RNA Search and Motif Discovery

Lectures 17-19
CSE 527
Autumn 2006
The Human Parts List, circa 2001

3 billion nucleotides, containing:

• 25,000 protein-coding genes
  (only ~1% of the DNA)

• Messenger RNAs made from each

• Plus a double-handful of other RNA genes
Noncoding RNAs

Dramatic discoveries in last 5 years

100s of new families

Many roles: Regulation, transport, stability, catalysis, …

1% of DNA codes for protein, but 30% of it is copied into RNA, i.e. ncRNA >> mRNA
Outline

Task 1: RNA 2\textsuperscript{ary} Structure Prediction (last time)

Task 2: RNA Motif Models
   Covariance Models
   Training & “Mutual Information”

Task 3: Search
   Rigorous & heuristic filtering

Task 4: Motif discovery
Task 2: Motif Description
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

AACAAAGccggccaggcuuuuAGUF
GAAUAUCUuuugggauu......AGUF
GAA..CA.................AGUF
GAAUAUCUuuaugauu......AGUF

mostly G del ins all G
How to model an RNA “Motif”?

Add “column pairs” and pair emission probabilities for base-paired regions.
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 sloooow
“RNA sequence analysis using covariance models”

Eddy & Durbin
Nucleic Acids Research, 1994
vol 22 #11, 2079-2088
(see also, Ch 10 of Durbin et al.)
What

A probabilistic model for RNA families
The “Covariance Model” ≈ 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
### Alignment Quality

#### Trusted:

| DF6280 | GCGGAUUUAGUCAGUU | GGG | AGAGCAGCAGACUGAAG | AUCUGGAG | GUCUCUGUUUCGAAUCGACAGAAUUCGACCA |
|        | GCGGAUUUAGUCAGUU | GGG | AGAGCAGCAGACUGAAGAAUAUCUUCGUCACAGUUUACUGGAG | GUCUCUGUUUCGAAUCGACAGAAUUCGACCA |
| DD6280 | UCCUGAAGUUGUUUAAU | GUGGCAAUUGGGCAGCACGUCUCUGG | CUGCCAG | A UCCGGGUUCAAUUCGCCCUGCCGAGCGCA |
| DX1661 | CGCGGGGUGGAGCACCGUUGGUAGCGGAGUCAUA | ACCCGAA | GUCUGCGGUUCAAUUCGCCCAGACCA |
| DS6280 | GGCACACUUGGGCGAGUCUUAAGC | AGAAGAGA | CCAGGAGAAGAUAAGAAA | GUCUCUGUUUCGAAUCGACAGAAUUCGACCA |

#### U100:

| DF6280 | GCGGAUUUAGUCAGUU | UGGGAAGAGCCCAAGAUC | GA | AG | AUCUGGAA | GUCUCUGUUUCGAAUCGACAGAAUUCGACCA |
|        | GCGGAUUUAGUCAGUU | UGGGAAGAGCCCAAGAUC | GA | AG | AUCUGGAA | GUCUCUGUUUCGAAUCGACAGAAUUCGACCA |
| DD6280 | UCCUGAAGUUGUUUAAU | UGGGCAAGAUAGGGCAGCUU | GU | CG | CGUGCCA | GAU CGGGGCUCAUUCGCCCUGGGAGCC |
| DX1661 | CGCGGGGUGGAGCACCGUUGGUAGCGGAGUCAUA | CA | UA | ACCCGAA | GUCUGCGGUUCAAUUCGCCCAGACCA |
| DS6280 | GGCACACUUGGGCGAGUCUUAAGC | UGGGAAGAGCCCAAGAUC | GA | AG | AUCUGGAA | GUCUCUGUUUCGAAUCGACAGAAUUCGACCA |

