Prediction of Structural RNA Genes through Computational Phylogenetic Shadowing

Kushal Chakrabarti and Daniel L. Ong *

December 1, 2003

1 Introduction

The amount of biological information produced by the Human Genome Project, other genome sequencing projects, and subsequent analysis is absolutely staggering. Biological sequences in GenBank, the major database of the National Institutes of Health (NIH), alone constitutes more than 35GB – the entire database was over 188GB. Even so, these biological databases are not all inclusive – for instance, genome databases at the University of California, Santa Cruz (UCSC) have more than 128GB of genomic sequence and associated analysis.

The availability of such extraordinary amounts of information poses several problems for the biological community, but at the same times creates incredible opportunity for computer scientists. The availability of such information, perhaps most importantly, simply precludes the possibility of manual, experimental analysis – the former workhorse of biological science. Instead, computational techniques – everything from formal language design [4] to complexity theory [5] and machine learning [3] – have proven useful.

Here, we are concerned with the application of machine learning techniques to the prediction of an important class of genes, known as *structural RNA genes*, within complete genomic sequences. The products of these genes, the structural RNAs themselves, fold into stable secondary structures through pseudo-palindromic pattern of complementary nucleotides that bind to each other. For instance, one could observe the sequence `AUUCGGUAA ... UUUCCGAAU`; since the nucleotide adenine (A) complements uracil (U) and guanine (G) complements cytosine (C), one could observe the binding pattern

```
AUUCGGUAA ...
|||||| || ...
UAAGCCUUU
```

where the sequence has been "folded" to allow binding. In fact, the products of these genes are necessary under the famous *central dogma*, which claims that DNA is first used to produce messenger RNA (mRNA), that is then used in the synthesis of proteins. More recently, though, these genes have been implicated in the regulation of a wide variety of critical biochemical pathways [8]. What is intriguing, however, is the observation that these genes are inherently difficult to discover through experimentation. Successful computation prediction of such genes would allow biological researchers to focus on the systematic analysis of putative genes instead of hit-and-miss search that typifies experimental biology.

Nevertheless, no successful approach to the systematic prediction of such genes has been published. Although significant work [3, 7, 1] has produced reasonably successful predictions of protein-coding genes, the analogous literature for structural RNA gene prediction is, at best, sparse and, at worst, unsuccessful. This is somewhat understandable given that (1) the statistical signals exploited in protein-coding gene prediction approaches either do not exist or are substantially degraded in structural RNA genes, and (2) the signals that do exist require significantly more complex statistical models for sensible analysis. In fact, the standard implementation of these statistical models (stochastic context free grammars, or SCFGs) is intractable for the problem sizes that are commonly encountered in computational biology.

*The authors have been listed in alphabetical order.

2 Background

Traditional approaches in protein-coding gene prediction have exploited the characteristic genomic structure of such genes. It is, for instance, established that protein-coding genes in higher eukaryotes are encoded as a set of *exons* interspersed among *intronic* sequence. During the production of mature mRNA from genomic DNA (through the process of *transcription*), these introns are removed. Since proteins are ultimately synthesized from mature mRNA through *translation*, the sequence of exons of a gene characterize its protein. Traditional gene prediction approaches exploit, among other things, the observation that

1. specific signals, known as *splice sites*, disproportionately occur at intronic/exonic boundaries [1]
2. nucleotide triplets within exonic sequences are drawn from organism-specific distributions (*codon bias*),
3. the lengths of exons and introns follow well-known distributions (normal and geometric, respectively) [3], and
4. specific signals, e.g. certain nucleotide triplets, indicate the beginning and end of genes.

We will observe, without further elaboration, that these characteristics can be straightforwardly implemented as a Hidden Markov Model (HMM).

In addition, the computational biology community has also begun to exploit the observation that functionally important regions of the genome are disproportionately similar across related species. The basis of this approach, known as the *comparative genomics paradigm*, is essentially that

1. genomic regions that lack functional purpose are likely to accumulate mutations over time from occasional errors in DNA replication, whereas
2. genomic regions critical to the survival of a species are unlikely to accumulate such mutations to the extent that their accumulation would cause the region to lose its critical function.

