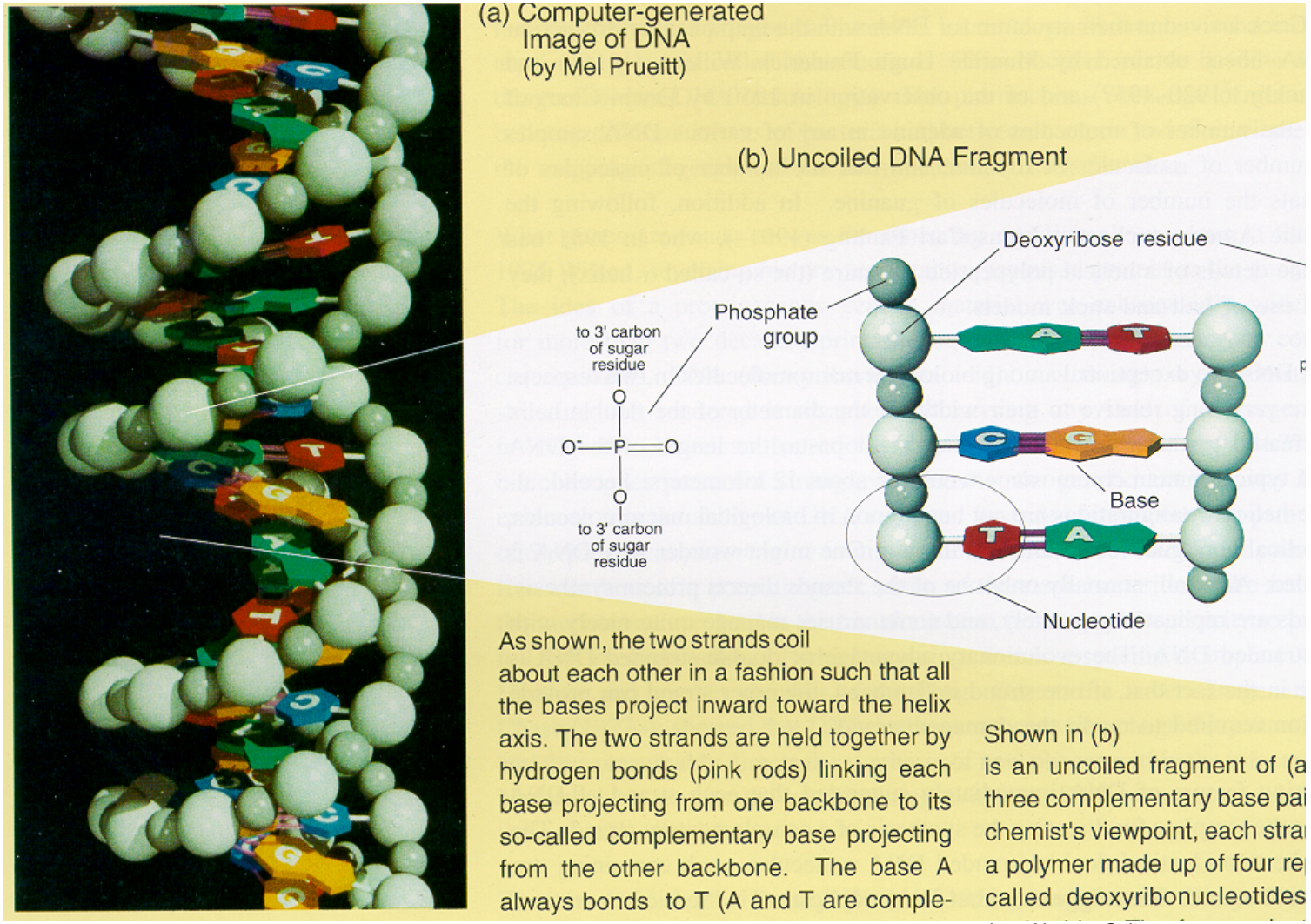
RNA Secondary Structure

CSE 417

W.L. Ruzzo

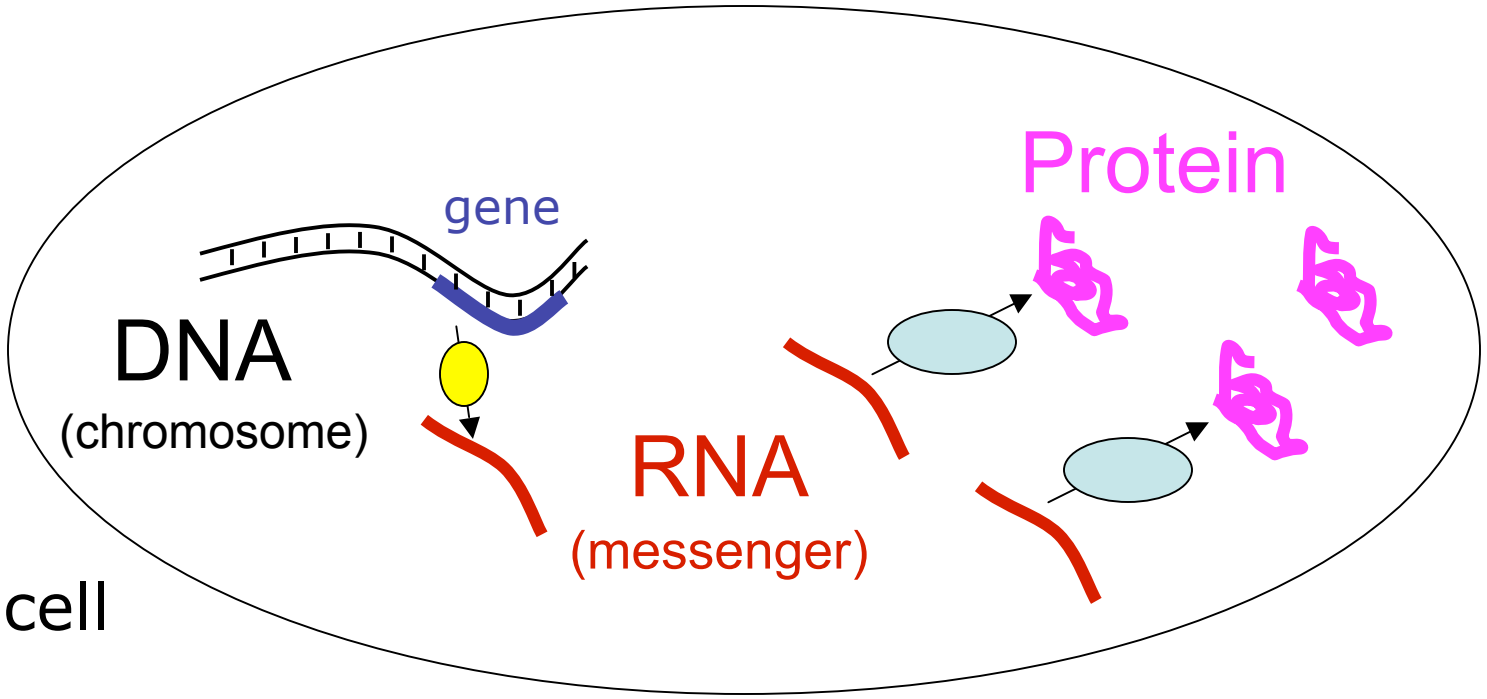
The Double Helix



Los Alamos Science

The “Central Dogma” of Molecular Biology

DNA → RNA → Protein



Non-coding RNA

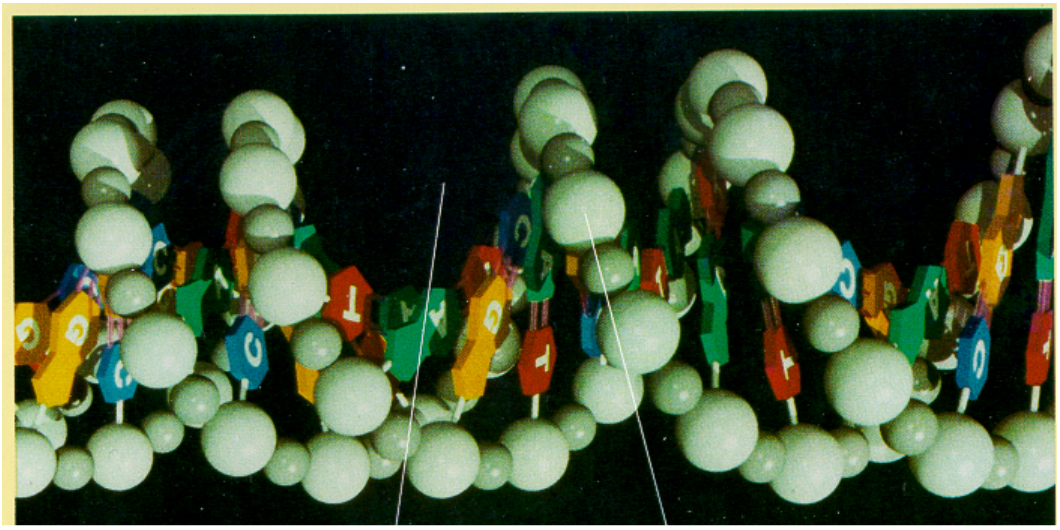
- Messenger RNA - codes for proteins
- Non-coding RNA - all the rest
  - Before, say, mid 1990's, 1-2 dozen known (critically important, but narrow roles: e.g. ribosomal and transfer RNA, splicing, SRP)
- Since mid 90's dramatic discoveries
    - Regulation, transport, stability/degradation
    - E.g.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

