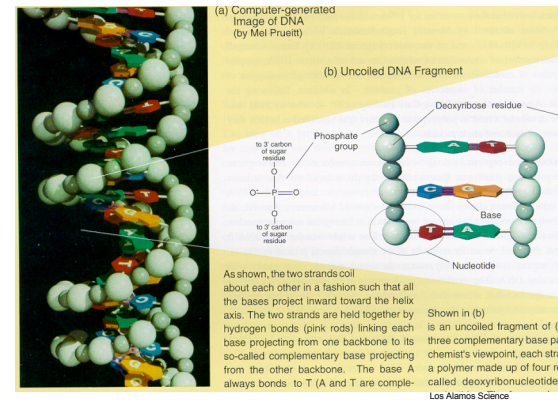


# RNA Secondary Structure

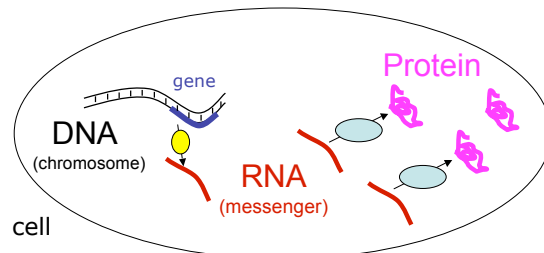
CSE 417  
W.L. Ruzzo

## The Double Helix



## The “Central Dogma” of Molecular Biology

DNA → RNA → Protein

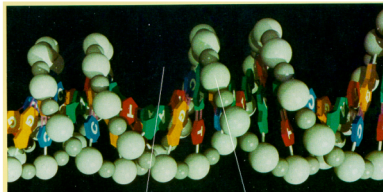


## Non-coding RNA

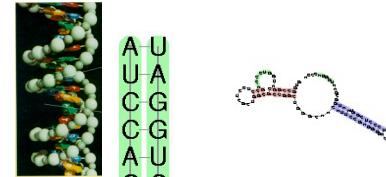
- Messenger RNA - codes for proteins
- Non-coding RNA - all the rest
  - Before, say, mid 1990's, 1-2 dozen known (critically important, but narrow roles: e.g. ribosomal and transfer RNA, splicing, SRP)
- Since mid 90's dramatic discoveries
  - Regulation, transport, stability/degradation
  - E.g. “microRNA”: hundreds in humans
  - E.g. “riboswitches”: thousands in bacteria

## DNA structure: dull

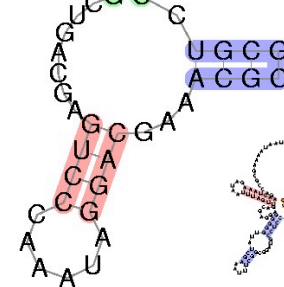
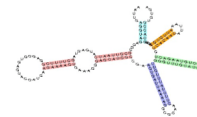
...ACCGCTAGATG...  
| | | | | | | | | |  
...TGGCGATCTAC...



## RNA Structure: Rich



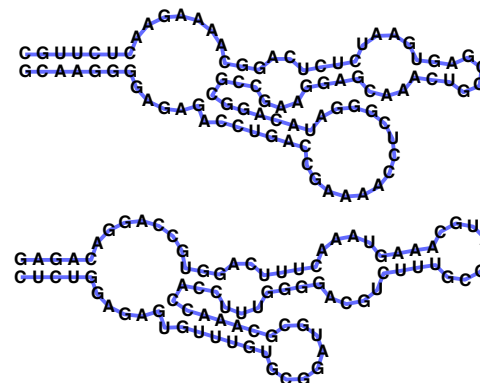
- RNA's fold, and function
- Nature uses what works



