Computational Analysis of RNAs

S.R. Eddy

Howard Hughes Medical Institute and Department of Genetics, Washington University School of Medicine, Saint Louis, Missouri 63108

Genome sequence analysis of RNAs presents special challenges to computational biology, because conserved RNA secondary structure plays a large part in RNA analysis. Algorithms well suited for RNA secondary structure and sequence analysis have been borrowed from computational linguistics. These "stochastic context-free grammar" (SCFG) algorithms have enabled the development of new RNA genefinding and RNA homology search software. The aim of this paper is to provide an accessible introduction to the strengths and weaknesses of SCFG methods and to describe the state of the art in one particular kind of application: SCFG-based RNA similarity searching. The INFERNAL and RSEARCH programs are capable of identifying distant RNA homologs in a database search by looking for both sequence and secondary structure conservation.

A fundamental goal of genomics is to compile a comprehensive parts list for every organism: a catalog of all genes, regulatory elements, and other functional sequences in the genome (ENCODE Project Consortium 2004). But words such as "all" and "comprehensive" are terms of art in genomics. They mean as many as possible, for a reasonable cost and in reasonable time, of the kinds of functional sequences we know how to identify. For some kinds of sequence elements, we are only beginning to be able to take genome-wide approaches. Functional noncoding RNA elements are a striking example.

There are many known functional RNAs, ranging from catalysts in the ribosome and RNase P, to guide RNAs for RNA editing, to structural RNAs in the spliceosome, and more (Eddy 2001; Szymanski et al. 2003; Mattick and Makunin 2005). A series of discoveries in the past decade made it clear how incomplete our knowledge of functional RNA still is, including the discoveries of several large families of RNA genes, such as microRNAs (miRNAs) involved in posttranscriptional regulation of mRNAs (Bartel 2004), C/D small nucleolar RNAs (snoRNAs) directing site-specific 2′-O-methylation of target RNAs, and H/ACA snoRNAs directing site-specific pseudouridylation of target RNAs (Bachellerie et al. 2002; Brown et al. 2003). Even in *Escherichia coli*, many small new RNA genes have been discovered, at least some of which are posttranscriptional regulators (Storz et al. 2005), and numerous *cis*-regulatory elements called riboswitches have been identified in bacteria as well (Tucker and Breaker 2005; Winkler 2005).

For molecular biologists to discover whole new families of RNA elements in well-studied organisms is both embarrassing and exciting. These discoveries serve to remind us that unbiased discovery methods do not exist. Consider the recent explosion of papers on miRNAs (Bartel 2004). Up until 2001, tiny 21–25-nucleotide miRNA genes were not within the parameters of what we expected genes to look like, aside from two oddball *Caenorhabditis elegans lin-4* and *let-7* genes. miRNAs are often biochemically abundant, but they are only noticed if tiny RNAs are not run off the end of the gel. miRNAs are readily cloned and sequenced, but not when RNA samples are enriched for capped poly(A)$^+$ mRNA to eliminate the "uninteresting" background of poly(A)$^-$ rRNAs and tRNAs. miRNA genes show mutant genetic phenotypes, but if the mutation maps to an interval that contains no protein-coding genes, it takes intestinal fortitude to persevere and find the gene (as Victor Ambros's lab did with *lin-4*; Lee et al. 1993) as opposed to giving up. Specialized computational genefinding programs readily predict miRNA genes, but standard genefinders are looking for open reading frames and codon bias, which noncoding RNA genes do not have.

We probably do not know the full extent to which organisms use RNA regulatory motifs and noncoding RNA genes. We need better systematic genome-wide approaches for identifying functional RNA elements. One approach is to map complete transcriptomes, including both mRNA and noncoding RNA populations, by cDNA sequencing and tiled whole-genome microarrays (Okazaki et al. 2002; Carninci et al. 2005; Cheng et al. 2005). In mammalian genomes, these approaches have resulted in claims of thousands of putative noncoding RNA transcripts (Okazaki et al. 2002; Numata et al. 2003; Furuno et al. 2006; Ravasi et al. 2006). However, it remains unclear how many of these cDNA transcripts represent functional noncoding RNAs, as opposed to being artifacts of cryptic low-level promoters, pre-mRNA contamination, missplicing, unannotated alternative splicing, unrecognized small protein-coding genes, RNA degradation intermediates, and other sources of apparently noncoding RNA one should expect to find in a total cellular RNA population (Wang et al. 2004; Hüttenhofer et al. 2005; Babak et al. 2005; Lee et al. 2006). Additional analysis is required to distinguish functional noncoding RNAs from other transcripts. Additionally, although transcriptome mapping can identify novel independent transcripts, it does not help in identifying new *cis*-regulatory RNA elements contained in known mRNA transcripts.

Another approach is to systematically identify evolutionarily conserved elements by comparative genome sequence analysis. More than half of the conserved sequence in mouse/human genome comparisons appears to be in noncoding regions (Waterston et al. 2002). Large numbers of comparative genome sequences are enabling higher-resolution identification of short and/or weakly conserved elements (Eddy 2005; Stone et al. 2005). An advantage of comparative genome sequence analysis is that it can identify both conserved noncoding RNA genes and conserved *cis*-regulatory RNA elements. A disadvantage is that many other kinds of functional elements show sequence conservation, not just functional RNAs. Sequence conservation suggests that a genomic region is functional, but some kind of additional analysis is required to distinguish whether that function is at the level of DNA, RNA, or protein.

A focus of my lab has been on the development of computational analysis methods for identifying functional RNAs. The heart of our work is a general class of statistical models called "stochastic context-free grammars" (SCFGs), which we use to create computational methods that treat RNA as both primary sequence and base-paired secondary structure (Durbin et al. 1998). With different kinds of SCFGs, we have developed strong RNA similarity search methods (Eddy and Durbin 1994; Eddy 2002; Klein and Eddy 2003), reasonable noncoding RNA genefinding methods (Rivas and Eddy 2001; Rivas et al. 2001), and promising prototypes of RNA structure prediction methods (Dowell and Eddy 2004 and in prep.). Despite the off-putting jargon "stochastic context-free grammar," which we inherited from the field of computational linguistics where SCFGs were first developed (Lari and Young 1990), SCFGs are in fact a natural extension from familiar primary sequence analysis methods to RNA secondary structure methods.

RNA SEQUENCE ANALYSIS OUGHT TO MODEL RNA SECONDARY STRUCTURE

Computational tools are an essential part of comprehensive genome annotation. We rely on *similarity search* programs such as BLAST (Altschul et al. 1997) to identify informative protein homologies and to deduce gene structures by mapping cDNA and EST (expressed sequence tags) sequences onto genome sequence. *Genefinding* programs are used to identify novel genes, by looking for general statistical properties of that class of feature, such as the presence of an open reading frame and codon bias in protein-coding genes. *Motif identification* programs try to identify *cis*-regulatory elements, such as transcription-factor-binding sites, by identifying short conserved and/or overrepresented DNA sequences.

Most tools only look at linear primary sequence, scoring one residue (or aligned pair or column of homologous residues) at a time. In RNA analysis, linear sequence models are inadequate. Many (although not all) functional RNAs conserve a base-paired secondary structure. We want RNA computational analysis tools to be able to model both sequence and RNA secondary structure.

Why are we mostly satisfied with primary sequence analysis tools like BLAST for proteins, but not for RNA? Surely, *any* computational sequence analysis method would be more powerful if it took structural constraints into account. Both RNAs and proteins fold into three-dimensional structures composed of stereotyped secondary structure elements, and these structures constrain primary sequence evolution. That is, in general, if artificial sequences are produced with good primary sequence similarity to a known protein or RNA, few will fold properly (Socolich et al. 2005).

In making a practical computational tool, it is not sufficient to know that structure imposes constraints on sequence. These constraints also have to have predictable effects on sequence, and these effects have to be computable with time-efficient algorithms. Additionally, one uses the simplest tool that gets the job done. A simple linear sequence model is preferred over a more biologically realistic model if the simple model does just as well in a fraction of the time.

