CSE 417: Algorithms and Computational Complexity

W. L. Ruzzo

Dynamic Programming, II
RNA Folding

Outline

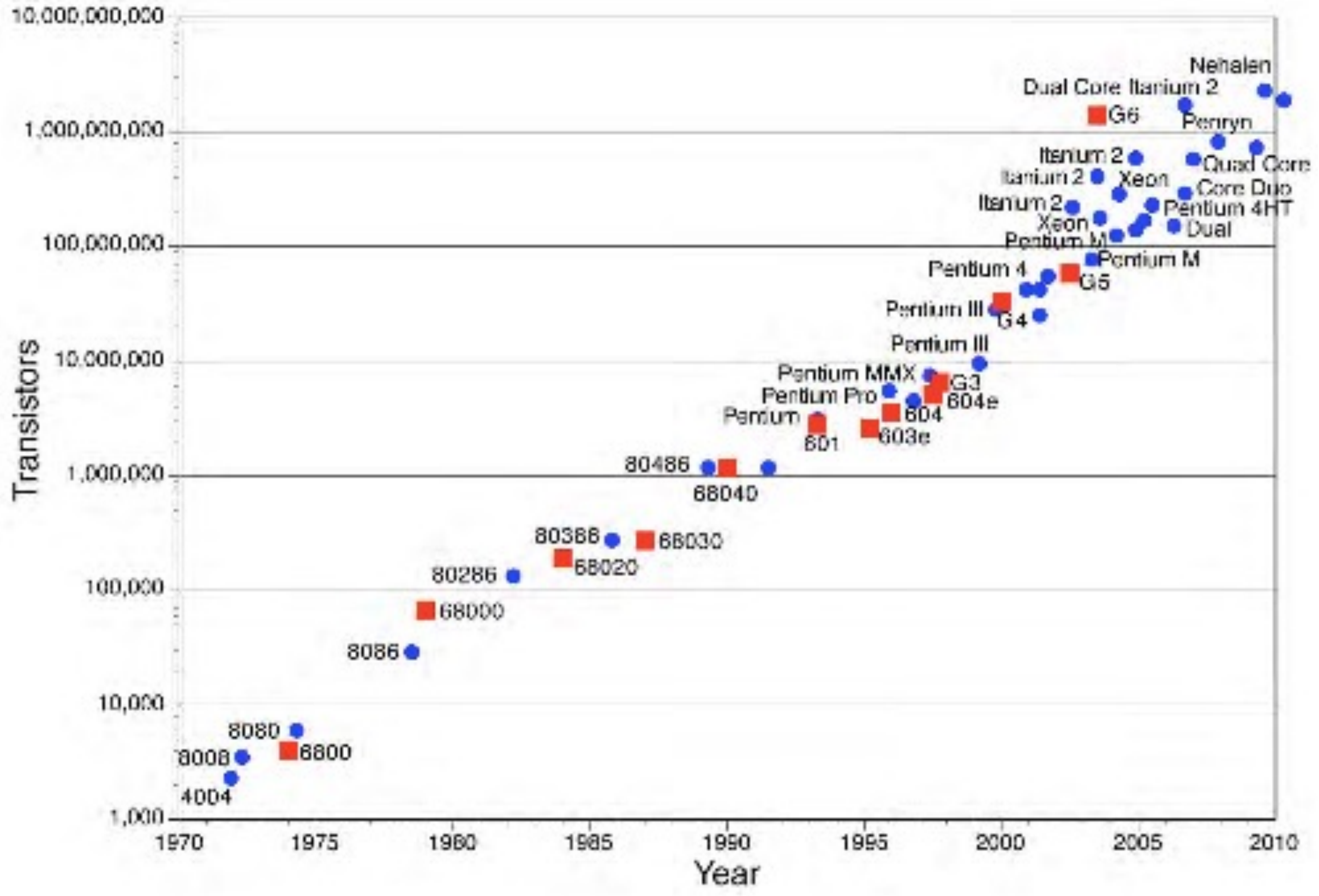
A few (well, ~25) slides on *applications* of dynamic programming in biology (You might enjoy a slightly deeper look at the use of some of the algorithms we study)

- Sequence alignment
- RNA structure

Algorithms for RNA structure (probable HW)

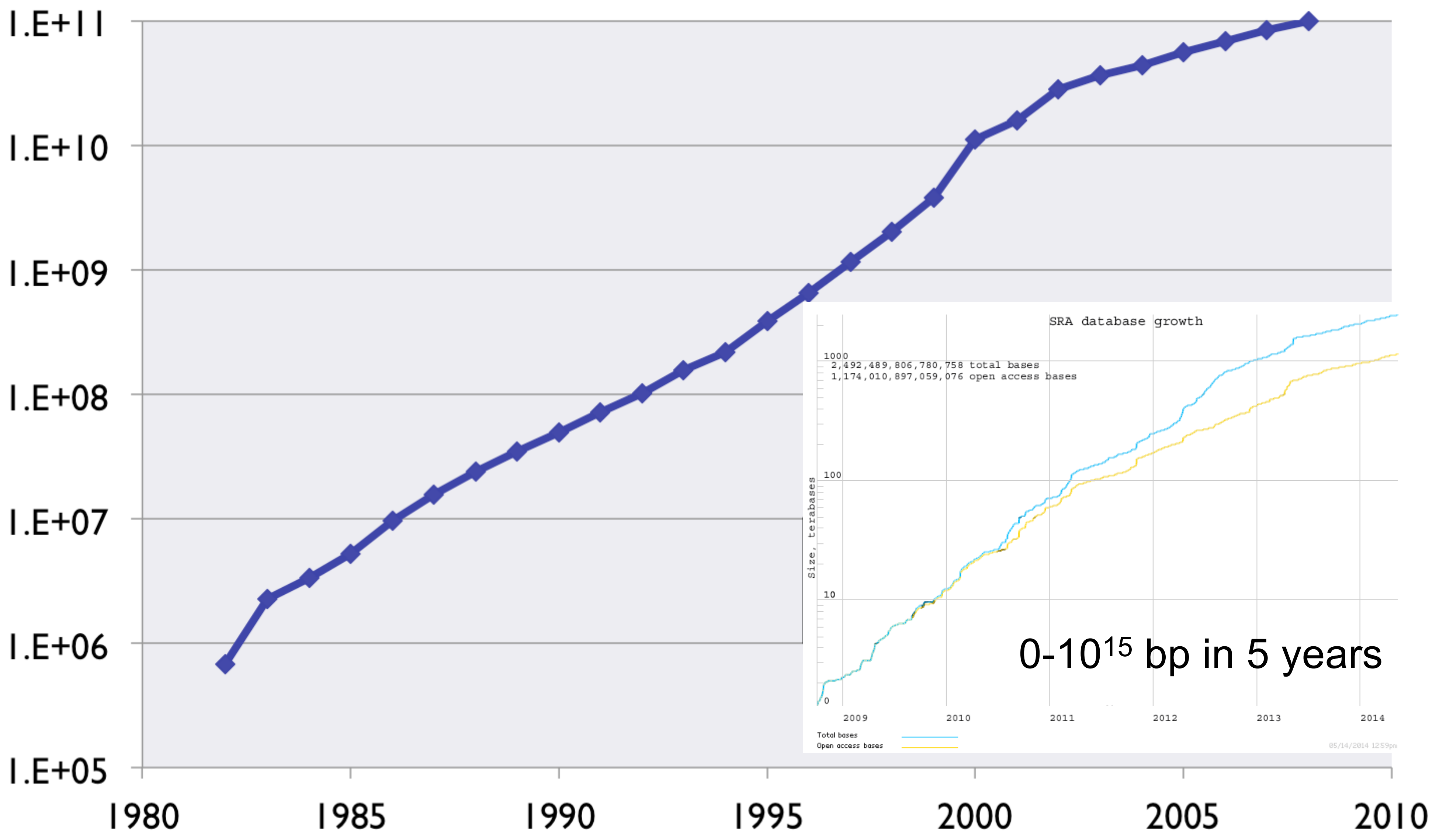
Application 1: Sequence Search

Moore’s Law



10,000,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
Transistors
1970
1975
1980
1985
1990
1995
2000
2005
2010
Year
4004
8008
8080
6800
8086
68000
80286
68020
80386
68030
80486
68040
Pentium
601
Pentium Pro
Pentium MMX
603e
604
604e
G3
Pentium III
Pentium III
G4
Pentium 4
G5
Pentium M
Pentium M
Xeon
Itanium 2
Itanium 2
Xeon
Itanium 2
G6
Dual Core Itanium 2
Nehalen
Penryn
Quad Core
Core Duo
Pentium 4HT
Dual

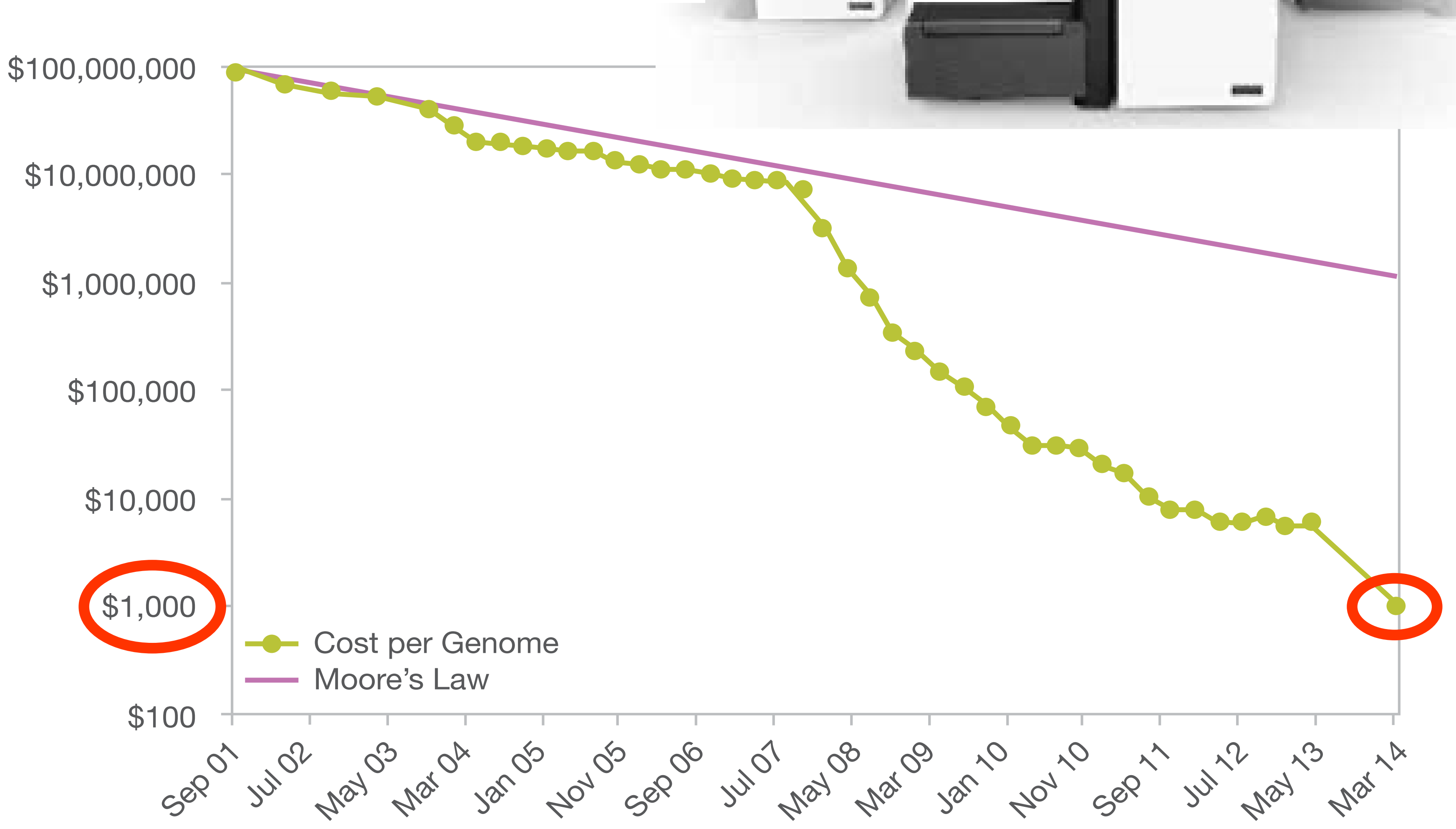
Growth of GenBank (Base Pairs)



