

Can we distinguish RNA genes from random sequences?

István Miklós

***Theoretical Biology and Ecology Group,
Hungarian Academy of Science and Eötvös University of Science***

(Joint work with Irmtraud M. Meyer, EMBL-EBI, Cambridge and Borbála Nagy, ELTE-TTK, Budapest)

Collegium Budapest, 2004-11-24

Outline

- RNA sequences, secondary structure
- Dynamic programming, Nussinov algorithm
- Context Free Grammars and RNA secondary structures
- Zuker-Tinoco-Turner energy model and the Zuker-Sankoff algorithm
- Previous results on identifying RNA genes
 - tRNAscan
 - The Rfam database
 - Negative results
- Algebraic dynamic programming
- Moments of the Boltzmann distribution
- Co-transcriptional folding

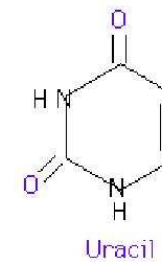
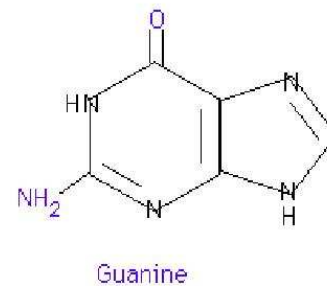
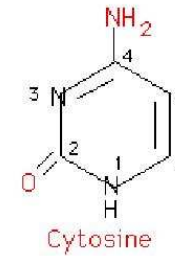
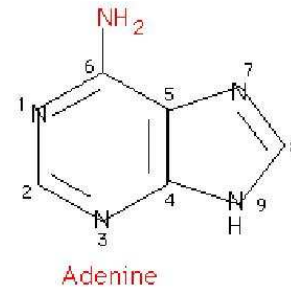
RNA Sequences

RNA:

A long sequence on a sugar-phosphate bone, 4 types of nucleotides.

Mathematical point of view:

A finite sequence over an alphabet $\{A, C, G, U\}$

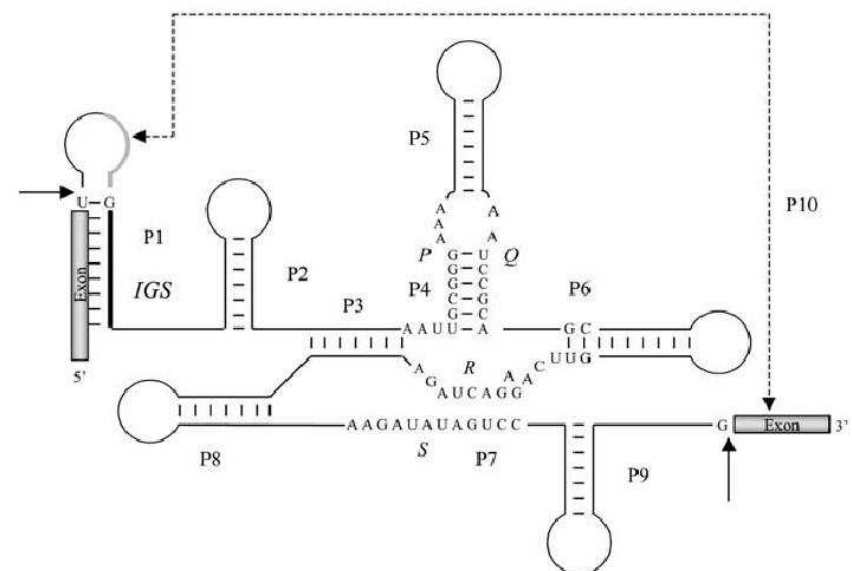


Secondary structure:

Base pairs: A-U, C-G, G-C, U-A,

Nested and juxtaposed pairs

Others form *pseudoknots*.



Dynamic programming

Easiest example: calculating the first n factorials.

$$1! = 1$$

$$2! = 1 \times 2$$

$$3! = 1 \times 2 \times 3$$

\vdots

$$n! = 1 \times 2 \times 3 \times \dots \times n$$

$O(n^2)$ time

$$1! = 1$$

$$2! = 1 \times 2$$

$$3! = 2 \times 3$$

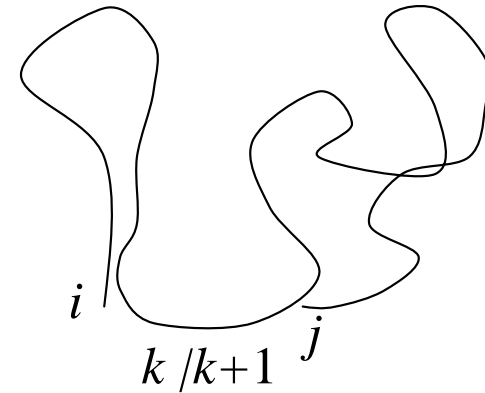
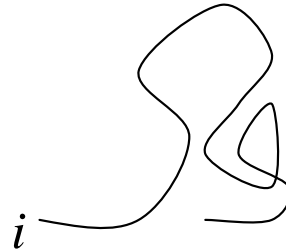
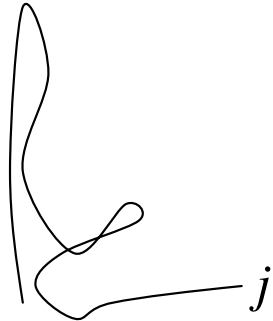
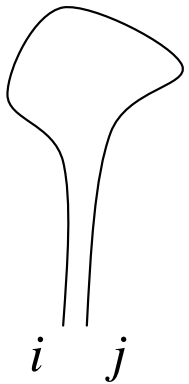
\vdots

$$n! = (n-1) \times n$$

$O(n)$ time

Nussinov algorithm

Find the secondary structure having the maximum number of basepairs



$$P(i+1, j-1) + 1 \times (s_i \text{ bp? } s_j)$$

$$P(i, j-1)$$

$$P(i+1, j)$$

$$P(i, k) + P(k+1, j)$$

$s_i \text{ bp? } s_j = 1$, if the i th character can form a basepair with the j th one, otherwise 0

Nussinov algorithm (cont'd)

For every $j-i = 2$ and $j-i = 3$, $P(i, j) = 0$

for $4 < j \leq l$

for $j-3 > i > 0$

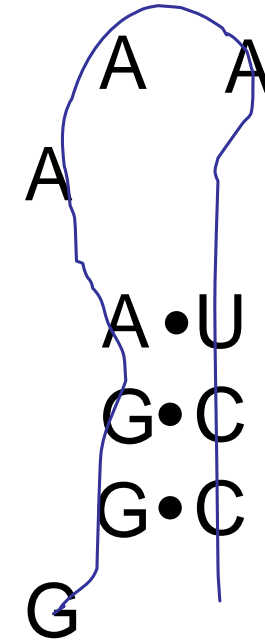
$$P(i, j) = \max \begin{cases} P(i+1, j-1) + 1 \times (s_i \text{ bp } ? s_j) \\ P(i, j-1) \\ P(i+1, j) \\ \max_{i+3 < k < j-4} \{P(i, k) + P(k+1, j)\} \end{cases}$$

