“RNA sequence analysis using covariance models”

Eddy & Durbin

Nucleic Acids Research, 1994
vol 22 #11, 2079-2088

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

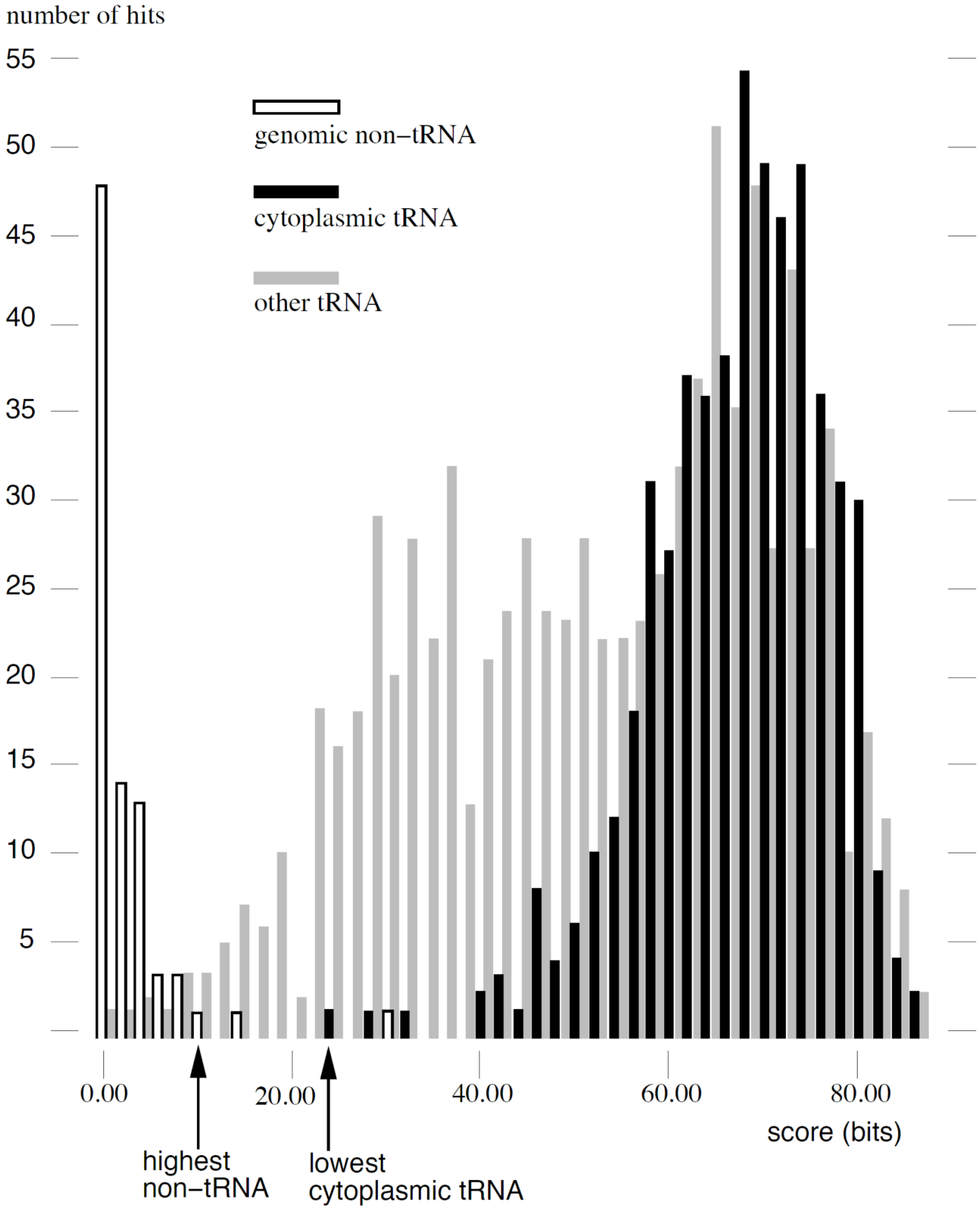
Main Results

- 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 llikelihood 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 AGAGCGCCAGACUGAAGAAAUACUUCGGUCAAGUUAUCUGGAG            GUCCUGUGUUCGAUCCACAGAAUUCGCA
DD6280    UCCGUGAUAGUUUAAU  GGUCAGAAUGGGCGCUUGUCG                  CGUGCCAG            A UCGGGGUUCAAUUCCCCGUCGCGGAGCCA
DX1661    CGCGGGGUGGAGCAGCCUGGU AGCUCGUCGGGCUCAUA                  ACCCGAAG            GUCGUCGGUUCAAAUCCGGCCCCCGCAACCA
DS6280    GGCAACUUGGCCGAGU  GGUUAAGGCGAAAGAUUAGAA                  AUCUUUU  GGGCUUUGCCCG CGCAGGUUCGAGUCCUGCAGUUGUCGCCA
```

U100:

```
DF6280    GCGGAUUUAGCUCAG UUGGGAGAGCGCCAGACU           GA   AG    AUCUGGA           GGUCCUGUGUUCGAUCCACAGAAUUCGCAcca
DF6280G   GCGGAUUUAGCUCAG UUGGGAGAGCGCCAGACUgaagaaauacuUCgguCAaguuAUCUGGA           GGUCCUGUGUUCGAUCCACAGAAUUCGCA
DD6280    UCCGUGAUAGUUUAA UGGUCAGAAUGGGCGCUU           GU   CG    CGUGCCA           GAU CGGGGUUCAAUUCCCCGUCGCGGAGcca
DX1661    CGCGGGGUGGAGCAGcCUGGUAGCUCGUCGGGCU           CA   UA    ACCCGAA           GGUCGUCGGUUCAAAUCCGGCCCCCGCAAcca
DS6280    GGCAACUUGGCCGAG UGGUUAAGGCGAAAGAUU           AG   AA    AUCUUUUgggcuuugcccG  CGCAGGUUCGAGUCCUGCAGUUGUCGcca
```

ClustalV:

```
DF6280    GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGA                  UCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA
DF6280G   GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGAAAUACUUCGGUCAAGUUAUCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCA
DD6280    UCCGUGAUAGUUUAAU       G GUCAGAAUGG GCG    CUUG  UCGCGUGCC   AGAUCGG GGUUCAAUUCCCCGUCGCGGAGCCA
DX1661    CGCGGGGUGGAGCAGC       CUGGUAGCUCGUCGGG    CUCA  UAACCCGA    AGGUCGUCGGUUCAAAUCCGGCCCCCGCAACCA
DS6280    GGCAACUUGGCCGAGUGGUUAAGGCGAAAGAUU  AGAAAUCUUUUGGGC  UUUGCCCG     CGCAGGUUCGAGUCCUGCAGUUGUCGCCA
```

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

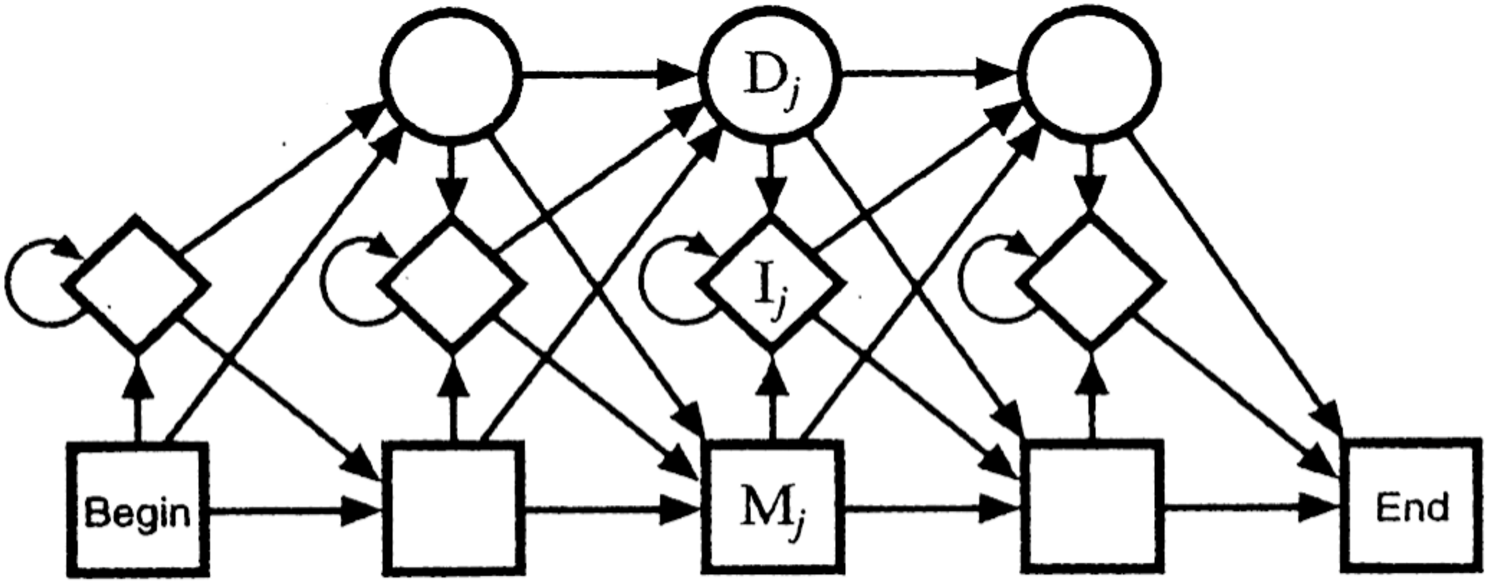


Figure 5.2 *The transition structure of a profile HMM.*

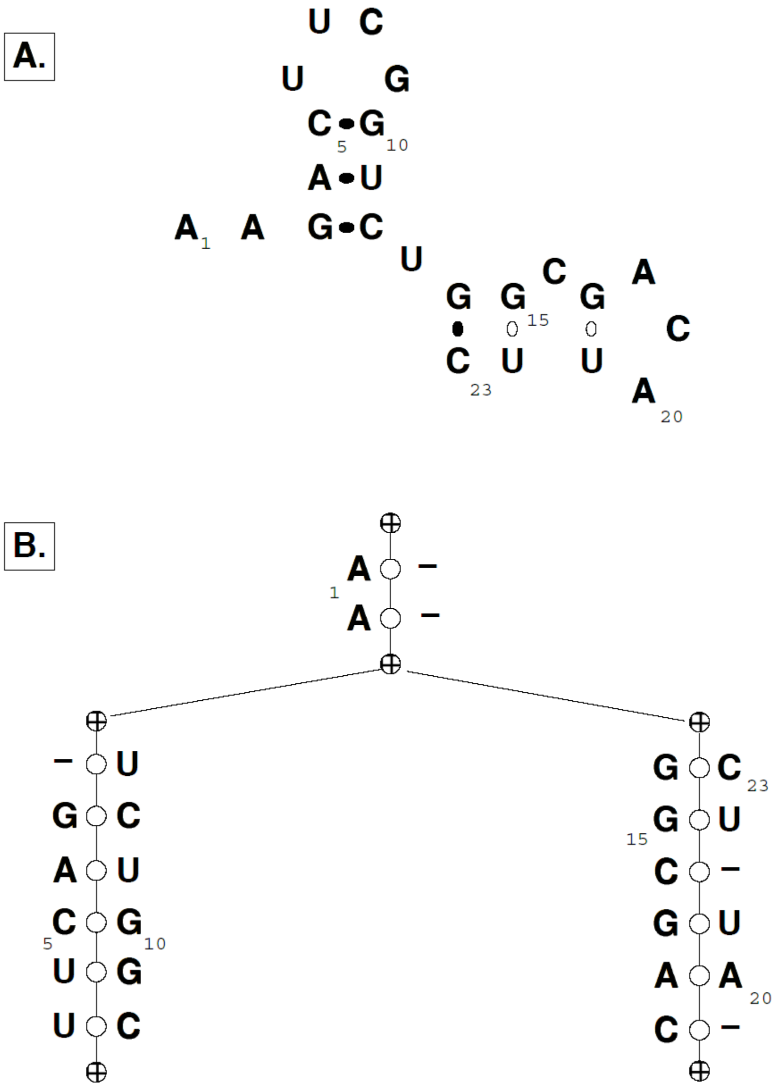
M_j: Match states (20 emission probabilities)

I_j: Insert states (Background emission probabilities)

D_j: Delete states (silent - no emission)

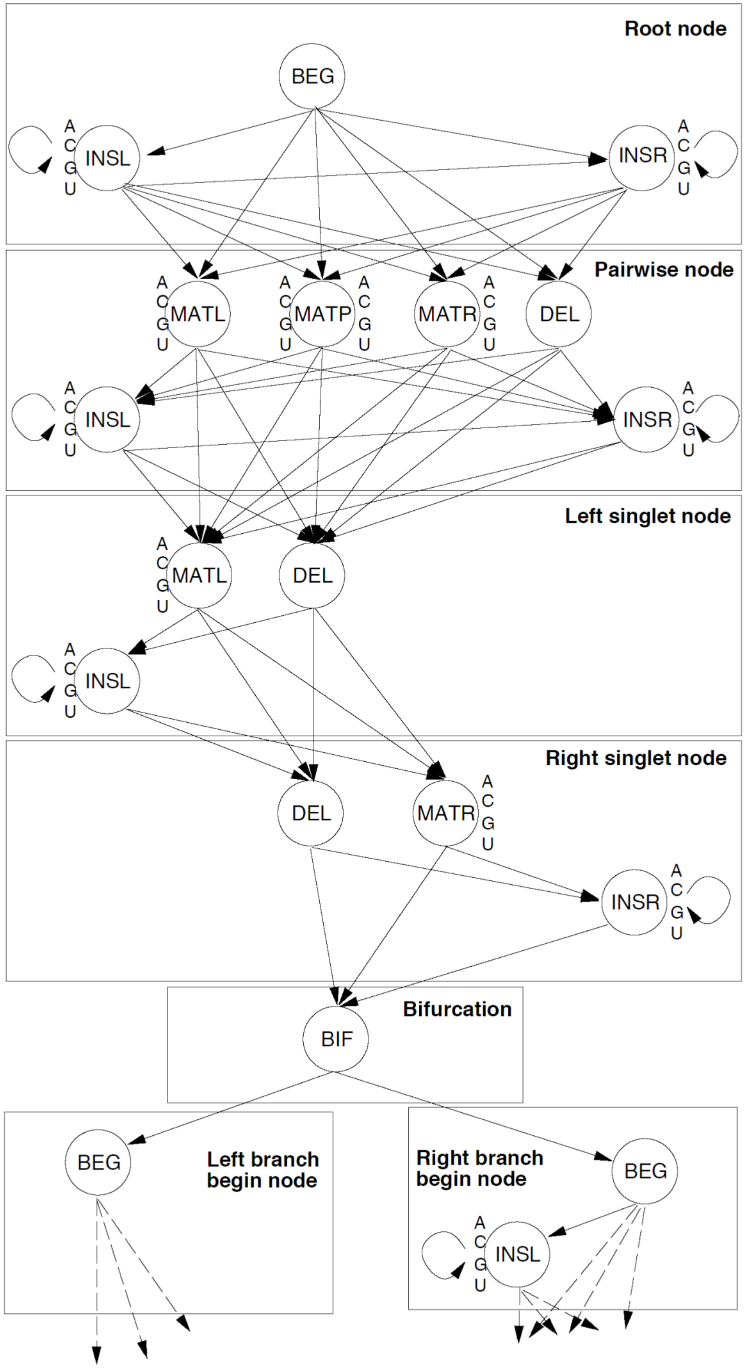
CM Structure

- 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

$x_i = i^{th}$ letter of input

$x_{ij} =$ substring $i,\ldots,j$ of input

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

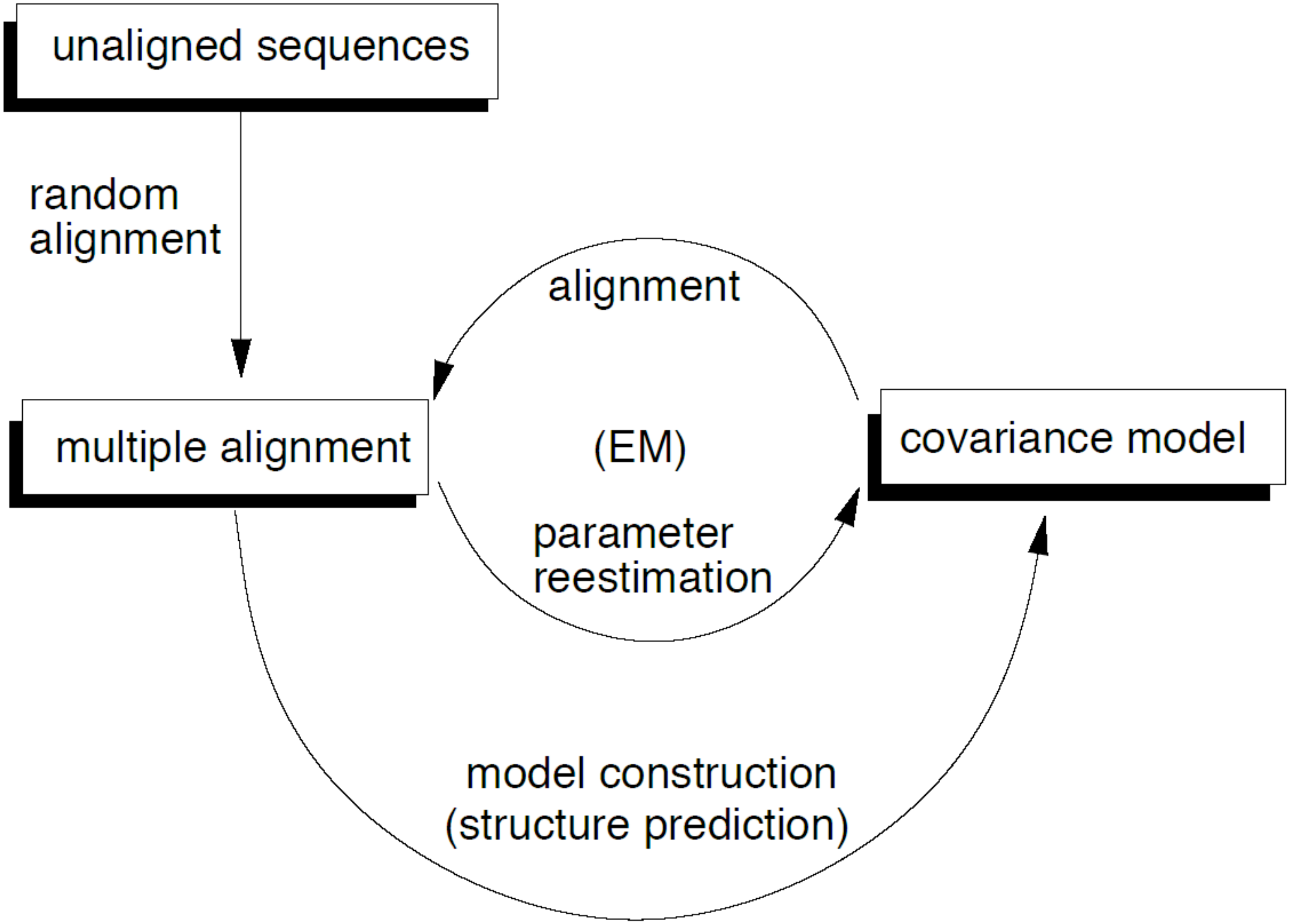
$E^y_{x_i,x_j} = P(\text{emission of } x_i, x_j \text{ from state } y)$

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

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