0-10^{15} bp in 5 years

Source: http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html

Sequencing Costs Outpace Moore's Law

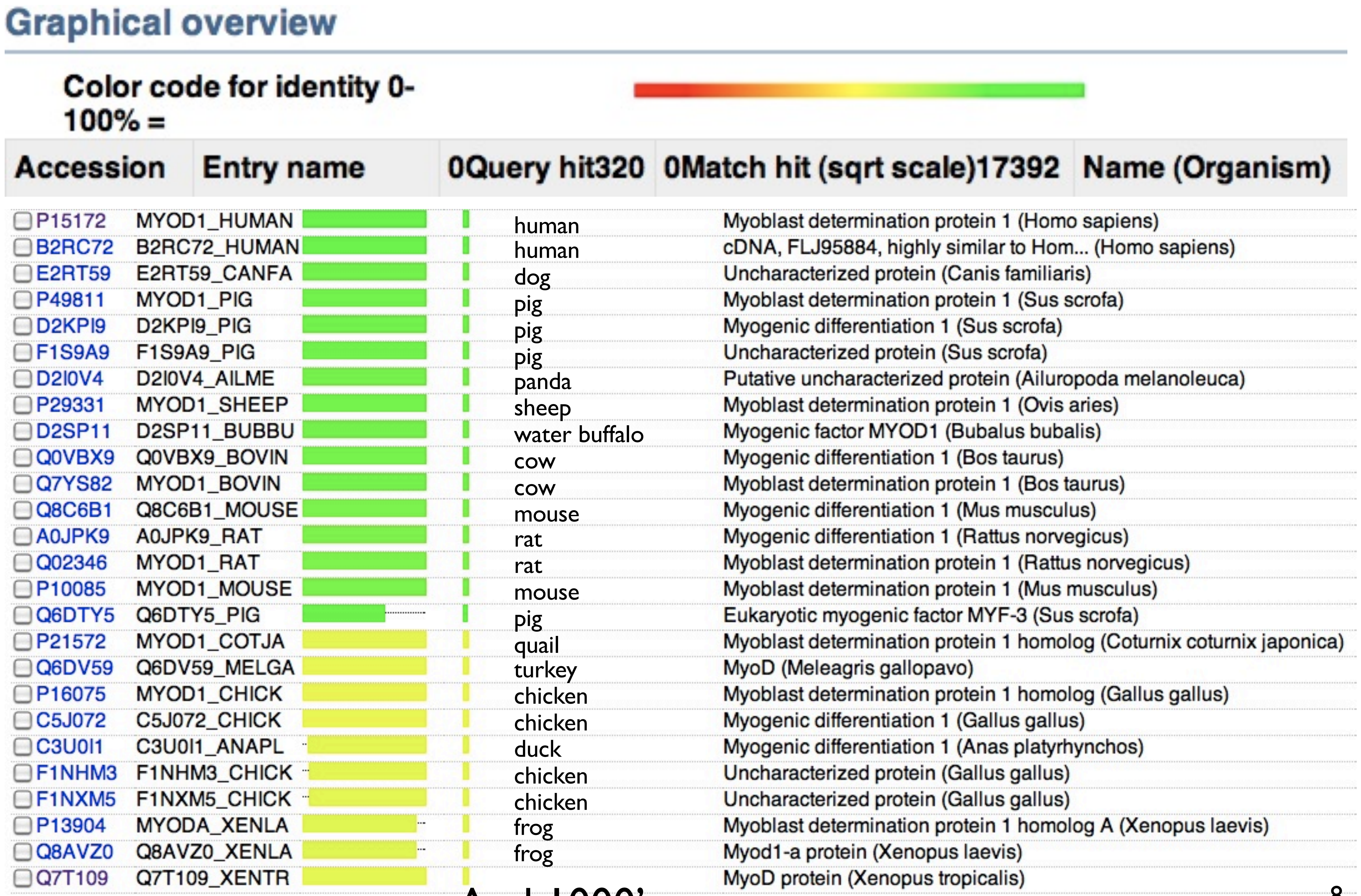


A Database Search

go to, e.g., http://www.uniprot.org/, “blast” tab, and paste in this:

```
>sp|P15172|MYOD1_HUMAN Myoblast determination protein 1 OS=Homo
   sapiens GN=MYOD1 PE=1 SV=3
MELLSPPLRDVDLTAPDGSLCSFATTDDFYDDPCFDSPDLRFFEDLDPRLMHVGALLKPE
EHSHFPAAVHPAPGAREDEHVRAPSGHHQAGRCLLWACKACKRKTTNADRRKAATMRERR
RLSKVNEAFETLKRCTSSNPNQRLPKVEILRNAIRYIEGLQALLRDQDAAPPGAAAAFYA
PGPLPPGRGGEHYSGDSDASSPRSNCSDGMMDYSGPPSGARRRNCYEGAYYNEAPSEPRP
GKSAAVSSLDCLSSIVERISTESPAAPALLLADVPSESPPRRQEAAAPSEGESSGDPTQS
PDAAPQCPAGANPNPIYQVL
```

A Few seconds Later…

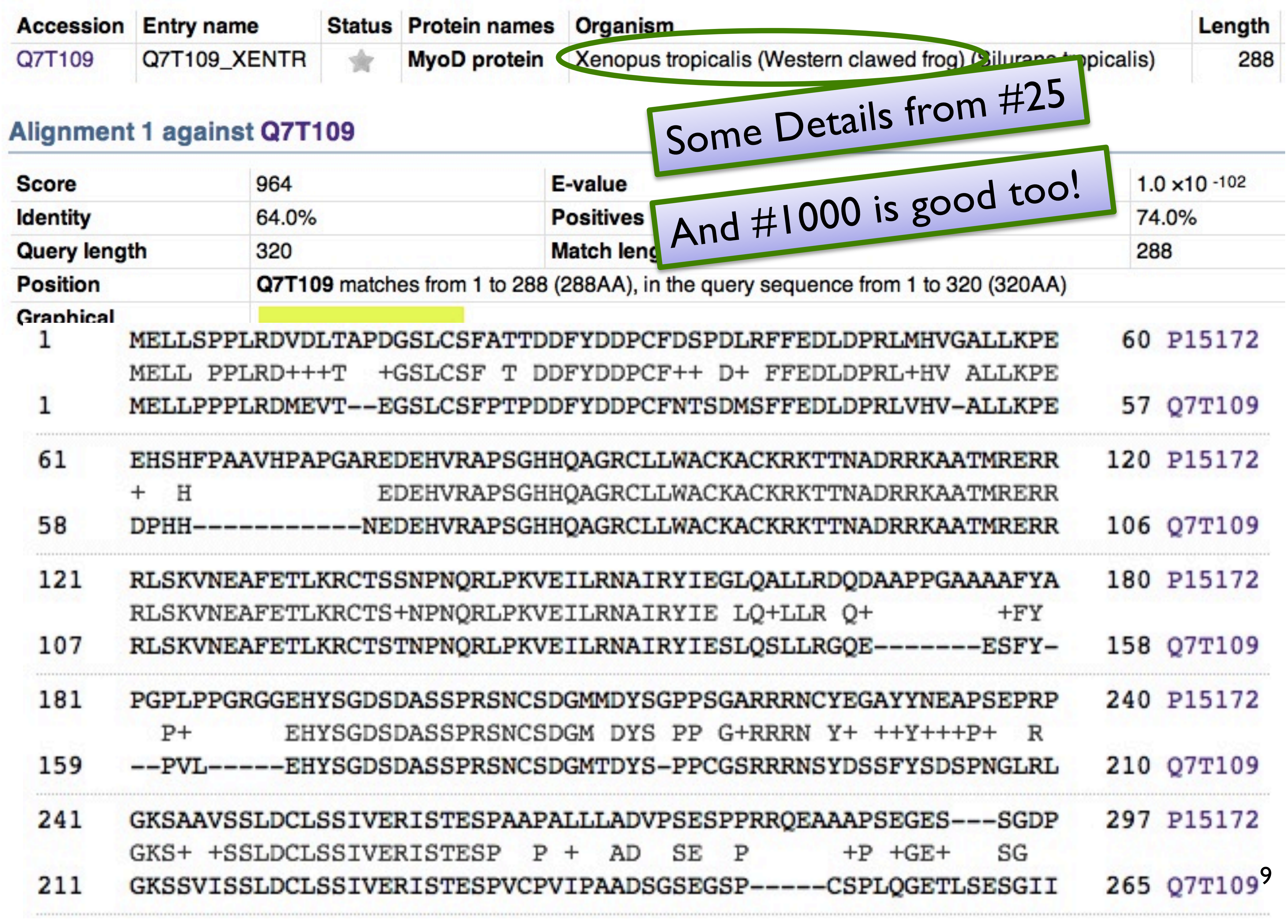


Graphical overview

Color code for identity 0-100% = 🟥🟨🟩

Accession	Entry name	0Query hit320	0Match hit (sqrt scale)17392		Name (Organism)
P15172	MYOD1_HUMAN	🟩	🟩	human	Myoblast determination protein 1 (Homo sapiens)
B2RC72	B2RC72_HUMAN	🟩	🟩	human	cDNA, FLJ95884, highly similar to Hom... (Homo sapiens)
E2RT59	E2RT59_CANFA	🟩	🟩	dog	Uncharacterized protein (Canis familiaris)
P49811	MYOD1_PIG	🟩	🟩	pig	Myoblast determination protein 1 (Sus scrofa)
D2KPI9	D2KPI9_PIG	🟩	🟩	pig	Myogenic differentiation 1 (Sus scrofa)
F1S9A9	F1S9A9_PIG	🟩	🟩	pig	Uncharacterized protein (Sus scrofa)
D2I0V4	D2I0V4_AILME	🟩	🟩	panda	Putative uncharacterized protein (Ailuropoda melanoleuca)
P29331	MYOD1_SHEEP	🟩	🟩	sheep	Myoblast determination protein 1 (Ovis aries)
D2SP11	D2SP11_BUBBU	🟩	🟩	water buffalo	Myogenic factor MYOD1 (Bubalus bubalis)
Q0VBX9	Q0VBX9_BOVIN	🟩	🟩	cow	Myogenic differentiation 1 (Bos taurus)
Q7YS82	MYOD1_BOVIN	🟩	🟩	cow	Myoblast determination protein 1 (Bos taurus)
Q8C6B1	Q8C6B1_MOUSE	🟩	🟩	mouse	Myogenic differentiation 1 (Mus musculus)
A0JPK9	A0JPK9_RAT	🟩	🟩	rat	Myogenic differentiation 1 (Rattus norvegicus)
Q02346	MYOD1_RAT	🟩	🟩	rat	Myoblast determination protein 1 (Rattus norvegicus)
P10085	MYOD1_MOUSE	🟩	🟩	mouse	Myoblast determination protein 1 (Mus musculus)
Q6DTY5	Q6DTY5_PIG	🟩	🟩	pig	Eukaryotic myogenic factor MYF-3 (Sus scrofa)
P21572	MYOD1_COTJA	🟨	🟨	quail	Myoblast determination protein 1 homolog (Coturnix coturnix japonica)
Q6DV59	Q6DV59_MELGA	🟨	🟨	turkey	MyoD (Meleagris gallopavo)
P16075	MYOD1_CHICK	🟨	🟨	chicken	Myoblast determination protein 1 homolog (Gallus gallus)
C5J072	C5J072_CHICK	🟨	🟨	chicken	Myogenic differentiation 1 (Gallus gallus)
C3U0I1	C3U0I1_ANAPL	🟨	🟨	duck	Myogenic differentiation 1 (Anas platyrhynchos)
F1NHM3	F1NHM3_CHICK	🟨	🟨	chicken	Uncharacterized protein (Gallus gallus)
F1NXM5	F1NXM5_CHICK	🟨	🟨	chicken	Uncharacterized protein (Gallus gallus)
P13904	MYODA_XENLA	🟨	🟨	frog	Myoblast determination protein 1 homolog A (Xenopus laevis)
Q8AVZ0	Q8AVZ0_XENLA	🟨	🟨	frog	Myod1-a protein (Xenopus laevis)
Q7T109	Q7T109_XENTR	🟨	🟨		MyoD protein (Xenopus tropicalis)

…And 1000's more…



Accession	Entry name	Status	Protein names	Organism	Length
Q7T109	Q7T109_XENTR	★	**MyoD protein**	Xenopus tropicalis (Western clawed frog) (Silurana tropicalis)	288

Some Details from #25

And #1000 is good too!