In the case of proteins, for the task of similarity searching, BLAST analysis has substantial power, routinely identifying significant homologies down to 20–30% amino acid sequence identity. Many proteins are conserved at this level across billions of years of divergence. Moreover, although protein structure clearly constrains primary sequence, we do not really understand *how* (i.e., we cannot yet predict very well which sequences will fold into active structures), nor do we know how to compute efficiently with what we do know (most existing protein folding or "threading" algorithms are very compute-intensive). Higher-order tools for protein analysis do exist (Godzik 2003), but they gain relatively little power at a high computational cost.

In the case of RNA, BLAST analysis is often unsatisfactory. Significant nucleic acid sequence alignments are only detected down to about 60–70% nucleotide sequence identity, largely as a consequence of the smaller nucleotide alphabet. Although some RNAs are highly conserved (notably ribosomal RNAs), many conserved RNAs will diverge below a 60–70% identity threshold in just tens or hundreds of millions of years. Thus, BLAST comparisons of RNAs are often unable to see reliably across important evolutionary divergences, such as across different animal phyla. The contrast between protein and RNA similarity searches is perhaps most striking when one looks at genome annotations of conserved homologs of the components of well-studied ribonucleoprotein (RNP) complexes. Often, the protein components of RNPs are annotated and the RNA components are not. If BLAST is used to search the fly, nematode, or yeast genome for homologs of human RNase P RNA, for example, no significant hits are seen, whereas conserved RNase P protein components are readily detectable. The presence of small nucleolar RNA homologs in Archaea was suspected based on BLAST detection of homologs of snoRNA-associated proteins, but detection of Archaeal snoRNAs required a combination of experiment and more sophisticated computational modeling (Omer et al. 2000).

Moreover, RNA structure is dominated by base pairs, and base pairs induce highly predictable patterns of long-distance pairwise residue complementarity in RNA primary sequence (Gutell et al. 2002). These patterns of structure-induced complementarity are so obvious in aligned RNA sequences that human analysts are often capable of accurately deducing the conserved secondary structure of an RNA sequence family solely from observed pairwise correlations in sequence alignments (*comparative sequence analysis*) (Pace et al. 1989). Robin Gutell and coworkers, for example, correctly predicted 97–98% of the conserved base pairs in ribosomal RNAs, essentially by eye (Gutell et al. 2002).

Thus, we need more powerful methods of RNA analysis than primary sequence analysis, and the constraints that conserved base-pairing imposes on RNA sequences are understood and easily predictable—by humans, at least. But are base-pairing constraints something we can use in efficient computer programs?

SCORING BOTH SEQUENCE AND RNA STRUCTURE

We want to be able to use a combination of sequence and structure information for a variety of RNA analysis problems, but for clarity, it will be useful to focus on a specific problem. Consider the problem of identifying homologs of a known RNA sequence family. Given a multiple sequence alignment of a family of homologous RNAs, and a consensus secondary structure for that family, we want to build a position-specific scoring model and use that model to search a sequence database and identify more homologs. (The model is called the *query*, and each database sequence is considered one at a time as a *target*.)

In a standard linear sequence profile, we assign 4 scores (for A, C, G, U) at each aligned column. If residue *a* occurs at some aligned position with probability p_a, compared to its average overall background frequency f_a, we calculate the score for that residue as:

$$S_a = \log_2 \frac{p_a}{f_a}$$

(The base two on the logarithm is an arbitrary and traditional choice, which makes scores in units of "bits.") For example, a completely conserved adenine ($p_A = 1.0$) gets a score of +2 (assuming uniform background expectation of 25% for each base). A position that allows either purine ($p_A = p_G = 0.5$) scores +1 for either purine. All standard sequence alignment methods (BLAST, Smith–Waterman, profile hidden Markov models) use essentially this same additive *log-odds scoring* method, which is well-grounded in statistics (Durbin et al. 1998).

Now, imagine that we have a perfectly conserved Watson–Crick base pair, but no primary sequence conservation in either column individually. That is, one column can be any of A, C, G, U with uniform probability, but when the residue there is an A, the residue in the other column is a U, and so on, with the two aligned columns maintaining a complementary Watson–Crick base pair. A primary sequence method assigns a score of 0 to any base in both columns, regardless of whether the two bases can base-pair or not. Substantial information is lost.

To capture that information, we must be able to score pairs of residues simultaneously. Log-odds scores are readily applied to pairs of positions. We obtain the score s_{ab} for residue pair *a,b* from the joint probability p_{ab} we expect to see that pair occur and the expected frequency that we would see *a,b* occur by chance independently, the product $f_a f_b$:

$$s_{ab} = \log_2 \frac{p_{ab}}{f_a f_b}$$

A perfectly conserved Watson–Crick pair could thus get a score of +2, when the individual positions are freely varying. That is, one base pair potentially conveys as much information as one completely conserved residue in a sequence profile.

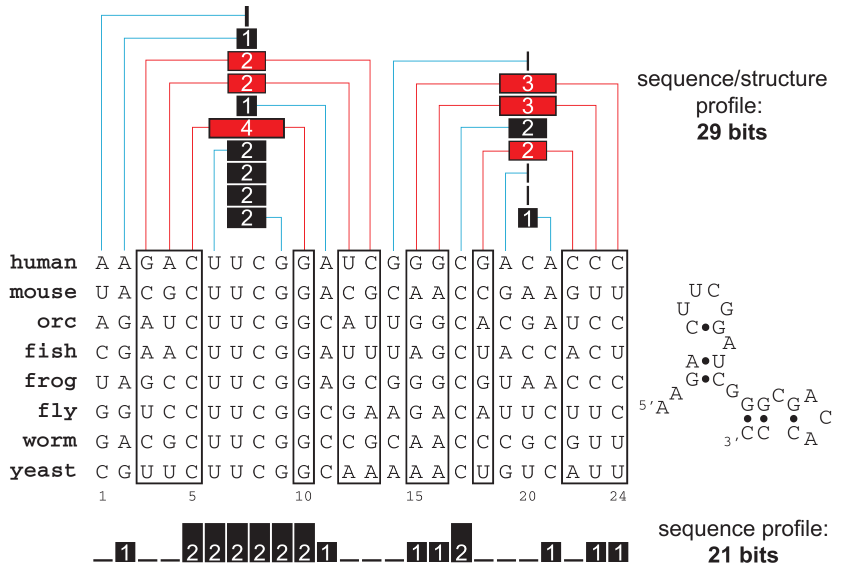
Importantly, the pairwise score s_{ab} contains information about *both* primary sequence and base-pairing conservation. If both positions were completely conserved residues (say an A/U), s_{AU} would be +4, the same as if we scored the two columns independently by sequence as +2, +2. It is also important that the score s_{ab} is a general pairwise residue score, and it does not restrict the two residues to canonical Watson–Crick pairs. The pairwise residue score s_{ab} can deal with any pairwise correlation, including GU and noncanonical RNA pairs.

The amount of extra information in pairwise residue correlations induced by base-pairing is significant (Fig. 1). Generally speaking, in a typical structural RNA sequence family, about 50–60% of residues are involved in base pairs. Empirically, a scoring model that captures base-pairing typically has about 50% more information content as a sequence-only model (albeit with wide variation, depending on the RNA).

The point here is that formally grounded, statistical scoring of conserved RNA base pairs is straightforward and well understood. There is no reason for any proposed RNA alignment method to use arbitrary scores. The real difficulty is not in scoring residues, but in how to align a model to a target sequence when insertions and deletions are allowed. If we allow insertions and deletions, we do not know which target residues to score as which consensus base pairs and consensus singlet positions. We need the scoring system, and we also need an optimization algorithm that can look at all possible predicted structures and alignments of the target sequence, and find the best-scoring one(s). There are an astronomical number of possible solutions for typical alignment problems, unfortunately. Brute force enumeration of all possible solutions is not feasible. Either we need to simplify the problem, or we need a clever efficient algorithm.

