Supplementary Data

FindNonCoding: rapid and simple detection of non-coding RNAs in genomes

By Erik S. Wright

Supplemental Methods

The approach to identifying non-coding RNAs is split into two phases: training (*LearnNonCoding*) and searching (*FindNonCoding*). In the first phase, four patterns are learned from a multiple sequence alignment provided as input. The quality of the multiple sequence alignment is therefore important for accurate pattern recognition. Multiple sequence alignments can be constructed with DECIPHER using the *AlignSeqs* function with unaligned sequences as input. However, a higher-quality alignment can sometimes be obtained by aligning sequences to the Rfam seed alignment one-by-one using *AlignProfiles* and then removing the seed sequences from the final alignment. This approach is preferred when a large (> 100 sequences) seed alignment is available.

The goal of *LearnNonCoding* is to calculate parameters for a log-odds model of the form:

$$score = \sum_{i \in features} log\left(\frac{foreground_i}{background_i}\right)$$

Where *score* is the reported log-odds score for a putative hit, *features* is the set of extracted patterns representing a non-coding RNA family, *foreground* is the prevalence of a feature among training representatives from that family, and *background* is the feature's prevalence in random sequence.

In stage 1, *LearnNonCoding* identifies conserved sequence motifs represented in the form of a position weight matrix. Candidate motifs are found by applying a center point moving average to the entropy at each position in the multiple sequence alignment (with fewer than 50% gaps), and recording regions with average entropy less than 1.8 bits. To lower the chance candidate motifs will fail to match new sequences, regions are split when they are too long or when more than 5% of training sequences have a different length in the region due to insertions or deletions.

Each candidate motif is searched in the input sequences and in random background sequences of the same base composition (i.e., fraction of A, C, G, and U nucleotides). Log-odds scores are recorded for each position weight matrix hit, where the foreground is the probability of observing a base at a position in the motif, and the background is a uniform distribution (i.e., 25% of each nucleotide). Scores are discretized by binning into up to 10 bins that are uniformly distributed among observed scores. The resulting motifs are ranked by their discerning power, defined as the sum of absolute values of log-odds scores assigned to each bin multiplied by the

relative frequency of observing each bin, and up to the top 20 are kept (by default). This results in a set of sequence motifs (i.e., position weight matrices) defined by the 10th and 90th percentile of their positions relative to the beginning and end of the input sequences.

For example, the tRNA (RF00005) motifs look like:

	begin_low	begin_high	end_low	end_high	motif	pwm	minscore	prevalence	background
1	0	0	56	69	GsssssgTrGCtcAry	0.042959....	0, 4.875....	0.020442....	0.989092....
2	16	17	53	65	GGt	0.004460....	0.768894....	0.108749....	0.863800....
3	19	21	45	57	AraGCrc	0.870938....	1.015461....	0.150488....	0.885651....
4	27	29	38	50	gSmyTb	0.208338....	1.517247....	0.146883....	0.858252....
5	34	36	31	43	tAAkSc	0.229318....	1.938167....	0.369712....	0.914679....
6	41	45	28	38	kaG	0.200282....	1.562279....	0.614610....	0.777599....
7	45	54	26	29	T	0.033929....	0, Inf	0.184573....	0.366546....
8	46	58	22	24	CGbg	0.020719....	1.276844....	0.348993....	0.661977....
9	50	63	17	17	GGTTC	0.129566....	0, 4.855....	0.001602....	0.987247....
10	55	68	0	0	GArTCCygyyssssCr	0.221396....	0, 5.887....	0.003134....	0.993409....

In stage 2, *LearnNonCoding* identifies conserved secondary structure patterns in the input multiple sequence alignment. First, unless one is provided by the user, a consensus secondary structure is predicted using the *PredictDBN* function within DECIPHER. Second, all possible palindromes are recorded with a minimum stem length of four nucleotides, minimum loop length of three nucleotides, a maximum loop length of 500 nucleotides (by default), and up to one mismatch where wobble base pairs (i.e., G/U and U/G) are not considered mismatches. Free energy of all palindromes is predicted with RNA/RNA nearest neighbor parameters. For each input sequence, the stem loop with lowest free energy overlapping each predicted base pairing in the alignment is recorded.