For all practical purposes, the claim is that corresponding functionally important regions in related species will exhibit particular patterns of evolution, i.e. those patterns that do not induce loss of function. With respect to protein-coding genes, this pattern is one that maintains generally similar sequence in the final protein product. This roughly translates into maintaining particular nucleotides at corresponding positions of corresponding regions in different genomes. It is important to note that this approach has proven successful [1, 7, 2].

With structural RNA genes, however, the matter of strict sequence similarity is relatively inconsequential in light of maintaining the specific secondary structure induced by the base-pairing of different positions in the structural RNA gene. Here, roughly speaking, a particular position in the structural RNA gene can mutate as long as either (1) it was not base-paired to another position, or (2) the position with which it was base-paired also mutated to a nucleotide that can base-pair with it. To this end, it is expected that evolution in structural RNA genes will exhibit a pattern of *compensating mutation* – if a base-paired position in a structural RNA gene mutates, the position to which it was base-paired will mutate with high probability. Unfortunately, this pattern of evolution cannot be captured by HMMs – it can, however, be captured by stochastic context-free grammars [10].

In fact, it turns out the comparative genomics approach is critical to structural RNA gene prediction. Eddy *et al* showed that sufficient signal does not exist in genomic sequence from a single species to accurately predict structural RNA genes [9]. More recent work, however, has shown that two sequences provides a nominal amount of statistical significant signal for structural RNA gene prediction [10].

In order to model evolutionary processes, we must first *align* the input sequence of nucleotides. This process allows for insertions of the gap symbol, in order to model deletions in this sequence or insertions in other sequences. For example, consider the sequences:

```
ATGCAAAATTTTTTTT
ATCGAATTTTGTTTT
```

One possible alignment inserts two gaps in the second segment and one gap in the first:

```
ATGCAAAATTTT-TTTT
ATGCAA--TTTTGTTTT
```

The preferred alignment minimizes a metric similar to edit distance, with insertions of the gap character and mismatches possible. Unlike edit distance, however, the cost function of each operation is arbitrary.

Aligning two sequences can be extended to aligning n sequences. A *multiple alignment* is one that matches as many of the sequences as possible, again allowing insertions of the gap character or mismatches between sequences. As n increases, more information is known about the likelihood of biological significance of a subsequence.

Here, we propose a model that can be used to (1) predict structural RNA genes in whole genomes and (2) simultaneously produce alignments that are consistent with structural RNA evolution. This model extends previous work by (1) meaningfully scaling to whole genome analysis, (2) handling an arbitrary number of genomes, and (3) not being susceptible to poor input alignments.

3 Structural RNA Gene Prediction

Our model employs three types of graphical models: phylogenetic trees, generalized hidden Markov models, and stochastic context-free grammars. A stochastic context-free grammar (also known as a probabilistic context-free grammar) is a context-free grammar, that instead of expanding an arbitrary rule, chooses one with a particular probability. A generalized Hidden Markov Model is simply a Hidden Markov Model where each state can emit sequences of symbols instead of single symbols.

The model is conceptually comprised of four distinct submodels: a single global submodel and three local submodels. The global submodel describes the general structure and layout of the input genomes, in terms of the locations of protein-coding and structural rna-coding genes. The local models, on the other hand, describe the actual sequences of protein-coding genes, structural rna-coding, and noncoding regions.

Attending first to some necessary preliminaries, define the alphabet $\Sigma = \{A, C, G, T\}$ of DNA sequences and the alphabet $\Sigma_- = \Sigma \cup \{-\}$ of aligned DNA sequences. As a measure of technical convenience, let $\Sigma_-^{(N)} = \Sigma_- \times \Sigma_- \times ... \times \Sigma_-$ be the alphabet of multiple alignments on N sequences.[1]

The model takes as input a set of approximately aligned sequences $S = \langle S_1, S_2, ..., S_{|S|} \rangle$, where $S_i = S_{i,1} S_{i,2} ... S_{i,|S_i|} \in \Sigma_-^*$, and a binary phylogenetic tree T defining the homology between the sequences S_i. As output, the model

1. computes an alignment $Y = Y_1 Y_2, ... Y_{|Y|} \in [\Sigma_\epsilon^{(N)}]^*$, where $|Y| = |S|$, and
2. infers the (hidden) sequence of states $X = X_1 X_2 ... X_{|X|}$, where $|X| = |Y|$ and X_t indicates whether a position t in the multiple alignment is in a protein-coding, structural rna-coding, or noncoding region.