If we restrict alignments to a limited number of possible gaps—by assuming that individual helices behave as ungapped blocks, for instance—then it becomes possible to exhaustively enumerate all possible alignments, as in Gautheret and Lambert's RNA profile search program ERPIN (Gautheret and Lambert 2001). But restricting where gaps are allowed is worrisome. It falls short of the general alignment methods we are accustomed to in primary sequence analysis.

This is where stochastic context-free grammars come in. SCFGs give us efficient and general alignment algorithms



```
human  A A G A C U U C G G A U C G G G C G A C A C C C
mouse  U A C G C U U C G G A C G C A A C C G A A G U U
  orc  A G A U C U U C G G C A U U G G C A C G A U C C
 fish  C G A A C U U C G G A U U U A G C U A C C A C U
 frog  U A G C C U U C G G A G C G G G C G U A A C C C
  fly  G G U C C U U C G G C G A A G A C A U U C U U C
 worm  G A C G C U U C G G C C G C A A C C C G C G U U
yeast  C G U U C U U C G G C A A A A A C U G U C A U U
       1       5         10        15        20      24
```

Figure 1. An example of how an RNA sequence/structure profile captures more information than a standard sequence profile. A multiple alignment is shown for eight homologous RNAs, all of which share the same consensus secondary structure (shown at right, for the "human" sequence). Below the alignment is a bar graph of the average expected score (information content) per position, in bits, for a sequence profile. In this contrived example, the only values are 2 bits (a 100% conserved residue), 1 bit (2 possible residues, 50% each), or 0 bits (no conservation, 25% each for all 4 residues). Above the alignment is a sequence/structure profile, a binary tree (instead of a linear array of columns) in which the six consensus base pairs are captured as six pairwise states (*red bars*) instead of as 12 single uncorrelated columns.

for base-paired RNA structure (Eddy and Durbin 1994; Sakakibara et al. 1994). To understand the generality of SCFGs, it is useful to make a brief digression into the current state of the art in linear sequence analysis.

PROBABILISTIC MODELS OF BIOLOGICAL SEQUENCES

Historically, sequence alignment algorithms were developed independently of statistical scoring methods. Different analysis applications typically used their own special algorithm(s) and parameterization methods. However, today, many primary sequence analysis methods, including similarity search, motif identification, and genefinding applications, are viewed by many computational biologists in a single formal framework called *hidden Markov models* (HMMs) (or more generally, *stochastic regular grammars*, and related models) (Durbin et al. 1998). Using HMM formalisms, every score is based on a probabilistic model—not just residue scores, but also any insertion/deletion penalties. HMMs allow straightforward creation of more complex yet consistent models that combine multiple information sources (protein-coding genefinders, for example [Burge and Karlin 1997], or position-specific profiles of conserved protein domains [Krogh et al. 1994]).

A key point is that any HMM, no matter how complicated, can be optimally aligned to a target sequence using a general algorithm called the Viterbi algorithm. For the special case of simple HMMs for sequence alignment, the Viterbi algorithm becomes identical to the well-known Smith/Waterman or Needleman/Wunsch sequence alignment algorithms (Needleman and Wunsch 1970; Smith and Waterman 1981).

Thus, adopting an explicit HMM framework has the advantage of splitting a sequence analysis problem into three pieces, two of which are standardized. The first piece is specifying the structure of the HMM. This is the interesting bit, where one decides what biological information to capture. The second piece is calculating the probability parameters (scores) of an HMM, which is standard probability theory. The third piece is how to align HMMs to target sequences, which is done with the standard Viterbi algorithm (and related HMM algorithms). By focusing specialization effort on the design of new models for different biological problems, rather than on the shared computational and statistical foundation, probabilistic models like HMMs give us powerful, biologically intuitive, and general toolkits for building a wide variety of sequence analysis methods of varying complexity and realism (Eddy 2004).

An HMM, however, is still "just" a primary sequence analysis method, scoring linear sequence one (or a few) residue at a time. HMMs cannot efficiently capture the long-distance pairwise correlations in RNA secondary structure.

STOCHASTIC CONTEXT-FREE GRAMMARS

In computational linguistics, HMMs are *stochastic regular grammars*, at the lowest level of a hierarchy of formal grammars originally defined by Noam Chomsky (1956) for the purpose of understanding the structure of natural languages. The next level in Chomsky's hierarchy are the so-called *context-free grammars,* or in probabilistic form as opposed to pattern-matching form, *stochastic context-free grammars* (SCFGs). One thing SCFGs can do that HMMs cannot is to efficiently model nested, long-

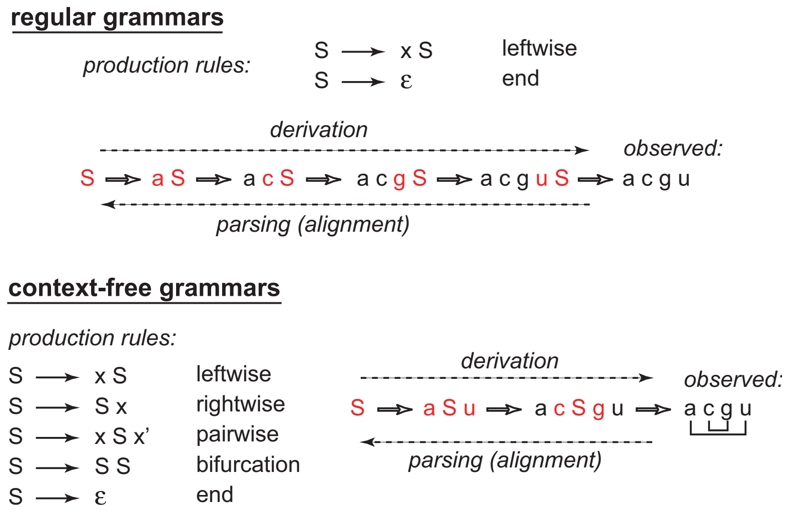


Figure 2. Comparison of regular grammars (linear sequence models) and context-free grammars, which can capture nested pairwise correlations. Productions of possible residues are shown as single rules generating a generic x (or ε, for end productions that generate a null symbol), rather than enumerating all possible 4 residues or 16 residue pairs. See text for more explanation.

distance pairwise correlations in strings of symbols—rare in natural languages, as it happens, but exactly what we need for RNA analysis. Shortly after HMMs were introduced into computational biology as general models of primary sequence analysis, SCFGs were brought into the field as general models of RNA sequence and structure (Eddy and Durbin 1994; Sakakibara et al. 1994).

Figure 2 briefly sketches the salient features of linear sequence algorithms (using regular grammars) and RNA sequence/structure algorithms (using context-free grammars). Both are *generative models*, consisting of a set of *production rules* that generate good sequences (those that belong to a homologous family, or align to a homologous query, or fit a gene model) with higher probability than other sequences. Production rules consist of *nonterminal symbols* (also called states) and *terminal symbols* (the observed A, C, G, U residues). Production rules describe the probabilistic expectation for what residues are favored in different places and for what states follow others. In essence, we use one state or production rule for each different way that we might want to score a residue. For example, a linear sequence profile of a multiple alignment might consist of a linear array of one state per consensus alignment column. An RNA structure profile might consist of one state per consensus base pair and one state per consensus single-stranded position. A genefinder might use two or more states to describe different residue compositions in exons versus introns.

Regular grammars are linear models. Production rules simply generate a symbol and move to a new state, from left to right. Still, regular grammar rules can capture a fair amount of complexity. For instance, to deal with insertions and deletions in a profile, we might move to the next consensus state, or move to an insertion state (and possibly stay there for a few self-transitions) to model a traditional gap-open/gap-extend penalty, or skip one or more of the next consensus states to model a deletion penalty.

Context-free grammars (CFGs) generate sequence outside-in, rather than left to right. For RNA, the crucial property of a CFG is that one production rule can generate a correlated pair of residues, then another correlated pair inside that. In an SCFG, a base-pair production is associated with a 4 x 4 probability table for all possible residue pairs, including Watson–Crick as well as noncanonical pairs. CFGs also allow one to fork off two or more substructures, so they can describe complex RNA secondary structures with multiple stem-loops.