This process typically results in a large set of hairpins and their positions relative to the beginning and end of the sequences. Next, steps are taken to rank hairpins relative to their discerning power and remove redundant hairpins covering overlapping positions in the input sequence alignment. The set of free energies for hairpins spanning two positions in the alignment are binned into up to 10 bins, and log-odds scores are determined based on the prevalence of each bin relative to a random background. Unlike motifs, hairpins are defined by two of three possible positions: their 10th and 90th percentiles of distances relative to the beginning of the sequences, end of the sequences, or the total number of nucleotides they span (i.e., width). The two distances are selected that have the smallest percentile range to minimize the likelihood the hairpin is observed by chance.

For example, the tRNA (RF00005) hairpins capture three of four hairpins in the canonical clover leaf secondary structure:

	begin_low	begin_high	end_low	end_high	width_low	width_high	length_low	length_high	dG	prevalence	background
1	-4	0	-3	1	71	92	8	14	-Inf, -1....	0.109306....	0.002576....
2	24	27	28	40	17	21	7	9	-Inf, -9....	0.096349....	0.006896....
3	47	60	8	10	15	17	6	7	-Inf, -6....	0.122790....	0.010533....

In stage 3, *LearnNonCoding* records the frequencies of k-mers in the input sequences. The value of k (between 1 and 4) is determined automatically from the diversity of input sequences such that all k-mers are observed at least 10 times. These k-mer frequencies are later used as the foreground in scoring, with the background being drawn from a window of up to 10,000 nucleotides centered around each position in the genome. This allows the background k-mer

distribution to vary across the length of the genome and avoids assigning too high of scores to k-mers in regions with GC-content more closely matching the input training sequences.

In stage 4, *LearnNonCoding* fits a sigmoidal function to the cumulative distribution of input sequence lengths. Care is taken to fit a smooth sigmoid, rather than a square wave, when the distribution of input sequences is very narrow. The probability density function is derived from the derivative of this sigmoid, and represents the expected distribution of foreground sequence lengths. The background is assumed to be a uniform distribution between 0.5 and 2.0 times the length of the shortest and longest input sequence, respectively. This sets the upper and lower bound of sequence lengths that can be detected.

In stage 5, *LearnNonCoding* calibrates the log-odds scores of the model so that they will be consistent across models and account for any dependencies that violate the assumption of independence among features. This is performed by searching random sequence with *FindNonCoding* and recording the scores of any hits. This process is repeated until a minimum number of observations are recorded, a maximum number of iterations is reached, or the observed scores of random hits fall well under the expected number (i.e., the false discovery rate of a high score should be less than e^{-score}). The right tail of the score distribution is fit to a censored log-normal distribution using maximum likelihood estimation. This results in two calibration parameters (i.e., the mean and standard deviation) that are used to transform reported scores for a *NonCoding* model.

FindNonCoding is designed to quickly find the beginning and ending position of hits in an input sequence (i.e., genome). To accomplish this, it first searches for each motif in the input sequence (and its reverse complement) and adds scores to each position of two numeric vectors: one for the beginning and one for the end position of matches. Log-odds scores for a motif hit are distributed between the beginning and ending positions of a candidate match relative to their span. For example, the first tRNA motif (above) starts at position zero relative to the beginning of the sequences and ends 56 to 69 positions from the end of the sequences. Therefore, the beginning has a span of 1 and the end has a span of 14 positions. Hits to this motif would have $14/15^{ths}$ of their score added to the "begin" vector at position zero from the start of the hit and $1/15^{th}$ of their score added to the "end" vector at positions 56 to 69 from the end of the hit. The remaining positions (i.e., those without any hits) are given a negative score according to the log-odds of not having observed the motif.

The resulting score vector is assessed for matches to the non-coding RNA family by adding the start and end scores at pairs of positions between the minimum and maximum length of the sequence (i.e., defined in stage 4 above). The combined score is calculated from the addition of the motif score, k-mer score, and length score. To carry forward, at least 60% of the desired minimum score must come from the combined score and 40% from the motif score at this point. Candidate matches that carry forward are searched for hairpins in the next phase. The observation of a hairpin is given a log-odds score based on its free energy (i.e., defined in stage 2 above) and the absence of an expected hairpin is given a negative score. These scores are added to the combined score and transformed by the calibration (i.e., see stage 5 above) to

report the final total score. In the last phase, the best scoring hit is selected when multiple hits are significantly overlapping, unless indicated otherwise by the user.