Get the maximum with a traceback

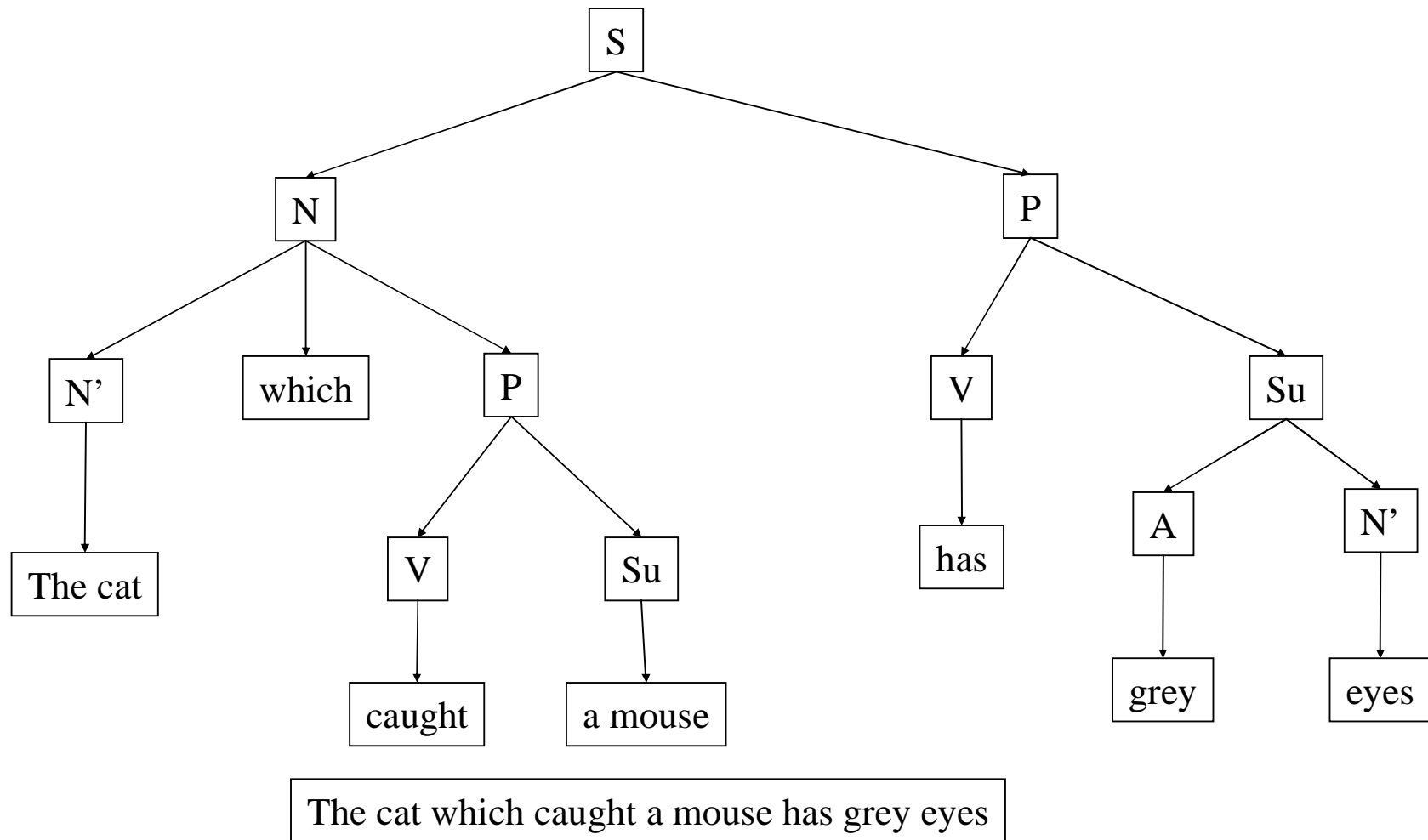
Nussinov algorithm (example)

Find the optimal structure of GGGAAAUCC!

	G	G	G	A	A	A	A	U	C	C
G	X	X	0	0	0	0	0	1	2	3
G	X	X	X	0	0	0	0	1	2	3
G	X	X	X	X	0	0	0	1	2	2
A	X	X	X	X	X	0	0	1	1	1
A	X	X	X	X	X	X	0	0	0	0
A	X	X	X	X	X	X	X	0	0	0
A	X	X	X	X	X	X	X	X	0	0
U	X	X	X	X	X	X	X	X	X	0
C	X	X	X	X	X	X	X	X	X	X
C	X	X	X	X	X	X	X	X	X	X



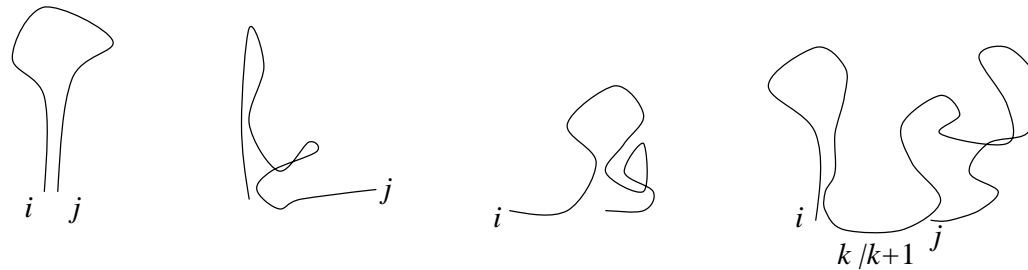
(Stochastic) Context Free Grammars



Starting non-terminal: S
Non terminals: N,P, etc.
terminals: which, a mouse, etc.

Derivations (Rules):
 $S \rightarrow NP$
 $Su \rightarrow a\ mouse \mid AN'$
etc.

Nussinov as (S)CFG



$$\begin{aligned}
 S &\rightarrow aSu \mid cSg \mid gSc \mid uSa \mid gSu \mid uSg \mid \\
 &Sa \mid Sc \mid Sg \mid Su \mid \\
 &aS \mid cS \mid gS \mid uS \mid \\
 &SS \mid \\
 &\varepsilon
 \end{aligned}$$

Each line stands for one case in the Nussinov algorithm.

Each derivation corresponds to a secondary structure

However, a secondary structure can be derived in several ways, hence it is an ambiguous grammar. There is an unambiguous version, not discussed here.

CYK algorithm

It gives the most probable derivation of a sequence.

The grammar must be given in Chomsky normal form

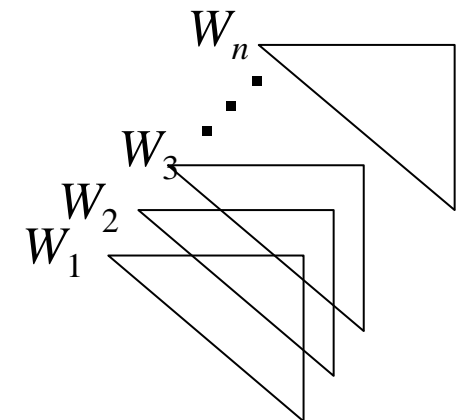
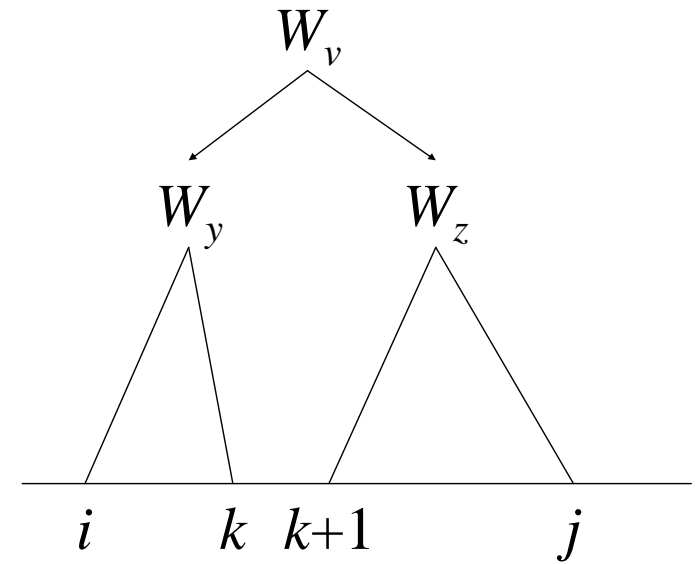
$$P(W_v \rightarrow W_y W_z) = t_v(y, z), \quad P(W_v \rightarrow a) = e_v(a)$$