Alignment 1 against Q7T109

Score	964	**E-value**	1.0×10^{-102}
Identity	64.0%	**Positives**	74.0%
Query length	320	**Match leng**	288
Position	**Q7T109** matches from 1 to 288 (288AA), in the query sequence from 1 to 320 (320AA)		
Graphical			

```
1      MELLSPPLRDVDLTAPDGSLCSFATTDDFYDDPCFDSPDLRFFEDLDPRLMHVGALLKPE    60 P15172
       MELL PPLRD+++T   +GSLCSF T DDFYDDPCF++ D+ FFEDLDPRL+HV ALLKPE
1      MELLPPPLRDMEVT--EGSLCSFPTPDDFYDDPCFNTSDMSFFEDLDPRLVHV-ALLKPE    57 Q7T109

61     EHSHFPAAVHPAPGAREDEHVRAPSGHHQAGRCLLWACKACKRKTTNADRRKAATMRERR   120 P15172
       +  H            EDEHVRAPSGHHQAGRCLLWACKACKRKTTNADRRKAATMRERR
58     DPHH-----------NEDEHVRAPSGHHQAGRCLLWACKACKRKTTNADRRKAATMRERR   106 Q7T109

121    RLSKVNEAFETLKRCTSSNPNQRLPKVEILRNAIRYIEGLQALLRDQDAAPPGAAAAFYA   180 P15172
       RLSKVNEAFETLKRCTS+NPNQRLPKVEILRNAIRYIE LQ+LLR Q+         +FY
107    RLSKVNEAFETLKRCTSTNPNQRLPKVEILRNAIRYIESLQSLLRGQE-------ESFY-   158 Q7T109

181    PGPLPPGRGGEHYSGDSDASSPRSNCSDGMMDYSGPPSGARRRNCYEGAYYNEAPSEPRP   240 P15172
         P+       EHYSGDSDASSPRSNCSDGM DYS PP G+RRRN Y+ ++Y+++P+   R
159    --PVL-----EHYSGDSDASSPRSNCSDGMTDYS-PPCGSRRRNSYDSSFYSDSPNGLRL   210 Q7T109

241    GKSAAVSSLDCLSSIVERISTESPAAPALLLADVPSESPPRRQEAAAPSEGES---SGDP   297 P15172
       GKS+ +SSLDCLSSIVERISTESP  P +  AD  SE  P        +P +GE+   SG
211    GKSSVISSLDCLSSIVERISTESPVCPVIPAADSGSEGSP-----CSPLQGETLSESGII   265 Q7T109
```

The foregoing search capability is a *huge* deal

the "google" of molecular biology

millions of searches daily

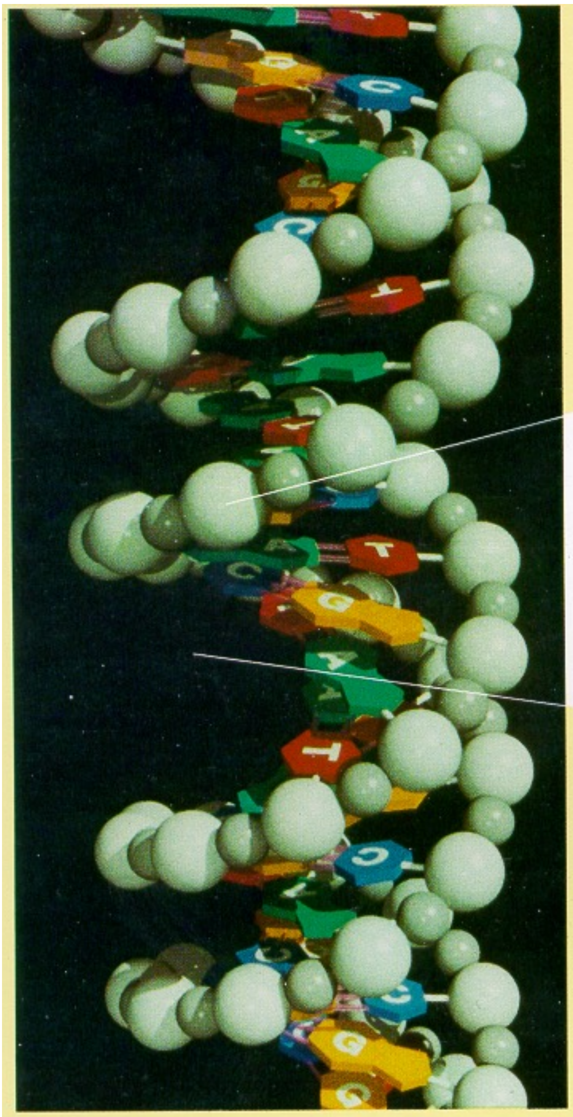
biologists (not just "computational" biologists) use this routinely

it connects information about *all* living things

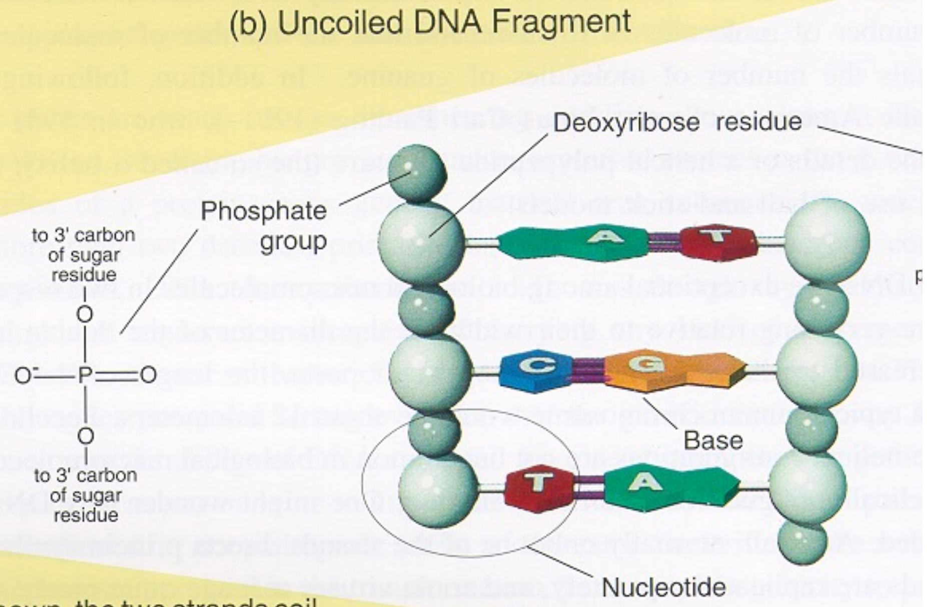
Time permitting, $\overbrace{\text{more on algorithm}}^{\text{(dynamic programming)}}$ later …

Application 2: RNA Structure

The Double Helix



(a) Computer-generated Image of DNA (by Mel Prueitt)



(b) Uncoiled DNA Fragment
Deoxyribose residue
Phosphate group
to 3' carbon of sugar residue
O
O^-—P═O
O
to 3' carbon of sugar residue
A
T
C
G
Base
T
A
Nucleotide

As shown, the two strands coil about each other in a fashion such that all the bases project inward toward the helix axis. The two strands are held together by hydrogen bonds (pink rods) linking each base projecting from one backbone to its so-called complementary base projecting from the other backbone. The base A always bonds to T (A and T are comple-

Shown in (b) is an uncoiled fragment of (a three complementary base pai chemist's viewpoint, each stra a polymer made up of four re called deoxyribonucleotides

Los Alamos Science

http://www.rcsb.org/pdb/explore.do?structureId=1GAT

NATURE VOL. 227 AUGUST 8 1970

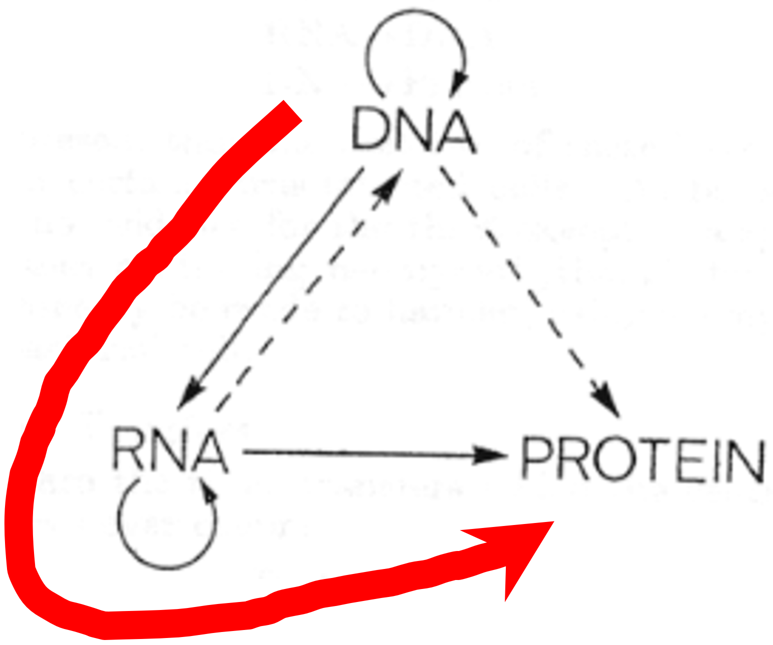
Central Dogma of Molecular Biology

by
FRANCIS CRICK
MRC Laboratory
Hills Road,
Cambridge CB2 2QH

The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred from protein to either protein or nucleic acid.

"The central dogma, enunciated by Crick in 1958 and the keystone of molecular biology ever since, is likely to prove a considerable over-simplification."

Fig. 2. The arrows show the situation as it seemed in 1958. Solid arrows represent probable transfers, dotted arrows possible transfers. The absent arrows (compare Fig. 1) represent the impossible transfers postulated by the central dogma. They are the three possible arrows starting from 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., tRNA)

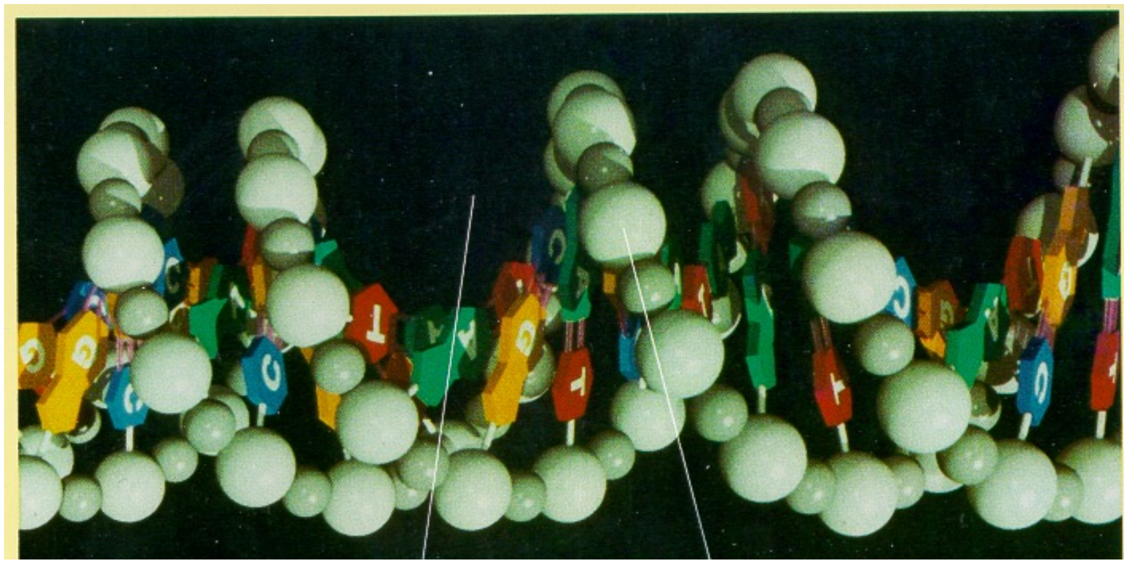
Since mid 90's dramatic discoveries

- Regulation, transport, stability/degradation
- E.g. "miRNA": >1000 in humans; regulate >50% of genes
- E.g. "riboswitches": 10000's in bacteria

By some estimates, ncRNA >> mRNA

DNA structure: dull

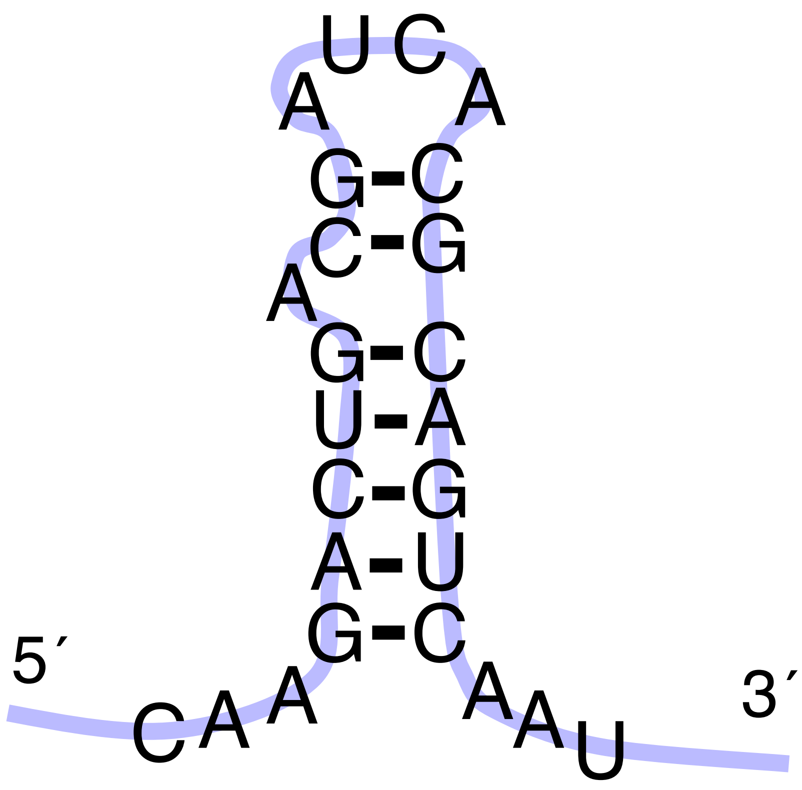
```
5’…ACCGCTAGATG…3’
    | | | | | | | | | | |
3’…TGGCGATCTAC…5’
```



