Searching Genomes for Noncoding RNA Using FastR

Shaojie Zhang, Brian Haas, Eleazar Eskin, and Vineet Bafna

Abstract—The discovery of novel noncoding RNAs has been among the most exciting recent developments in biology. It has been hypothesized that there is, in fact, an *abundance* of functional noncoding RNAs (ncRNAs) with various catalytic and regulatory functions. However, the inherent signal for ncRNA is weaker than the signal for protein coding genes, making these harder to identify. We consider the following problem: Given an RNA sequence with a known secondary structure, efficiently detect all structural homologs in a genomic database by computing the sequence and structure similarity to the query. Our approach, based on structural filters that eliminate a large portion of the database while retaining the true homologs, allows us to search a typical bacterial genome in minutes on a standard PC. The results are two orders of magnitude better than the currently available software for the problem. We applied FastR to the discovery of novel riboswitches, which are a class of RNA domains found in the untranslated regions. They are of interest because they regulate metabolite synthesis by directly binding metabolites. We searched all available eubacterial and archaeal genomes for riboswitches from purine, lysine, thiamin, and riboflavin subfamilies. Our results point to a number of novel candidates for each of these subfamilies and include genomes that were not known to contain riboswitches.

Index Terms—Noncoding RNA, database search, filtration, riboswitch, bacterial genome.

1 Introduction

Not all genes encode proteins. Noncoding RNAs (ncRNAs) form transcripts that are functional molecules by themselves. The involvement of ncRNA in translation (tRNA), splicing, and other cellular functions is well-known. As early as 1961, Jacob and Monod hypothesized complementary roles for the two classes of genes, proposing that *"Structural genes encode proteins, and regulatory genes produce ncRNA"* [16]. Until recently, however, most novel gene discovery was in the form of protein coding genes and discovery of ncRNA was limited to finding novel homologs of commonly occurring ncRNAs (such as tRNA and rRNA). In part, these discoveries were fuelled by advances in genomic and computational technologies as well as large scale genome sequencing projects leading up to publications of large Eukaryotic genomes [21], [38], [41]. The complete genomic sequence allowed us to refine the estimates of the number of human (coding) genes. Surprisingly, these current estimates (30,000-40,000 genes) comprise less than 2 percent of the genome, far lower than earlier estimates and only twice as many as in *Drosophila*. It is an intriguing question if these genes and their (alternatively spliced) protein products are sufficient to carry out complex cellular functions. Could it be that many cellular functions are carried out by as yet undiscovered ncRNA?

Recent discoveries show that this idea of a treasure trove of undiscovered ncRNA is not without merit. The discovery of endogenous small interfering RNA (RNAi) has generated a lot of excitement [30]. Targeted search for other noncoding RNA (ncRNA) [2], [23], [25] has led to surprising discoveries of novel subfamilies of ncRNA. Some ncRNA are not independently transcribed but occur as part of the untranslated regions of mRNA. For example, Riboswitches are ncRNA elements that often occur in the 5′ untranslated regions (UTRs) and regulate the transcription of the downstream gene by directly binding to metabolites [29], [39]. It is hypothesized that there is in fact an *abundance* of undiscovered, functional ncRNAs with various catalytic and regulatory functions (the modern RNA world [9]). The reason these genes remain undiscovered is because genomic and computational tools for finding ncRNAs are not as advanced as those for protein coding genes.

Various computational approaches to detecting noncoding genes are under investigation. Some of these are attempts at *de novo* prediction, looking for signals that might suggest a functional RNA in the molecule. The most promising approach seemed to be the use of secondary structure as a signal [4], [15], [22] to discover RNA. This approach builds upon extensive earlier research into predicting the secondary structure of an RNA molecule [17], [46]. However, recent reports [31], [44] have concluded that the secondary structure signal is not sufficient to detect ncRNA. Random sequences with a biased GC composition, or with a *di-nucleotide* composition similar to true RNA sequences, usually allow folding into energetically favorable secondary structures. Other *de novo* approaches include looking for the transcription start and similar signals, but have had limited success. The consensus is that the ncRNA signals in a genome are not as strong as the signals for protein coding genes.

- *S. Zhang, E. Eskin, and V. Bafna are with the Department of Computer Science and Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0114. E-mail: {shzhang, eeskin, vbafna}@cs.ucsd.edu.*
- *B. Haas is with the Bioinformatics Program, Department of Bioengineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0419. E-mail: bchaas@ucsd.edu.*

Manuscript received 23 Nov. 2004; revised 2 Mar. 2005; accepted 12 July 2005; published online 1 Nov. 2005.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0194-1104.

Therefore, a natural way to solve this problem is based on comparative methods. One approach is to consider the evidence for RNA structure in sequences that are conserved through evolution. QRNA [32] tries to find ncRNA genes by scanning the conserved region alignments from two distant species. The program has been used to find ncRNAs in *E. coli* [33] and in *Saccharomyces cerevisiae* [28]. Other programs, such as ddbRNA [6], MSARI [5], and alignfold [40], use multiple alignments as input to detect conserved RNA secondary structures. However, if the sequences have diverged, constructing accurate multiple alignment itself is a challenging problem. Further selecting appropriate genomic subsequences to align is also challenging because of the divergence in primary sequences.

Here, instead of trying to identify novel ncRNA families, we address the relatively easier problem of identifying subsequences that are similar in structure and sequence to query. This approach has been used to find homologs of a specific RNA, such as tRNA [26]. This has also been extended to searching for homologs of ncRNA families by querying with a statistical representations of a multiple alignments of the family. Examples include CMsearch [12] using covariance model (CM) [10] and ERPIN using secondary structure profiles [11], [20]. Recently, Klein and Eddy developed a tool, RSEARCH [19], for searching a database with a single ncRNA query. This method depends upon existing algorithms for computing alignments between an RNA sequence and substrings of a database, where the alignment score is a function of sequence and structural similarity. Known algorithms for computing such alignments are computationally intensive, which is approximately $O(mw^2n)$, where m is the length of the query sequence, n is the length of the database sequence, and w is the maximum length of a database substring that is aligned to the query. For a test run on an Intel/Linux PC with 2.8 GHz, 1 Gb memory, a microbial database of size 1.67 Mb, and a query 5SrRNA sequence, RSEARCH took over 6.5 hours to run. This makes it intractable for a large genome database.

In this paper, we describe FastR, an efficient database search tool for ncRNA. FastR is generally two orders of magnitude faster and, as an example, FastR reduces the compute time of the previously mentioned query to 103 seconds. Are such algorithmic improvements worth investigating? An analogy can be made with the BLAST [1] algorithm, which has had tremendous influence on the growth of sequence databases such as Genbank and bioinformatics as a discipline. While tools for sequence alignment, based on the Smith-Waterman algorithm, had been available for a long time, BLAST changed the landscape largely by its speed and accuracy in searching for sequence homologs. The main idea here is the development of filters that efficiently prune most of the database while retaining the true homologs. This has also been tried for ncRNAs. For example, to improve speed, Rfam employs an initial BLAST search to filter genomic sequences before running the CMsearch [12]. Weinberg and Ruzzo [42] described filters based on Markov models, which can provably retain all hits that a covariance model could find. Because these two filters are based on primary sequences conservation, many compensatory mutations in ncRNA sequences that affect the sequence similarity may reduce their sensitivity or speed. Other approaches to filters are also studied (see, for example, [7]), which search for simple motifs which might be shared by many ncRNA families. Whereas, the idea in FastR is the use of RNA structural features as filters, where the filters are specific to a family. Most ncRNAs appear to be selected more for maintenance of a particular base-paired secondary structure than conservation of primary sequences. After filtering, we compute the alignments between the query ncRNA and all possible hits to find the true homologs.

We apply FastR to the discovery of novel riboswitches, which are a class of RNA domains found in the UTRs. They are of interest because they regulate metabolite synthesis by directly binding metabolites. We searched all available eubacterial and archaeal genomes (508 mega bases) for riboswitches from purine, lysine, thiamin, and riboflavin subfamilies. Our results point to a number of novel candidates for each of these subfamilies and include genomes that were not previously known to contain riboswitches. As an example, a search with the purine riboswitch (Z99107.2/14363-14264) took 19 hours on a standard PC and resulted in the discovery of 180 homologs, including 33 of 35 known riboswitches. Nine of these are of interest as they lie less than 500 bases upstream of a gene involved in Purine metabolism. Thus, FastR is a viable tool for discovering novel homologs of ncRNA.

We describe details of the FastR algorithm in Section 2. In Sections 3 and 4, the algorithm is validated by testing its speed and accuracy on known ncRNA subfamilies. Finally, we describe our findings from a search of the entire microbial database for novel riboswitches.

2 Methods

FastR solves following problem: Given an RNA sequence with known secondary structure, efficiently compute all structural homologs (computed as a function of both sequence and structural similarity) in a genomic database. There are two stages in FastR. In the first stage, the database is filtered to identify substrings which have structural features similar to the query (see Sections 2.1 and 2.2). In the second stage, the selected substrings are locally aligned to the query using a sequence structure alignment (see Section 2.5). Finally, p-values are assigned to the top hits.

