RNA secondary structure prediction

Jon Malkin/Mukund Narasimhan

December 08, 2004

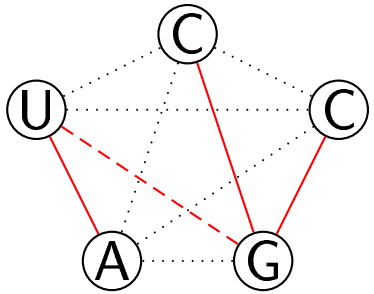
Agenda

1. Techniques that can handle pseudoknots
   - Graph Matching.
   - Iterative Loop Matching.
   - Dynamic Programming.
2. Techniques for improving performance of CMs.
   - Algorithm for reduction of memory consumption.

Graph Matchings for RNA structure prediction

- Given a graph $G = (V, E)$, a matching is a set of edges $S \subseteq E$ such that every vertex is adjacent to at most one edge.
- Maximum cardinality matching : Find matching S so that $|S|$ is maximized.
- Maximum weight matching : Given a weight function $w : E \to \mathbb{R}$, find a matching S so that $\sum_{e \in S} w(e)$ is maximized.
- The maximum weight matching can be solved in $O(|V|^3)$ time and $O(|V|^2)$ space (Gabow's Algorithm).

Graph Matchings for RNA structure prediction



- Given a sequence $AGCCU$, we construct the graph shown above.
- Different edges have different weights.
- Positive/negative weights allowed.

Graph Matchings for RNA structure prediction

[Cary/Stormo 1995, Tabaska et al. 1998]

- Suppose that $b_1 b_2 \ldots b_n$ is the RNA base sequence.
- Construct a graph $G = (V, E)$, where $V = \{b_1, b_2, \ldots, b_n\}$, and $E = \{(v_i, v_j) : 1 \leq i, j \leq n\}$.
- Suppose that for each base pair (v_i, v_j), we have a "score" $w(v_i, v_j)$. Find a matching $S \subseteq E$ that maximizes $\sum_{(v_i,v_j) \in S} w(v_i, v_j)$.
- No "nesting" constraints - handles pseudoknots.

Picking base scores

- If $w(v_i, v_j) = 1$ for all (v_i, v_j), then we get algorithm to maximize the number of base pairs.
- Pick $w(v_i, v_j)$ based on base-pair likelihood ratio.
- Pick $w(v_i, v_j)$ based on thermodynamic scores to minimize energy.
- Pick $w(v_i, v_j)$ to be the mutual information between v_i and v_j to maximize "covarying" structure.
- Pick $w(v_i, v_j)$ based on *Helix plots*.

Helix plots

1. Assign a small positive 'good score' for Watson-Crick and G-U pairs, and larger negative 'bad score' for other base paris, and a still larger negative 'paired gap' score for penalizing single strand deletions. Construct matrix with these initial scores.

2. Scan matrix for potential helices. Scores of base pairs comprising helices of less than some minimum length are changed to the 'bad score'. Base paris forming sufficiently long helices are are increased are increased by adding bonus score proportional to the length of the helix.

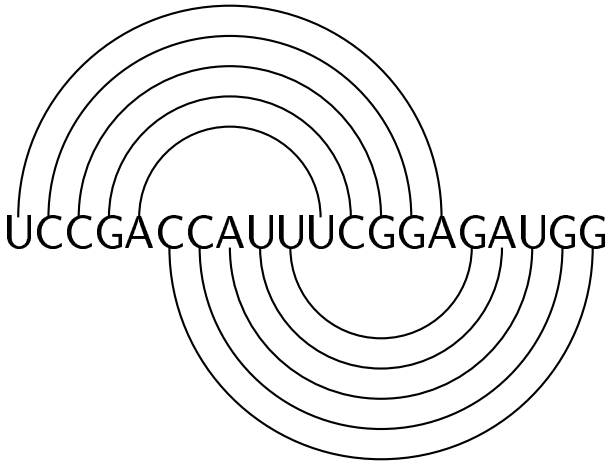
3. Repeat for each sequence in the alignment. Add individual score matrix to get a cumulative matrix.

Advantages/Disadvantages of Matching based methods.