RNA Secondary Structure:
RNA makes helices too

Base pairs

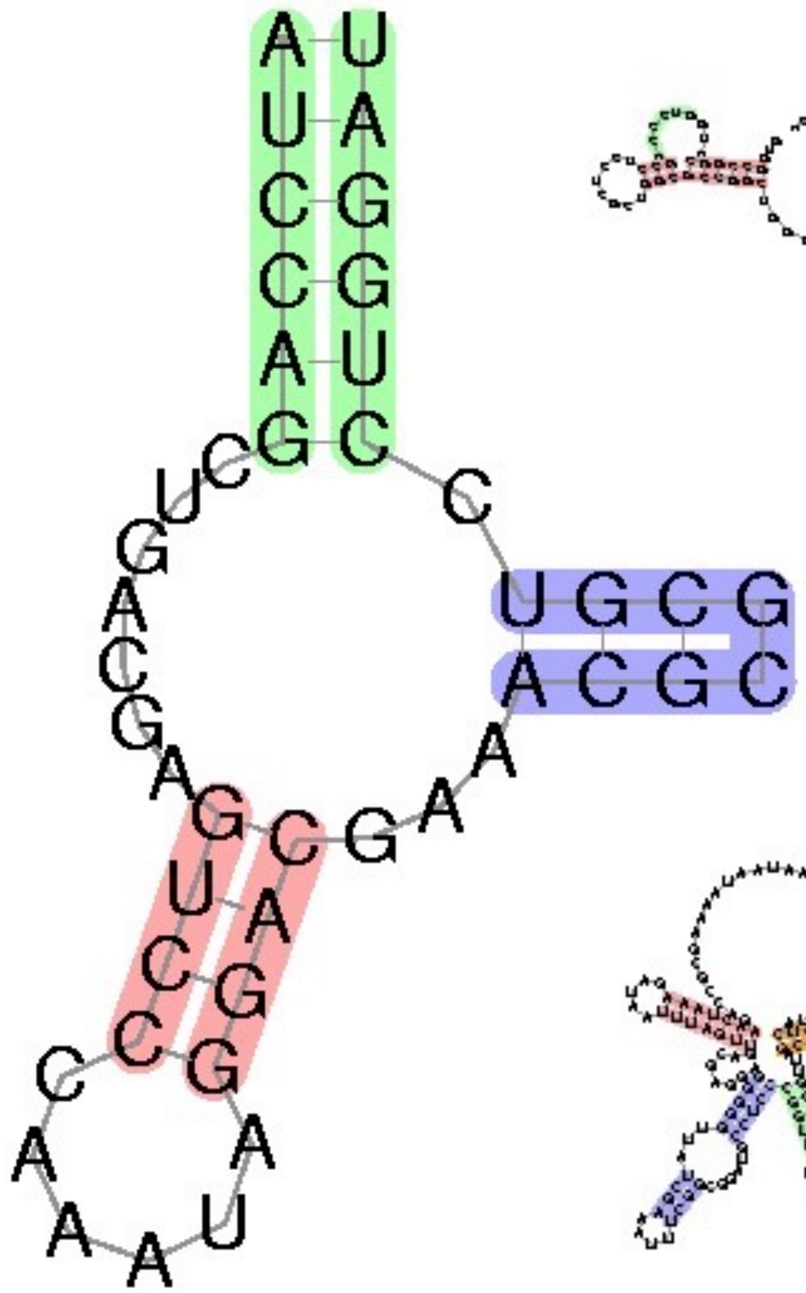
A–U
C–G

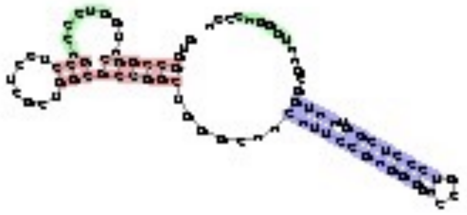


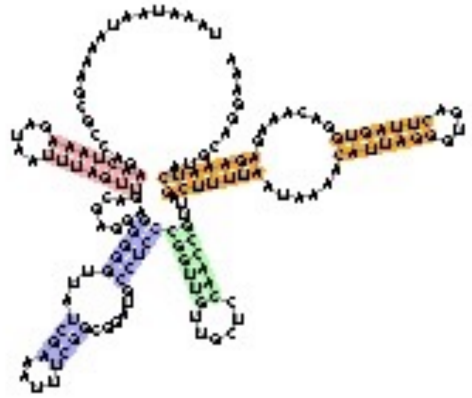
Usually *single* stranded

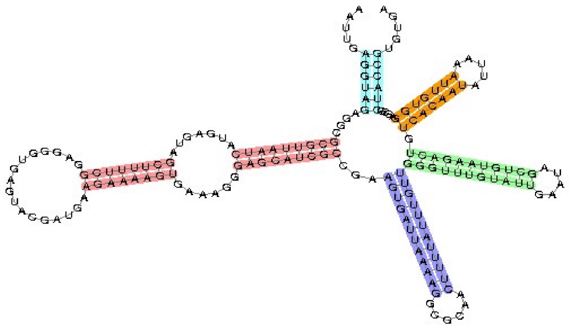
RNA Secondary Structure:

Not everything, but important, easier than 3d





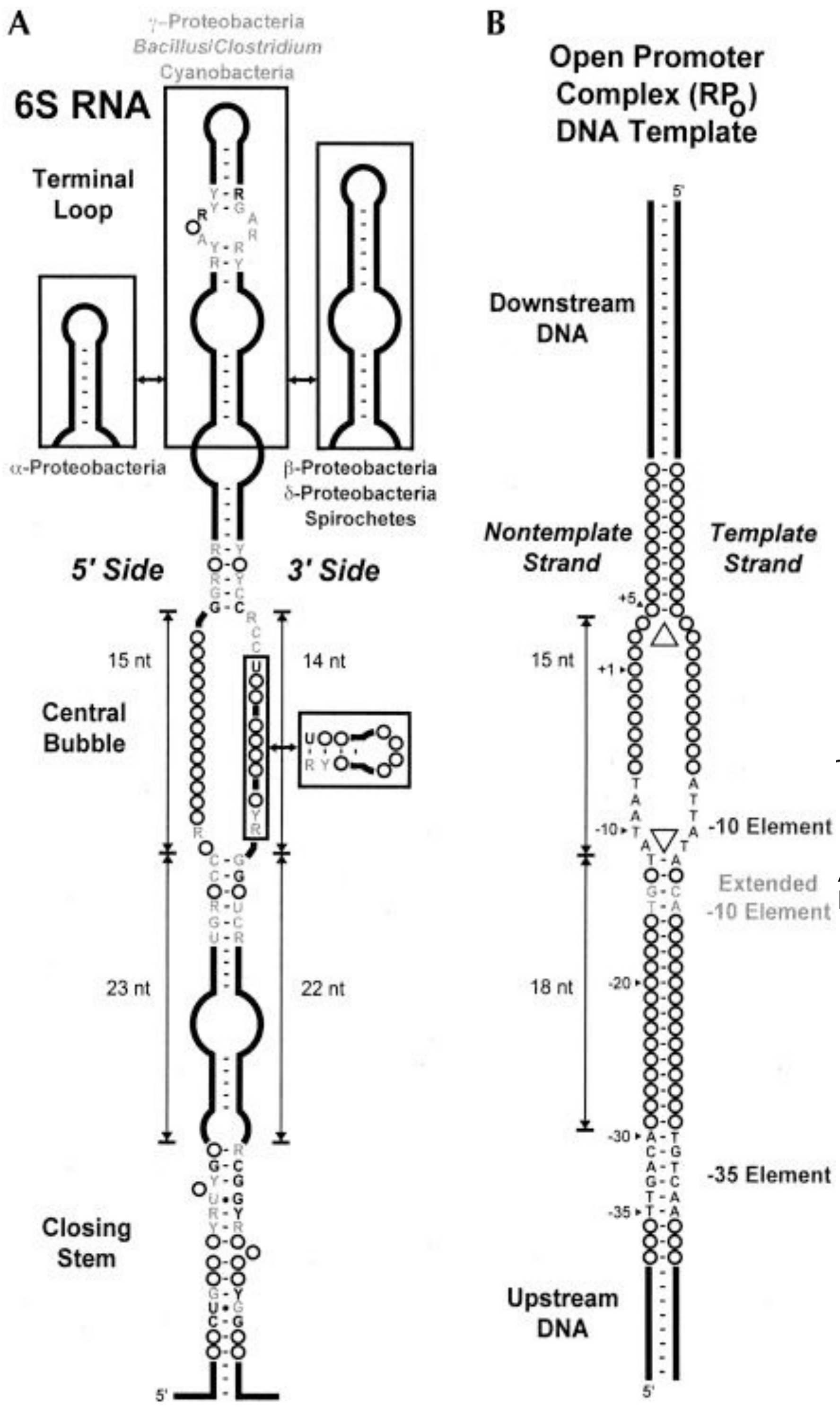


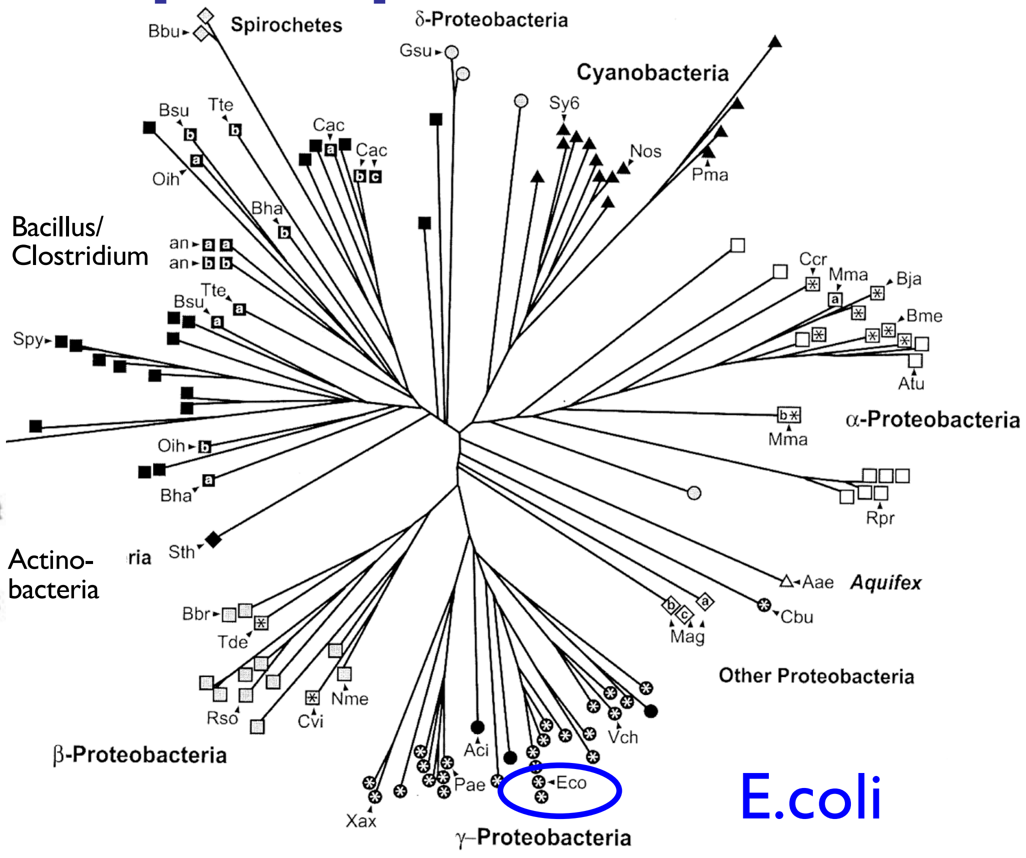


Why is structure important?