DNA structure: dull

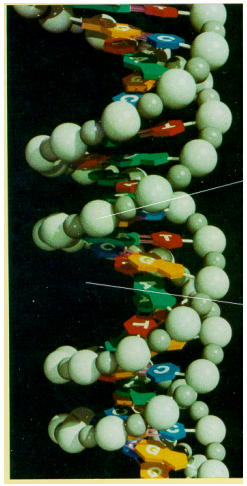
…ACCGCTAGATG…

| | | | | | | | | | |

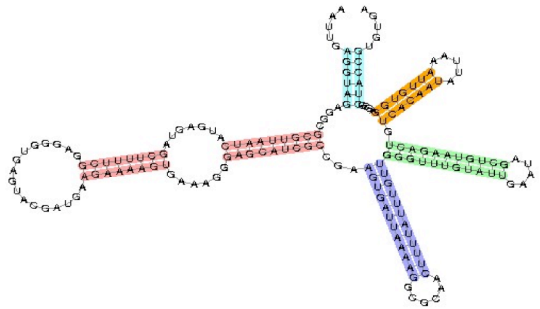
…TGGCGATCTAC…

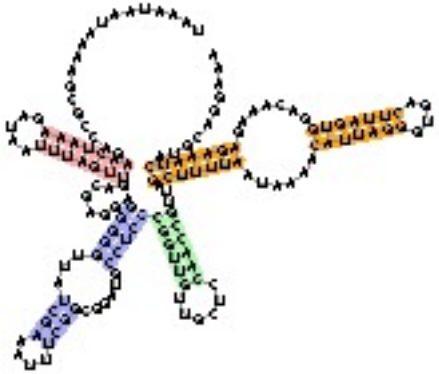