#### ClustalV:

| DF6280 | GCGGAUUUAGUCAGUU | UGGGAAGAGCCCAAGAUC | GA | AG | AUCUGGAA | UCCGGAGGCGUGGUUCGGAUCCCAACAGAAUUCGCA |
|        | GCGGAUUUAGUCAGUU | UGGGAAGAGCCCAAGAUC | GA | AG | AUCUGGAA | UCCGGAGGCGUGGUUCGGAUCCCAACAGAAUUCGCA |
| DD6280 | UCCUGAAGUUGUUUAAU | GUGGGAGAGCCGAGCAGUGAAGAAUAUCUUCGUCACAGUUUACUGGAG | GUGGCGGUGGUUCGGAUCCCAACAGAAUUCGCA |
| DX1661 | CGCGGGGUGGAGCACCGUUGGUAGCGGAGUCAUA | ACCCGAA | UCCGGAGGCGUGGUUCGGAUCCCAACAGAAUUCGCA |
| DS6280 | GGCACACUUGGGCGAGUCUUAAGC | UGGGAAGAGCCCAAGAUC | GA | AG | AUCUGGAA | UCCGGAGGCGUGGUUCGGAUCCCAACAGAAUUCGCA |
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.
**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)
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)

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

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

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

Time \(O(qn^3)\), \(q\) states, seq len \(n\)
compare: \(O(qn)\) for profile HMM
Model Training

unaligned sequences

random alignment

multiple alignment

(EM)
alignment
parameter reestimation

covariance model

model construction
(structure prediction)
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
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)**

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.
MI-Based Structure-Learning

Find best (max total MI) subset of column pairs among $i \ldots j$, 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
Pseudoknots

disallowed   allowed

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Avg.</th>
<th>Min</th>
<th>Max</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>

Table 1: Statistics of the training and test sets of 100 tRNA sequences each. The average identity in an alignment is the average pairwise identity of all aligned symbol pairs, with gap/symbol alignments counted as mismatches. Primary sequence information content is calculated according to [48]. Calculating pairwise mutual information content is an NP-complete problem of finding an optimum partition of columns into pairs. A lower bound is calculated by using the model construction procedure to find an optimal partition subject to a non-pseudoknotting restriction. An upper bound is calculated as sum of the single best pairwise covariation for each position, divided by two; this includes all pairwise tertiary interactions but overcounts because it does not guarantee a disjoint set of pairs. For the meaning of multiple alignment accuracy of ClustalV, see the text.
<table>
<thead>
<tr>
<th>Model</th>
<th>training set</th>
<th>iterations</th>
<th>score (bits)</th>
<th>alignment accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1415</td>
<td>all sequences (aligned)</td>
<td>3</td>
<td>58.7</td>
<td>95%</td>
</tr>
<tr>
<td>A100</td>
<td>SIM100 (aligned)</td>
<td>3</td>
<td>57.3</td>
<td>94%</td>
</tr>
<tr>
<td>A65</td>
<td>SIM65 (aligned)</td>
<td>3</td>
<td>46.7</td>
<td>93%</td>
</tr>
<tr>
<td>U100</td>
<td>SIM100 (degapped)</td>
<td>23</td>
<td>56.7</td>
<td>90%</td>
</tr>
<tr>
<td>U65</td>
<td>SIM65 (degapped)</td>
<td>29</td>
<td>47.2</td>
<td>91%</td>
</tr>
</tbody>
</table>

Table 2: Training and multiple alignment results from models trained from the trusted alignments (A models) and models trained from no prior knowledge of tRNA (U models).
tRNASEScanSE

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
Rfam – an RNA family DB
Griffiths-Jones, et al., NAR ‘03, ’05

Biggest scientific computing 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, >300k instances
Input (hand-curated):
- MSA “seed alignment”
- SS_cons
- Score Thresh T
- Window Len W

Output:
- CM
- scan results & “full alignment”

**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>UUUCUUC.UUCAACAGUGUUUGGAUGGAAC</td>
</tr>
<tr>
<td>Hom. sap.</td>
<td>UUUCUGUUUCAACAGUGUUUGGAUGGAAC</td>
</tr>
<tr>
<td>Hom. sap.</td>
<td>UUUAUC..AGUGACAGAGUUCACU.AUAAA</td>
</tr>
<tr>
<td>Hom. sap.</td>
<td>UCUCUUGCUUCAACAGUGUUUGGAUGGAAC</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..GGAGACAGUGAUCUCC.AUAUG</td>
</tr>
<tr>
<td>Hom. sap.</td>
<td>AUUAUC..GGAGACAGUGCCUCCC.AUAAU</td>
</tr>
<tr>
<td>Cav. por.</td>
<td>UCUCCUGCUUCAACAGUGCUUGGACGGAAC</td>
</tr>
<tr>
<td>Mus. mus.</td>
<td>UUAUAC..GGAGACAGUGAUCUCC.AUAUG</td>
</tr>
<tr>
<td>Mus. mus.</td>
<td>UUCCUGCUUCAACAGUGCUUGAACGGAAC</td>
</tr>
<tr>
<td>Mus. mus.</td>
<td>GUACUUGCUUCAACAGUGUUUGAACGGAAC</td>
</tr>
<tr>
<td>Rat. nor.</td>
<td>UUAUAC..GGAGACAGUGACCUCUCC.AUAUG</td>
</tr>
<tr>
<td>Rat. nor.</td>
<td>UAUCUUGCUUCAACAGUGUUUGGACGGAAC</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>
Figure 2. Taxonomic distribution of Rfam family members in the three kingdoms of life.
Rfam – key issues

