Combinatorial Optimization in Bioinformatics

RNA Structure Prediction—Part II

Sebastian Will · Yann Ponty

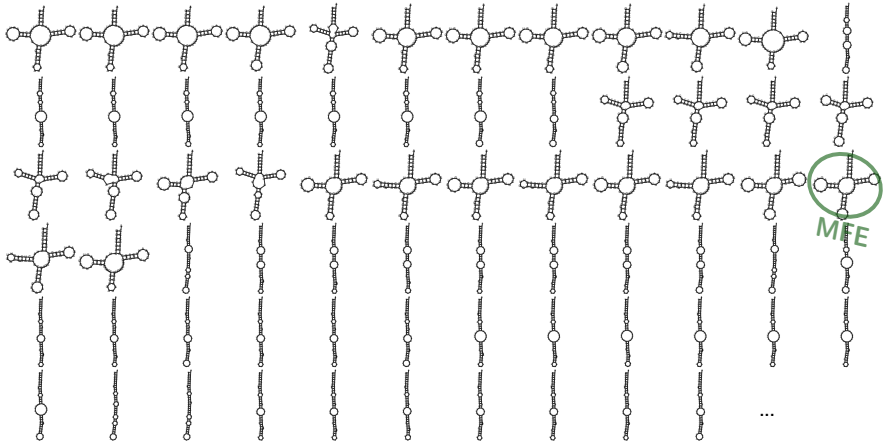
sebastian.will@polytechnique.edu · yann.ponty@lix.polytechnique.fr

Beyond predicting single RNA structures

- so far: optimization over all (non-crossing) RNA structures
  - from simple to more complex objective functions
  - (further) bioinformatics example of optimization by DP: "optimal substructure", recursive decompsosition
  - first look beyond: relevance of ambiguity, counting
- today: beyond finding the optimal solution—more power of DP
  - **statistical mechanics**: energies correspond to probabilities
  - here: directly **probabilities of structures** in thermodynamic equilibrium, but technique is generally useful to assign probabilities to combinatoric objects.
  - calculate partition functions by DP—**sum over all structures** (in place of optimize)
  - inside + outside algorithm—**probability of features**
  - structure prediction from **multiple sequences**, **comparative analysis**

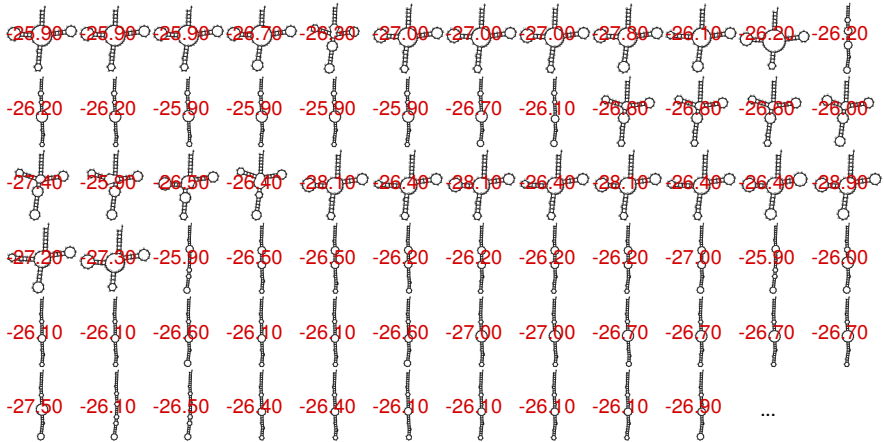
RNA structure ensembles, Probabilities

GGGCUAUUAGCUCAGUUGGUUAGAGCGCACCCCUGAUAAGGGUGAGGUCGCUGAUUCGAAUUCAGCAUAGCCCA



RNA structure ensembles, Probabilities

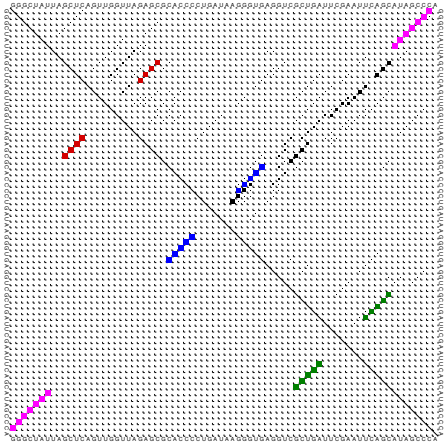
GGGCUAUUAGCUCAGUUGGUUAGAGCGCACCCCUGAUAAGGGUGAGGUCGCUGAUUCGAAUUCAGCAUAGCCCA

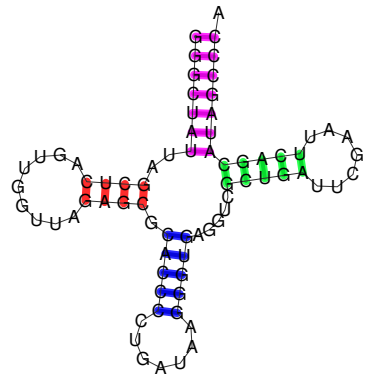


Energies → Structure Probabilities

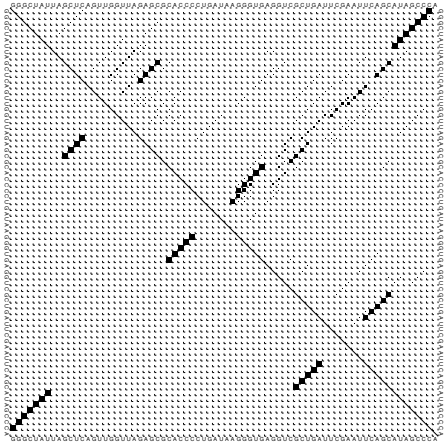
Structure Probabilities → Base Pair Probabilities

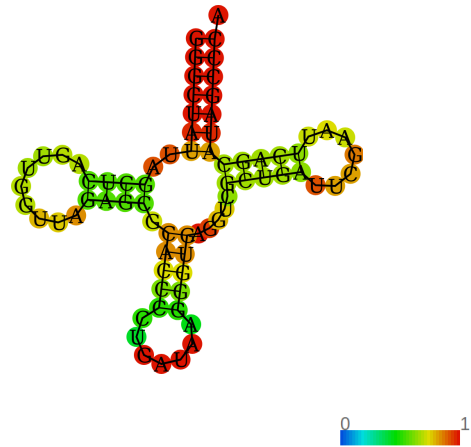
Dotplots and Reliabilities





Dotplots and Reliabilities





Thermodynamic Equilibrium / Statistical Mechanics

We assume that the probability of each possible structure P (in thermodynamic equilibrium) is proportional to its Boltzmann weight ($\implies$ **Boltzmann distribution**).

$$Pr(P) \propto exp(-\frac{E(P)}{RT})$$

(R = gas constant; T = temperature in Kelvin).