3.1 Phylogenetic Tree

A *phylogenetic tree* is a graphical model that considers each site independently (*site independent evolution*), meaning evolution at one point in the sequence does not affect another part. With this assumption, we use the tree to calculate the likelihood of the given sequences. The structure (also called topology in the literature) of the tree specifies the relationship between species. Intuitively, it serves as a "family tree" recording speciation. The aligned sequences being taken in as input constitutes the leaf nodes, since the sequences are observed. Internal (non-leaf) nodes represent hypothetical common ancestor species of the sample species. Each node of the tree represents a random variable that can take values corresponding to the four nucleotides or the gap character. Furthermore, one each edge connecting parent and child species, a branch length represents evolutionary time.

The site independent evolution model is not suitable for our purpose, since the nucleotide in one position will have a high probability of binding to a nucleotide in another part of the sequence. This invalidates the independence assumption, but it can be overcome by assuming sites are at most pairwise dependent (*pairwise sited dependent evolution*).

Without a phylogenetic tree, the parameter space is exponential, $O(|\Sigma_-|^n)$. Given the phylogenetic tree T, it becomes constant. To compute this, we use the following formula to find the probability of the kth nucleotide of all sequence s_i, for $i = 1$ to n. Let the internal nodes be $r_1 ... r_{n-1}$, where r_1 is the root. [6]

$$\Pr(s_{1,k} ... s_{n,k} | T) = \sum_{r_1 ... r_n} \Pr(r_1 = a) \prod_{i=1}^{2n-2} \Pr(a_i | a^{\alpha(i)}, i_i) \prod_{i=1}^{n} \Pr(s_{i,k} | a^{\alpha(i)}, t_i)$$

[1] Strictly speaking, the alphabet of multiple alignments on N sequences is $\Sigma_-^{(N)} = \Sigma_- \times \Sigma_- \times ... \times \Sigma_- \backslash \{(-, -, ..., -)\}$ because it is not sensible to incur the cost of a gap in every sequence.

where $\Pr(a_i|a^{\alpha(i)}, i_i)$ is the probability that nucleotide a evolves from its parent in time t_i and $\Pr(s_{i,k}|a^{\alpha(i)}, t_i)$ is the probability of the leaf observation in sequence i at position k evolves from its parent nucleotide in time t_i.

3.2 Genome Architecture Submodel

The genome architecture, ie. global, submodel is very simple. In particular, it is a *generalized Hidden Markov Model (GHMM)* $G = \langle Q^G, \pi^G, \delta^G, \tau^G \rangle$ of three states – one for each of the local submodels – with transitions between every two states.
In particular,

1. the state set $Q^G = \{q_P, q_S, q_N\}$ includes exactly one state for each local model; and,
2. the initial probability $\pi^G_{q_i} = Pr(X_1 = q_i)$ defines the probability that a prefix of the multiple alignment was generated by local submodel i; and,
3. the transition probability $\delta^G_{q_i,q_{i'}} = Pr(X_{t+l} = q_{i'}|X_t = q_i, l)$ defines the probability that the multiple alignment column at position $t + l$ was generated by local submodel i' given that local submodel i generated the multiple alignment columns $t, t+1, ..., t+l-1$; and, finally,
4. the length prior $\tau^G_{q_i}$ defines a probability distribution $Pr(l|q_i)$ that submodel i generates l multiple alignment columns.

In order to implement a feasible version of this model, we need to be able to efficiently compute the probability $Pr(S_i, S_{i+1}, ..., S_{i+l-1}|q_i, l)$ that a local submodel i emits a sequence of multiple alignment columns $S_i, S_{i+1}, ..., S_{i+l-1}$.

3.3 Protein-Coding Submodel

The protein-coding submodel is entirely derived from previous (successful) work in comparative genomic protein-coding region prediction. Specifically, we extend the SLAM GHMM to handle multiple alignments of an arbitrary number N of sequences.

