

Novel computational methods for detecting functional structures in RNA molecules

Irmtraud Meyer

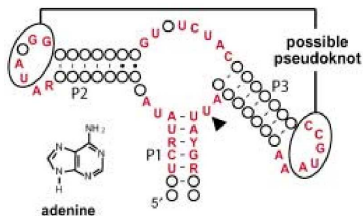
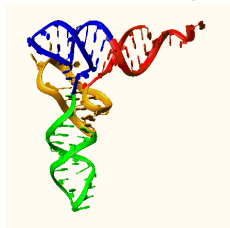
UBC Bioinformatics Centre &
Department of Computer Science

irmtraud@cs.ubc.ca

January 11, 2007

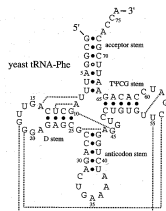
RNA structures fulfill many functional roles in the cell

- tRNAs map codons of mRNA to amino-acids
- rRNAs determine ribosome's structure and function
- RNA structural elements in protein-coding genes can
 - function as translation initiation sites
 - influence translation efficiency
 - regulate mRNA degradation
 - function as pre-mRNA editing sites
 - function as zip-codes for mRNA localization
 - bind small metabolites and repress or activate translation and transcription (riboswitches)



RNA structure: some definitions

- **RNA structure** (for us): set of base-paired sequence positions
- **consensus base-pairs**: $\{C, G\}$, $\{A, U\}$ and $\{G, U\}$
- **functional RNA structure**: structure that conveys observed functionality to the molecule in the cell



tRNA tertiary structure

secondary structure

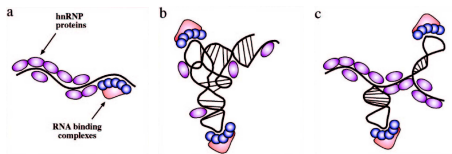
```
GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAAAUCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA
(((((((.....))))).((((.....)))).....((((.....)))))).....
```

dot-bracket notation

Difficulties in predicting functional RNA structure

1. an RNA sequence has many potential structures
Which is the functional structure?
2. entire RNA sequence need not fold into the most stable structure because
 - RNA may be bound by other molecules
 - RNA may not have enough time to fold
 - transient structures form during co-transcriptional folding

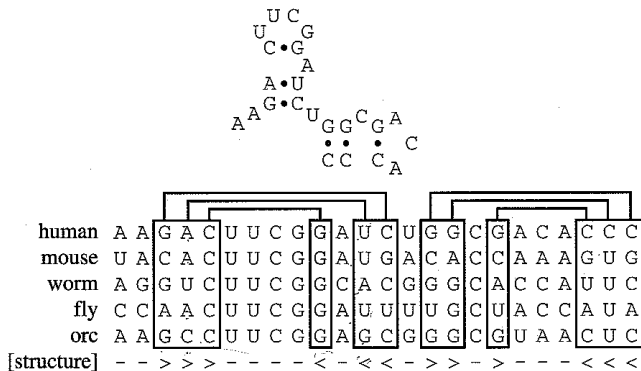
How do we know which regions of a given RNA sequence are structured?



[Buratti and Baralle, Mol and Cell Biol (2004) 24(24) 10505]

Prediction of functional RNA structure: main ideas

1. align **functionally equivalent** sequences from related organisms
2. find pairs of **co-varying columns** in the alignment
3. combine co-varying columns into an RNA secondary structure



[Durbin et al., Biological sequence analysis, CUP (1998)]

RNA-DECODER

*How can we detect structural regions in long RNA sequences?
Can functional RNA structures overlap protein-coding regions?*

Prediction program RNA-DECODER

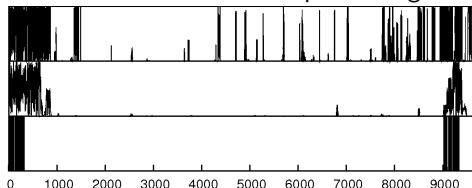
Input:

- fixed multiple sequence alignment with known coding regions
- fixed evolutionary tree relating the sequences in the alignment

Two types of output:

- 1: base-pairing probability for each column in the alignment
- 2: secondary structure with highest overall probability

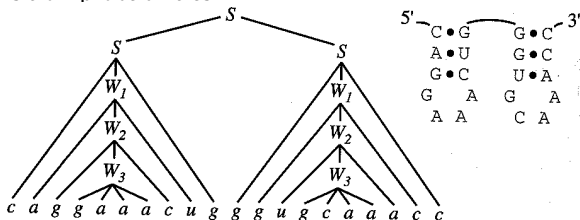
RNA-DECODER scan of Hepatitis C genome



[Nucleic Acids Research (2004) 32(16) 4925]

Computational method used by RNA-DECODER

- decompose any RNA structure into
 - base-paired and unpaired nucleotide (**emission probabilities**)
 - elementary structural elements (**transition probabilities**)
- define **probability of every RNA structure** := product of emission and transition probabilities



- for a given RNA sequence of length L , use CYK algorithm to derive the RNA structure with the highest overall probability in $\mathcal{O}(L^3)$ time and $\mathcal{O}(L^2)$ memory

⇒ overall concept of stochastic context free grammars (**SCFGs**)

Summary RNA-DECODER

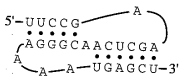
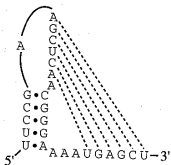
Advantages:

- + explicitly models unstructured regions
- + can detect evolutionarily conserved structure, also in coding regions
- + assigns confidence values to its predictions (probabilistic model)

Disadvantages:

- SCFGs cannot model **pseudo-knots**, i.e. non-nested pairs [()]
- need evolutionarily diverse data to get good sequence signals
- diverse data make it difficult to generate a good input alignment without already knowing the conserved structure

⇒ **chicken-egg problem**



Work in progress: SIMULFOLD

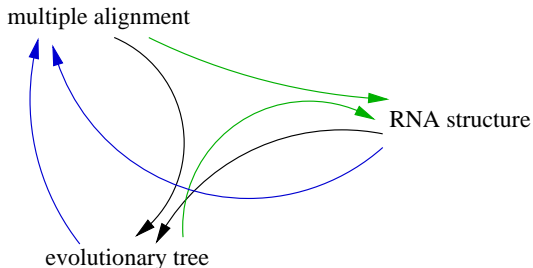
Can we simultaneously infer alignment, tree and structure?

Wish list for a novel method

Given a set D of several un-aligned, evolutionarily related sequences, we want to simultaneously infer:

- an evolutionary tree T relating the sequences
- a conserved RNA structure S which may include pseudo-knots
- a multiple sequence alignment A

Why simultaneous?



Review of existing work

Co-estimation:

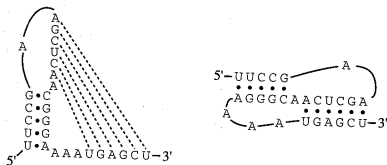
- Sankoff (1985): simultaneous estimation of alignment and structure
 - exact, but too slow, no pseudo-knots
- “alignment-free” graph-based methods [e.g. CARNAC by Touzet et al. (2004)], COMRNA by Ji et al. (2004)]
 - report only single predicted structure, no polynomial algorithm
- n-SCFGs take $\mathcal{O}(L^{3n})$ time and $\mathcal{O}(L^{2n})$ memory to simultaneously align n RNA sequences and to predict their structure [e.g. pair-SCFGs CONSAN by Dowell and Eddy (2006), STEMLOC by Holmes (2005)]
 - too slow, no pseudo-knots

Simulations of kinetic folding process:

- have some success [KINEFOLD by Isambert and Siggia (2000), Gulyaev et al. (1995)], but error is multiplicative and method considers only single sequence

