RNA Search and Motif Discovery

CSEP 590 B
Computational Biology
Many biologically interesting roles for RNA
RNA secondary structure prediction
Many interesting RNAs, e.g. Riboswitches
Approaches to Structure Prediction

Maximum Pairing
+ works on single sequences
+ simple
- too inaccurate

Minimum Energy
+ works on single sequences
- ignores pseudoknots
- only finds “optimal” fold

Partition Function
+ finds all folds
- ignores pseudoknots
Nussinov:

Computation Order

\[ B(i,j) = \# \text{ pairs in optimal pairing of } r_i \ldots r_j \]

\[ B(i,j) = 0 \text{ for all } i, j \text{ with } i \geq j-4; \text{ otherwise} \]

\[ B(i,j) = \max \text{ of:} \]

\[ \begin{cases} 
B(i,j-1) \\
\max \{ B(i,k-1)+1+B(k+1,j-1) \mid i \leq k < j-4 \text{ and } r_k-r_j \text{ may pair} \} 
\end{cases} \]

Time: \( O(n^3) \)

Loop-based energy version is better; recurrences similar, slightly messier
Optimal pairing of $r_i \ldots r_j$

Two possibilities

j Unpaired:
Find best pairing of $r_i \ldots r_{j-1}$

j Paired (with some $k$):
Find best $r_i \ldots r_{k-1}$ +
best $r_{k+1} \ldots r_{j-1}$ plus 1

Why is it slow?
Why do pseudoknots matter?
Structure prediction via comparative analysis
Covariance Models (CMs) represent RNA sequence/structure motifs
Fast CM search
Motif Discovery
Applications in prokaryotes & vertebrates
Approaches, II

Comparative sequence analysis
  + handles all pairings (potentially incl. pseudoknots)
  - requires several (many?) aligned, appropriately diverged sequences

Stochastic Context-free Grammars
  Roughly combines min energy & comparative, but no pseudoknots

Physical experiments (x-ray crystallography, NMR)
Covariation is strong evidence for base pairing
A  L19 (rptS) mRNA leader

Example: Ribosomal Autoregulation:
Excess L19 represses L19 (RF00556; 555-559 similar)

B  B. subtilis L19 mRNA leader

C  G • A  Watson-Crick base pair
G - C  other base interaction

nucleotide identity nucleotide present
N  97%  97%
N  90%  90%
N  75%  75%
N  50%  

compensatory mutations compatible mutations

stem loop always present

5' 3'
Mutual Information

\[ M_{ij} = \sum_{x_i,x_j} f_{x_i,x_j} \log_2 \frac{f_{x_i,x_j}}{f_{x_i}f_{x_j}}; \quad 0 \leq M_{ij} \leq 2 \]

Max when *no* seq conservation but perfect pairing

MI = expected score gain from using a pair state (below)

Finding optimal MI, (i.e. opt pairing of cols) is hard(?)

Finding optimal MI *without pseudoknots* can be done by dynamic programming
### M.I. Example (Artificial)

<table>
<thead>
<tr>
<th>Cols</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>U</td>
<td>A</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>Y</td>
<td>C</td>
<td>A</td>
<td>Y</td>
<td>C</td>
<td>U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>Y</td>
<td>C</td>
<td>G</td>
<td>U</td>
<td>Y</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Y</td>
<td>U</td>
<td>Y</td>
<td>U</td>
<td>Y</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>C</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>G</td>
<td>A</td>
<td>U</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>G</td>
<td>U</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>G</td>
<td>U</td>
<td>G</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>C</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>G</td>
<td>U</td>
<td>G</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>U</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>A</td>
<td>C</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>U</td>
<td>U</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>C</td>
<td>A</td>
<td>C</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>A</td>
<td>C</td>
<td>U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>16</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>16</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>16</td>
</tr>
</tbody>
</table>

- **Cols 1 & 9, 2 & 8:** perfect conservation & *might* be base-paired, but unclear whether they are. M.I. = 0
- **Cols 3 & 7:** *No* conservation, but always W-C pairs, so seems likely they do base-pair. M.I. = 2 bits.
- **Cols 7->6:** unconserved, but each letter in 7 has only 2 possible mates in 6. M.I. = 1 bit.
Figure 10.6 A mutual information plot of a tRNA alignment (top) shows four strong diagonals of covarying positions, corresponding to the four stems of the tRNA cloverleaf structure (bottom; the secondary structure of yeast phenylalanine tRNA is shown). Dashed lines indicate some of the additional tertiary contacts observed in the yeast tRNA-Phe crystal structure. Some of these tertiary contacts produce correlated pairs which can be seen weakly in the mutual information plot.
Find best (max total MI) subset of column pairs among i…j, subject to absence of pseudo-knots

\[ S_{i,j} = \max \left\{ S_{i,j-1}, \max_{i \leq k < j-4} S_{i,k-1} + M_{k,j} + S_{k+1,j-1} \right\} \]

“Just like Nussinov/Zucker folding”

BUT, need enough data---enough sequences at right phylogenetic distance
Computational Problems

How to predict secondary structure
How to model an RNA “motif” (i.e., sequence/structure pattern)
Given a motif, how to search for instances
Given (unaligned) sequences, find motifs
How to score discovered motifs
How to leverage prior knowledge
Motif Description
RNA Motif Models

“Covariance Models” (Eddy & Durbin 1994)
aka profile stochastic context-free grammars
aka hidden Markov models on steroids

Model position-specific nucleotide preferences and base-pair preferences

Pro: accurate
Con: model building hard, search slow
Eddy & Durbin 1994: What

A probabilistic model for RNA families
  The “Covariance Model”
  \approx A Stochastic Context-Free Grammar
  A generalization of a profile HMM

Algorithms for Training
  From aligned or unaligned sequences
  Automates “comparative analysis”
  Complements Nusinov/Zucker RNA folding

Algorithms for searching
Main Results

Very accurate search for tRNA
  (Precursor to tRNAscanSE - current favorite)
Given sufficient data, model construction comparable to, but not quite as good as, human experts
Some quantitative info on importance of pseudoknots and other tertiary features
Probabilistic Model Search

As with HMMs, given a sequence, you calculate likelihood ratio that the model could generate the sequence, vs a background model.

You set a score threshold. Anything above threshold → a “hit”

Scoring:

“Forward” / “Inside” algorithm - sum over all paths
Viterbi approximation - find single best path
(Bonus: alignment & structure prediction)
Example: searching for tRNAs
**Profile Hmm Structure**

*Figure 5.2* The transition structure of a profile HMM.

- **Mj**: Match states (20 emission probabilities)
- **Ij**: Insert states (Background emission probabilities)
- **Dj**: Delete states (silent - no emission)
How to model an RNA “Motif”?

Conceptually, start with a profile HMM:

- from a multiple alignment, estimate nucleotide/insert/delete preferences for each position
- given a new seq, estimate likelihood that it could be generated by the model, & align it to the model

```
AACAAAGccggccaggcuuucAGUF
GAAUAUCUuuugggauu......AGUF
GAAA..CA.................AGUF
GAAUAUCUuuauggauu......AGUF
```

mostly G

del

ins

all G
How to model an RNA “Motif”? 

Add “column pairs” and pair emission probabilities for base-paired regions
Table 5.2. The transition structure of a profile HMM.