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

Model Training

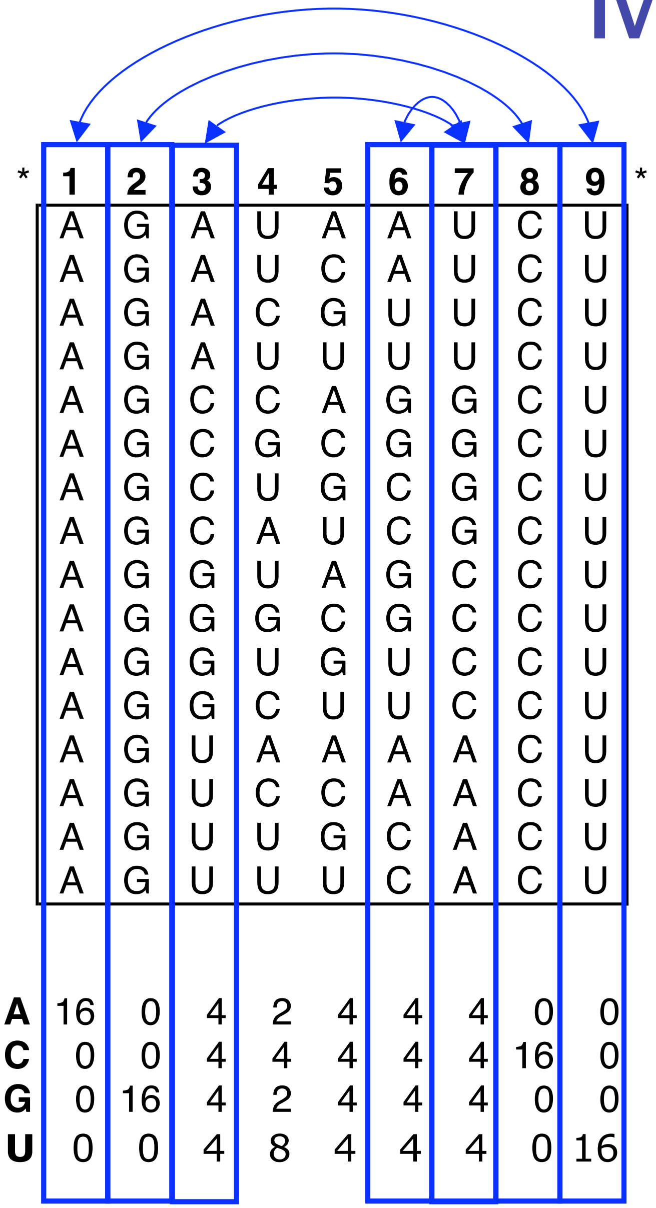


Mutual Information

$$M_{ij} = \sum_{xi,xj} f_{xi,xj} \log_2 \frac{f_{xi,xj}}{f_{xi} f_{xj}}; \quad 0 \leq M_{ij} \leq 2$$

- Max when *no* sequence conservation but perfect pairing
- MI = expected score gain from using a pair state
- Finding optimal MI, (i.e. optimal pairing of columns) is NP-hard(?)
- Finding optimal MI *without pseudoknots* can be done by dynamic programming

M.I. Example



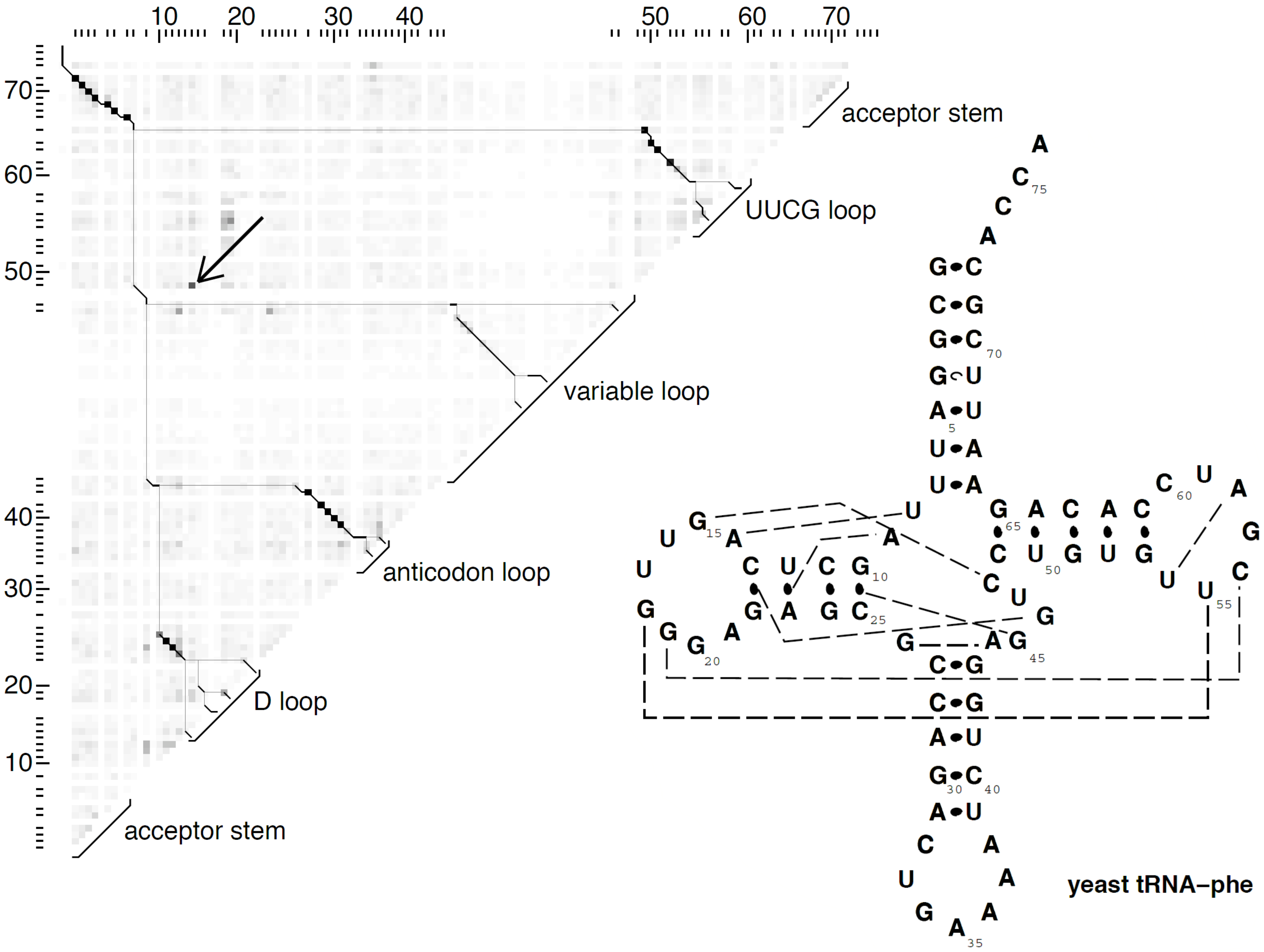
*	1	2	3	4	5	6	7	8	9	*
	A	G	A	U	A	A	U	C	U	
	A	G	A	U	C	A	U	C	U	
	A	G	A	C	G	U	U	C	U	
	A	G	A	U	U	U	U	C	U	
	A	G	C	C	A	G	G	C	U	
	A	G	C	G	C	G	G	C	U	
	A	G	C	U	G	C	G	C	U	
	A	G	C	A	U	C	G	C	U	
	A	G	G	U	A	G	C	C	U	
	A	G	G	G	C	G	C	C	U	
	A	G	G	U	G	U	C	C	U	
	A	G	G	C	U	U	C	C	U	
	A	G	U	A	A	A	A	C	U	
	A	G	U	C	C	A	A	C	U	
	A	G	U	U	G	C	A	C	U	
	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 and *might* be base-paired, but unclear whether they are. M.I. = 0

Cols 3 & 7: completely unconserved, but always W-C pairs, so seems likely that 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.


10
20
30
40
50
60
70
acceptor stem
UUCG loop
variable loop
anticodon loop
D loop
acceptor stem
yeast tRNA−phe

MI-Based Structure-Learning

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

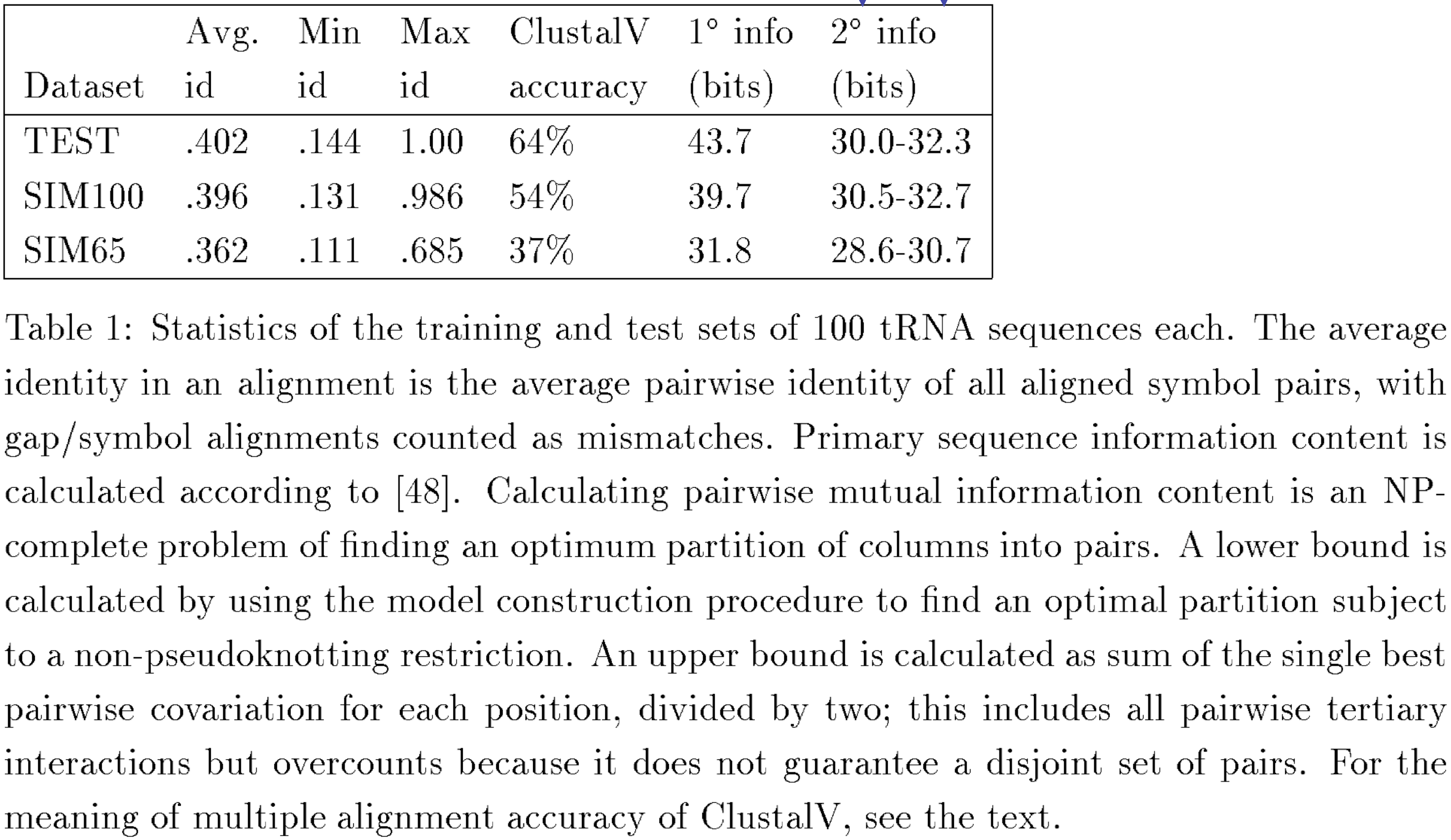
$$S_{i,j} = \max \begin{cases} S_{i+1,j} \\ S_{i,j-1} \\ S_{i+1,j-1} + M_{i,j} \\ \max_{i<j<k} S_{i,k} + S_{k+1,j} \end{cases}$$