Table S1. Comparison of *FindNonCoding* and *StructRNAfinder* on the genome of *Chlamydia trachomatis* (NC_000117).

StructRNAfinder					*FindNonCoding*				
Name	**From**	**To**	**Score**	**E-value**	**Name**	**From**	**To**	**Score**	**Notes**
SSU_rRNA_bacteria-RF00177	854124	855676	1541	0	rRNA_16S-RF00177	854124	855676	169	
SSU_rRNA_archaea-RF01959	854129	855674	1031	0					Taxonomy = Archaea
LSU_rRNA_archaea-RF02540	855924	858861	1809	0					Taxonomy = Archaea
LSU_rRNA_bacteria-RF02541	855925	858862	2702	0	rRNA_23S-RF02541	855924	858861	208	
LSU_rRNA_eukarya-RF02543	856085	858850	1229	0					Taxonomy = Eukarya
SSU_rRNA_bacteria-RF00177	876170	877722	1541	0	rRNA_16S-RF00177	876170	877722	169	
SSU_rRNA_archaea-RF01959	876175	877720	1031	0					Taxonomy = Archaea
LSU_rRNA_archaea-RF02540	877970	880906	1821	0					Taxonomy = Archaea
LSU_rRNA_bacteria-RF02541	877971	880907	2714	0	rRNA_23S-RF02541	877970	880906	211	
LSU_rRNA_eukarya-RF02543	878131	880895	1235	0					Taxonomy = Eukarya
SSU_rRNA_microsporidia-RF02542	854129	855671	744	3.50E-227					Taxonomy = Microsporidia
SSU_rRNA_microsporidia-RF02542	876175	877717	744	3.50E-227					Taxonomy = Microsporidia
SSU_rRNA_eukarya-RF01960	854129	855671	696	1.20E-209					Taxonomy = Eukarya
SSU_rRNA_eukarya-RF01960	876175	877717	696	1.20E-209					Taxonomy = Eukarya
RNaseP_bact_a-RF00010	457003	457408	222	8.50E-72	RNase_P_class_A-RF00010	457003	457408	70	
tmRNA-RF00023	21082	20663	150	8.80E-44	tmRNA-RF00023	21082	20663	88	
PK-G12rRNA-RF01118	858256	858363	115	9.60E-30					Overlaps LSU rRNA
PK-G12rRNA-RF01118	880302	880409	115	9.60E-30					Overlaps LSU rRNA
tRNA-RF00005	752671	752743	77	5.80E-18	tRNA-Arg	752671	752746	67	
tRNA-RF00005	775336	775264	76	1.00E-17	tRNA-Lys	775336	775262	65	
tRNA-RF00005	202414	202341	73	1.30E-16	tRNA-Ile	202414	202339	69	
tRNA-RF00005	202492	202420	71	2.90E-16	tRNA-Ala	202492	202418	60	
tRNA-RF00005	363281	363210	70	5.90E-16	tRNA-Thr	363281	363207	60	
tRNA-RF00005	368396	368468	70	6.40E-16	tRNA-Met	368396	368471	77	