- **M**: Match states (20 emission probabilities)
- **I**: Insert states (Background emission probabilities)
- **D**: Delete states (silent - no emission)
CM Structure

A: Sequence + structure
B: the CM “guide tree”
C: probabilities of letters/ pairs & of indels

Think of each branch being an HMM emitting both sides of a helix (but 3’ side emitted in reverse order)
Overall CM Architecture

One box (“node”) per node of guide tree

BEG/MATL/INS/DEL just like an HMM

MATP & BIF are the key additions: MATP emits pairs of symbols, modeling base-pairs; BIF allows multiple helices
CM Viterbi Alignment (the “inside” algorithm)

\[ x_i = \text{ }^{th} \text{ letter of input} \]
\[ x_{ij} = \text{substring } i,\ldots,j \text{ of input} \]
\[ T_{yz} = P(\text{transition } y \rightarrow z) \]
\[ E_{x_i x_j}^y = P(\text{emission of } x_i, x_j \text{ from state } y) \]
\[ S_{ij}^y = \max_\pi \log P(x_{ij} \text{ gen'd starting in state } y \text{ via path } \pi) \]
CM Viterbi Alignment
(the “inside” algorithm)

\[ S_{ij}^y = \max_\pi \log P(x_{ij} \text{ generated starting in state } y \text{ via path } \pi) \]

\[ S_{ij}^y = \begin{cases} 
\max_z [S_{i+1,j-1}^z + \log T_{yz} + \log E_{x_i,x_j}^y] & \text{match pair} \\
\max_z [S_{i+1,j}^z + \log T_{yz} + \log E_{x_i}^y] & \text{match/insert left} \\
\max_z [S_{i,j-1}^z + \log T_{yz} + \log E_{x_j}^y] & \text{match/insert right} \\
\max_z [S_{i,j}^z + \log T_{yz}] & \text{delete} \\
\max_{i<k<j} [S_{i,k}^{y,\text{left}} + S_{k+1,j}^{y,\text{right}}] & \text{bifurcation} 
\end{cases} \]

Time \( O(qn^3) \), \( q \) states, seq len \( n \)
compare: \( O(qn) \) for profile HMM
Primary vs Secondary Info

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Avg. id</th>
<th>Min id</th>
<th>Max id</th>
<th>ClustalV Accuracy</th>
<th>1° info (bits)</th>
<th>2° info (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>.402</td>
<td>.144</td>
<td>1.00</td>
<td>64%</td>
<td>43.7</td>
<td>30.0-32.3</td>
</tr>
<tr>
<td>SIM100</td>
<td>.396</td>
<td>.131</td>
<td>.986</td>
<td>54%</td>
<td>39.7</td>
<td>30.5-32.7</td>
</tr>
<tr>
<td>SIM65</td>
<td>.362</td>
<td>.111</td>
<td>.685</td>
<td>37%</td>
<td>31.8</td>
<td>28.6-30.7</td>
</tr>
</tbody>
</table>

\[ \left( \sum_{i=1}^{n} \max_{j} M_{i,j} \right) / 2 \]

disallowing / allowing pseudoknots
Model Training

unaligned sequences

random alignment

multiple alignment

alignment

(EM)

parameter reestimation

covariance model

model construction (structure prediction)
Comparison to TRNASCAN

Fichant & Burks - best heuristic then
  97.5% true positive
  0.37 false positives per MB
CM A1415 (trained on trusted alignment)
  > 99.98% true positives
  < 0.2 false positives per MB

Current method-of-choice is “tRNAscanSE”, a CM-based scan with heuristic pre-filtering (including TRNASCAN?) for performance reasons.
tRNAScanSE

Uses 3 older heuristic tRNA finders as prefilter

Uses CM built as described for final scoring

Actually 3(?) different CMs
  - eukaryotic nuclear
  - prokaryotic
  - organellar

Used in all genome annotation projects
An Important Application: Rfam
Rfam – an RNA family DB
Griffiths-Jones, et al., NAR ’03, ’05, ’08

Was biggest scientific comp user in Europe - 1000 cpu cluster for a month per release

Rapidly growing:
Rel 1.0, 1/03: 25 families, 55k instances
Rel 7.0, 3/05: 503 families, 363k instances
Rel 9.0, 7/08: 603 families, 636k instances
Rel 9.1, 1/09: 1372 families, 1148k instances
Rel 10.0, 1/10: 1446 families, 3193k instances

DB size:
~8GB
~160GB
**RF00037: Example Rfam Family**

**Input** (hand-curated):
- MSA “seed alignment”
- SS_cons

**Score Thresh T**

**Window Len W**

**Output:**
- CM
- scan results & “full alignment”
- phylogeny, etc.

### IRE (partial seed alignment):

<table>
<thead>
<tr>
<th>Species</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hom.sap.</td>
<td>GUUCCUGCUUCAACAGUGUUUGGAUGGAAC</td>
</tr>
<tr>
<td>Hom.sap.</td>
<td>UUUUCUC.UUCAACAGUGUUUGGAUGGAAC</td>
</tr>
<tr>
<td>Hom.sap.</td>
<td>UUUCUGUUUCAACAGUGCUUGGA.GGAAC</td>
</tr>
<tr>
<td>Hom.sap.</td>
<td>UUUAUC..AGUGACAGAGUUCACU.AUAAA</td>
</tr>
<tr>
<td>Hom.sap.</td>
<td>UCUUCUUGCUUCAACAGUGUUUGGAUGGAAC</td>
</tr>
<tr>
<td>Hom.sap.</td>
<td>AUUAUC..GGGAACAGUGUUUCCC.AUAAU</td>
</tr>
<tr>
<td>Hom.sap.</td>
<td>UCUUGC..UUCAACAGUGUUUGGACGGAAG</td>
</tr>
<tr>
<td>Hom.sap.</td>
<td>UGUAUC..GGAGACAGUGACCUCC.AUAUG</td>
</tr>
<tr>
<td>Hom.sap.</td>
<td>AUUAUC..GGAGACAGUGCUUCUCELL.AUAAU</td>
</tr>
<tr>
<td>Cav.por.</td>
<td>UCUCCUCUUBAACAGUGCUUGGACGGAGC</td>
</tr>
<tr>
<td>Mus.mus.</td>
<td>UAUAGACAGUGACCUCC.AUAUG</td>
</tr>
<tr>
<td>Mus.mus.</td>
<td>UUUCUGCUUCAACAGUGCUUGGAACGGAAC</td>
</tr>
<tr>
<td>Mus.mus.</td>
<td>GUACUUGCUUCAACAGUGUUUGGAACGGAAC</td>
</tr>
<tr>
<td>Rat.nor.</td>
<td>UAUAGACAGUGACCUCC.AUAUG</td>
</tr>
<tr>
<td>Rat.nor.</td>
<td>UAUACUUGCUUCAACAGUGUUUGGAACGGAAC</td>
</tr>
<tr>
<td>SS_cons</td>
<td>&lt;&lt;&lt;&lt;&lt;&lt;...&lt;&lt;&lt;&lt;&lt;&lt;....&gt;&gt;&gt;&gt;&gt;.&gt;&gt;&gt;&gt;&gt;</td>
</tr>
</tbody>
</table>
Rfam – key issues

Overly narrow families
Variant structures/unstructured RNAs
Spliced RNAs
RNA pseudogenes
  Human ALU is SRP related w/ 1.1m copies
  Mouse B2 repeat (350k copies) tRNA related
Speed & sensitivity
Motif discovery/hand-made models
Homology search

“Homolog” – similar by descent from common ancestor

Sequence-based
  Smith-Waterman
  FASTA
  BLAST

For RNA, sharp decline in sensitivity at ~60-70% identity

So, use structure, too
Impact of RNA homology search

(Barrick, et al., 2004)

**glycine riboswitch**

**operon**

- **B. subtilis**
- **L. innocua**
- **A. tumefaciens**
- **V. cholera**
- **M. tuberculosis**

(and 19 more species)
Impact of RNA homology search

(Barrick, et al., 2004)

B. subtilis

L. innocua

A. tumefaciens

V. cholera

M. tuberculosis

(and 19 more species)

BLAST-based

(Mandal, et al., 2004)

(and 42 more species)

CM-based

glycine riboswitch

operon
Faster Genome Annotation of Non-coding RNAs Without Loss of Accuracy

Zasha Weinberg

& W.L. Ruzzo

Recomb ‘04, ISMB ‘04, Bioinfo ‘06
RaveNnA: Genome Scale RNA Search

Typically 100x speedup over raw CM, w/ no loss in accuracy:
- Drop structure from CM to create a (faster) HMM
- Use that to pre-filter sequence;
- Discard parts where, provably, CM score < threshold;
- Actually run CM on the rest (the promising parts)
- Assignment of HMM transition/emission scores is key
  (a large convex optimization problem)