RNA Structure: Rich



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

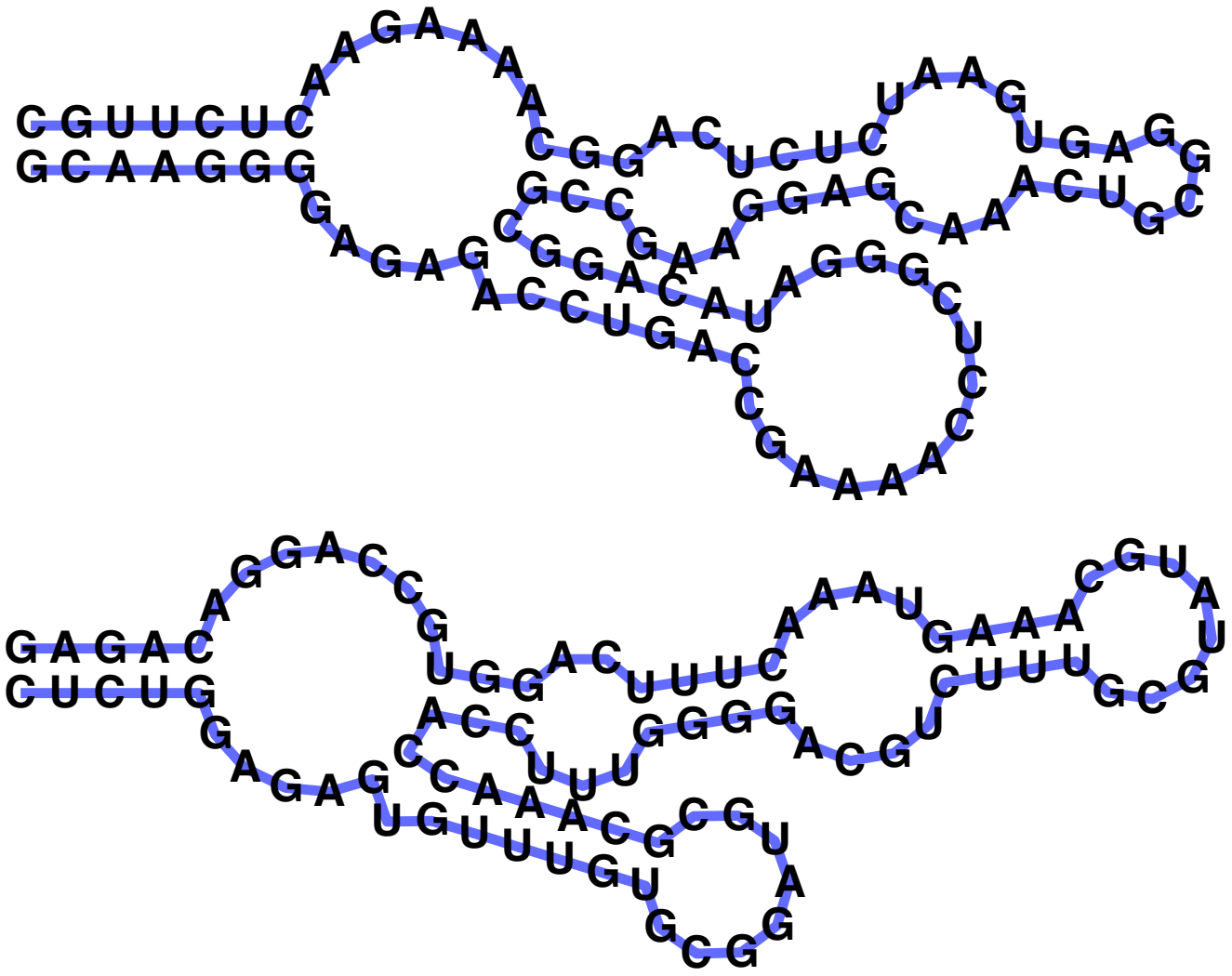




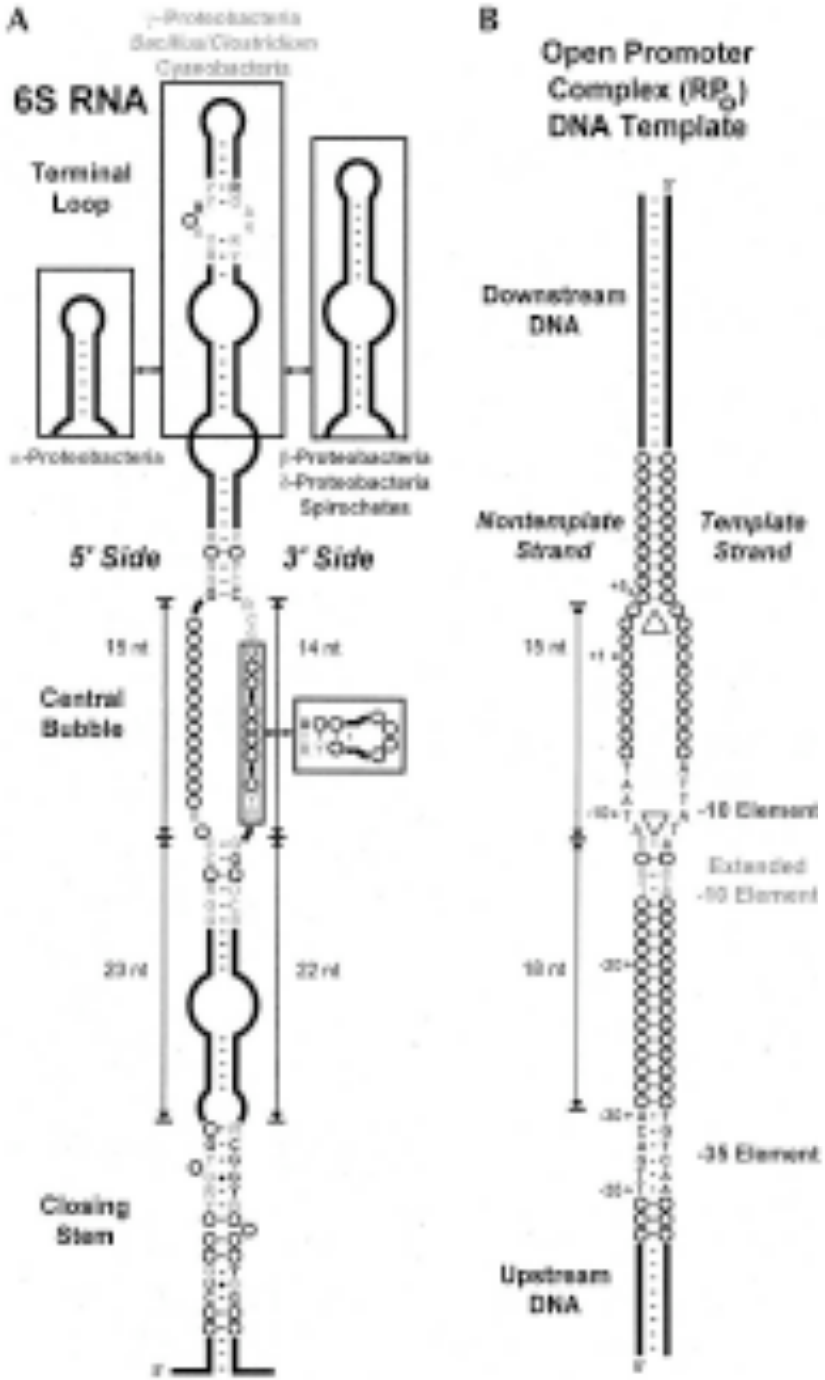
Why is structure Important?