The protein-coding local submodel GHMM $P = \langle Q^P, \pi^P, \delta^P, \tau^P \rangle$ is superficially similar to the global model described above. The most important difference between the two is that the state set Q includes different states corresponding to introns, single-exon genes, and exons of different phase shifts.

We further extend the SLAM GHMM to allow emission of multiple alignment of an arbitrary number of sequences. However, the model P is not explicitly modified; while the specific parameterizations may change, the ideas underlying the state set Q^P, transition matrix δ^P, initial distrbution π^P, and length priors τ^P remain constant. Instead, the (uncharacterized) (sub)submodels that describe exonic and intronic sequences are redefined over multiple alignments on N sequences. It is important to note there that the probabilities of emission in the GHMM states are computed according to the site-independent model of evolution.

For the purposes of computing $Pr(S_i, S_{i+1}, ..., S_{i+l-1}|q_P, l)$, we can implement the GHMM in the standard fashion and return this probability through the $\alpha-$ and $\gamma-$ algorithms.

3.4 Structural RNA-Coding Submodel

The local submodel describing structural RNA genes requires greater specificity than HMMs or similar variants. Since structural RNA genes are often characterized by long-range, nested base-pairing relationships between nucleotides, and HMMs can only (efficiently) describe local relationships, HMMs are insufficient. In contrast, *stochastic context free grammars (SCFGs)* have been shown to provide a theoretically coherent framework for the description of structural RNA genes.

Hence, we use an SCFG $S = \langle Q^S, R^S, \delta^S, \pi^S \rangle$ to define the local submodel describing structural RNA-coding genes. Specifically,

1. the set of *nonterminals* $Q^S = \{B, C\}$, which includes a nonterminal C that corresponds to long-range, nested base-paired nucleotides, and a nonterminal B for non-base-paired nucleotides; and,
2. the (implicit) set of *symbols* $Z = Q^S \cup \Sigma_\epsilon^{(N)}$; and,
3. the rewriting rules R^S, which define the sequences of symbols $\lambda \in R^S_q \subseteq Z*$ to which a nonterminal $q \in Q$ may be rewritten; and,

4. the initial probability π_q^S, which parameterizes the probability that nonterminal q characterizes a (possibly trivial) prefix and suffix of the sequence; and finally,

5. the transition probability $\delta_{q,\lambda}^S$ that a nonterminal $q \in Q^S$ will be replaced by a sequence of symbols $\lambda \in Z*$.

Furthermore, the structure of the SCFG may be qualitatively represented by the rewriting rules

$$R^S = \left\{ \begin{array}{lllllll} B & \rightarrow & aB & & C & \rightarrow & aCa' \\ & \rightarrow & Ba & & & \rightarrow & aB \\ & \rightarrow & aCa' & & & \rightarrow & Ba \\ & \rightarrow & BB & & & \rightarrow & a \end{array} \right\}$$

for all $a, a' \in \Sigma_\epsilon^{(N)}$.

The probability $\delta_{q,\lambda}^N$ that a nonterminal q rewrites to a sequence of symbols λ is crucial to (correct) operation. Roughly speaking, if a rewrite rule models base-paired nucleotides, eg. $C \rightarrow aCa'$, the probability of the rewrite is derived from the pairwise site-dependent model of evolution; conversely, if a rewrite-rule models non-base-paired nucleotides, eg. $B \rightarrow aB$, the probability of the rewrite is derived from the site-independent model of evolution. In particular, if q rewrites to λ and

1. if λ contains exactly two terminal symbols a and a', $Pr(q \rightarrow \lambda) \propto Pr_{PSD}(a, a'|T)$; or,

2. if λ contains exactly one terminal symbol a, $Pr(q \rightarrow \lambda) \propto Pr_{SI}(a|T)$,

where T is the previously mentioned phylogenetic tree. The necessary probability $Pr(S_i, S_{i+1}, ..., S_{i+l-1}|q_S, l)$ is exactly the probability computed by the inside algorithm.

3.5 Noncoding Submodel

The noncoding submodel is quite trivial. It is simply a single-state HMM that is defined over single columns of a multiple alignment on n sequences. We approximate the expected increased rate of evolution using a tree T' which is simply the tree T, with all branch lengths scaled by some factor greater than one. Although this approach is admittedly crude, it has been used with reasonable success elsewhere [2], and is a sufficient heuristic for our purposes. Probabilities of emission are then computed according to the tree T' using the site-independent model of evolution. The probability $Pr(S_i, S_{i+1}, ..., S_{i+l-1}|q_N, l)$ can be computed in a fashion that is similar to the protein-coding model.