In an actual application, the problem is not to generate simulated sequences, but to align a model to a given target sequence and assign it a score. The alignment problem is called *parsing* in linguistics. We aim to determine the optimal (highest probability) series of production rules that would have generated the target sequence. Just as all HMMs have a common parsing algorithm (the Viterbi algorithm), all SCFGs have a common parsing algorithm, the CYK algorithm.

The CYK algorithm identifies the best-scoring manner in which the model can generate the target sequence. The resulting so-called *parse tree* is the RNA secondary structure analog of a sequence alignment (Fig. 3),

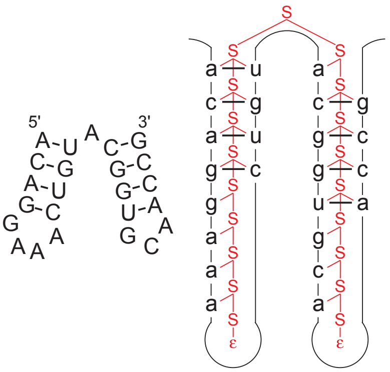


Figure 3. An SCFG parse tree (*right*) corresponding to a small example RNA secondary structure (*left*), for the set of five types of production rules in Fig. 2.

describing an assignment of residues in the target sequence to states of the query SCFG. A parse tree essentially specifies how to factorize an RNA alignment into a sum of additive scoring units, such as base pairs and singlet residues, or in more complicated models, base stacks and different lengths and types of loops.

Different SCFGs can be drawn for different problems (structure prediction, similarity search, or genefinding), depending on what statistical information one wants to capture in the model. Like HMMs for sequence analysis, SCFGs are a general toolkit for probabilistic modeling of RNA sequence and secondary structure.

LIMITATIONS OF SCFGs

The most important limitation of SCFGs is their computational complexity. Sequence analysis algorithms such as BLAST typically take time and memory proportional to L^2 for comparing two sequences of length L in residues, because the set of all possible alignments must try every residue in the query against every residue in the target. SCFG-based RNA analysis algorithms require time and memory proportional to at least L^3, because every possible pair of residues (L^2) must be tried against up to $L/2$ base-pairing states in the model (and in most RNA SCFGs, the time required more typically scales as L^4.) Thus, roughly speaking, for a typical RNA of length 100–1000 residues, compared to linear sequence analysis, SCFG-based algorithms take 100–1000-fold more memory and 10^4–10^6-fold more time. Although this is high, it should be noted that this kind of compute complexity is not foreign to biological sequence analysis. Well-known RNA secondary structure prediction programs such as the Zuker MFOLD program (Zuker 2000) have the same computational complexity (not a coincidence, because the MFOLD folding algorithm is essentially a special case of the SCFG CYK algorithm). Nonetheless, computational complexity is a very serious problem for SCFGs, especially in database-searching applications. Although the standard CYK algorithm is a useful general starting point for all SCFGs, much of the work in my lab is devoted toward finding more efficient algorithms.

A second limitation of SCFGs is that they can only model *nested* pairwise correlations. RNA pseudoknots, which involve nonnested interactions, cannot be described by SCFG formalisms. More powerful grammar classes (so-called "mildly context-sensitive grammars") can model RNA pseudoknots, but their computational complexity is currently prohibitive for many applications (Rivas and Eddy 1999). Likewise, base triples are usually prohibited, because they also usually involve nonnested pairing. These losses are unfortunate and would be especially serious if three-dimensional RNA structure prediction were the goal. But for many objectives, including homology search, motif identification, and genefinding, this information loss is an acceptable tradeoff. RNA pseudoknots typically account for something like 5–10% of the base pairs in most RNA structures. There is still a large gain in being able to capture most of the pairwise correlation information in an RNA structure in an efficient computational model.

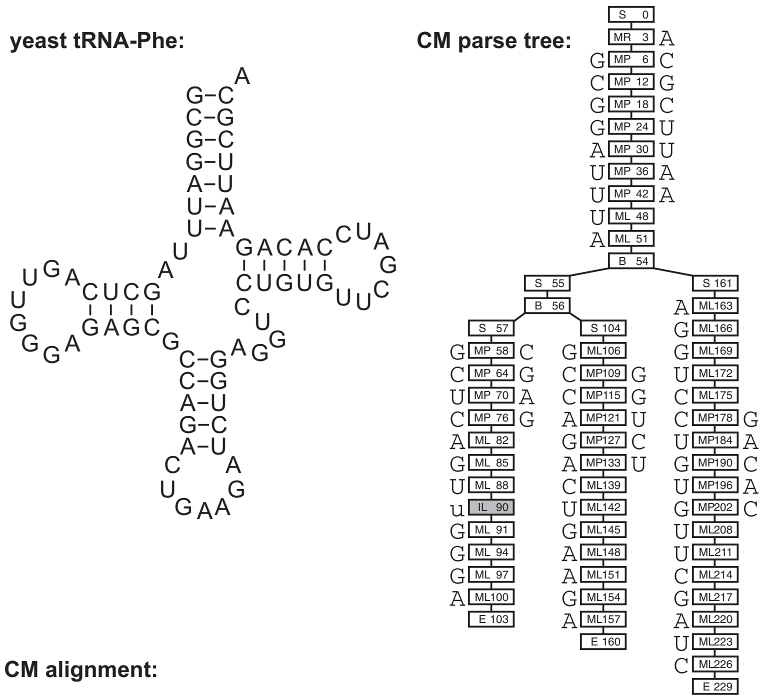
SCFG-BASED RNA SIMILARITY SEARCH PROGRAMS

My lab has been exploring the use of SCFGs for a variety of tasks, including RNA structure prediction (Dowell and Eddy 2004 and in prep.) and noncoding RNA genefinding (Rivas and Eddy 2001). Rather than survey all this work, I continue to focus here on RNA similarity searching applications. So far, I have discussed the formal benefits of SCFGs from a mathematical and computational point of view. This leaves an important question—How well do they actually work?

Again, our problem is, given either a single RNA sequence and its secondary structure, or a RNA multiple alignment and a known consensus structure, we want to search a sequence database for similar sequences. A position-specific SCFG is constructed which has a set of 4 scores for each single-stranded position, 16 scores for each base pair, and appropriate extra states and state transitions that allow for insertions and deletions. Because there are many ways to deal with insertions and deletions, in terms of where to allow them and how to score them, there are many ways one can convert an RNA structure query into SCFG production rules. I adopted one particular general convention for building SCFGs for similarity searching, called "covariance models" (CMs) (Eddy and Durbin 1994; Eddy 2002). The conventions in CMs follow, as closely as possible, conventions in linear sequence analysis. Insertions and deletions are allowed anywhere, and are assigned gap-open and gap-extend penalties (affine gap penalties). Figure 4 shows an example of a CM alignment and parse tree for a tRNA profile aligned to yeast phenylalanine tRNA.

Essentially the same CM structure and alignment algorithms are the basis of three software packages from my lab. Two are for consensus profiles of multiple alignments—COVE (Eddy and Durbin 1994) (now obsolete) and COVE's replacement INFERNAL (Eddy 2002). The third, Robbie Klein's RSEARCH, is for searching with single RNA sequence/structure queries (Klein and Eddy 2003). INFERNAL is the RNA secondary structure analog of the HMMER profile HMM software for sequence analysis (Eddy 1998), and RSEARCH is the analog of Smith/Waterman single query sequence alignment (Smith and Waterman 1981).