Thermodynamic Equilibrium / Statistical Mechanics

We assume that the probability of each possible structure P (in thermodynamic equilibrium) is proportional to its Boltzmann weight ($\implies$ **Boltzmann distribution**).

$$Pr(P) \propto exp(-\frac{E(P)}{RT})$$

(R = gas constant; T = temperature in Kelvin).

Reason to assume **Boltzmann distribution**:

- Formally justified as distribution with maximal entropy in a closed physical system (with constant average energy).
- Boltzmann distributions models the distribution of free, dynamically folding structures after infinite time (and without additional forces).

Normalization: Partition function

$$Pr(P) \propto exp(-\frac{E(P)}{RT})$$

Normalize Boltzmann weights by Z:

$$Pr(P) = exp(-\frac{E(P)}{RT})/Z, \text{ such that}$$

$$\sum_{P} Pr(P) = 1$$

Normalization: Partition function

$$Pr(P) \propto exp(-\frac{E(P)}{RT})$$

Normalize Boltzmann weights by Z:

$$Pr(P) = exp(-\frac{E(P)}{RT})/Z, \text{ such that}$$

$$\sum_P Pr(P) = 1$$

$$\implies Z = \sum_P exp(-\frac{E(P)}{RT})$$

Z is called partition function.

Normalization: Partition function

$$Pr(P) \propto exp(-\frac{E(P)}{RT})$$

Normalize Boltzmann weights by Z:

$$Pr(P) = exp(-\frac{E(P)}{RT})/Z, \text{ such that}$$

$$\sum_P Pr(P) = 1$$

$$\implies Z = \sum_P exp(-\frac{E(P)}{RT})$$

Z is called partition function.

Efficient computation?

- weighted sum has similarities to counting (DP!)
- weights depend on free energy (Zuker-like DP?!)

Recall: Counting of Structures

$$N_{i,j} = \max \begin{cases} N_{i+1,j} \\ \max\limits_{\substack{i<k\leq j \\ S[i],S[j] compl.}} 1 + N_{i+1,k-1} + N_{k+1,j} \end{cases}$$

Recall: Counting of Structures

$$N_{i,j} = \max \begin{cases} N_{i+1,j} \\ \max\limits_{\substack{i<k\leq j \\ S[i],S[j]\,compl.}} 1 + N_{i+1,k-1} + N_{k+1,j} \end{cases}$$

$$\Downarrow$$

$$C_{i,j} = C_{i+1,j} + \sum_{\substack{i<k\leq j \\ S[i],S[j]\,compl.}} 1 \cdot C_{i+1,k-1} \cdot C_{k+1,j}$$

Exchange of operators: $\max \to \sum/+$, $\sum/+ \to \prod/\cdot$

From counting to partition function

$$C_{i,j} = C_{i+1,j} + \sum_{\substack{i<k\leq j \\ S[i],S[j] compl.}} 1 \cdot C_{i+1,k-1} \cdot C_{k+1,j}$$

$$\Longrightarrow Z_{i,j} = Z_{i+1,j} + \sum_{\substack{i<k\leq j \\ S[i],S[j] compl.}} \exp\left(\frac{-E_{bp}(i,k)}{RT}\right) \cdot Z_{i+1,k-1} \cdot Z_{k+1,j}$$

From counting to partition function

$$C_{i,j} = C_{i+1,j} + \sum_{\substack{i<k\leq j \\ S[i],S[j] compl.}} 1 \cdot C_{i+1,k-1} \cdot C_{k+1,j}$$

$$\Longrightarrow Z_{i,j} = Z_{i+1,j} + \sum_{\substack{i<k\leq j \\ S[i],S[j] compl.}} \exp(\frac{-E_{bp}(i,k)}{RT}) \cdot Z_{i+1,k-1} \cdot Z_{k+1,j}$$

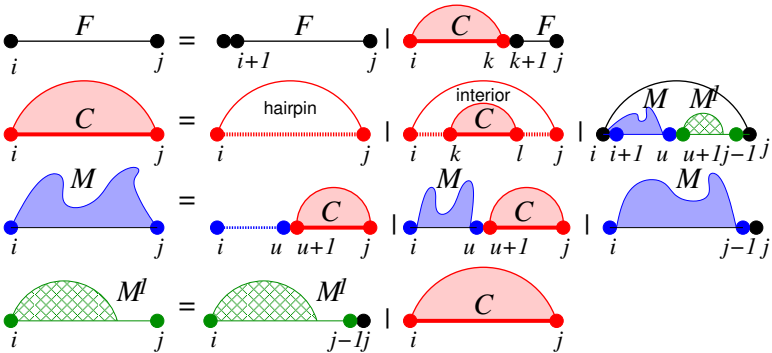
Validity?

$$\exp(\frac{-E_{bp}(i,k)}{RT}) \cdot Z_{i+1,k-1} \cdot Z_{k+1,j} = \exp(\frac{-E_{bp}(i,k)}{RT}) \cdot \sum_x \exp(\frac{-E(x)}{RT}) \cdot \sum_y \exp(\frac{-E(y)}{RT})$$

$$= \sum_{x,y} \exp(\frac{-E_{bp}(i,k)}{RT}) \cdot \exp(\frac{-E(x)}{RT}) \cdot \exp(\frac{-E(y)}{RT})$$

$$= \sum_{x,y} \exp(\frac{-(E_{bp}(i,k) + E(x) + E(y)}{RT}$$

Free energy minimization ('Zuker' algorithm)