- (+) Can handle pseudoknots very easily.
- (+) Can incorporate (partial) experimental evidence.
- (-) MI/base-scores/uniform-weights does not perform well.
- (-) In general, generates many spurious base paris, and so the output needs to be "filtered".

RNA structure by Iterative Loop Matching

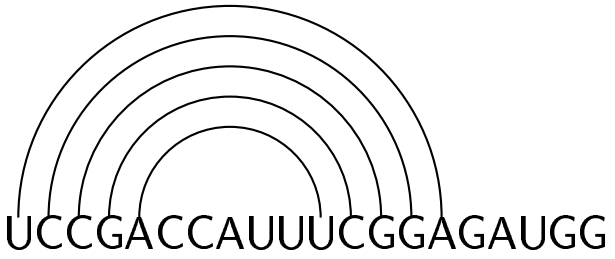
[Ruan/Stormo/Zhang 2004]



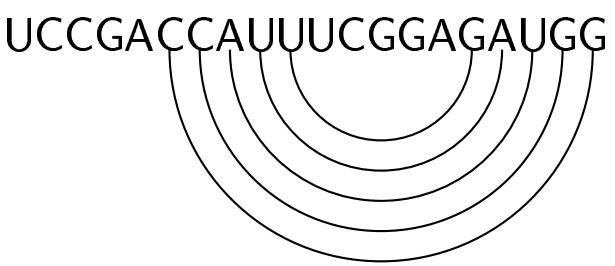
- Problem : Want to pick both sets of base pairs (top and bottom).

RNA secondary structure by Iterative Loop Matching

- Standard Dynamic programming algorithm to maximize number of base-pairs cannot pick all the pairs (top and bottom) because it has a pseudoknot structure.
- Oversimplified solution : Run the DP iteratively, after removing each set of matched pairs
- First run of DP algo finds the top base pairs as the optimal set.
- Run (modified) DP algo again by disabling matches to the already matched set.



Base pairs found by first loop of DP



Base pairs found by second loop of DP

Iterated Loop Matching Algorithm

1: Prepare base-pairing scoring matrix $B[1..n][1..n]$ from aligned sequences. Set $S \leftarrow \phi$.
2: Run the basic LM algorithm using B to produce a base-pair list L.
3: Identify all helices in L and combine helices separeted by small internal loops or bulges. If no helix is found skip to step 7.
4: Compute score of each helix. Pick helix H with highest score and add base pairs to S.
5: 'Remove' positions of H from initial sequence. Update B accordingly.
6: Repeat steps 2-5 till no further bases remain.
7: Report the set S.

Dynamic programming for pseudoknots

[Akutsu 2000]

- A dynamic programming algorithm for predicting pseudoknots
- Time complexity is $O(n^4)$ for simple pseudoknots and $O(n^5)$ for recursive pseudoknots.
- Simplification of the tree adjoining grammar based algorithm of Uemura et al. (removes need for tree adjoining grammar)

Types of psuedoknots

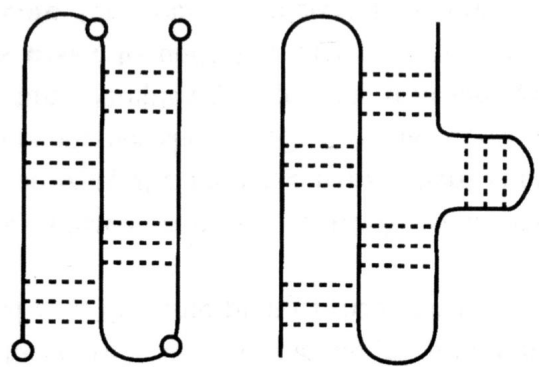


Figure 1: Psuedoknots : Simple and recursive

- A simple pseudoknot has no base pairs between pairs in the pseudoknot
- A recursive pseudoknot may have unknotted intervening pairs
- Generalizations of pseudoknots include any structure with a planar graph with bases connected to at most one non-adjacent base

Algorithmic overview

- Looks for simple pseudoknots using DP on triples, not pairs
- DP equation for overall structure includes pseudoknot option when finding max score
- Generalizing to recursive pseudoknots possible, but slower
- Proves that generalized pseudoknot solutions are NP-hard, although author unsure if such structures exist naturally

Covariance models for RNA structure prediction

- Find RNA structure alignment using a Stochastic Context Free Grammar.
- Can handle pairwise dependences in nonknotted structures.
- The CKY Algorithm (inside algorithm) is a dynamic programming algorithm that guarantees finding the optimal solution in polynomial time.
- Simple extensions allow one to find the top N solutions (also in polynomial time).