2.1 Filters

Before introducing our structure-based filtering method, we first address the question whether sequence similarity with the query string is sufficient to get an initial set of candidate regions. To test this, we queried the whole genome of *A. pernix* (GenBank NC_000854.1) with an Asn-tRNA sequence. With default parameters, BLASTN selected four hits with an E-value < 0.001 and 24 hits with E-value < 10. Three of the four and 10 of the 24 matched the 43 hits produced by RSEARCH. Most of the alignments were less than 20bp in length and would have been discarded. Another example is presented in Fig. 1. The alignment of two tRNA sequences (Acc#: X07778.1/115-45 and AF200843.1/3014-3079) from *Drosophila* show complete conservation of structure, but low sequence similarity. From

```
GCAUCGGUGGUUCAGUGGUAGAAUGCUCGCCUGCCACGCGGGCG
<<<<<<<..<<<<.......>>>>.<<<<<.......>>>>>..
UCUAAUAUGGCAGAUU...AGUGCAAUAGAUUUAAGCUCUAUAU

GCCCGGGUUCGAUUCCCGACCGAUGCA
..<<<<<.......>>>>>>>>>>>.
AUAAAGU.AUUUU.ACUUUUAUUAGAA
```

Fig. 1. Alignment of two tRNA sequences from *Drosophila melanogaster* (tRNA_Gly, Acc#: X07778.1/115-45) (top) and *Drosophila simulans* (tRNA_Leu, Acc#: AF200843.1/3014-3079) (bottom). The two molecules have identical secondary structure (there are four stacks and two same-colored blocks form a stack), but very low sequence similarity (only four bases are matched in stacked region). Note that these are diverged members of a large superfamily. However, they underscore the need for structure-based alignments.

this and similar tests not included here, we do not anticipate a tool based on sequence similarity to be effective in finding RNA homologs. Therefore, we turn to the secondary structure of the query RNA sequence as the basis for our filter design. We will continue to use sequence similarity in computing the final alignments.

As shown in Fig. 2, the secondary structure of an RNA has a tree like shape and can be decomposed into various loops (Interior loops, bulges, multiloops) and stack regions. Each stem in this tree contains energetically favorable stacked base-pairs. The stacks are stabilized by hydrogen bonds between base-pairs. The Watson-Crick base-pairing (A $\leftrightarrow$ U, C $\leftrightarrow$ G) is energetically the most favorable, but other pairings such as the wobble base-pair (G $\leftrightarrow$ U) are possible as well. Fig. 2b provides a "stretched" view of the RNA structure.

Each stack corresponds to a pair of substrings. These pairs are typically noninterleaving. While interleaved stacks, or *pseudoknots* (such as the pair (f, f'), and (h, h') in Fig. 2), do occur, they can be ignored for filtering purposes.

Consider a nucleotide string s with $|s| = n$. We define a (k, w)-*stack* as a pair of indices $(i, j), i < j$ if $(j - i) \leq w$, $s[i \ldots i + k - 1]$, and $s[j \ldots j + k - 1]$ can form an energetically favorable base-pair stack. As an example, the indices of the substring (a, a') in Fig. 2 form a $(5, w)$-stack if they are at most w bases apart. A simple filter choice for an RNA structure is the set of all starting positions i which contain a (k, w)-stack for appropriately chosen k and w. Let p be the probability that a pair of randomly chosen bases is part of a stack. The probability that a pair of indices (i, j) with $(j - i) \leq w$ forms a (k, w)-stack is p^k. Define $X_{i,j}$ as the indicator variable with $X_{i,j} = 1$ if and only if (i, j) forms a (k, w)-stack. Using linearity of expectation, the expected number of hits in a random string of length n is

$$E\left(\sum_{i=1}^{n}\sum_{j=i+k}^{i+w} X_{i,j}\right) = \sum_{i=1}^{n}\sum_{j=i+k}^{i+w} E(X_{i,j}) \leq nwp^k.$$

See Table 1 for the expected number of hits per starting position ($\simeq wp^k$). Obviously, for large k and small w, even this simple filter can be quite powerful. Assume for exposition purposes that the base-pairing is limited to the Watson-Crick base (A $\leftrightarrow$ U, C $\leftrightarrow$ G) and the wobble base-pair (G $\leftrightarrow$ U). For a randomly and uniformly chosen pair of bases, the probability p of pairing is $p = \frac{3}{8}$. As an example, typical tRNA structures have a clover-leaf shape with the outermost stem having a seven base-pair stack separated by about 70 bases. The $(7, 70)$ filter would eliminate over 90 percent of the starting positions from consideration. In fact, we can do better as this base-pair unit is in fact separated by at least 50 bases in all tRNA, therefore making w effectively 20 ($50 \leq w \leq 70$), eliminating 98 percent of the starting positions. Note that the assumption that the bases are independent and identically distributed (i.i.d.) is not valid for real genomic sequence. However, the same principle applies and similar results are observed in practice.

2.2 Filter Design

We will use the (k, w)-stack as the basis for our filter design. However, we need to design more sophisticated filters as indels may sometimes disrupt base-pair stacks (decreasing the effective value of k), and variability in separation may increase the effective value of w. We quantify some design goals for filters to evaluate different designs, spur further

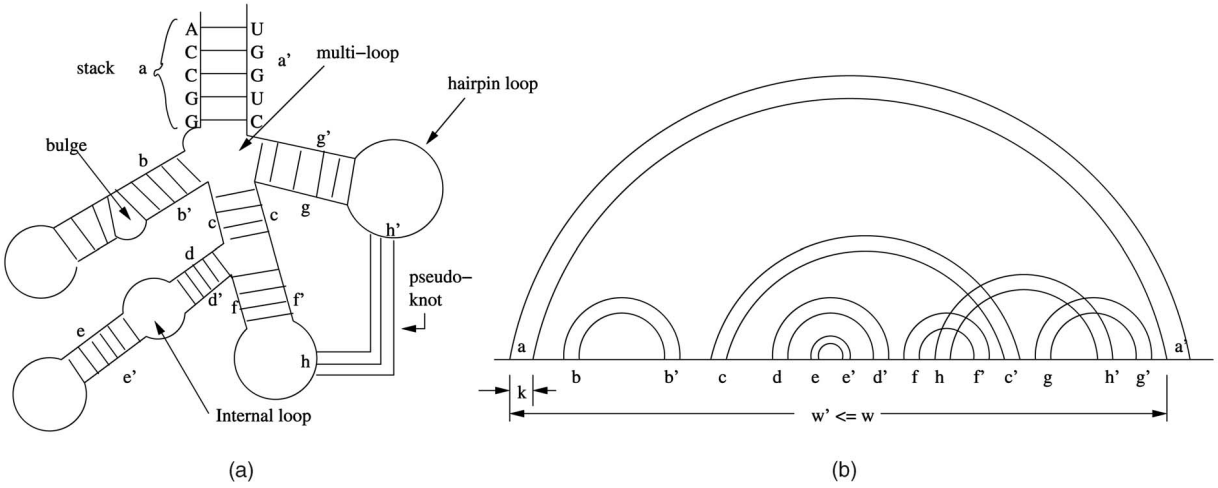


Fig. 2. (a) An RNA structure with various structural elements including stacked base-pairs, bulges, hairpin, and multiloops. (b) An alternative view. The set of bases in (a, a') forms a (k, w)-stack. Two substrings a and a' are w' bases apart where $w' <= w$.

TABLE 1
Expected Number of Hits in a Random String in a (k, w)-Filter

$w\backslash k$	4	5	6	7	8	9	10
20	0.3955	0.1483	0.0556	0.0208	0.0078	0.0029	0.001
40	0.791	0.2966	0.1112	0.0417	0.0156	0.0058	0.0021
60	1.1865	0.4449	0.1668	0.0625	0.0234	0.0087	0.0032
80	1.582	0.5932	0.2224	0.0834	0.0312	0.0117	0.0043
100	1.9775	0.7415	0.278	0.1042	0.0391	0.0146	0.0054

research in this area. A good filter must be *efficient*. The time to filter should be no more than the time to align and score the filtered hits and preferably as small as possible. Additionally, the filters must have high *sensitivity* and *specificity*. Sensitivity is described as the fraction of all members of the ncRNA family that is admitted by the filter, and should be as close to 1 as possible. It may be acceptable to work with lower sensitivities, for example, to look for members in a subfamily. We define specificity as the expected number of hits per base-pair and should be as small as possible. Finally, the filters must be general and simply described so as to be applicable (with appropriate parameter tuning) to every ncRNA family. We propose the following *Nested and Multiloop* filters:

Nested Filters: Considering the RNA secondary structure as a tree and going depth first down a path (see Fig. 2), we have many nested (k, w)-stacks. Consider (k, w)-stacks $s_1 = (i_1, j_1)$ and $s_2 = (i_2, j_2)$. Stack s_1 is *nested* in stack s_2 if $i_1 \leq i_2 + k$ and $j_2 \geq j_1 + k$. A (k, w, l)-*nested_stack* is a collection of l (k, w)-stacks $s_1, s_2, \ldots, s_l$ such that, for all $i \in [1, l-1]$, s_{i+1} is nested in s_i. For example, in Fig. 2, the configuration $(a, a'), (c, c'), (d, d'), (e, e')$ is a $(k, w, 4)$-nested_stack.