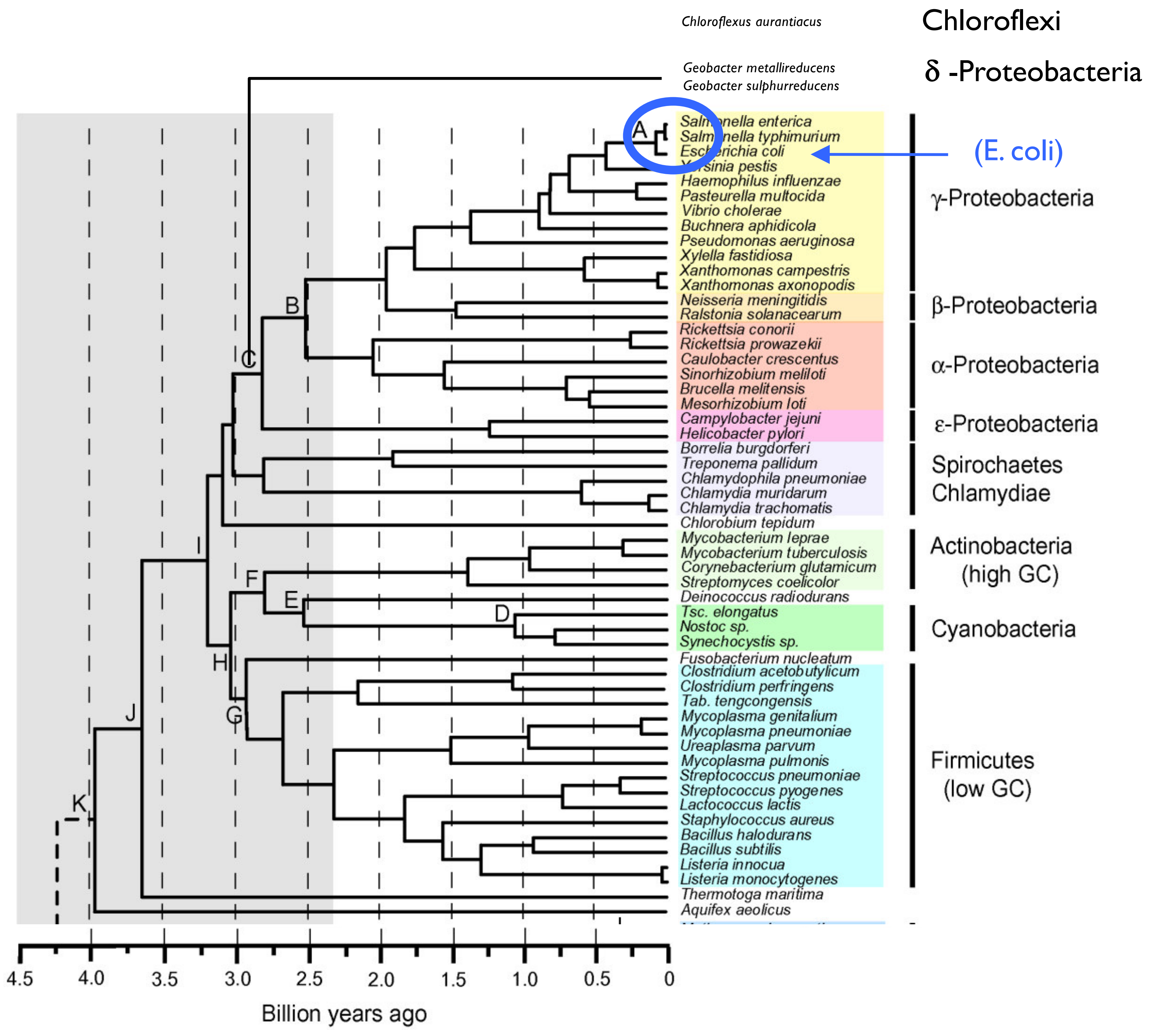
- For protein-coding, similarity in sequence is a powerful tool for finding related sequences
  - e.g. "hemoglobin," "MyoD" and many others are easily recognized in all animals
- For many non-coding RNAs, *different sequences* can have the *same structure*, and structure is most important for function.
  - So, using structure plus sequence, can find related sequences at much greater evolutionary distances
  - 2 Examples below

6S mimics an open promoter

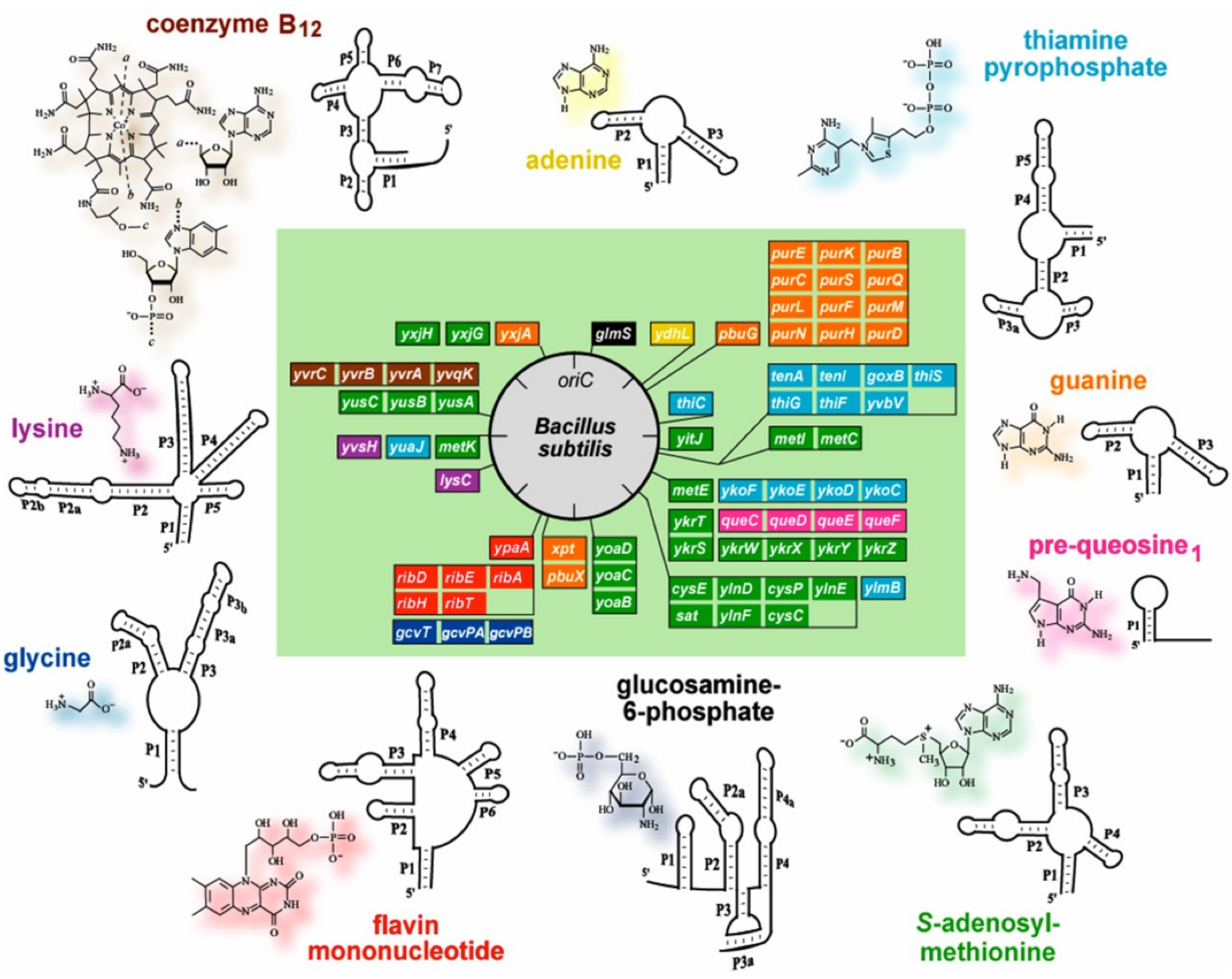




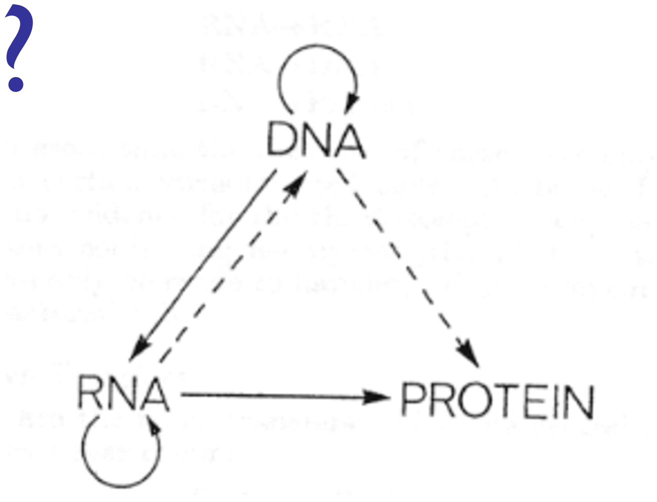
Barrick et al. *RNA* 2005
Trotochaud et al. *NSMB* 2005
Willkomm et al. NAR 2005


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
(E. coli)
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
Corynebacterium glutamicum
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

“Riboswitches”: small molecule sensors & gene on/off switches



Origin of Life?

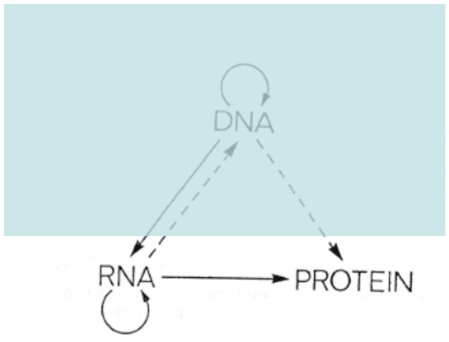


Life needs

- information carrier: DNA
- molecular machines, like enzymes: Protein
- making proteins needs DNA + RNA + proteins
- making (duplicating) DNA needs proteins

Horrible circularities! How could it have arisen in an abiotic environment?

Origin of Life?



RNA can carry information, too

- RNA double helix; RNA-directed RNA polymerase

RNA can form complex structures

RNA enzymes exist (ribozymes)

RNA can control, do logic (riboswitches)

The "RNA world" hypothesis:
1st life was RNA-based

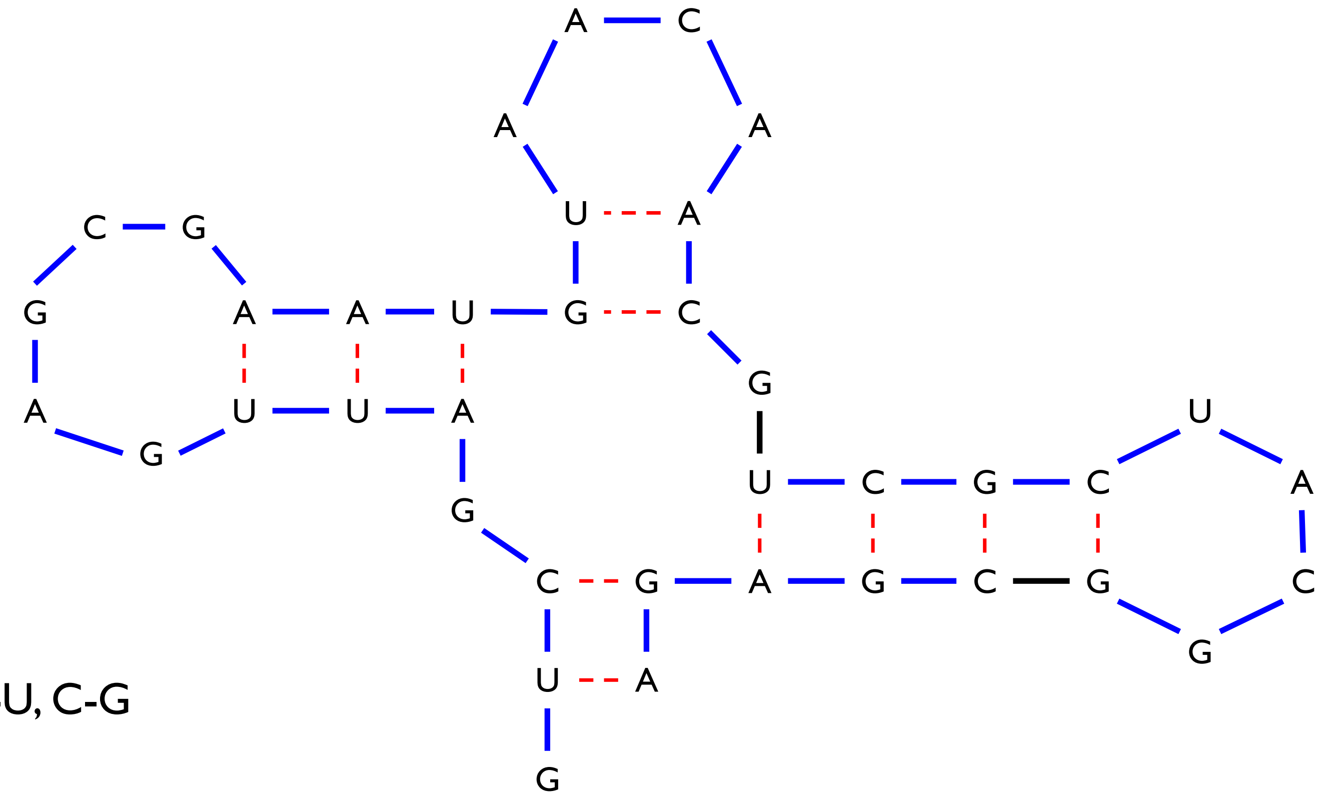
6.5 RNA Secondary Structure

Nussinov's Algorithm – core technology
for RNA structure prediction

RNA Secondary Structure

RNA. String B = $b_1 b_2 \ldots b_n$ over alphabet { A, C, G, U }.

Secondary structure. RNA is usually single-stranded, and tends to loop back and form base pairs with itself. This structure is essential for understanding molecular behavior.



