Outline

Whirlwind tour of ncRNA search & discovery
  Covariance Model Review
  Algorithms for Training
    “Mutual Information”
  Algorithms for searching
    Rigorous & heuristic filtering
  Motif discovery

Wrap up
Course Evals
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
“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
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   GCGGAUUUGACUGAGU   GGG   AGAGCGCCAGACGUGAAG   AUCUGGA   GUCUCCUGUGUUCGAAUCAGAAGAUAUGCACCA
DF6280G  GCGGAUUUGACUGAGU   GGG   AGAGCGCCAGACGUGAAGAUAACUUCGCGUCAAGUUAUCUGGAG   GUCUCCUGUGUUCGAAUCAGAAGAUAUGCACCA
DD6280   CCUGAGAUGUUAUAAU   GUACGAAAGAUGCCGCGUCUGG   CGUCCAG   AGUCCGGUCAUCCCCGUCGCGGAGCA
DX1661   CCGCGGUGGAGGACGCCGAGAU   GACUCGCGGAGGGCGUCAUA   ACCCGAAG   GUCUCGCGGUGCAAAUCCGCCCGGCAACCA
DS6280   GGCACACUUGGCGGAGU   GGUUAAGGCGAAAGAUAGAA   AUCUUUU   GGCUUUGCCCG   CGCAGGUUCGAGUCCGCAUUGUGGCACCA
```

U100:

```
DF6280   GCGGAUUUGACUGAGU   UGGGAAGAGCGCGCAAGCA   GA   AG   AUCUGGA   GUCUCCUGUGUUCGAAUCAGAAGAUAUGCACCA
DF6280G  GCGGAUUUGACUGAGU   UGGGAAGAGCGCGCAAGCA   UCaUCCGCAAGAAGAUAUCGCUAAGUUAUCUGGAG   GUCUCCUGUGUUCGAAUCAGAAGAUAUGCACCA
DD6280   CCUGAGAUGUUAUAAU   UGGAGAAUGCCGCGAGU   GU   CG   CGUGCAG   GAU   CGGCUUCAUCCCCGUCGCGGAGCA
DX1661   CCGCGGUGGAGGACGCCGAGAU   CUGGUAGCGUUCGGGCU   CA   UA   ACCCGAAG   GUCUCGCGGUGCAAAUCCGCCCGGCAACCA
DS6280   GGCACACUUGGCGGAGU   UGGGUAAAGCGCAGAAGAUU   AG   AA   AUCUUUUUg gccu uu gu ccccG   CGAAGGGUCCGAGUCCGCAUUGUGGCACCA
```

ClustalV:

```
DF6280   GCGGAUUUGACUGAGU   UGGGAAGAGCGCGCAAGCA   GACUCGCGGAGGGCGUCAUA   UCCGAGGUGCUGGUUCGAAUCAGAAGAUAUGCACCA
DF6280G  GCGGAUUUGACUGAGU   UGGGAAGAGCGCGCAAGCA   UCCAUCGUGUAUGCGGUGGAAUGGCGCCAAGAAGAUAACUUCGCGUCAAGUUAUCUGGAG   GUCUCCUGUGUUCGAAUCAGAAGAUAUGCACCA
DD6280   CCUGAGAUGUUAUAAU   GUGUCCUGGUCC   CUUG   UCCGUGCC   AGAUGCAGGCGGCAGAUGCACCA
DX1661   CCGCGGUGGAGGACGCCGAGAU   CUGGUAGCGUUCGGGCU   CA   UA   ACCCGAAG   GUCUCGCGGUGCAAAUCCGCCCGGCAACCA
DS6280   GGCACACUUGGCGGAGU   UGGGUAAAGCGCAGAAGAUU   AGAUAUCUUGGCG   UCUGGCCCG   CGCAGGUUCGAGUCCGCAUUGUGGCACCA
```
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.

M_j: Match states (20 emission probabilities)
I_j: Insert states (Background emission probabilities)
D_j: 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

\( x_i = \text{ith letter of input} \)

\( x_{ij} = \text{substring i,...,j of input} \)

\( T_{yz} = P(\text{transition } y \rightarrow z) \)

\( E_{x_i,x_j}^y = P(\text{emission of } x_i,x_j \text{ from state } y) \)

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

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

Time \( O(qn^3) \), \( q \) states, seq len \( n \)
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)**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>G</td>
<td>A</td>
<td>U</td>
<td>A</td>
<td>A</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>A</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>C</td>
<td>C</td>
<td>G</td>
<td>G</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>C</td>
<td>U</td>
<td>G</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>C</td>
<td>U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>G</td>
<td>U</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</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>
<td></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></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>C</td>
<td>U</td>
<td>G</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>U</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>U</td>
<td>U</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>A</td>
<td>C</td>
<td>U</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MI:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0.30</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.55</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0.30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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.
MI-Based Structure-Learning

Find best (max total MI) subset of column pairs among \( i \ldots j \), subject to absence of pseudo-knots

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

“Just like Nussinov/Zucker folding”

BUT, need enough data---enough sequences at right phylogenetic distance
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).
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):**

- Hom.sap.
- Cav.por.
- Mus.mus.
- Rat.nor.
- SS_cons

**Score Thresh T**

**Window Len W**

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

**Rfam**
Figure 2. Taxonomic distribution of Rfam family members in the three kingdoms of life.
Faster Genome Annotation of Non-coding RNAs Without Loss of Accuracy

Zasha Weinberg
& W.L. Ruzzo

Recomb ‘04, ISMB ‘04, Bioinfo ‘06
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 ➔ BLAST ➔ CM ➔ junk, hits
1 month, 1000 computers

Our Work

EMBL ➔ Ravenna ➔ CM ➔ hits
~2 months, 1000 computers

Rfam Goal

EMBL ➔ CM ➔ junk ➔ hits
10 years, 1000 computers
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

$\text{Score}(\pi) = \text{product of \text{“probabilities” on } \pi}$

NB: ok if \text{“probs” aren’t, e.g. } \sum \neq 1
(e.g. in CM, emissions are odds ratios vs 0th-order background)

For any nucleotide sequence $x$:

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

$\text{Forward-score}(x) = \sum \{ \text{score}(\pi) \mid \pi \text{ emits } x \}$
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
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 ≈ 25%

Opt 1:
\[ L_U + (R_A + R_G)/2 = 4 \]

Opt 2:
\[ L_U + (R_A + R_G)/2 = 2.5 \]
Optimizing filtering

For any nucleotide sequence \( x \):

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

Expected Forward Score

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

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↓ ⇒ Viterbi↓ ⇒ filter↓")

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

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

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

$$\frac{\partial E(L_1, L_2, \ldots)}{\partial L_i}$$
## 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 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)
Filter Chains

Figure 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

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 (B₁₂) riboswitch
tRNA
SECIS
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
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.
Approaches

Align sequences, then look for common structure

Predict structures, then try to align them

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 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
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
M-step uses M.I. + folding energy for structure prediction
CMfinder Accuracy
(on Rfam families with flanking sequence)
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
Footprinter finds patterns of conservation

Upstream of folC

11S_MUTANS_UA159
16S_AGALACTIAE_NEM316
15S_PYOGENES_SSI-1
65_AUREUS_SUBSP.
8L_LACTIS
9E_Faecalis_V583
190_YELLOWS_PHYTOPLASMA
17S_PNEUMONIA_R5
10L_PLANTARUM
2B_THURINGIENSIS_SEROVAR_KONKU
3B_CEREUS_ATCC_10987
4L_INNOCUA_CHROMOSOME
5L_MONOCYTOGENES_STRI
1B_SUBTILIS
13C_ACETOBUTYVICUM_ATCC824
14C_TETANI_E88
18C_PERRINGENS
12T_TENGCONGENS
7B_HALODURANS
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)

tyrS T box structure
- **CMfinder**: 9 instances
- **Found by Scan**: 447 hits
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>
Gene Regulation: The MET Repressor

Protein

DNA

Alberts, et al, 3e.
Riboswitch alternative → The protein way

Corbino et al., Genome Biol. 2005
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, 2 more in progress
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>
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
Course Wrap Up
What is DNA? RNA?
How many Amino Acids are there?
Did human beings, as we know them, develop from earlier species of animals?
What are stem cells?
What did Viterbi invent?
What is dynamic programming?
What is a likelihood ratio test?
What is the EM algorithm?
How would you find the maximum of \( f(x) = ax^3 + bx^2 + cx +d \) in the interval \(-10 < x < 25\)?
“High-Throughput BioTech”

Sensors
- DNA sequencing
- Microarrays/Gene expression
- Mass Spectrometry/Proteomics
- Protein/protein & DNA/protein interaction

Controls
- Cloning
- Gene knock out/knock in
- RNAi

Floods of data

“Grand Challenge” problems
CS Points of Contact

Scientific visualization
  Gene expression patterns

Databases
  Integration of disparate, overlapping data sources
  Distributed genome annotation in face of shifting underlying coordinates

AI/NLP/Text Mining
  Information extraction from journal texts with inconsistent nomenclature, indirect interactions, incomplete/inaccurate models,…

Machine learning
  System level synthesis of cell behavior from low-level heterogeneous data (DNA sequence, gene expression, protein interaction, mass spec,

Algorithms

…
Frontiers & Opportunities

New data:
Proteomics, SNP, arrays CGH, comparative sequence information, methylation, chromatin structure, ncRNA, interactome

New methods:
graphical models? rigorous filtering?

Data integration
many, complex, noisy sources
Frontiers & Opportunities

Open Problems:
- splicing, alternative splicing
- multiple sequence alignment (genome scale, w/ RNA etc.)
- protein & RNA structure
- interaction modeling
- network models
- RNA trafficking
- ncRNA discovery
- ...
Exciting Times

Lots to do
Various skills needed
I hope I’ve given you a taste of it
Thanks!