BIOINFORMATICS ORIGINAL PAPER
Vol. 22 no. 1 2006, pages 35–39
doi:10.1093/bioinformatics/bti743

Sequence analysis

Sequence-based heuristics for faster annotation of non-coding RNA families

Zasha Weinberg[1,*] and Walter L. Ruzzo[1,2]

[1]Department of Computer Science & Engineering and [2]Department of Genome Sciences, University of Washington, Seattle, WA 98195, USA

Received on December 13, 2004; revised on October 13, 2005; accepted on October 22, 2005
Advance Access publication November 2, 2005
Associate Editor: Steven L. Salzberg

ABSTRACT

Motivation: Non-coding RNAs (ncRNAs) are functional RNA molecules that do not code for proteins. Covariance Models (CMs) are a useful statistical tool to find new members of an ncRNA gene family in a large genome database, using both sequence and, importantly, RNA secondary structure information. Unfortunately, CM searches are extremely slow. Previously, we created rigorous filters, which provably sacrifice none of a CM's accuracy, while making searches significantly faster for virtually all ncRNA families. However, these rigorous filters make searches slower than heuristics could be.

Results: In this paper we introduce profile HMM-based heuristic filters. We show that their accuracy is usually superior to heuristics based on BLAST. Moreover, we compared our heuristics with those used in tRNAscan-SE, whose heuristics incorporate a significant amount of work specific to tRNAs, where our heuristics are generic to any ncRNA. Performance was roughly comparable, so we expect that our heuristics provide a high-quality solution that—unlike family-specific solutions—can scale to hundreds of ncRNA families.

Availability: The source code is available under GNU Public License at the supplementary web site.

Contact: zasha@cs.washington.edu

Supplementary information: http://bio.cs.washington.edu/supplements/zasha-HeurHmm-2004/ (Technical details, results, C++ code)

1 INTRODUCTION

Non-coding RNAs (ncRNAs) are RNAs that function without being translated to proteins. ncRNAs include *trans*-acting RNAs, e.g. tRNAs and microRNAs (Wagner and Flardh, 2002), and *cis* regulatory elements, such as riboswitches (Winkler and Breaker, 2003). Since roughly the late 1990s, research has shown ncRNAs to be much more numerous and significant than previously thought. For reviews, see Kennedy, 2002; Storz, 2002; Eddy, 2002.

This paper addresses a fundamental task for RNA research: given a family of related RNAs, scan genomes for homologs. Several techniques exist for this task, but this paper focuses on covariance models (CMs). A discussion of such techniques and our motivation for improving CMs appears in our previous paper (Weinberg and Ruzzo, 2004b). The present paper proposes a novel technique to speed CM searches. For brevity, we assume basic familiarity with our earlier paper.

Prior work has sped CM scans with sequence filters, running the CM only on promising subsequences. However, these filters have limitations; we see a need for filters that have the following features:

(1) Generic—work for any ncRNA family
(2) Sensitive and fast
(3) Able to trade sensitivity for speed, e.g. if computer resources are limited
(4) Easy to adapt to improvements on CMs—better CM technology should not invalidate the filters.

This paper describes a novel heuristic filter called the ML-heuristic ('Maximum-Likelihood heuristic'), which meets all the above criteria, although sensitivity and speed could (as always) be improved further. In terms of sensitivity and speed, it is comparable with (though not as good as) highly tuned heuristics in the tRNAscan-SE program (Lowe and Eddy, 1997). This suggests that generic heuristic filters are a reasonable approach to design filters for the hundreds of families in Rfam (Griffiths-Jones *et al*., 2005), one that—unlike the creation of specialized tools like tRNAscan-SE—requires no human effort.

In the ML-heuristic, a profile HMM is created, in which HMM transition and emission probabilities are set to make the HMM maximally similar to the CM.

This paper introduces a novel methodology to analyze filters on real biological data, comparing the various heuristic methods with each other in terms of speed and sensitivity. Realistic heuristic filters must make a trade-off between missing true positives versus submitting too much to the CM (thus not accelerating searches).