Initialising:

$$\alpha_{\max}(i, i, v) = e_v(a_i)$$

Recursion:

$$\alpha_{\max}(i, j, v) = \max_{y, z, i \leq k < j} \{ \alpha_{\max}(i, k, y) t_v(y, z) \alpha_{\max}(k + 1, j, z) \}$$



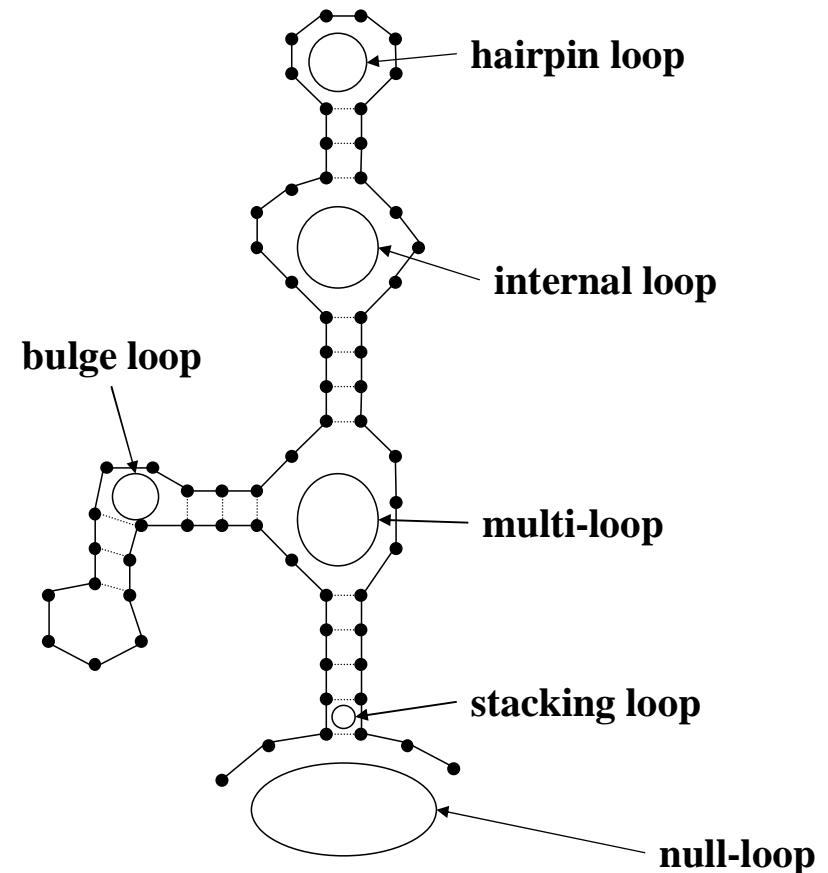
Zuker-Tinoco-Turner energy model

K-loop decomposition:

The free energy of a structure is the sum of free energies of loops

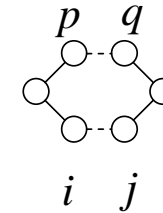
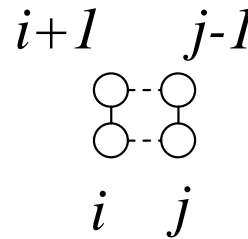
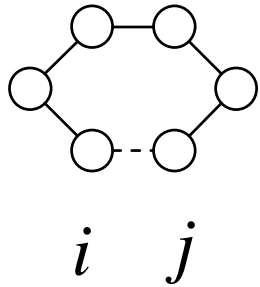
Problem: find the minimum free energy structure

Solution: also with dynamic programming called Zuker-Sankoff algorithm



Zuker-Sankoff algorithm

It finds the minimum energy structure, highly reminiscent of CYK.
 Example (part of the algorithm):

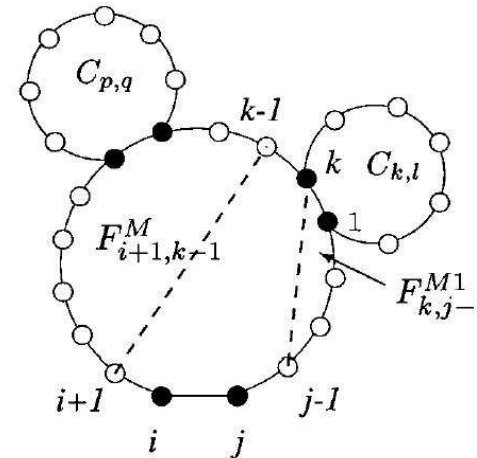


$$C_{i,j} = \min \left\{ H_{i,j}, \right.$$

$$C_{i+1,j-1} + \textit{Stacking}_{i,i+1,j-1,j},$$

$$\min_{\substack{i+1 \leq p \leq j-m-2 \\ p+m+1 \leq q \leq j-1 \\ p=i+1 \Rightarrow q \neq j-1}} \left\{ C_{p,q} + L_{i,p,q,j} \right\},$$

$$\min_{i+m+3 \leq k \leq j-m-2} \left\{ F_{i+1,k-1}^M + F_{k,j-1}^{M1} + a \right\} \left. \right\}$$



F^M and F^{M1} provide that each structure is considered only once
 (will be important later on)

Outline

- RNA sequences, secondary structure
- Dynamic programming, Nussinov algorithm
- Context Free Grammars and RNA secondary structures
- Zuker-Tinoco-Turner energy model and the Zuker-Sankoff algorithm
- Previous results on identifying RNA genes
 - tRNAscan
 - The Rfam database
 - Negative results
- Algebraic dynamic programming
- Moments of the Boltzmann distribution
- Co-transcriptional folding

Previous results:

Covariance models and tRNAscan-SE

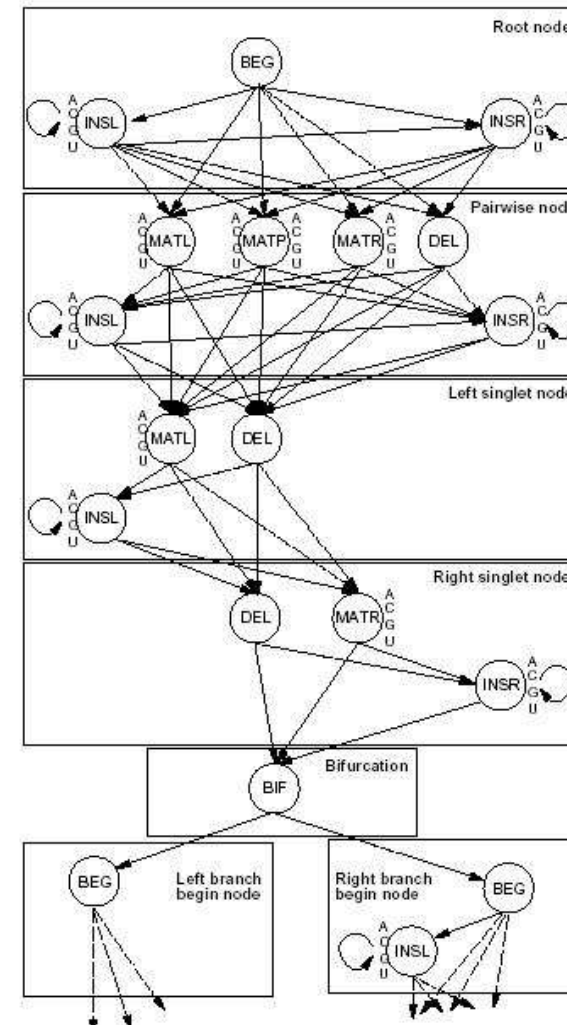
Eddy & Durbin (1994) NAR

Low & Eddy (1997) NAR

Using CM models and other techniques, they identified almost all tRNAs in several genomes available in 1996.

