

RNA Search and Motif Discovery

Lectures 18-19

CSE 527

Autumn 2007

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

```
1 gagccccggcc cgggggacgg gcggcgggat agcgggaccc cggcgcggcg gtgcgcttca
61 gggcgcagcg gcggccgcag accgagcccc gggcgcggca agaggcggcg ggagccgggtg
121 ggggtggcgc atcatgcttc gaggcgtct gctgagatc gccctgggat ttaccgtgct
181 ttcaaggctc caatcggcgc agggggggga ggggggggga gaggctggga acgtgaagga
241 aaccagagcc agtcggggcca agagaagagg cgggtggagga cacgacgcgc ttaaaggacc
301 caatcggcgc atcatgcttc gaggcgtct gctgagatc gccctgggat ttaccgtgct
361 cggaaatcag tctattgtcc ccatttgccc gctattcctgt ggggatggat tttgttcgag
421 gctgagatc caatcggcgc agggggggga ggggggggga gaggctggga acgtgaagga
481 acaacactgc aatattcgct gtatgaatgg aggtagctgc agtgacgatc actgtctatg
541 ccaaaagga tacatagcgc ttttactgtg acacactgtt tgtaaagtg gctgtctcaa
601 tggaggaagg tctgtggccc caaatcgatg tgcattgcact tctggattta ctggaccca
661 gtgtgaaaga gattacagga caggccgatg ttttactgtg atgagcaacc agatgtgcca
721 gggacactc atcatgcttc gaggcgtct gctgagatc gccctgggat ttaccgtgct
781 ctggggccac ccctgtgaga tgtgtcctgc ccagcctcac ccctgccgcc gtggcttcat
841 tccaaatata cgcacgggag cttgtcaaga tgtggatgaa tgccaggcca tccccgggct
901 ctgtcagggg ggaattgca ttaatactgt tgggtctttt gagtgcaaat gccctgctgg
961 acacaaactt aatgaagtgt cacaaaaatg tgaagatatt gatgaatgca gcaccattcc
1021 ...
```



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^{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

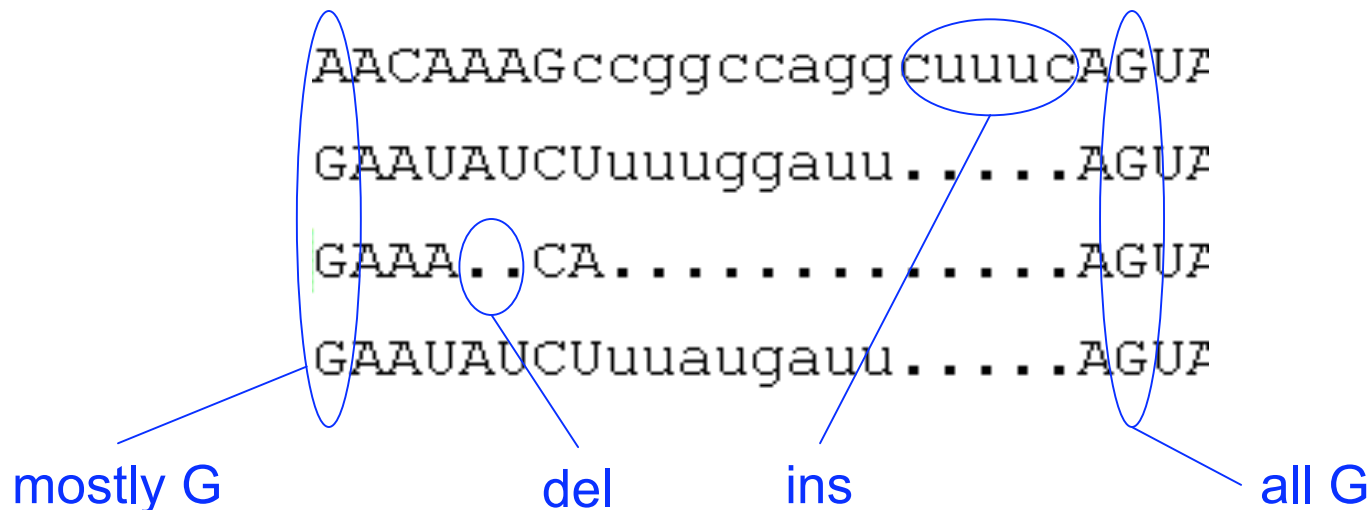
Task 2: Motif Description

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



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 Nussinov/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”

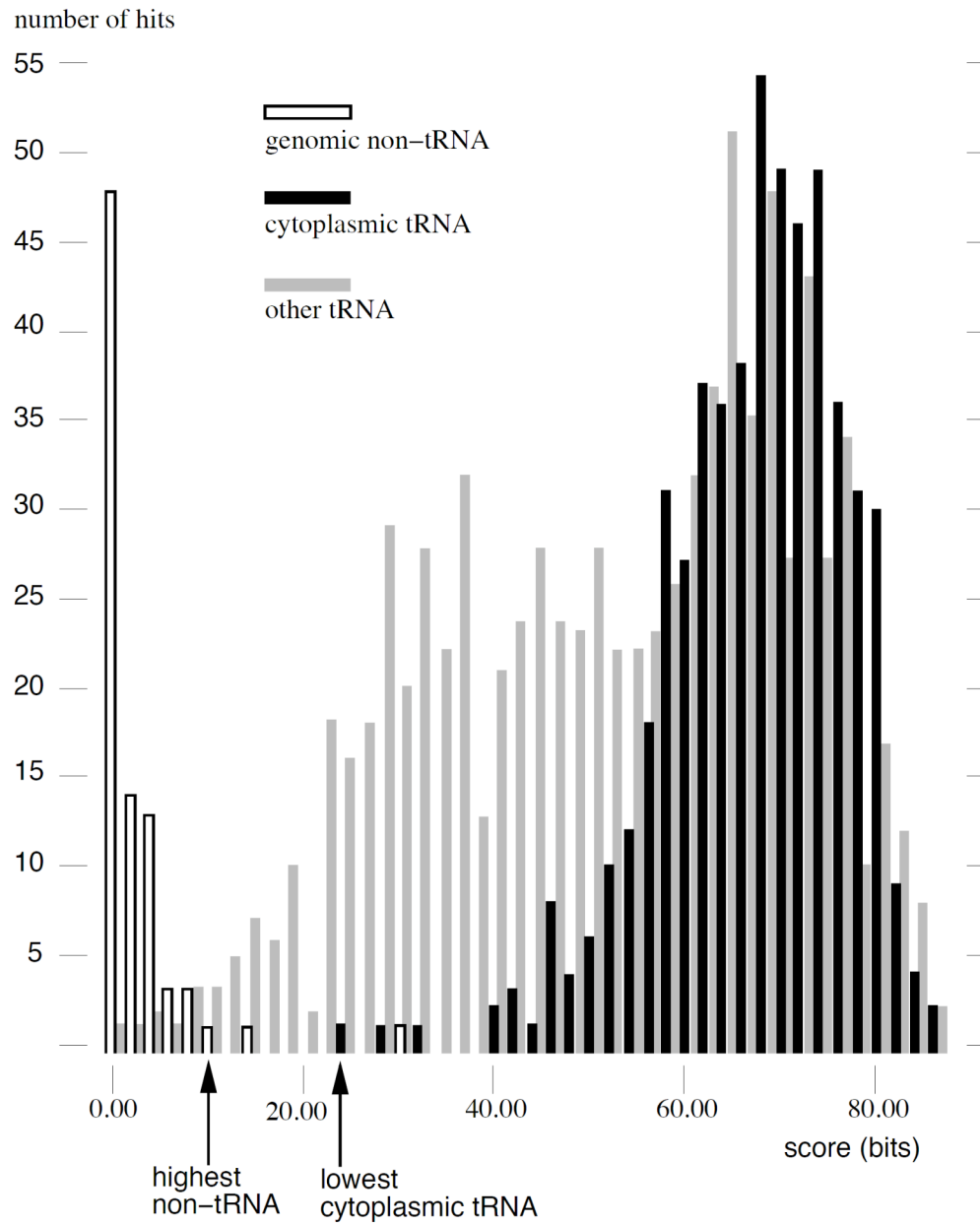
Scoring:

- “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 GCGGAUUUAGCUCAGUU GGG AGAGCGCCAGACUGAAG AUCUGGAG GUCCUGUGUUCGAUCCACAGAAUUCGCACCA
DF6280G GCGGAUUUAGCUCAGUU GGG AGAGCGCCAGACUGAAGAAAUAUCUUCGGUCAAGUUAUCUGGAG GUCCUGUGUUCGAUCCACAGAAUUCGCA
DD6280 UCCGUGAUAGUUUAAU GGUCAGAAUGGGCGCUUGUCG CGUGCCAG A UCGGGGUCAAUUCGCCGUCGCGGAGCCA
DX1661 CGCGGGGUGGAGCAGCCUGGU AGCUCGUCGGGCUCAUA ACCCGAAG GUCGUCGGUCAAUUCGGCCCCCGCAACCA
DS6280 GGCAACUUGGCCGAGU GGUUAAGGCGAAAGAUUAGAA AUCUUUU GGGCUUUGCCC GCGAGGUUCGAGUCCUGCAGUUGUCGCCA
```

U100:

```
DF6280 GCGGAUUUAGCUCAG UUGGGAGAGCGCCAGACU GA AG AUCUGGA GGUCCUGUGUUCGAUCCACAGAAUUCGCacca
DF6280G GCGGAUUUAGCUCAG UUGGGAGAGCGCCAGACUgaagaaauacuUCgguCAaguuAUCUGGA GGUCCUGUGUUCGAUCCACAGAAUUCGCA
DD6280 UCCGUGAUAGUUUAA UGGUCAGAAUGGGCGCUU GU CG CGUGCCA GAU CGGGGUCAAUUCGCCGUCGCGGAGcca
DX1661 CGCGGGGUGGAGCAGcCUGGUAGCUCGUCGGGCU CA UA ACCCGAA GGUCGUCGGUCAAUUCGGCCCCCGCAacca
DS6280 GGCAACUUGGCCGAG UGUUAAGGCGAAAGAUU AG AA AUCUUUUgggcuuugcccG GCGAGGUUCGAGUCCUGCAGUUGUCgcca
```

ClustalV:

```
DF6280 GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGA UCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA
DF6280G GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGAAAUAUCUUCGGUCAAGUUAUCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCA
DD6280 UCCGUGAUAGUUUAAU G GUCAGAAUGG GCG CUUG UCGGUGCC AGAUCGG GGUCAAUUCGCCGUCGCGGAGCCA
DX1661 CGCGGGGUGGAGCAGC CUGGUAGCUCGUCGGG CUCA UAACCCGA AGGUCGUCGGUCAAUUCGGCCCCCGCAACCA
DS6280 GGCAACUUGGCCGAGUGGUUAAGGCGAAAGAUU AGAAAUCUUUUGGGC UUUGCCCC GCGAGGUUCGAGUCCUGCAGUUGUCGCCA
```

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.

Slightly different
evaluation criteria

Profile HMM Structure

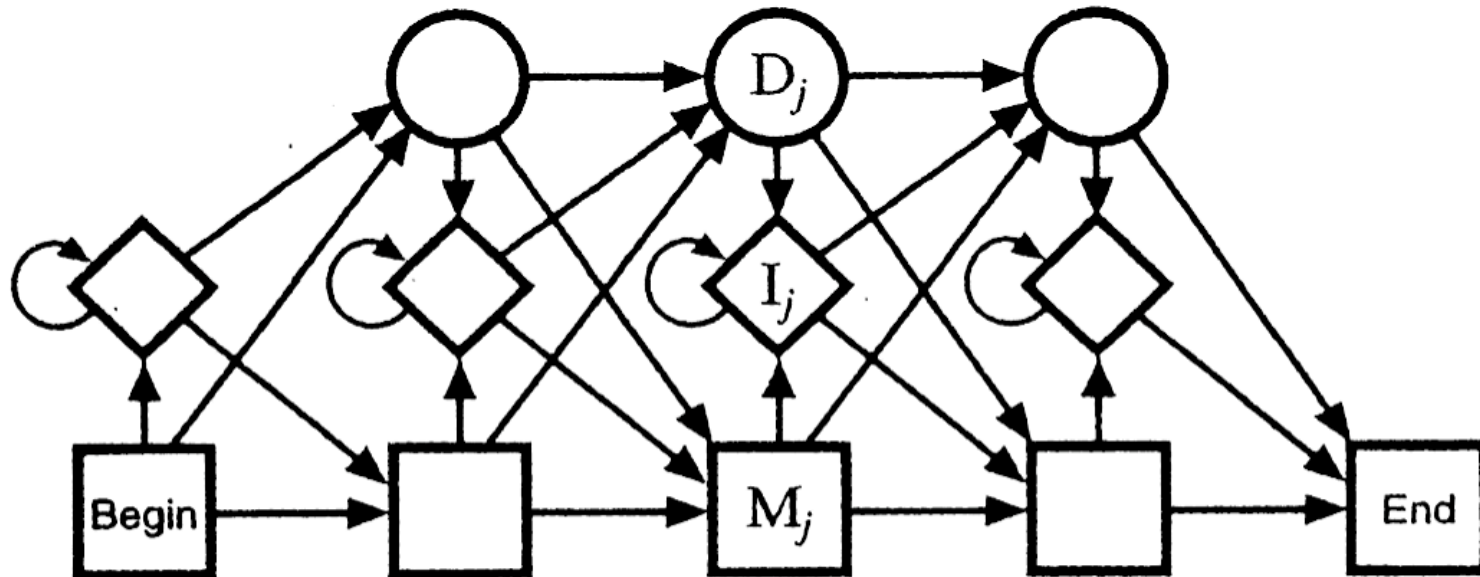


Figure 5.2 *The transition structure of a profile HMM.*

M_j: Match states (20 emission probabilities)

I_j: Insert states (Background emission probabilities)

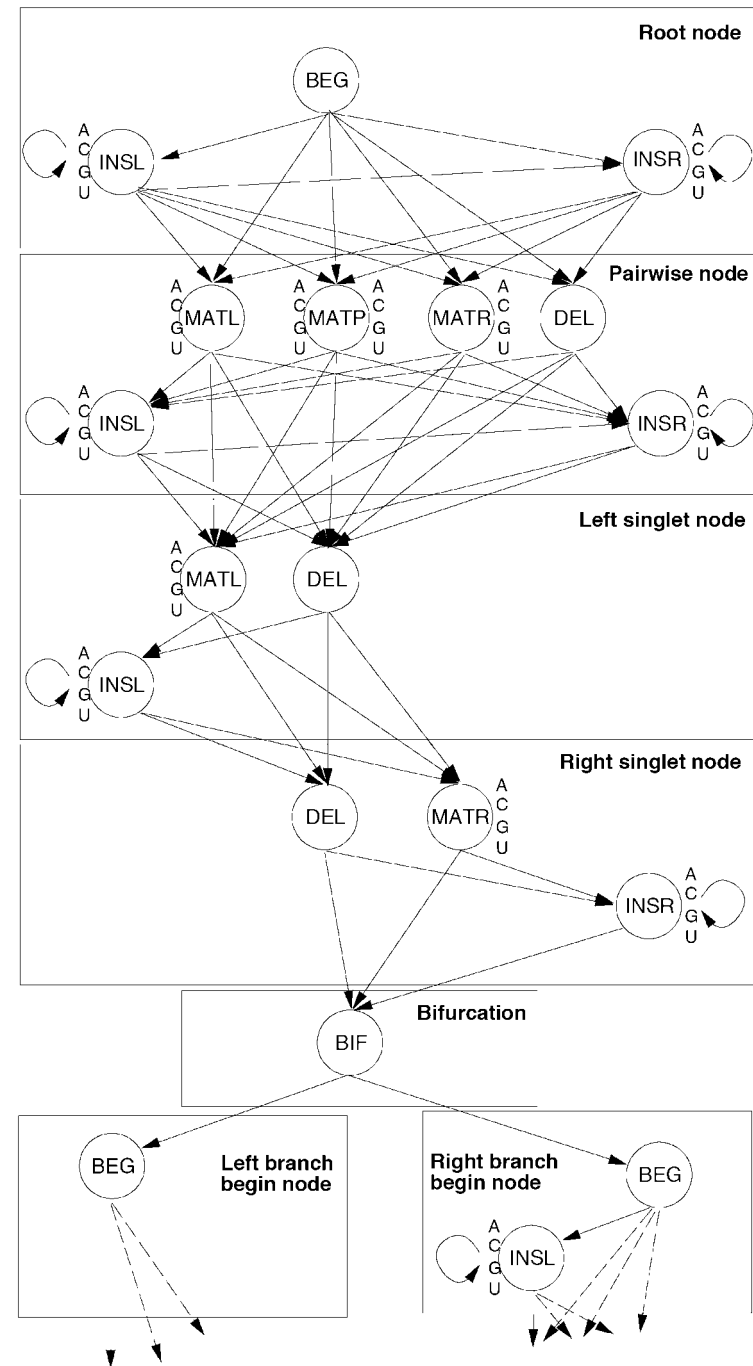
D_j: Delete states (silent - no emission)

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} letter of input

x_{ij} = substring i, \dots, 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 = $\max_{\pi} \log P(x_{ij} \text{ gen'd starting in state } y \text{ via path } \pi)$

CM Viterbi Alignment

(the “inside” algorithm)

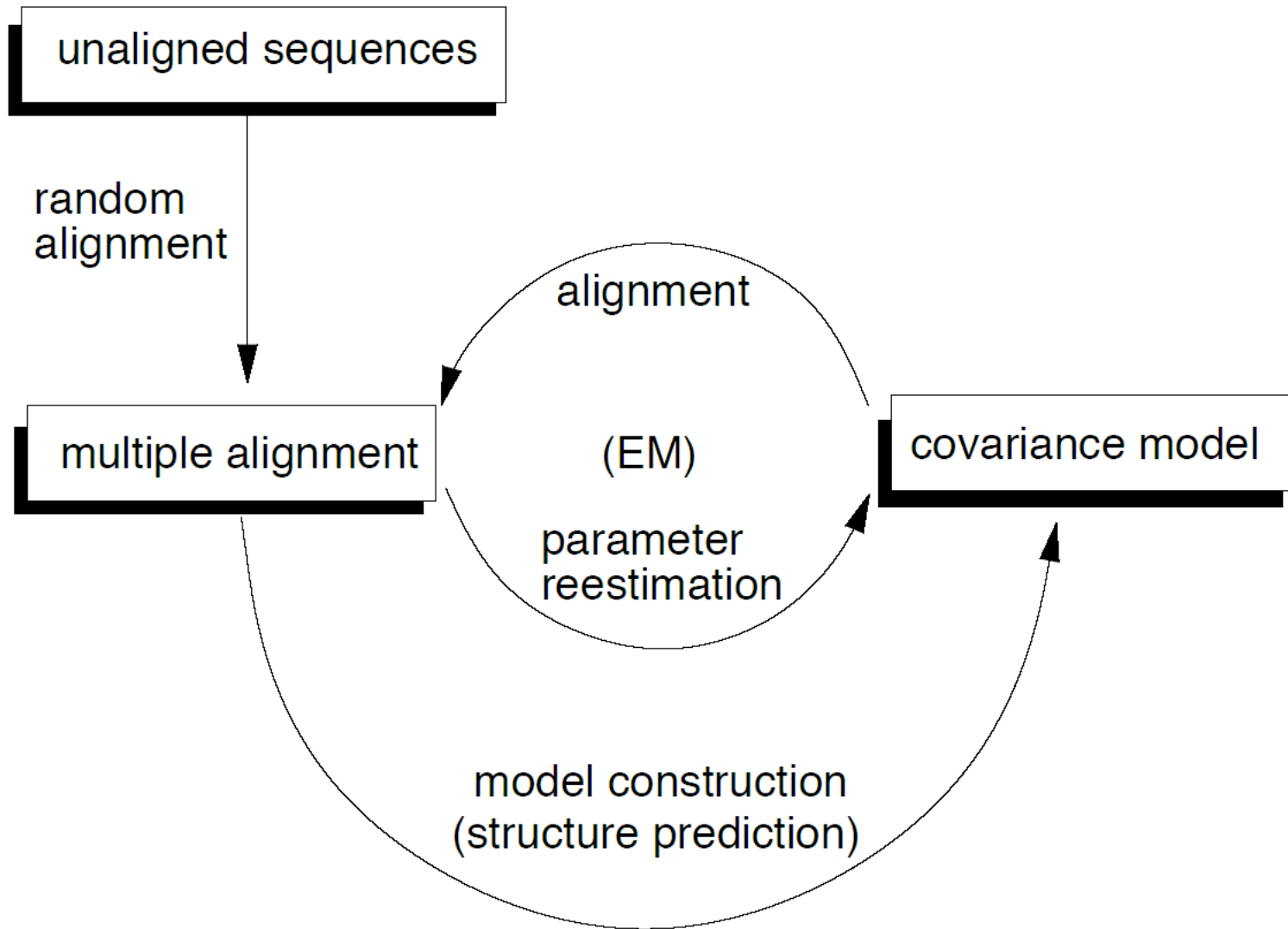
$$S_{ij}^y = \max_{\pi} \log P(x_{ij} \text{ generated starting in state } y \text{ via path } \pi)$$

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



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

Model Training



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)

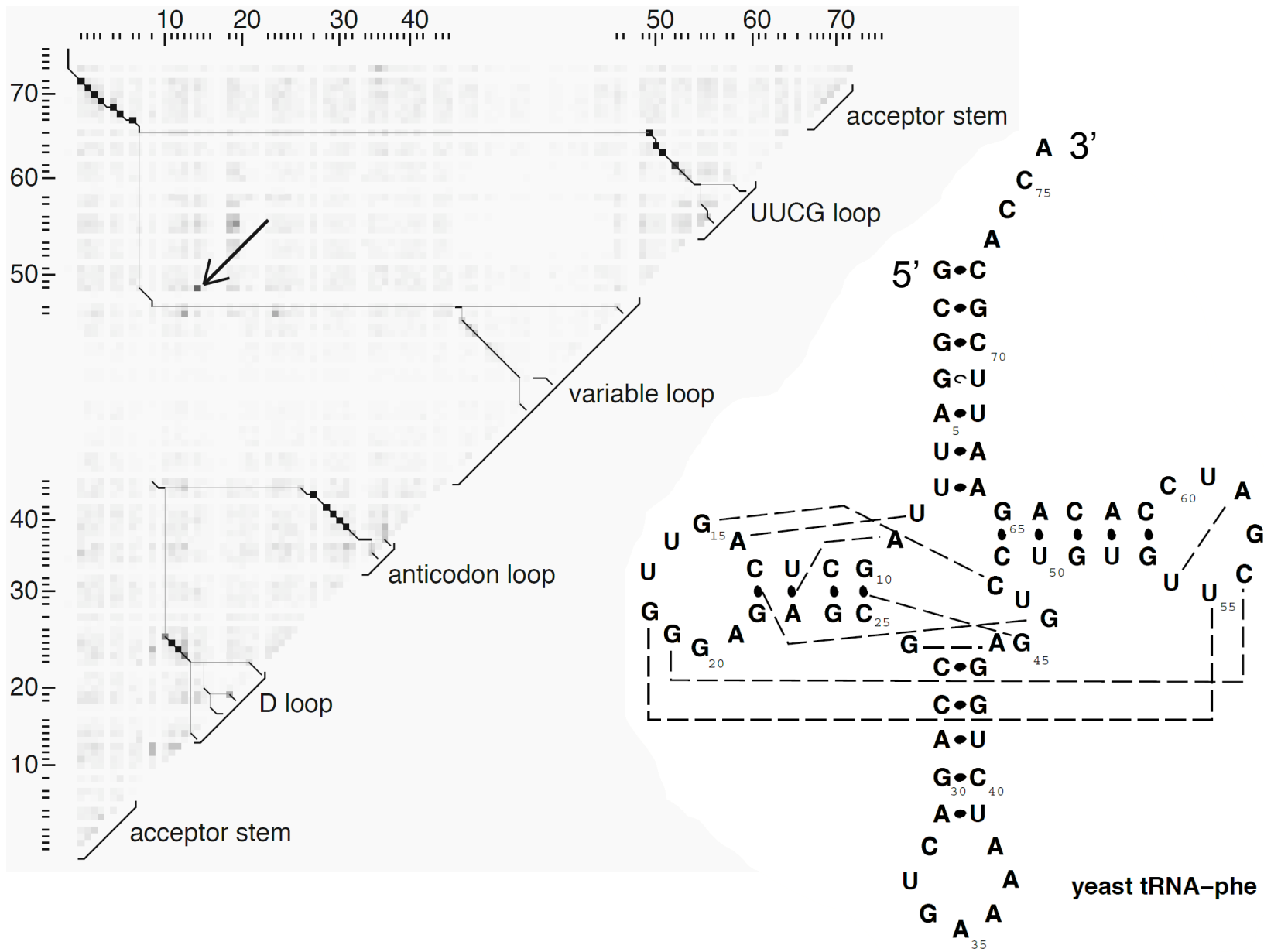
	1	2	3	4	5	6	7	8	9
A	A	G	A	U	A	A	U	C	U
A	A	G	A	U	C	A	U	C	U
A	A	G	A	C	G	U	U	C	U
A	A	G	A	U	U	U	U	C	U
A	A	G	C	C	A	G	G	C	U
A	A	G	C	G	C	G	G	C	U
A	A	G	C	U	G	C	G	C	U
A	A	G	C	A	U	C	G	C	U
A	A	G	G	U	A	G	C	C	U
A	A	G	G	G	C	G	U	C	U
A	A	G	G	C	U	U	C	C	U
A	A	G	U	A	A	A	A	C	U
A	A	G	U	C	C	A	A	C	U
A	A	G	U	U	G	C	A	C	U
A	A	G	U	U	U	C	A	C	U
A	16	0	4	2	4	4	4	0	0
C	0	0	4	4	4	4	4	16	0
G	0	16	4	2	4	4	4	0	0
U	0	0	4	8	4	4	4	0	16

MI:	1	2	3	4	5	6	7	8	9
9	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0		
7	0	0	2	0.30	0	1			
6	0	0	1	0.55	1				
5	0	0	0	0.42					
4	0	0	0.30						