Overly narrow families
Variant structures/unstructured RNAs
Spliced RNAs
RNA pseudogenes
  Human ALU is SRP-related, with $1.1 \times 10^6$ copies
  Mouse B2 repeat (350k copies) tRNA related
Speed & sensitivity
Motif discovery
Task 3: Faster Search
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 will score < threshold;
- actually run CM on the rest (the promising parts)
- assignment of HMM transition/emission scores is key
  (large convex optimization problem)

Weinberg & Ruzzo, Bioinformatics, 2004, 2006
Covariance Model

Key difference of CM vs HMM: Pair states emit paired symbols, corresponding to base-paired nucleotides; 16 emission probabilities here.
CM’s are good, but slow

Rfam Reality

EMBL \rightarrow BLAST \rightarrow CM \rightarrow junk, hits
1 month, 1000 computers

Our Work

EMBL \rightarrow Ravenna \rightarrow CM \rightarrow junk, hits
\sim 2 months, 1000 computers

Rfam Goal

EMBL \rightarrow CM \rightarrow junk, hits
10 years, 1000 computers
Simplified CM
(for pedagogical purposes only)
CM to HMM

25 emissions per state

5 emissions per state, 2x states
Key Issue: 25 scores $\rightarrow$ 10

Need: log Viterbi scores $\text{CM} \leq \text{HMM}$
Viterbi/Forward Scoring

Path $\pi$ defines transitions/emissions
Score($\pi$) = product of “probabilities” on $\pi$
NB: ok if “probs” aren’t, e.g. $\Sigma \neq 1$
(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($x$) = $\Sigma${ score($\pi$) | $\pi$ emits $x$ }
Key Issue: 25 scores $\rightarrow$ 10

Need: log Viterbi scores $CM \leq HMM$

- $P_{AA} \leq L_A + R_A$
- $P_{AC} \leq L_A + R_C$
- $P_{AG} \leq L_A + R_G$
- $P_{AU} \leq L_A + R_U$
- $P_{A-} \leq L_A + R_-$
- $P_{CA} \leq L_C + R_A$
- $P_{CC} \leq L_C + R_C$
- $P_{CG} \leq L_C + R_G$
- $P_{CU} \leq L_C + R_U$
- $P_{C-} \leq L_C + R_-$

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)
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 \( \approx 25\% \)

<table>
<thead>
<tr>
<th>Opt 1:</th>
<th>Opt 2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ L_U + (R_A + R_G)/2 = 4 ]</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**

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

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 ↓ $\Rightarrow$ Viterbi ↓ $\Rightarrow$ filter ↓”)

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

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

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:
\[
\begin{align*}
  f_k(i) &= P(x_1 \ldots x_i, \pi_i = k) \\
  f_i(i + 1) &= e_i(x_{i+1}) \sum_k f_k(i) a_{k,l}
\end{align*}
\]

Viterbi:
\[
\begin{align*}
  v_i(i + 1) &= e_i(x_{i+1}) \cdot \max_k (v_k(i) a_{k,l})
\end{align*}
\]

\[
\frac{\partial E(L_1, L_2, \ldots)}{\partial L_i}
\]
## 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
## Results: New ncRNA’s?