Previous heuristics include tRNAscan-SE and a BLAST-based (Altschul *et al*., 1997) heuristic used by Rfam (Griffiths-Jones *et al*., 2005). We show that BLAST filtering has problems with sensitivity.

tRNAscan-SE (Lowe and Eddy, 1997) is applied to annotate tRNAs in most genome projects and uses CMs. tRNAscan-SE runs two programs previously created specifically for tRNA searches; if either program reports a possible tRNA, the CM is run. Since they use tRNA-specific programs, tRNAscan-SE's heuristics are not generic.

Previously, we created rigorous filters (Weinberg and Ruzzo, 2004a,b), which provably guarantee that all homologs detectable by a given CM are selected by the filter. Although rigorous filters'

*To whom correspondence should be addressed.

guarantees can be advantageous, they fail on some of our criteria. First, they are slower than heuristics can be, because rigorous guarantees are required.

Moreover, rigorous filters cannot trade sensitivity for increased speed, since they must guarantee maximal sensitivity. For example, our rigorous scans of 1.7 Gb of bacterial sequences for glycine riboswitch homologs (Mandal *et al*., 2004), took 9.7 CPU days, while our heuristics took 1.8 days, yet missed only 1 putative homolog (out of 609). Our rigorous filtering techniques could not run faster without sacrificing rigorousness (although we do explore using the underlying filters as heuristics).

Next, rigorous filters failed to improve scanning time for two families, while maintaining guarantees. Although two families (out of 139 tested) is a modest problem, it would be helpful to search these families in a practical amount of time. Moreover, as ncRNA research progresses, we anticipate that more known families will become bigger, and their secondary structure thus more prominent. So, more highly structured families may emerge that challenge our rigorous filters, which are best at exploiting primary sequence conservation.

Finally, the proofs used in rigorous filters may not translate to improvements on CMs. (In contrast, although improved CMs may make the probabilistic analysis used in the ML-heuristic difficult, it should always be possible to create a non-improved CM, and use the resulting ML-heuristic to filter searches with the improved CM. This solution is not ideal, but yields a valid heuristic, and should have reasonable sensitivity for its given ncRNA. Indeed, the existing BLAST-heuristic entirely ignores the CM.)

Our results show that the ML-heuristic discriminates better than BLAST. Although BLAST itself runs faster than the profile HMM scan (∼2–4 times faster), profile HMMs seem a better choice when high levels of sensitivity are demanded for challenging families, since overall scan time will be dominated by the CM. For a common example, to achieve maximal sensitivity for the cobalamin riboswitch (Rfam ID RF00174), BLAST requires the CM to scan 7% of RFAMSEQ (Rfam's 8 Gb sequence database), where our heuristics only require 0.001%. Our approach is about 10 times faster overall. (The HMM itself is roughly 600 times faster than the CM.) In an extreme case, BLAST missed 90% of SECIS element homologs found with a profile HMM filter in the same run time.

We also compare the ML-heuristic with tRNAscan-SE's heuristics. Despite the work that went into tRNAscan-SE—particularly the dedicated tRNA detectors used in its heuristics—the discriminative power of the ML-heuristic is similar to that of tRNAscan-SE's heuristics. A weakness of the ML-heuristic is that it is a factor of 3–12 times slower than tRNAscan-SE's heuristics. However, its speed seems close enough to be immediately practical and we are optimistic that further improvements are possible.

In summary, this paper introduces the novel ML-heuristic, designed to make profile HMMs as close as possible to the CM. In contrast, our previous algorithms set scores to facilitate rigorous filtering. We also consider filters in the context of heuristics, rather than the rigorous context of our previous papers. Heuristics enable increased speed in exchange for slightly reduced sensitivity—an important practical advantage in many scenarios. This paper presents a method to evaluate heuristics and shows the ML-heuristic is the best available generic method, and is even comparable with tRNAscan-SE's specialized heuristics.

2 METHODS

2.1 Simplified Covariance Models

This technical section assumes familiarity with Sections 3, 4.1 and 4.2 of our earlier paper (Weinberg and Ruzzo, 2004b). However, for ease of exposition, we will use the probabilistic form of CMs (not odds ratios or logarithmic scores).

Our earlier paper did not explain how CMs are created. CM rules are created from a multiple sequence alignment (MSA) with annotations indicating which columns are base paired (Eddy and Durbin, 1994). For base-paired columns, $S_i \rightarrow x_L S_{i+1} x_R$ rules are created. For unpaired columns, $x_R = \epsilon$.