3	0	0							
2	0								
1									

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.



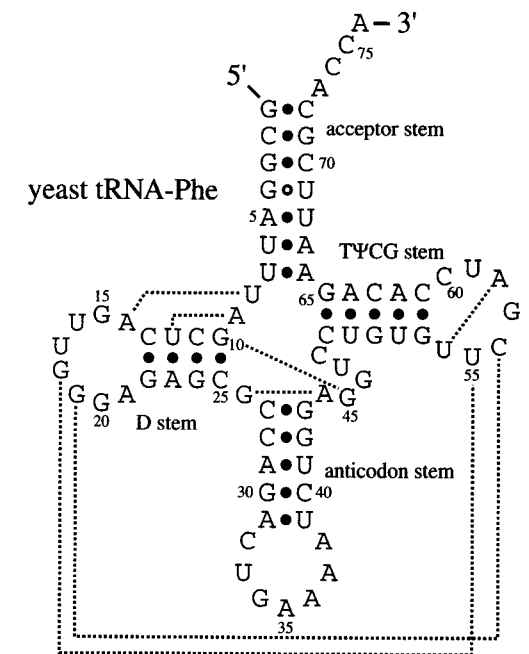
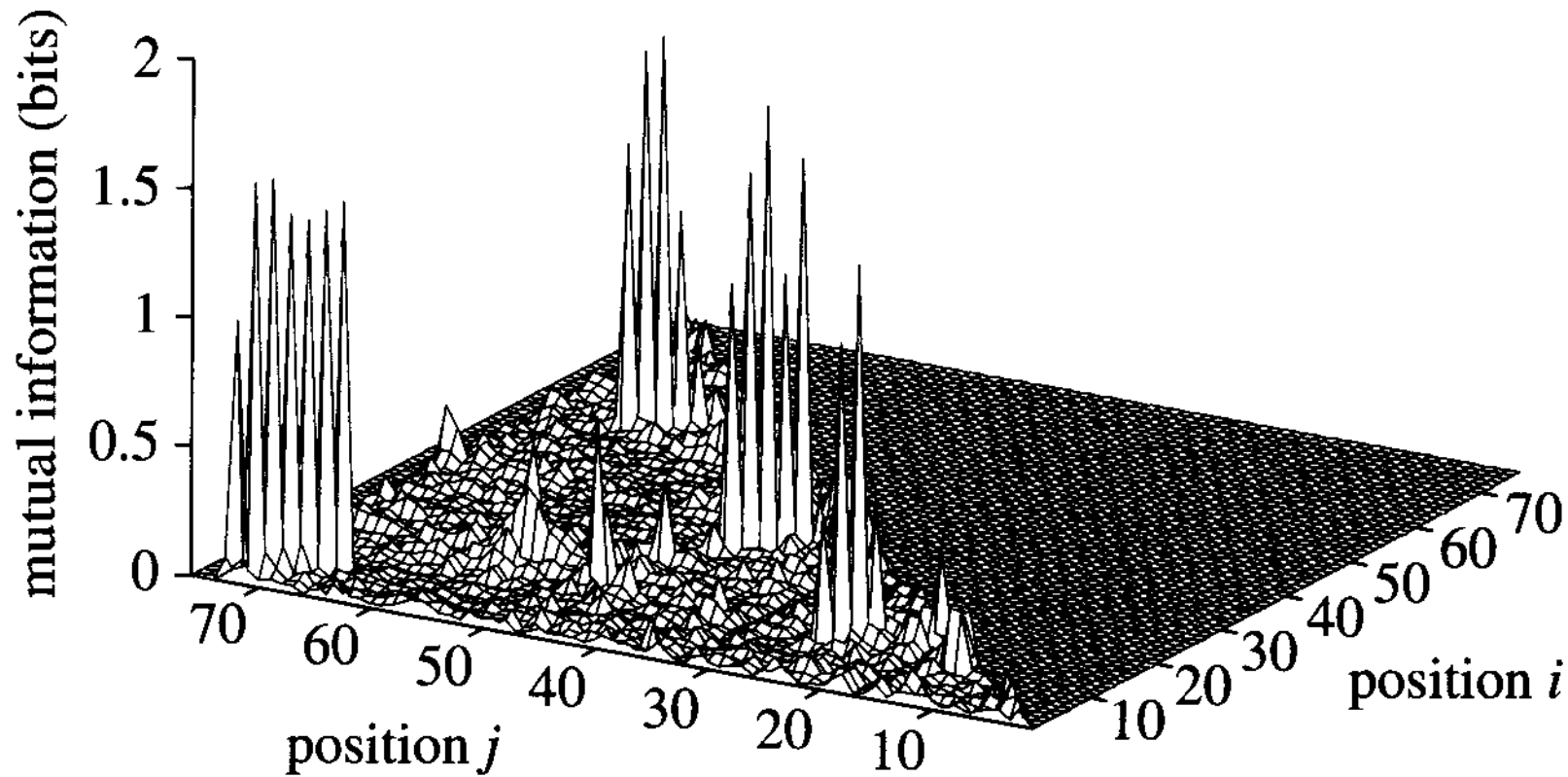


Figure 10.6 A mutual information plot of a tRNA alignment (top) shows four strong diagonals of covarying positions, corresponding to the four stems of the tRNA cloverleaf structure (bottom; the secondary structure of yeast phenylalanine tRNA is shown). Dashed lines indicate some of the additional tertiary 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.

MI-Based Structure-Learning

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

$$S_{i,j} = \max \left\{ \begin{array}{l} S_{i,j-1} \\ \max_{i \leq k < j-4} S_{i,k-1} + M_{k,j} + S_{k+1,j-1} \end{array} \right.$$

“Just like Nussinov/Zucker folding”

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

Pseudoknots
disallowed allowed $\left(\sum_{i=1}^n \max_j M_{i,j}\right)/2$

	Avg. id	Min id	Max id	ClustalV accuracy	1° info (bits)	2° info (bits)
TEST	.402	.144	1.00	64%	43.7	30.0-32.3
SIM100	.396	.131	.986	54%	39.7	30.5-32.7
SIM65	.362	.111	.685	37%	31.8	28.6-30.7

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.

Model	training set	iterations	score (bits)	alignment accuracy
A1415	all sequences (aligned)	3	58.7	95%
A100	SIM100 (aligned)	3	57.3	94%
A65	SIM65 (aligned)	3	46.7	93%
U100	SIM100 (degapped)	23	56.7	90%
U65	SIM65 (degapped)	29	47.2	91%

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).

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

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

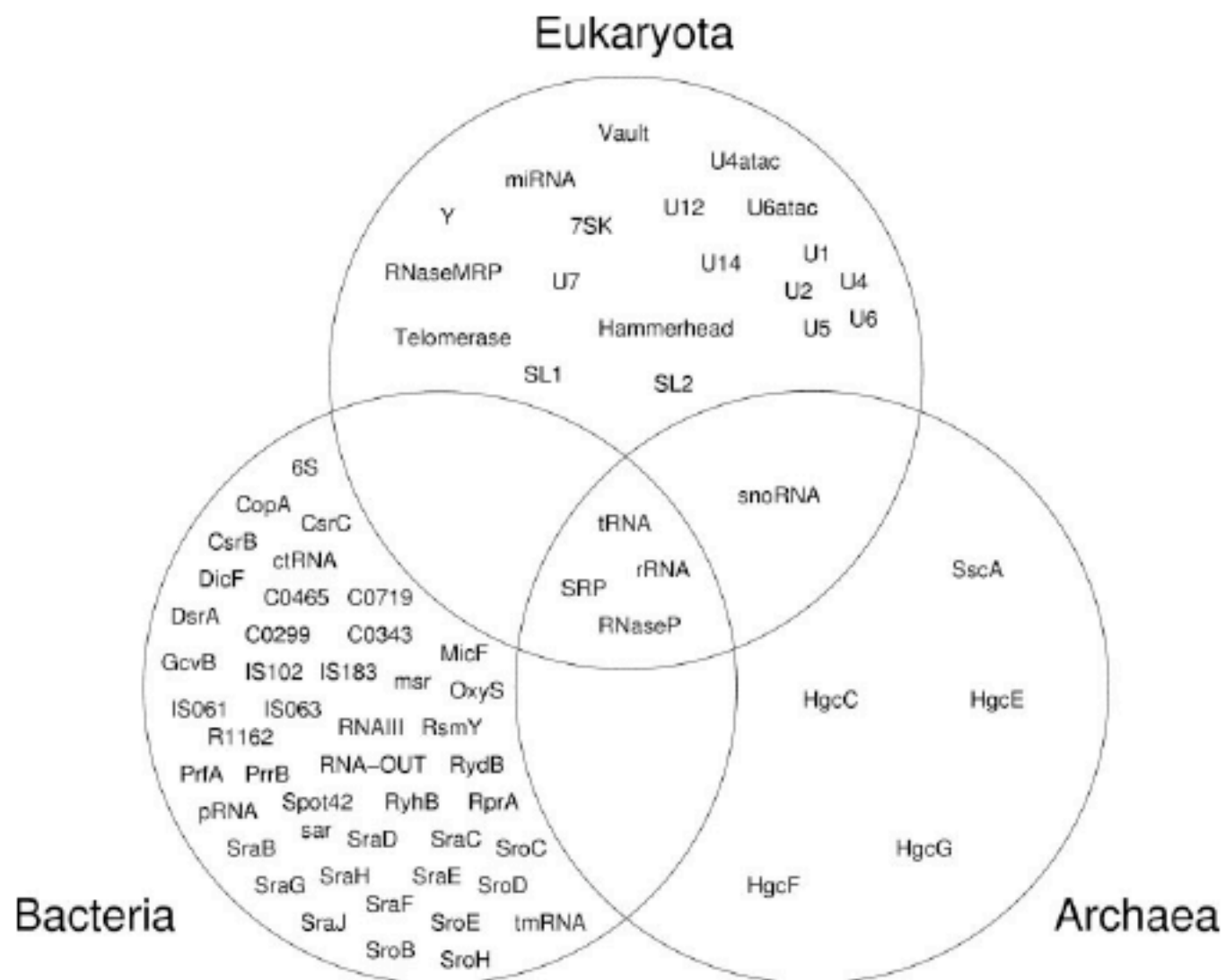


Figure 2. Taxonomic distribution of Rfam family members in the three kingdoms of life.

Rfam – key issues

Overly narrow families

Variant structures/unstructured RNAs

Spliced RNAs

RNA pseudogenes

Human ALU is SRP-related, with 1.1×10^6 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

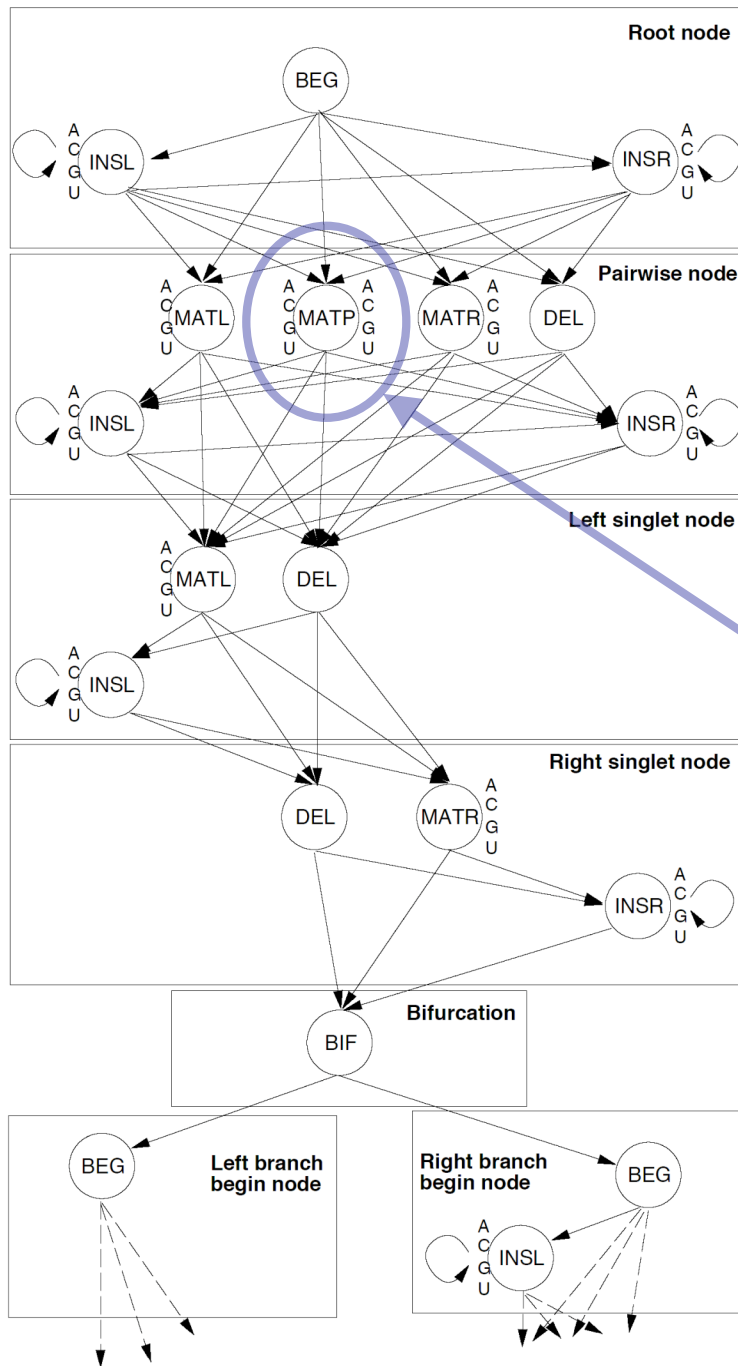
Recomb '04, ISMB '04, Bioinfo '06

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)

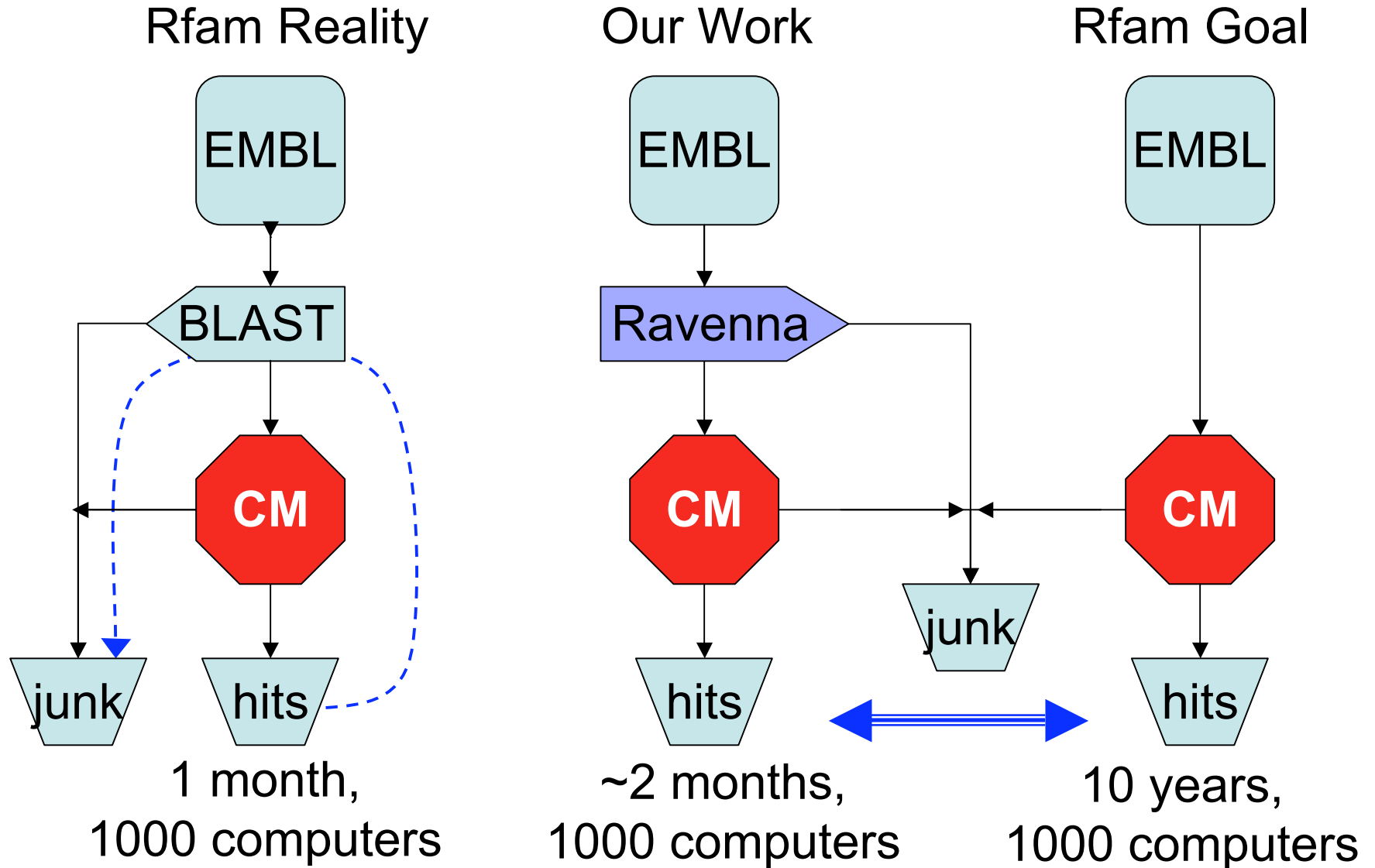
Weinberg & Ruzzo, *Bioinformatics*, 2004, 2006

Covariance Model



