Day 1

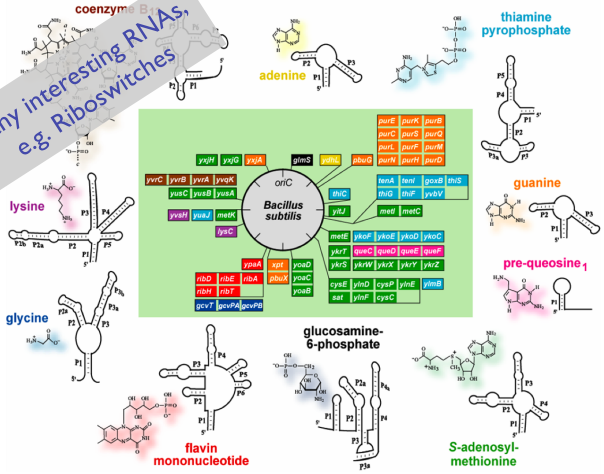
RNA Search and Motif Discovery

Genome 541
Intro to Computational
Molecular Biology

Many biologically interesting roles for RNA

RNA secondary structure prediction

Many interesting RNAs, e.g. Riboswitches



Structure Prediction

Approaches to Structure Prediction

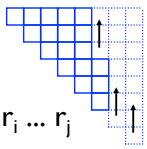
Maximum Pairing
- \+ works on single sequences
- \+ simple
- \- too inaccurate

Minimum Energy
- \+ works on single sequences
- \- ignores pseudoknots
- \- only finds "optimal" fold

Partition Function
- \+ finds all folds
- \- ignores pseudoknots

Structure Prediction

Nussinov: A Computation Order



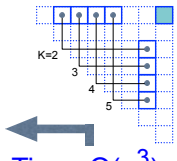
Or energy

B(i,j) = # pairs in optimal pairing of r_i ... r_j

B(i,j) = 0 for all i, j with i ≥ j-4; otherwise

B(i,j) = max of:

- B(i,j-1)
- max { B(i,k-1)+1+B(k+1,j-1) | i ≤ k < j-4 and r_k-r_j may pair}



Time: $O(n^3)$

Structure Prediction

Approaches, II

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

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

Physical experiments (x-ray crystalography, NMR)

Day 2

Day 1:
- Many biologically interesting roles for RNA
- RNA secondary structure prediction

Today:
- Covariance Models (CMs) represent RNA sequence/structure motifs
- Fast CM search

Computational Problems

- ~~How to predict secondary structure~~
- How to model an RNA "motif" (I.e., sequence/structure pattern)
- Given a motif, how to search for instances
- Given (unaligned) sequences, find motifs
- How to score discovered motifs
- How to leverage prior knowledge

Motif Description

RNA Motif Models

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

Model position-specific nucleotide preferences *and* base-pair preferences

Pro: accurate

Con: model building hard, search slow

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

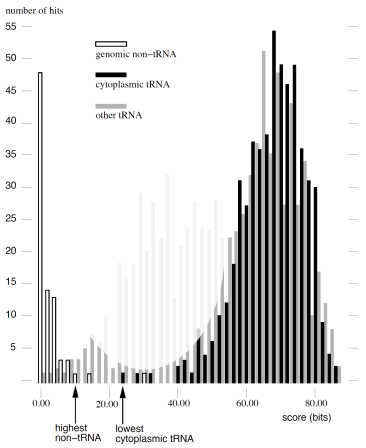
You set a score threshold

Anything above threshold → a "hit"

Scoring:

- "Forward" / "Inside" algorithm - sum over all paths
- Viterbi approximation - find single best path
  (Bonus: alignment & structure prediction)

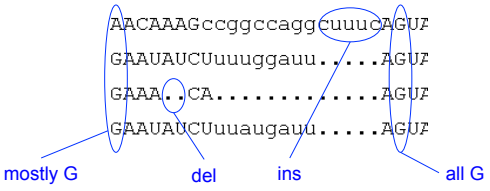
Example: searching for tRNAs



How to model an RNA "Motif"?