StructRNAfinder					*FindNonCoding*				
Name	**From**	**To**	**Score**	**E-value**	**Name**	**From**	**To**	**Score**	**Notes**
5S_rRNA-RF00001	858982	859096	74	9.00E-16	rRNA_5S-RF00001	858982	859096	106	
5S_rRNA-RF00001	881027	881141	74	9.00E-16	rRNA_5S-RF00001	881027	881141	105	
tRNA-RF00005	853628	853555	69	1.10E-15	tRNA-His	853628	853554	83	
tRNA-RF00005	42801	42730	69	1.40E-15	tRNA-Asn	42801	42727	70	
tRNA-RF00005	158662	158734	69	1.60E-15	tRNA-Thr	158662	158736	56	
tRNA-RF00005	778661	778591	68	3.20E-15	tRNA-Gly	778661	778590	61	
tRNA-RF00005	250442	250370	67	3.50E-15	tRNA-Val	250442	250368	62	
tRNA-RF00005	682286	682214	67	5.90E-15	tRNA-Arg	682286	682211	58	
tRNA-RF00005	984330	984414	66	6.60E-15	tRNA-Ser	984330	984417	69	
tRNA-RF00005	68995	68921	66	7.10E-15	tRNA-Pro	68995	68920	79	
tRNA-RF00005	814611	814540	65	1.40E-14	tRNA-Gly	814611	814539	77	
tRNA-RF00005	409238	409324	65	2.10E-14	tRNA-Ser	409238	409327	59	
tRNA-RF00005	234447	234375	64	2.70E-14	tRNA-Met	234447	234372	59	
RNaseP_bact_b-RF00011	457007	457397	55	3.00E-14					Overlaps RNaseP (Bact A)
tRNA-RF00005	574985	574897	64	3.20E-14	tRNA-Ser	574985	574894	65	
tRNA-RF00005	490055	489983	64	3.30E-14	tRNA-Phe	490056	489982	82	
tRNA-RF00005	485330	485244	64	4.10E-14	tRNA-Ser	485330	485242	89	
tRNA-RF00005	979594	979521	63	4.90E-14	tRNA-Val	979594	979520	55	
tRNA-RF00005	543862	543935	63	5.20E-14	tRNA-Arg	543862	543937	64	
RNaseP_arch-RF00373	457004	457403	53	6.70E-14					Taxonomy = Archaea
tRNA-RF00005	773399	773471	63	8.10E-14	tRNA-Thr	773399	773471	61	
tRNA-RF00005	250362	250289	62	9.10E-14	tRNA-Asp	250362	250288	64	
tRNA-RF00005	541063	541145	62	9.10E-14	tRNA-Leu	541062	541146	58	
tRNA-RF00005	361939	361867	62	9.50E-14	tRNA-Trp	361939	361865	51	
tRNA-RF00005	888006	887936	62	1.20E-13	tRNA-Cys	888006	887935	80	
tRNA-RF00005	158744	158826	61	1.60E-13	tRNA-Tyr	158744	158827	52	

StructRNAfinder					*FindNonCoding*				
Name	**From**	**To**	**Score**	**E-value**	**Name**	**From**	**To**	**Score**	**Notes**
tRNA-RF00005	1018111	1018192	61	2.50E-13	tRNA-Leu	1018111	1018195	60	
tRNA-RF00005	234356	234283	61	2.60E-13	tRNA-Met	234356	234281	70	
tRNA-RF00005	937062	937133	61	2.90E-13	tRNA-Gln	937062	937135	71	
tRNA-RF00005	582069	581987	60	4.10E-13	tRNA-Leu	582069	581985	41	
tRNA-RF00005	725508	725436	59	6.90E-13	tRNA-Ala	725508	725434	77	
tRNA-RF00005	718303	718376	58	1.70E-12	tRNA-Pro	718303	718378	76	
tRNA-RF00005	775427	775353	56	5.00E-12	tRNA-Glu	775425	775352	61	
tRNA-RF00005	605752	605835	54	1.50E-11	tRNA-Leu	605752	605837	56	
Bacteria_small_SRP-RF00169	286644	286546	53	2.30E-11	SmallSRP-RF00169	286644	286547	20	
tRNA-RF00005	546070	545989	51	1.40E-10	tRNA-Leu	546070	545987	63	
tRNA-Sec-RF01852	752670	752743	44	8.10E-08					Overlaps tRNA
tRNA-Sec-RF01852	582070	581987	41	5.10E-07					Overlaps tRNA
tRNA-Sec-RF01852	984330	984413	39	1.10E-06					Overlaps tRNA
tRNA-Sec-RF01852	541063	541144	38	2.40E-06					Overlaps tRNA
tmRNA-RF00023	202492	202417	27	2.90E-06					Overlaps tRNA
Bacteria_large_SRP-RF01854	286647	286545	45	3.20E-06					Overlaps small SRP
tRNA-Sec-RF01852	368395	368468	37	3.50E-06					Overlaps tRNA
tRNA-Sec-RF01852	409238	409323	37	5.00E-06					Overlaps tRNA
tRNA-Sec-RF01852	778661	778592	36	5.90E-06					Overlaps tRNA
tRNA-Sec-RF01852	158744	158825	35	1.30E-05					Overlaps tRNA
tRNA-Sec-RF01852	682286	682215	34	2.00E-05					Overlaps tRNA
tRNA-Sec-RF01852	485330	485245	34	2.40E-05					Overlaps tRNA
tRNA-Sec-RF01852	574985	574898	34	2.40E-05					Overlaps tRNA
tRNA-Sec-RF01852	234446	234377	33	3.50E-05					Overlaps tRNA
tRNA-Sec-RF01852	543861	543935	33	5.40E-05					Overlaps tRNA
U1-RF00003	939656	939817	24	9.00E-05					Taxonomy = Eukarya