Parallel and Multiloop Stacks: Yet another way of looking at RNA structural elements is to locate nonnested, nonoverlapping (k, w)-stacks. Consider stacks $s_1 = (i_1, j_1)$ and $s_2 = (i_2, j_2)$. Stack s_1 is *parallel* to stack s_2 if $j_1 < i_2$ or $j_2 < i_1$. A (k, w, l)-*parallel_stack* is a set of stacks $s_1, s_2, \ldots, s_l$ such that any pair of stacks is parallel to each other. This definition can be extended to a multiloop_stack. A (k, w, l)-*multiloop_stack* is a configuration in which a $(k, w, l-1)$-parallel_stack and each of the stacks is nested in a (k, w)-stack. The units (b, b'), (d, d'), (f, f'), and (g, g') in Fig. 2 form a $(k, w, 4)$-parallel_stack. Correspondingly, $\{(a, a'), (b, b'), (d, d'), (f, f'), (g, g')\}$ is a $(k, w, 5)$-multiloop_stack.

The nested, parallel, and multiloop stacks are all generalizations of the (k, w)-stack and, therefore, applicable to all families of ncRNA. There are conserved structural elements in every ncRNA family that enforce the correct folding, so it should be possible to find multiloop and nested structures with high sensitivity. Also, the simple description allows us to compute specificity using combinatorial techniques. To increase the specificity of these filters, we need to extend the design to include distance constraints (number of base-pairs) in between the various (k, w)-stacks. For a filter with l (k, w)-stacks, there are $2l$ substrings of length k each with $2l - 1$ distances between adjacent substrings. To this, we add an additional distance between the first and the last substring and we have a vector of $2l$ distances. We constrain the distances by a $2l$-dimensional vector $\vec{w}$ containing the allowed ranges for each of these distances. Choose w_0 to be the range of distances between the first and last substring, and $w_j, j > 1$ to be the range of distances in the substrings ordered from left to right. A (multiloop/nested) filter satisfying these constraints is a $(k, \vec{w}, l)$-filter. Note that (k, w, l)-multiloop_stack can be redefined by choosing $\vec{w}$ such that $w_j = (0, w)$ for all j. A $(4, \vec{w}, 4)$-multiloop_stack for tRNA with appropriate distance constraints is shown in Fig. 3.

Specificity and Sensitivity: To compute the specificity of a (multiloop or nested) filter, we address the following combinatorial problem: What is the probability that an arbitrary position in the random database is the start of a $(k, \vec{w}, l)$-multiloop_stack or nested_stack? In general, this is hard to compute because of the various dependencies between overlapping units, so we approach it indirectly. Consider a $2l$-dimensional vector $\vec{v}$. If the distances in $\vec{v}$ are within the range specified by $\vec{w}$, then $\vec{v}$ denotes a *configuration* of a $(k, \vec{w}, l)$-multiloop_stack obtained by fixing the $2l$ positions of the l (k, w)-stacks using distances in $\vec{v}$. The probability of occurrence of an arbitrary configuration is exactly p^{kl}. For an arbitrary starting position and a configuration $\vec{v}$, define an indicator variable

$$X_{\vec{v}} = \begin{cases} 1 & \text{if a}(k, \vec{w}, l)\text{-multiloop\_stack occurs} \\ & \text{with configuration } \vec{v} \\ 0 & \text{otherwise.} \end{cases}$$

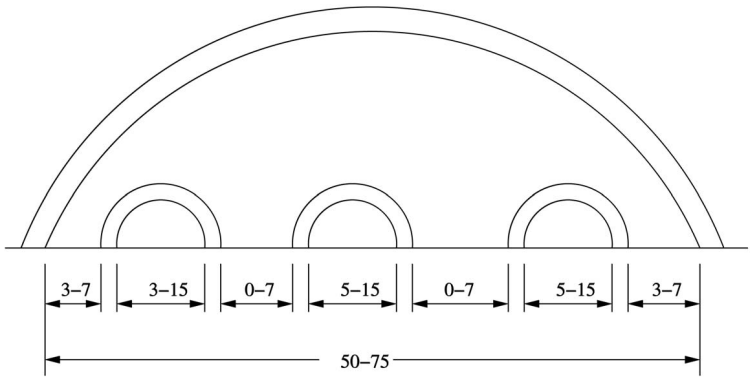


Fig. 3. A $(k, \vec{w}, 4)$-multiloop_stack for tRNA with distance constraints, with $\vec{w} = [(50, 75), (3, 7), (3, 15), (0, 7), (5, 15), (0, 7), (5, 15), (3, 7)]$.

Let $Y = \sum_{\vec{v}} X_{\vec{v}}$. We are interested in computing $Pr[Y > 0]$. By linearity of expectation, $E(Y) = \sum_{\vec{v}} E(X_{\vec{v}}) = n_{k,\vec{w},l} p^{kl}$, where $n_{k,\vec{w},l}$ is number of possible configurations of a $(k, \vec{w}, l)$-multiloop_stack. $n_{k,\vec{w},l}$ can be computed using standard combinatorial arguments. We consider two special cases:

1. Let $0 \leq w_j \leq w$ for all j. Then,
$$n_{k,\vec{w}e,l} = \binom{w - 2(k-1)l - 1}{2l - 1}.$$
2. Let $0 \leq w_0 \leq \infty$ and, for all $j > 0$, let $0 \leq w_j < x$. Then, $n_{k,\vec{w},l} = x^{2l-1}$.

Ideally, we choose the distance constraints so that $n_{k,\vec{w},l} p^{kl} << 1$. For those values, we can use the Markov inequality ($Pr[Y > 0] < E(Y)$) to get the desired bound. For higher values of $E(Y)$, we need other techniques to bound the probability. These computations allow us to quantify the sensitivity-specificity trade-off due to a change in the distance constraints. Further increases in specificity are obtained by using intersections of nested and multiloop_stacks. In Section 3, we describe our filtering results on various test cases. Informally, making a filter restrictive increases specificity at the cost of sensitivity. However, in most families of interest, we can design effective filters that reduce the database size by two orders of magnitude.

2.3 Optimal Filter Design

Given a family $\mathcal{R}$ of ncRNA sequences, an ideal (nested or multiloop) filter would seek to minimize $n_{k,\vec{w},l} p^{kl}$ (increase specificity) while admitting a large fraction of the members (sensitivity) and allow efficient filtering. Initial tests on the purine filters resulted in a ten-fold improvement in total running time with no loss of sensitivity. We will describe our results on optimal filter design elsewhere.

As the input to FastR is a single query ncRNA, we employ a dynamic programming algorithm that automatically generates nested and multiloop filters with high specificity. The algorithm takes advantage of the tree-like structure of RNA. It iterates over every value of k, l. For each such pair of values and every node v in the tree, it checks if a (k, l)-nested (multiloop) filter is possible. The final filter chosen is one that maximizes kl, while keeping k as low as possible. The software then allows users to tweak the computed parameters to get the desired sensitivity while retaining specificity. However, our results test results show that the automatically generated filters have sensitivity that is comparable to the fine tuned filters.

2.4 Filtering Algorithms

Filtering speed is critical to fast homolog computation. We use a combination of string matching and dynamic programming techniques (see, for example, [13]) to filter databases with multiloop and nested filters.

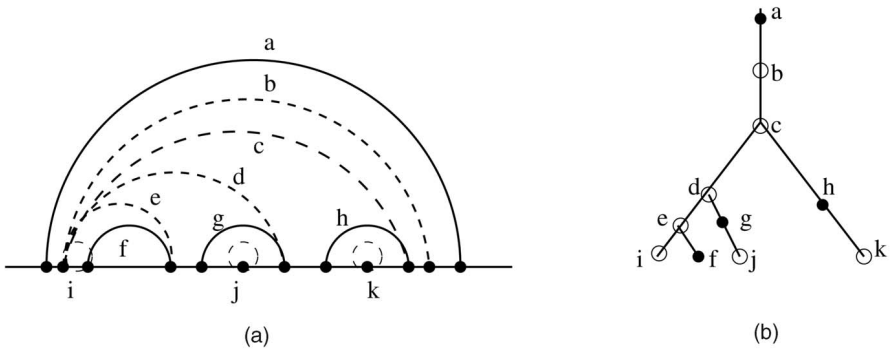
1. **Hash:** Build a hash table to compute all $k$mer positions in the database. The time taken is $O(m)$, where m is the size of the database.
2. **Identify (k, w)-stacks**: Let s_i denote the $k$mer at an arbitrary position i in the database. For each s_i, compute a neighborhood $N(s_i)$ of all "complementary" $k$mers. To identify (k, w)-stacks efficiently, we use the hash table to compute all positions j such that $s_j \in N(s_i)$, and $j - i$ satisfies distance constraints. The time taken is linear in the number of (k, w)-stacks, which is typically smaller than the size of the database.
3. **Filters:** Note that multiloop and nested filters are combinations of (k, w)-stacks. We scan the database with a moving window of size w. An "active" list of (k, w)-stacks within the window is maintained and a dynamic programming technique is used to compute filters from this list. The total computation is bounded by $O(m_k w)$, where m_k is the number of (k, w)-stacks. Typically, $m_k < \frac{m}{w}$.