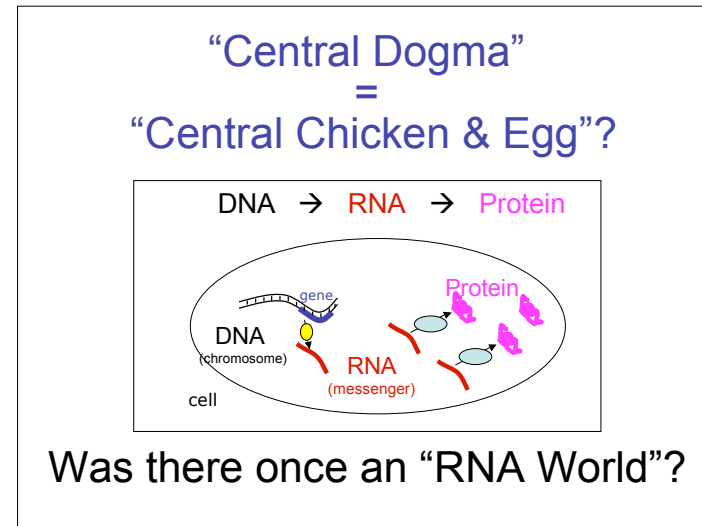
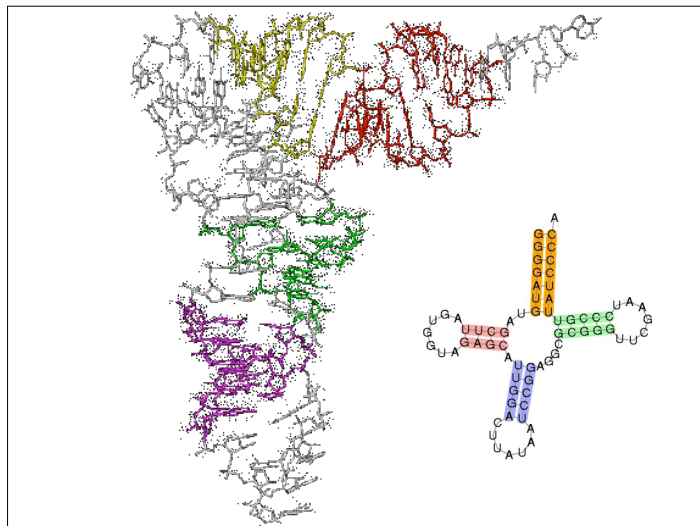
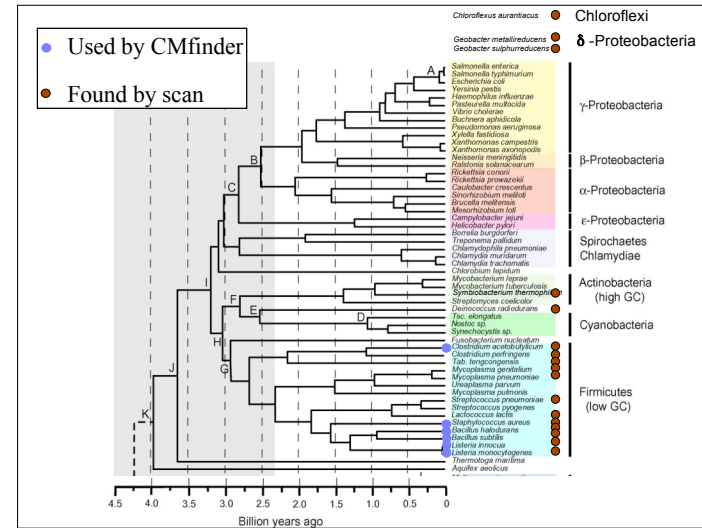
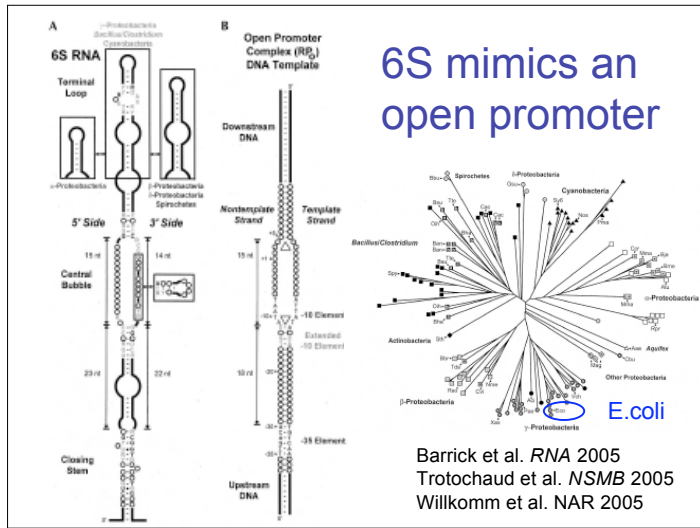
## Why is structure Important?

- For protein-coding, similarity in sequence is a powerful tool for finding related sequences
  - e.g. “hemoglobin” is easily recognized in all vertebrates
- For non-coding RNA, many different sequences have the same structure, and structure is most important for function.
  - So, using structure plus sequence, can find related sequences at much greater evolutionary distances

## Q: What's so hard?



A: Structure often more important than sequence



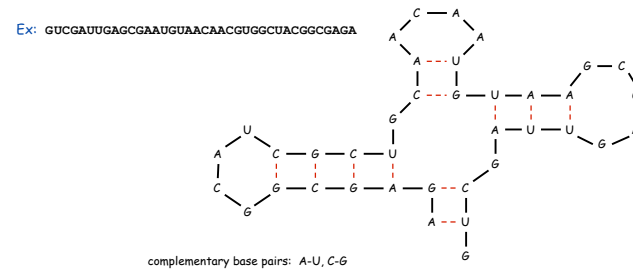
## 6.5 RNA Secondary Structure

### Algorithms

### RNA Secondary Structure

RNA. String  $B = b_1b_2\dots b_n$  over alphabet  $\{A, C, G, U\}$ .