- “just like Nussinov/Zucker folding”
- BUT, need enough data---enough sequences at right phylogenetic distance

Pseudoknots

disallowed allowed

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



Dataset	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.

			score	alignment
Model	training set	iterations	(bits)	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%

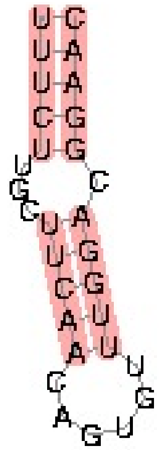
Table 2: Training and multiple alignment results from models trained from the trusted alignments (A models) and models trained from no prior knowledge of tRNA (U models).

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

Rfam



- 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.   GUUCCUGCUUCAACAGUGUUUGGAUGGAAC
Hom.sap.   UUUCUUC.UUCAACAGUGUUUGGAUGGAAC
Hom.sap.   UUUCCUGUUUCAACAGUGCUUGGA.GGAAC
Hom.sap.   UUUAUC..AGUGACAGAGUUCACU.AUAAA
Hom.sap.   UCUCUUGCUUCAACAGUGUUUGGAUGGAAC
Hom.sap.   AUUAUC..GGGAACAGUGUUUCCC.AUAAU
Hom.sap.   UCUUGC..UUCAACAGUGUUUGGACGGAAG
Hom.sap.   UGUAUC..GGAGACAGUGAUCUCC.AUAUG
Hom.sap.   AUUAUC..GGAAGCAGUGCCUUCC.AUAAU
Cav.por.   UCUCCUGCUUCAACAGUGCUUGGACGGAGC
Mus.mus.   UAUAUC..GGAGACAGUGAUCUCC.AUAUG
Mus.mus.   UUUCCUGCUUCAACAGUGCUUGAACGGAAC
Mus.mus.   GUACUUGCUUCAACAGUGUUUGAACGGAAC
Rat.nor.   UAUAUC..GGAGACAGUGACCUCC.AUAUG
Rat.nor.   UAUCUUGCUUCAACAGUGUUUGGACGGAAC
SS_cons    <<<<<...<<<<<......>>>>>.>>>>>
```