Results:

- Run on Silicon Indigo 200MHz
- 3000 Mbp in 36.6 hours
- 99.5% of tRNAs found
- False positives less than 0.001/Mbp



Rfam: an RNA family database

Sam Griffiths-Jones*, Alex Bateman, Mhairi Marshall, Ajay Khanna¹ and Sean R. Eddy¹

The Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SA, UK and
¹Howard Hughes Medical Institute and Department of Genetics, Washington University School of Medicine,
St Louis, MO 63110, USA

<http://www.sanger.ac.uk/Software/Rfam/>

- Sequences from more than 100 genomes
- 379 families, each family is represented with a covariance model and a seed alignment generated by that model
- Searching homologous sequences, downloading sequences, covariance models, multiple alignments, annotations, etc.

General RNA gene finding?

- Previous results seem very promising
- For protein coding genes, it works
 - No need for homologous sequences (training or teaching set)
 - A general protein coding gene does have information which separates it from random sequences (abundance of STOP codons, codon usage, UTR info, etc)
 - HMMs vs. SCFGs: HMMs are stochastic regular grammars, they have very similar dynamic programming algorithms
- Aim is to find sequences having a previously unknown structure

Working hypothesis:

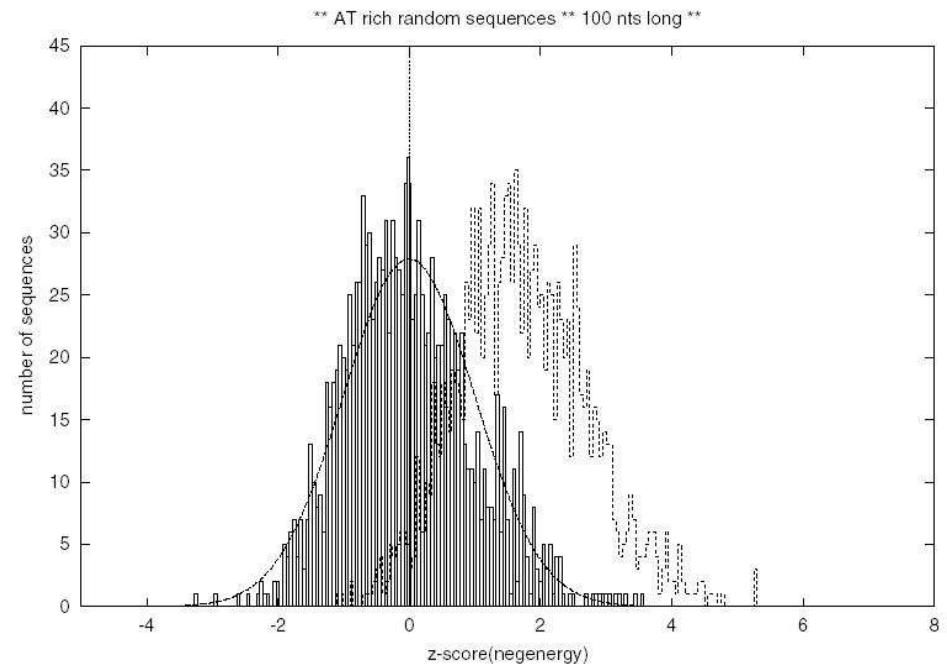
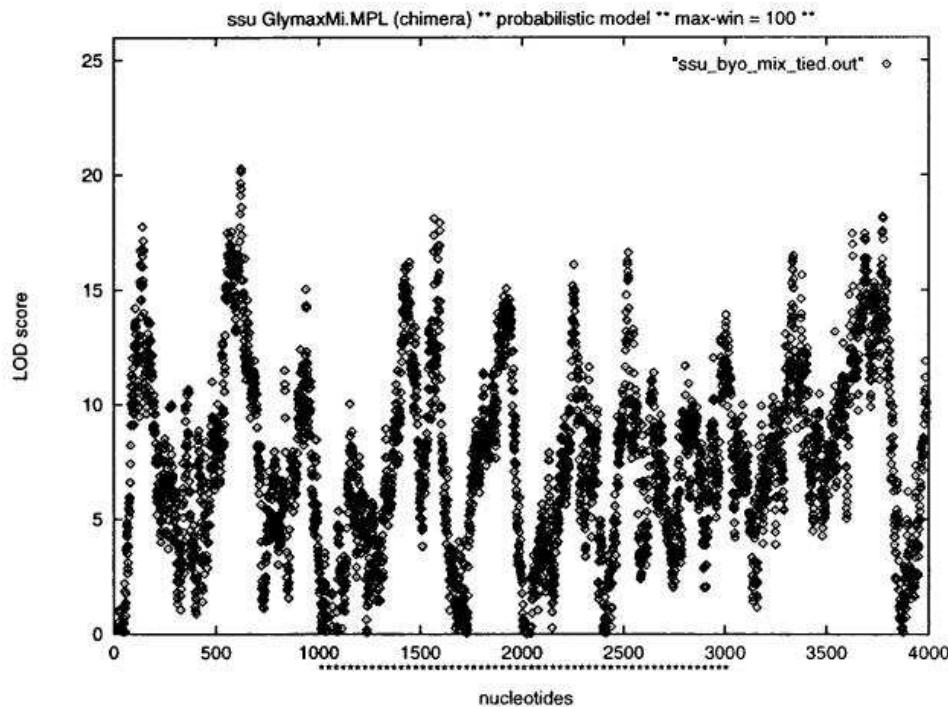
An RNA gene has 'interesting' secondary structure which is significantly different than that of random sequences.

Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs

*Elena Rivas and Sean R. Eddy**

Department of Genetics, Washington University, St. Louis, MO 63110, USA

Received on August 4, 1999; revised on December 15, 1999; accepted on December 21, 1999



Our idea: compare the best structure with suboptimal solutions

Idea: The stability of an RNA structure depends not on its free energy but the difference between its free energy and those of competing structures

The distribution of structures follows the Boltzmann distribution:

$$P(S) \propto e^{-\frac{\Delta G(S)}{RT}}$$

where $\Delta G(S)$ is the free energy of structure S , $P(S)$ is the probability of structure S , and \propto stands “proportional to”. The normalising coefficient is the partition function

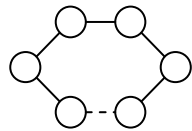
$$Z = \sum_S e^{-\frac{\Delta G(S)}{RT}}$$