CM’s are good, but slow

Rfam Reality

EMBL → BLAST → CM → hits → junk

1 month, 1000 computers

Our Work

EMBL → Ravenna → CM → hits

~2 months, 1000 computers

Rfam Goal

EMBL → CM → junk → hits

10 years, 1000 computers
Covariance Model

Key difference of CM vs HMM: Pair states emit paired symbols, corresponding to base-paired nucleotides; 16 emission probabilities here.
Oversimplified CM
(for pedagogical purposes only)
CM to HMM

25 emissions per state       5 emissions per state, 2x states
Key Issue: 25 scores → 10

Need: log Viterbi scores CM ≤ HMM
Viterbi/Forward Scoring

Path $\pi$ defines transitions/emissions
Score($\pi$) = product of “probabilities” on $\pi$
NB: ok if “probs” aren’t, e.g. $\sum \neq 1$
(e.g. in CM, emissions are odds ratios vs 0th-order background)

For any nucleotide sequence $x$:

Viterbi-score($x$) = max{ score($\pi$) | $\pi$ emits $x$}

Forward-score($x$) = $\sum$ { score($\pi$) | $\pi$ emits $x$}
Key Issue: 25 scores → 10

Need: log Viterbi scores CM ≤ HMM

\[
\begin{align*}
P_{AA} & \leq L_A + R_A \\
P_{AC} & \leq L_A + R_C \\
P_{AG} & \leq L_A + R_G \\
P_{AU} & \leq L_A + R_U \\
P_{A-} & \leq L_A + R_- \\
P_{CA} & \leq L_C + R_A \\
P_{CC} & \leq L_C + R_C \\
P_{CG} & \leq L_C + R_G \\
P_{CU} & \leq L_C + R_U \\
P_{C-} & \leq L_C + R_- \\
\end{align*}
\]

NB: HMM not a prob. model
Rigorous Filtering

Any scores satisfying the linear inequalities give rigorous filtering

Proof:

CM Viterbi path score
≤ “corresponding” HMM path score
≤ Viterbi HMM path score

(even if it does not correspond to any CM path)

\[
\begin{align*}
P_{AA} & \leq L_A + R_A \\
P_{AC} & \leq L_A + R_C \\
P_{AG} & \leq L_A + R_G \\
P_{AU} & \leq L_A + R_U \\
P_{A-} & \leq L_A + R_- \\
\end{align*}
\]
Some scores filter better

\[ P_{UA} = 1 \leq L_U + R_A \]
\[ P_{UG} = 4 \leq L_U + R_G \]

Option 1:
\[ L_U = R_A = R_G = 2 \]

Option 2:
\[ L_U = 0, R_A = 1, R_G = 4 \]

Assuming ACGU ≈ 25%

<table>
<thead>
<tr>
<th>Opt 1:</th>
<th>( L_U + (R_A + R_G)/2 = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt 2:</td>
<td>( L_U + (R_A + R_G)/2 = 2.5 )</td>
</tr>
</tbody>
</table>
Optimizing filtering

For any nucleotide sequence $x$:

- Viterbi-score($x$) = $\max\{ \text{score}(\pi) \mid \pi \text{ emits } x \}$
- Forward-score($x$) = $\sum\{ \text{score}(\pi) \mid \pi \text{ emits } x \}$

Expected Forward Score

$$E(L_i, R_i) = \sum_{\text{all sequences } x} \text{Forward-score}(x) \ast \text{Pr}(x)$$

NB: $E$ is a function of $L_i, R_i$ only

Optimization:

Minimize $E(L_i, R_i)$ subject to score Lin.Ineq.s

This is heuristic ("forward $\downarrow$ $\Rightarrow$ Viterbi $\downarrow$ $\Rightarrow$ filter $\downarrow$")

But still rigorous because "subject to score Lin.Ineq.s"
Calculating $E(L_i, R_i)$

$$E(L_i, R_i) = \sum_x \text{Forward-score}(x) \times \text{Pr}(x)$$

Forward-like: for every state, calculate expected score for all paths ending there; easily calculated from expected scores of predecessors & transition/emission probabilities/scores
Minimizing $E(L_i, R_i)$

Calculate $E(L_i, R_i)$ symbolically, in terms of emission scores, so we can do partial derivatives for numerical convex optimization algorithm.

Forward:

- $f_k(i) = P(x_1 \ldots x_i, \pi_i = k)$
- $f_i(i + 1) = e_i(x_{i+1}) \sum_k f_k(i) a_{k,l}$

Viterbi:

- $v_i(i + 1) = e_i(x_{i+1}) \cdot \max_k (v_k(i) a_{k,l})$
Assignment of scores/ “probabilities”

Convex optimization problem

- **Constraints**: enforce rigorous property
- **Objective function**: filter as aggressively as possible

Problem sizes:
- 1000-10000 variables
- 10000-100000 inequality constraints
“Convex” Optimization

Convex:
local max = global max;
simple “hill climbing” works

Nonconvex:
can be many local maxima,
≪ global max;
“hill-climbing” fails
## Estimated Filtering Efficiency

(139 Rfam 4.0 families)

<table>
<thead>
<tr>
<th>Filtering fraction</th>
<th># families (compact)</th>
<th># families (expanded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; $10^{-4}$</td>
<td>105</td>
<td>110</td>
</tr>
<tr>
<td>$10^{-4}$ - $10^{-2}$</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>.01 - .10</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>.10 - .25</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>.25 - .99</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>.99 - 1.0</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

~100x speedup

≈ break even

Averages 283 times faster than CM
## Results: new ncRNAs (?)

<table>
<thead>
<tr>
<th>Name</th>
<th># Known (BLAST + CM)</th>
<th># New (rigorous filter + CM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pyrococcus snoRNA</td>
<td>57</td>
<td>123</td>
</tr>
<tr>
<td>Iron response element</td>
<td>201</td>
<td>121</td>
</tr>
<tr>
<td>Histone 3’ element</td>
<td>1004</td>
<td>102*</td>
</tr>
<tr>
<td>Retron msr</td>
<td>11</td>
<td>48</td>
</tr>
<tr>
<td>Hammerhead I</td>
<td>167</td>
<td>26</td>
</tr>
<tr>
<td>Hammerhead III</td>
<td>251</td>
<td>13</td>
</tr>
<tr>
<td>U6 snRNA</td>
<td>1462</td>
<td>2</td>
</tr>
<tr>
<td>U7 snRNA</td>
<td>312</td>
<td>1</td>
</tr>
<tr>
<td>cobalamin riboswitch</td>
<td>170</td>
<td>7</td>
</tr>
<tr>
<td>13 other families</td>
<td>5-1107</td>
<td>0</td>
</tr>
</tbody>
</table>
## Results: With additional work

<table>
<thead>
<tr>
<th></th>
<th># with BLAST+CM</th>
<th># with rigorous filter series + CM</th>
<th># new</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rfam tRNA</td>
<td>58609</td>
<td>63767</td>
<td>5158</td>
</tr>
<tr>
<td>Group II intron</td>
<td>5708</td>
<td>6039</td>
<td>331</td>
</tr>
<tr>
<td>tRNAscan-SE (human)</td>
<td>608</td>
<td>729</td>
<td>121</td>
</tr>
<tr>
<td>tmRNA</td>
<td>226</td>
<td>247</td>
<td>21</td>
</tr>
<tr>
<td>Lysine riboswitch</td>
<td>60</td>
<td>71</td>
<td>11</td>
</tr>
</tbody>
</table>

And more…
“Additional work”

Profile HMM filters use no 2\textsuperscript{ary} structure info

They work well because, tho structure can be critical to function, there is (usually) enough primary sequence conservation to exclude most of DB

But not on all families (and may get worse?)