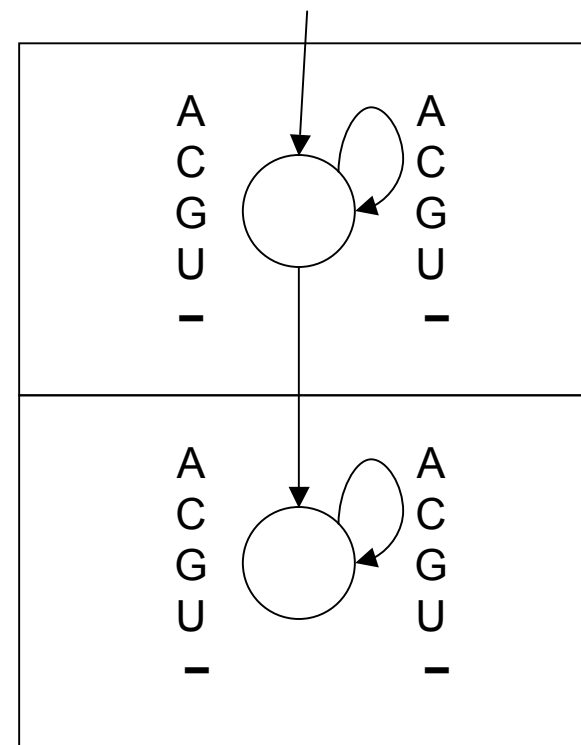
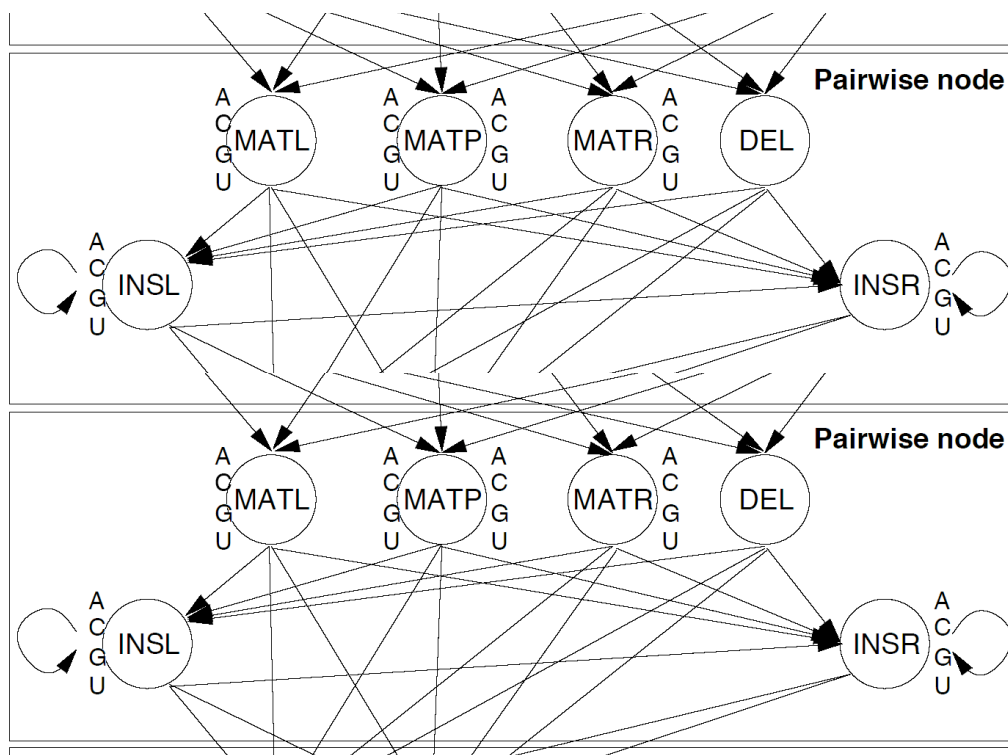
Key difference of CM vs HMM: Pair states emit paired symbols, corresponding to base-paired nucleotides; 16 emission probabilities here.

CM's are good, but slow



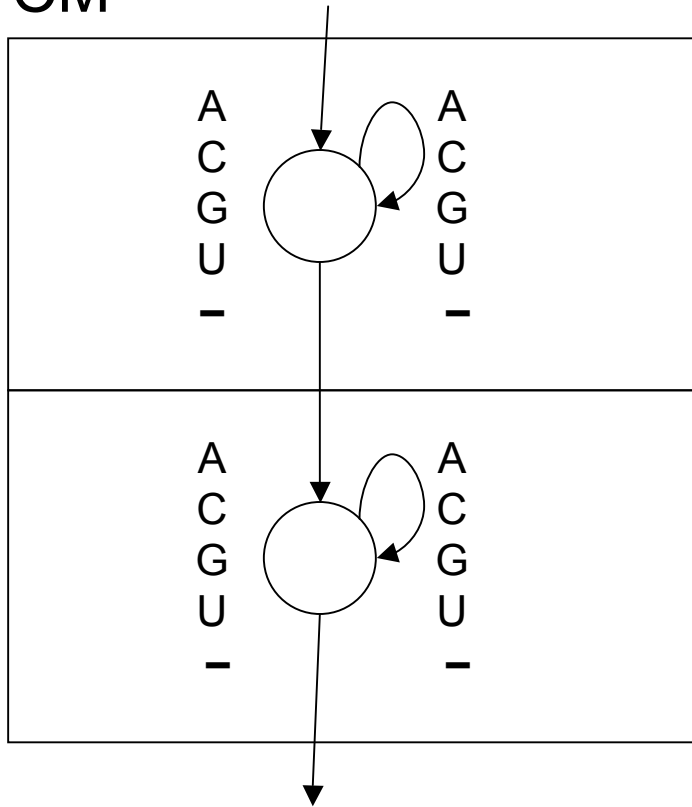
Simplified CM

(for pedagogical purposes only)



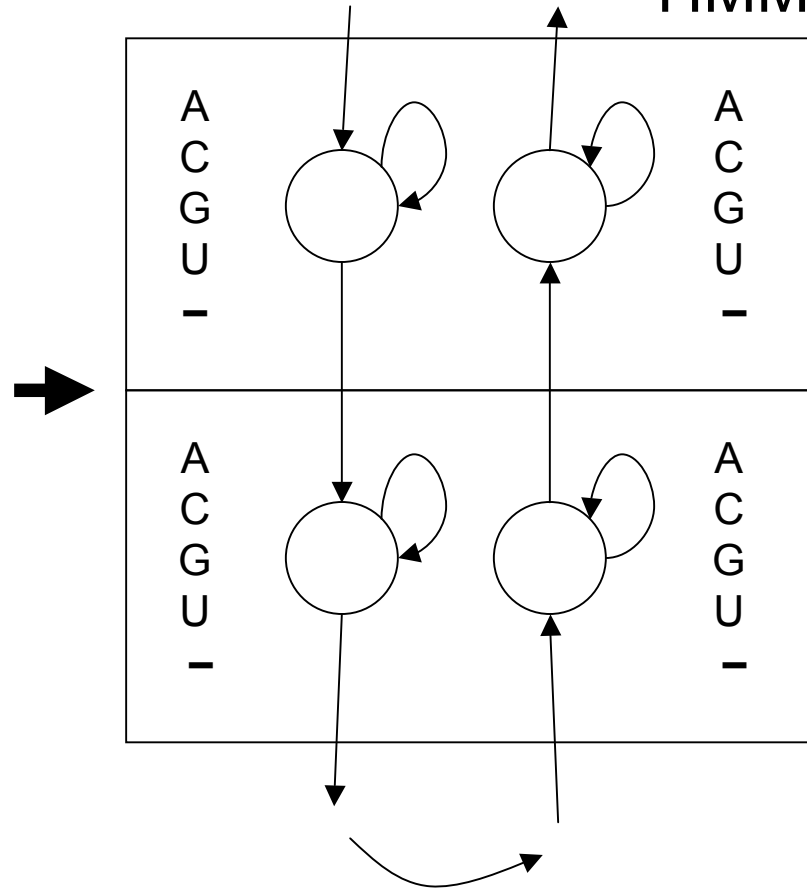
CM to HMM

CM



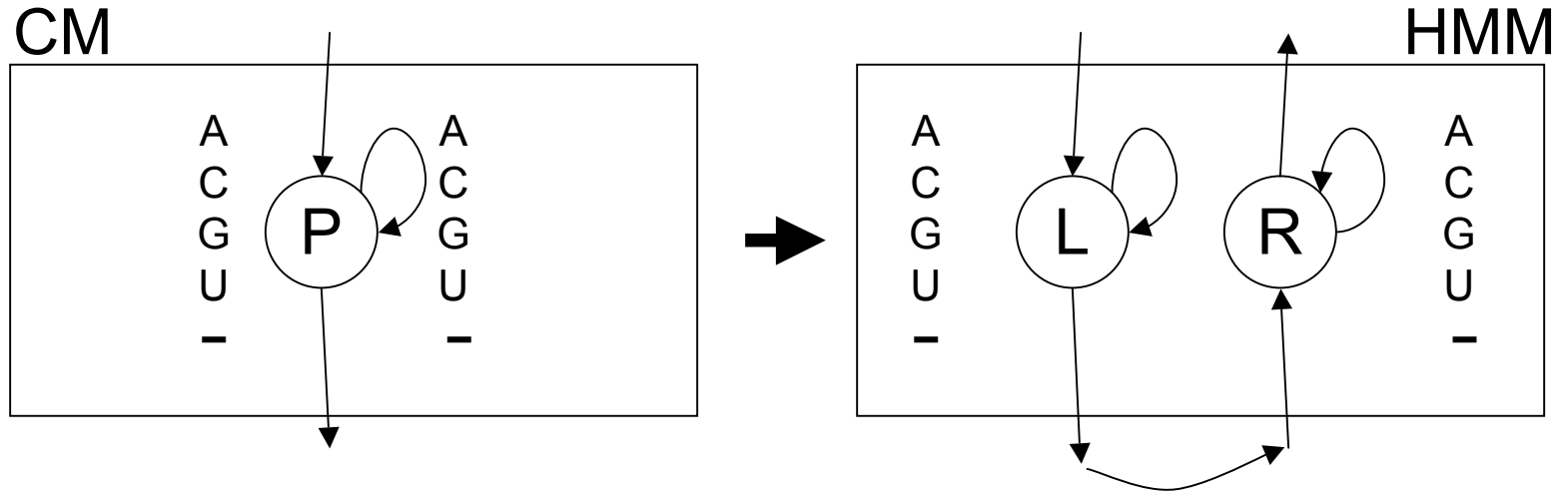
25 emissions per state

HMM



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 π defines transitions/emissions

Score(π) = product of “probabilities” on π

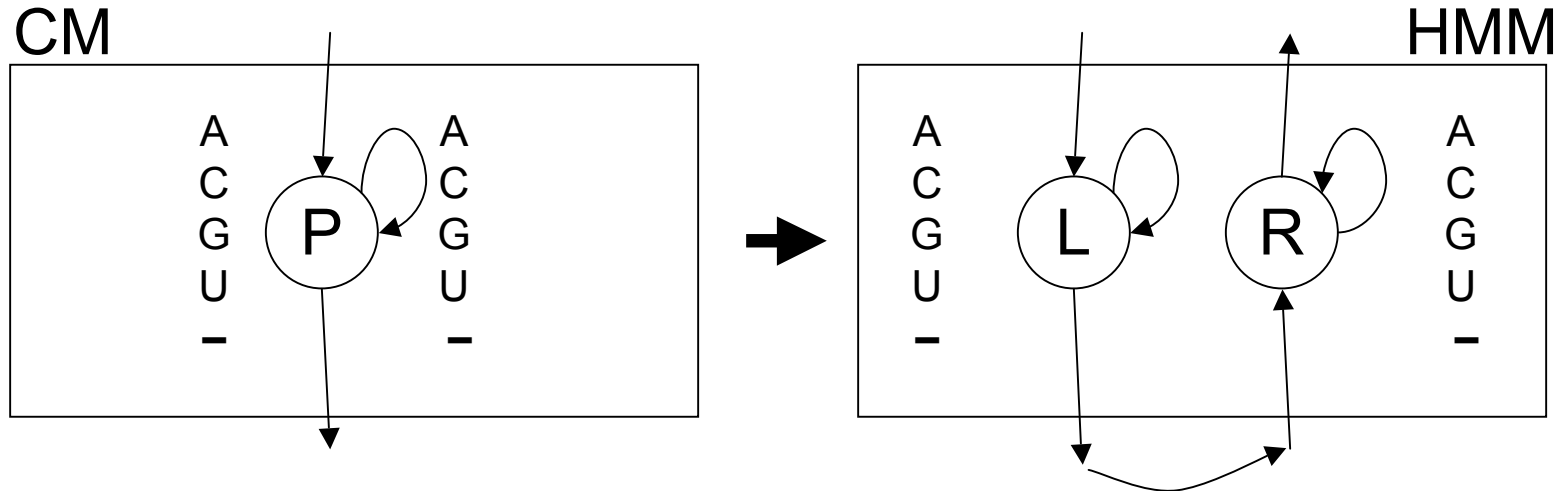
NB: ok if “probs” aren’t, e.g. $\sum \neq 1$
(e.g. in CM, emissions are odds ratios vs
0th-order background)

For any nucleotide sequence x :

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

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

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

Rigorous Filtering

$$\begin{aligned} P_{AA} &\leq L_A + R_A \\ P_{AC} &\leq L_A + R_C \\ P_{AG} &\leq L_A + R_G \\ P_{AU} &\leq L_A + R_U \\ P_{A-} &\leq L_A + R_- \\ &\dots \end{aligned}$$

Any scores satisfying the linear inequalities give rigorous filtering

Proof:

CM Viterbi path score
 \leq “corresponding” HMM path score
 \leq 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%

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

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

NB: E is a function of L_i, R_i only

Under 0th-order
background model

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 $E(L_i, R_i)$

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

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:

$$f_k(i) = P(x_1 \dots x_i, \pi_i = k)$$
$$f_l(i+1) = e_l(x_{i+1}) \sum_k f_k(i) a_{k,l}$$

Viterbi:

$$v_l(i+1) = e_l(x_{i+1}) \cdot \max_k (v_k(i) a_{k,l})$$

$$\frac{\partial E(L_1, L_2, \dots)}{\partial L_i}$$

Estimated Filtering Efficiency

(139 Rfam 4.0 families)

Filtering fraction	# families (compact)	# families (expanded)
$< 10^{-4}$	105	110
$10^{-4} - 10^{-2}$	8	17
.01 - .10	11	3
.10 - .25	2	2
.25 - .99	6	4
.99 - 1.0	7	3

} ~100x speedup