Can we calculate this? YES, using algebraic dynamic programming

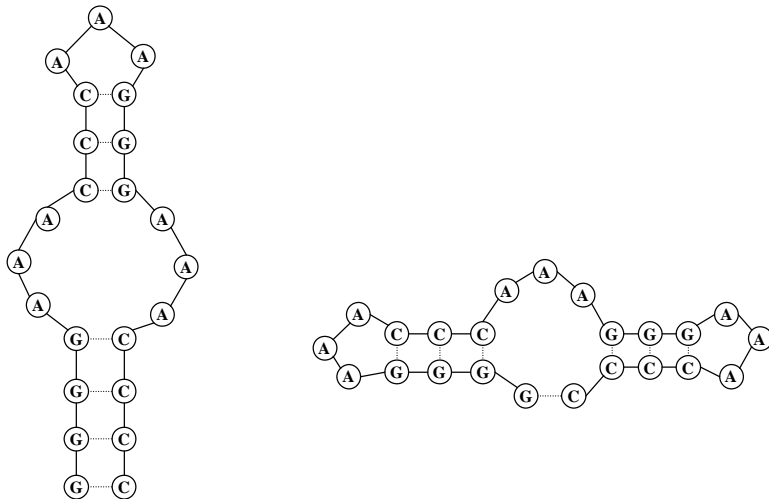
Algebraic dynamic programming

The main idea: separate concrete calculations from the recursion

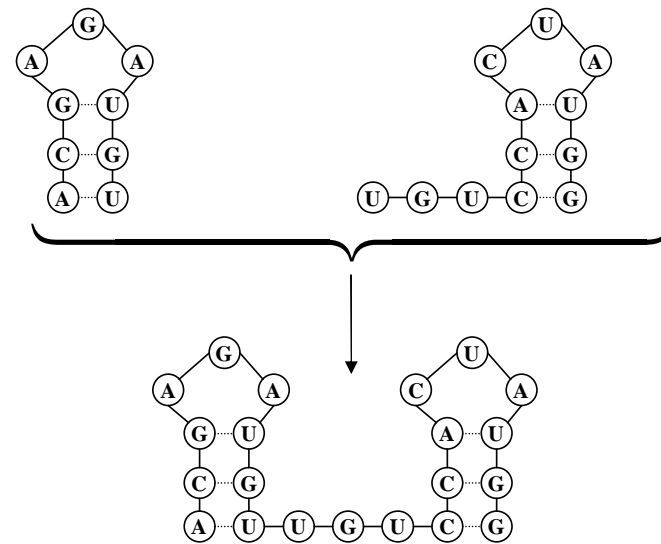
In case of RNA secondary structures, the recursion has three different functions:



Initial function



Choose function



Merge function

Best structure vs partition function

	Best structure	Partition function
Initial function	$G(S)$	$e^{-\frac{G(S)}{RT}}$
Choose function	$\min\{G(S_1), G(S_2)\}$	$Z_1 + Z_2$
Merge function	$G(S_1) + G(S_2)$	$Z_1 \times Z_2$

Proving merge function for partition function calculation

$$\sum_i \sum_j e^{-\frac{G(S_i) + G(S_j)}{RT}} = \sum_i e^{-\frac{G(S_i)}{RT}} \times \sum_j e^{-\frac{G(S_j)}{RT}} = Z_1 \times Z_2$$

Same recursion, different calculation

$$C_{i,j} = \min \left\{ H_{i,j}, \right. \\
 \left. C_{i+1,j-1} + \textit{Stacking}_{i,i+1,j-1,j}, \right. \\
 \left. \min_{\substack{i+1 \leq p \leq j-m-2 \\ p+m+1 \leq q \leq j-1 \\ p=i+1 \Rightarrow q \neq j-1}} \left\{ C_{p,q} + L_{i,p,q,j} \right\}, \right. \\
 \left. \min_{i+m+3 \leq k \leq j-m-2} \left\{ F_{i+1,k-1}^M + F_{k,j-1}^{M1} + a \right\} \right\}$$

 Classes

 Initial function

 Choose function

 Merge function

Classes have members; functions operate on members according to the calculations above

Expected value and variance

Expected value

$$E_B[G(S)] = \frac{\sum_{S_i} G(S_i) e^{-\frac{G(S_i)}{RT}}}{Z}$$

Variance

$$V_B[G(S)] = \frac{\sum_{S_i} (G(S_i) - E_B[G(S)])^2 e^{-\frac{G(S_i)}{RT}}}{Z} = E_B[G^2(S)] - E_B^2[G(S)]$$

Hence we would like to calculate

$$X = \sum_{S_i} G(S_i) e^{-\frac{G(S_i)}{RT}} \quad \text{and} \quad Y = \sum_{S_i} G^2(S_i) e^{-\frac{G(S_i)}{RT}}$$

Getting X and Y

	X	Y
Initial function	$G(S)e^{-\frac{G(S)}{RT}}$	$G^2(S)e^{-\frac{G(S)}{RT}}$
Choose function	$X_1 + X_2$	$Y_1 + Y_2$
Merge function	$X_1 \times Z_2 + X_2 \times Z_1$	$Y_1 \times Z_2 + 2 \times X_1 \times X_2 + Y_2 \times Z_1$

Proofs:

$$\sum_i \sum_j (G(S_i) + G(S_j)) e^{-\frac{G(S_i) + G(S_j)}{RT}} = \sum_i G(S_i) e^{-\frac{G(S_i)}{RT}} \times \sum_j e^{-\frac{G(S_j)}{RT}} + \sum_j G(S_j) e^{-\frac{G(S_j)}{RT}} \times \sum_i e^{-\frac{G(S_i)}{RT}} = X_1 \times Z_2 + X_2 \times Z_1$$

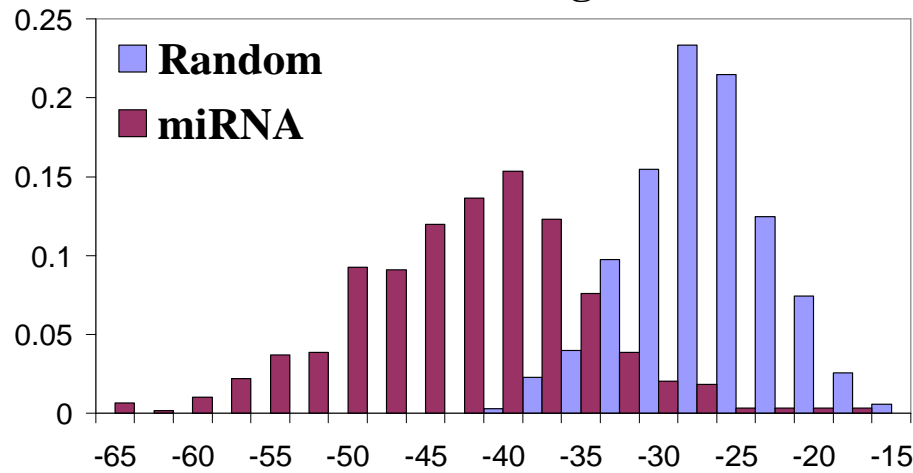
Similarly for $Y = \sum_i \sum_j (G(S_i) + G(S_j))^2 e^{-\frac{G(S_i) + G(S_j)}{RT}}$