— backbone

-- pairing

complementary base pairs: A-U, C-G

Ex: **GUCGAUUGAGCGAAUGUAACAACGUGGCUACGGCGAGA**

RNA Secondary Structure (≈ oversimplified)

T in DNA → U in RNA

RNA: String $B = b_1 b_2 \ldots b_n$ over alphabet { A, C, G, U }.

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

- [Watson-Crick Pairing.]
  - S is a *matching:* each base pairs with ≤ other, and
  - each pair in S is a Watson-Crick pair: A-U, U-A, C-G, or G-C.
- [No sharp turns.] Pairs are separated by ≥ 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 $i < k < j < l$. (Violation is called a *pseudoknot.*)

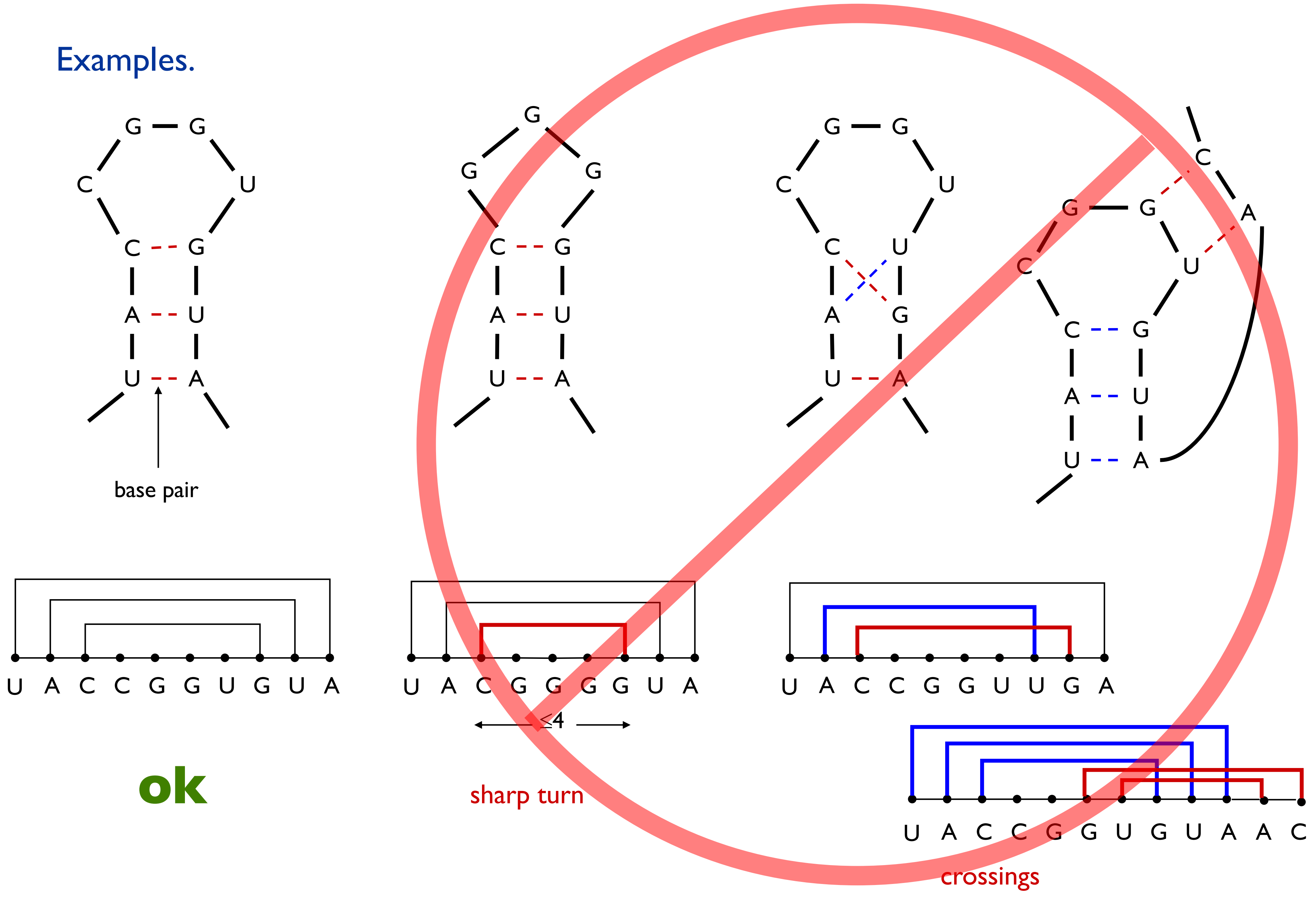
What's Best: RNA will form the structure that *minimizes* free energy.

approximated by maximizing number of base pairs

Goal: find a secondary structure S *maximizing* the number of base pairs.

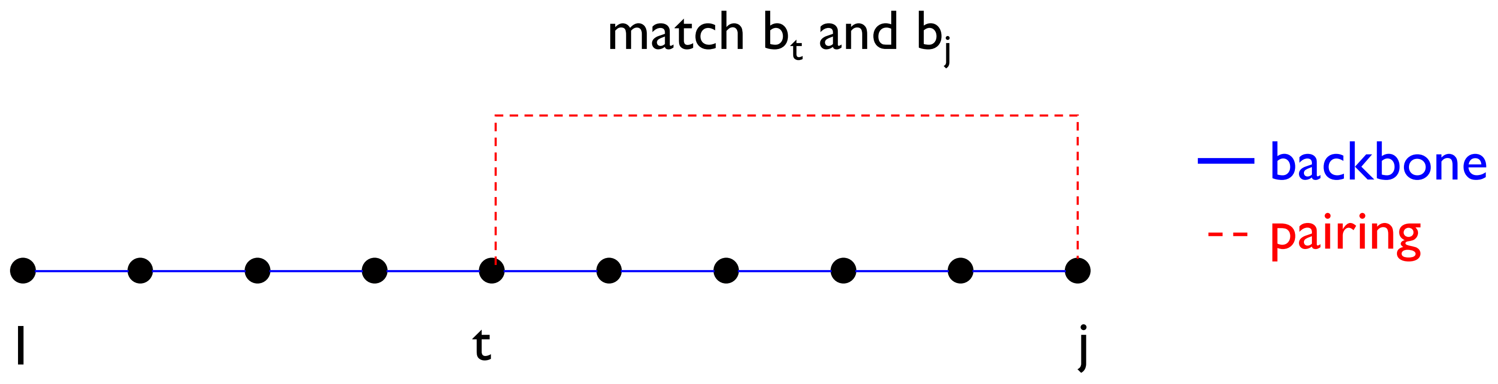
RNA Secondary Structure: Examples

Examples.



RNA Secondary Structure: Subproblems

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



Results in two sub-problems.

- Find secondary structure in: $b_1 b_2 \ldots b_{t-1}$. ← OPT(t-1) ; good!
- Find secondary structure in: $b_{t+1} b_{t+2} \ldots b_{j-1}$. ←

DIFFICULTY: this isn't "OPT" of anything; need more flexible set of sub-problems

Dynamic Programming Over Intervals: (R. Nussinov's algorithm)

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 1a. If $i \geq j - 4$ (and base b_j is not paired):

  OPT[i, j] = 0 by no-sharp turns condition.

- Case 1b. If $i < j - 4$, but base b_j is not paired:

  OPT[i, j] = OPT[i, j-1]

Key point: Either last base is unpaired (case 1a,b) or paired (case 2)

- Case 2. Base b_j pairs with b_t for some $i \leq t < j - 4$.
  non-crossing constraint decouples resulting sub-problems

$$\text{OPT}[i, j] = 1 + \max_t \{ \text{OPT}[i, t-1] + \text{OPT}[t+1, j-1] \}$$

take max over t such that $i \leq t < j-4$ and b_t and b_j are Watson-Crick complements

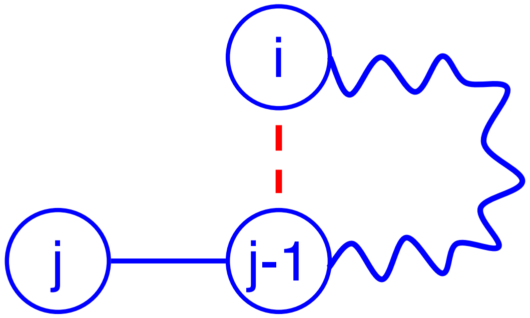
omit when t == i (see next slide)

"Optimal pairing of b_i ... b_j"

Two possibilities:

j Unpaired:

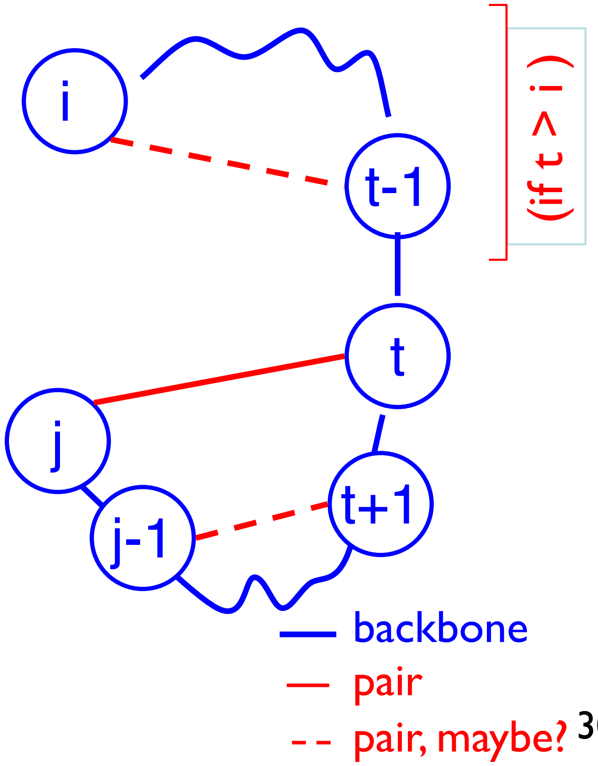
Find best pairing of b_i ... b_{j-1}



j Paired (with some t):

Find best b_i ... b_{t-1} +

best b_{t+1} ... b_{j-1} plus 1



Why is it slow?

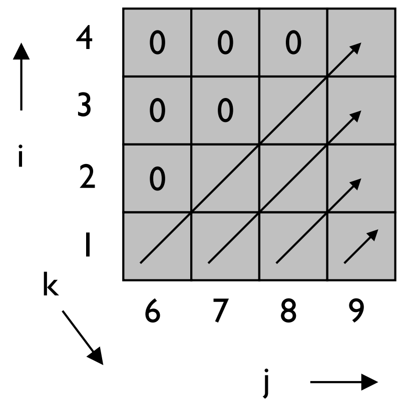
Why do pseudoknots matter?

Bottom Up Dynamic Programming Over Intervals

Q. What order to solve the sub-problems?
A1. Book way–do shortest intervals first, then earliest start:

Interval length →
Start position →
End position →

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

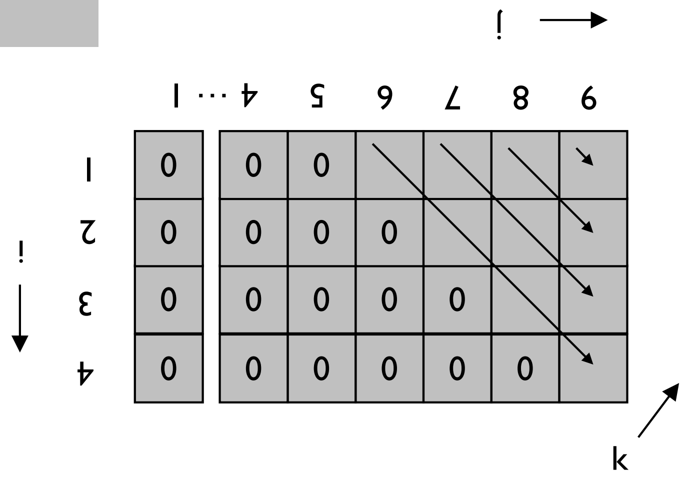


book

A2. Slides way: earliest start first, then shortest intervals (next slides)

+ HW

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



slides

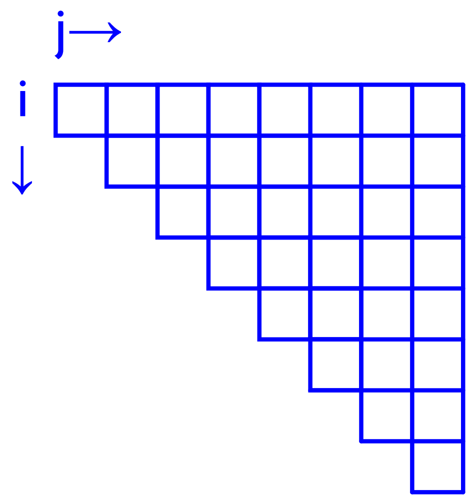
Nussinov: Max Pairing

Opt[i,j] = # pairs in optimal pairing of $b_i \ldots b_j$

Opt[i,j] = 0 for all i, j with $i \geq j-4$; otherwise

Opt[i,j] = max of:

$$\begin{cases} \text{Opt}[i,j-1] \\ \max \{ \text{Opt}[i,\overset{\text{if } t > i}{t}-1]+1+\text{Opt}[t+1,j-1] \mid \\ \qquad i \leq t < j-4 \text{ and } b_t\text{-}b_j \text{ may pair}\} \end{cases}$$



R Nussinov, AB Jacobson, "Fast algorithm for predicting the secondary structure of single-stranded RNA." PNAS 1980.