<table>
<thead>
<tr>
<th>Name</th>
<th># found BLAST + CM</th>
<th># found rigorous filter + CM</th>
<th># new</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Pyrococcus</em> snoRNA</td>
<td>57</td>
<td>180</td>
<td>123</td>
</tr>
<tr>
<td>Iron response element</td>
<td>201</td>
<td>322</td>
<td>121</td>
</tr>
<tr>
<td>Histone 3’ element</td>
<td>1004</td>
<td>1106</td>
<td>102</td>
</tr>
<tr>
<td>Purine riboswitch</td>
<td>69</td>
<td>123</td>
<td>54</td>
</tr>
<tr>
<td>Retron msr</td>
<td>11</td>
<td>59</td>
<td>48</td>
</tr>
<tr>
<td>Hammerhead I</td>
<td>167</td>
<td>193</td>
<td>26</td>
</tr>
<tr>
<td>Hammerhead III</td>
<td>251</td>
<td>264</td>
<td>13</td>
</tr>
<tr>
<td>U4 snRNA</td>
<td>283</td>
<td>290</td>
<td>7</td>
</tr>
<tr>
<td>S-box</td>
<td>128</td>
<td>131</td>
<td>3</td>
</tr>
<tr>
<td>U6 snRNA</td>
<td>1462</td>
<td>1464</td>
<td>2</td>
</tr>
<tr>
<td>U5 snRNA</td>
<td>199</td>
<td>200</td>
<td>1</td>
</tr>
<tr>
<td>U7 snRNA</td>
<td>312</td>
<td>313</td>
<td>1</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 binary structure info.

- They work well because, though 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)
Fig. 2. Filter creation and selection. Filters for Rfam tRNA (RF00005) generated by the store-pair and sub-CM techniques and those selected for actual filtering are plotted by filtering fraction and run time. The CM runs at 3.5 secs/kbase. The four selected filters are run one after another, from highest to lowest fraction.
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
“ML heuristic”: train HMM from the infinite alignment generated by the CM
Often 10x faster, modest loss in sensitivity
Heuristic Filters

* rigorous HMM, not rigorous threshold

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
Task 4: Motif Discovery
RNA Motif Discovery

Typical problem: given a ~10-20 unaligned sequences of ~1kb, most of which contain instances of one RNA motif of, say, 150bp -- find it.

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

Align sequences, then look for common structure

Predict structures, then try to align them

Do both together
“Obvious” Approach I: Align First, Predict from Multiple Sequence Alignment

Compensatory mutations reveal structure, (core of “comparative sequence analysis”) but usual alignment algorithms penalize them (twice)
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 sequences, then look for common structure

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

Do both together
  - Sankoff – good but slow
  - Heuristic
“Obvious” Approach II: Fold First

Predict secondary RNA structure using MFOLD or Vienna

Problems
- false folding predictions
- comparing structures
Our Approach: CMfinder

Simultaneous alignment, folding and CM-based motif description using an EM-style learning procedure

Yao, Weinberg & Ruzzo, *Bioinformatics*, 2006
Cmfinder--A Covariance Model Based RNA Motif Finding Algorithm

*Bioinformatics, 2006, 22(4): 445-452*

Zizhen Yao
Zasha Weinberg
Walter L. Ruzzo

University of Washington, Seattle
Design Goals

Find RNA motifs in unaligned sequences
Seq conservation exploited, but not required
Robust to inclusion of unrelated sequences
Robust to inclusion of flanking sequence
Reasonably fast and scalable
Produce a probabilistic model of the motif that can be directly used for homolog search
Alignment $\rightarrow$ CM $\rightarrow$ Alignment

Similar to HMM, but much slower
Builds on Eddy & Durbin, ‘94
But new way to infer which columns to pair, via a principled combination of mutual information and predicted folding energy
And, it’s local, not global, alignment (harder)
CMfinder Outline

M-step uses M.I. + folding energy for structure prediction
Initial Alignment Heuristics

fold sequences separately

candidates: regions with low folding energy

compare candidates via “tree edit” algorithm

find best “central” candidates & align to them

BLAST anchors
\( L_i = \text{column } i; \ \sigma = (\alpha, \beta) \) the 2\textsuperscript{ary} struct, \( \alpha = \text{unpaired}, \ \beta = \text{paired cols} \)

Our goal is to find \( \hat{\sigma} = \arg \max_{\sigma} P(D, \sigma) \). Assuming independence of non-base paired columns, then

\[
P(D|\sigma) = \prod_{k \in \alpha} P(L_k) \prod_{(i,j) \in \beta} P(L_i L_j) \tag{2}
\]

\[
= \prod_{1 \leq k \leq l} P(L_k) \prod_{(i,j) \in \beta} \frac{P(L_i L_j)}{P(L_i)P(L_j)} \tag{3}
\]