Interesting statistics

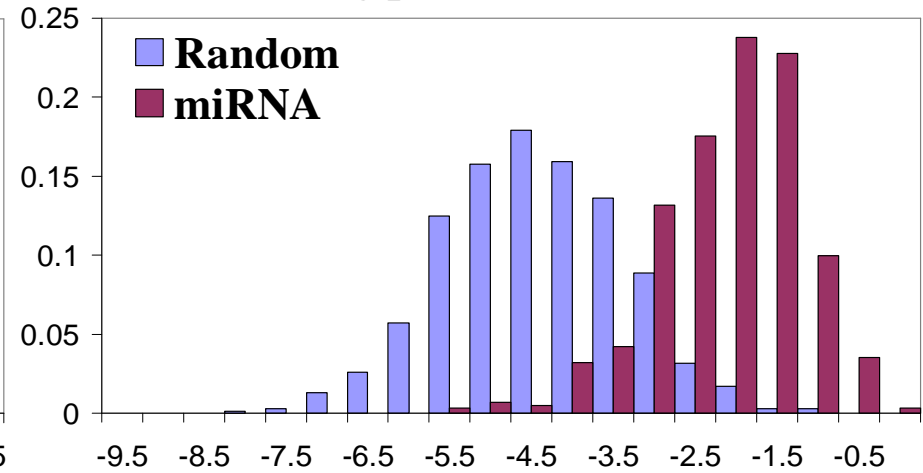
- Energy of the best structure
- Probability of the best structure in the Boltzmann ensemble
- Deviation of the best energy from the average energy
 - Exclude the best energy from the expected energy
 - Exclude structures similar to the best structure with counting only structures with maximal helices
- Variance of the Boltzmann distribution, with the same conditions than above

Results: miRNA

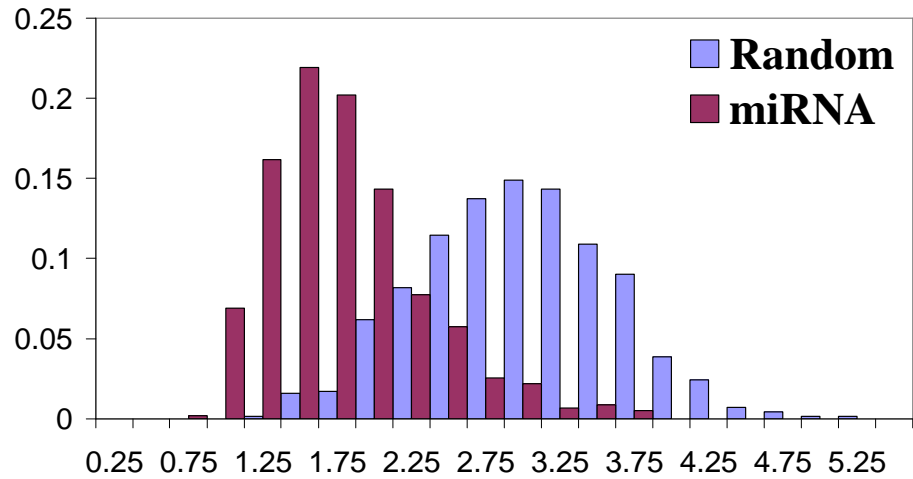
minimum energies



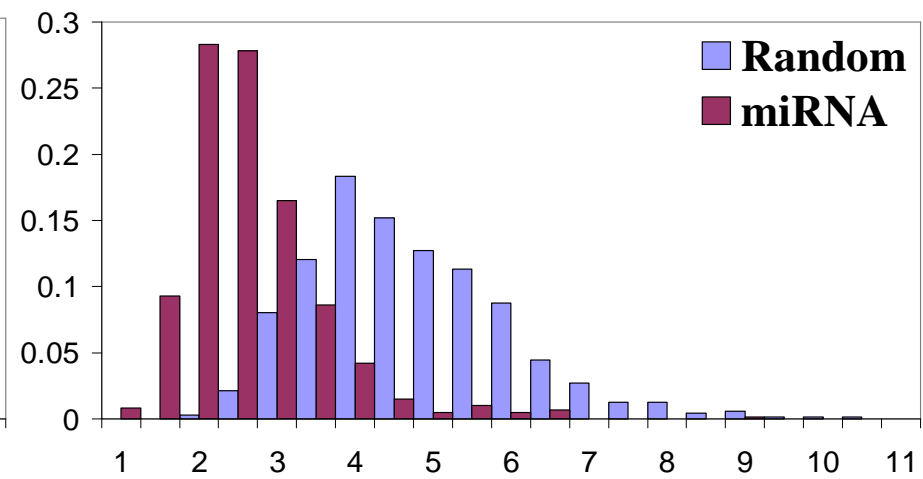
log probabilities



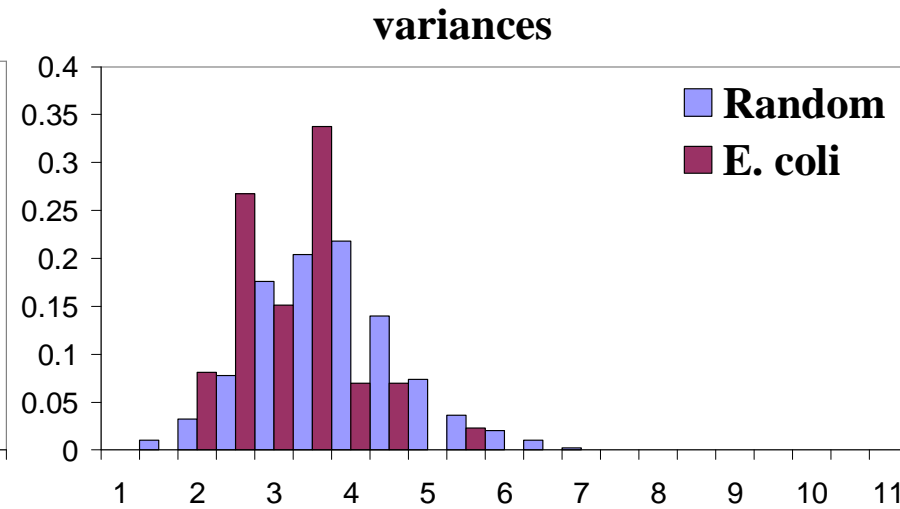
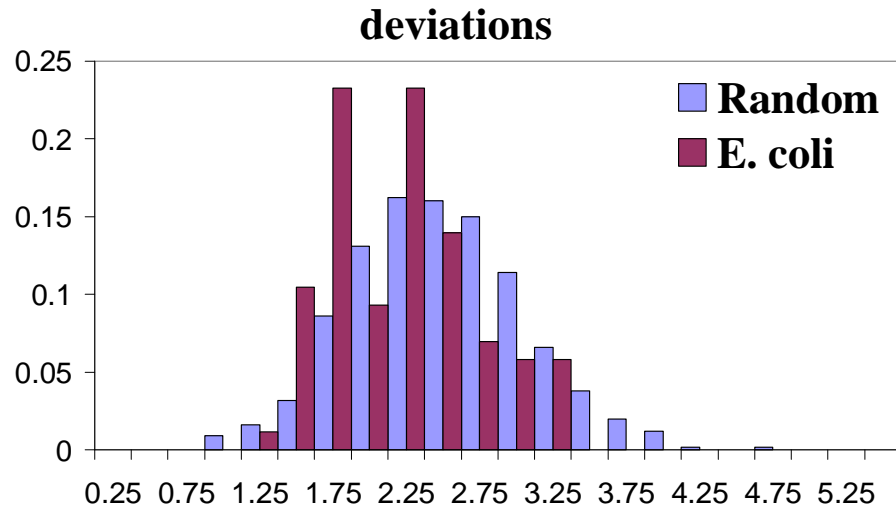
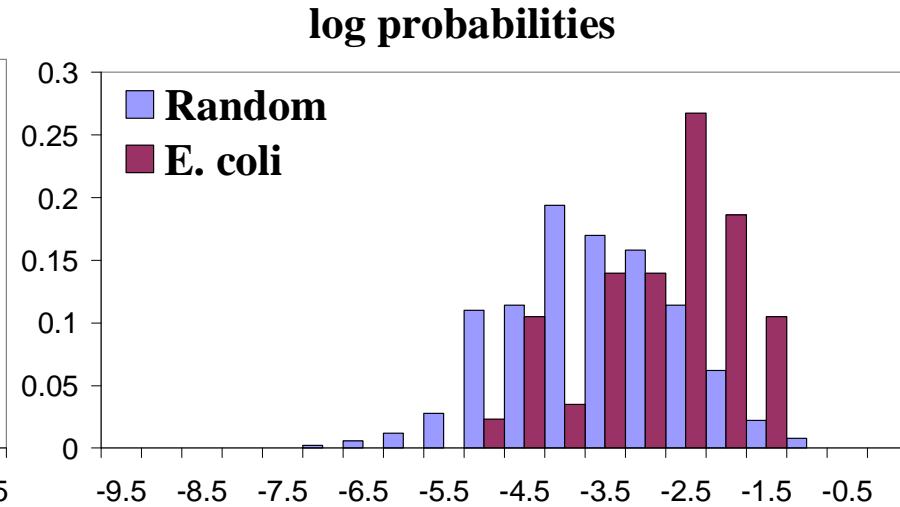
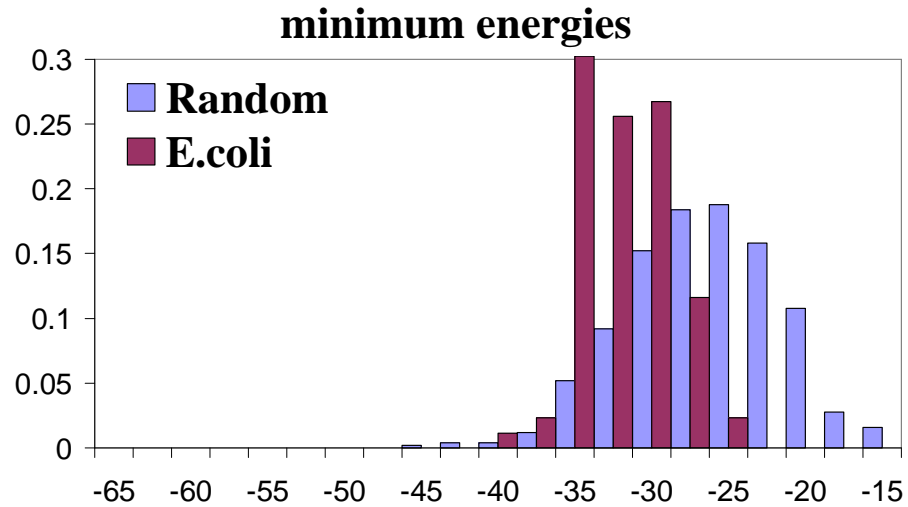
deviations