We assign probabilities for the rules based on the MSA. Each CM rule's probability is set based on how frequently it is used in parsing the MSA sequences constrained by the annotated secondary structure. A maximum-likelihood estimate derives a probability estimate by counting the number of times the rule is used and dividing by the counts of all rules with the same CM state in their left-hand side (Durbin *et al*., 1998). To avoid zero-probability rules, however, pseudocounts are typically used; all rules have 1 count added before considering the MSA.

2.2 The ML-heuristic profile HMM

2.2.1 Profile HMM grammar The profile HMM grammar is created from the CM as in Weinberg and Ruzzo (2004b).

This grammar transformation is equivalent to a transformation of MSAs. Given the MSA (with secondary structure) used to create a CM, a profile HMM can be created by removing the secondary structure (i.e. base pair annotations), and using the CM creation method. With no structure, this 'CM' will be equivalent to a profile HMM; the profile HMM will capture the primary sequence information of the original MSA, but not its secondary structure.

2.2.2 Profile HMM probabilities In our previous work, we assigned scores in order to ensure rigorous filtering. In this paper, we discard rigorousness in favor of a heuristic. Intuitively, we wish to assign probabilities to the profile HMM so that it is as similar as possible to the CM, to make the heuristic more accurate. (Although our rigorous and heuristic HMM grammars are the same, the method of assigning rule probabilities is completely different. The supplement shows a simple example where the heuristic has improved filtering.)

In Section 2.2.1 we suggested training a profile HMM on the MSA used to create the CM. Unfortunately, issues like pseudocounts affect the profile HMM differently from the CM, so the resulting profile HMM is not as similar as it could be. To avoid this problem, we train the HMM directly from the CM. (Results show empirically that training directly from the CM is more accurate; see supplement.) Suppose we generate a random MSA from the CM. If this MSA has many sequences, it accurately reflects the CM's probability distribution. We can now learn a profile HMM from this MSA without pseudocounting, i.e. using maximum likelihood. The larger the MSA, the closer the profile HMM's distribution is to that of the CM, at least in the positional sequence information that the profile HMM can model. In fact, it is possible to simulate an MSA with infinitely many sequences, i.e. the limiting case.

The correspondence between CM and profile HMM in the MSA is the same as the correspondence between rules. Suppose a sequence

in the MSA uses CM rule $S_i \rightarrow x_L S_{i+1} x_R$. With the MSA structure removed, this will correspond to profile HMM rules $\bar{S}_i^L \rightarrow x_L \bar{S}_{i+1}^L$ and $\bar{S}_i^R \rightarrow x_R \bar{S}_{i-1}^R$.

The counts used to set the probability for $\bar{S}_i^L \rightarrow x_L \bar{S}_{i+1}^L$ should be proportional to the frequency with which this rule is used by the profile HMM in parsing sequences from the CM-generated infinite MSA. This HMM rule is used for CM rules $S_i \rightarrow x_L S_{i+1} x_R$ for any x_R, so,

$$\mathcal{C}(\bar{S}_i^L \rightarrow x_L \bar{S}_{i+1}^L) \propto \sum_{x_R \in \{a,c,g,u,\varepsilon\}} \Pr(S_i \rightarrow x_L S_{i+1} x_R)$$

The virtual counts ($\mathcal{C}$) with the same left-hand side ($\bar{S}_i^L$) are then normalized into probabilities. Similar equations are used for the right-side HMM rule. (The algorithm is more complicated with fully general CMs; see supplement.) This method of setting profile HMM probabilities can be viewed as learning a maximum likelihood profile HMM from the distribution of MSAs induced by the CM.

2.2.3 Filtering with the profile HMM It is as in Weinberg and Ruzzo (2004b), but the threshold is independent of the CM's threshold. Our results indicate a rational basis for selecting the filter's heuristic threshold; even for difficult families, a filtering fraction of 0.01 was sufficient to find the majority of family members and provides a roughly hundred-fold speedup. (See supplement.)

2.3 Calculating ROC-like curves

In Section 3, we compare profile HMMs and BLAST. Both filter types have a tunable parameter, either heuristic probability threshold or *E*-value threshold. By varying the parameter, we can find more CM hits at the expense of less selective filtering, and therefore more CPU time, or the opposite.