In general, any $k$mer that can form an energetically favorable stack with s should be in $N(s_k)$. In our current implementation, we do not allow indels and allow at most two G $\leftrightarrow$ U pairs. To test this, a scan of all of the structures in Rfam 5.0 [12] showed that at least 93 percent of all stacks contain an ungapped base-pairing of at size at least four. Note that even the absence of an ungapped stack does not preclude the formation of a filter using other stacks in the same molecule. Therefore, this is a reasonable choice that does not affect sensitivity too much. The current filter time is a few seconds per Mb of sequence, which is easily dominated by the time for computing alignments. Also, the filters are very effective in eliminating a large fraction of the database while retaining most of the true hits.

2.5 Computing RNA Sequence Structure Alignment

After filtering, we need to align the filtered regions to the query. There are three types of alignments for RNA sequences: 1) RNA plain sequence alignment, which takes into account the secondary structures in the sequences [27], [36], 2) RNA structure structure alignment, which aligns tree-like secondary structures together [14], [45], and 3) RNA sequence structure alignment, which aligns a plain sequence to a secondary structure or a structure profile [3], [8], [24]. In this paper, we are dealing with the third type of alignment: The filtered database substrings must be structurally aligned to the query to identify true homologs. This problem has been well-studied in the literature, with scoring based on a Nussinov like counting model [3], [18], [36] and probabilistic models such as Covariance Models and Stochastic Context free grammars for RNA [8], [35]. It is also possible to extend the Zuker-Turner thermodynamic model [17], [46] for scoring sequence structure alignments.

Here, we extend the approach from Bafna et al. [3] to include a new binarizing procedure, banded alignment for efficient computation, and more realistic score functions. We use the scoring matrix (RIBOSUM) from Klein and Eddy [19] and empirically generated affine gap penalties to score the alignments. We note that our filtering approach generates candidates which can be used in conjunction with *any* alignment method. However, we use the extra information from the filter match to speed up alignment computation using banding techniques.




```
procedure Binarize(i,j)  (* Binarize the interval (i, j). *)
if (i = j)
    return (create_node(i,j,dotted,Nil)); (* A dotted node with 0 child. *)
if (i, j) ∈ S
    v = Binarize(i+1,j-1);
    return (create_node(i,j,solid,v)); (* A solid node with 1 child v. *)
if (k, j) ∈ S for some i < k < j
    vl = Binarize(1,k-1);
    vr = Binarize(k,j);
    return (create_node(i,j,dotted,vl,vr)); (*A dotted node with 2 children, vl and vr. *)
if (i < j)
    v = Binarize(i,j-1);
    return (create_node(i,j,dotted,v)); (* A dotted node with 1 child v. *)
end if
```

(c)

Fig. 4. Procedure to create a Binary tree for s with structure S, having $O(m)$ nodes such that each node has at most two children. (a) Nodes in the horizontal line represent sequence s. a, f, g, and h are paired bases. i, j, k, and b are unpaired bases. c, d, and e are representing the branches. The solid edges correspond to base-pairs in S, while the dotted edges correspond to augmented spurious edges. (b) A binary tree representation for (a), by changing solid edges into solid nodes and dotted edges into void nodes. (c) The Binarize procedure.

Given two RNA sequences, $s[1,\ldots,m]$ and $t[1,\ldots,n]$. We know the secondary structure of s, which is a set of base-pairs, S, where $(i,j) \in S$ implies that $s[i]$ bonds with $s[j]$. The alignment A of two RNA strings s and t can be described by a matrix of two rows. The first row $A[1,*]$ contains the string s with gaps, and the second row $A[2,*]$ contains the string t with some interspersed gaps. Each column has at most one gap in it. If $A[1,i]$ and $A[1,j]$ form a base-pair, we score for both sequence and structure using the function $\delta(A[1,i], A[1,j], A[2,i], A[2,j])$. As long as $A[2,i]$ and $A[2,j]$ also form a base-pair, we will give a high score to capture complementary mutations. Additionally, we score each column that does not participate in base-pairing by a function $\gamma(A[1,i]A[2,i])$ that measures sequence conservation. Alignments are scored by summing up the contributions of sequence and structural alignments.

A naive algorithm would iterate over all pairs of intervals in s and t. We can do better by exploiting the structure of s. Ignoring pseudoknots, each base-pair has a unique enclosing base-pair; thus, S can be shown to be a tree with each node denoting a base-pair, and the obvious parent-child relation. First, we augment the tree (see algorithm and an illustration in Fig. 4) by adding spurious base-pairs so that each nucleotide (originally base-paired or not) is in some base-pair, each node has at most two children, and the number of nodes is $O(m)$, where $|s| = m$. For any unpaired base, there should be a spurious edge added between this base and the most left base without crossing real base-pairing edges. Additionally, each node $v \in S$ has at most one child in the augmented structure which is denoted by S'.

A schematic algorithm for aligning an RNA query against a sequence is given in Fig. 5. Note that this algorithm uses linear gap penalties. In our implementation, we use a slightly more sophisticated affine gap function (omitted in Fig. 5 for exposition). Our alignment is local in the subject sequence (there is no penalty for aligning ends of the sequence), but global in the query sequence (the entire query must be aligned).

We limit the intervals in s to nodes $v \in S'$, which are bounded by $O(m)$. Fig. 5 describes the algorithm for aligning sequence t against sequence s (with known structure). Each node v in the tree structure of s is aligned against each interval (i,j) of t. Suppose $v \in S$ and let l_v and r_v denote the indices of the left and right end-points of v. If, for example, $s[l_v] = t[i]$ and $s[r_v] = t[j]$, then clearly

$$A[i,j,v] = A[i+1, j-1, child(v)] + \delta(t[i], t[j], s[l_v], s[r_v]).$$

procedure alignRNA
(*S is the set of base-pairs in RNA structure of s. S' is the augmented set. *)
for all intervals (i,j), $1 \le i < j \le n$, all nodes $v \in S'$
  if $v \in S$

$$A[i,j,v] = \max \begin{cases} A[i+1, j-1, \text{child}(v)] + \delta(t[i], t[j], s[l_v], s[r_v]), \\ A[i, j-1, v] + \gamma('-', t[j]), \\ A[i+1, j, v] + \gamma('-', t[i]), \\ A[i+1, j, \text{child}[v]] + \gamma(s[l_v], t[i]) + \gamma(s[r_v], '-'), \\ A[i, j-1, \text{child}[v]] + \gamma(s[l_v], '-') + \gamma(s[r_v], t[j]), \\ A[i, j, \text{child}[v]] + \gamma(s[l_v], '-') + \gamma(s[r_v], '-'), \end{cases}$$

  else if $v \in S' - S$, and v has one child

$$A[i,j,v] = \max \begin{cases} A[i, j-1, \text{child}[v]] + \gamma(s[r_v], t[j]), \\ A[i, j, \text{child}[v]] + \gamma(s[r_v], '-'), \\ A[i, j-1, v] + \gamma('-', t[j]), \\ A[i+1, j, v] + \gamma('-', t[i]), \end{cases}$$

  else if $v \in S' - S$, and v has two children
    $A[i,j,v] = \max_{i \le k \le j} \{A[i, k-1, \text{left\_child}[v]] + A[k, j, \text{right\_child}[v]]\}$
  end if
end for

Fig. 5. An algorithm for aligning a query RNA s of length m with a database string t of length n. The query structure S has been *Binarized* to get S'. The index pair in s corresponding to each node $v \in S'$ is denoted by (l_v, r_v).

If, on the other hand, $v \in S' - S$ and has two children, then we need to iterate over all k such that the right_child(v) can align with the interval (k, j). The procedure alignRNA (Fig. 5) describes the dynamic programming algorithm to handle all the cases. Let m_1 and $m_2 = m - m_1$ denote the number of nodes with one and two children, respectively. The complexity of alignRNA, with a query of length m and a target of length n, is $O(n^2 m_1 + n^3 m_2)$. This parameterization is useful because in typical structures $m_2 << m$. In our case, the complexity can be further reduced. The sequence pairs that need to be aligned have been filtered for an underlying substructure. The preliminary alignment obtained by this filter allows us to limit the nodes in S' that can be applied to a position i in t, based on the left endpoint of v and the width. This *banding* reduces the number of nodes to a constant, effectively making the complexity $O(n^2 \delta_m^2)$, where $\delta_m << m$ is the size of the banded region. The banding forces a trade-off. Overlapping hits from the filter can either be aligned independently with a tight band or merged and aligned once with larger band size.