variances

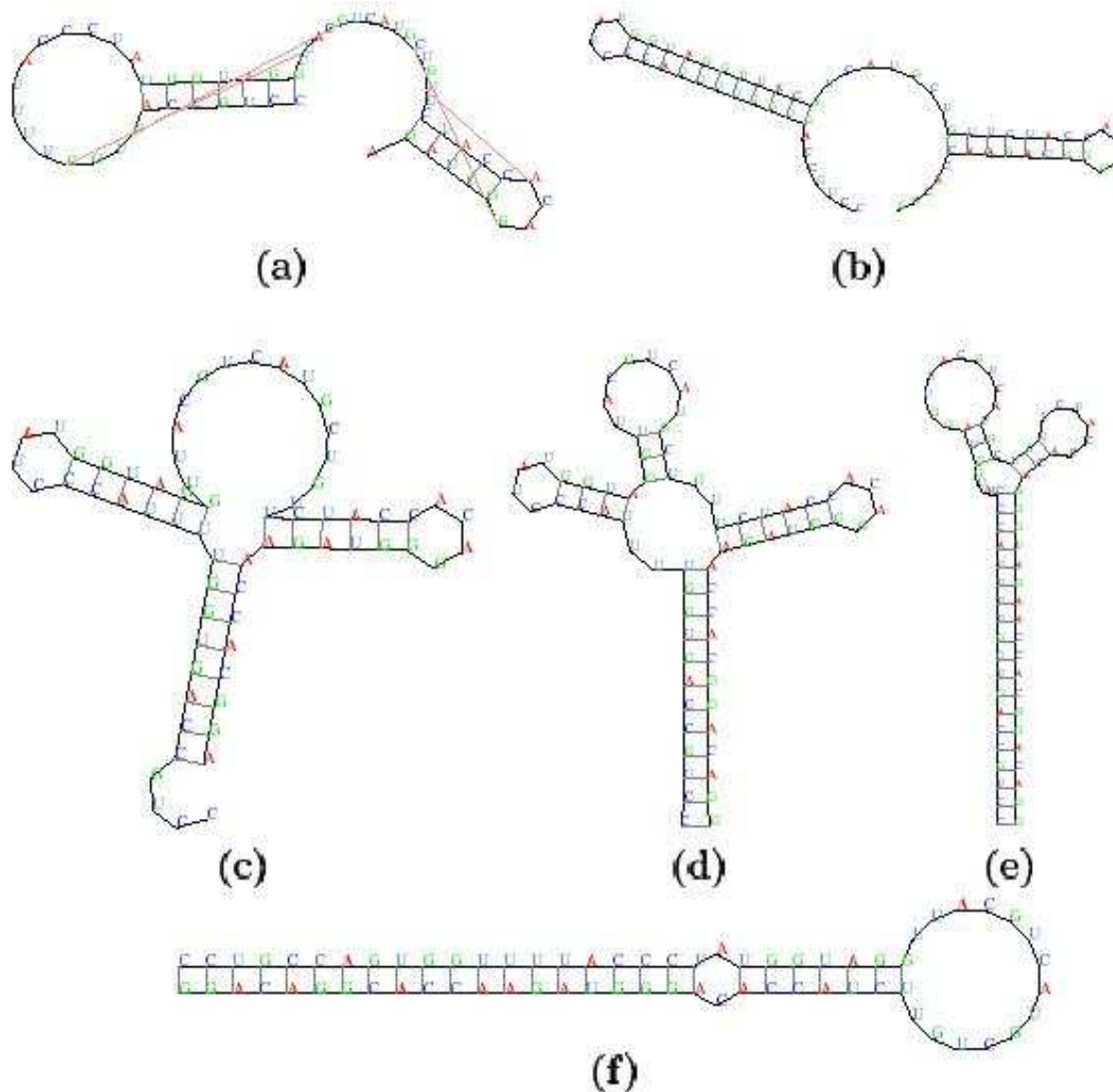


Results: E.coli tRNAs



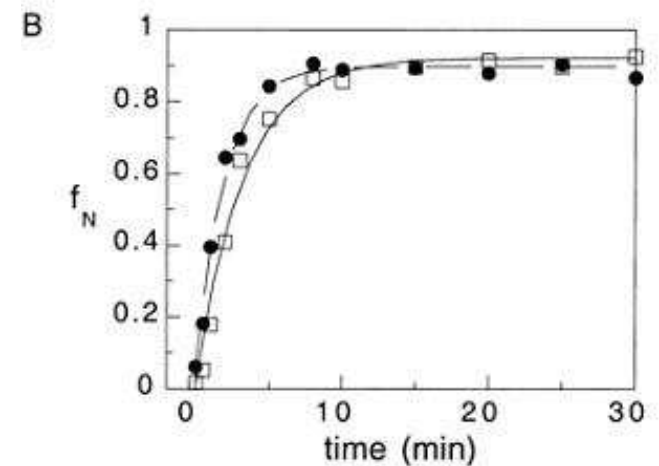
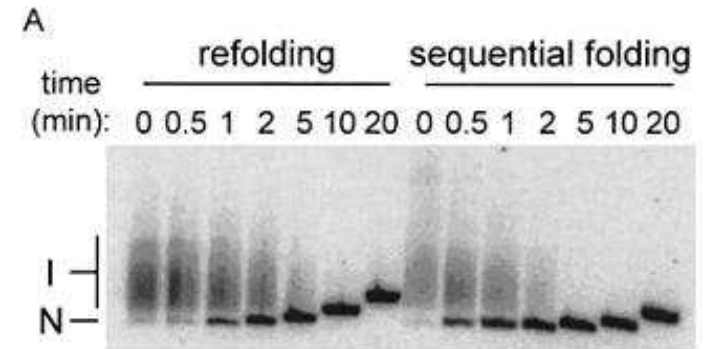
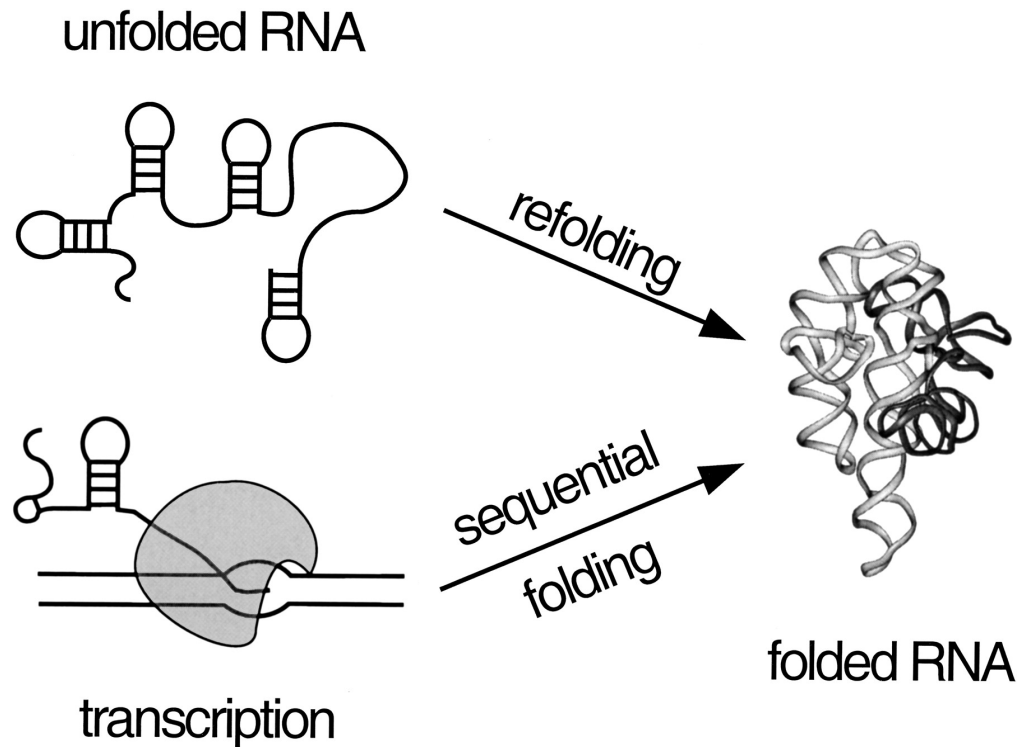
Cotranscriptional folding of miRNA?

As predicted by the Kinefold server



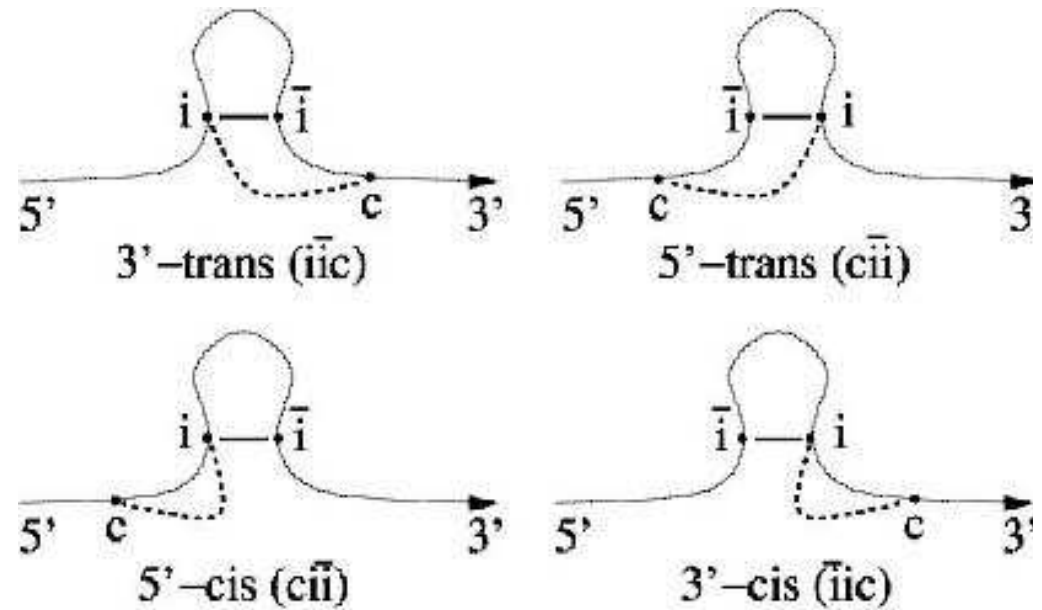
Co-transcriptional folding

Heilman-Miller & Woodson (2003) *RNA* 9:722-733.



Meyer & Miklós (2004) *BMC Mol. Biol.* 5:10.

Taxonomic unit	all 16S rRNA	23S rRNA	Group I	Group II
Data set A				
Archea	28	22	6	0
Bacteria	277	232	45	0
Eukaryotes	41	35	6	0
Chloroplasts	6	6	0	0
Mitochondria	9	9	0	0
Sum	361	304	57	0
Data set B				
Eukaryotes	15	0	0	15
Bacteria	5	0	5	0
Chloroplasts	5	0	5	0
Mitochondria	23	0	17	6
Sum	48	0	27	6



dataset	A		B	
	p-value for t-test	p-value for pos	p-value for t-test	p-value for pos
\overline{Cis}_p	< 0.0001	< 0.0001	0.5733	0.6137
\overline{Cis}_g	< 0.0001	< 0.0001	0.5650	0.6137
\overline{Trans}_p	0.0012	< 0.0001	0.3093	0.8068
\overline{Trans}_g	0.0021	< 0.0001	0.3011	0.5000

- Significant deviation from the H_0 hypothesis only for sequences which fold co-transcriptional, there is no deviation for sequences which are edited after transcription
- “cis” values are smaller than “trans” values, hence stability after folding is also important
- For sequences folding co-transcriptional,

$$5'\text{-trans} < 3'\text{-trans}$$

hence competing structures suppressed

- For sequences folding co-transcriptional,

$$5'\text{-cis} > 3'\text{-cis}$$

hence several temporary structures form during transcription

Conclusions + possible further directions of research

RNA folding is more complicated than we thought a few years ago

Open questions:

- Is the folding path conservative?
- Can we use folding information for
 - Structure prediction?
 - Evolutionary inferring?
- Can we improve folding simulations? Note that the error is multiplicative here!