- 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

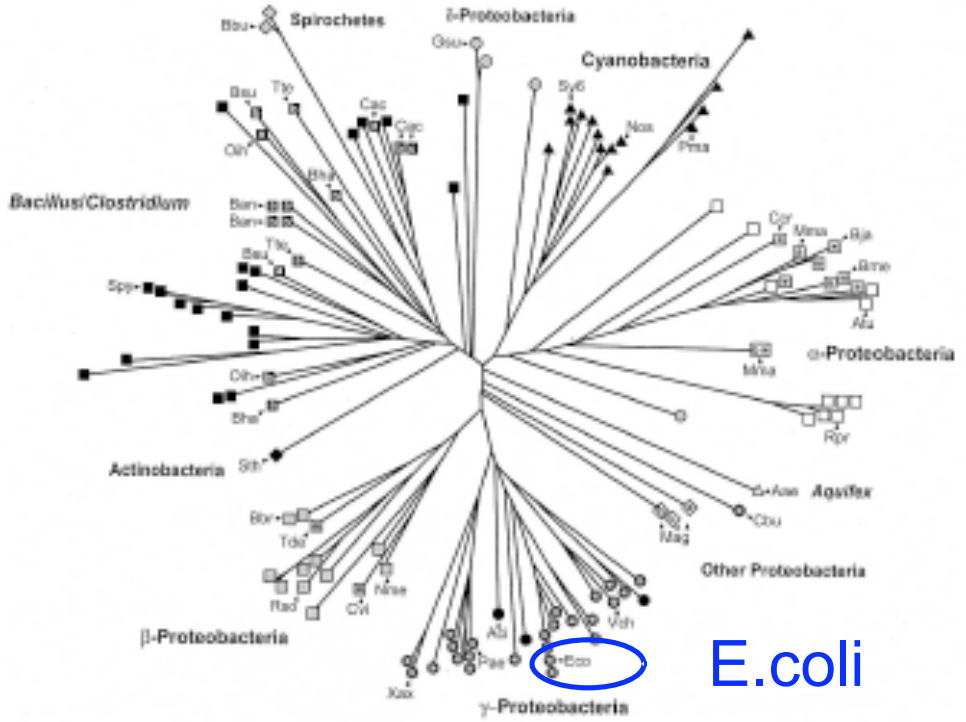
Q: What's so hard?



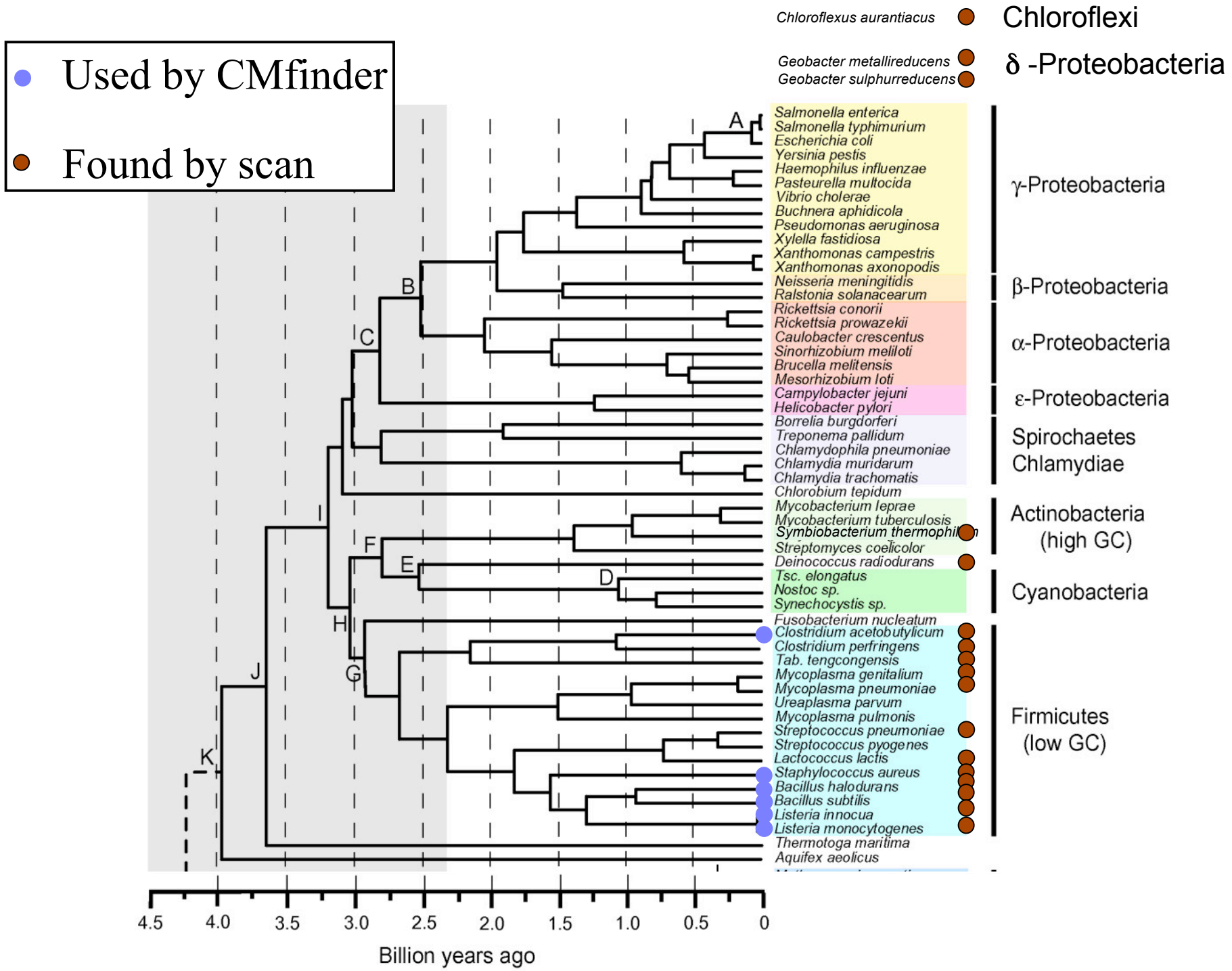
A: Structure often more important than sequence