One of the first practical applications of CMs for similarity search demonstrated both the power and the limitations of SCFG-based approaches. Todd Lowe in my lab developed a program for tRNA gene identification, TRNASCAN-SE, as a wrapper script around a CM built from a large alignment of known tRNAs (Lowe and Eddy 1997). At the time, the best tRNA gene identification programs had false-positive rates of about 0.2–0.3 per megabase and sensitivities of about 95–99% for known tRNAs (Fichant and Burks 1991; Pavesi et al. 1994). These programs were quite adequate for small genomes, such as *E. coli* or *Saccharomyces cerevisiae*, where they identified only a small number of false positives (≤10), but we realized that for large genomes like the 3000 Mb human genome, these false positive rates would become a problem. We expected only about 500 or so true tRNA



CM alignment:

```
    (((((((,,<<<<___ ____>>>>,<<<<<_______>>>>>,,,,,<<<<<_______
  1 GggggauaUAGCUcAGU.GGUAgAGCaccgGaCUuauAAuCcggaGGUCgcgGGUUCGAaU 59
    G:G AU:UAGCUCAGU GG AGAGC+CC:GACU+A+ AUC:GGAGGUC::G:GUUCGA
  1 GCGGAUUUAGCUCAGUuGGGAGAGCGCCAGACUGAAGAUCUGGAGGUCCUGUGUUCGAUC 60

    >>>>>))))))).
 60 CCcgcuaucccCA 72
    C:C:::AU C:CA
 61 CACAGAAUUCGCA 73
```

Figure 4. A real example of a CM parse tree (*top right*) for yeast phenylalanine tRNA (*top left*), for a CM constructed from a tRNA alignment, similar to the CM used by the TRNASCAN-SE program. Below, the two-dimensional parse tree is represented by the program as a four-line sequence alignment between the consensus of the query CM (*second line*) and the target yeast tRNA sequence (*fourth line*), in a format akin to BLAST output format. The output format is augmented in two ways to indicate RNA secondary structure. First, : symbols in the identity line (*third line*) indicate positive-scoring compensatory base pairs. (+ symbols indicate positive-scoring single residues as in BLAST). Second, an extra line of annotation (first line) annotates the base pairs in the consensus secondary structure with <> and ( ) pairs (and other symbols annotating different single-stranded residues).

genes in the human genome, but the programs were going to predict more than 1000 false positives. We needed to increase the specificity of tRNA gene identification by orders of magnitude if we were going to be able to annotate the tRNA gene family in large genomes.

Lowe showed that a CM built automatically by the COVE software from a large tRNA sequence alignment database (Steinberg et al. 1993) achieved 99.8% sensitivity with less than 0.002 false positives per megabase (Lowe and Eddy 1997). On the other hand, the search speed was far too slow for whole-genome analysis. We estimated needing 10 CPU (central processing unit)-years for a human genome, on 1997 CPUs. Since the existing programs were fast and sensitive, just not specific enough, Lowe solved the speed problem by using a combination of two existing programs as prefilters (Fichant and Burks 1991; Pavesi et al. 1994), and only passing their proposed tRNAs to COVE and the tRNA CM. The combination of programs resulted in TRNASCAN-SE, which showed 99.5% average sensitivity on tRNA genes, a false positive rate below the detection limit of our simulations (<1 per 15 gigabases), and we could process the human genome in about 2 CPU-days (on 1997 CPUs).

Today, TRNASCAN-SE seems to still be the standard for whole-genome annotation of tRNA genes, although a newer heuristic program, ARAGORN, now has comparable performance (Laslett and Canback 2004). TRNASCAN-SE has, on occasion, even produced interesting new tRNA biology. For example, the CM in TRNASCAN-SE detected the noncanonical tRNA for the "22nd amino acid," pyrrolysine, in the *Methanosarcina barkeri* genome (Srinivasan et al. 2002). The principal failings of the program are largely unrelated to the similarity detection power of its CM. It has trouble distinguishing tRNA pseudogenes from true tRNA genes, and some genomes, such as the rat, have thousands of tRNA pseudogenes (Gibbs et al. 2004). It also has trouble correctly annotating some tRNA isoacceptor types in cases where a tRNA anticodon is posttranscriptionally modified.

TRNASCAN-SE was a nice demonstration of SCFGs. However, tRNAs are an unusually ideal case, in several ways:

1. tRNAs are small (~75 nucleotides), so the $O(L^3)$ memory requirement of CM alignment algorithms did not have serious impact. tRNA-sized RNAs can be aligned in about 1 MB RAM with the standard CYK algorithm. However, for larger RNAs, memory requirements could become prohibitive. We estimated that RNase P alignments (~400 nucleotides) would take 340 MB RAM, and LSU rRNA alignments (~2900 nucleotides) would take 150 GB RAM.
2. We had a highly reliable, deep, manually curated alignment of 1415 tRNAs to use to estimate the tRNA CM's parameters (Steinberg et al. 1993). Thus, we could use observed frequencies of single-stranded residues, base pairs, and indels as parameters, without any need for more sophisticated parameterization methods, such as the use of mixture Dirichlet priors in profile HMM parameter estimation (Sjölander et al. 1996), or the determination of general RNA substitution score matrices akin to the use of BLOSUM matrices in BLAST.
3. Structural variation in tRNAs is minimal. Almost all tRNAs adopt a four-stemmed cloverleaf consensus structure. Most structural RNAs—even ribosomal RNA—show much more substantial structural variation over evolutionary time, with stems or even whole substructural domains coming and going. Global alignment to a single consensus model works fine for tRNA, but fails for most other RNAs.
4. The $O(L^4)$ time requirement of the CM search algorithm is punitive, even for RNAs as small as tRNA. The only reason we could make a practical tRNA search program based on CM algorithms is that fast rule-based search programs were already available, which we could use as effective prescreens.

Several lines of research in my lab have focused on addressing each of these four problems in order to make RNA similarity searching practical.

IMPROVED MEMORY USAGE

The memory requirement of CMs was at first the most serious barrier. This issue was essentially solved in 2002 (Eddy 2002). Using an approach analogous to approaches already in common use in sequence alignment, I developed a "divide and conquer" (Hirschberg 1975) variant of the CYK algorithm that reduces the cubic $O(L^3)$ dynamic programming lattice for CMs to $O(L^2 \log L)$ while still guaranteeing a mathematically optimal alignment (Eddy 2002). The price is a relatively negligible extra factor in time (an extra 20% on average, as measured empirically). tRNA alignments now cost 0.1 megabytes, RNase P costs 4 MB, and LSU rRNA costs 270 MB, all well within the capabilities of standard current computers.

IMPROVED PARAMETERIZATION

Techniques for parameterizing probabilistic models are well understood. The same methods that are used for sequence alignment scores work for CMs, so once the memory problem was no longer limiting, it just required applying known methods to CMs.

For CMs built from single RNA query structures, Robbie Klein developed the RIBOSUM substitution matrices, 4 x 4 matrices for scoring single-residue alignments and 16 x 16 for scoring base-pair alignments (Klein and Eddy 2003). Klein implemented these in a program called RSEARCH. RSEARCH uses CM formalisms and the INFERNAL CM alignment engines, but parameterizes the models with RIBOSUM substitution scores and arbitrary gap penalities, much like standard sequence alignment algorithms.

For CMs built as profiles of a multiple RNA sequence alignment of known consensus structure, Eric Nawrocki estimated informative mixture Dirichlet priors from known RNA structural alignments and implemented these priors in INFERNAL (E.P. Nawrocki and S.R. Eddy, unpubl.). INFERNAL profiles built from only a few aligned sequences (five, for example) now seem to perform reasonably. (Quantifying "reasonably" would require a digression into how similarity search programs are benchmarked, which I will forego here.)

LOCAL STRUCTURAL ALIGNMENT

In most structural RNAs, not all of the secondary structure is conserved over evolutionary time. A computational model that requires a global RNA structural alignment is not ideal. We extended CMs to allow local structure alignment (Eddy 2002; Klein and Eddy 2003). The basic idea is in two parts. One is to allow a parse tree to start at any consensus position, instead of always starting at the root of the tree. The second is to allow the model to end prematurely from any consensus state and generate zero or more nonhomologous random residues before stopping. These rules allow large deletions and truncations of structural subdomains, where that deletion is consistent with the rest of the conserved secondary structure. For example, Figure 5 shows an INFERNAL alignment of a gamma-proteobacterial RNase P RNA profile to the *Bacillus subtilis* RNase P RNA, in which four substructural domains have to be deleted or inserted to see the homology.