Can we exploit some structure (quickly)?

Idea 1: “sub-CM”

Idea 2: extra HMM states remember mate

Idea 3: try lots of combinations of “some hairpins”

Idea 4: chain together several filters (select via Dijkstra)
Sub-CM filters

Full CM

Profile HMM

ACUCCCCAGAAGAGAAGAGAGAUUA

A sub-CM

Sub-CM

AAGAGAGAUUA

Sub-profile-HMM
Store-pair filters

Full CM

Store pair

“Profile” HMM:
Filter Chains

ACCGAT
GGACA

Rigorous filter

Rigorous filter

Rigorous filter

CM

ncRNAs
Why run filters in series?

<table>
<thead>
<tr>
<th></th>
<th>Filtering fraction</th>
<th>Run time (sec/Kbase)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter 1</td>
<td>0.25</td>
<td>1</td>
</tr>
<tr>
<td>Filter 2</td>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td>CM</td>
<td>N/A</td>
<td>200</td>
</tr>
</tbody>
</table>

CM alone: 200 s/Kb

Filter 1 → CM: \(1 + 0.25 \times 200 = 51\) s/Kb

Filter 2 → CM: \(10 + 0.01 \times 200 = 12\) s/Kb

Filter 1 → Filter 2 → CM:
\(1 + 0.25 \times 10 + 0.01 \times 200 = 5.5\) s/Kb
Properties of a filter:
• Filtering fraction
• Run time (sec/Kb)
Simplified performance model (selectivity & speed)

Independence assumptions for base pairs

Use dynamic programming to rapidly explore base pair combinations
Store pair

Sub-CM

Run time (sec/Kb)

Filtering fraction

Selected rigorous filter chain

97
Results: faster

CM: 30 years (your career)
Filters: 1 month (time between school terms)

Rigorous series of filters + CM time (days)
Results: more sensitive than BLAST

<table>
<thead>
<tr>
<th></th>
<th># with BLAST+CM</th>
<th># with rigorous filters + CM</th>
<th># new</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rfam tRNA</td>
<td>58609</td>
<td>63767</td>
<td>5158</td>
</tr>
<tr>
<td>Group II intron</td>
<td>5708</td>
<td>6039</td>
<td>331</td>
</tr>
<tr>
<td>Iron response element</td>
<td>201</td>
<td>322</td>
<td>121</td>
</tr>
<tr>
<td>tmRNA</td>
<td>226</td>
<td>247</td>
<td>21</td>
</tr>
<tr>
<td>Lysine riboswitch</td>
<td>60</td>
<td>71</td>
<td>11</td>
</tr>
</tbody>
</table>

And more…
Is there anything more to do?

Rigorous filters can be too cautious
  E.g., 10 times slower than heuristic filters
  Yet only 1-3% more sensitive

We want to
  Run scans faster with minimal loss of sensitivity
  Know empirically what sensitivity we’re losing
Heuristic Filters

Rigorous filters optimized for worst case
Possible to trade improved speed for small loss in sensitivity?
Yes – profile HMMs as before, but optimized for average case
Often 10x faster, modest loss in sensitivity
Heuristic Filters

ROC-like curves
(lysine riboswitch)

Filtering fraction

Filter sensitivity

Filter sends 80% of hits to CM
Heuristic Filters

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.

cobalamine (B_{12}) riboswitch
tRNA
SECIS
Software

Ravenna implements both rigorous and heuristic filters.

Infernal (engine behind Rfam) implements heuristic filters and some other (important) accelerations.

E.g., dynamic “banding” of dynamic programming matrix based on the insight that large deviations from consensus length must have low scores.
CM Search Summary

Still slower than we might like, but dramatic speedup over raw CM is possible with:

- No loss in sensitivity (provably), or
- Even faster with modest (and estimable) loss in sensitivity
Last Lecture

Part I
Many interesting RNAs, e.g. Riboswitches

Bacillus subtilis

coenzyme B1

adenine

thiamine pyrophosphate

lysine

guanine

pre-queosine

glycine

flavin mononucleotide

S-adenosyl-methionine

glucosamine-6-phosphate
Nussinov:

**Computation Order**

\[ B(i,j) = \text{# pairs in optimal pairing of } r_i \ldots r_j \]

\[ B(i,j) = 0 \text{ for all } i, j \text{ with } i \geq j-4; \text{ otherwise} \]

\[ B(i,j) = \max \text{ of:} \]

\[ \begin{align*}
B(i,j-1) \\
\max \{ B(i,k-1)+1+B(k+1,j-1) \mid i \leq k < j-4 \text{ and } r_k-r_j \text{ may pair} \}
\end{align*} \]

**Time:** \( O(n^3) \)

Loop-based energy version is better; recurrences similar, slightly messier
A: Sequence + structure

B: the CM “guide tree”

C: probabilities of letters/ pairs & of indels

Think of each branch being an HMM emitting both sides of a helix (but 3’ side emitted in reverse order)
CM to HMM

25 emissions per state

5 emissions per state, 2x states
Motif Discovery
RNA Motif Discovery

CM’s are great, but where do they come from?

Key approach: comparative genomics
  Search for motifs with common secondary structure in a set of functionally related sequences.

Challenges
  Three related tasks
    Locate the motif regions.
    Align the motif instances.
    Predict the consensus secondary structure.

Motif search space is huge!
  Motif location space, alignment space, structure space.
RNA Motif Discovery

Would be great if: given 100 complete genomes from diverse species, we could automatically find all the RNAs.

State of the art: that’s hopeless

Hope: can we exploit biological knowledge to narrow the search space?
RNA Motif Discovery

More promising problem: given a 10-20 unaligned sequences of a few kb, most of which contain instances of one RNA motif of 100-200bp -- find it.

Example: 5’ UTRs of orthologous glycine cleavage genes from γ-proteobacteria

Example: corresponding introns of orthogolous vertebrate genes

Orthologs = counterparts in different species
Approaches

Align-First: Align sequences, then look for common structure

Fold-First: Predict structures, then try to align them

Joint: Do both together
Pitfall for sequence-alignment-first approach

Structural conservation ≠ Sequence conservation
Alignment without structure information is unreliable

CLUSTALW alignment of SECIS elements with flanking regions

same-colored boxes should be aligned
Approaches

Align-first: align sequences, then look for common structure

Fold-first: Predict structures, then try to align them

  single-seq struct prediction only ~ 60% accurate; exacerbated by flanking seq; no biologically-validated model for structural alignment

Joint: Do both together

  Sankoff – good but slow

  Heuristic
Our Approach: CMfinder

Simultaneous \textit{local} alignment, folding and CM-based motif description using an EM-style learning procedure

Yao, Weinberg & Ruzzo, \textit{Bioinformatics}, 2006
CMFinder

Simultaneous alignment, folding & motif description
Yao, Weinberg & Ruzzo, *Bioinformatics*, 2006

Combines folding & mutual information in a principled way.