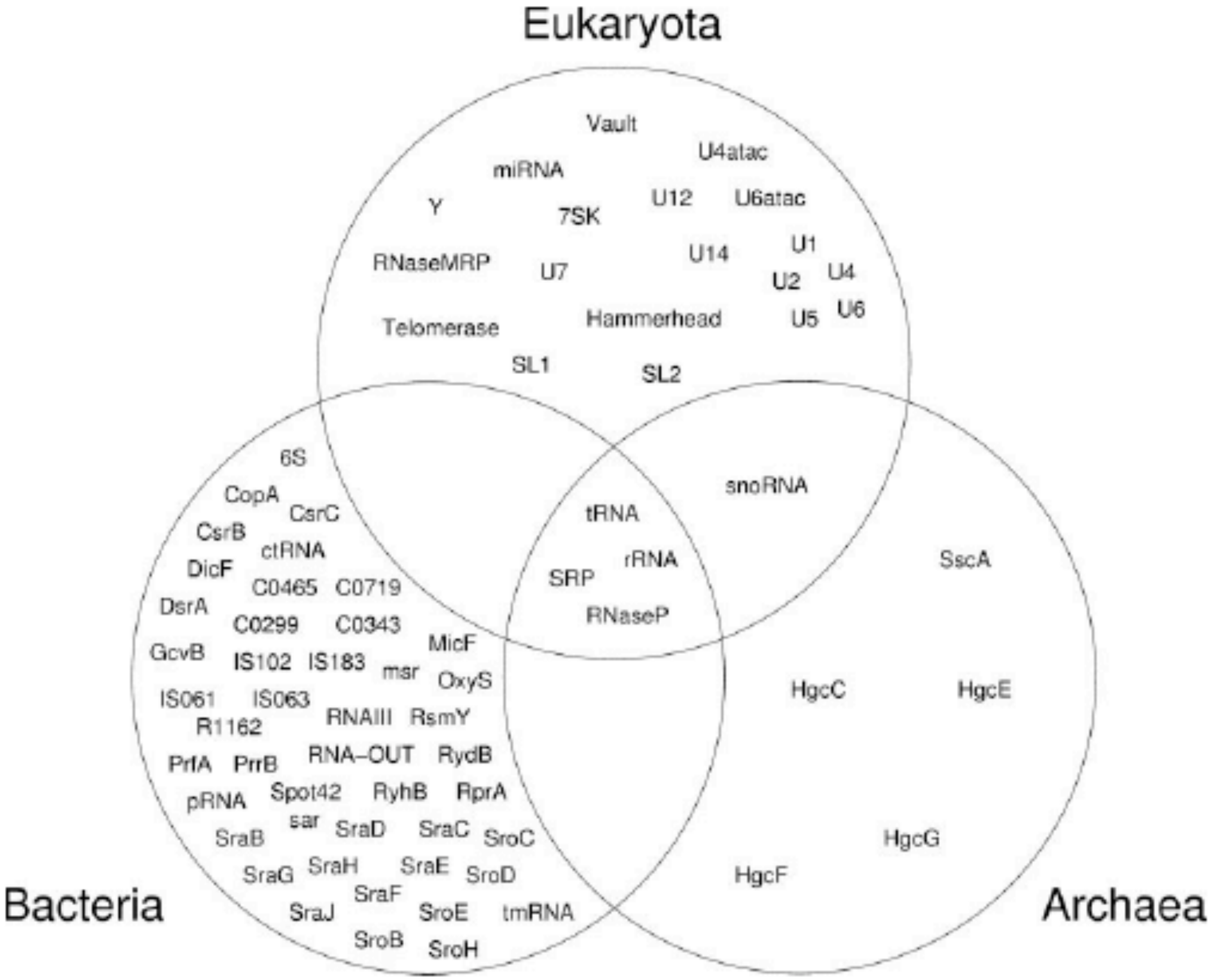


Figure 2. Taxonomic distribution of Rfam family members in the three kingdoms of life.