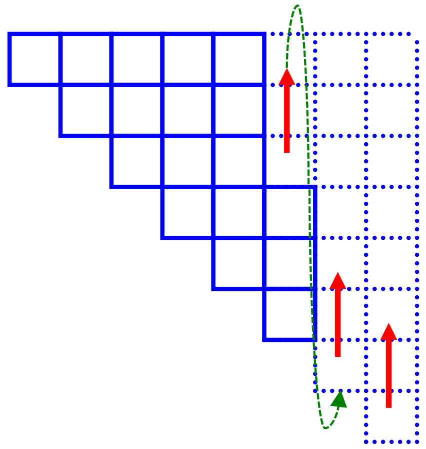
Another Computation Order

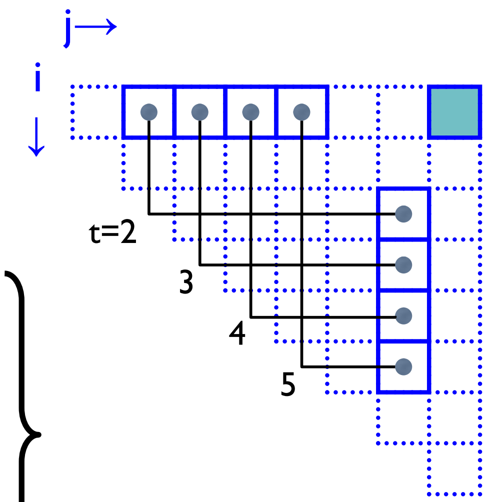
Opt[i, j] = optimal # pairs in $b_i \dots b_j$

```
for(j = 1 to n)
  for(i = j downto 1)
    Opt[i, j] = 0 if i ≥ j-4 else:
      max of:
        { Opt[i, j-1]
          max { Opt[i, t-1] + 1 + Opt[t+1, j-1] |
                i ≤ t < j-4 and b_t - b_j may pair } }
```

if t > i

Time: $O(n^3)$





Which Pairs?

Usual dynamic programming “trace-back” tells you *which* base pairs are in the optimal solution, not just how many

Details? : homework

Computing one cell: OPT[2,18] = ?

(Examples here and below assume 1-based indexing)

G	G	G	A	A	A	A	C	C	C	A	A	A	G	G	G	G	$\boxed{U}$	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	2	2	2	2	2	3	3	3	$\boxed{4}$	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

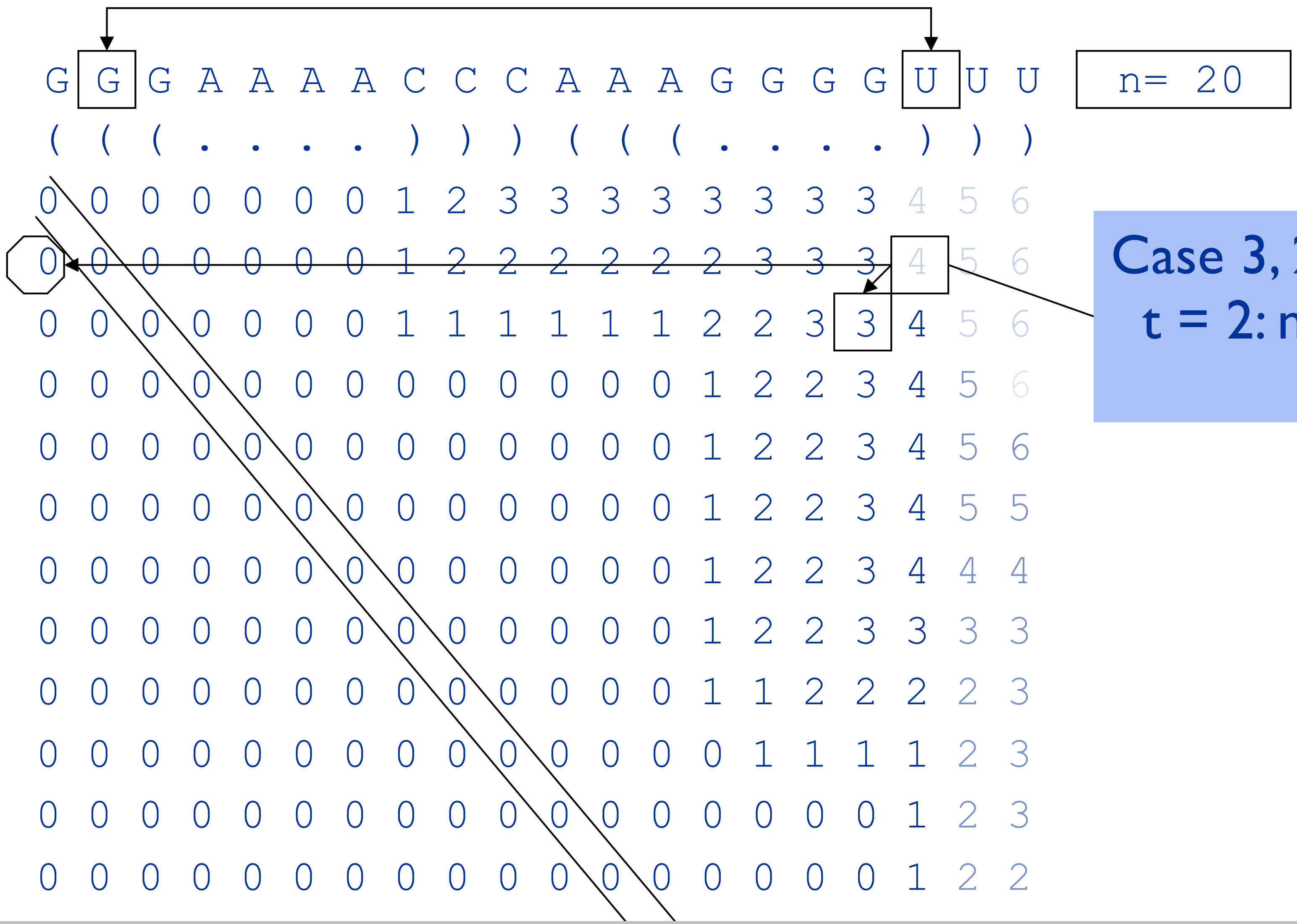
n= 20

Case 1:
 $2 \geq 18-4$? no.
Case 2:
 B_{18} unpaired?
 Always a possibility;
 then OPT[2,18] ≥ 3

```
GGAAAACCCAAAGGGGU
((....))(....)...
```

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & \text{if } i \geq j-4 \\ \max\begin{Bmatrix} \mathrm{OPT}[i,j-1] \\ 1+\max_t(\mathrm{OPT}[i,t-1]+\mathrm{OPT}[t+1,j-1]) \end{Bmatrix} & \text{otherwise} \end{cases}$$

Computing one cell: OPT[2,18] = ?



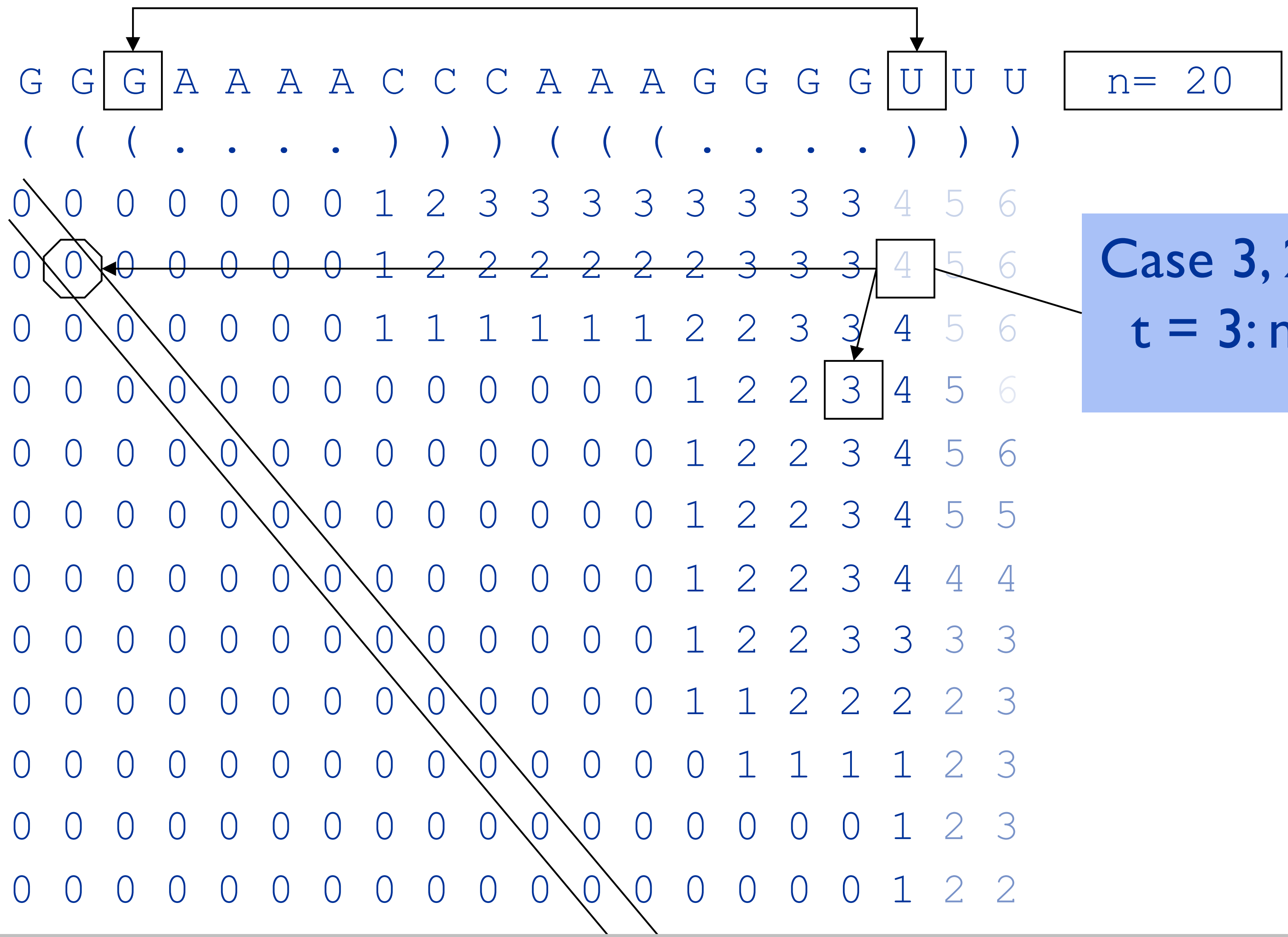
G	G	G	A	A	A	A	C	C	C	A	A	A	G	G	G	G	U	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	2	2	2	2	2	3	3	3	4	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

n= 20

Case 3, $2 \le t < 18-4$:
$t = 2$: no pair

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & \text{if } i \ge j-4 \\ \max\left\{ \begin{array}{c} \mathrm{OPT}[i,j-1] \\ 1 + \max_t(\mathrm{OPT}[i,t-1] + \mathrm{OPT}[t+1,j-1]) \end{array} \right\} & \text{otherwise} \end{cases}$$

Computing one cell: OPT[2,18] = ?