4 Implementation

It is well-known that GHMMs can be implemented in time linear to the input sequence length and maximum length that can be output by a single state. On the other hand, the complexity of stochastic context-free parsing in the structural RNA submodel, implemented in the standard fashion, is cubic in the length of the input. Since the length of the input commonly ranges from 10^5 to 10^8, the standard implementations are clearly intractable.

The following discussion will accordingly focus on bounding and optimizing the worst-case time complexity of stochastic context-free parsing. In order to make the former argument concrete, we first, very briefly, describe the Inside algorithm and show that it imposes an intractable time complexity for the application of phylogenetic shadowing. We then describe and derive worst-case complexities for a series of improvements on the standard CYK implementation, and ultimately show that the structural RNA submodel can be parsed in time linear in the length of the alignment.

4.1 Stochastic Context-Free Parsing

4.1.1 Preliminaries

The Inside algorithm computes a three-dimensional matrix T of probabilities, where entry $T_{i,j,q}$ is the probability that nonterminal q was ultimately rewritten to the multiple alignment columns $i, i+1, ..., j$.

4.1.2 Algorithm

Although the Inside algorithm traditionally accepts only grammars in Chomsky normal form (CNF), it can be straightforwardly extended to handle the grammar specified for the structural RNA submodel.

```
INSIDE(S)
T = ALLOCATENEWMATRIX(|S|, |S|, |Q|);
for i = 1, ..., |S|
    do for j = 1, ..., |S|
            do for q → λ ∈ R
                    do if λ ~ aCa'
                            then T_{i,j,q} = T_{i,j,q} + Pr(q → λ) * T_{i+1,j-1,C} * Pr(a, a'|PSD)

                       if λ ~ aB
                            then T_{i,j,q} = T_{i,j,q} + Pr(q → λ) * T_{i+1,j,B} * Pr_SI(a)

                       if λ ~ Ba
                            then T_{i,j,q} = T_{i,j,q} + Pr(q → λ) * T_{i,j-1,B} * Pr_SI(a)

                       if λ ~ BB
                            then for k = i + 1, ..., j - 2
                                     do T_{i,j,q} = T_{i,j,q} + Pr(q → λ) * T_{i,k,B} * T_{k+1,j,C}
```

There are a couple matters worth mentioning here. First, the terminal symbols a and a' are, in fact, temporary placeholders for specific terminals from the alphabet $\Sigma^{(N)}{}_{-}$. As noted earlier, it is important to model the actual emission of these terminals using the phylogenetic tree. To incorporate this probability, we factor the probability $Pr(q \to \lambda^*)$ that q rewrites to a specific rule λ^*, i.e. with actual terminals from the alphabet $\Sigma_{-}^{(N)}$ to the product of

1. the marginal probability $Pr(q \to \lambda)$ over specific terminals that q rewrites to some rule of the form λ; and,
2. the conditional probability that specific terminals a (and possibly a') were emitted given a particular form λ.

Second, the fact that we have modified the Inside algorithm – instead of converting our grammar to CNF form – is indeed important and discussed in more detail below.

4.1.3 Time Complexity

The Inside algorithm clearly iterates over three indices that are, in the worst case, proportional to the length of the input alignment. Since these iterations are, in fact, nested amongst each other, the time complexity is cubic in the length of alignment, i.e. $O(T^3)$, where the length of the alignment is T. This, in light of the fact that alignment lengths can easily exceed 10^8, causes the standard implementation of the Inside algorithm to fail.

4.2 Genome Architecture Length Prior

Certain classes of length priors in the genome architecture model coupled with approximate alignments, allow reduction of the time complexity to $O(T * L^3)$, for some constant L.

Specifically, if $Pr(l|q_i) = 0$ for all $l > L$, then there is zero probability that the structural RNA submodel generated a sequence that is longer than L. Conversely, during parsing, the structural RNA submodel will never need to parse a subalignment with more than L sites. We can naively implement this by naively computing the Inside algorithm on windows of length L at each position $i = 1, 2, ..., T - L + 1$.