Rfam – key issues

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

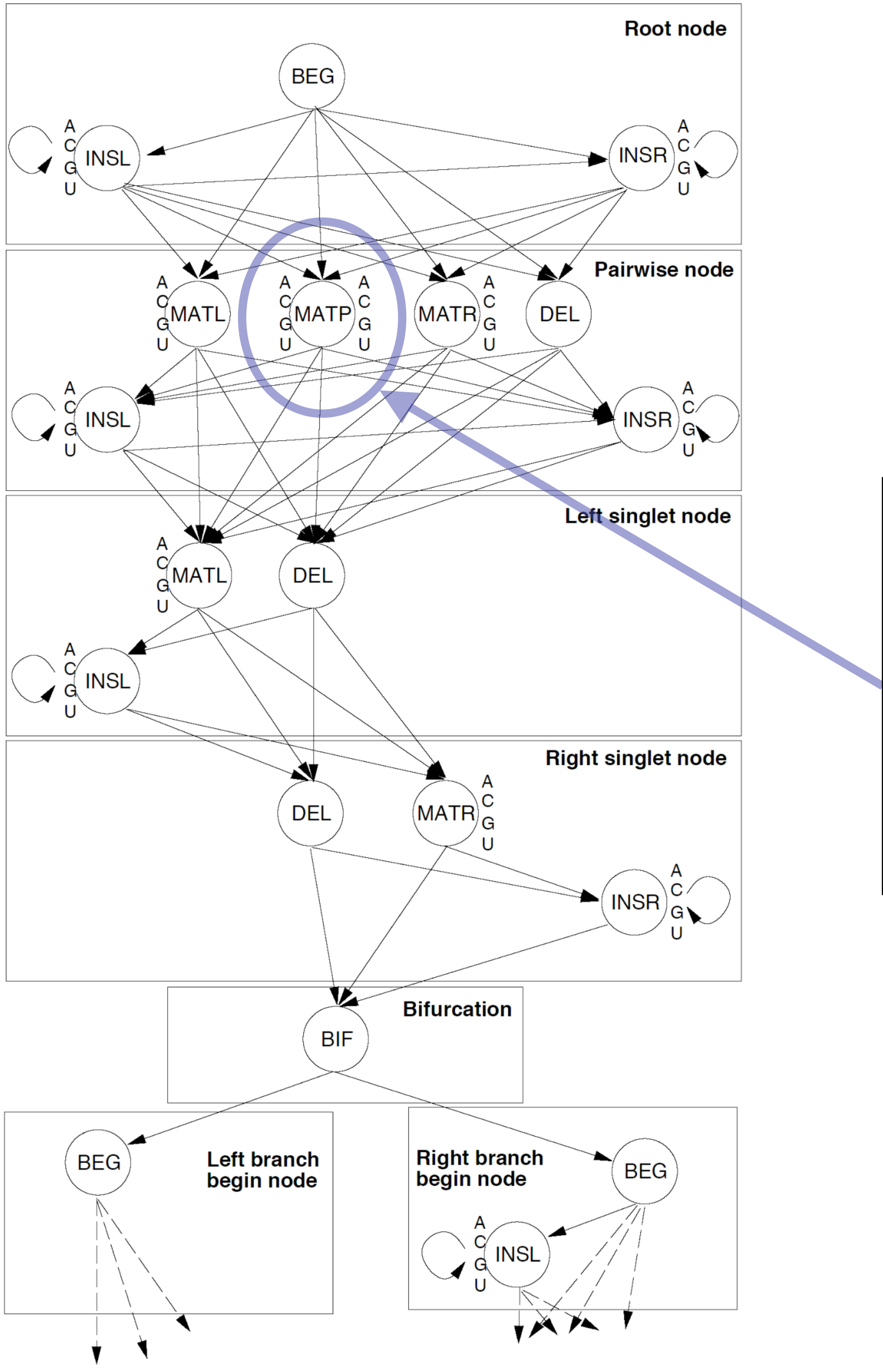
Faster Genome Annotation of Non-coding RNAs Without Loss of Accuracy

Zasha Weinberg

& W.L. Ruzzo

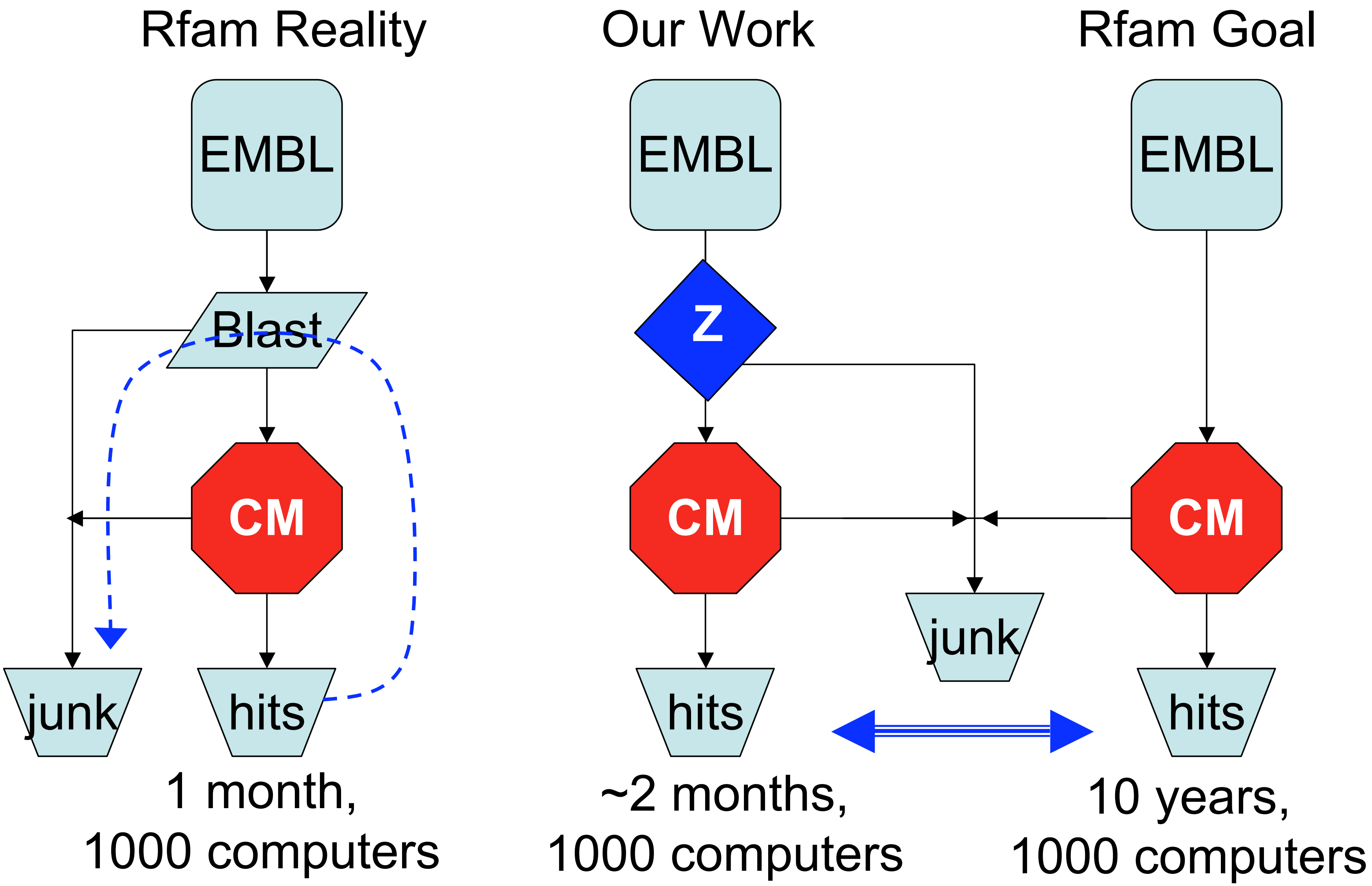
Recomb ‘04, ISMB ‘04

Covariance Model



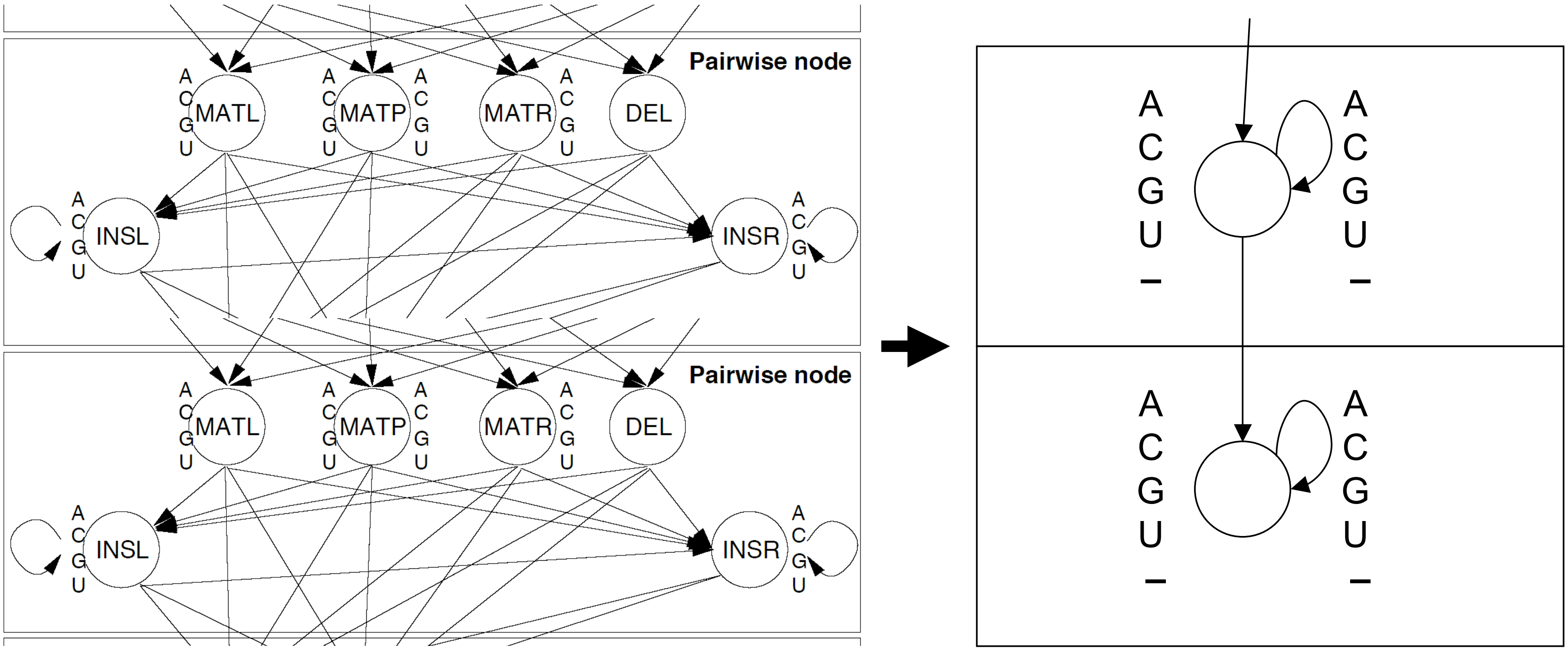
Key difference of CM vs HMM: Pair states emit paired symbols, corresponding to base-paired nucleotides; 16 emission probabilities here.