- Folding predictions
- Smart heuristics
- Mutual Information
- Candidate alignment
  - CM
  - Realign
  - EM
Structure Inference

Part of M-step is to pick a structure that maximizes data likelihood

We combine:

- mutual information
- position-specific priors for paired/unpaired
  (based on single sequence thermodynamic folding predictions)

intuition: for similar seqs, little MI; fall back on single-sequence folding predictions

data-dependent, so not strictly Bayesian

Details: see paper
CMfinder Accuracy
(on Rfam families with flanking sequence)
Summary of Rfam test families and results

<table>
<thead>
<tr>
<th>ID</th>
<th>Family</th>
<th>Rfam ID</th>
<th>#seqs</th>
<th>%id</th>
<th>length</th>
<th>#hp</th>
<th>CMfinder</th>
<th>CW/Pfold</th>
<th>CW/RNAalifold</th>
<th>Carnac</th>
<th>Foldalign</th>
<th>ComRNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cobalamin</td>
<td>RF00174</td>
<td>71</td>
<td>49</td>
<td>216</td>
<td>4</td>
<td>0.59</td>
<td>0.05</td>
<td>0</td>
<td>X</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ctRNA_pGA1</td>
<td>RF00236</td>
<td>17</td>
<td>74</td>
<td>83</td>
<td>2</td>
<td>0.91</td>
<td>0.70</td>
<td>0.72</td>
<td>0</td>
<td>0.86</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Entero_CRE</td>
<td>RF00048</td>
<td>56</td>
<td>81</td>
<td>61</td>
<td>1</td>
<td>0.89</td>
<td>0.74</td>
<td>0.22</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Entero_OriR</td>
<td>RF00041</td>
<td>35</td>
<td>77</td>
<td>73</td>
<td>2</td>
<td>0.94</td>
<td>0.75</td>
<td>0.76</td>
<td>0.80</td>
<td>0.52</td>
<td>0.52</td>
</tr>
<tr>
<td>5</td>
<td>glmS</td>
<td>RF00234</td>
<td>14</td>
<td>58</td>
<td>188</td>
<td>4</td>
<td>0.83</td>
<td>0.12</td>
<td>0.18</td>
<td>0</td>
<td>-</td>
<td>0.13</td>
</tr>
<tr>
<td>6</td>
<td>Histone3</td>
<td>RF00032</td>
<td>63</td>
<td>77</td>
<td>26</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Intron_gpII</td>
<td>RF00029</td>
<td>75</td>
<td>55</td>
<td>92</td>
<td>2</td>
<td>0.80</td>
<td>0.30</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>IRE</td>
<td>RF00037</td>
<td>30</td>
<td>68</td>
<td>30</td>
<td>1</td>
<td>0.77</td>
<td>0.22</td>
<td>0</td>
<td>0</td>
<td>0.38</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>let-7</td>
<td>RF00027</td>
<td>9</td>
<td>69</td>
<td>84</td>
<td>1</td>
<td>0.87</td>
<td>0.08</td>
<td>0.42</td>
<td>0</td>
<td>0.71</td>
<td>0.78</td>
</tr>
<tr>
<td>10</td>
<td>lin-4</td>
<td>RF00052</td>
<td>9</td>
<td>69</td>
<td>72</td>
<td>1</td>
<td>0.78</td>
<td>0.51</td>
<td>0.75</td>
<td>0.41</td>
<td>0.65</td>
<td>0.24</td>
</tr>
<tr>
<td>11</td>
<td>Lysine</td>
<td>RF00168</td>
<td>48</td>
<td>48</td>
<td>183</td>
<td>4</td>
<td>0.77</td>
<td>0.24</td>
<td>0</td>
<td>X</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>mir-10</td>
<td>RF00104</td>
<td>11</td>
<td>66</td>
<td>75</td>
<td>1</td>
<td>0.66</td>
<td>0.59</td>
<td>0.60</td>
<td>0</td>
<td>0.48</td>
<td>0.33</td>
</tr>
<tr>
<td>13</td>
<td>Purine</td>
<td>RF00167</td>
<td>29</td>
<td>55</td>
<td>103</td>
<td>2</td>
<td>0.91</td>
<td>0.07</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0.27</td>
</tr>
<tr>
<td>14</td>
<td>RFN</td>
<td>RF00050</td>
<td>47</td>
<td>66</td>
<td>139</td>
<td>4</td>
<td>0.39</td>
<td>0.68</td>
<td>0.26</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>Rhino_CRE</td>
<td>RF00220</td>
<td>12</td>
<td>71</td>
<td>86</td>
<td>1</td>
<td>0.88</td>
<td>0.52</td>
<td>0.52</td>
<td>0.69</td>
<td>0.41</td>
<td>0.61</td>
</tr>
<tr>
<td>16</td>
<td>s2m</td>
<td>RF00164</td>
<td>23</td>
<td>80</td>
<td>43</td>
<td>1</td>
<td>0.67</td>
<td>0.80</td>
<td>0.45</td>
<td>0.64</td>
<td>0.63</td>
<td>0.29</td>
</tr>
<tr>
<td>17</td>
<td>S_box</td>
<td>RF00162</td>
<td>64</td>
<td>66</td>
<td>112</td>
<td>3</td>
<td>0.72</td>
<td>0.11</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>SECIS</td>
<td>RF00031</td>
<td>43</td>
<td>43</td>
<td>68</td>
<td>1</td>
<td>0.73</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>Tymo_tRNA-like</td>
<td>RF00233</td>
<td>22</td>
<td>72</td>
<td>86</td>
<td>4</td>
<td>0.81</td>
<td>0.33</td>
<td>0.36</td>
<td>0.30</td>
<td>0.80</td>
<td>0.48</td>
</tr>
</tbody>
</table>

Average Accuracy: 0.79 0.36 0.28 0.17 0.60 0.19
Average Specificity: 0.81 0.42 0.57 0.83 0.60 0.65
Average Sensitivity: 0.77 0.36 0.23 0.13 0.61 0.17

Min/Max in col **Bold** = best in row
Discovery in Bacteria

A Computational Pipeline for High-Throughput Discovery of cis-Regulatory Noncoding RNA in Prokaryotes

Zizhen Yao¹*, Jeffrey Barrick², Zasha Weinberg³, Shane Neph¹,⁴, Ronald Breaker²,³,⁵, Martin Tompa¹,⁴, Walter L. Ruzzo¹,⁴

Published online 9 July 2007

Identification of 22 candidate structured RNAs in bacteria using the CMfinder comparative genomics pipeline

Zasha Weinberg¹*, Jeffrey E. Barrick²,³, Zizhen Yao⁴, Adam Roth², Jane N. Kim¹, Jeremy Gore¹, Joy Xin Wang¹,², Elaine R. Lee¹, Kirsten F. Block¹, Narasimhan Sudarsan¹, Shane Neph⁵, Martin Tompa⁴,⁵, Walter L. Ruzzo⁴,⁵ and Ronald R. Breaker¹,²,³

doi:10.1093/nar/gkm487
Use the Right Data; Do Genome Scale Search
Right Data: Why/How

We can recognize, say, 5-10 good examples amidst 20 extraneous ones (but not 5 in 200 or 2000) of length 1k or 10k (but not 100k)

Regulators often near regulatees (protein coding genes), which are usually recognizable cross-species
So, look near similar genes (“homologs”)
Many riboswitches, e.g., are present in ~5 copies per genome
(Not strategy used in vertebrates - 1000x larger genomes)
A pipeline for RNA motif genome scans

Input from ~70 complete Firmicute genomes available in late 2005-early 2006, totaling ~200 megabases

Processing Times

- Identify CDD group members: 2946 CDD groups
- Retrieve upstream sequences
- Footprinter ranking: < 10 CPU days
- CMfinder: 35975 motifs
- Motif postprocessing: 1740 motifs
- RaveNnA: 10 CPU months
- CMfinder refinement: < 1 CPU month
- Motif postprocessing: 1466 motifs

Footprinter ranking:
- < 10 CPU days
- Footprinter ranking:
- CMfinder: 35975 motifs
- Motif postprocessing: 1740 motifs
- RaveNnA: 10 CPU months
- CMfinder refinement: < 1 CPU month
- Motif postprocessing: 1466 motifs
Table 1: Motifs that correspond to Rfam families