Previous attempts: pseudo-knot prediction

- several polynomial-time algorithms for limited classes of structures [e.g. Rivas and Eddy (2000), Dirks and Pierce (2003), Reeder and Giegerich (2004)]
- general problem is NP-complete
- approaches which use maximum weighted matching algorithm (**MWM**) [e.g. Tabaska et al. (1998), HXMATCH by Witwer et al. (2004)] and heuristics have success, but require a good input alignment and many sequences



Wish list for novel method SIMULFOLD

- make best use of comparative information from related organisms
 - ⇒ use more than 2 related sequences as input
- include pseudo-knots in structure prediction
 - ⇒ cannot use SCFGs
- be able to test different folding hypotheses, e.g. 5' → 3' asymmetries due to co-transcriptional folding
 - ⇒ cannot use SCFGs
- quantify uncertainty of predictions
 - ⇒ do not only report one maximum likelihood solution

⇒ need conceptually new theoretical framework

Theoretical framework: basic ideas

Aim:

- find triples (A, T, S) that are best supported by the data D
- sample from the joint posterior probability $P(A, T, S|D)$

Strategy:

- using $P(X|Y) \cdot P(Y) = P(X, Y) = P(Y|X) \cdot P(X)$, write the posterior probability as

$$P(A, T, S|D) = \frac{1}{Z} P(D|A, T, S) \cdot P(A, T, S)$$

where

- $Z = P(D)$ is the **partition function** (unknown and not needed)
- $P(A, T, S|D)$ the **posterior probability**
- $P(D|A, T, S)$ the **likelihood**
- $P(A, T, S)$ the **prior**

⇒ use Bayesian Markov chain Monte Carlo (**MCMC**)

Theoretical framework: basic ideas

Strategy:

$$P(A, T, S|D) = \frac{1}{Z} P(D|A, T, S) \cdot P(A, T, S)$$

- use expressions on the right hand side to calculate values for specific (A, T, S) triples
- sample (A, T, S) values with a MCMC whose equilibrium density is the posterior probability

Basic MCMC facts: Theorem of Metropolis (1953)

For any ergodic and reversible Markov chain $Q(X'|X)$, called **proposal density**, we can define a Markov chain $X_i \mapsto X_{i+1}$ as follows

1. choose X' according to $Q(X'|X_i)$
2. with probability

$$\max\left\{1, \frac{P(X)Q(X|X')}{P(X')Q(X'|X)}\right\}$$

set $X_{i+1} = X'$ (**accept proposal**), otherwise

set $X_{i+1} = X$ (**reject proposal**)

The resulting Markov chain $X_i \mapsto X_{i+1}$ is then ergodic and converges to $P(X)$, called **equilibrium density**.

[In our case: $X = (A, T, S)$ and P is the posterior probability.]

Calculating the likelihood $P(D|A, T, S)$

$$P(A, T, S|D) = \frac{1}{Z} P(D|A, T, S) \cdot P(A, T, S)$$

What is the probability of observing the data (individual sequences), given an alignment, evolutionary tree and RNA structure?



Use Felsenstein's algorithm to calculate the likelihood:

- for un-paired columns, see ● above, single nucleotides evolve on tree using a Markov chain described by a 4x4 rate matrix
- for base-paired columns, see ● above, **pairs of nucleotides** evolve on tree using a Markov chain described by a 16x16 rate matrix
- treat gaps as missing information

Calculating the prior $P(A, T, S)$

$$P(A, T, S|D) = \frac{1}{Z} P(D|A, T, S) \cdot P(A, T, S)$$

The prior has the purpose to describe a goodness-distribution over all (A, T, S) triples. We decompose the prior into

$$F_1(S, A) \cdot F_2(A, T) \cdot F_3(T)$$

because

$F_1(S, A)$: the common structure depends on the alignment, e.g. its length

$F_2(A, T)$: the observed alignment is the result of an evolutionary process described by the evolutionary tree

