structural elements in ribosomal rna. *PNAS* 86:3119–3122.

Woese, C. R.; Gutell, R. R.; Gupta, R.; and 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.; Kandler, O.; and Wheelis, M. 1990. Towards a natural system of organisms. *PNAS* 87:4576–4579.

Woese, C. 1987. Bacterial evolution. *Microbiological Reviews* 51(2):221–271.

Zardoya, R.; Costas, E.; Lopex-Rodas, V.; Garrido-Pertierra, A.; and Bautista, J. 1995. Revised dinoflagellate phylogeny inferred from molecular analysis of large subunit ribosomal rna gene sequences. *Journal of Molecular Evolution* 41:637–645.

Zuker, M.; Mathews, D.; and Turner, D. 1999. Algorithms and thermodynamics for rna secondary structure prediction. In Barciszewski, J., and Clark, B., eds., *RNA Biochemistry and Biotechnology*. Kluwer.