StructRNAfinder					*FindNonCoding*				
Name	**From**	**To**	**Score**	**E-value**	**Name**	**From**	**To**	**Score**	**Notes**
tRNA-Sec-RF01852	888006	887937	32	0.0001					Overlaps tRNA

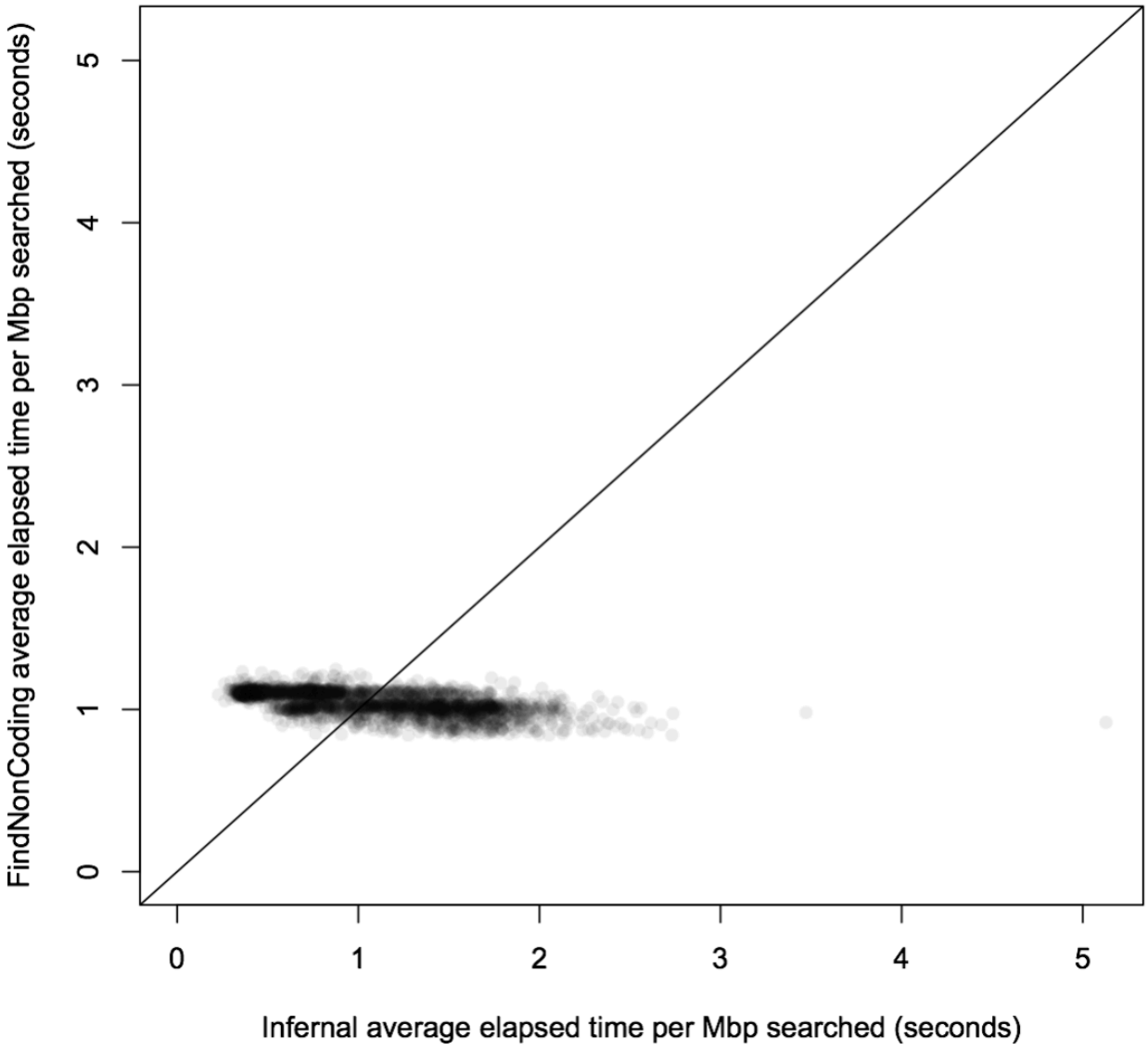

Figure S1. *FindNonCoding* and Infernal have similar search times on average. The plot shows the time required to search each million base pairs (Mbp) of 2,774 genomes (points). Although Infernal had a wider distribution of search times, on average it took 1.15 seconds per Mbp, whereas *FindNonCoding* took 1.04 seconds per Mbp. Both programs were forced to use a single processor in this comparison.