The combination of the above three improvements has made it possible to routinely see remote RNA homologies that were previously below the radar of existing sequence-based approaches. RNase P RNAs are an interesting example. RNase P RNA is one of the best-studied catalytic RNAs and is thought to be nearly universally conserved in all domains of life. However, few metazoan RNase Ps had been identified until recently (Marquez et al. 2005; Piccinelli et al. 2005), even in sequenced model organisms like *Caenorhabditis elegans* and *Drosophila melanogaster*, because many RNase P RNA homologs are not detected by BLAST searches. A single RSEARCH search, using the human RNase P RNA structure as a query, cleanly identifies single RNase P RNA homologs in *C. elegans*, *D. melanogaster*, and several other eukaryotic genomes, with significant E-values. The predicted *C. elegans* RNase P is shown in Figure 6, along with the structure of the human RNase P query and the alignment output from RSEARCH.

```
    {{{{{{{{{{{{{{{{{{,<<<<<<<<<<<<<-<<<<<____>>>>>>>>>->>>>>>>>
  1 ggAGuggGgcaGgCaguCGCugcuucggccuuGuucaguuaacugaaaaggAccgaagga
    +: :::G::C:GG:A:UCGCU+C::::            U+            ::::G+A
  4 CUUAACGUUCGGGUAAUCGCUGCAGAUC-----------UUG----------AAUCUGUA
        P1       P2        P3                              P3'

    >,,,,,,,,,,,,,[[[ [--------[[[[[       ((---(((((,,,,      )
 61 GAGGAAAGUCCGGGCUC.CACAGGGCAgGGUG       GGAAAGUGCCACAG      G
    GAGGAAAGUCC  GCUC C  A GG   :G G       :GAAAGUGCCACAG      G
 43 GAGGAAAGUCCAUGCUCgC--ACGGUGCUGAG*[102]*UGAAAGUGCCACAG*[37]*G
                    P5          [P7]     [P10]   P11

    ))--))))]]]]]] ]]],,,      ,,,,,,,,,,}}}}}}}--
230 GUAAACCCCACCcG.GAGCAA      CuAGAUGAAUGacuGcCCA..............
    GUAAACC:C C: G GAG AA       UAGAU++AUGA:U:CC
227 GUAAACCCCUCGAGcGAGAAA*[64]*GUAGAUAGAUGAUUGCC--gccugaguacgagg
   P11'  P10' P7'  P5'                    P2'

                              -----------------}-}}}}}}}}}}....
345 ..........................CGACAGAACCCGGCUUAuagcCccaCUccucuu
                                ACA AAC  GGCUUA:AG::C::: :+ C
343 ugaugagccguuugcaguacgaugga--ACAAAACAUGGCUUACAGAACGUUAGACCAC
                                                    P1'
```

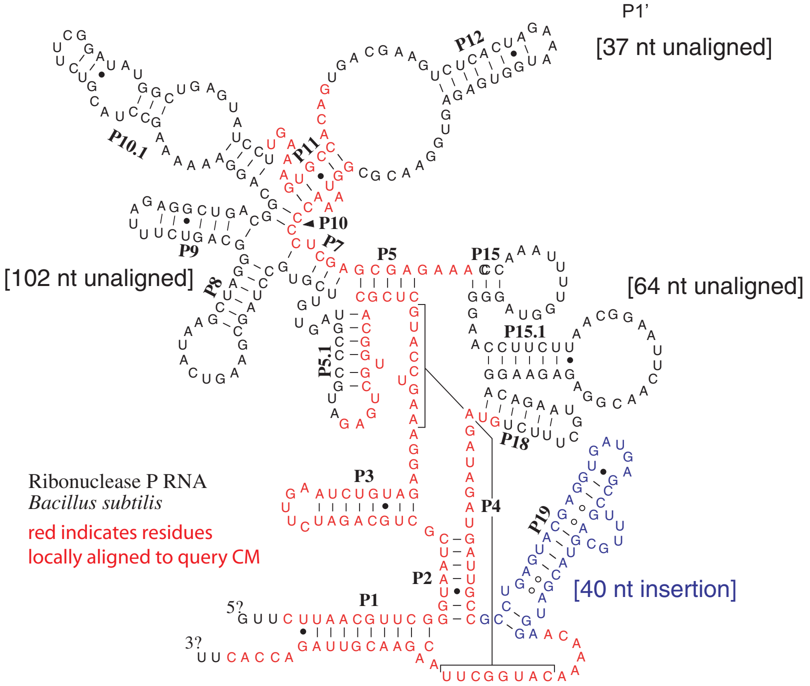
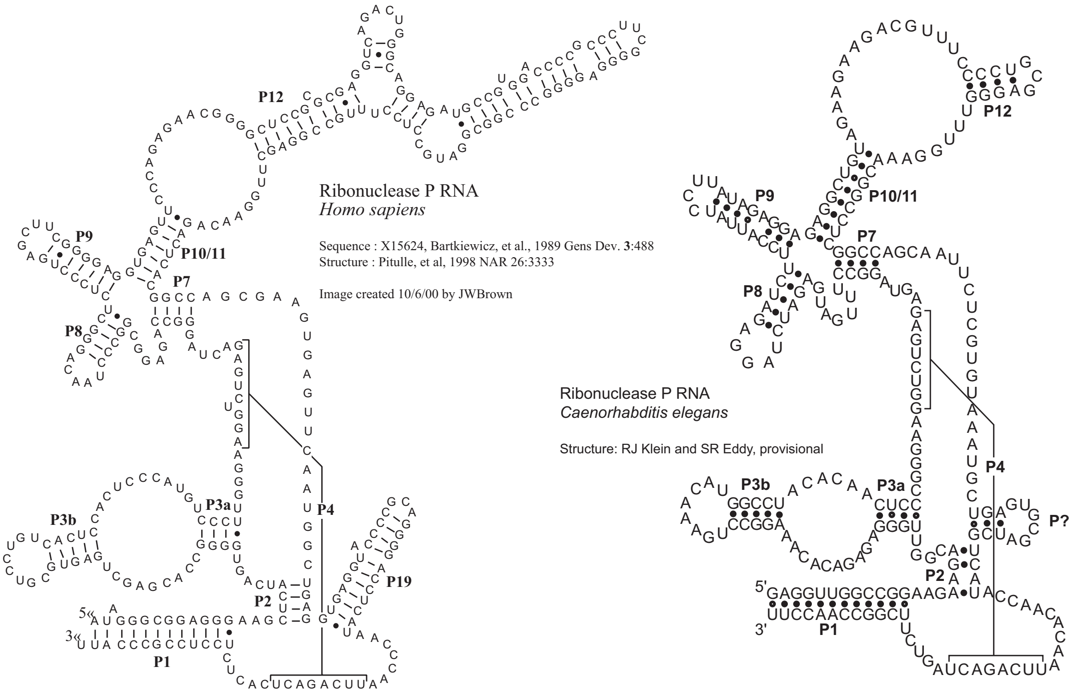


Figure 5. Example of output for a local structural alignment, when a CM built from five homologous gamma-proteobacterial RNase P RNAs (including *E. coli*) successfully detects a homologous alignment in the RNase P RNA of *B. subtilis* RNase P with a significant score of 49.4 bits. (*Top*) In the maximum likelihood alignment, three substructural domains of 102 (P8, P9, P10.1 helices), 37 (P12), and 64 (P15, P15.1, P18) nucleotides in the *B. subtilis* structure are treated as nonhomologous by the CM's local alignment rules, and an additional 40-nucleotide domain (P19) is handled as an insertion. The predicted P1–P11 stem regions are annotated beneath the output lines, with italics indicating mispredictions of the P7 and P10 stems as the result of the large structural variations in these regions. (*Bottom*) The accepted secondary structure of the *B. subtilis* RNase P is shown (Brown 1999), with red indicating which residues were aligned to the query model. These aligned residues roughly correspond to the conserved core of the RNase P three-dimensional structure.