To estimate CPU time, we measure filtering fraction (Weinberg and Ruzzo, 2004b). If the CM dominates overall scan time, the speedup from filtering is the reciprocal of the filtering fraction.

In Section 3, we will use an ROC-like curve, inspired by Metz (1978). ROC-like curves plot sensitivity versus filtering fraction at every threshold.

It is possible to calculate an ROC-like curve via a single scan of the database. Suppose a heuristic probability threshold is chosen, and consider under what circumstances a given nucleotide position will be scanned by the CM (i.e. will be in the numerator of the filtering fraction): the position will be scanned if the maximum of the probabilities within the W nucleotides to the right is above the threshold. So, for each nucleotide position, we compute the maximum of the probabilities of its W right neighbors, calling this the position's inclusion *point*. To obtain a filtering fraction f, we select a threshold that is less than a fraction f of the inclusion points. Thus, keeping a sorted list of inclusion points allows us to quickly look up a threshold for a given fraction, or the reverse.

Later, we will analyze the sensitivity of heuristics relative to a known set of ncRNAs detected by a given CM: how many of these can the heuristic filter detect at a given heuristic threshold? A given ncRNA will be detected by the filter—and submitted to the CM—if the inclusion points of each nucleotide within the ncRNA are above the selected heuristic threshold. Thus, for each ncRNA, we can calculate the heuristic threshold necessary to detect it.

We use this scheme to plot ROC-like curves. The scheme can also be used for the BLAST heuristic if we define analogous inclusion *E*-values for BLAST.

3 RESULTS

We compare profile HMMs with BLAST as a heuristic for CMs. Three types of profile HMMs are tested: ML-heuristic, ignore-SS (HMM built from the CM's input MSA, but ignoring secondary structure) and rigorous profile HMMs (Weinberg and Ruzzo, 2004a). Rigorous HMMs are used as heuristics by making their probability threshold a free parameter, instead of setting it to the CM's threshold. For tRNAs, we compare the ML-heuristic with tRNAscan-SE's heuristics.

3.1 ROC-like curves

Both profile HMMs and BLAST have a tunable parameter (heuristic threshold or *E*-value cutoff). Varying this parameter yields a curve showing different sensitivity versus filtering fraction points. By running profile HMMs on RFAMSEQ, we are able to plot sensitivity versus filtering fraction at all possible score thresholds, as described in Section 2. (The supplement says more about ROC-like curves for BLAST.)

3.2 Analysis of ROC-like curves

To test the heuristics on relatively difficult (highly structured) ncRNA families, we used families that could not be efficiently scanned using a rigorous profile HMM: 5S rRNA (Rfam ID RF00001), tRNA (RF00005), eubacterial RNase P (RF00010), the group II intron (RF00029), SECIS element (RF00031), and thiamin, lysine and cobalamin riboswitches (RF00059, RF00168, RF00174). Figure 1 shows a selection of the ROC-like curves. (The complete set is in the supplement.)

We scanned these families using sophisticated rigorous filters (Weinberg and Ruzzo, 2004b). However, rigorous filters were not feasible for RF00031, so we took the union of scans with all 4 heuristics (BLAST and HMMs) at relatively high filtering fractions. Based on a raw CM scan of a subset of vertebrate sequences, we estimate this union has ∼90% of RF00031 hits. In ROC-like curves, we assume for convenience that this ∼90% is the full set of SECIS elements.

In all cases, ignore-SS was no better than the ML-heuristic, and was typically much worse. Although ignore-SS uses the input MSA in a similar way to the CM, pseudocounts affect the CM differently from the profile HMM, so ignore-SS only accurately reflects sequence information when there are many training examples. More generally, an advantage of the ML-heuristic is that it allows any scheme for transforming an input MSA into a CM, since it only uses the CM.

The rigorous profile HMM's performance was slightly better than the ML-heuristic in small parts of some ROC-like curves, but its performance was more often noticeably worse than the ML-heuristic. Since rigorous filters must guarantee perfect sensitivity, they optimize for rare RNAs. The ML-heuristic optimizes for average-case performance, so is expected to be usually more accurate.