Initialization: $F_{ii} = 0$; $C_{ii} = M_{ii} = M^1_{ii} = \infty$

Recursions: $F_{ij} = \min \left\{ F_{i+1,j}, \min\limits_{i<k\leq j} C_{ik} + F_{k+1,j} \right\}$

$$C_{ij} = \min \left\{ H(i,j), \min_{i<k<l<j} C_{kl} + I(i,j;k,l), \min_{i<u<j} M_{i+1,u} + M^1_{u+1,j-1} + a \right\}$$

$$M_{ij} = \min \left\{ \min_{i<u<j} (u-i+1)c + C_{u+i,j} + b, \min_{i<u<j} M_{i,u} + C_{u+1,j} + b, M_{i,j-1} + c \right\}$$

$$M^1_{ij} = \min \left\{ M_{i,j-1} + c, C_{ij} + b \right\}$$

Efficient computation of partition functions

Efficient computation: partition function algorithm (McCaskill, 1990) **can be inferred from the MFE algorithm** by *replacing minimum operations with sums and additions with multiplications*.

$$F_{ij} = \min \left\{ F_{i+1,j}, \min_{i<k\leq j} C_{ik} + F_{k+1,j} \right\}$$

$$C_{ij} = \min \left\{ H(i,j), \min_{i<k<l<j} C_{kl} + I(i,j;k,l), \min_{i<u<j} M_{i+1,u} + M^1_{u+1,j-1} + a \right\}$$

$$M_{ij} = \min \left\{ \min_{i<u<j} (u-i+1)c + C_{u+i,j} + b, \min_{i<u<j} M_{i,u} + C_{u+1,j} + b, M_{i,j-1} + c \right\}$$

$$M^1_{ij} = \min \{ M_{i,j-1} + c, C_{ij} + b \}$$

Efficient computation of partition functions

Efficient computation: partition function algorithm (McCaskill, 1990) **can be inferred from the MFE algorithm** by *replacing minimum operations with sums and additions with multiplications*.

$$
\begin{aligned}
Z_{i,j} &= Z_{i+1,j} + \sum_{i<k\leq j} Z^C_{i,k} Z_{k+1,j} \\
Z^C_{i,j} &= e^{-\beta H(i,j)} + \sum_{i<k<l<j} Z^C_{k,l} e^{-\beta I(i,j,k,l)} + \sum_{i<u<j} Z^M_{i+1,u} Z^{M1}_{u+1,j-1} e^{-\beta a} \\
Z^M_{i,j} &= \sum_{i<u<j} e^{-\beta(u-i+1)c} Z^M_{u+1,j} + \sum_{i<u<j} Z^M_{i,u} Z^C_{u+1,j} e^{-\beta b} + Z^M_{i,j-1} e^{-\beta c} \\
Z^{M1}_{i,j} &= Z^{M1}_{i,j-1} e^{-\beta c} + Z^C_{i,j} e^{-\beta b}
\end{aligned}
$$

(Un)-Ambiguity

Partition function by operator exchange “trick” would not work on ambigous recursions. Why?

(Un)-Ambiguity

Partition function by operator exchange “trick” would not work on ambigous recursions. Why?

prerequisite, violated by ambiguous decomposition

$$\text{traceback} : \text{structure} = 1 : 1$$

(Un)-Ambiguity

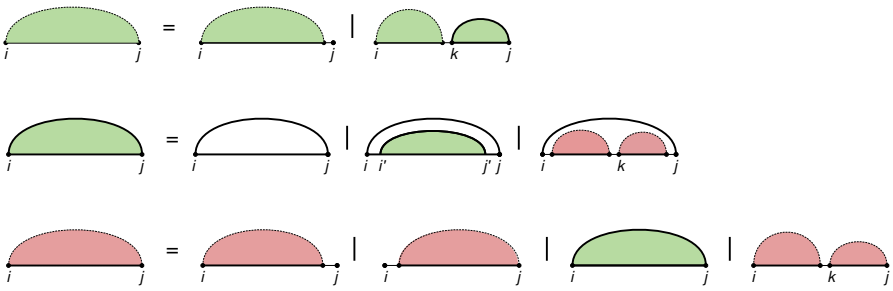
Partition function by operator exchange "trick" would not work on ambigous recursions. Why?

prerequisite, violated by ambiguous decomposition

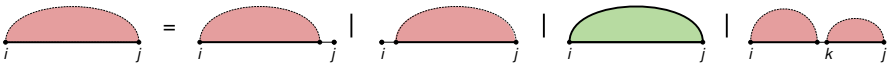
$$\text{traceback} : \text{structure} = 1 : 1$$

Challenge of ambiguity in (original) Zuker-recursions: split of multiloops

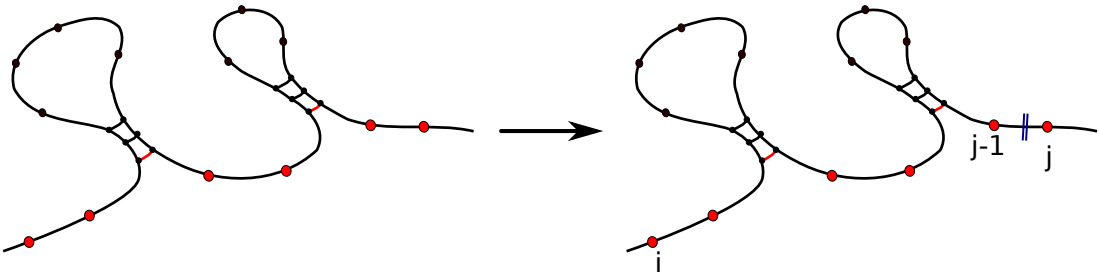
Decomposition of Structure Space by (original) Zuker