2.6 P-Value

For an effective database search, we need to have p-values for the probability that a hit was obtained by chance. Klein and Eddy make the argument that the distribution of scores of RNA structural alignments follow the Gumbel distribution. As this is a strong assumption, and determination of a true p-value is a challenging research problem. Therefore, we choose to express the p-value by using the nonparametric Chebyshev's inequality. To obtain the mean and variance, the query is aligned against randomly generated sequence with a similar GC-content as the database after each query. The bound provided by this inequality is conservative and overestimates the probability of obtaining a similar score by chance. We have found that a cut-off of 0.03 is a reasonable value in practice.

3 Results

We describe the results on filtering and alignment independently before giving combined results. To test our algorithms, we worked with ncRNA subfamilies of known/predicted structure from the Rfam [12] and the 5S Ribosomal RNA database [37]. Four subfamilies are considered here, tRNA, 5S rRNA, the hammerhead ribozyme, and four riboswitches, purine, lysine, thiamin, and riboflavin [39], [43]. Of these, tRNA and rRNA are well-studied subfamilies. Most genome annotations include screening and annotation for tRNA. The different riboswitches are of great interest because they regulate metabolite (nucleic-acids, amino-acids, vitamins) synthesis by direct binding to metabolites. In subsequent tests, we search the entire complement of eubacterial and archaeal genomes for novel riboswitches.

For every subfamily, we chose representative members, inserted them in a random database of size 1Mb, and tested our algorithms on the composite sequence. The probability of finding stacks at random depends on the GC-content, so, in some cases, the random database was created by first choosing the GC-content and subsequently generating bases with appropriate fixed probability. $G + C$ probabilities of 0.35, 0.5, and 0.75 were chosen to study the effect of GC-content. All experiments were performed on an Intel PC (3.4 GHz, 1 Gb RAM), running Linux.

3.1 Filtering for ncRNA

Table 2 describes results of applying various filters. As expected, as the filters become more stringent (higher k, l, less variable distances), the number of false negatives increases. However, for each family, there exist appropriate filters that filter out a large portion of the database while retaining most of the members of the family. Also, as the GC-content is biased away from 0.5, the number of false hits increases. The false negatives are all explained by one of three possibilities: 1) The proposed structure contains

TABLE 2
The Results of Applying Nested and Multiloop Filters (with Various Parameters) to Random Databases that Contain True Positives

ncRNA	GC	k	l	#Hits (/Mb)	True Pos. /Tot.	Non canonical pairing	Missing (k,w)-stack	Deviant $\vec{w}$
tRNA	0.50	4	3	21120	89/100	10	1	0
tRNA	0.35	4	3	29379	89/100	10	1	0
tRNA	0.7	4	3	37208	89/100	10	1	0
5S rRNA	0.7	5	2	7502	80/100	0	20	0
5S rRNA	0.5	5	2	3307	80/100	0	20	0
Hammerhead	0.5	4(*)	2	6250	50/57	1	0	6
Purine-Rs	0.5	4	2	10263	33/35	0	2	0
Thiamin-Rs	0.5	4(*)	2	10822	84/115	0	18	13
Lysine-Rs	0.5	5	4	2749	28/32	0	4	0
Riboflavin-Rs	0.5	3(*)	0	558	38/41	0	2	1

As the filters become more stringent, the number of hits decrease and the number of false-negatives increase. In all but (*) cases, at most two G ↔ U base-pairs are allowed in a stack. (*) refers to cases where only one G ↔ U base-pair is allowed by the multiloop filter. The false negatives are due to noncanonical base-pairing, small stacks ($k = 3$), or the distance constraints being out of range, as described in the last three columns.

noncanonical base-pairs, which are not allowed by FastR. For example, $10/100$ tRNA sequences contain A ↔ G, A ↔ A, or A ↔ C base-pairs. 2) One of the (k,w)-stacks is missing due to indels, mismatches or short stacks. 3) Distance constraints are not satisfied. In ongoing work, we plan to change the neighborhood computation to include all $k$mers that form low energy stacks according to the Zuker-Turner [17] thermodynamic considerations. With respect to varying distance constraints, one has to choose the correct speed/sensitivity trade-off. The filters we have selected are all fast and lead to very few hits in the random database. This number will increase as the distance constraints are increased.

3.2 Alignment

To test alignment quality, we computed alignments on a randomly generated 300Kb database sequence with set of ncRNA sequence for each family inserted in it. No filtering was used for FastR. Fig. 6 shows ROC plots for the two alignment algorithms. The two are comparable, with RSEARCH performance better for distant homologs. The time taken for FastR (banded) tRNA alignment is 3 minutes and 48 seconds, compared to 20 minutes and 42 seconds for RSEARCH.

Finally, we evaluate FastR after combining filtering and alignment. We randomly select the query for each family from Rfam seed alignment and search the random sequences using FastR and RSEARCH. Table 3 summarizes the results of our search. Similar results are achieved when repeating the tests with different queries. As can be seen, FastR is close to two orders of magnitude faster than RSEARCH while maintaining comparable sensitivity. Much of the loss of sensitivity is due to filtering. As seen in the previous section, FastR alignments and scores are good for the high quality hits, but decrease thereafter leading to a loss of sensitivity. As expected, much of the loss of sensitivity can be attributed to filtering. For 5S rRNA, the filter allows 80 of the 100 true positives, which are almost completely retrieved by FastR. In contrast, RSEARCH gets the top 97 but needs two orders of magnitude more time, making it much harder to conduct large scale searches. It should also be pointed out that many of the true positives were initially discovered using covariance models which are not unlike the model used by RSEARCH. As a final validation of the FastR algorithm, we apply it to the discovery of novel members of Riboswitches. Our results point to a number of interesting findings.

3.3 Riboswitches

Riboswitches are cis-regulatory elements typically found in the 5′ untranslated region of the gene they regulate. To date, six such motifs have been identified that control the anabolism of three vitamins (riboflavin, thiamin, cobalamin), as well as the biosynthesis of methionine, lysine, and purine [29], [34], [39], [43]. Similar to previously characterized RNA regulatory structures, each riboswitch is capable of folding into a consensus structure which may result in either transcription attenuation or translation inhibition. However, the riboswitch element is unique in that it binds directly to ligands and is therefore able to sense the level of cellular metabolites without the need of transacting protein factors.

It is believed that this class of ncRNA appeared early in evolution and, accordingly, riboswitch elements have been found in a wide range of bacterial species. The vitamin riboswitches are the most diverse and can be identified in archaea and eubacteria. In particular, the thiamin riboswitch has been characterized in fungi and plants such as rice and *Arabidopsis* [39]. Conversely, the methionine, lysine, and purine riboswitches are more common in gram-positive bacteria. The repression mechanism is also biased by a bacterium's phylogeny. Gram positive bacteria typically prefer transcription termination, whereas gram-negative microorganisms tend to mediate gene repression by inhibiting translation.