The CKY Algorithm

1: $\forall\,(\text{state}, \text{start}, \text{len}) \qquad \alpha_{\text{state}}[\text{start}, \text{len}] \leftarrow -\infty$
2: **for all** len $\in 0$ to $n-1$ **do**
3: **for all** start $\in 0$ to $n - \text{len}$ **do**
4: **for all** state $\in 0$ to M **do**
5: Process Insert/Delete/Match States
6: **if** state is a bifurcation (rule : state $\rightarrow \text{child}_0\,\text{child}_1$) **then**
7: $\forall$split $\in 1$ to len $- 1$
8: $$\alpha_{\text{state}}[\text{start}, \text{len}] \leftarrow \max \begin{cases} \alpha_{\text{state}}[\text{start}, \text{len}] \\ \alpha_{\text{child}_0}[\text{start}, \text{split}] + \alpha_{\text{child}_1}[\text{split}, \text{len}] + \log p(r \end{cases}$$
9: **end if**
10: **end for**
11: **end for**
12: **end for**

Complexity of the CKY algorithm

- Time complexity $O(n^2 \cdot M + B \cdot n^3)$.
- Space complexity is $O(n^2 \cdot M)$.
  - n is the length of the sequence.
  - M is the number of states in the CM (M is $\Theta(n)$).
  - B is the number of bifurcation states (B is $\Theta(1)$).
- Yields total time complexity of $O(n^3)$, space complexity of $O(n^3)$.

Reducing Memory Requirements

[Eddy, 2002]

- Often, memory constraint is more of an issue than the time constraint.
- Large memory requirement leads to increased time requirement due to issues with caches etc.
- The paper gives a way of trading time for space : Memory requirement goes from $O(n^3)$ to $O(n^2 \log n)$.

A sketch of a simplified version for sequences

- Consider problem of aligning two sequences $x_1 x_2 \ldots x_N$ and $y_1 y_2 \ldots y_M$.
- The dynamic programming relation is given by

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + \text{ score for aligning } (x_i, y_j) \\ F(i-1, j) - \text{ cost of a gap} \\ F(i, j+1) - \text{ cost of a gap} \end{cases}$$

- $F(i,j)$ is the cost of best alignment of prefix $x_1 x_2 \ldots x_i$ with $y_1 y_2 \ldots y_j$
- Can similary compute $B(i,j)$ as the cost of best alighment of suffix $x_{i+1} x_{i+2} \ldots x_N$ and $y_{j+1} y_{j+2} \ldots y_M$.

- $F(i,j) + B(i,j)$ is the cost of the best alighment that uses the cell (i,j)
- We know that every alignment must pass through row i somewhere.
- The Basic Idea :
  1. Fix $i \sim N/2$.
  2. Find optimal j so that (i,j) is part of best structure (i.e., $j \in \arg\max F(i,j) + B(i,j)$).
  3. Now align sequence $(1,i)$ to $(1,j)$ and $(i+1,N)$ to $(j+1,M)$.
- The new algorithm is at most $1 + \frac{2}{4} + \frac{4}{16} + \cdots \approx 2$ times as expensive (in time) as the old algorithm.
- However, only requires $O(N)$ memory.

References

[1] P. P. Gardner and R. Giegerich, *A comprehensive comparison of comparative RNA structure prediction approaches*, BMC Bioinformatics 2004 5:140

[2] S. R. Eddy, *A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure*, BMC Bioinformatics 2002, 3:18

[3] J. Ruan, G. D. Stormo and W. Zhang, *An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots*, Bioinformatics 2004, 20(1)

[4] J. E. Tabaska, R. B. Cary, H. N. Gabow and G. D. Stormo, *An RNA folding method capable of identifying pseudoknots and base triples*, Bioinformatics 1998 14(8)

[5] Y. Ji, X. Xu and G. D. Stormo, *A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences*, Bioinformatics 2004 20(10)

[6] T. Akutsu, *Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots*, Discrete Applied Mathematics 2000 (104)