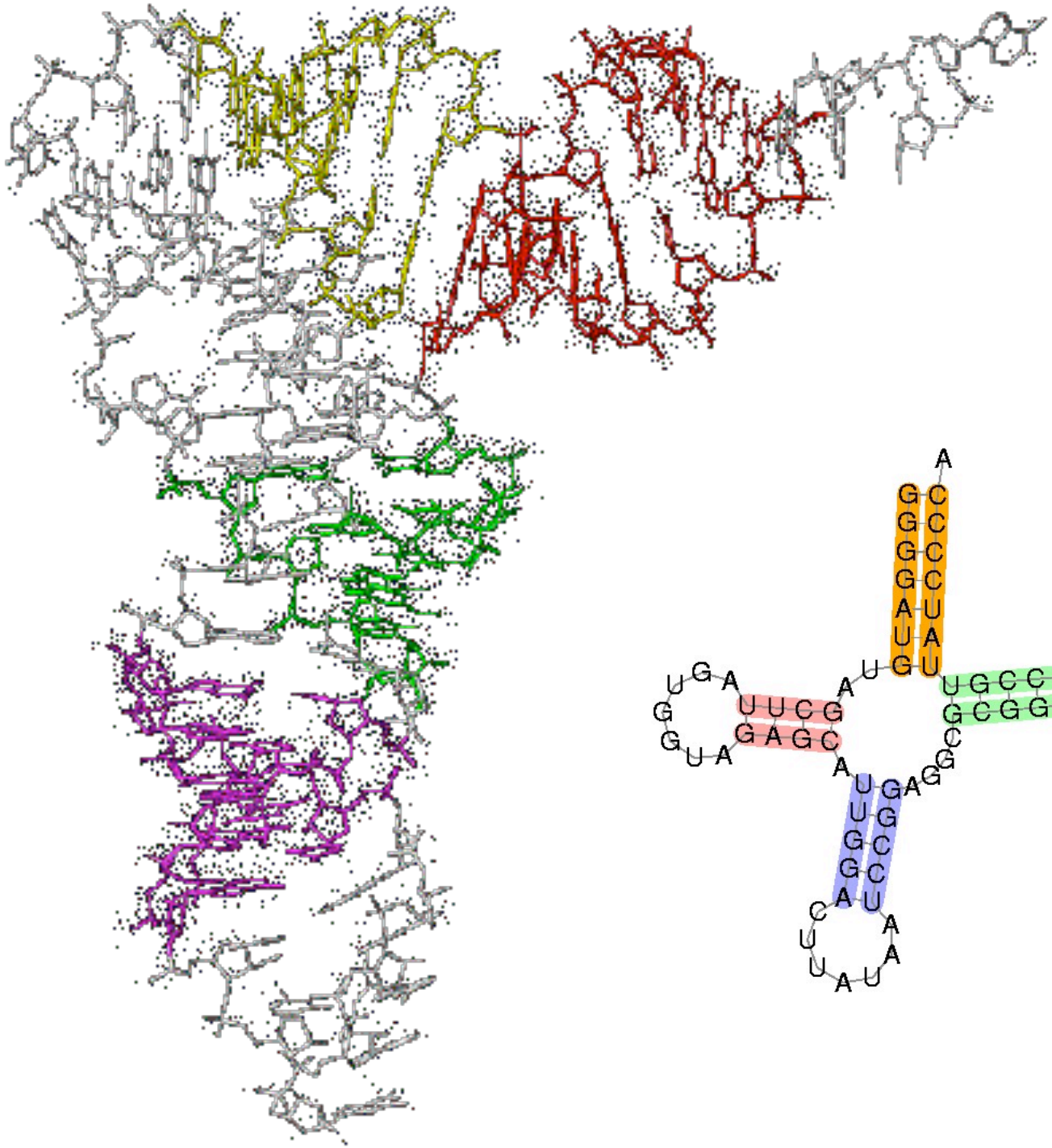


6S mimics an open promoter

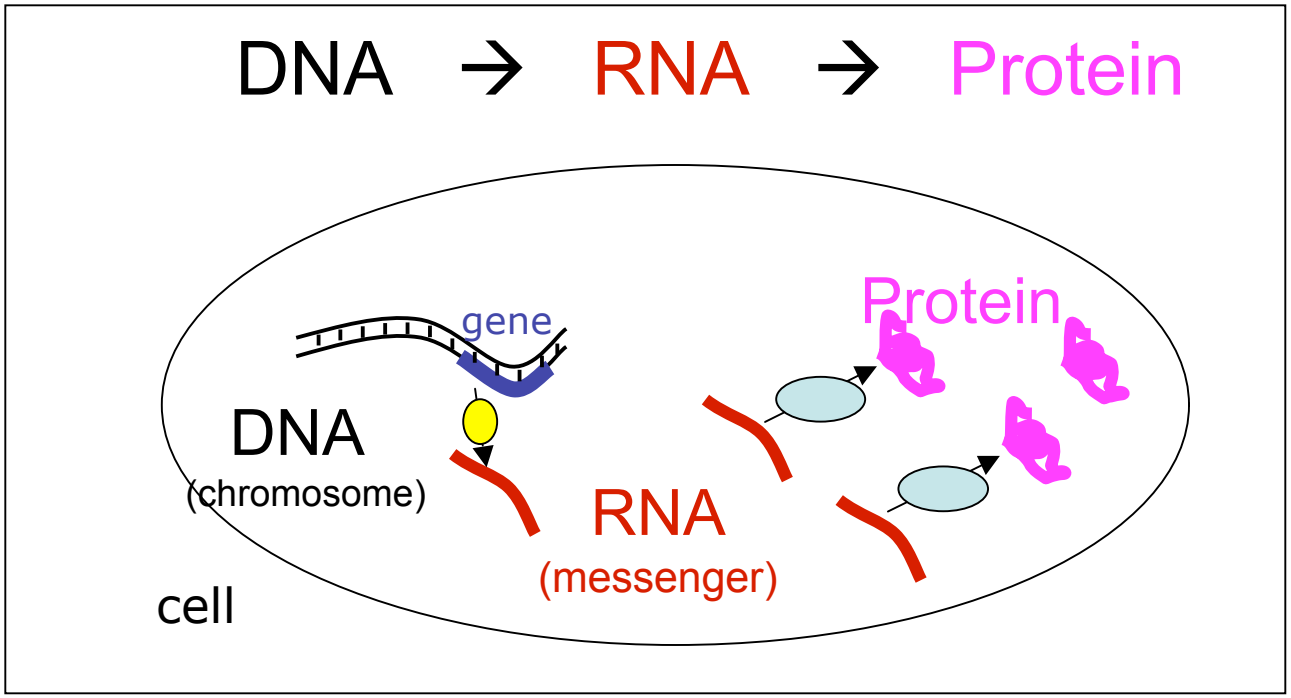


Barrick et al. *RNA* 2005
Trotochaud et al. *NSMB* 2005
Willkomm et al. NAR 2005


Used by CMfinder
Found by scan
Chloroflexus aurantiacus
Chloroflexi
Geobacter metallireducens
Geobacter sulphurreducens
δ -Proteobacteria
Salmonella enterica
Salmonella typhimurium
Escherichia coli
Yersinia pestis
Haemophilus influenzae
Pasteurella multocida
Vibrio cholerae
Buchnera aphidicola
Pseudomonas aeruginosa
Xylella fastidiosa
Xanthomonas campestris
Xanthomonas axonopodis
γ-Proteobacteria
Neisseria meningitidis
Ralstonia solanacearum
β-Proteobacteria
Rickettsia conorii
Rickettsia prowazekii
Caulobacter crescentus
Sinorhizobium meliloti
Brucella melitensis
Mesorhizobium loti
α-Proteobacteria
Campylobacter jejuni
Helicobacter pylori
ε-Proteobacteria
Borrelia burgdorferi
Treponema pallidum
Chlamydophila pneumoniae
Chlamydia muridarum
Chlamydia trachomatis
Spirochaetes
Chlamydiae
Chlorobium tepidum
Mycobacterium leprae
Mycobacterium tuberculosis
Symbiobacterium thermophil
Streptomyces coelicolor
Actinobacteria
(high GC)
Deinococcus radiodurans
Tsc. elongatus
Nostoc sp.
Synechocystis sp.
Cyanobacteria
Fusobacterium nucleatum
Clostridium acetobutylicum
Clostridium perfringens
Tab. tengcongensis
Mycoplasma genitalium
Mycoplasma pneumoniae
Ureaplasma parvum
Mycoplasma pulmonis
Streptococcus pneumoniae
Streptococcus pyogenes
Lactococcus lactis
Staphylococcus aureus
Bacillus halodurans
Bacillus subtilis
Listeria innocua
Listeria monocytogenes
Firmicutes
(low GC)
Thermotoga maritima
Aquifex aeolicus
A
B
C
D
E
F
G
H
I
J
K
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
Billion years ago



“Central Dogma”
=
“Central Chicken & Egg”?



Was there once an “RNA World”?