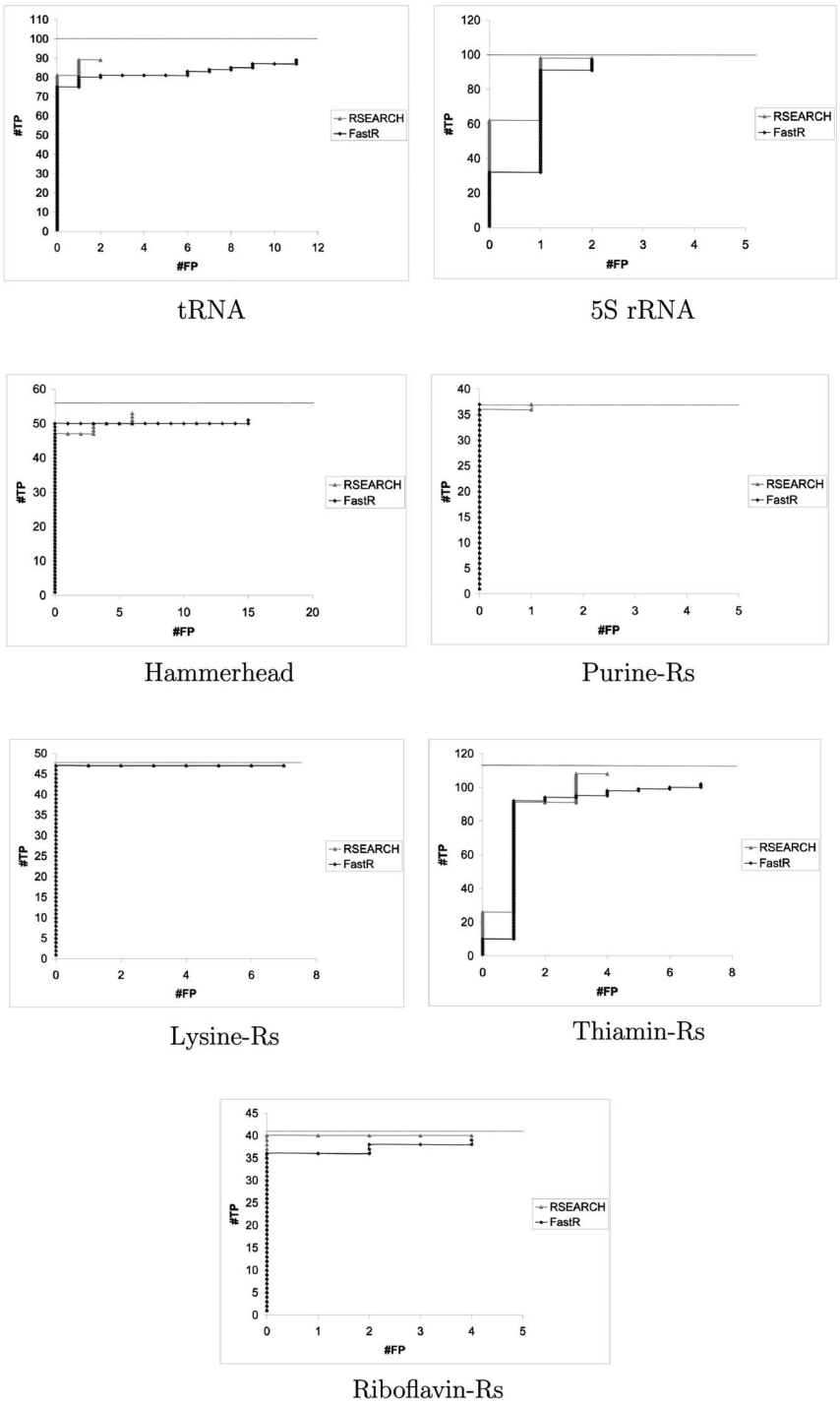


Fig. 6. ROC plots for the alignments generated by RSEARCH and FastR. Alignments were tested using a 300 kb random sequence with a set of true ncRNAs inserted in it. The X-axis represents the number of false-positives and the y-axis represents the number of true positives. The horizontal line represents the number of true hits in the random sequences.

While riboswitches are ubiquitous, homologs show little sequence similarity. Even in the most conserved regions, typically for ligand binding, the sequence identity may be less than seven nucleotides. We used FastR to search both plus and minus strands of bacterial and archaeal genomes with queries from purine, thiamin, lysine, and riboflavin riboswitches. A data set of nonredundant, known riboswitches existing within our genome files was assembled from the Rfam database. This data set was used to determine the p-value cutoffs and a single member used as the query sequence. A total of 245 genomes comprising 508 Mb were searched in both strands. Candidate riboswitch sequences generated by FastR were filtered in order to find the best predictions. First, known riboswitches from the Rfam database and low-complexity and AT-rich predictions were discarded. The remaining predictions were filtered by their distance from the 5′ start of an exon. Finally, the predictions were manually examined to

TABLE 3
Comparison of FastR and RSEARCH

	Query	Hits (TP/Tot)	Filtered Hits	Time
RSEARCH	Asn-tRNA (AE001087.1/4936-5008)	85/93	100	3411s
FastR		77/93	82	52s
RSEARCH	5S rRNA (AE016770.1/210436-210555)	97/97	100	14939s
FastR		80/80	80	44s
RSEARCH	Hammerhead (M83545.1/56-3)	50/58	50	2741s
FastR		47/47	47	34s
RSEARCH	Purine-Rs (Z99107.2/14363-14264)	34/35	35	5461s
FastR		33/33	33	77s
RSEARCH	Lysine-Rs (Z75208.1/54883-55062)	32/39	32	26581s
FastR		28/28	28	159s
RSEARCH	Thiamin-Rs (Z99110.2/31833-31942)	109/116	115	7850s
FastR		71/81	84	234s
RSEARCH	Riboflavin-Rs (L09228.1/7992-8136)	41/45	41	14385s
FastR		31/31	38	79s

A p-value cutoff for FastR, 0.05, was chosen that approximately matched the total number of hits in RSEARCH with cutoff E-value of 10. The hits column refers to the number of true positives out of the total hits found. The filtered hits column represents the number of true positives passed the filters. No filtering is used for RSEARCH.

determine if the downstream gene was biologically relevant. The results are summarized in Table 4 (which can be found on the Computer Society Digital Library at http://computer.org/tcbb/archives.htm).

We focus on the 18 most promising hits, even though the remaining hits are likely to contain many interesting candidates. See Table 5. These predictions represent either those elements which are upstream from genes involved in the metabolic pathway under regulation or predictions with strong sequence similarity in regions thought to mediate ligand binding. There are six of the nine putative purine riboswitches that are found in the 5′ UTR of either the xanthine transport protein, xanthine phosphoribosyltransferase, purine nucleotide phosphorylase, adenine deaminase, or GMP synthase. Moreover, prediction 2 (gi|42519879) lies upstream from a hypothetical protein with homology to the xanthine permease family. This observation highlights a hidden value in identification of riboswitches—the possibility of assigning annotations to genes of unknown function. Similarly, of the seven reported lysine riboswitch predictions, there are five predictions that are located upstream of genes encoding an amino acid permease, diaminopimelate decarboxylase, dihydrodipicolinate synthase, or lysine specific permease. The final predictions for the riboflavin and thiamin riboswitches are found upstream of genes encoding diaminohydroxyphosphoribosylaminopyrimidine deaminase and phosphomethylpyrimidine kinase, respectively.

Of the 16 novel purine and lysine riboswitch predictions, there are 13 predictions from gram-positive bacteria, supporting earlier conclusions. There are four of the seven novel purine hits that are to *Lactobacillus johnsonii* and *Lactobacillus plantarum*, which have no previously identified purine riboswitches. Likewise, four lysine predictions and one riboflavin prediction are from genomes with no previous riboswitches from that family. While none of these predictions appears in Rfam, it has been brought to our attention that some of these predictions overlap with the predictions in [34]. As these were made using completely different techniques, they provide additional validation of our approach.

Free energy minimization approaches to secondary structure prediction are not well suited to riboswitches because the repressing structure is contingent upon ligand binding. FastR offers an advantage for such RNA motifs in that the biologically significant structure can be inferred from the alignment. The secondary structures derived from the top predictions in each riboswitch family in Table 5 can be seen in Fig. 7.

4 Discussion

Our results show that FastR is an effective tool for finding novel homologs of query ncRNA sequences. In general, the development of fast filtering and searching tools for ncRNA is a natural area of research, analogous to the development of sequence similarity tools like BLAST and Fasta. However, as the discussion above shows, the underlying structure and diversity of ncRNA makes this problem quite different in character. Consequently, the filters must be more complex than the (approximate) keyword matches used for sequence similarity. The ideas presented here open many lines of research, which we are actively pursuing.

TABLE 4
Summary of the FastR Riboswitch Search

Riboswitch	Query	Time hrs.(Secs./Mb)	P_value cutoff[a]	Novel hits[b]	Filtered hits[c]
Purine	Z99107.2/14363-14264	19.12(67.7)	0.03	3350	180
Lysine	Z75208.1/54883-55062	45.82(162.23)	0.03	2190	200
Thiamin	Z99110.2/31833-31942	42.00(148.71)	0.04	1592	85
Riboflavin	L09228.1/7992-8136	22.86(80.94)	0.03	10	3

(a) The p-value cutoffs were determined from alignments of known riboswitches. (b) Number of novel hits returned by FastR after removing the annotated hits in Rfam database. (c) Number of novel hits after removing annotated hits in Rfam database and filtering for low-complexity, AT-content, and distance from a gene.

The first is regarding sensitivity. For diverged families, the filters miss out a few true homologs. Our analysis showed that, in many cases, this was due to a stem loop not being recognized as a (k, w)-stack. This can be due to too few base-pairs, bulges, and noncanonical base-pairing. However, the stem must still have low-energy that allows it to maintain that conformation. Therefore, we plan to generalize the definition of a (k, w)-stack allowing all pairs that form energetically favorable structures. While non-canonical base-pairs are easy to handle, bulges and interior loops are computationally more challenging. It will be interesting to see how these changes affect sensitivity. Some homologs are filtered out because they do not satisfy distance constraints. Relaxing the distance constraints could decrease specificity. One approach to increasing sensitivity without compromising specificity is to relax the distance constraints, but employ multiple nested and multiloop filters. While keyword matches are not good filters, Weinberg and Ruzzo [42] show that filters based on Markov Models can be very effective. These capture conservation in

TABLE 5
Description of the 18 Most Promising Candidates from the 468 Putative Riboswitches Discovered by FastR