$F_3(T)$: we need a tree prior as the likelihood, $P(D|A, T, S)$, does not tend to zero for infinite branch lengths

The structure prior $F_1(S, A)$

$$P(A, T, S) = F_1(S, A) \cdot F_2(A, T) \cdot F_3(T)$$

- probabilities of **base-pairs** already in likelihood, $P(D|A, T, S)$
- incorporate free energy of structure's **topology**, τ , which is usually purely entropic into structure prior \Rightarrow probability is temperature independent

$$P(\tau) \propto e^{-\frac{G(\tau)}{RT}} = e^{-\frac{S(\tau)T}{RT}} = e^{-\frac{S(\tau)}{R}}$$

- entropic terms in the structure prior:
 - pseudo-knot free structures can be decomposed into loops [Zuker and Sankoff (1984)], entropy for loop of length L is

$$\begin{aligned} S(L) &\propto 1.75 \cdot RT \cdot \ln(L) \\ \Rightarrow P(L) &\propto L^{-1.75} \end{aligned}$$

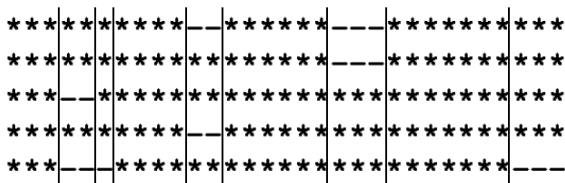
- pseudo-knots are more complicated, use simple model where each stretch of un-paired nucleotides of length L in a pseudo-knots gets

$$P(L) \propto L^{-1.75}$$

Alignment prior $F_2(A, T)$: treatment of gaps

$$P(A, T, S) = F_1(S, A) \cdot F_2(A, T) \cdot F_3(T)$$

Idea: decompose alignment into homogeneous groups



assign gap-opening and gap-extension penalties

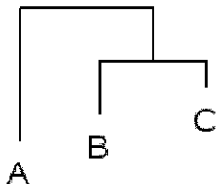
- the probability of an alignment is proportional to the exponentiated penalty score, the basis of exponentiation depends on the edge lengths in the tree

Tree prior $F_3(T)$

$$P(A, T, S) = F_1(S, A) \cdot F_2(A, T) \cdot F_3(T)$$

- need tree prior as likelihood, $P(D|A, T, S)$, does not tend to zero for infinite branch lengths
- choose uninformative tree prior

$$P(T) \propto e^{-\sum_i \text{edge length}_i}$$



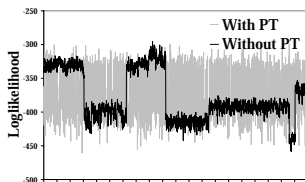
Efficiency of MCMC

There are three cases when MCMC is not efficient:

- acceptance ratio is small
- chain gets stuck in local optimum
- computational time to perform a step is large

To avoid these problems we use partial Metropolis importance sampling

- quickly propose movements with replace only part of the data
- keep rejection probability and autocorrelation low
- use sophisticated technique of parallel tempering, if required



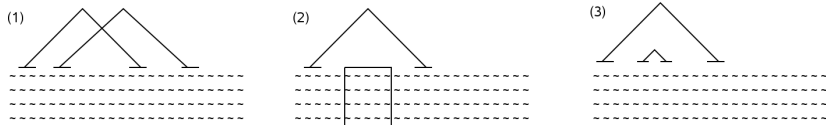
Log-likelihood as function of the Markov chain states
with and without parallel tempering (PT)

Sampling strategy

We use two types of sampling:

1. sample tree, keep structure and alignment fixed
2. sample alignment and structure, keep tree fixed

Strategy for sampling alignment and structure



Sampling alignments

- cut our part of the alignment in random window, remove all gaps and realign it
- realignment method: stochastic version of iterative alignment
- sample an alignment from the posterior of a pair-HMM using forward-backward sampling

Sampling structures

Need one-to-one correspondence between proposal and back-proposals

- Proposals
 - remove a set of random number of helices, A
 - propose a set of random number of helices, B
- Back-proposals
 - remove the set of helices, B
 - add the set of helices, A

We have to make sure we have a **unique way** of proposing helices, otherwise we cannot easily calculate proposal and back-proposal probabilities.