<table>
<thead>
<tr>
<th>Rank</th>
<th>Score RAV</th>
<th>Score CMF</th>
<th>Score FP</th>
<th># RAV</th>
<th># CMF</th>
<th>ID</th>
<th>Gene</th>
<th>Description</th>
<th>CDD</th>
<th>Rfam</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>43</td>
<td>107</td>
<td></td>
<td>3400</td>
<td></td>
<td></td>
<td></td>
<td>Thiamine pyrophosphate-requiring enzymes</td>
<td></td>
<td>RF00230 T-box</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>344</td>
<td></td>
<td>3115</td>
<td></td>
<td></td>
<td></td>
<td>Predicted membrane protein</td>
<td></td>
<td>RF00059 THI</td>
</tr>
<tr>
<td>2</td>
<td>77</td>
<td>1284</td>
<td></td>
<td>2376</td>
<td></td>
<td></td>
<td></td>
<td>Methionine synthase I specific DNA methylase</td>
<td></td>
<td>RF00162 S_box</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>5</td>
<td></td>
<td>2327</td>
<td></td>
<td></td>
<td></td>
<td>Predicted N6-adenine-specific DNA methylase</td>
<td></td>
<td>RF00011</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>66</td>
<td></td>
<td>2228</td>
<td></td>
<td></td>
<td></td>
<td>3,4-dihydroxy-2-butanoate 4-phosphate synthase</td>
<td></td>
<td>RF00050 RFN</td>
</tr>
<tr>
<td>5</td>
<td>145</td>
<td>952</td>
<td></td>
<td>1429</td>
<td></td>
<td></td>
<td></td>
<td>GMP synthase</td>
<td></td>
<td>RF00167 Purine</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>108</td>
<td></td>
<td>1322</td>
<td></td>
<td></td>
<td></td>
<td>Glycine cleavage system protein P</td>
<td></td>
<td>RF00504 Glycine</td>
</tr>
<tr>
<td>7</td>
<td>37</td>
<td>749</td>
<td></td>
<td>1235</td>
<td></td>
<td></td>
<td></td>
<td>Uncharacterised BCR, YbaB family COG0718</td>
<td></td>
<td>RF00169 SRP_bact</td>
</tr>
<tr>
<td>8</td>
<td>123</td>
<td>1358</td>
<td></td>
<td>1222</td>
<td></td>
<td></td>
<td></td>
<td>Cobalamin biosynthesis protein CobD/CbiB</td>
<td></td>
<td>RF00174 Cobalamin</td>
</tr>
<tr>
<td>9</td>
<td>137</td>
<td>1133</td>
<td></td>
<td>899</td>
<td></td>
<td></td>
<td></td>
<td>Diaminopimelate decarboxylase</td>
<td></td>
<td>RF00168 Lysine</td>
</tr>
<tr>
<td>10</td>
<td>36</td>
<td>141</td>
<td></td>
<td>896</td>
<td></td>
<td></td>
<td></td>
<td>Membrane protein TerC</td>
<td></td>
<td>RF00080 yybP-ykoY</td>
</tr>
<tr>
<td>11</td>
<td>202</td>
<td>684</td>
<td></td>
<td>664</td>
<td></td>
<td></td>
<td></td>
<td>Mg/Co/Ni transporter MgtE</td>
<td></td>
<td>RF00380 ykoK</td>
</tr>
<tr>
<td>12</td>
<td>26</td>
<td>74</td>
<td></td>
<td>645</td>
<td></td>
<td></td>
<td></td>
<td>Glucosamine 6-phosphate synthetase</td>
<td></td>
<td>RF00234 glmS</td>
</tr>
<tr>
<td>13</td>
<td>208</td>
<td>192</td>
<td></td>
<td>561</td>
<td></td>
<td></td>
<td></td>
<td>ABC-type proline/glycine betaine transport systems</td>
<td></td>
<td>RF00005 tRNA1</td>
</tr>
<tr>
<td>14</td>
<td>99</td>
<td>239</td>
<td></td>
<td>413</td>
<td></td>
<td></td>
<td></td>
<td>Membrane transporters of cations and cationic drug</td>
<td></td>
<td>RF00442 ykkC-ykkD</td>
</tr>
<tr>
<td>15</td>
<td>392</td>
<td>281</td>
<td></td>
<td>268</td>
<td></td>
<td></td>
<td></td>
<td>Uncharacterized conserved protein</td>
<td></td>
<td>RF00023 tmRNA</td>
</tr>
</tbody>
</table>

Table 1: Motifs that correspond to Rfam families. “Rank”: the three columns show ranks for refined motif clusters after genome scans (“RAV”), CMfinder motifs before genome scans (“CMF”), and FootPrint results (“FP”). We used the same ranking scheme for RAV and CMF. “Score”
<table>
<thead>
<tr>
<th>Rfam</th>
<th>Membership</th>
<th>Overlap</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Sn Sp</td>
<td>nt Sn Sp</td>
<td>bp Sn Sp</td>
</tr>
<tr>
<td>RF00174 Cobalamin</td>
<td>183 0.74 (^1) 0.97</td>
<td>152 0.75 0.85</td>
<td>20 0.60 0.77</td>
</tr>
<tr>
<td>RF00504 Glycine</td>
<td>92 0.56 (^1) 0.96</td>
<td>94 0.94 0.68</td>
<td>17 0.84 0.82</td>
</tr>
<tr>
<td>RF00234 glmS</td>
<td>34 0.92 1.00</td>
<td>100 0.54 1.00</td>
<td>27 0.96 0.97</td>
</tr>
<tr>
<td>RF00168 Lysine</td>
<td>80 0.82 0.98</td>
<td>111 0.61 0.68</td>
<td>26 0.76 0.87</td>
</tr>
<tr>
<td>RF00167 Purine</td>
<td>86 0.86 0.93</td>
<td>83 0.83 0.55</td>
<td>17 0.90 0.95</td>
</tr>
<tr>
<td>RF00050 RFN</td>
<td>133 0.98 0.99</td>
<td>139 0.96 1.00</td>
<td>12 0.66 0.65</td>
</tr>
<tr>
<td>RF00011 RNaseP_bact_b</td>
<td>144 0.99 0.99</td>
<td>194 0.53 1.00</td>
<td>38 0.72 0.78</td>
</tr>
<tr>
<td>RF00162 S_box</td>
<td>208 0.95 0.97</td>
<td>110 1.00 0.69</td>
<td>23 0.91 0.78</td>
</tr>
<tr>
<td>RF00169 SRP_bact</td>
<td>177 0.92 0.95</td>
<td>99 1.00 0.65 25 0.89 0.81</td>
<td></td>
</tr>
<tr>
<td>RF00230 T-box</td>
<td>453 0.96 0.61</td>
<td>187 0.77 1.00</td>
<td>5 0.32 0.38</td>
</tr>
<tr>
<td>RF00059 THI</td>
<td>326 0.89 1.00</td>
<td>99 0.91 0.69</td>
<td>13 0.56 0.74</td>
</tr>
<tr>
<td>RF00442 ykkC-yxkD</td>
<td>19 0.90 0.53</td>
<td>99 0.94 0.81</td>
<td>18 0.94 0.68</td>
</tr>
<tr>
<td>RF00380 ykoK</td>
<td>49 0.92 1.00</td>
<td>125 0.75 1.00</td>
<td>27 0.80 0.95</td>
</tr>
<tr>
<td>RF00080 yybP-ykoY</td>
<td>41 0.32 0.89</td>
<td>100 0.78 0.90</td>
<td>18 0.63 0.66</td>
</tr>
<tr>
<td>mean</td>
<td>145 0.84 0.91</td>
<td>121 0.81 0.82</td>
<td>21 0.75 0.77</td>
</tr>
<tr>
<td>median</td>
<td>113 0.91 0.97</td>
<td>105 0.81 0.83</td>
<td>19 0.78 0.78</td>
</tr>
</tbody>
</table>