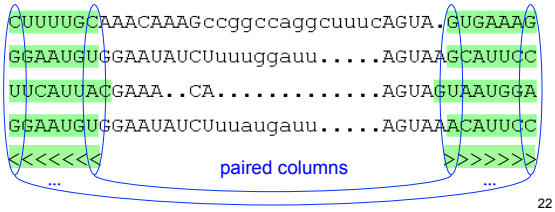
Conceptually, start with a profile HMM:

- 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



How to model an RNA "Motif"?

Add "column pairs" and pair emission probabilities for base-paired regions



Profile Hmm Structure

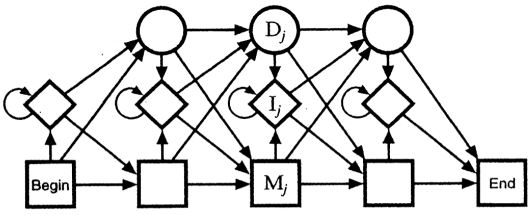


Figure 5.2 *The transition structure of a profile HMM.*

M_j: Match states (20 emission probabilities)
I_j: Insert states (Background emission probabilities)
D_j: Delete states (silent - no emission)

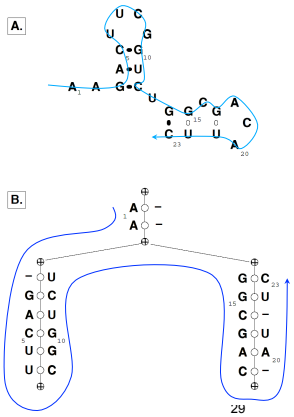
CM Structure

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)

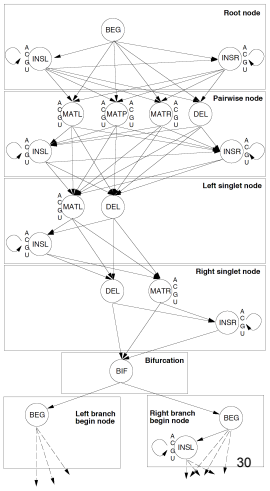


Overall CM Architecture

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
(the "inside" algorithm)

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

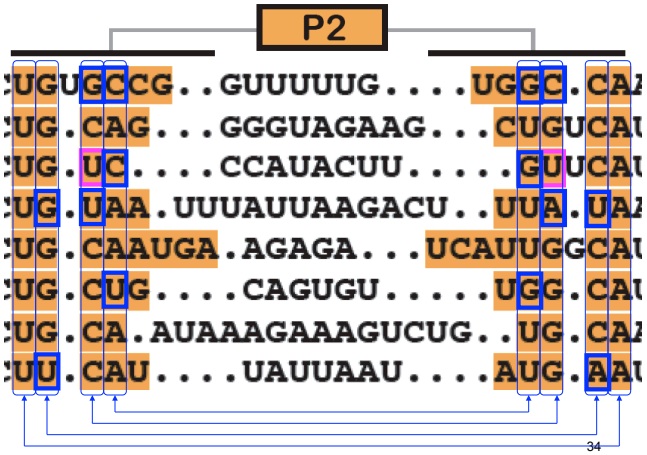
CM Viterbi Alignment
(the "inside" algorithm)

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

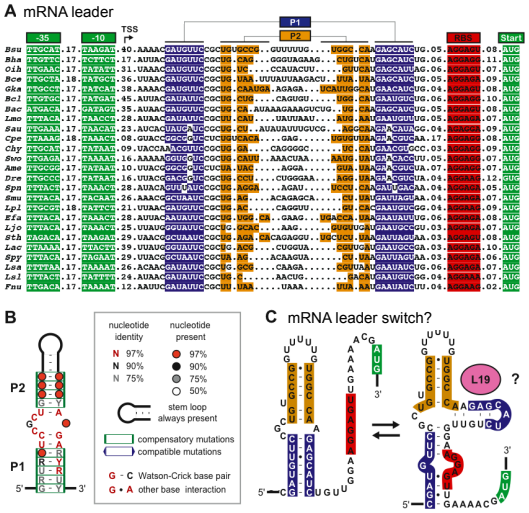
$$S^y_{ij} = \begin{cases} \max_z [S^z_{i+1,j-1} + \log T_{yz} + \log E^y_{x_i,x_j}] & \text{match pair} \\ \max_z [S^z_{i+1,j} + \log T_{yz} + \log E^y_{x_i}] & \text{match/insert left} \\ \max_z [S^z_{i,j-1} + \log T_{yz} + \log E^y_{x_j}] & \text{match/insert right} \\ \max_z [S^z_{i,j} + \log T_{yz}] & \text{delete} \\ \max_{i<k\le j} [S^{y_{left}}_{i,k} + S^{y_{right}}_{k+1,j}] & \text{bifurcation} \end{cases}$$