6.5 RNA Secondary Structure

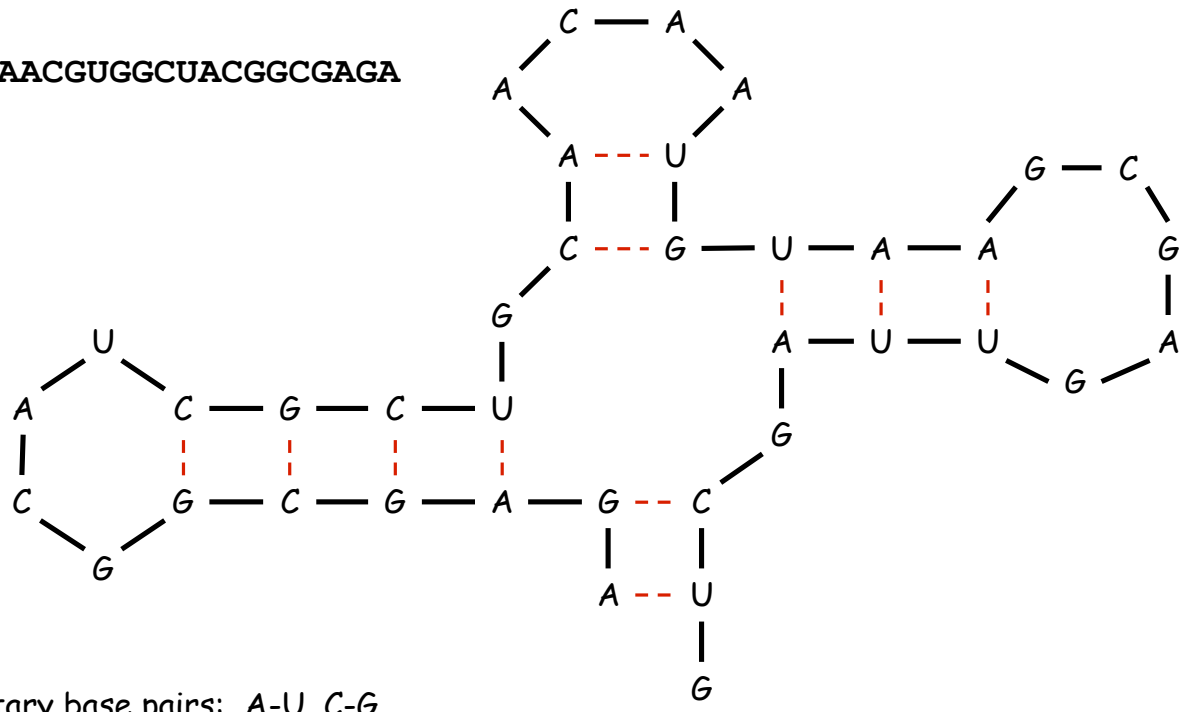
Algorithms

RNA Secondary Structure

RNA. String B = $b_1 b_2 \ldots b_n$ 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.

Ex: **GUCGAUUGAGCGAAUGUAACAACGUGGCUACGGCGAGA**



complementary base pairs: A-U, C-G

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 $i < k < j < l$.

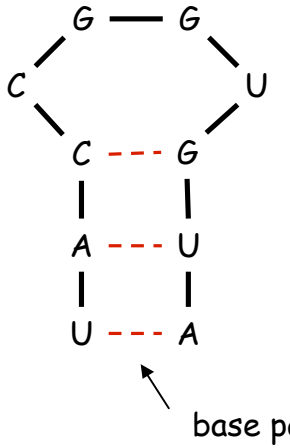
Free energy. Usual hypothesis is that an RNA molecule will form the secondary structure with the optimum total free energy.

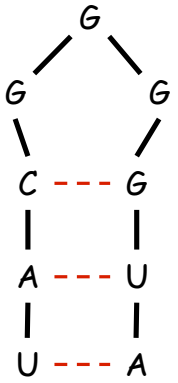
approximate by number of base pairs

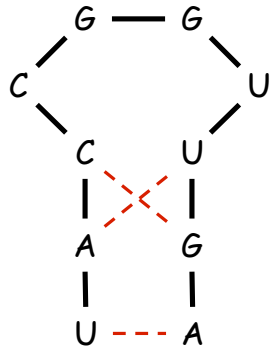
Goal. Given an RNA molecule $B = b_1 b_2 \ldots b_n$, find a secondary structure S that maximizes the number of base pairs.

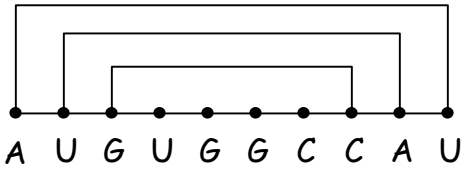
RNA Secondary Structure: Examples

Examples.

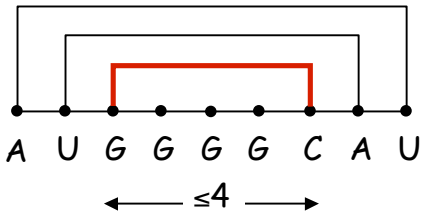




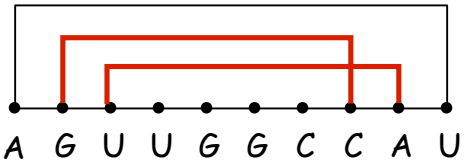




ok



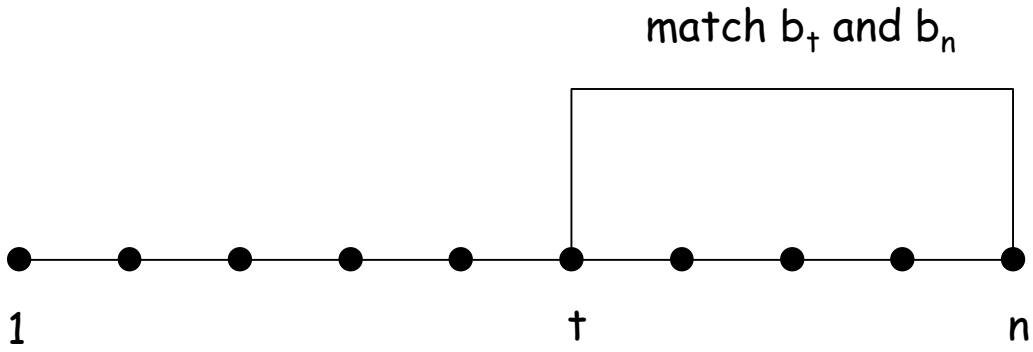
sharp turn



crossing

RNA Secondary Structure: Subproblems

First attempt. OPT(j) = maximum number of base pairs in a secondary structure of the substring $b_1 b_2 \ldots b_j$.



Difficulty. Results in two sub-problems.