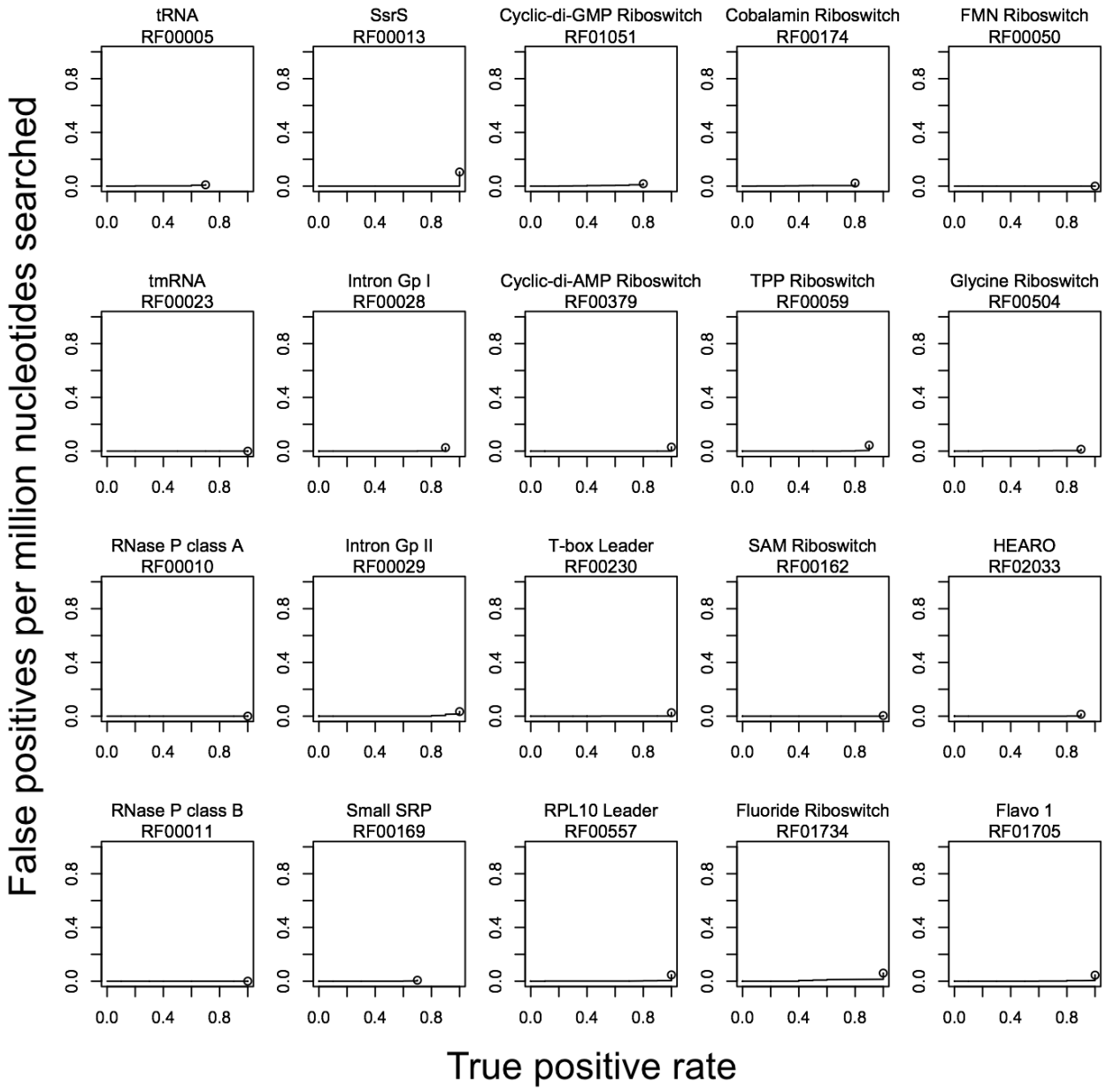

Figure S2. Analysis of sensitivity versus specificity using synthetic genomes. A single test sequence was randomly drawn from each Rfam family consisting of up to 1000 non-coding RNAs identified by Infernal. All sequences above 60% sequence identity to the test sequence were removed from the Rfam family before training with *LearnNonCoding*. The resulting *NonCoding* model was used to search 10 million random DNA nucleotides of even base content, within which the representative sequence was embedded. *FindNonCoding* hits were recorded with a log-odds score of at least 16 (i.e., the end point on each curve), corresponding to a maximum predicted false discovery rate of 0.1 per 10^6 nucleotides (i.e., e^{-16}). This process was repeated ten times per non-coding RNA family to determine true and false positive rates under the assumption that all the original representative sequences (found by Infernal) actually belong to the non-coding RNA family.