Riboswitch	Genome	Location[a]	p-value	D[b]	Gene Annotation
Purine	*Bacillus anthracis*	794079-794178(+)	0.016	264	GMP synthase
	*Lactobacillus johnsonii**	1949385-1949485(+)	0.018	181	Hypothetical protein (xanthine permease family)
	*Lactobacillus plantarum**	2410480-2410573(+)	0.019	156	Xanthine / uracil transport protein
	*Lactobacillus plantarum**	339446-339540(−)	0.020	175	Adenine deaminase
	*Lactobacillus johnsonii**	1729531-1729628(−)	0.021	168	Xanthine phosphoribosyltransferase
	Bacillus anthracis	4574871-4574970(+)	0.021	319	Conserved hypothetical protein
	Clostridium perfringens	512985-513085(+)	0.024	417	Purine nucleoside phosphorylase
	Bdellovibrio bacteriovorus	1778017-1778113(−)	0.026	93	Hypothetical protein
	Bacillus cereus	2435592-2435693(−)	0.026	50	Adenine deaminase
Lysine	*Bacillus subtilis*	794406-794582(−)	0.010	282	ABC transporter (amino acid permease)
	Bacillus halodurans	1619231-1619417(+)	0.011	296	Diaminopimelate decarboxylase
	*Fusobacterium nucleatum**	1813295-1813475(−)	0.015	277	Hypothetical protein
	*Onion yellows phytoplasma**	191443-191627(−)	0.022	200	ABC-type amino acid transport system
	*Lactococcus lactis**	2276234-2276412(+)	0.026	284	Lysine specific permease
	*Lactococcus lactis**	699673-699852(−)	0.026	273	Dihydrodipicolinate synthase
	Shewanella oneidensis	1689148-1689335(−)	0.027	276	Hypothetical Na+/H+ antiporte
Riboflavin	*Thermus thermophilus**	1210336-1210467(−)	0.016	123	Diaminohydroxyphosphoribosy-laminopyrimidine deaminase family
Thiamin	*Streptococcus pneumonia*	1469400-1469498(−)	0.029	181	Phosphomethylpyrimidine kinase

Each predicted riboswitch is either upstream from a biologically relevant gene, or contains strong sequence similarity in regions thought to mediate ligand binding. (a) Genome coordinates of the riboswitches. "+" and "−" refer to the strand. (b) Distance between the start of the riboswitch and the 5' end of an exon. (*) Genomes with no previously identified riboswitches from that family.

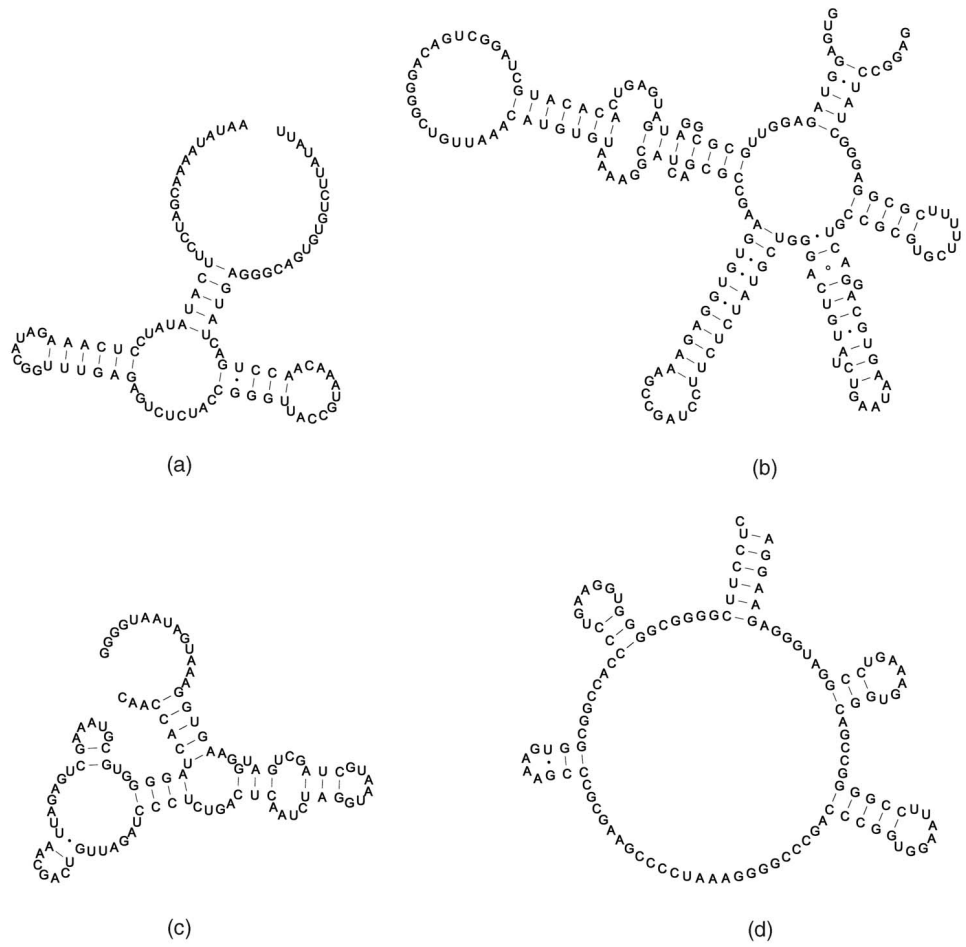


Fig. 7. Representative riboswitch secondary structures derived from the alignments of the top novel hits for each query. (a) The secondary structure prediction for the top purine hit. (b) The secondary structure prediction for the top lysine hit. (c) The secondary structure prediction for the top thiamine hit. (d) The secondary structure prediction for the top riboflavin hit.

sequence, but not structure. However, they are constructed from covariance models in a way that ensures the same level of sensitivity as the CM. It will be interesting to combine the two filters to see how well they perform.

Another direction is the design of optimal multiloop and nested filters. Currently, the filters were constructed by changing parameters empirically. We are working to automate the design of optimal (multiloop and nested) filters for an ncRNA family. Preliminary results on the purine riboswitch show a 10-fold speedup with no loss of sensitivity and we are testing the methodology on other families. These filters are constructed when the secondary structure is known for all members of the ncRNA family. In general, the secondary structure might not be known and it would be interesting to automate the filter design after inferring common structural elements. This is a simpler and more tractable version of the well-studied problem of RNA multiple alignment because it only requires an alignment of the stacks involved in the filters.

In many cases, the FastR results themselves need to be filtered to remove obvious false positives. However, the advantage of using the tool is that good candidates can be found with relatively little effort. If the "Modern RNA world" hypothesis is true, many ncRNA sequences will be discovered in the coming years. Our tool can be used to rapidly identify novel homologs of these ncRNA. Finally, many RNA motifs, including riboswitches, fold into the correct structure only in combination with other molecules. Programs that predict structure based on *de novo* energy minimization are challenged in their ability to find the correct structure for these molecules. In contrast, comparative tools such as ours can be used to infer structure relatively easily.

Similar to most bioinformatics tools, the results of the search must be validated in the lab for final confirmation. We plan to collaborate with experimental scientists in systematically testing the results on various families of interest. Our software is freely available for use upon request.

REFERENCES

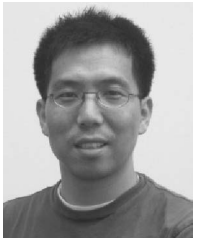
[1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman, "Basic Local Alignment Search Tool," *J. Molecular Biology,* vol. 215, pp. 403-410, 1990.

[2] L. Argaman et al., "Novel Small RNA-Encoding Genes in the Ontergenic Regions of *Escherischia Coli*," *Current Biology,* vol. 11, pp. 941-950, 2001.

[3] V. Bafna, S. Muthukrishnan, and R. Ravi, "Computing Similarity between RNA Strings," *Combinatorial Pattern Matching Conf.,* vol. 937, pp. 1-14, 1995.

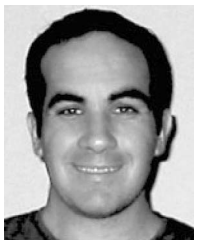
[4] J.-H. Chen, S.-Y. Lee, and B. Shapiro, "A Computational Procedure for Assessing the Significance of RNA Secondary Structure," *Computer Applications in the Biosciences,* vol. 6, pp. 7-18, 1990.