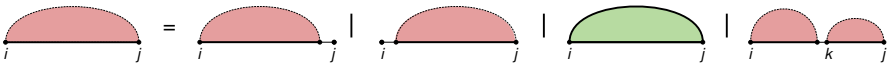
Ambiguity Example

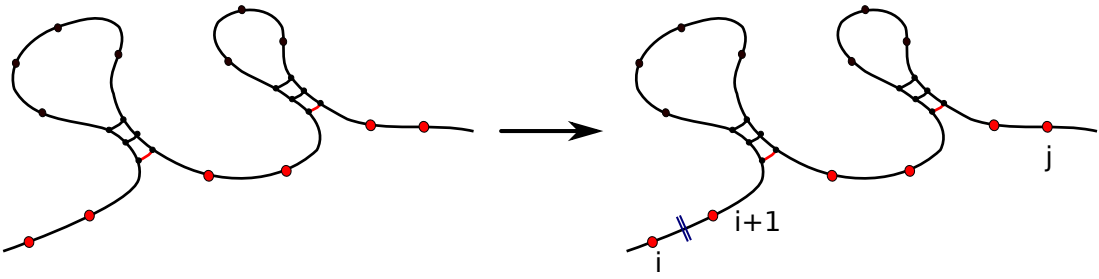


Ambiguity Example

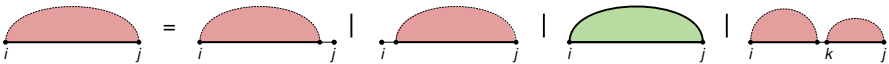


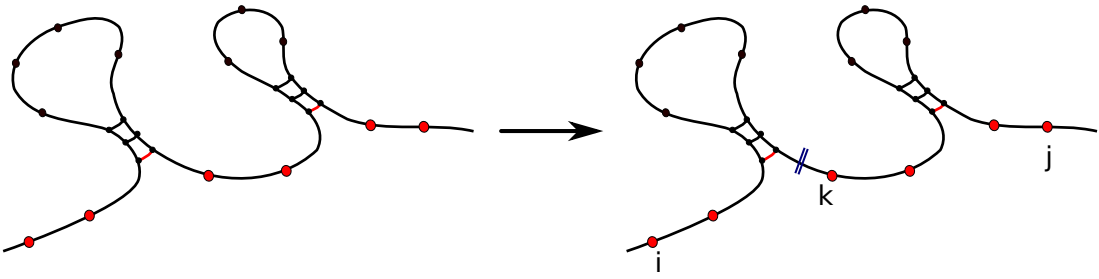
Ambiguity Example



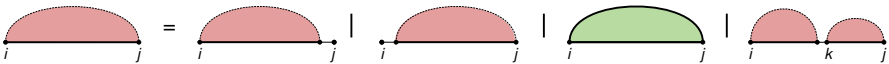


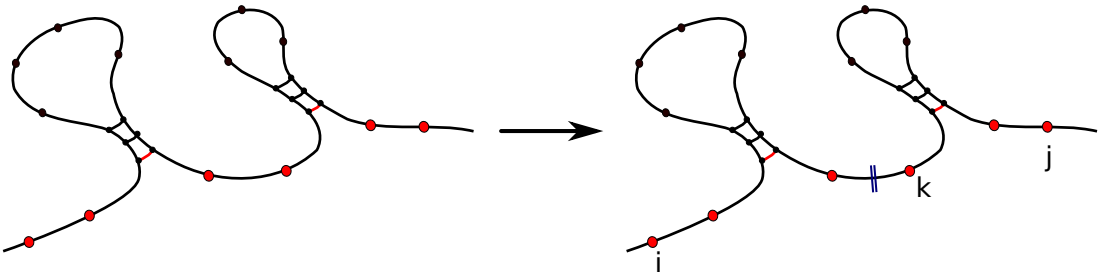
Ambiguity Example



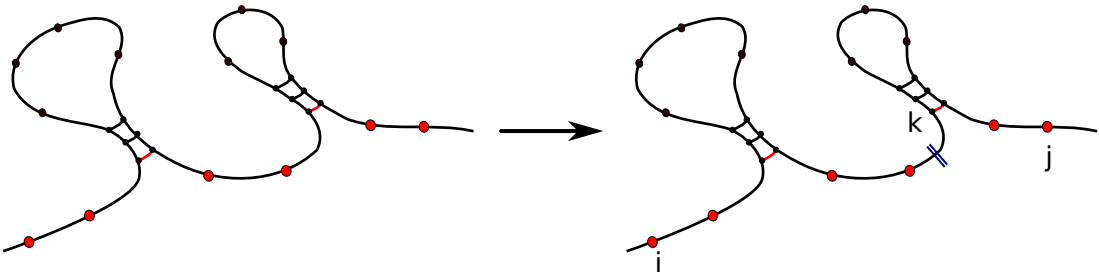


Ambiguity Example

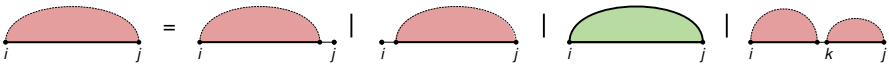


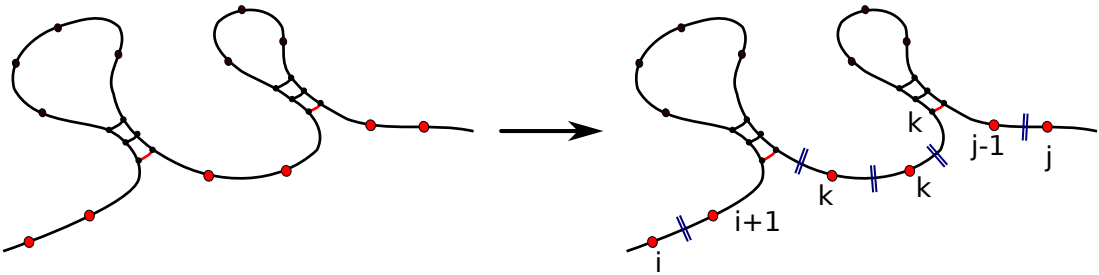


Ambiguity Example

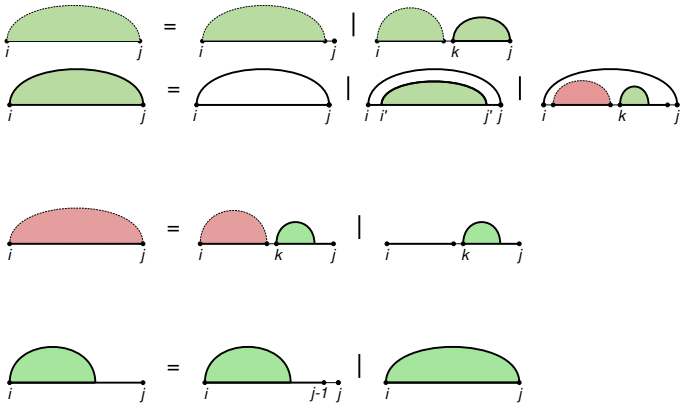


Ambiguity Example





Discarding Ambiguity



Disambiguation due to “fourth” matrix.

Structure and base pair probabilities

The **probability of a specific structure** is given by:

$$Pr(P) = \frac{exp(-\frac{E(P)}{RT})}{Z}$$

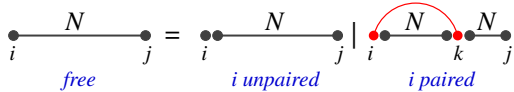
The **probability of a base pair** (i, j) is the fraction of all structures having this base pair and the number of all possible structures.

$$p_{ij} = \sum_{P \ni (i,j)} Pr(P)$$

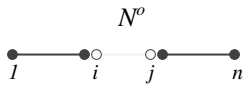
This can be computed efficiently from the product of the inside and *outside* partition functions of a base pair:

$$p_{ij} = \frac{Z_{\texttt{inside}(i,j)} \cdot Z_{\texttt{outside}(i,j)}}{Z}$$

Outside algorithms

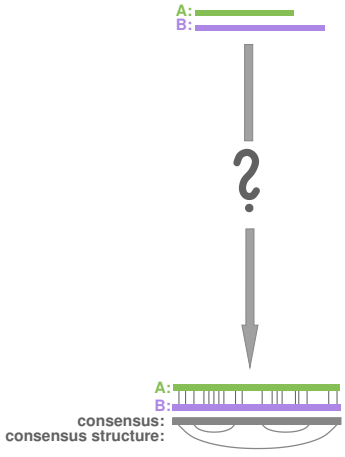


$$N_{i,j} = \max \begin{cases} N_{i-1,j} \\ \max_{i \le k < j} 1 + N_{i+1,k-1} + 1 + N_{k+1,j} : S_i \text{ and } S_k \text{ compl.} \end{cases}$$



$$N^o_{i,j} = \max \begin{cases} N^o_{i-1,j} & i-1 \text{ unpaired} \\ \max\limits_{S_k,\ S_{i-1} \text{ compl.}}^{1 \le k < i} N^o_{k,j} + 1 + N_{k+1,i-2} & i-1 \text{ paired left} \\ \max\limits_{S_{i-1},\ S_k \text{ compl.}}^{j \le k < n} N^o_{i-1,k} + 1 + N_{j,k-1} & i-1 \text{ paired right} \end{cases}$$

Comparative RNA Analysis



adopted from:
[Gardner & Giegerich BMC 2004]

Typical Scenario

Given: set of related RNA sequences

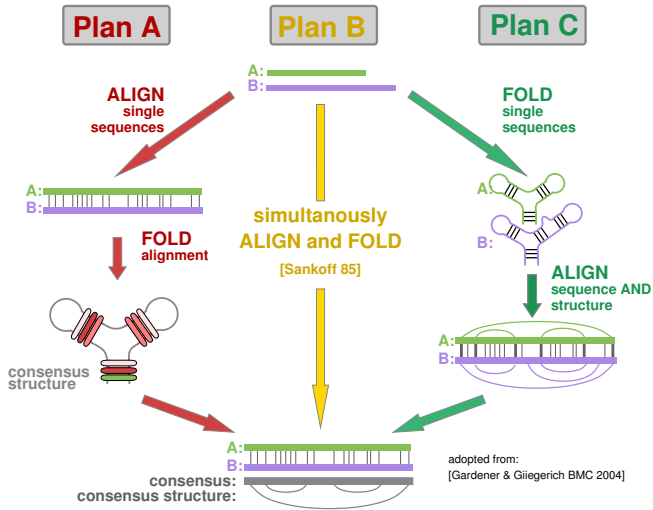
```
>AF008220
GGAGGAUUAGCUCAGCUGGGAGAGCAUCUGCCUUACAAGCAGAGGGUCGGCGGUUCGAGCCCGUCAUCCUCCA
>M68929
GCGGAUAUAACUUAGGGGUUAAAGUUGCAGAUUGUGGCUCUGAAAACACGGGUUCGAAUCCCGUUAUUCGCC
>X02172
GCCUUUAUAGCUUAGUGGUAAAGCGAUAAACUGAAGAUUUAUUUACAUGUAGUUCGAUUCUCAUUAAGGGCA
>Z11880
GCCUUCCUAGCUCAGUGGUAGAGCGCACGGCUUUUAACCGUGUGGUCGUGGGUUCGAUCCCCACGGAAGGCG
>D10744
GGAAAAUUGAUCAUCGGCAAGAUAAGUUAUUUACUAAAUAAUAGGAUUUAAUAACCUGGUGAGUUCGAAUCUCACAUUUUCCG
```

Wanted:

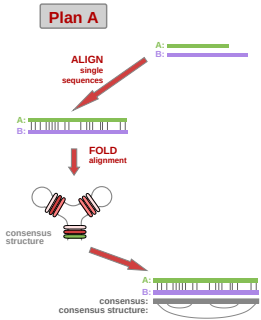
```
AF008220  GGAGGAUU-AGCUCAGCUGGGAGAGCAUCUGCCUUACAAGC---------AGAGGGUCGGCGGUUCGAGCCCGUCAUCCUCCA
M68929    GCGGAUAU-AACUUAGGGGUUAAAGUUGCAGAUUGUGGCUC---------UGAAAA-CACGGGUUCGAAUCCCGUUAUUCGCC
X02172    GCCUUUAU-AGCUUAG-UGGUAAAGCGAUAAACUGAAGAUU---------UAUUUACAUGUAGUUCGAUUCUCAUUAAGGGCA
Z11880    GCCUUCCU-AGCUCAG-UGGUAGAGCGCACGGCUUUUAACC---------GUGUGGUCGUGGGUUCGAUCCCCACGGAAGGCG
D10744    GGAAAAUUGAUCAUCGGCAAGAUAAGUUAUUUACUAAAUAAUAGGAUUUAAUAACCUGGUGAGUUCGAAUCUCACAUUUUCCG

consens   (((((((...((((........))))((((((.......)).........))))....(((((.......)))))))))))).
```

- Typically, only sequences are known, not their structures
- Evolutionary conserved for many RNAs: sequence and structure

RNA Alignment Approaches



Comparative RNA Analysis: Plan A

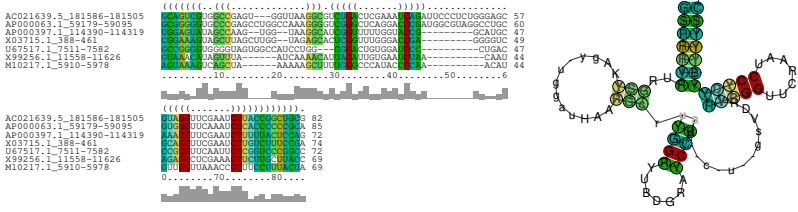


Remarks

- ALIGN: multiple sequence alignment
  (recall Needleman-Wunsch, progressive alignment)
- FOLD: generalize prediction for single sequences

Predict (optimal) consensus structure of an RNA alignment

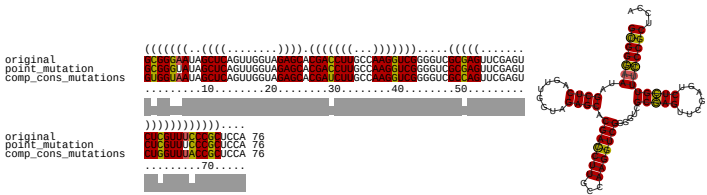
```
CLUSTAL W --- LocARNA 2.0.0RC7


AC021639.5_181586-181505 GCAGUCGUGGCCGAGU---GGUUAAGGCGUCUGACUCGAAAUCAGAUUCCCUCUGGGAGCGUAGGUUCGAAUCCUACCGGCUGCG
AP000063.1_59179-59095   GCGGGGGUGCCCGAGCCUGGCCAAAGGGGUCGGGCUCAGGACCCGAUGGCGUAGGCCUGCGUGGGUUCAAAUCCCACCCCCCGCA
AP000397.1_114390-114319 UGGAGUAUAGCCAAG--UGG--UAAGGCAUCGGUUUUUGGUACCG---------GCAUGCAAAGGUUCGAAUCCUUUUACUCCAG
X03715.1_388-461         CGGAAAGUAGCUUAGCUUGG--UAGAGCACUCCGGUUUUGGGACCGA---------GGGGUCGCAGGUUCGAAUCCUGUCUUUCCGA
U67517.1_7511-7582       GCCGGGGUGGGGUAGUGGCCAUCCUGG---GGGACUGUGGAUCCC----------CUGACCCGGGUUCAAUUCCCGGUCCCGGCC
X99256.1_11558-11626     GUAAACAUAGUUUA------AUCAAAACAUUAGAUUGUGAAUCUAA----------CAAUAGAGGCUCGAAACCUCUUGCUUACC
M10217.1_5910-5978       AGUAAAGUCAGCUA------AAAAAGCUUUUGGGCCCAUACCCCAA----------ACAUGUUGGUUAAACCCCUUCCUUUACUA
```



RNAalifold

IN: RNA alignment

OUT: **Consensus structure of aligned RNAs**

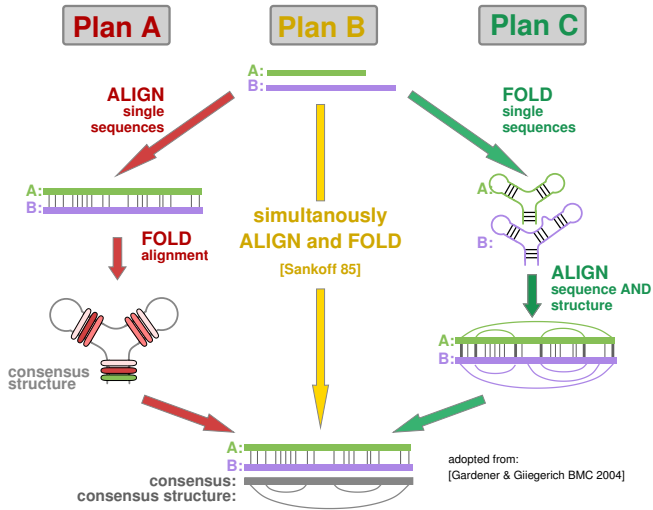


Optimizes **free energy** + **conservation score**

conservation score = compensatory and consistent mutations

- alignment as sequence of alignment columns. Folding of this sequence is analogous to folding of an RNA sequence, 'base pairs' are formed between alignment columns
- Thus, same decomposition as Zuker; but modified scoring: sum of loop energies over all sequences & add conservation score