**Tbl 2: Prediction accuracy compared to prokaryotic subset of Rfam full alignments.**

Membership: # of seqs in overlap between our predictions and Rfam’s, the sensitivity (Sn) and specificity (Sp) of our membership predictions. Overlap: the avg len of overlap between our predictions and Rfam’s (nt), the fractional lengths of the overlapped region in Rfam’s predictions (Sn) and in ours (Sp). Structure: the avg # of correctly predicted canonical base pairs (in overlapped regions) in the secondary structure (bp), and sensitivity and specificity of our predictions. \(^1\)After 2nd RaveNnA scan, membership Sn of Glycine, Cobalamin increased to 76% and 98% resp., Glycine Sp unchanged, but Cobalamin Sp dropped to 84%.
<table>
<thead>
<tr>
<th>Rank</th>
<th>#</th>
<th>CDD</th>
<th>Gene: Description</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>69</td>
<td>28178</td>
<td>DHOase IIa: Dihydroorotase</td>
<td>PyrR attenuator [22]</td>
</tr>
<tr>
<td>15</td>
<td>33</td>
<td>10097</td>
<td>RplL: Ribosomal protein L7/L1</td>
<td>L10 r-protein leader; see Supp [30]</td>
</tr>
<tr>
<td>19</td>
<td>36</td>
<td>10234</td>
<td>RpsF: Ribosomal protein S6</td>
<td>S6 r-protein leader</td>
</tr>
<tr>
<td>22</td>
<td>32</td>
<td>10897</td>
<td>COG1179: Dinucleotide-utilizing enzymes</td>
<td>6S RNA [25]</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>9926</td>
<td>RpsJ: Ribosomal protein S10</td>
<td>S10 r-protein leader; see Supp [46]</td>
</tr>
<tr>
<td>29</td>
<td>11</td>
<td>15150</td>
<td>Resolvase: N terminal domain</td>
<td>IF-3 r-protein leader; see Supp [47]</td>
</tr>
<tr>
<td>31</td>
<td>31</td>
<td>10164</td>
<td>InFC: Translation initiation factor 3</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>26</td>
<td>10393</td>
<td>RpsD: Ribosomal protein S4 and related proteins</td>
<td>S4 r-protein leader; see Supp [51]</td>
</tr>
<tr>
<td>44</td>
<td>30</td>
<td>10332</td>
<td>GroL: Chaperonin GroEL</td>
<td>HrcA DNA binding site</td>
</tr>
<tr>
<td>46</td>
<td>33</td>
<td>25629</td>
<td>Ribosomal L21p: Ribosomal prokaryotic L21 protein</td>
<td>L21 r-protein leader; see Supp [52]</td>
</tr>
<tr>
<td>50</td>
<td>11</td>
<td>5638</td>
<td>Cad: Cadmium resistance transporter</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>19</td>
<td>9965</td>
<td>RplB: Ribosomal protein L2</td>
<td>S10 r-protein leader</td>
</tr>
<tr>
<td>55</td>
<td>7</td>
<td>26270</td>
<td>RNA pol Rpb2 1: RNA polymerase beta subunit</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>9</td>
<td>13148</td>
<td>COG3830: ACT domain-containing protein</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>28</td>
<td>4174</td>
<td>Ribosomal S2: Ribosomal protein S2</td>
<td>S2 r-protein leader</td>
</tr>
<tr>
<td>74</td>
<td>9</td>
<td>9924</td>
<td>RpsG: Ribosomal protein S7</td>
<td>S12 r-protein leader</td>
</tr>
<tr>
<td>86</td>
<td>6</td>
<td>12328</td>
<td>COG2984: ABC-type uncharacterized transport system</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>19</td>
<td>24072</td>
<td>CtsR: Firmicutes transcriptional repressor of class III</td>
<td>CtsR DNA binding site [48]</td>
</tr>
<tr>
<td>100</td>
<td>21</td>
<td>23019</td>
<td>Formyl trans N: Formyl transferase</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>8</td>
<td>9916</td>
<td>PurE: Phosphoribosylcarboxyaminomimidazole</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>5</td>
<td>13411</td>
<td>COG4129: Predicted membrane protein</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>10</td>
<td>10075</td>
<td>RplO: Ribosomal protein L15</td>
<td>L15 r-protein leader</td>
</tr>
<tr>
<td>121</td>
<td>9</td>
<td>10132</td>
<td>RpmJ: Ribosomal protein L36</td>
<td>IF-1 r-protein leader</td>
</tr>
<tr>
<td>129</td>
<td>4</td>
<td>23962</td>
<td>Cna B: Cna protein B-type domain</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>9</td>
<td>25424</td>
<td>Ribosomal S12: Ribosomal protein S12</td>
<td>S12 r-protein leader</td>
</tr>
<tr>
<td>131</td>
<td>9</td>
<td>16769</td>
<td>Ribosomal L4: Ribosomal protein L4/L1 family</td>
<td>L3 r-protein leader</td>
</tr>
<tr>
<td>140</td>
<td>12</td>
<td>8892</td>
<td>Pencillinase R: Pencillinase repressor</td>
<td>Blal, MecI DNA binding site [49]</td>
</tr>
<tr>
<td>157</td>
<td>25</td>
<td>24415</td>
<td>Ribosomal S9: Ribosomal protein S9/S16</td>
<td>L13 r-protein leader; Fig 3</td>
</tr>
<tr>
<td>160</td>
<td>27</td>
<td>1790</td>
<td>Ribosomal L19: Ribosomal protein L19</td>
<td>L19 r-protein leader; Fig 2</td>
</tr>
<tr>
<td>164</td>
<td>6</td>
<td>9932</td>
<td>GapA: Glyceraldehyde-3-phosphate dehydrogenase/erythrose</td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>8</td>
<td>13849</td>
<td>COG4708: Predicted membrane protein</td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>7</td>
<td>10199</td>
<td>COG0325: Predicted enzyme with a TIM-barrel fold</td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>9</td>
<td>10207</td>
<td>RpmF: Ribosomal protein L32</td>
<td>L32 r-protein leader</td>
</tr>
<tr>
<td>187</td>
<td>11</td>
<td>27850</td>
<td>LDH: L-lactate dehydrogenases</td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>11</td>
<td>10094</td>
<td>CspR: Predicted rRNA methylase</td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>9</td>
<td>10353</td>
<td>FusA: Translation elongation factors</td>
<td>EF-G r-protein leader</td>
</tr>
</tbody>
</table>
**A** L19 (rplS) mRNA leader

<table>
<thead>
<tr>
<th>RBS</th>
<th>Start</th>
<th>Leader</th>
<th>TSS</th>
<th>P1</th>
<th>P2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Up</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example: Ribosomal Autoregulation**

Excess L19 represses L19 (RF00556; 555-559 similar)

**B**

**Compensatory and Compatible Mutations**

<table>
<thead>
<tr>
<th>Nucleotide</th>
<th>Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>97%</td>
</tr>
<tr>
<td>N</td>
<td>90%</td>
</tr>
<tr>
<td>N</td>
<td>75%</td>
</tr>
</tbody>
</table>

- **G - C** Watson-Crick base pair
- **G • A** other base interaction

**C**

B. subtilis L19 mRNA leader
Estimating Motif Significance

Red: top 100 motifs.
Black: 50 permutations of ClustalW alignment of each of those input sets

This likely underestimates significance, but nevertheless all real motifs have p < .01, and 73/100 better than all perms of their own input set
Examples: 6 (of 22) Representative motifs


MoCo

sucA

COG4708

boxed = confirmed riboswitch

Vertebrate ncRNAs

Some Results
Human Predictions

EvoFold
48,479 candidates (~70% FDR?)

RNAz
S Washietl, IL Hofacker, IL Lukasser, A Hutenhofer, PF Stadler, "Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome." 
30,000 structured RNA elements
1,000 conserved across all vertebrates.
~1/3 in introns of known genes, ~1/6 in UTRs
~1/2 located far from any known gene

FOLDALIGN
E Torarinsson, M Siver, JH Havgaard, M Fredholm, J Gorodkin, "Thousands of corresponding human and mouse genomic regions unalignable in primary sequence contain common RNA structure." 
1800 candidates from 36970 (of 100,000) pairs

CMfinder
Torarinsson, Yao, Wiklund, Bramsen, Hansen, Kjems, Tommerup, Ruzzo and Gorodkin. 
Comparative genomics beyond sequence based alignments: RNA structures in the ENCODE regions. 
6500 candidates in ENCODE alone (better FDR, but still high)

Some details below
CMfinder Search in Vertebrates

Extract ENCODE* Multiz alignments
   Remove exons, most conserved elements.
   56017 blocks, 8.7M bps.
Apply CMfinder to both strands.
10,106 predictions, 6,587 clusters.
   High false positive rate, but still suggests 1000’s of RNAs.