CM's are good, but slow

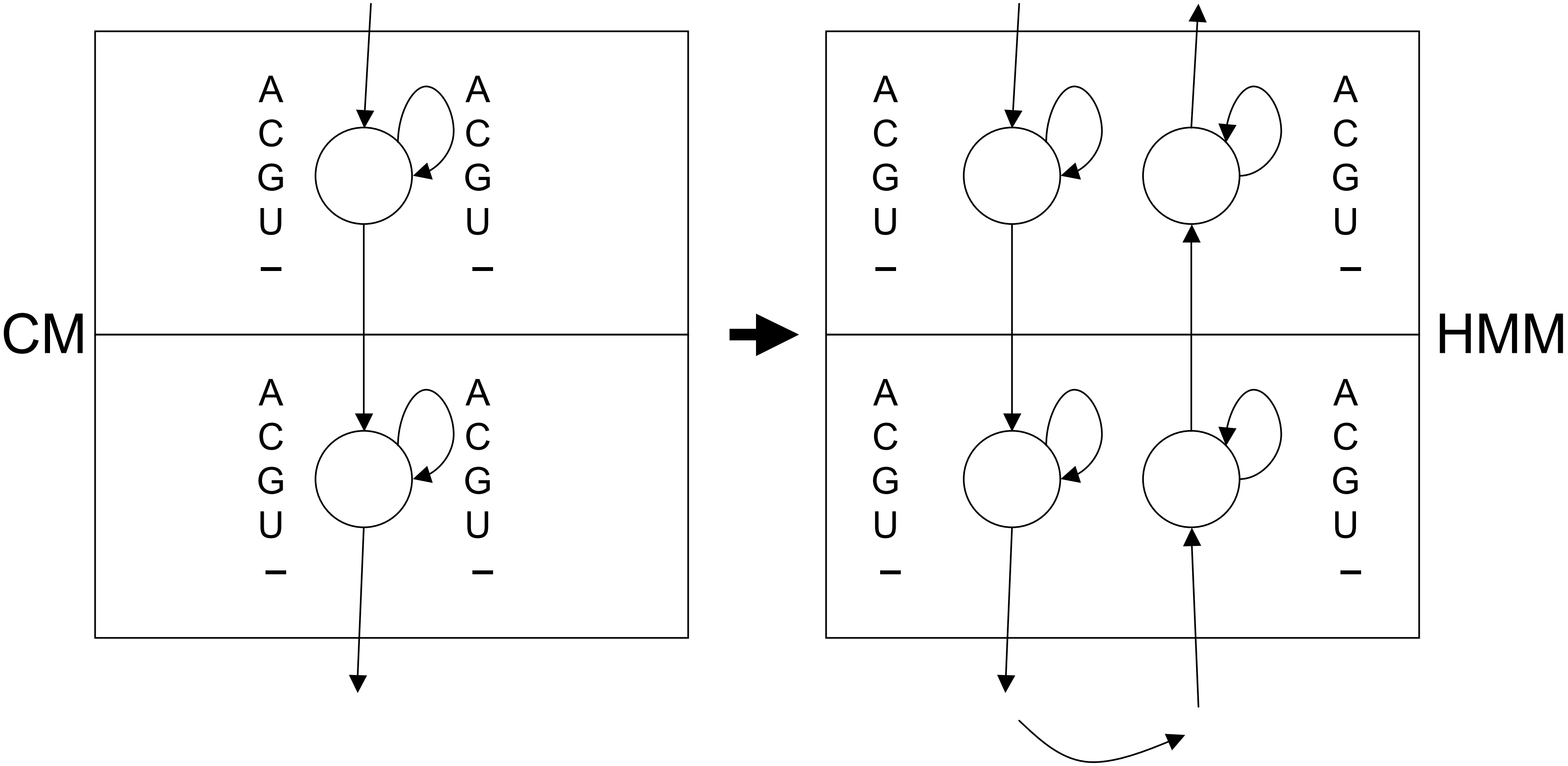


Oversimplified CM

(for pedagogical purposes only)