RNA Alignment Approaches

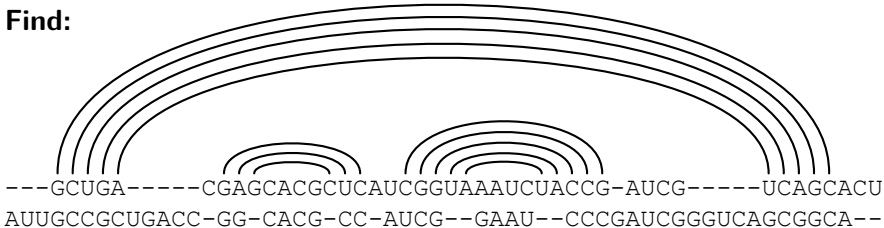


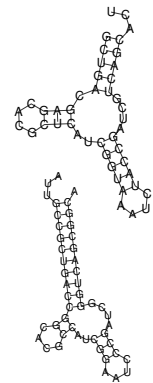
Simultaneous Alignment and Folding [Sankoff]

Given: A = GCUGACGAGCACGCUCAUCGGUAAAUCUACCGAUCGUCAGCACU

& B = AUUGCCGCUGACCGGCACGCCAUCGGAAUCCCGAUCGGGUCAGCGGCA

Find:





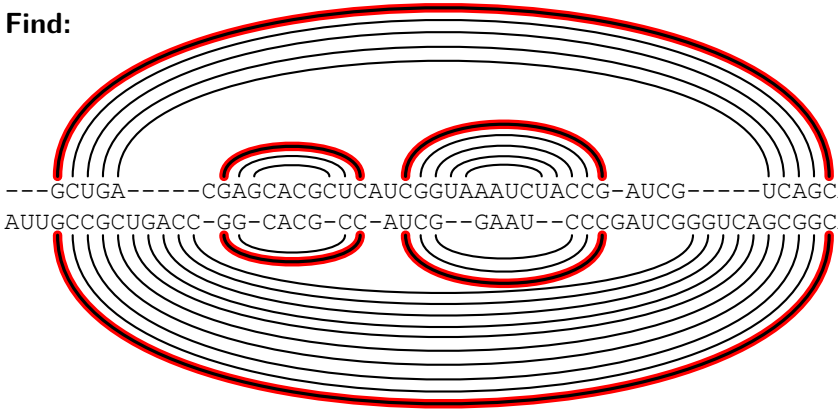
edit distance + energy A + energy B → opt

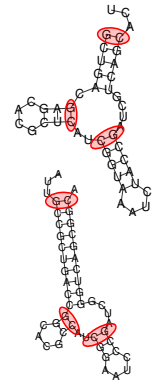
Simultaneous Alignment and Folding [Sankoff]

Given: A = GCUGACGAGCACGCUCAUCGGUAAAUCUACCGAUCGUCAGCACU

& B = AUUGCCGCUGACCGGCACGCCAUCGGAAUCCCGAUCGGGUCAGCGGCA

Find:





edit distance + energy A + energy B → opt

Sankoff Problem Definition

- Idea: Sankoff = Zuker Folding $\times$ Needleman/Wunsch Alignment

- IN: two sequences a and b
- find two **equivalent** structures P_a and P_b
  and **compatible** alignment A of a and b
  such that $Energy(a, P_a) + Energy(b, P_b) + EditDistance(A)$ minimal

- where: *Energy* yields (loop-based) Turner free energy,
  EditDistance is edit distance (base mismatch x, indel y)

- what means *compatible*
  A must align all base pairs $(i_f, i_g) \in P_a$ and $(j_f, j_g) \in P_b$ that enclose branches
  (red base pairs of previous slide)

Constraints

Summarizing, we need to find the optimal structures + alignment with the following constraints:

constraints on the predicted structures:

- must be equivalent (same shape)

constraints on the alignment:

- multiloops must be aligned to their equivalent partner
- hairpin loops must be aligned to their equivalent partner
- each 2-loop (includes stacking and bulge) must be aligned to exactly one other 2-loop or must be aligned to gaps entirely.

Edit distance of sub-sequences

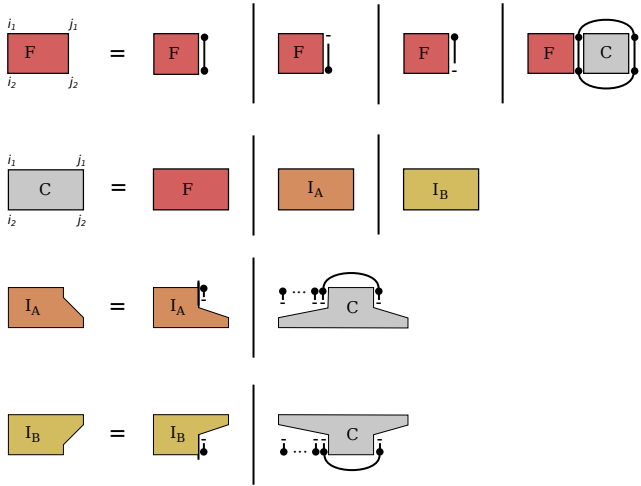
- distance based score
  - x = base mismatch
  - y = base deletion/insertion
- $D(i_1, j_1; i_2, j_2)$ minimum sequence alignment cost between sequences $a_{i_1} \dots a_{j_1}$ and $b_{i_2} \dots b_{j_2}$.
- Recursion: $D(i_1, j_1; i_2, j_2) = \min \begin{cases} D(i_1, j_1 - 1; i_2, j_2 - 1) + \begin{cases} x & \text{if } a_{j_1} \neq b_{j_2} \\ 0 & \text{otherwise} \end{cases} \\ D(i_1, j_1 - 1; i_2, j_2) + y \\ D(i_1, j_1; i_2, j_2 - 1) + y \end{cases}$
- Initialization: $D(i_1, i_1; i_2, i_2) = \begin{cases} x & \text{if } a_{i_1} \neq b_{i_2} \\ 0 & \text{else} \end{cases}$

Simplified Sankoff (without full energy model)

We define the following functions (which will be evaluated recursively):

- $F(i_1, j_1, i_2, j_2)$ best alignment between subsequences $a_{i_1} \dots a_{j_1}$ and $b_{i_2} \dots b_{j_2}$.
- $C(i_1, j_1, i_2, j_2)$ best "closed"alignment between the same subsequences, where we predict the base pairs (i_1, j_1) and (i_2, j_2).
- $I_A(i_1, j_1, i_2, j_2)$ alignment of the same subsequences, which deletes j_1
- $I_B(i_1, j_1, i_2, j_2)$ symmetrically (...which inserts j_2)