Moreover, creation of an ML-heuristic profile HMM takes ∼1 sec on a 2.8 GHz Pentium 4, versus 30 sec to several hours for a

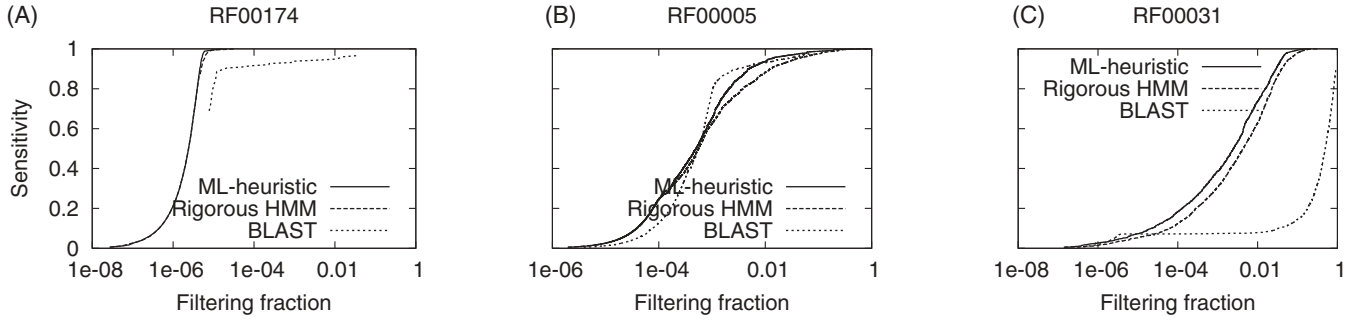

Fig. 1. Selected ROC-like curves. All plot sensitivity against filtering fraction, with filtering fraction in log scale. (**A**) RF00174 is typical of the other families; the ML-heuristic is slightly better than the rigorous profile HMM, and both often dramatically exceed BLAST. (**B**) Atypically, in RF00005, BLAST is superior, although only in one region. (**C**) BLAST performs especially poorly for RF00031. (Recall that rigorous scans were not possible for RF00031, so only ∼90% of hits are known; see text.) The supplement includes all ROC-like curves, and the inferior ignore-SS.

Table 1. ML-heuristic versus tRNAscan-SE heuristics

Sequence data	Size (Mb)	t-SE filtering fraction	Sensitivity of heuristic (%) t-SE	ML $W = 500$	ML $W = 100$	Total run time, heuristic + CM (CPU hours) t-SE	ML $W = 500$	ML $W = 100$	Raw CM
Archaea	47	0.0034	98.5	77.6	99.3	0.21	1.54	0.67	503
Eubacteria	640	0.0034	99.4	99.6	99.8	2.79	21.43	10.03	6553
Caenorhabditis elegans	100	0.0012	98.1	55.1	97.5	0.13	3.42	1.03	1056
Drosophila	117	0.00036	99.7	56.9	99.3	0.08	1.33	1.12	1233
Human	3070	0.00055	83.4	86.8	90.4	3.41	53.75	30.86	32422

First two columns: genome sequences tested and size in megabases. The next column is tRNAscan-SE's filtering fraction with default settings (with domain of life specified on the command line); the ML-heuristic scans were run at this same filtering fraction. The next three columns are sensitivity relative to rigorous scans for tRNAscan-SE ('t-SE'), the ML-heuristic with window length W=500 and $W = 100$. (See text for why $W = 500$ is worse.) The next four columns give measured run times, including the time used to run the CM. (Last column is without any filter.) Most of the increase in profile HMM run time is because the profile HMM is slower than tRNAscan-SE's heuristics.

rigorous profile HMM. Rigorous and ML-heuristic HMMs run at about the same speed; the ML-heuristic is preferred because it is faster to create, and its sensitivity (based on ROC-like curves) is more reliable. When rigorous HMMs are used rigorously, they guarantee perfect sensitivity, but the ability to use a less stringent threshold gives heuristics a considerable speed advantage.

BLAST was almost always worse than the profile HMMs. In the most extreme case, the BLAST heuristic found only 10% of RF00031 hits even with a filtering fraction as high as 0.1, where the ML-heuristic found an estimated 90% of them at this filtering fraction. There was a small, but interesting part of the ROC curve for tRNA (RF00005) for which BLAST was better; the very large number of training tRNA sequences may benefit BLAST.