Results: New ncRNA's?

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

Results: With additional work

	# with BLAST+CM	# with rigorous filter series + CM	# new
Rfam tRNA	58609	63767	5158
Group II intron	5708	6039	331
tRNAscan-SE (human)	608	729	121
tmRNA	226	247	21
Lysine riboswitch	60	71	11
And more...			

“Additional work”

Profile HMM filters use *no* 2^{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)

} for some
} hairpins

Filter Chains

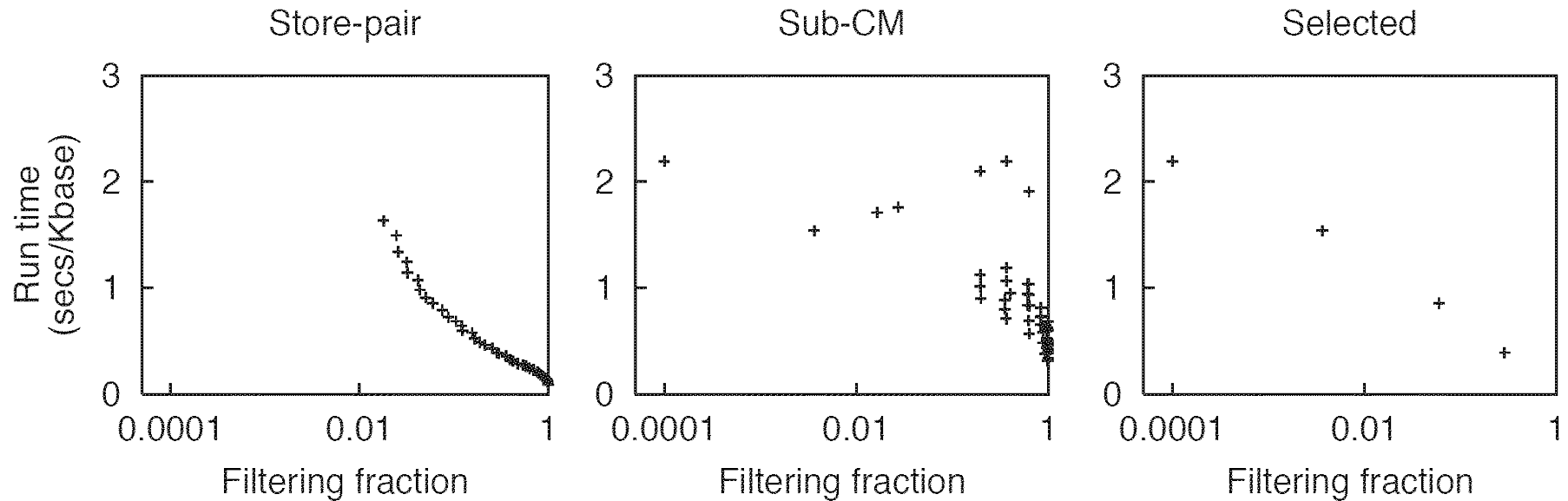


Fig. 2. Filter creation and selection. Filters for Rfam tRNA (RF00005) generated by the store-pair and sub-CM techniques and those selected for actual filtering are plotted by filtering fraction and run time. The CM runs at 3.5 secs/kbase. The four selected filters are run one after another, from highest to lowest fraction.

Heuristic Filters

Rigorous filters optimized for worst case

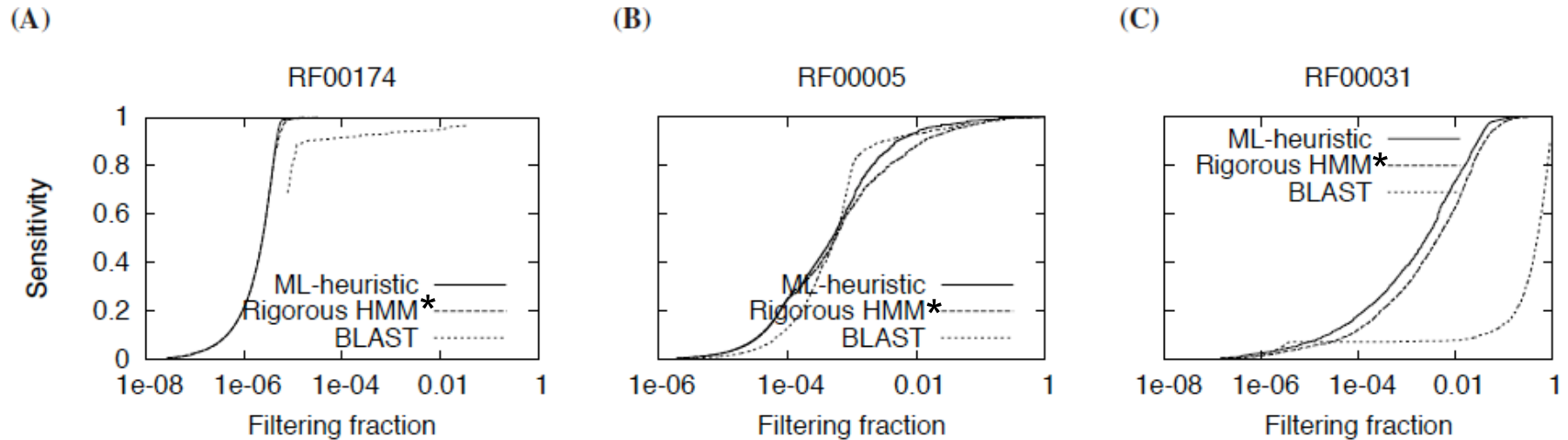
Possible to trade improved speed for small loss in sensitivity?

Yes – profile HMMs as before, but optimized for average case

“ML heuristic”: train HMM from the infinite alignment generated by the CM

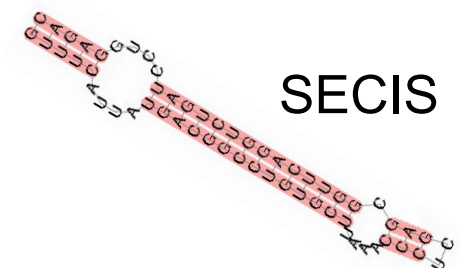
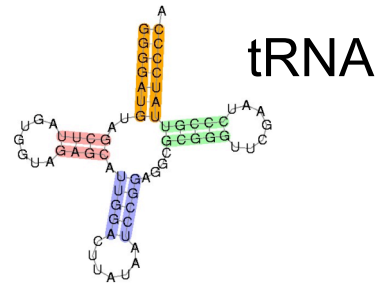
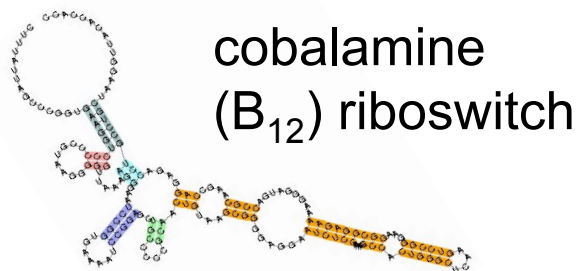
Often 10x faster, modest loss in sensitivity

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.



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

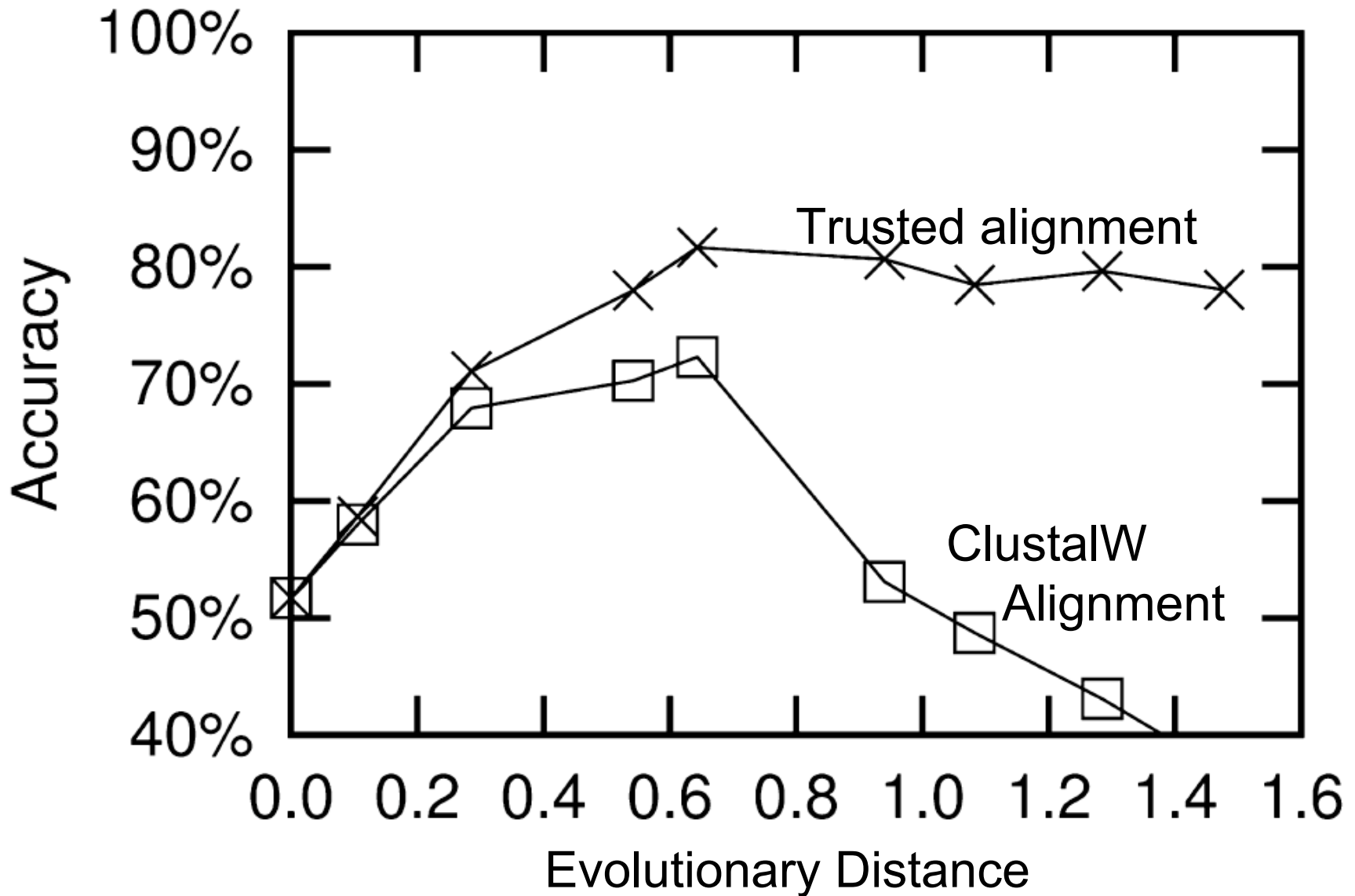
“Obvious” Approach I: Align First, Predict from Multiple Sequence Alignment

... GA ... UC ...
... GA ... UC ...
... GA ... UC ...
... CA ... UG ...
... CC ... GG ...
... UA ... UA ...



Compensatory mutations reveal structure, (core of “comparative sequence analysis”) *but* usual alignment algorithms penalize them (twice)

Pfold (KH03) Test Set D



Knudsen & Hein, Pfold: RNA secondary structure prediction using stochastic context-free grammars, Nucleic Acids Research, 2003, v 31,3423–3428

Pitfall for sequence-alignment- first approach

Structural conservation \neq Sequence conservation
Alignment without structure information is unreliable

CLUSTALW alignment of SECIS elements with flanking regions