Secondary structure. RNA is single-stranded so it tends to loop back and form base pairs with itself. This structure is essential for understanding behavior of molecule.



### RNA Secondary Structure

Secondary structure. A set of pairs  $S = \{(b_i, b_j)\}$  that satisfy:

- [Watson-Crick.]
  - $S$  is a *matching* and
  - each pair in  $S$  is a Watson-Crick pair: A-U, U-A, C-G, or G-C.
- [No sharp turns.] The ends of each pair are separated by at least 4 intervening bases. If  $(b_i, b_j) \in S$ , then  $i < j - 4$ .
- [Non-crossing.] If  $(b_i, b_j)$  and  $(b_k, b_l)$  are two pairs in  $S$ , then we cannot have  $i < k < j < l$ .

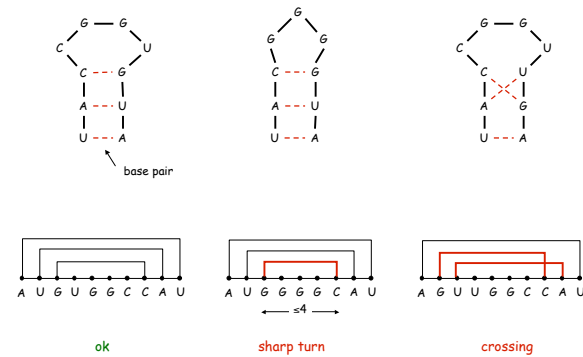
Free energy. Usual hypothesis is that an RNA molecule will form the secondary structure with the optimum total free energy.

approximate by number of base pairs

Goal. Given an RNA molecule  $B = b_1b_2\dots b_n$ , find a secondary structure  $S$  that maximizes the number of base pairs.

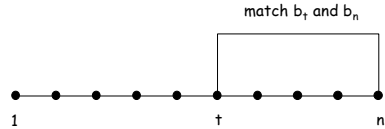
### RNA Secondary Structure: Examples

Examples.



### RNA Secondary Structure: Subproblems

First attempt.  $OPT(j) =$  maximum number of base pairs in a secondary structure of the substring  $b_1b_2\dots b_j$ .



Difficulty. Results in two sub-problems.

- Finding secondary structure in:  $b_1b_2\dots b_{t-1}$ . ←  $OPT(t-1)$
- Finding secondary structure in:  $b_{t+1}b_{t+2}\dots b_{n-1}$ . ← need more sub-problems

### Dynamic Programming Over Intervals

Notation.  $OPT(i, j) =$  maximum number of base pairs in a secondary structure of the substring  $b_i b_{i+1} \dots b_j$ .

- Case 1. If  $i \geq j - 4$ .
  - $OPT(i, j) = 0$  by no-sharp turns condition.
- Case 2. Base  $b_j$  is not involved in a pair.
  - $OPT(i, j) = OPT(i, j-1)$
- Case 3. Base  $b_j$  pairs with  $b_t$  for some  $i \leq t < j - 4$ .
  - non-crossing constraint decouples resulting sub-problems
  - $OPT(i, j) = 1 + \max_t \{ OPT(i, t-1) + OPT(t+1, j-1) \}$
  - ↑  
take max over  $t$  such that  $i \leq t < j-4$  and  $b_t$  and  $b_j$  are Watson-Crick complements

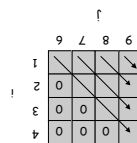
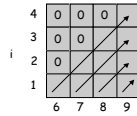
Remark. Same core idea in CKY algorithm to parse context-free grammars.

### Bottom Up Dynamic Programming Over Intervals

- Q. What order to solve the sub-problems?  
A. Do shortest intervals first.

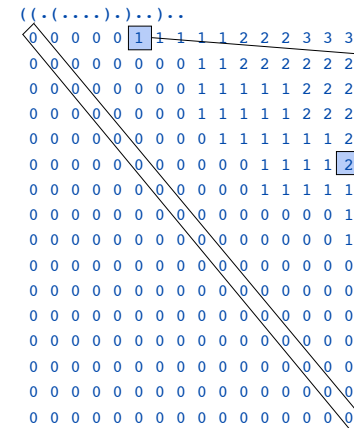
```

RNA( $b_1, \dots, b_n$ ) {
  for  $k = 5, 6, \dots, n-1$ 
    for  $i = 1, 2, \dots, n-k$ 
       $j = i + k$ 
      Compute  $M[i, j]$ 
  return  $M[1, n]$  using recurrence
}
    
```



Running time.  $O(n^3)$ .

CUCCGGUUGCAAUGUC n=16



E.g.:  
 $OPT(1,6) = 1$ :  
CUCCGG  
(...)

E.g.:  
 $OPT(6,16) = 2$ :  
GUUGCAAUGUC  
(- (...)-...)