(We’ve applied CMfinder to whole human genome:
   many 100’s of CPU years. Analysis in progress.)

* ENCODE: deeply annotated 1% of human genome
These observations are further quantified below.

The overrepresentation of long ungapped segments (Figure 1B) is caused by indel events thought to evolve predominantly neutrally only alignments of ARs (see Materials and Methods). These purifying selection, a similar histogram was constructed using ungapped segments shorter than 20 bp were ignored.

This phenomenon does not have a single alignment gap when this more parsimoniously gap attraction [14], by which two nearby indel events give rise to a similar fit in the range 20–50 bp. The resulting histograms markably close fit to the geometric distribution (a straight line in log-linear coordinates) within the length range 20–50 bp, with the model explaining 99.996% of the variance.

Outside of the range of 20–50 bp, histogram counts deviate from the neutral model (log$_10$ scale). Parameters were obtained by simultaneously introducing co-dependencies in the survival probabilities of nearby sites. In other words, the probability that an ancestral nucleotide survives as a homologous nucleotides surviving in between, follows a geometric distribution. Note that this conclusion holds irrespective of the distribution of indel lengths themselves, because of the assumption of independence. In this paper, we refer to the probability of an individual nucleotide in two descendan

The underrepresentation of short intergap distances is caused by a systematic alignment artefact termed gap attraction [14], by which two nearby indel events give rise to a similarly tight fit in the range 20–80 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond 300 is expected genome-wide under the neutral model; however the whole-genome histogram shows a similarly tight fit in the range 20–50 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond 300 is expected genome-wide under the neutral model; however the whole-genome histogram shows a similarly tight fit in the range 20–80 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond 300 is expected genome-wide under the neutral model; however the whole-genome histogram shows a similarly tight fit in the range 20–80 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond 300 is expected genome-wide under the neutral model; however the whole-genome histogram shows a similarly tight fit in the range 20–80 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond 300 is expected genome-wide under the neutral model; however the whole-genome histogram shows a similarly tight fit in the range 20–80 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond 300 is expected genome-wide under the neutral model; however the whole-genome histogram shows a similarly tight fit in the range 20–80 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond 300 is expected genome-wide under the neutral model; however the whole-genome histogram shows a similarly tight fit in the range 20–80 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond 300 is expected genome-wide under the neutral model; however the whole-genome histogram shows a similarly tight fit in the range 20–80 bp, and a large excess of long intergap distances over neutral model predictions (green) beyond
Overlap w/ Indel Purified Segments

IPS presumed to signal purifying selection
Majority (64%) of candidates have >45% G+C
Strong P-value for their overlap w/ IPS

<table>
<thead>
<tr>
<th>G+C</th>
<th>data</th>
<th>P</th>
<th>N</th>
<th>Expected</th>
<th>Observed</th>
<th>P-value</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-35</td>
<td>igs</td>
<td>0.062</td>
<td>380</td>
<td>23</td>
<td>24.5</td>
<td>0.430</td>
<td>5.8%</td>
</tr>
<tr>
<td>35-40</td>
<td>igs</td>
<td>0.082</td>
<td>742</td>
<td>61</td>
<td>70.5</td>
<td>0.103</td>
<td>11.3%</td>
</tr>
<tr>
<td>40-45</td>
<td>igs</td>
<td>0.082</td>
<td>1216</td>
<td>99</td>
<td>129.5</td>
<td>0.00079</td>
<td>18.5%</td>
</tr>
<tr>
<td>45-50</td>
<td>igs</td>
<td>0.079</td>
<td>1377</td>
<td>109</td>
<td>162.5</td>
<td>5.16E-08</td>
<td>20.9%</td>
</tr>
<tr>
<td>50-100</td>
<td>igs</td>
<td>0.070</td>
<td>2866</td>
<td>200</td>
<td>358.5</td>
<td>2.70E-31</td>
<td>43.5%</td>
</tr>
<tr>
<td>all</td>
<td>igs</td>
<td>0.075</td>
<td>6581</td>
<td>491</td>
<td>747.5</td>
<td>1.54E-33</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
Realignment

Average pairwise sequence similarity

% realigned

216
The original MULTIZ alignment without flanking regions. [RNAz Score: 0.132 (no RNA)]

**Human**
GGTCACTTCAAAGAGGGCTT--GTGGGGCTGTGAAACCAAGAGG--CTTAACAGTATGACCAAAAACCTGAAAGT

**Chimp**
GGACATTTCAATGCGGGCTC--ATGGGGCTGTGAAACCAAGAGG--CTTAACACTATGACCAAGGACTGAATAT

**Cow**
GGTCATTTCAGAGGGCTTTGCTGAAACCAAGAGG--CTTAACACTATGACCAAGGACTGAATAT

**Dog**
GGTCATTTCAGAGGGCTTTGCTGAAACCAAGAGG--CTTAACACTATGACCAAGGACTGAATAT

**Rabbit**
GATCATTTCAATGCGGGCTC--ATGGGGCTGTGAAACCAAGAGG--CTTAACACTATGACCAAGGACTGAATAT

**Rhesus**
GGTCACTTCAAAGAGGGCTT--GTGGGGCTGTGAAACCAAGAGG--CTTAACACTATGACCAAGGACTGAATAT

**Str**
(((((((.................................))))))(((((.................................))))))

The local CMfinder re-alignment of the MULTIZ block. [RNAz Score: 0.709 (RNA)]

**Human**
GGTCACTTCAAAGAGGGCTT--GTGGGGCTGTGAAACCAAGAGG--AGAGTCTTAACAGTATGACCAAAAACCTGAAAGT

**Chimp**
GGACATTTCAATGCGGGCTC--ATGGGGCTGTGAAACCAAGAGG--AGAGCTATTAACACTATGACCAAGGACTGAATAT

**Cow**
GGTCATTTCAGAGGGCTTTGCTGAAACCAAGAGG--AGAGCTATTAACACTATGACCAAGGACTGAATAT

**Dog**
GGTCATTTCAGAGGGCTTTGCTGAAACCAAGAGG--AGAGCTATTAACACTATGACCAAGGACTGAATAT

**Rabbit**
GATCATTTCAATGCGGGCTC--ATGGGGCTGTGAAACCAAGAGG--AGAGCTATTAACACTATGACCAAGGACTGAATAT

**Rhesus**
GGTCACTTCAAAGAGGGCTT--GTGGGGCTGTGAAACCAAGAGG--AGAGCTATTAACACTATGACCAAGGACTGAATAT

**Str**
(((((((.................................))))))(((((.................................))))))

218
10 of 11 top (differentially) expressed
Open Problems - Better CM’s

Optional- and variable-length stems
Riboswitches & other regulatory RNAs often switch between conformations; better search & alignment exploiting both alternatives?
“Augmented” CM handling pseudoknots probably too slow for scan, but plausibly could be used for alignment
Better use of prior knowledge? (GNRA tetraloops, single-stranded A’s…)
Open Problems - Better algorithms & scoring

incorporating phylogeny in model construction & scoring
  e.g. “mutual information” ignores it
improve scoring by “shuffling”
other ideas for scan filtering
comparing & clustering RNA structures
search/alignment/inference with splicing
Open Problems - Applications & Biology

clustering intergenic sequences, esp prokaryotic
systematic look at eukaryotic UTRs
  how to cluster? how to score?
  “swiss-cheese phylogenies”
evidence for selection (no dN/dS)
ncRNA Summary

ncRNA is a “hot” topic
For family homology modeling: CMs
Training & search like HMM (but slower)
Dramatic acceleration possible
Automated model construction possible
New computational methods yield new discoveries
Many open problems