```
-----CCCCCCCCAGGCTCCTGGTGCCCGG--ATGATGACGACCTGGGTG-GAA-A---CCTACCCTGTGGGCACCC-ATGTCGA-GCCCCCTGGCATT
GGGATCATTGCAAGAGCAGCGTG--ACTGACATTA--TGAAGGCCTGTACTGAAGACAGCAA--GCTGTTAGTACAGACC---AGATG---CTTTCCTGGCAGGCCTCGTTGTACCTCTTGGAAAACCTCAAT
AGGTTTGCATTAATGAGGATTACACAGAAAACCTTI-GTTAAGGGTTTGTGTGATCTGCTAA--TTGGCAAATTTTTATTTTTTAAAT---ATTCTTACAGAAGAGTTCCATTTAAGAATGTTTCGTATAGG
AGTGTGCGGATGATAACTACTGACGAAAGAGTCATCGACTCAGTTAGTGGTTGGATGTAGTCACATTAGTPTGCCTCTCCCCATCTTTG---TCTCCCTGGCAAGGAGAATATGCGGGACATGATGCTAAGAG
TGGACTGATAGGTA-GCCATGGC--TTCATCTGTC--ATG--TCTGCTTCTTTTTATATTG--TGATGATGGTCACAGTGTAAG-G---TTCCACAGCTGTGACTTGATTTTTAA-AAATGTCGGAAGA
TAAACTCGAACTCGAGCGGGCAATTGCTGATTACGA-TTAAACCACTGATTCTGGGTGCTGTC--TTCGTGGCCGTCGTCGGTTCCA-----TTTATCAACTATTAGCTCCAATACATAGCTACAGGTTTTT
AAATTCTCGCTATATGACGATGGCAATCTCAAATGT-TCATTTGGTTGCCATTIGATGAAATCAGTTTTGTGTGCACCTGATTGCAGAATTTTGTTTACCTTGCTCATTTTTTTTCATTGAA-ACCACTTCTCAGA
GGGGCGGGAGTACAAGGTGCGTGTGACTGGAGCCA--CCCCTCCGACTCTGCAGGTGTTG--CAAATGACGACCGATTTTGAATG---GTCACACGGCCAAAACTCGTGTCCGACATCAACCCCTTC
TTCTCCAGTGTCTAGTTACATTTGATGAGAACAGAA-ACATAAACTATGACCTAGGGGTTTCT--GTTGGATAGCTCGTAATTAAGAACGGAGAAAGAACAACAAGACATATTTCCAGTTTTTTTTCTTTAC
CAAATGATGGATA-GCCATTGGTATTTCATCTATT--TTAACTCTGTGTCITTACATATTG--TTTATGATGGCCACAGCCTAAA-G---TACACACGGCTGTGACTTGATTCAAAA-GAAA-----
TGAGCAACTTGTCT-GATGACTGGGAAAGGAGGAC--CTGCAACCATCTGACTTGGTCTCTG--TTAATGACGTCTCTCCCTCTAA-A---CCC-CATTAAGGACTGGGAGAGGCAGA-GCAAGCCTCAGAG
GATTACTGGCTGCACCTCTGGGGGGCGTTCTTCCA--TGATGGTGTTCCTCTAAATTTGCA--CGGAGAAACACCTGATTTCCAGGAAA-ATCCCTCAGATGGGCGCTGGTCCCATCCATTCCCGATGCCT
AGACCAGGCAAGACAACATGTGAGC-GCGATGGCCG--TGTACCCAGGTGAGGGGTGGTGTG--TCTATGAAGGAGGGGCCGAAG-----CCCTTGTGGGCGGGCCTCCCTGAGCCCGTCTGTGGTGCCAG
CACTTCAGAAGGCT-TCTGAATGGAACCATCTCTT--GACA-TTTGTTTCTATA-ATATTG--T-CATGACAGTCACAGCATAAA-G---CGCAGACGGCTGTGACTGATTTTAGA-AAATATTTTTAGA
```

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 → CM → Alignment

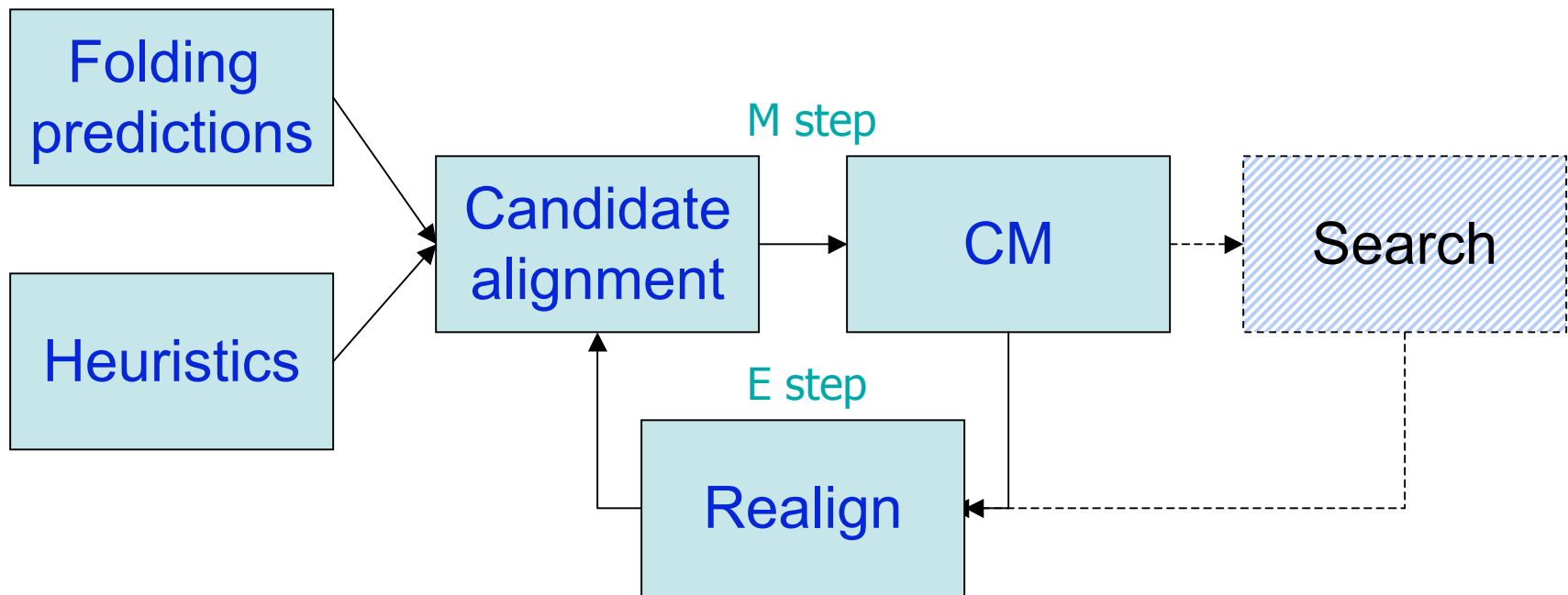
Similar to HMM, but much slower

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 = column i ; $\sigma = (\alpha, \beta)$ the 2^{ary} struct, α = unpaired, β = paired cols