- Finding secondary structure in: $b_1 b_2 \ldots b_{t-1}$. ← OPT(t-1)
- Finding secondary structure in: $b_{t+1} b_{t+2} \ldots b_{n-1}$. ← need more sub-problems

Dynamic Programming Over Intervals

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

- 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

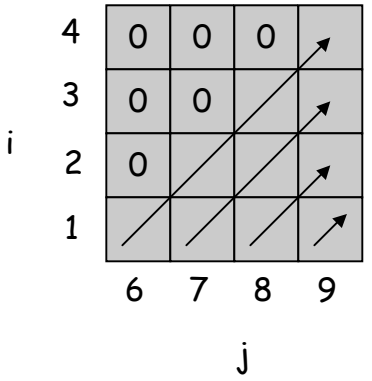
Remark. Same core idea in CKY algorithm to parse context-free grammars.

Bottom Up Dynamic Programming Over Intervals

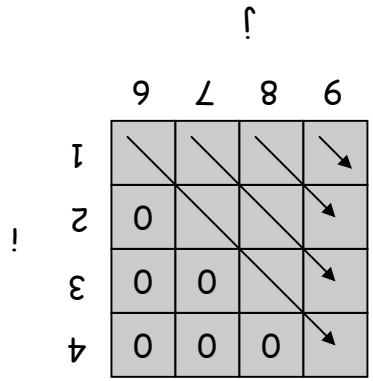
Q. What order to solve the sub-problems?

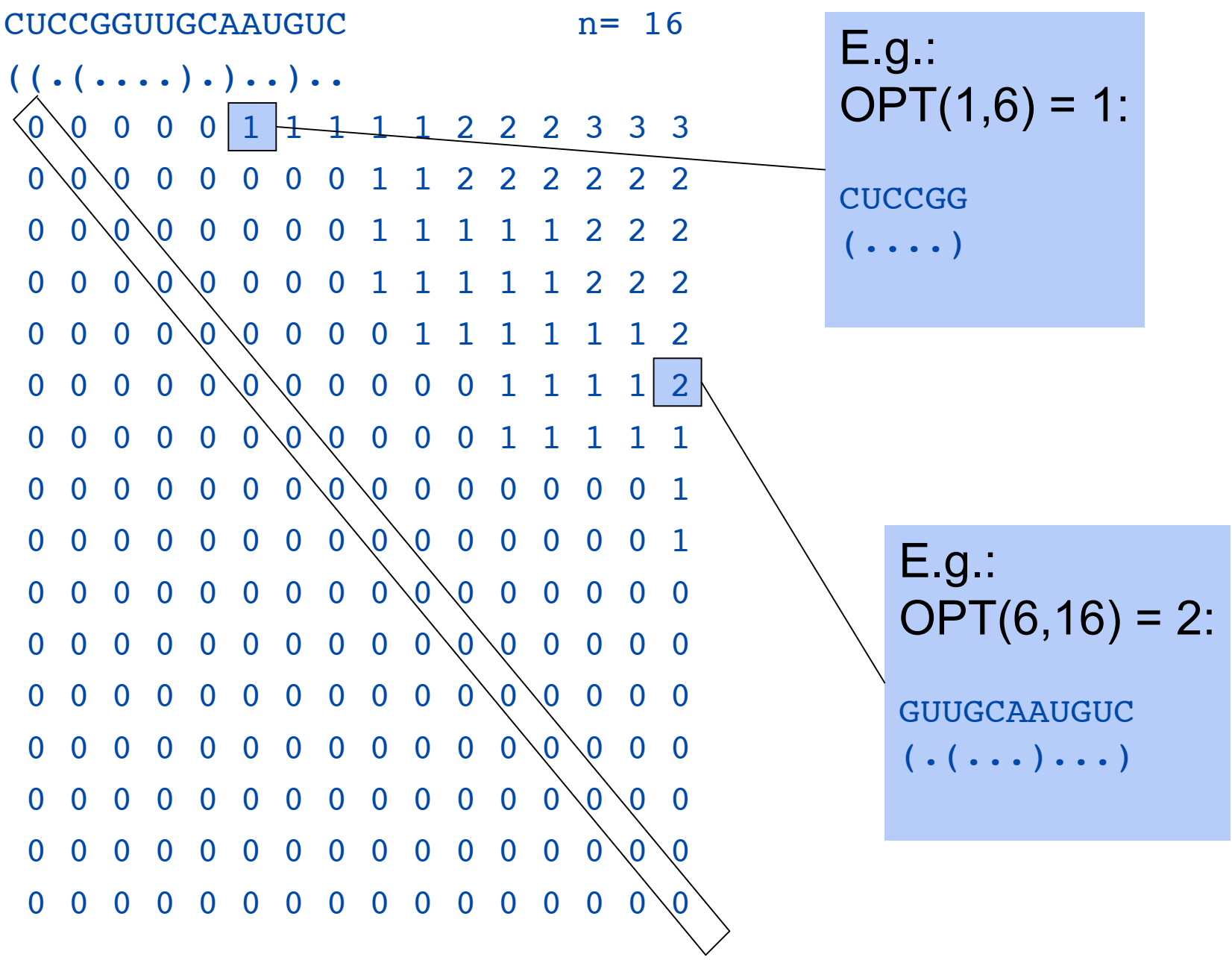
A. Do shortest intervals first.

```
RNA(b1,…,bn) {
    for k = 5, 6, …, n-1
        for i = 1, 2, …, n-k
            j = i + k
            Compute M[i, j]
                                  ↖ using recurrence
    return M[1, n]
}
```



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




CUCCGGUUGCAAUGUC n= 16
((.(....).)..)..
0 0 0 0 0 1 1 1 1 1 2 2 2 3 3 3
0 0 0 0 0 0 0 0 1 1 2 2 2 2 2 2
0 0 0 0 0 0 0 0 1 1 1 1 1 2 2 2
0 0 0 0 0 0 0 0 1 1 1 1 1 2 2 2
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 2
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 2
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
E.g.:
OPT(1,6) = 1:
CUCCGG
(....)
E.g.:
OPT(6,16) = 2:
GUUGCAAUGUC
(.(...)...)