4.3 Subparse Forwarding

The aforementioned optimization, however, can be straightforwardly extended to further reduce the time complexity of the algorithm to $O(L^3 + T * L^2)$, for the same constant L.

In particular, one can incrementally compute the window beginning at i given the window beginning at $i-1$ in time L^2. Thus, the only full computation is that of the firt window at position 1 at L^3. This optimization is obvious given the observation that the window computed at position i differs from the window computed at position $i-1$ only at its last position, i.e. $i+L$. We essentially need to only compute the inside probabilities for all multiple alignment column sequences ending at position $i+L$. Since there are L such column sequences, and each require $O(L)$ time, the time necessary to incrementally compute windows is $O(L^2)$.

4.4 Rule Compression

The previous discussions have glossed over a subtle aspect: the size of the grammar itself is exponential in the number of sequences, given the terminal symbol variables a and a'.

The entire grammar need not be explicitly represented. In fact, it is not necessary to expand the terminal symbol variables until parsing. Hence, it is possible to simply store the *compressed rules* – rules that have not yet been substituted – of which there are a constant number. If a grammar stores only compressed rules, it is said to be a *compressed grammar.*

Outside of the obvious benefit of decreasing the size of the grammar, grammar compression allows us to enforce particular structure upon the rewriting probability distribution. It allows us, in particular, to enforce that nonterminals rewrite to rules that include them with higher probability. For instance, because base-paired nucleotides occur in runs of 5-10 at a time, we expect that if we are in the base-pair nonterminal C that we will rewrite to another base-pair nonterminal C, i.e. through some rule of the form $C \to aCa'$, with high probability – and, in the very least, with probability higher than to a rule of the form $C \to aB$. We can formalize this by letting each compressed rule i maintain a corresponding marginal probability p_i of being rewritten, and then computing the joint probability of a specific, uncompressed rule by multiplying in the probability of the specific column(s) of nucleotides emitted using the appropriate phylogenetic model.

5 Conclusion

We have implemented the GHMM, SCFG, and phylogenetic tree models in JAVA. We, however, were unable acquire the necessary biological sequence data in time to run meaningful experiments. In the last few days, we have been able to get into contact with the necessary collaborators and will be able to begin experimental evaluation of this model. Nonetheless, we believe that this report is interesting in and of itself as a novel (albeit unproven) theoretical framework for the simultaneous prediction and alignment of structural RNAs in genomic sequence and hope that the reader agrees.

References

[1] Marina Alexandersson, Lior Pachter, and Simon Cawley. Slam: Cross-species gene finding and alignment with a generalized pair hidden markov model. *Genome Research*, 2003.

[2] D. Boffelli, J. McAuliffe, D. Ovcharenko, K.D. Lewis, I. Ovcharenko, L.Pachter, and E.M. Rubin. Phylogenetic shadowing of primate sequences to find functional regions of the human genome. *Science*, 2003.

[3] Chris Burge. *Identificationof genes in human genomic DNA*. PhD thesis, Stanford University, 1997.

[4] Bor-Yuh Evan Chang and Manu Sridharan. Pml: Toward a high-level formal language for biological systems. Technical report, University of California, Berkeley, 2003.

[5] Pierluigi Crescenzi, Deborah Goldman, Christos H. Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 1998.

[6] Richard Durbin, Sean Eddy, Anders Krogh, and Graeme. Mitchison. *Biological Sequence Analysis.* Cambridge University Press, 1998.

[7] I. Korf, P. Flicek, D. Duan, and MR Brent. Integrating genomic homology into gene structure prediction. *Bioinformatics*, 2001.

[8] Baojie Li, Josep Vilardell, and Jonathan R. Warner. An rna structure involved in feedback regulation of splicing and of translation is critical for biological fitness. *Proceedings of the National Academy of Sciences*, 1996.

[9] Elena Rivas and Sean Eddy. Secondary structure alone is generally not statistically significant for the detection of noncoding rnas. *Bioinformatics*, 2000.

[10] Elena Rivas and Sean Eddy. Noncoding rna gene detection using comparative sequence analysis. *BMC Bioinformatics*, 2001.