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

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

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

Let

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

With MLE params, I_{ij} is the *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 $P(\sigma) = \prod_{k \in \alpha} s_k \prod_{(i,j) \in \beta} p_{ij}$, and $P(D, \sigma)$ can be rewritten as

$$\begin{aligned} P(D, \sigma) &= P(D|\sigma)P(\sigma) \\ &= \prod_{1 \leq k \leq l} P(L_k) s_k \prod_{(i,j) \in \beta} \frac{P(L_i L_j)}{P(L_i)P(L_j)} \frac{p_{ij}}{s_i s_j} \quad (4) \end{aligned}$$

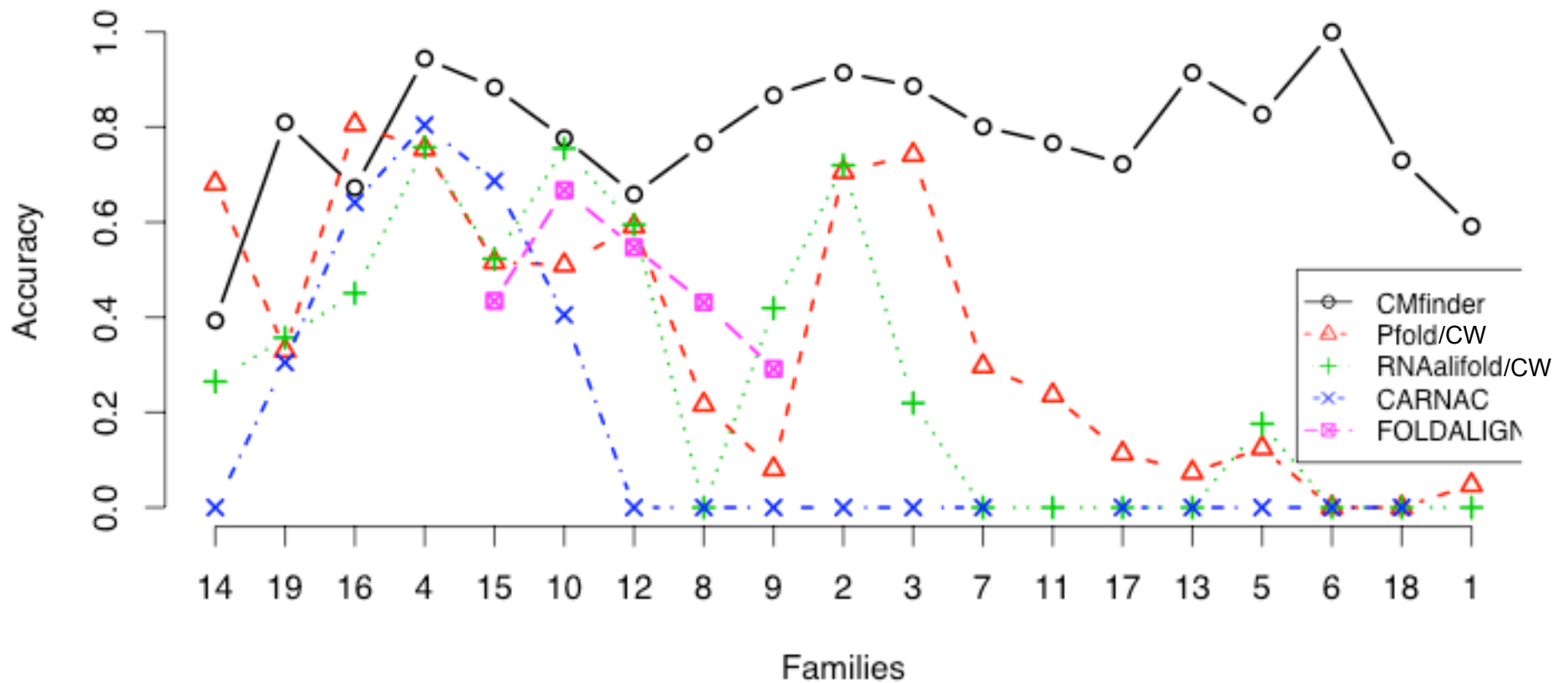
Let

$$K_{ij} = \log \left(\frac{P(L_i L_j)}{P(L_i)P(L_j)} \frac{p_{ij}}{s_i s_j} \right) = I_{ij} + \log \frac{p_{ij}}{s_i s_j},$$

then the maximum likelihood structure σ 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

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

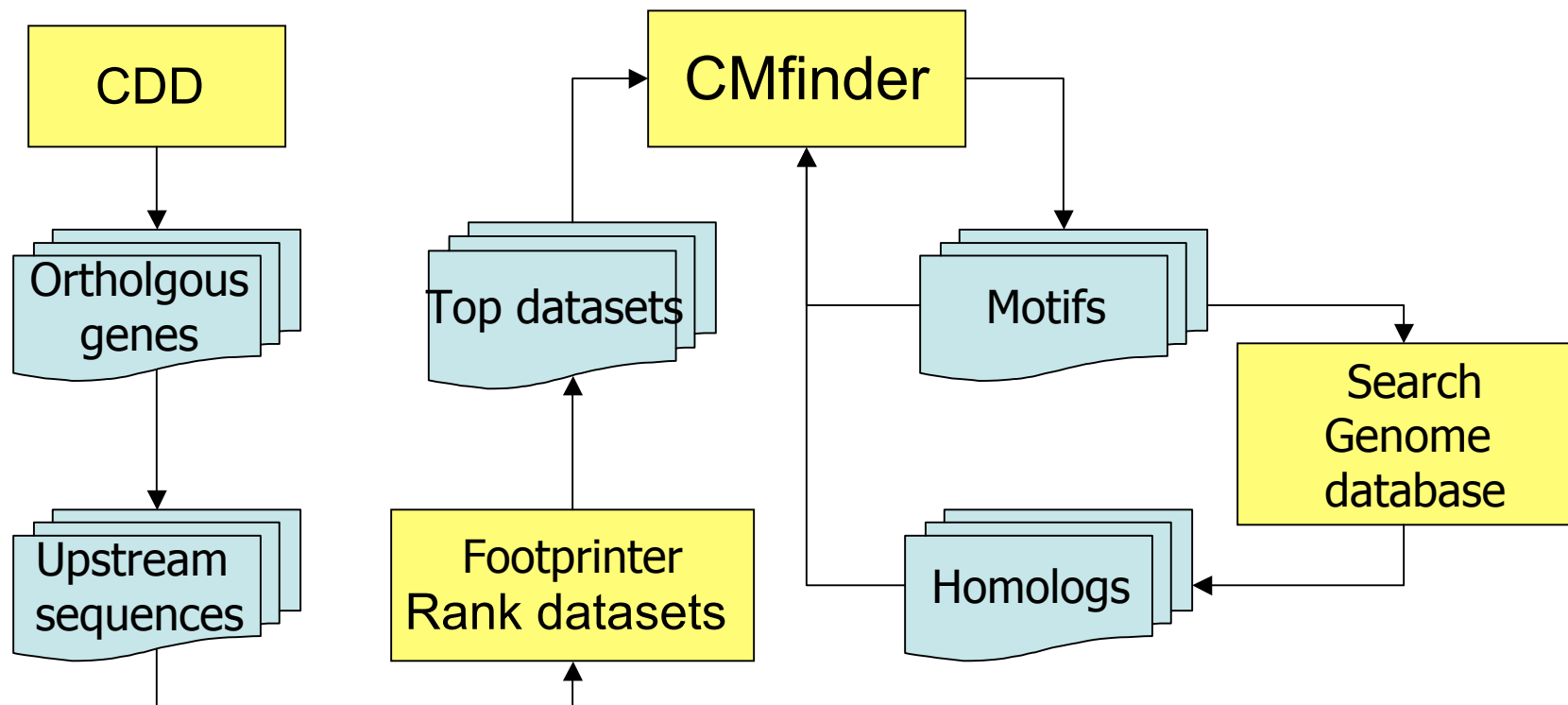
Find most promising genes, based on conserved sequence motifs (Footprinter)

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



Yao, Barrick, Weinberg, Neph, Breaker, Tompa and Ruzzo. A Computational Pipeline for High Throughput Discovery of cis-Regulatory Noncoding RNA in Prokaryotes. PLoS Computational Biology. 3(7): e126, July 6, 2007.

Genome Scale Search: Why

Most riboswitches, e.g., are present in ~5 copies per genome

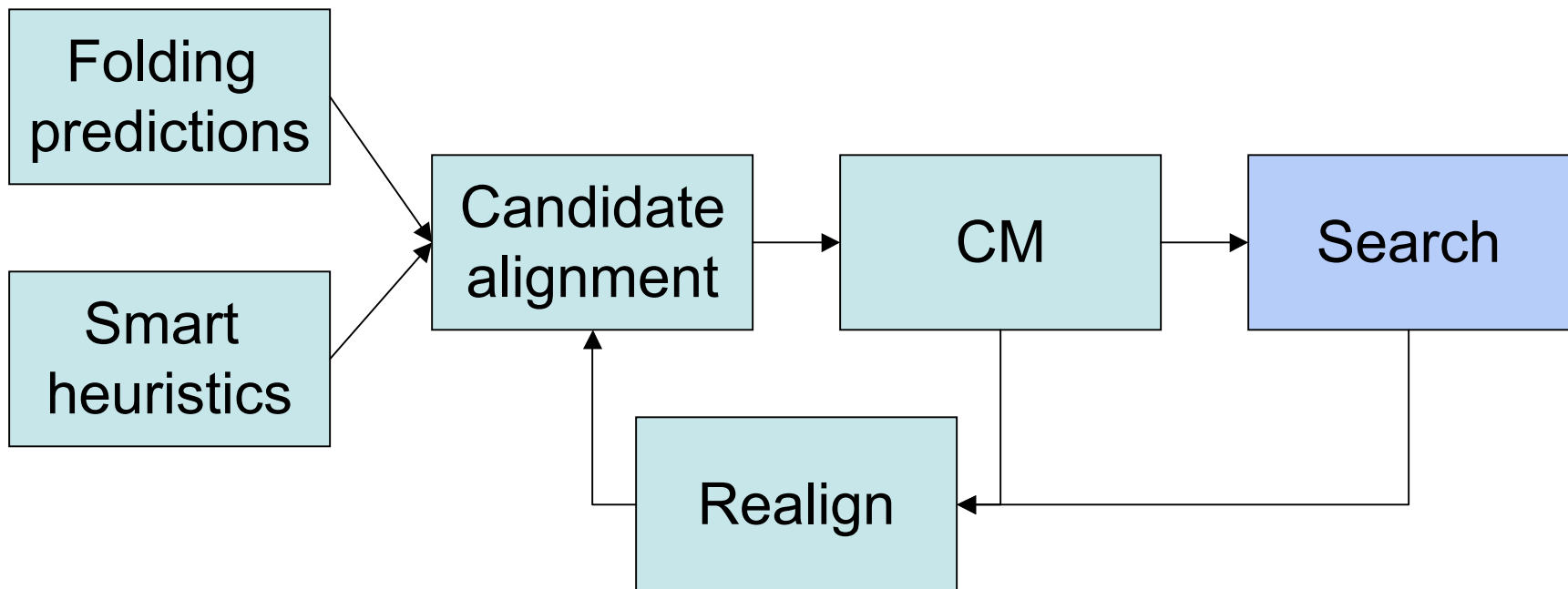
Throughout (most of) clade

More examples give better model, hence even more examples, fewer errors

More examples give more clues to function - critical for wet lab verification

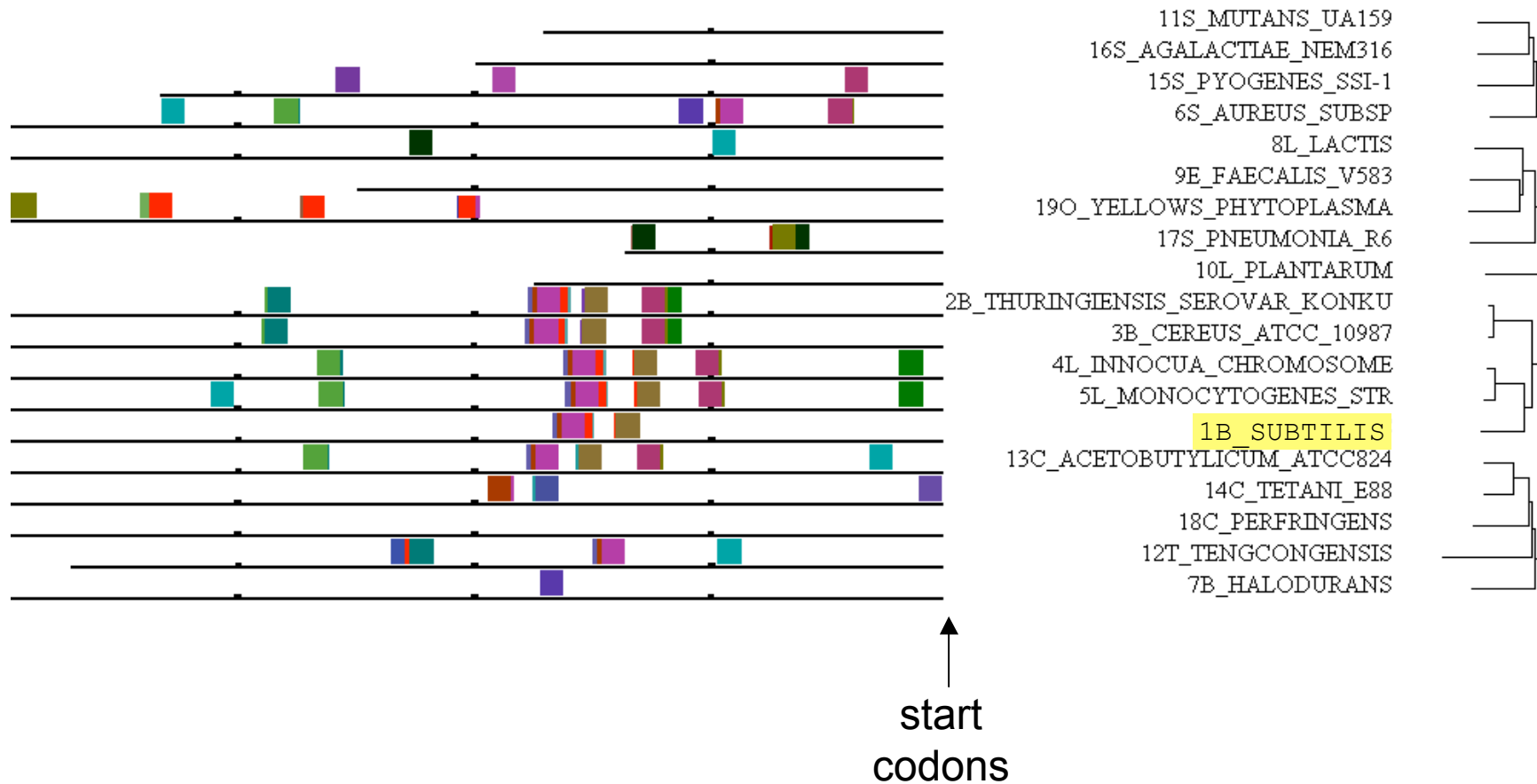
Genome Scale Search

CMfinder is directly usable for/with search



Footprinter finds patterns of conservation

Upstream of folC



A blind test

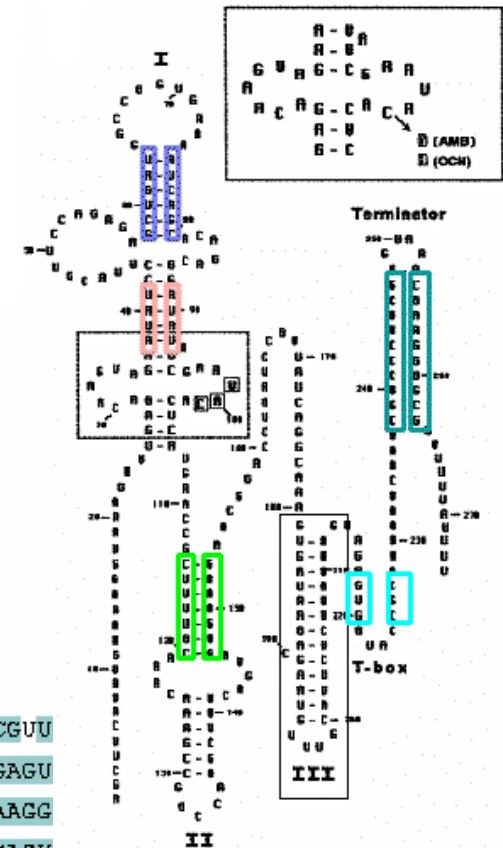
1ST genome scan: 234 sequences
 2ND genome scan: 447 sequences
The motif turned out to be T box
 Match to RFAM T box family: 299 OF 342
 False Positives: 89/148 are probable (upstream of annotated tRNA-synthetase genes)