Running BLAST is typically 2–4 times faster than a profile HMM scan, so the BLAST heuristic may be preferred for especially easy families where a low filtering fraction still yields high sensitivity. However, the single best heuristic as measured by ROC-like curves seems to be the ML-heuristic. This is particularly relevant at higher levels of sensitivity on these difficult families, where the time spent running the CM would dominate overall scan time.

3.3 tRNAscan-SE heuristics

We compared ML-heuristic profile HMMs with tRNAscan-SE. tRNAscan-SE is used to predict tRNAs in most genome projects. Its heuristics are based on two previously created tRNA annotation programs that were selected based on superior sensitivity and selectivity, from 7 tRNA annotation programs. Thus, tRNAscan-SE represents significant effort to create CM filters for a specific, well-studied family and exploits a substantial body of work on tRNA detectors. We were interested to see how well the generic ML-heuristic compares with this specialized case.

Because tRNAscan-SE has many complex parameters, we avoid ROC-like curves, using only the default parameters. Three eukaryotic nuclear genomes were scanned, as was archaeal and eubacterial DNA (all archael/eubacterial DNA in RFAMSEQ).

The filtering fraction of tRNAscan-SE was measured on each of the test genomic databases, ML-heuristic profile HMMs were run at the same filtering fraction, and run time and sensitivity were measured; see Table 1.

By default, tRNAscan-SE uses a window length of 500, but its heuristics can find intervals that are much smaller, and typically do. A window length of 500 is highly disadvantageous to our profile HMMs. For example, consider a 10 Kb sequence with 5 widely separated tRNAs of length 100. With window length 100, and filtering fraction 500/10 000, a profile HMM could potentially select all 5 tRNAs, attaining 100% sensitivity. However, with that same filtering fraction, but window length 500, even a perfect HMM could attain no better than 20% sensitivity, since only one of the tRNA hits (with 400 extraneous flanking nucleotides) could be reported without exceeding the filtering limit. This explains the

reduced sensitivity of the $W = 500$ case in Table 1. So, we also tried a window length of 100, the value used for Rfam's tRNA family. In all cases, the CM's window length was 500. (Overlapping windows of length 100 can create larger interval sizes.)

We emphasize that sensitivity measurements in Table 1 are the heuristic sensitivity relative to the CM, not the sensitivity relative to any experimental criteria. Both heuristics have low sensitivity on human, and tRNAscan-SE is lower. The HMM finds 64 hits that tRNAscan-SE does not. tRNAscan-SE has additional heuristics to predict pseudo-tRNAs, which indicate that 29/64 are likely pseudo-tRNAs. Of the 16 putative tRNAs found with tRNAscan-SE but not HMMs, 16/16 are predicted pseudo-tRNAs. Among the 592 hits common to tRNAscan-SE and the HMM, 91 are likely pseudo-tRNAs, so the change in number of pseudo-tRNAs is probably not drastic.

tRNAscan-SE's filters seem preferable to the ML-heuristic for tRNA detection, certainly in speed. However, if the results extrapolate to other ncRNA families, they suggest that generic heuristics are a more cost-effective solution for these other families than family-specific tools. The sensitivity of the ML-heuristic is similar to that of tRNAscan-SE, and the speed is in the same league, and still practical. Since most other ncRNA families have no detection tools like those in tRNAscan-SE's heuristics, the ML-heuristic would require significantly less work than a family-specific scheme, yet can be expected to provide comparable results.

4 DISCUSSION

Profile HMMs appear to have superior accuracy to BLAST, although the region in Figure 1B where BLAST was superior suggests it may have advantages in families with many known members. BLAST itself runs 2–4 faster than the HMM, making it a logical tool for families with high sequence conservation where BLAST's accuracy is adequate. Tuning of BLAST, e.g. tuning the DNA substitution matrix, may improve its overall accuracy. In using BLAST, there are two main heuristics being used: BLAST's word-matching heuristic to seed gapped alignments, and the gapped alignments used as a heuristic for CMs. It is unclear which heuristic is hurting sensitivity. The issue is subtle because, even though most database subsequences have an exact word match to at least some known family member, it may not match the most useful member for the alignment phase.