SPEED IMPROVEMENTS

The remaining problem is the slow speed of CM searches. For example, whereas a BLAST search of the *C. elegans* genome with a mammalian RNase P RNA query takes CPU-seconds, an INFERNAL or RSEARCH search takes CPU-months. On the other hand, the BLAST search does not find anything significant, whereas SCFG searches do. Our first priority has been to get the right answer (however slowly), but now it is time to worry about speed. Up until now, we have been able to address the computational speed problem by brute force, by parallelizing our search programs and running them on a cluster (~300 Linux processors in the St. Louis lab), but this is not a satisfactory long-term solution. We and other investigators are working on accelerated CM search algorithms. Zasha Weinberg and Larry Ruzzo at the University of Washington in Seattle have developed a clever "rigorous filter" approach, which Diana Kolbe in my lab has incorporated into the INFERNAL codebase (Weinberg and Ruzzo 2004, 2006). Eric Nawrocki and I have developed a complementary method, query-dependent banding, a banded dynamic programming algorithm specific to CMs. We think that the combination of these acceleration methods


Ribonuclease P RNA
Homo sapiens
Sequence : X15624, Bartkiewicz, et al., 1989 Gens Dev. 3:488
Structure : Pitulle, et al, 1998 NAR 26:3333
Image created 10/6/00 by JWBrown
P1
P2
P3a
P3b
P4
P7
P8
P9
P10/11
P12
P19
5'
3'
Ribonuclease P RNA
Caenorhabditis elegans
Structure: RJ Klein and SR Eddy, provisional
P1
P2
P3a
P3b
P4
P?
P7
P8
P9
P10/11
P12
5'
3'

```
Score = 91.22 bits, E = 1.656e-05

-{{{{{{{{{{,,,[[[[,,,,,<<<<      >>>>,,,,,,,,,,,,,,,,,,,((((,,,,,,< <<<<_____>>>>><<<<<<<_____ >>>>>>><<<<<< <----------
AGGGCGGAGGGAAGCUCAUCAGUGGGG      CCCUUGGGAAGGUCUGAGACUAGGGCCAGAGGCG.GCCCUAACAGGGCUCUCCCUGAGCUU.CGGGGAGGUGAGU.UCCCAGAGAAC
AGG++GG++GGAAG+  + C GU G+G      C+C  GGGAAGGUCUGAGA UAGG CC G    G G++CUA  AG++CU++C +++  CUU +++ G++G+G++U U   AGA  AC
AGGUUGGCCGGAAGA-AGACGGUUGGG*[34]*CUCCCGGGAAGGUCUGAGAGUAGG-CCUUUGAUGaGAUCUAGGAGAUCUUCCAUUAUCCUUaUAGAGGAGAGGCUgUAGAAGAAGAC

---<<<__     >>>-------->>>>>>>)))),,,,,,,,,,,,,,,,,,,,]]] ]<<<<< <<     >>>> >>>,,,,,,,,,,,,,,,,,,,} }}}}}}}}}
GGGGCUCC     AGCUUGGAACAGACUCACGGCCAGCGAAGUGAGUUCAAUGGCUGA.GGUGAG.GU     ACCU.CAUAACCCAAUUCAGACCACUCU.CCUCCGCCC 338
G   C+CC     +G UUGGAA  G ++C+CGGCCAGC A     GU  AA  GC+ A ++ GA+ GU      C+U C +AAC CAAUUCAGAC A UCU C++CC++CC
GUUUCCCC*[6]*GGUUUGGAAACG-GCCUCGGCCAGCAAUUCUCGUGUAAAUGCUGAgUGCGAUcGU*[0]*-CAUaC-CAACACAAUUCAGACUAGUCUuCGGCCAACC 250
```

Figure 6. The RSEARCH program, with a human RNase P RNA query structure shown at top left (Brown 1999), detects one significant alignment in the *C. elegans* genome (*bottom*) with an E-value of 1.656e-5. We believe that this is the *C. elegans* RNase P RNA. The same sequence was detected by Steve Marquez and Norman Pace in 2005 (Marquez et al. 2005). Our tentative structure prediction, with a few manual corrections from the RSEARCH alignment, is shown in the upper right. The structure shown for the P2/P? region is uncertain (and different from that predicted by the RSEARCH alignment).

should soon give us about 10–100 times improvement in speed in INFERNAL's publicly distributed code. CM approaches will still be much more compute-intensive than BLAST, but they might start to become feasible on single desktop CPUs.

THE RFAM DATABASE

In collaboration with us, Sam Griffiths-Jones and coworkers at the Wellcome Trust Sanger Institute have developed a database called Rfam, which contains curated multiple alignments and CMs for known RNA sequence families (Griffiths-Jones et al. 2005). The current Rfam 7.0 release contains 503 families. Rfam is an RNA analog of the Pfam protein domain database, using INFERNAL software where Pfam uses the HMMER profile HMM software. Rfam makes it possible to automatically detect and annotate homologs of known RNA structures in genome sequences. At present, for speed reasons, Rfam processing relies on BLAST prefilters, however, so some of the added sensitivity that full CM searches could provide is sacrificed. We hope to gain this back as CM search speed increases.

CONCLUSION

Developing better computational sequence analysis tools is like building better telescopes. With more and more powerful tools, we are trying to peer into the genome and discern the subtle signals left by functional elements that have diverged by billions of years of evolution. As our resolution power goes up, features come into sharper focus. True breakthroughs are relatively rare, because most features have been seen already, albeit at lower resolution and in less detail. Nonetheless, over time, steady incremental advances in technology can amount to surprising overall gains in power.

The advent of SCFGs for RNA sequence analysis was a significant *theoretical* advance, making it possible to harness RNA secondary structure constraints in almost arbitrarily complex, fully automated computational methods while still using formally well-grounded probabilistic modeling. However, converting the promise of SCFG formalisms to *practice* in RNA analysis has been a more usual case of incremental progress, requiring practical software implementations and a lot of work. We have almost reached the point of routine practical applications in RNA similarity search, with only computational time as our remaining barrier. In other areas, such as SCFG-based noncoding RNA genefinders, SCFG-based RNA structure prediction by comparative analysis, and SCFG-based RNA structural motif discovery in unaligned sequences, even more serious barriers still remain, but practical applications are developing in those areas as well.

ACKNOWLEDGMENTS

I am grateful to Elena Rivas, Ariane Machado-Lima, and Eric Nawrocki for comments on the manuscript. I thank the National Institutes of Health National Human Genome Research Institute, the Howard Hughes Medical Institute, and Alvin Goldfarb for their financial support of my group.

REFERENCES

Altschul S.F., Madden T.L., Schaffer A.A., Zhang J., Zhang Z., Miller W., and Lipman D.J. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **25:** 3389.

Babak T., Blencowe B.J., and Hughes T.R. 2005. A systematic search for new mammalian noncoding RNAs indicates little conserved intergenic transcription. *BMC Genomics* **6:** 104.

Bachellerie J.P., Cavaille J., and Hüttenhofer A. 2002. The expanding snoRNA world. *Biochimie* **84:** 775.

Bartel D.P. 2004. MicroRNAs: Genomics, biogenesis, mechanism, and function. *Cell* **116:** 281.

Brown J.W. 1999. The ribonuclease P database. *Nucleic Acids Res.* **27:** 314.

Brown J.W., Echeverria M., and Qu L.H. 2003. Plant snoRNAs: Functional evolution and new modes of gene expression. *Trends Plant Sci.* **8:** 42.

Burge C. and Karlin S. 1997. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* **268:** 78.

Carninci P., Kasukawa T., Katayama S., Gough J., Frith M.C., Maeda N., Oyama R., Ravasi T., Lenhard B., Wells C., et al. 2005. The transcriptional landscape of the mammalian genome. *Science* **309:** 1559.

Cheng J., Kapranov P., Drenkow J., Dike S., Brubaker S., Patel S., Long J., Stern D., Tammana H., Helt G., et al. 2005. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science* **308:** 1149.

Chomsky N. 1956. Three models for the description of language. *IRE Trans. Inf. Theory* **2:** 113.

Dowell R.D. and Eddy S.R. 2004. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics* **5:** 71.

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

Eddy S.R. 1998. Profile hidden Markov models. *Bioinformatics* **14:** 755.