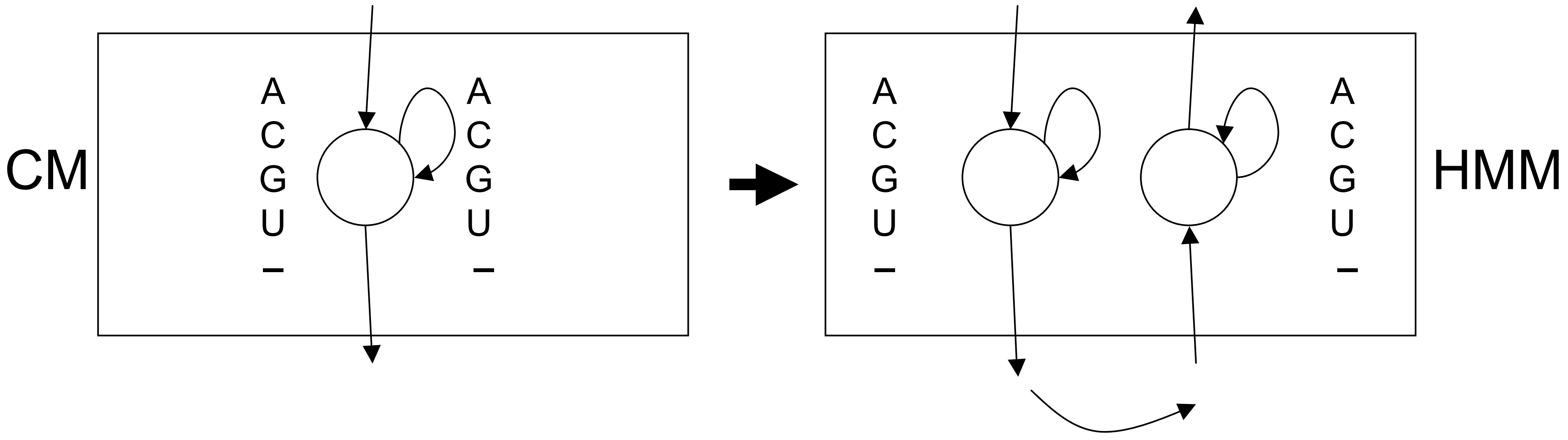
CM to HMM



25 emisions per state

5 emissions per state, 2x states

Key Issue: 25 scores → 10

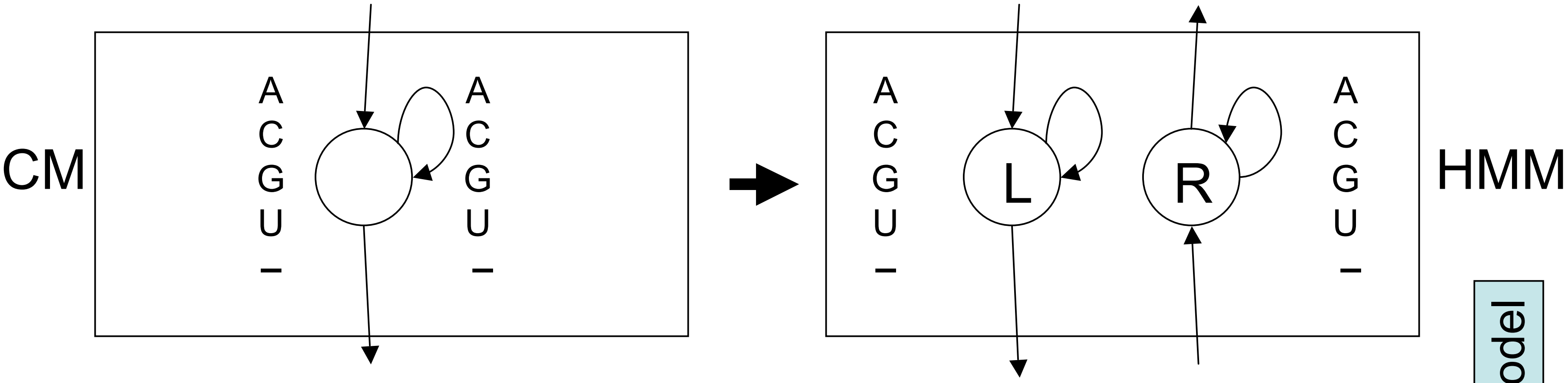


- Need: log Viterbi scores CM ≤ HMM

Viterbi/Forward Scoring

- Path π defines transitions/emissions
- Score(π) = product of “probabilities” on π
- 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(π) | π emits x}
  - Forward-score(x) = Σ{ score(π) | π emits x}

Key Issue: 25 scores → 10



- Need: log Viterbi scores CM ≤ HMM

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

NB:HMM not a prob. model

Rigorous Filtering

$P_{AA} \leq L_A + R_A$
$P_{AC} \leq L_A + R_C$
$P_{AG} \leq L_A + R_G$
$P_{AU} \leq L_A + R_U$
$P_{A-} \leq L_A + R_-$
…

- *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%

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:

  Viterbi-score(x) = max{ score(π) | π emits x }

  Forward-score(x) = Σ{ score(π) | π emits x }

- Expected Forward Score

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

  Under 0th-order background model

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

- Optimization:

  Minimize $E(L_i, R_i)$ subject to score L.I.s

  – This is heuristic ("forward↓ ⇒ Viterbi↓ ⇒ filter↓")

  – But still rigorous because "subject to score L.I.s"

Calculating $E(L_i, R_i)$

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

- 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)

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

Results: buried treasures

Name	# found BLAST + CM	# found rigorous filter + CM	# new
Pyrococcus 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

“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

    } for some hairpins
  - Idea 3: try lots of combinations of “some hairpins”
  - Idea 4: chain together several filters

Results: With additional work

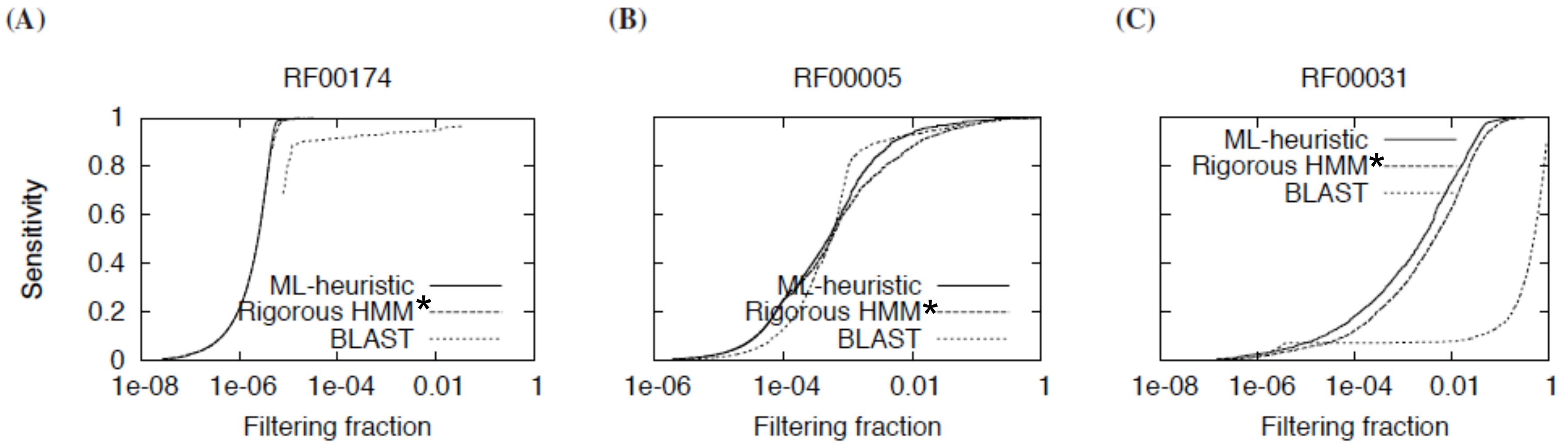
	# with BLAST+CM	# with rigorous filter series + CM	# new
Rfam tRNA	58609	63767	5158
Group II intron	5708	6039	331
tRNAscan-SE (human)	608	729	121
tmRNA	226	247	21
Lysine riboswitch	60	71	11

And more…

Heuristic Filters

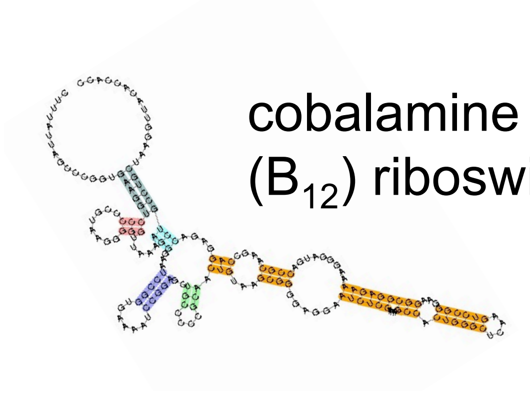
- Rigorous filters optimized for worst case
- Possible to trade improved speed for small loss in sensitivity?
- Yes – profile HMMs as before, but optimized for average case
- “ML heuristic”: train HMM from the infinite alignment generated by the CM
  - often 10x faster, modest loss in sensitivity

Heuristic Filters

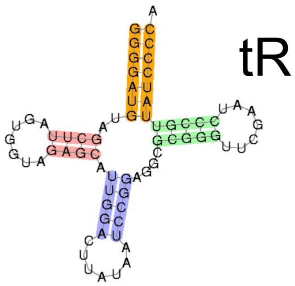


* rigorous HMM, not rigorous threshold

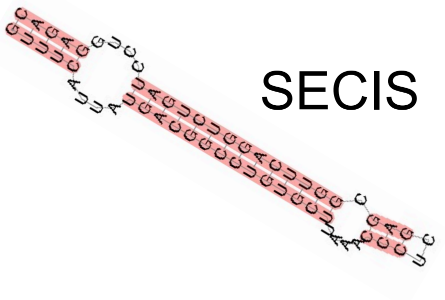
Fig. 1. Selected ROC-like curves. All plot sensitivity against filtering fraction, with filtering fraction in log scale. (A) RF00174 is typical of the other families; the ML-heuristic is slightly better than the rigorous profile HMM, and both often dramatically exceed BLAST. (B) Atypically, in RF00005, 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