Let

\[
I_{ij} = \log \frac{P(L_i L_j)}{P(L_i)P(L_j)}
\]

With MLE params, \( I_{ij} \) is the \textit{mutual information} between cols \( i \) and \( j \)
Let \( s_i \) be the prior for column \( i \) to be single stranded, and \( p_{ij} \) the prior for columns \( i, j \) to be base paired, then

\[
P(\sigma) = \prod_{k \in \alpha} s_k \prod_{(i, j) \in \beta} p_{ij}, \text{ and } P(D, \sigma) \text{ can be rewritten as}
\]

\[
P(D, \sigma) = P(D|\sigma)P(\sigma)
\]

\[
= \prod_{1 \leq k \leq l} P(L_k)s_k \prod_{(i, j) \in \beta} \frac{P(L_iL_j)}{P(L_i)P(L_j)} \frac{p_{ij}}{s_is_j}
\]

(4)

Let

\[
K_{ij} = \log \left( \frac{P(L_iL_j)}{P(L_i)P(L_j)} \frac{p_{ij}}{s_is_j} \right) = I_{ij} + \log \frac{p_{ij}}{s_is_j},
\]

then the maximum likelihood structure \( \sigma \) maximizes \( \sum_{(i, j) \in \beta} K_{ij} \). Can find it via a simple dynamic programming alg.
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>0</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>0</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
Task 5: Application

Genome-wide search for cis-regulatory RNA elements (in prokaryotes, initially)
Searching for noncoding RNAs

CM’s are great, but where do they come from?
An 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.
Predicting New cis-Regulatory RNA Elements

Goal:
Given unaligned UTRs of coexpressed or orthologous genes, find common structural motifs

Difficulties:
Low sequence similarity: alignment difficult
Varying flanking sequence
Motif missing from some input genes
Approach

Choose a bacterial genome
For each gene, get 10-30 close orthologs (CDD)
Find most promising genes, based on conserved sequence motifs (Footprinter)
From those, find structural motifs (CMfinder)
Genome-wide search for more instances (Ravenna)
Expert analyses (Breaker Lab, Yale)
A pipeline for RNA motif genome scans

Bacillus subtilis genes

- BLAST/CDD
  - Orthologous genes
    - Upstream sequences
  - Top datasets
    - Footprinter Rank datasets

- CMfinder
  - Motifs
    - Homologs
  - Search Genome database
Genome Scale Search: Why

Most riboswitches, e.g., are present in ~5 copies per genome
Throughout (most of) clade
More examples give better model, hence even more examples, fewer errors
More examples give more clues to function - critical for wet lab verification
Genome Scale Search

CMfinder is directly usable for/with search

- Folding predictions
- Smart heuristics
- Candidate alignment
- CM
- Search
- Realign
Footprinter finds patterns of conservation

Upstream of folC
A blind test

1ST genome scan: 234 sequences
2ND genome scan: 447 sequences

The motif turned out to be T box

Match to RFAM T box family: 299 OF 342
False Positives: 89/148 are probable (upstream of annotated tRNA-synthetase genes)

tyR S T box structure
- CMfinder: 9 instances
- Found by Scan: 447 hits
Results

Process largely complete in
bacillus/clostridia
gamma proteobacteria
cyanobacteria
actinobacteria
firmicutes

Analysis ongoing
Some Preliminary Actino Results
8 of 10 Rfam families found

<table>
<thead>
<tr>
<th>Rfam Family</th>
<th>Type (metabolite)</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>THI</td>
<td>riboswitch (thiamine)</td>
<td>4</td>
</tr>
<tr>
<td>ydaO-yuaA</td>
<td>riboswitch (unknown)</td>
<td>19</td>
</tr>
<tr>
<td>Cobalamin</td>
<td>riboswitch (cobalamin)</td>
<td>21</td>
</tr>
<tr>
<td>SRP_bact</td>
<td>gene</td>
<td>28</td>
</tr>
<tr>
<td>RFN</td>
<td>riboswitch (FMN)</td>
<td>39</td>
</tr>
<tr>
<td>yybP-ykoY</td>
<td>riboswitch (unknown)</td>
<td>48</td>
</tr>
<tr>
<td>gcvT</td>
<td>riboswitch (glycine)</td>
<td>53</td>
</tr>
<tr>
<td>S_box</td>
<td>riboswitch (SAM)</td>
<td>401</td>
</tr>
<tr>
<td>tmRNA</td>
<td>gene</td>
<td>Not found</td>
</tr>
<tr>
<td>RNaseP</td>
<td>gene</td>
<td>Not found</td>
</tr>
</tbody>
</table>

not cis-regulatory (got one anyway)
Preliminary results of genome scan

Top 115 datasets (some are redundant)
13 T box, 22 riboswitches, 30 ribosomal genes
RNase P, tRNA, CIRCE elements and other DNA binding sites