Efficient structure sampling

Before starting MCMC:

- for each sequence, create table D and vector E using standard dynamic programming
 - $D(i, j)$ = score of best helix with outer base-pair (i, j)
 - $E(i) = \sum_j D(i, j)$ score for helices starting at i on the 5' site

In MCMC:

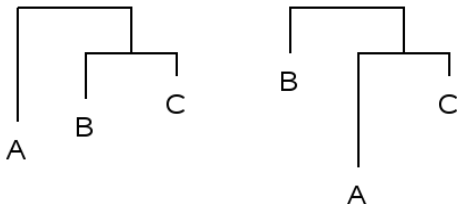
- scan alignment from left to right and make random decision where to start helix
- choose 5' start of new helix proportional to the sum of E values in that column (i.e. do importance sampling)
- for a chosen 5' start, choose 3' end of helix proportional to sum of corresponding D values
- choose length of helix based on goodness of that helix

⇒ can sample reasonable helix in linear rather than cubic time !

Sampling trees

Tree sampling is subdivided into two steps:

- sampling an edge length
- sampling a tree topology: swap niece and aunt, proven to be ergodic



Post-processing the sampled data

The MCMC returns a large amount of triples (A, T, S) which are distributed according to the posterior probability.

What can we do with it?

- trees: use standard approaches to derive a consensus tree and tree networks
- alignment: derive maximum posterior decoding alignment by dynamic programming on the acyclic, directed graph that represents all sampled alignments
- structure: use maximum weighted matching ([MWM](#)) algorithm to extract bi-secondary structure from base-pairing probabilities for each individual sequence in $\mathcal{O}(L^3)$ time and $\mathcal{O}(L^2)$ memory for sequence of length L

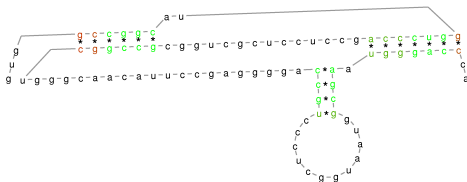
First results

	SIMULFOLD	CARNAC	HXMATCH
	co-estimates T , S and A	no pseudo-knots, no T , no A	requires good and fixed A , no T
	Sensitivity		
coronavirus (9 seqs)	0.78	0.50	0.94
enterovirus (12 seqs)	0.84	0.32	0.79
Hep. delta virus (12 seqs)	0.85* (0.63)	0.48	0.70
	Specificity		
coronavirus	0.67	1.00	0.81
enterovirus	0.86	1.00	0.64
Hep. delta virus	0.59* (0.53)	0.87	0.61

- T tree, A alignment, S RNA structure
- **sensitivity** $\in [0, 1]$ fraction of known base-pairs which are correctly predicted
- **specificity** $\in [0, 1]$ fraction of predicted base-pairs which are correct
- Simulfold: for each set, CPU time was ca. 30 minutes on a 3 GHz machine using 3 MB memory
- * using parallel tempering to avoid getting stuck in local minima
- used 10000 moves as burn-in, 2000 after burn-in and 100 between two samples
- acceptance ratios around 50%

Summary and outlook SIMULFOLD

- + first program to co-estimate alignment, tree and RNA structure including pseudo-knots !
 - + calculates reliability values for its predictions (see figure below)
 - + very respectable performance
- next compare SIMULFOLD to HXMATCH with automatically generated alignments
- next evaluate tree and alignment prediction quality
- next explore different structure formation hypotheses



HDV structure predicted by SIMULFOLD