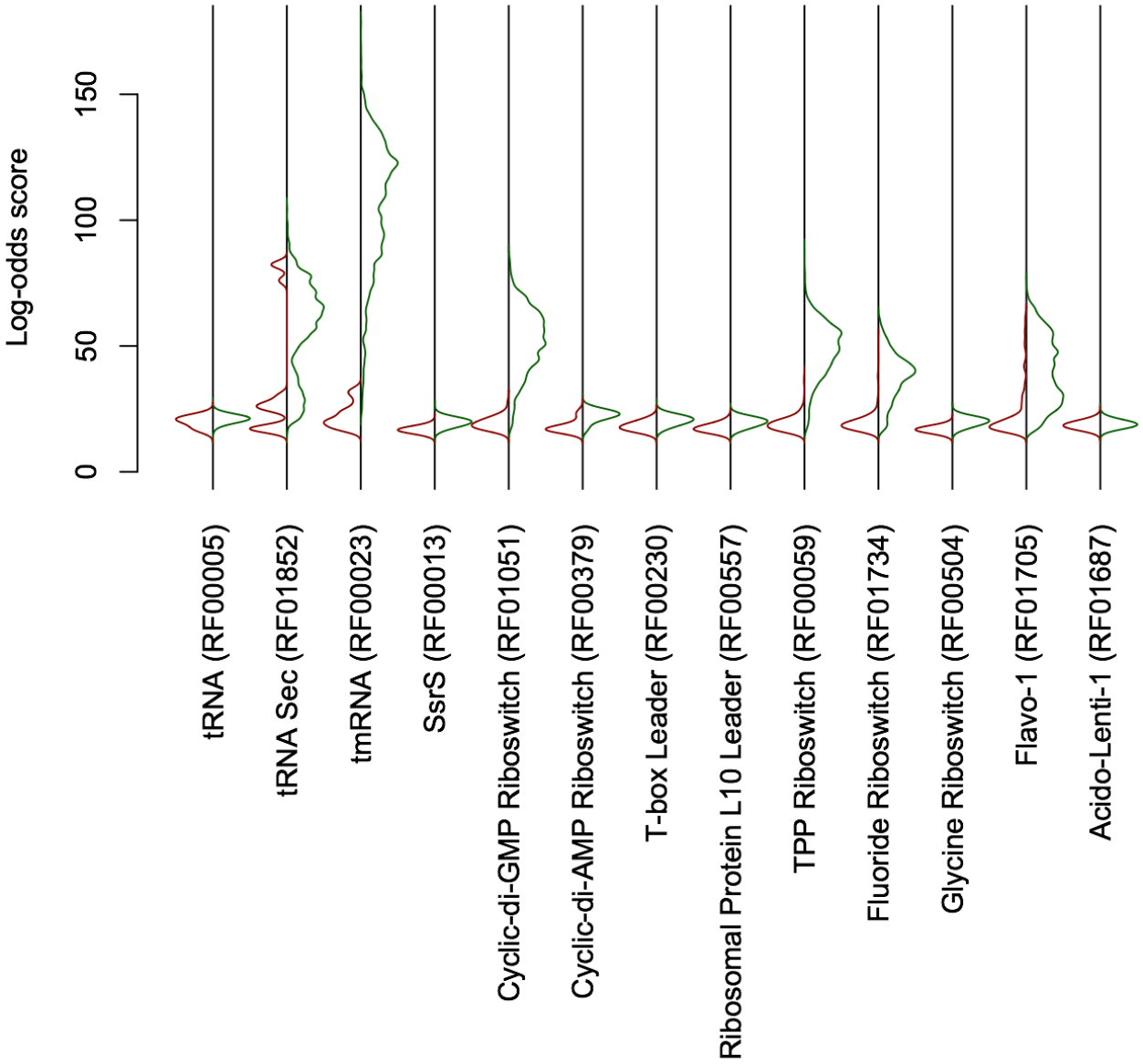

Figure S3. Distributions of scores for predicted non-coding RNAs. The distribution of scores are shown for false (red; left) and true (green; right) positive non-coding RNAs identified in bacterial genomes. Only non-coding RNA families with more than 10 false positives in 2,774 genomes are shown, with the distributions normalized to the maximum density of true and false positives in each non-coding RNA family. Only a small subset of false positives (i.e., non-coding RNA sequences overlapping with protein coding genes) had high scores.

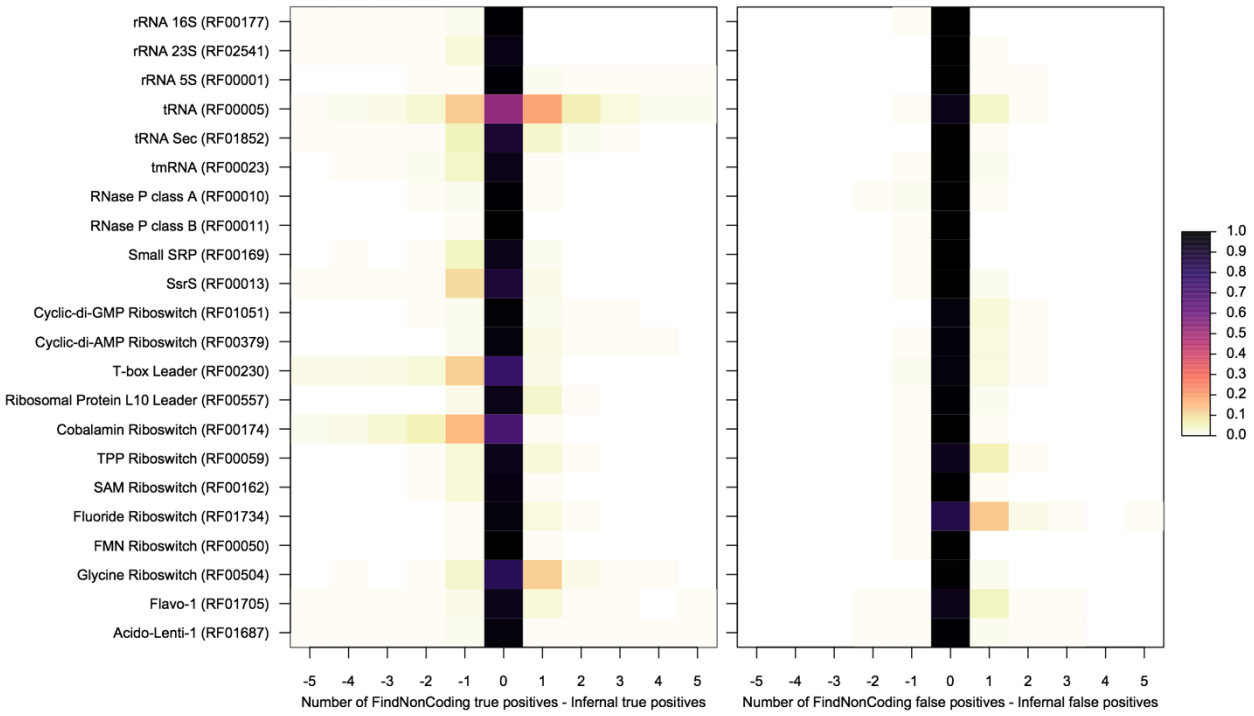

Figure S4. Relative number of true and false positives per genome. On average, Infernal identified more non-coding RNAs that do not significantly overlap with protein coding genes (left), although there are also some putative non-coding RNAs only identified by *FindNonCoding*. The color scale shows the difference in the number of assigned hits as a fraction of true or false positive hits per genome. Both programs yielded the same number of true and false positive hits in the vast majority of cases. Infernal is excellent at discriminating coding from non-coding sequence and identified fewer false positives than *FindNonCoding* (right). Only differences in the number of true or false positives of up to five per genome are shown because discrepancies beyond five were relatively rare (< 0.3%).