<table>
<thead>
<tr>
<th>Gene</th>
<th>#motif</th>
<th>hits</th>
<th>RFAM</th>
<th>#seed</th>
<th>#full</th>
<th>#TP</th>
<th>specificity</th>
<th>sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>metK</td>
<td>13</td>
<td>150</td>
<td>S_box</td>
<td>71</td>
<td>151</td>
<td>145</td>
<td>0.967</td>
<td>0.960</td>
</tr>
<tr>
<td>ribB</td>
<td>9</td>
<td>106</td>
<td>RFN</td>
<td>48</td>
<td>114</td>
<td>97</td>
<td>0.915</td>
<td>0.851</td>
</tr>
<tr>
<td>folC</td>
<td>9</td>
<td>447</td>
<td>T_box</td>
<td>67</td>
<td>342</td>
<td>299</td>
<td>0.669</td>
<td>0.874</td>
</tr>
<tr>
<td>xpt</td>
<td>14</td>
<td>106</td>
<td>Purine</td>
<td>37</td>
<td>100</td>
<td>97</td>
<td>0.915</td>
<td>0.970</td>
</tr>
<tr>
<td>glmS</td>
<td>16</td>
<td>33</td>
<td>glmS</td>
<td>14</td>
<td>37</td>
<td>33</td>
<td>1.000</td>
<td>0.892</td>
</tr>
<tr>
<td>thiA</td>
<td>16</td>
<td>305</td>
<td>THI</td>
<td>237</td>
<td>366</td>
<td>305</td>
<td>1.000</td>
<td>0.833</td>
</tr>
<tr>
<td>ykoY</td>
<td>10</td>
<td>34</td>
<td>yybP-ykoY</td>
<td>74</td>
<td>127</td>
<td>33</td>
<td>0.971</td>
<td>0.260</td>
</tr>
</tbody>
</table>
More Prelim Actino Results

Many others (not in Rfam) are likely real; of top 50:

- known (Rfam, 23S) 10
- probable (Tbox, CIRCE, LexA, parP, pyrR) 7
- probable (ribosomal genes) 9
- potentially interesting 12
- unknown or poor 12

One bench-verified, 3-4 more in progress
<table>
<thead>
<tr>
<th>mRNA leader switch?</th>
</tr>
</thead>
</table>

### A: mRNA leader

<table>
<thead>
<tr>
<th>m</th>
<th>n</th>
<th>R</th>
<th>a</th>
<th>n</th>
<th>a</th>
<th>l</th>
<th>e</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>mRNA</td>
<td>leader</td>
<td>switch?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### B: mRNA leader switch?

<table>
<thead>
<tr>
<th>nucleotide identity</th>
<th>nucleotide present</th>
</tr>
</thead>
<tbody>
<tr>
<td>N 97%</td>
<td>97%</td>
</tr>
<tr>
<td>N 90%</td>
<td>90%</td>
</tr>
<tr>
<td>N 75%</td>
<td>75%</td>
</tr>
<tr>
<td>N 50%</td>
<td>50%</td>
</tr>
</tbody>
</table>

### C: mRNA leader switch?

- **P1**: Watson-Crick base pair
- **P2**: Other base interaction
- **G • C**: Watson-Crick base pair
- **G • A**: Other base interaction

**compensatory mutations**: stem loop always present

**compatible mutations**:
Ongoing & Future Work

Still automating a few steps, e.g. identifying duplicates
Improved ranking/motif significance stats
Better ortholog clustering
Performance & scale-up

Eukaryotic mRNAs, e.g. UTRs
Summary

ncRNA - apparently widespread, much interest

Covariance Models - powerful but expensive tool for ncRNA motif representation, search, discovery

Rigorous/Heuristic filtering - typically 100x speedup in search with no/little loss in accuracy

CMfinder - CM-based motif discovery in unaligned sequences