We were surprised by how competitive ML-heuristic profile HMMs are to tRNAscan-SE's heuristics, given that tRNAscan-SE was explicitly designed to detect tRNAs. The profile HMM itself is slower than tRNAscan-SE's heuristics, though not in a totally different class. In terms of sensitivity at a given filtering fraction, the heuristics were comparable, usually within a fraction of a percentage point of each other.

These results suggest that the ML-heuristic is generally a preferred method for heuristic filters of new ncRNA families for which creation of a family-specific filter would be a large amount of work. Although some families may have features different from tRNAs that family-specific filters could exploit, it is possible that such features could be integrated into a generic filter. Moreover, there is clearly room for improvement to the ML-heuristic, such as augmenting the profile HMM with structural information (Weinberg and Ruzzo, 2004b).

Our heuristics make scans for SECIS elements (RF00031) 9 times more sensitive than with BLAST. This allowed the detection of known, more diverged SECIS elements, e.g. the first known viral SECIS (in a poxvirus). So, our filters may be useful in selenoprotein detection pipelines (Kryukov *et al*., 2003). (See supplement.)

CMs were recently augmented with a 'local alignment' feature, which helps them to detect anomalous ncRNAs with missing/added domains. This functionality is used in several Rfam families and in the RSEARCH program (Klein and Eddy, 2003), which attempts to find homologs of a single ncRNA. A forthcoming paper will introduce heuristic and rigorous profile HMM filters for this feature, to speed RSEARCH and Rfam. That paper addresses a question not addressed in the current paper: what is the best filter to find homologs highly diverged from all training ncRNAs? RSEARCH (even with heuristic filters) can, e.g. find archaeal SRPs from human or from eubacterial SRPs. In this context, BLAST's performance is very poor compared with the ML-heuristic.

We have designed a new heuristic filter for CM searches, the ML-heuristic, and shown it superior to previous heuristics. We have also shown a method to evaluate a heuristic CM filter, using an ROC-like curve, and how results of rigorous scans can be used to measure the sensitivity of the filter itself. We therefore anticipate further developments in heuristic filters for CM-based searches.

ACKNOWLEDGEMENTS

We thank J. E. Barrick, S. R. Eddy, A. Bateman, R. H. Waterston, P. Green and departmental colleagues for helpful discussions, and anonymous referees for their feedback. This work was supported by the National Institutes of Health grants R01 HG02602 and NIH HG-00035.

Conflict of Interest. none declared.

REFERENCES

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis*. Cambridge University Press, Cambridge, UK.

Eddy,S.R. (2002) Computational genomics of noncoding RNA genes. *Cell*, **109**, 137–140.

Eddy,S.R. and Durbin,R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res*., **22**, 2079–2088.

Griffiths-Jones,S. *et al*. (2005) Rfam: annotating non-coding RNAs in complete genomes.. *Nucleic Acids Res*., **33**, D121–D124.

Kennedy,D. (2002) Breakthrough of the year. *Science*, **298**, 2283.

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

Kryukov,G. *et al*. (2003) *Science*, **300**, 1439–1443.

Lowe,T. and Eddy,S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res*., **25**, 955–964.

Mandal,M. *et al*. (2004) A glycine-dependent riboswitch that uses cooperative binding to control gene expression. *Science*, **306**, 275–279.

Metz,C.E. (1978) Basic principles of ROC analysis. *Semin. Nucl. Med*., **8**, 283–298.

Storz,G. (2002) An expanding universe of noncoding RNAs. *Science*, **296**, 1260–1263.

Wagner,E. and Flardh,K. (2002) Antisense RNAs everywhere? *Trends Genet*., **18**, 223–226.

Weinberg,Z. and Ruzzo,W.L. (2004a) Faster genome annotation of non-coding RNA families without loss of accuracy. In *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology (RECOMB)*, pp. 243–251. ACM Press.

Weinberg,Z. and Ruzzo,W.L. (2004b) Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. *Bioinformatics*, **20**, suppl. i334–i340.

Winkler,W.C. and Breaker,R.R. (2003) Genetic control by metabolite-binding riboswitches. *Chembiochem*., **4**, 1024–1032.