Simplified Sankoff (without full energy model)



Sankoff's extreme complexity

space complexity $O(n^4)$

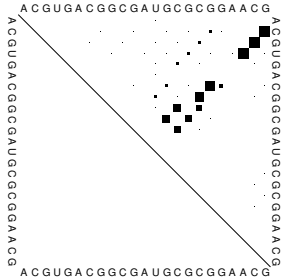
- constant number of matrices
- each of them has $O(n^4)$ entries

time complexity $O(n^6)$

- each entry of matrix D requires constant time
- each entry of F,C, and G requires $O(n^2)$ time (minimize over all h_1, h_2)
- hence: $n^4 \cdot n^2 = n^6$

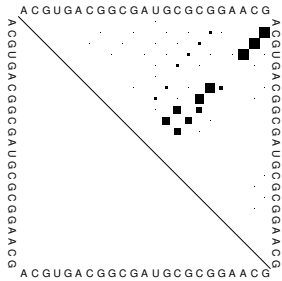
Ensemble-based sparsification (LocARNA)

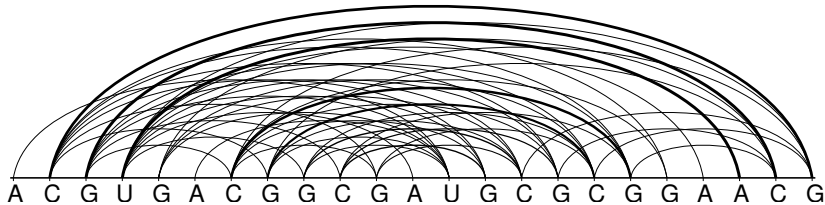
- **Sparsify structure ensemble**



Ensemble-based sparsification (LocARNA)

- **Sparsify structure ensemble**

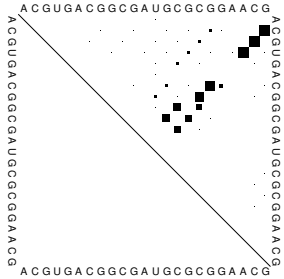


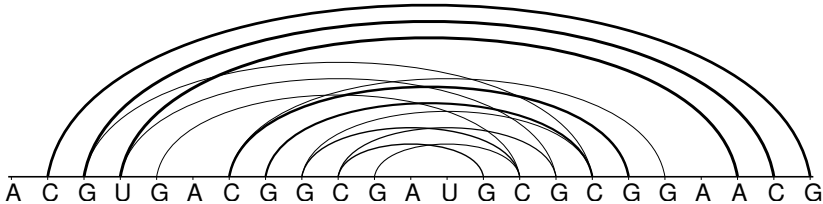


all base pairs

Ensemble-based sparsification (LocARNA)

- **Sparsify structure ensemble**

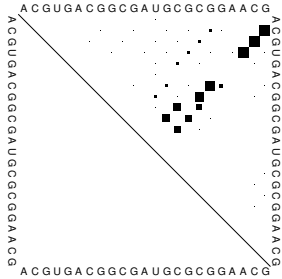


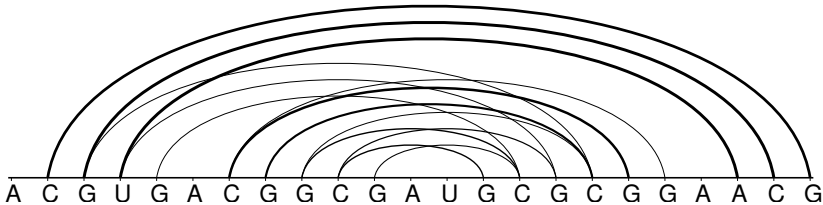


only probable base pairs

Ensemble-based sparsification (LocARNA)

- **Sparsify structure ensemble**





only probable base pairs

- **immediate time improvement** by $O(n^2)$
- **space improvement** by $O(n^2)$ requires rearrangement of recursion evaluation

Rfam / Infernal

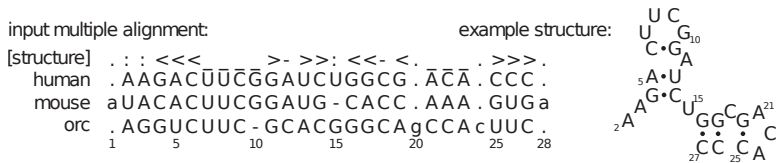
- Infernal: scan genomic data for RNA family members

  Inference of RNA alignments
- important tool for Rfam

  Rfam 10.1 (June 2011, 1973 families)

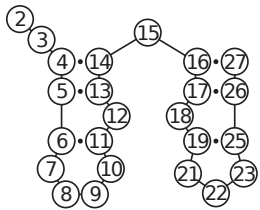
  `http://rfam.sanger.ac.uk/`
- in Rfam: 'hand-curated' seed alignments $\Rightarrow$ full alignments
- use Stochastic Context Free Grammars to model RNA families
- model of a family: *Consensus Model (CM)*



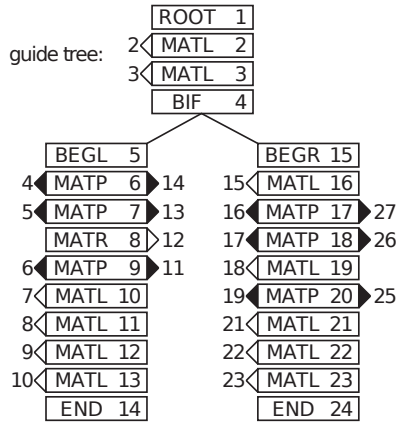
Infernal

Construct grammatical description

consensus structure:



guide tree:



Infernal

- Construct CM from guide tree
- Expand nodes of guide tree:
  Add match, insertion, and deletion states
- learn transition and output probabilities from alignment
- CM comparable to profile HMM for protein families (Pfam)