Time O(qn³), q states, seq len n

compare: O(qn) for profile HMM



Covariation is strong evidence for base pairing



Mutual Information

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

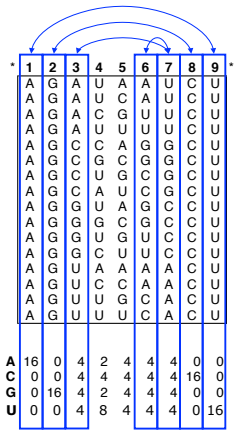
Max when *no* seq conservation but perfect pairing

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	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.

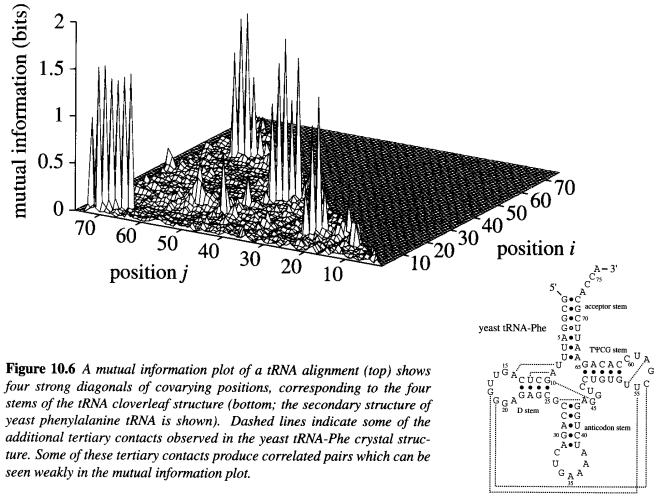


Figure 10.6 *A mutual information plot of a tRNA alignment (top) shows four strong diagonals of covarying positions, corresponding to the four stems of the tRNA cloverleaf structure (bottom; the secondary structure of yeast phenylalanine tRNA is shown). Dashed lines indicate some of the additional tertiary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

Primary vs Secondary Info

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

disallowing / allowing pseudoknots

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

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

Slightly different evaluation criteria

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

tRNAScanSE

Uses 3 older heuristic tRNA finders as prefilter

Uses CM built as described for final scoring

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

Used in all genome annotation projects

An Important Application: Rfam

Rfam – an RNA family DB

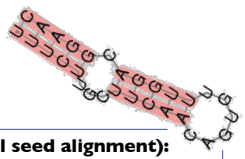
Griffiths-Jones, et al., NAR '03, '05, '08

Biggest scientific computing user in Europe - 1000 cpu cluster for a month per release

Rapidly growing:

			DB size:
Rel	1.0, 1/03:	25 families, 55k instances	~8GB
Rel	7.0, 3/05:	503 families, 363k instances	
Rel	9.0, 7/08:	603 families, 636k instances	
Rel	9.1, 1/09:	1372 families, 1148k instances	
Rel	10.0, 1/10:	1446 families, 3193k instances	~160GB

Example Rfam Family



Input (hand-curated):

- MSA "seed alignment"
- SS_cons
- Score Thresh T
- Window Len W

Output:

- CM
- scan results & "full alignment"
- phylogeny, etc.

IRE (partial seed alignment):

```
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   <<<<<...<<<<<.....>>>>>.>>>>>
```