———. 2001. Non-coding RNA genes and the modern RNA world. *Nat. Rev. Genet.* **2:** 919.

———. 2002. A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics* **3:** 18.

———. 2004. What is a hidden Markov model? *Nat. Biotechnol.* **22:** 1315.

———. 2005. A model of the statistical power of comparative genome sequence analysis. *PLoS Biol.* **3:** e10.

Eddy S.R. and Durbin R. 1994. RNA sequence analysis using covariance models. *Nucleic Acids Res.* **22:** 2079.

ENCODE Project Consortium. 2004. The ENCODE (ENCyclopedia of DNA Elements) project. *Science* **306:** 636.

Fichant G.A. and Burks C. 1991. Identifying potential tRNA genes in genomic DNA sequences. *J. Mol. Biol.* **220:** 659.

Furuno M., Pang K.C., Ninomiya N., Fukuda S., Frith M.C., Bult C., Kai C., Kawai J., Carninci P., Hayashizaki Y., et al. 2006. Clusters of internally primed transcripts reveal novel long noncoding RNAs. *PLoS Genet.* **2:** e37.

Gautheret D. and Lambert A. 2001. Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. *J. Mol. Biol.* **313:** 1003.

Gibbs R.A., Weinstock G.M., Metzker M.L., Muzny D.M., Sodergren E.J., Scherer S., Scott G., Steffen D., Worley K.C., Burch P.E., et al. 2004. Genome sequence of the Brown Norway rat yields insights into mammalian evolution. *Nature* **428:** 493.

Godzik A. 2003. Fold recognition methods. *Methods Biochem. Anal.* **44:** 525.

Griffiths-Jones S., Moxon S., Marshall M., Khanna A., Eddy S.R., and Bateman A. 2005. Rfam: Annotating non-coding RNAs in complete genomes. *Nucleic Acids Res.* **33:** D121.

Gutell R.R., Lee J.C., and Cannone J.J. 2002. The accuracy of ribosomal RNA comparative structure models. *Curr. Opin. Struct. Biol.* **12:** 301.

Hirschberg D.S. 1975. A linear space algorithm for computing maximal common subsequences. *Commun. ACM* **18:** 341.

Hüttenhofer A., Schattner P., and Polacek N. 2005. Non-coding RNAs: Hope or hype? *Trends Genet.* **21:** 289.

Klein R.J. and Eddy S.R. 2003. RSEARCH: Finding homologs of single structured RNA sequences. *BMC Bioinformatics* **4:** 44.

Krogh A., Brown M., Mian I.S., Sjölander K., and Haussler D. 1994. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.* **235:** 1501.

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

Laslett D. and Canback B. 2004. ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic Acids Res.* **32:** 11.

Lee L.J., Hughes T.R., and Frey B.J. 2006. How many new genes are there? *Science* **311:** 1709.

Lee R.C., Feinbaum R.L., and Ambros V. 1993. The *C. elegans* heterochronic gene *lin-4* encodes small RNAs with antisense complementarity to *lin-14*. *Cell* **75:** 843.

Lowe T.M. and Eddy S.R. 1997. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.* **25:** 955.

Marquez S.M., Harris J.K., Kelley S.T., Brown J.W., Dawson S.C., Roberts E.C., and Pace N.R. 2005. Structural implications of novel diversity in eucaryal RNase p RNA. *RNA* **11:** 739.

Mattick J.S. and Makunin I.V. 2005. Small regulatory RNAs in mammals. *Hum. Mol. Genet.* **14:** R121.

Needleman S.B. and Wunsch C.D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48:** 443.

Numata K., Kanai A., Saito R., Kondo S., Adachi J., Wilming L.G., Hume D.A., Hayashizaki Y., Tomita M., RIKEN GER Group, and GSL Members. 2003. Identification of putative noncoding RNAs among the RIKEN mouse full-length cDNA collection. *Genome Res.* **13:** 1301.

Okazaki Y., Furuno M., Kasukawa T., Adachi J., Bono H., Kondo S., Nikaido I., Osato N., Saito R., Suzuki H., et al. (FANTOM Consortium; RIKEN Genome Exploration Research Group Phase I & II Team). 2002. Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cDNAs. *Nature* **420:** 563.

Omer A.D., Lowe T.M., Russell A.G., Ebhardt H., Eddy S.R., and Dennis P.P. 2000. Homologs of small nucleolar RNAs in *Archaea*. *Science* **288:** 517.

Pace N.R., Smith D.K., Olsen G.J., and James B.D. 1989. Phylogenetic comparative analysis and the secondary structure of ribonuclease P RNA: A review. *Gene* **82:** 65.

Pavesi A., Conterlo F., Bolchi A., Dieci G., and Ottonello S. 1994. Identification of new eukaryotic tRNA genes in genomic DNA databases by a multistep weight matrix analysis of transcriptional control regions. *Nucleic Acids Res.* **22:** 1247.

Piccinelli P., Rosenblad M.A., and Samuelsson T. 2005. Identification and analysis of ribonuclease P and MRP RNA in a broad range of eukaryotes. *Nucleic Acids Res.* **33:** 4485.

Ravasi T., Suzuki H., Pang K.C., Katayama S., Furuno M., Okunishi R., Fukuda S., Ru K., Frith M.C., Gongora M.M., et al. 2006. Experimental validation of the regulated expression of large numbers of non-coding RNAs from the mouse genome. *Genome Res.* **16:** 11.

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

———. 2001. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics* **2:** 8.

Rivas E., Klein R.J., Jones T.A., and Eddy S.R. 2001. Computational identification of noncoding RNAs in *E. coli* by comparative genomics. *Curr. Biol.* **11:** 1369.

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

Sjölander K., Karplus K., Brown M., Hughey R., Krogh A., Mian I.S., and Haussler D. 1996. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *Comput. Appl. Biosci.* **12:** 327.

Smith T.F. and Waterman M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol.* **147:** 195.

Socolich M., Lockless S.W., Russ W.P., Lee H., Gardner K.H., and Ranganathan R. 2005. Evolutionary information for specifying a protein fold. *Nature* **437:** 512.

Srinivasan G., James C.M., and Krzycki J.A. 2002. Pyrrolysine encoded by UAG in Archaea: Charging of a UAG-decoding specialized tRNA. *Science* **296:** 1459.

Steinberg S., Misch A., and Sprinzl M. 1993. Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res.* **21:** 3011.

Stone E.A., Cooper G.M., and Sidow A. 2005. Trade-offs in detecting evolutionarily constrained sequence by comparative genomics. *Annu. Rev. Genomics Hum. Genet.* **6:** 143.

Storz G., Altuvia S., and Wassarman K.M. 2005. An abundance of RNA regulators. *Annu. Rev. Biochem.* **74:** 199.

Szymanski M., Barciszewska M.Z., Zywicki M., and Barciszewski J. 2003. Noncoding RNA transcripts. *J. Appl. Genet.* **44:** 1.

Tucker B.J. and Breaker R.R. 2005. Riboswitches as versatile gene control elements. *Curr. Opin. Struct. Biol.* **15:** 342.

Wang J., Zhang J., Zheng H., Li J., Liu D., Li H., Samudrala R., Yu J., and Wong G.K. 2004. Mouse transcriptome: Neutral evolution of "non-coding" complementary DNAs. *Nature* **431:** 757.

Waterston R.H., Lindblad-Toh K., Birney E., Rogers J., Abril J.F., Agarwal P., Agarwala R., Ainscough R., Alexandersson M., An P., et al. (Mouse Genome Sequencing Consortium). 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature* **420:** 520.

Weinberg Z. and Ruzzo W.L. 2004. Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. *Bioinformatics* (suppl. 1) **20:** I334.

———. 2006. Sequence-based heuristics for faster annotation of non-coding RNA families. *Bioinformatics* **22:** 35.

Winkler W.C. 2005. Riboswitches and the role of noncoding RNAs in bacterial metabolic control. *Curr. Opin. Chem. Biol.* **9:** 594.

Zuker M. 2000. Calculating nucleic acid secondary structure. *Curr. Opin. Struct. Biol.* **10**: 303.