```

AUAUC . CUUACGU . . UCCAGAGAGCUGAUGGCCGGUGAAA . AUCAGCACAGACGGAU AUAU
CAAU . GUCGUUUcUUAUAGAGAGUCGAUGGUUGGUGGAA . AUCGAUAG . . AAACAGUUUG
AAAAGUAGAACCG . AUCUAGCGAAUUGAGGAV . GGUGUGAGCUCAGUGC . GGAAAGCUUUU
CAAU . GUCGUUUcUUAUAGAGAGUCGAUGGUUGGUGGAA . AUCGAUAG . . AAACAGUUUG

CGAA . . UACACUCAUGAACC6 CUUUUGCAAACAAAGccggccaggcuuucAGUA . GUGAAAG
UGAA . . UCCAUCCUGGAU . . GGAUGUGGAAUAUCUuuuggauu . . . . . AGUAAACAUUC
AGAAAUC . ACUCUUGAGUU . UUCAUACGAAA . . CA . . . . . AGUAGUAAUGGA
UGAA . . UCCAUCCUGGAU . . GGAUGUGGAAUAUCUuuaugauu . . . . . AGUAAACAUUC

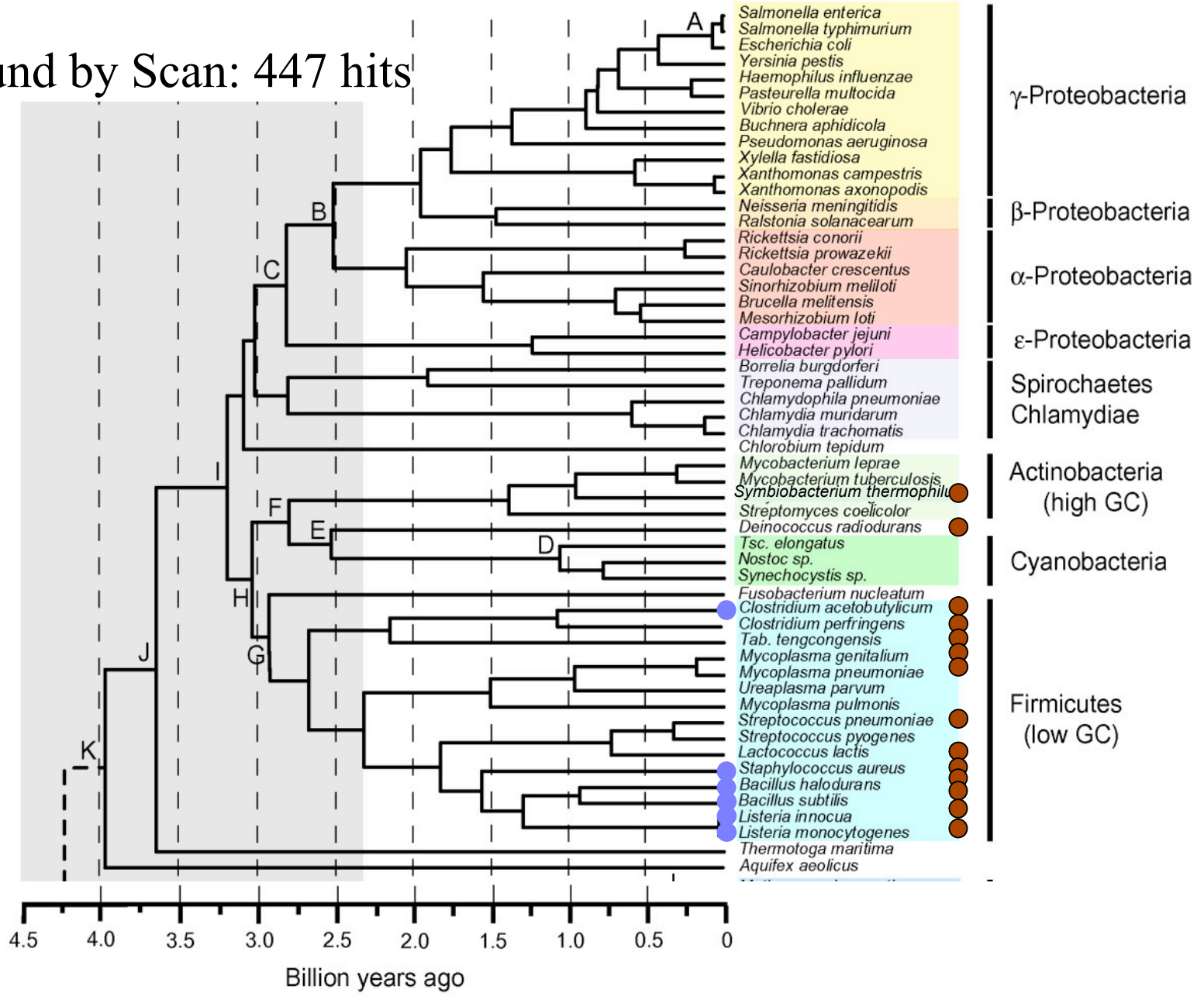
aCGGAC . CUGAUCGGUUAUCAGGCCAAAGUGGUACCGCGAUAUAUC AAUCGUCCCUUCGUGUAAa CGAAGGGGCGUU
. CGGUG . AAGAGCCGUAU . . . . . UCuASUGGCAACGCGG . . . . . GUUAACUCCCGUCCCUUUAUAU AGGGACGGGAGU
. CGGUUcAUC . UCCGUUAUCGAUCUUAUGUGGUACCGCGA . . . . . GUCUUCU CGUCCCUUUU . . . . . GGAUUAGAAGG
. CGGUG . AAGAGCCGUAU . . . . . UCuASUGGCAACGCGG . . . . . GUUAACUCCCGUCCCUUUAUAU AGGGACGGGAGU
    
```



tyrS T box structure

- CMfinder: 9 instances
- Found by Scan: 447 hits

Chloroflexus aurantiacus ● Chloroflexi
Geobacter metallireducens ● δ -Proteobacteria
Geobacter sulphurreducens ●



Results

Have analyzed most sequenced bacteria (~2005)

bacillus/clostridia

gamma proteobacteria

cyanobacteria

actinobacteria

firmicutes

Actino Results

8 of 10 Rfam families found

Rfam Family	Type (metabolite)	Rank	
THI	riboswitch (thiamine)	4	
ydaO-yuaA	riboswitch (unknown)	19	
Cobalamin	riboswitch (cobalamin)	21	
SRP_bact	gene	28	←
RFN	riboswitch (FMN)	39	
yybP-ykoY	riboswitch (unknown)	48	
gcvT	riboswitch (glycine)	53	
S_box	riboswitch (SAM)	401	
tmRNA	gene	Not found	←
RNaseP	gene	Not found	←

not cis-regulatory (got one anyway)

Results of genome scans

Top 115 datasets (some are redundant)

13 T box, 22 riboswitches, 30 ribosomal genes

RNase P, tRNA, CIRCE elements and other DNA binding sites

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

More Actino Results

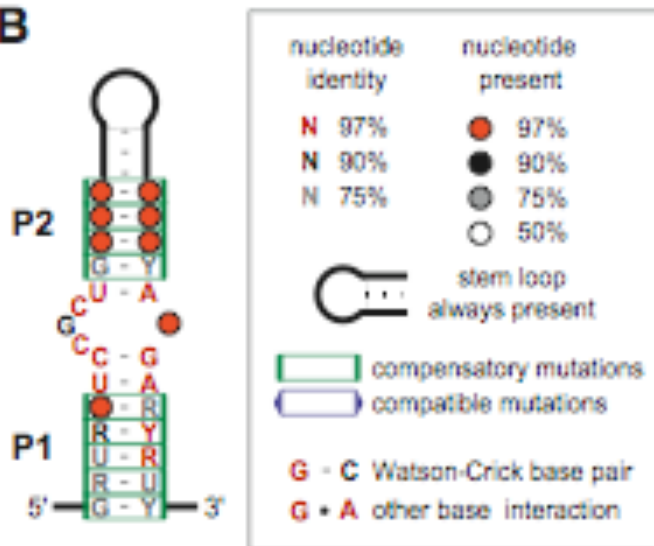
Many others (not in Rfam) are likely real;
of top 50:

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

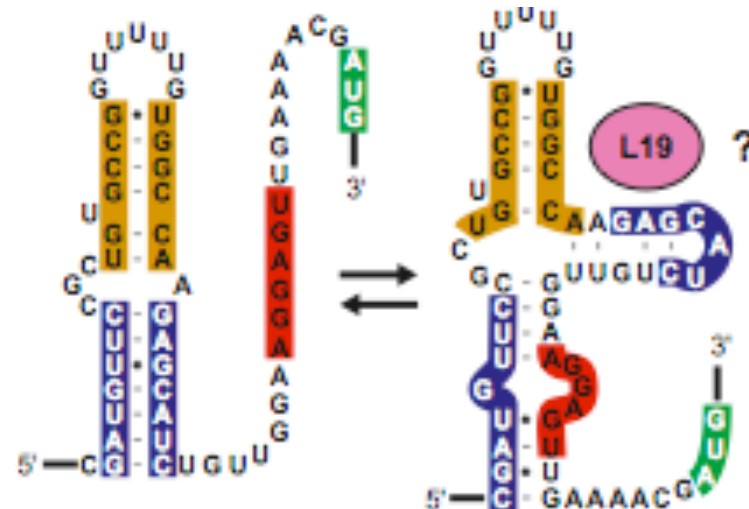
A mRNA leader

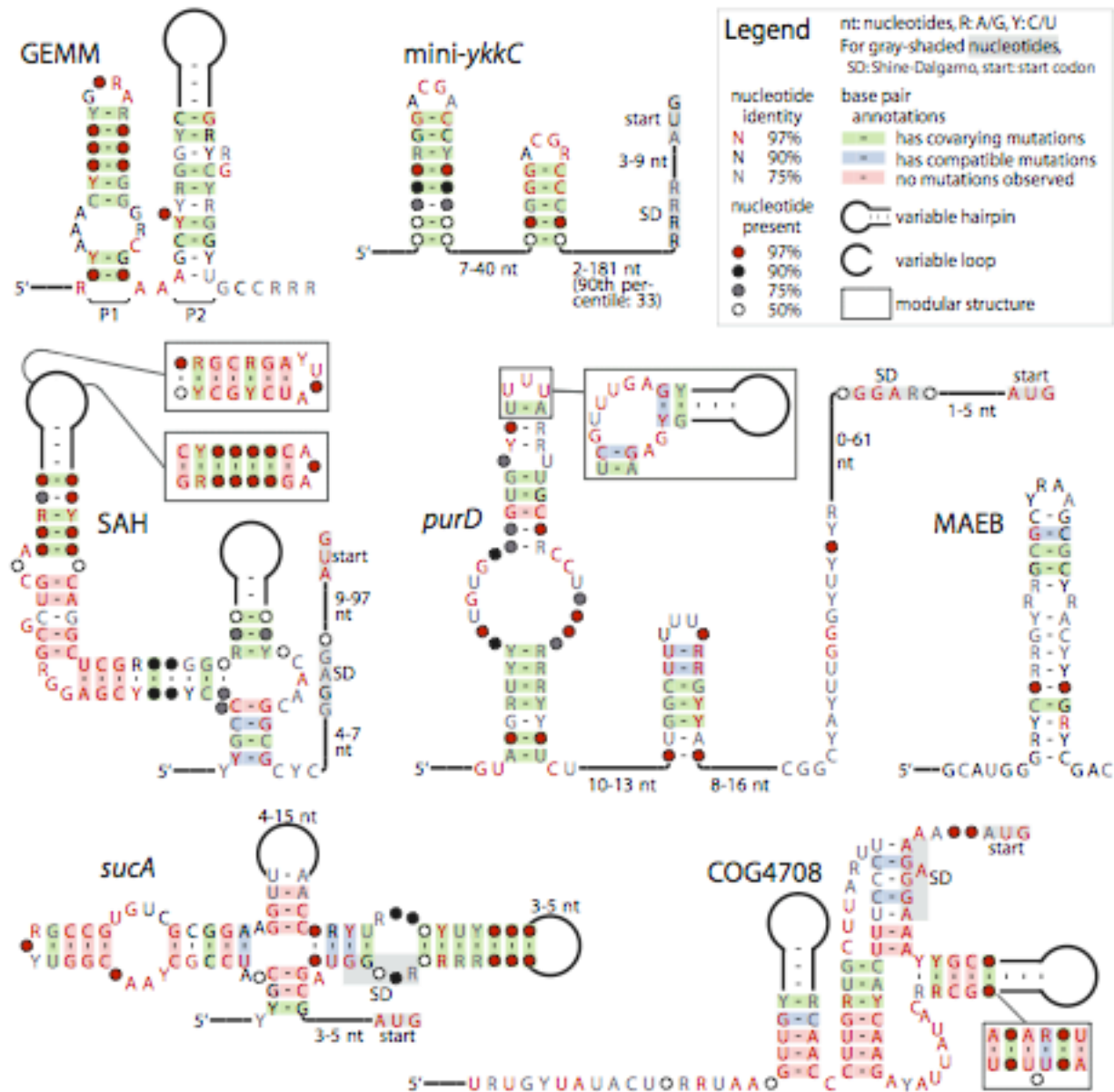


B



C mRNA leader switch?





Weinberg, Barrick, Yao, Roth, Kim, Gore, Wang, Lee, Block, Sudarsan, Neph, Tompa, Ruzzo and Breaker. Identification of 22 candidate structured RNAs in bacteria using the CMfinder comparative genomics pipeline. Nucl. Acids Res., July 2007 35: 4809-4819.

Ongoing & Future Work

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