[5] A. Coventry, D.J. Kleitman, and B. Berger, "MSARI: Multiple Sequence Alignments for Statistical Detection of RNA Secondary Structure," *Proc. Nat'l Academy of Sciences,* vol. 101, no. 33, pp. 12102-12107, 2004.
[6] D. di Bernardo, T. Down, and T. Hubbard, "ddbRNA: Detection of Conserved Secondary Structures in Multiple Alignments," *Bioinformatics,* vol. 19, no. 13, pp. 1606-1611, 2003.
[7] M. Dsouza, N. Larsen, and R. Overbeek, "Searching for Patterns in Genomic Data," *Trends in Genetics,* vol. 13, no. 12, pp. 497-498, 1997.
[8] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, "Covariance Models: SCFG-Based RNA Profiles," *Biological Sequence Analysis,* chapter 10.3, Cambridge Univ. Press, 1998.
[9] S.R. Eddy, "Non-Coding RNA Genes and the Modern RNA World," *Nature Rev. in Genetics,* vol. 2, pp. 919-929, 2001.
[10] S.R. Eddy and R. Durbin, "RNA Sequence Analysis Using Covariance Models," *Nucleic Acids Research,* vol. 22, pp. 2079-2088, 1994.
[11] D. Gautheret and A. Lambert, "Direct RNA Motif Definition and Identification from Multiple Sequence Alignments Using Secondary Structure Profiles," *J. Molecular Biology,* vol. 313, no. 5, pp. 1003-1011, 2001.
[12] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S.R. Eddy, "Rfam: An RNA Family Database," *Nucleic Acids Research,* vol. 31, no. 1, pp. 439-441, 2003.
[13] D. Gusfield, *Algorithms on Strings, Trees, and Sequences.* Cambridge Univ. Press, 1997.
[14] M. Höchsmann, T. Töller, R. Giegerich, and S. Kurtz, "Local Similarity in RNA Secondary Structures," *Proc. Second IEEE CS Bioinformatics Conf. (CSB 2003),* pp. 159-168, 2003.
[15] I.L. Hofacker, B. Priwitzer, and P.F. Stadler, "Prediction of Locally Stable RNA Secondary Structures for Genome-Wide Surveys," *Bioinformatics,* vol. 20, no. 2, pp. 186-190, 2004.
[16] F. Jacob and J. Monod, "Genetic Regulatory Mechanisms in the Synthesis of Proteins," *J. Molecular Biology,* vol. 3, pp. 318-356, 1961.
[17] J. Jaeger, D.H. Turner, and M. Zuker, "Improved Prediction of Secondary Structures for RNA," *Proc. Nat'l Academy of Sciences,* vol. 86, pp. 7706-7710, 1989.
[18] T. Jiang, G. Lin, B. Ma, and K. Zhang, "A General Edit Distance between RNA Structures," *J. Computational Biology,* vol. 9, pp. 371-388, 2002.
[19] R.J. Klein and S.R. Eddy, "Rsearch: Finding Homologs of Single Structured RNA Sequences," *BMC Bioinformatics,* vol. 4, no. 1, p. 44, 2003.
[20] A. Lambert et al., "The ERPIN Server: An Interface to Profile-Based RNA Motif Identification," *Nucleic Acids Research,* vol. 32, no. s2, pp. W160-165, 2004.
[21] E. Lander et al., "Initial Sequencing and Analysis of the Human Genome," *Nature,* vol. 409, pp. 860-921, 2001.
[22] S.Y. Le, J.H. Chen, and J. Maizel, *Structure and Methods: Human Genome Initiative and DNA Recombination,* vol. 1, pp. 127-136. Adenine Press, 1990.
[23] R.C. Lee and V. Ambros, "An Extensive Class of Small RNAs in *Caenorhabditis elegans*," *Science,* vol. 294, pp. 862-864, 2001.
[24] H.P. Lenhof, K. Reinert, and M. Vingron, "A Polyhedral Approach to RNA Sequence Structure Alignment," *J. Computational Biology,* vol. 5, no. 3, pp. 517-530, 1998.
[25] L.P. Lim, N.C. Lau, E.G. Weinstein, A. Abdelhakim, S. Yekta, M.W. Rhoades, C.B. Burge, and D.P. Bartel, "The MicroRNAs of *Caenorhabditis elegans*," *Genes and Developtment,* vol. 17, pp. 991-1008, 2003.
[26] T.R. Lowe and S.R. Eddy, "tRNAscan-SE: A Program for Improved Detection of Transfer RNA Genes in Genomic Sequence," *Nucleic Acids Research,* vol. 25, pp. 955-964, 1997.
[27] D.H. Mathews and D.H. Turner, "Dynalign: An Algorithm for Finding the Secondary Structure Common to Two RNA Sequences," *J. Molecular Biology,* vol. 317, no. 2, pp. 191-203, 2002.
[28] J.P. McCutcheon and S.R. Eddy, "Computational Identification of Non-Coding RNAs in Saccharomyces cerevisiae by Comparative Genomics," *Nucleic Acids Research,* vol. 31, no. 14, pp. 4119-4128, 2003.
[29] A. Nahvi, N. Sudarshan, M.S. Ebert, X. Zou, K.L. Brown, and R.R. Breaker, "Genetic Control by a Metabolite Binding mRNA," *Chemical Biology,* vol. 9, pp. 1043-1049, 2003.
[30] C.D. Novina and P.A. Sharp, "The RNAi Revolution," *Nature,* vol. 430, no. 6996, pp. 161-164, 2004.
[31] E. Rivas and S.R. Eddy, "Secondary Structure Alone Is Generally Not Statistically Significant for the Detection of Noncoding RNAs," *Bioinformatics,* vol. 16, no. 7, pp. 583-605, 2000.
[32] E. Rivas and S.R. Eddy, "Noncoding RNA Gene Detection Using Comparative Sequence Analysis," *BMC Bioinformatics,* vol. 2, pp. 8-26, 2001.
[33] E. Rivas, R.J. Klein, T.A. Jones, and S.R. Eddy, "Computational Identification of Noncoding RNAs in *E. coli* by Comparative Genomics," *Current Biology,* vol. 11, pp. 1369-1373, 2001.
[34] D.A. Rodinov, A.G. Vitreschak, A.A. Mironov, and M.S. Gelfand, "Regulation of Lysine Biosynthesis and Transport Genes in Bacteria: Yet Another RNA Riboswitch?" *Nucleic Acids Research,* vol. 31, no. 23, pp. 6748-6757, 2003.
[35] Y. Sakakibara, M. Brown, R. Hughey, I.S. Mian, K. Sjölander, R.C. Underwood, and D. Haussler, "Recent Methods for RNA Modeling Using Stochastic Context Free Grammars," *Proc. Combinatorial Pattern Matching Conf.,* vol. 807, 1994.
[36] D. Sankoff, "Simulations Solution of the RNA Folding, Alignment and Protosequence Problems," *SIAM J. Applied Math.,* vol. 45, no. 5, pp. 810-825, 1985.
[37] M. Szymanski, M.Z. Barciszewska, V.A. Erdmann, and J. Barciszewski, "5S Ribosomal RNA Database," *Nucleic Acids Research,* vol. 28, no. 1, pp. 166-167, 2002.
[38] J.C. Venter et al. "The Sequence of the Human Genome," *Science,* vol. 291, no. 5507, pp. 1304-1351, 2001.
[39] A.G. Vitreschak et al. "Riboswitches: The Oldest Mechanism for the Regulation of Gene Expression?" *Trends in Genetics,* vol. 20, no. 1, pp. 44-50, 2003.
[40] S. Washietl and I.L. Hofacker, "Consensus Folding of Aligned Sequences as a New Measure for the Detection of Functional RNAs by Comparative Genomics," *J. Molecular Biology,* vol. 342, no. 1, pp. 19-30, 2004.
[41] R.H. Waterson et al. "Initial Sequencing and Comparative Analysis of the Mouse Genome," *Nature,* vol. 420, no. 6915, pp. 520-562, 2002.
[42] Z. Weinberg and W.L. Ruzzo, "Faster Genome Annotation of Non-Coding RNA Families Without Loss of Accuracy," *Proc. Int'l Conf. Research in Computational Molecular Biology,* pp. 243-251, ACM Press, 2004.
[43] W.C. Winkler and R.R. Breaker, "Genetic Control by Metabolite-Binding Riboswitches," *Chembiochem,* vol. 4, no. 10, pp. 1024-1032, 2003.
[44] C. Workman and A. Krogh, "No Evidence that mRNA have Lower Folding Free Energy than Random Sequences with the same Dinucleotide Distribution," *Nucleic Acids Research,* vol. 27, no. 24, pp. 4816-4822, 1999.
[45] K. Zhang, L. Wang, and B. Ma, "Computing Similarity between RNA Structures," *Combinatorial Pattern Matching,* pp. 281-293, 1999.
[46] M. Zuker and D. Sankoff, "RNA Secondary Structures and their Prediction," *Bull. Math. Biology,* vol. 46, pp. 591-621, 1984.



Shaojie Zhang received the Bachelor of Science degree in computer science from Peking University, Beijing, China, and the Master of Engineering degree from Nanyang Technological University, Singapore. He is currently a PhD candidate in computer science at the University of California, San Diego. His research is focused on bioinformatics, which includes ncRNA gene finding, RNA analysis, and comparative genomics.

Brian Haas received the Bachelor of Science degree in biochemistry and computer science from the University of Wisconsin, Madison, in 2003. He is currently enrolled in the bioinformatics PhD program at the University of California, San Diego. His research interests include comparative genomics, RNA evolution, and disease association.



Eleazar Eskin received the PhD degree in computer science from Columbia University in 2002. He is currently an assistant professor in the computer science and engineering department at the University of California, San Diego. Previously, he was a postdoctoral researcher at the Hebrew University of Jerusalem in computer science with professors Yoram Singer and Nir Friedman.

Vineet Bafna is an assistant professor in computer science and engineering at the University of California, San Diego. Prior to joining UCSD in 2003, he worked at Celera Genomics, ultimately, as Director of Informatics Research. He also held positions at SmithKline Beecham and at The Center for Advancement of Genomics. His research is focused on bioinformatics, including population genetics, ncRNA gene-finding, and computational proteomics.