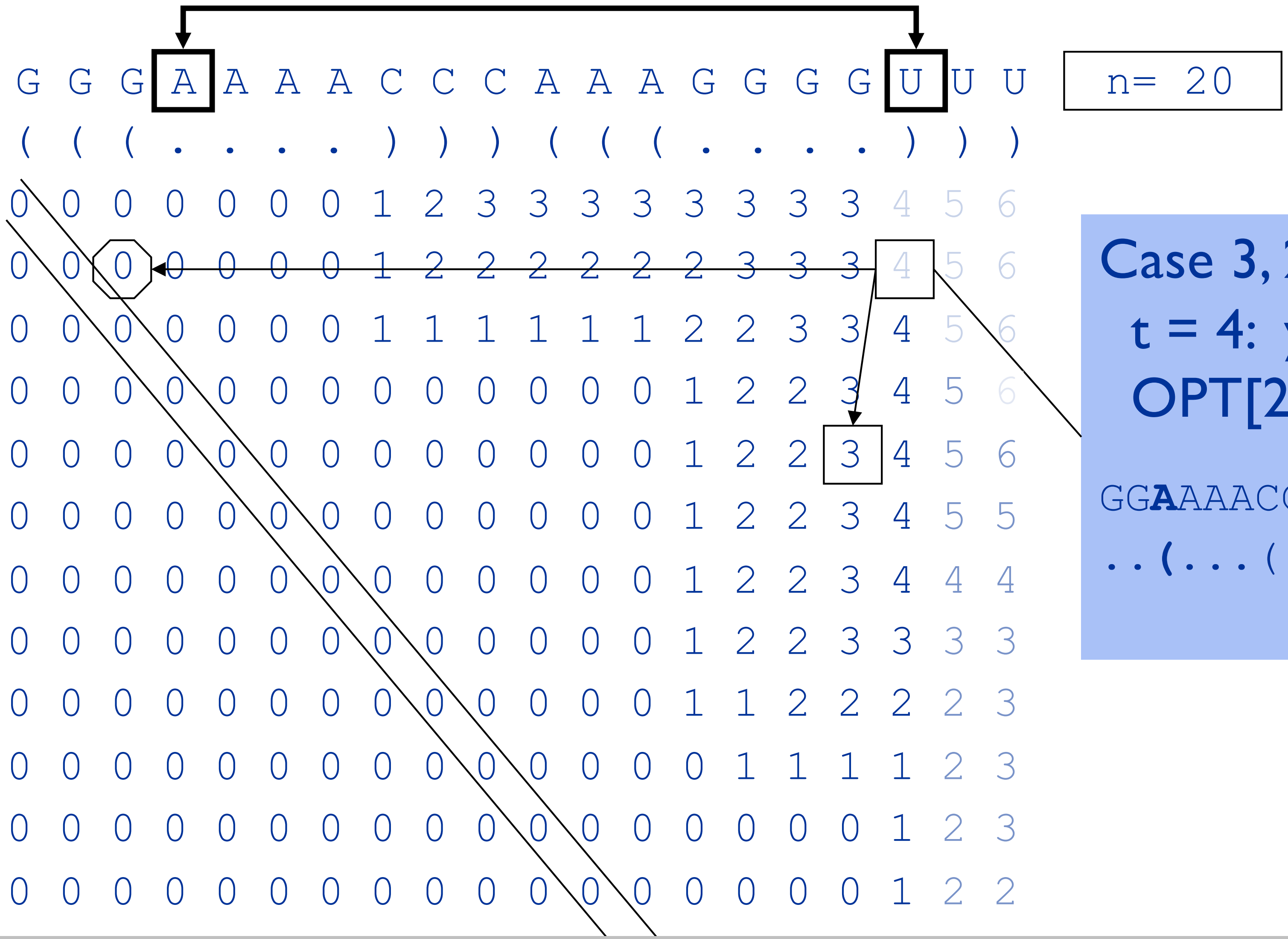


G	G	$\boxed{G}$	A	A	A	A	C	C	C	A	A	A	G	G	G	G	$\boxed{U}$	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	2	2	2	2	2	3	3	3	$\boxed{4}$	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	$\boxed{3}$	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

Case 3, 2 ≤ t <18-4:
t = 3: no pair

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & \text{if } i \ge j-4 \\ \max\left\{ \begin{array}{c} \mathrm{OPT}[i,j-1] \\ 1+\max_t(\mathrm{OPT}[i,t-1]+\mathrm{OPT}[t+1,j-1]) \end{array} \right\} & \text{otherwise} \end{cases}$$

Computing one cell: OPT[2,18] = ?



n= 20

G	G	G	A	A	A	A	C	C	C	A	A	A	G	G	G	G	U	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	2	2	2	2	2	3	3	3	4	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

Case 3, $2 \le t < 18-4$:

$t = 4$: yes pair

$OPT[2,18] \geq 1+0+3$

GG**A**AAACCCAAAGGGG**U**

. . **(** . . . . (((. . . .))) **)**

$$OPT(i,j) = \begin{cases} 0 & \text{if } i \geq j-4 \\ \max\begin{Bmatrix} OPT[i,j-1] \\ 1+\max_t(OPT[i,t-1]+OPT[t+1,j-1]) \end{Bmatrix} & \text{otherwise} \end{cases}$$

Computing one cell: OPT[2,18] = ?

n= 20

G	G	G	A	A	A	A	C	C	C	A	A	A	G	G	G	G	U	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	2	2	2	2	2	3	3	3	4	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

Case 3, 2 $\leq$ t <18-4:

t = 5: yes pair

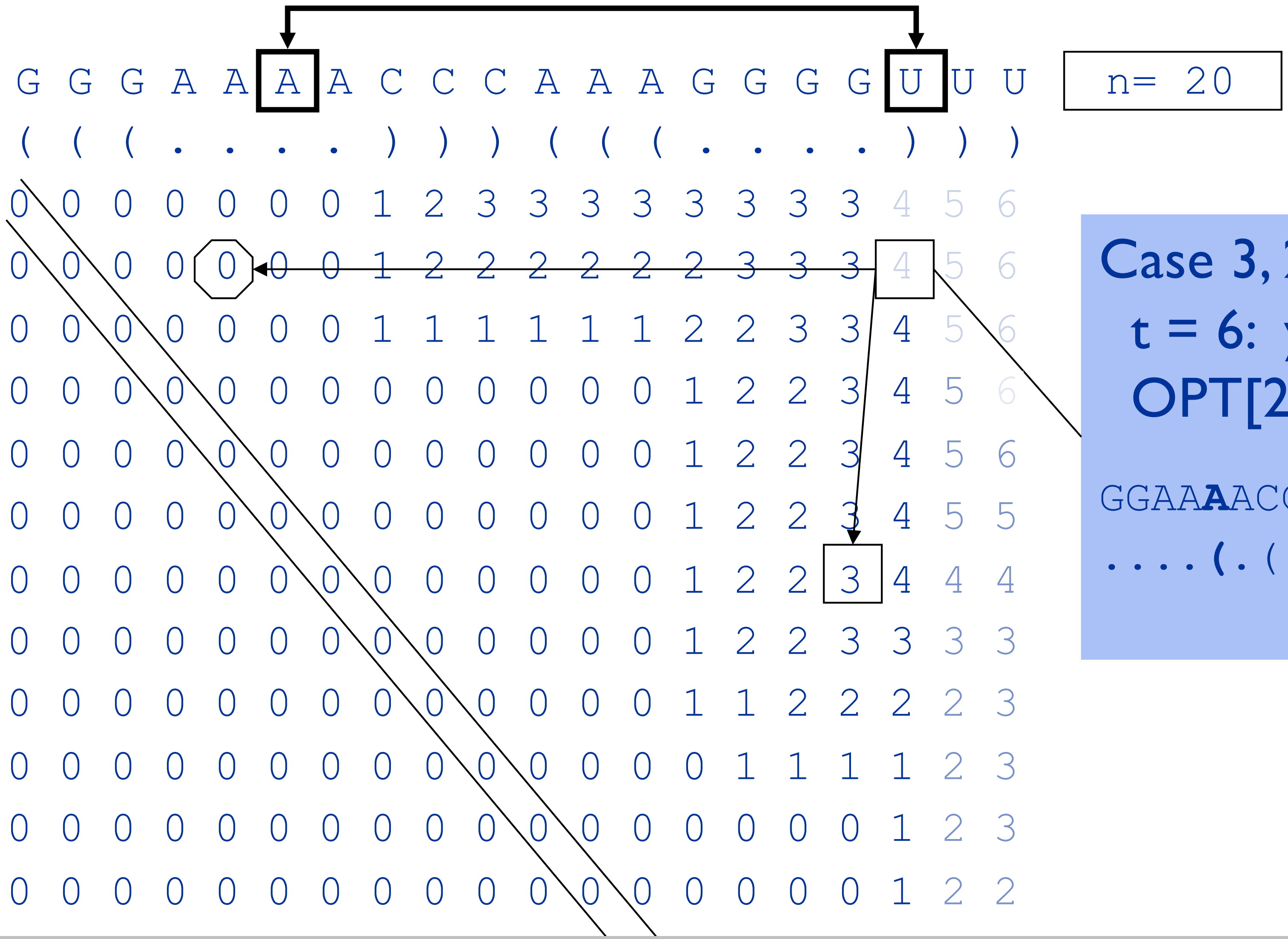
OPT[2,18]$\geq$1+0+3

GGA**A**AACCCAAAGGGG**U**

. . . **(** . . (((. . . .))) **)**

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & \text{if } i \geq j-4 \\ \max\begin{Bmatrix} \mathrm{OPT}[i,j-1] \\ 1+\max_t(\mathrm{OPT}[i,t-1]+\mathrm{OPT}[t+1,j-1]) \end{Bmatrix} & \text{otherwise} \end{cases}$$

Computing one cell: OPT[2,18] = ?



n= 20

G	G	G	A	A	A	A	C	C	C	A	A	A	G	G	G	G	U	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	2	2	2	2	2	3	3	3	4	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

Case 3, $2 \le t < 18-4$:

$t = 6$: yes pair

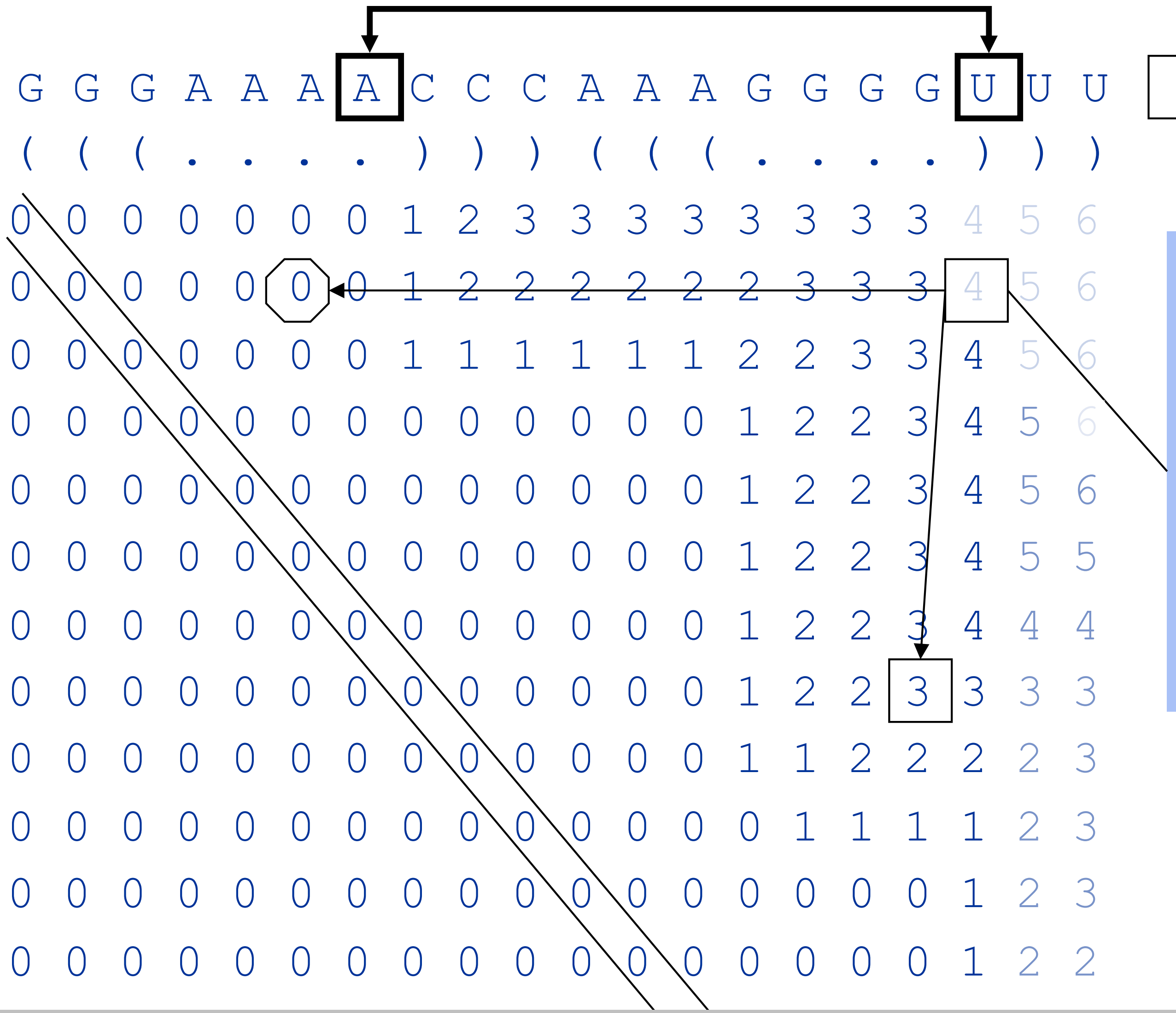
$\text{OPT}[2,18] \ge 1+0+3$

GGAA**A**ACCCAAAGGGG**U**

. . . . **(** . (((. . . .))) **)**

$$\text{OPT}(i,j) = \begin{cases} 0 & \text{if } i \ge j-4 \\ \max\begin{Bmatrix} \text{OPT}[i,j-1] \\ 1 + \max_t(\text{OPT}[i,t-1] + \text{OPT}[t+1,j-1]) \end{Bmatrix} & \text{otherwise} \end{cases}$$

Computing one cell: OPT[2,18] = ?



G	G	G	A	A	A	A	C	C	C	A	A	A	G	G	G	G	U	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	2	2	2	2	2	3	3	3	4	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

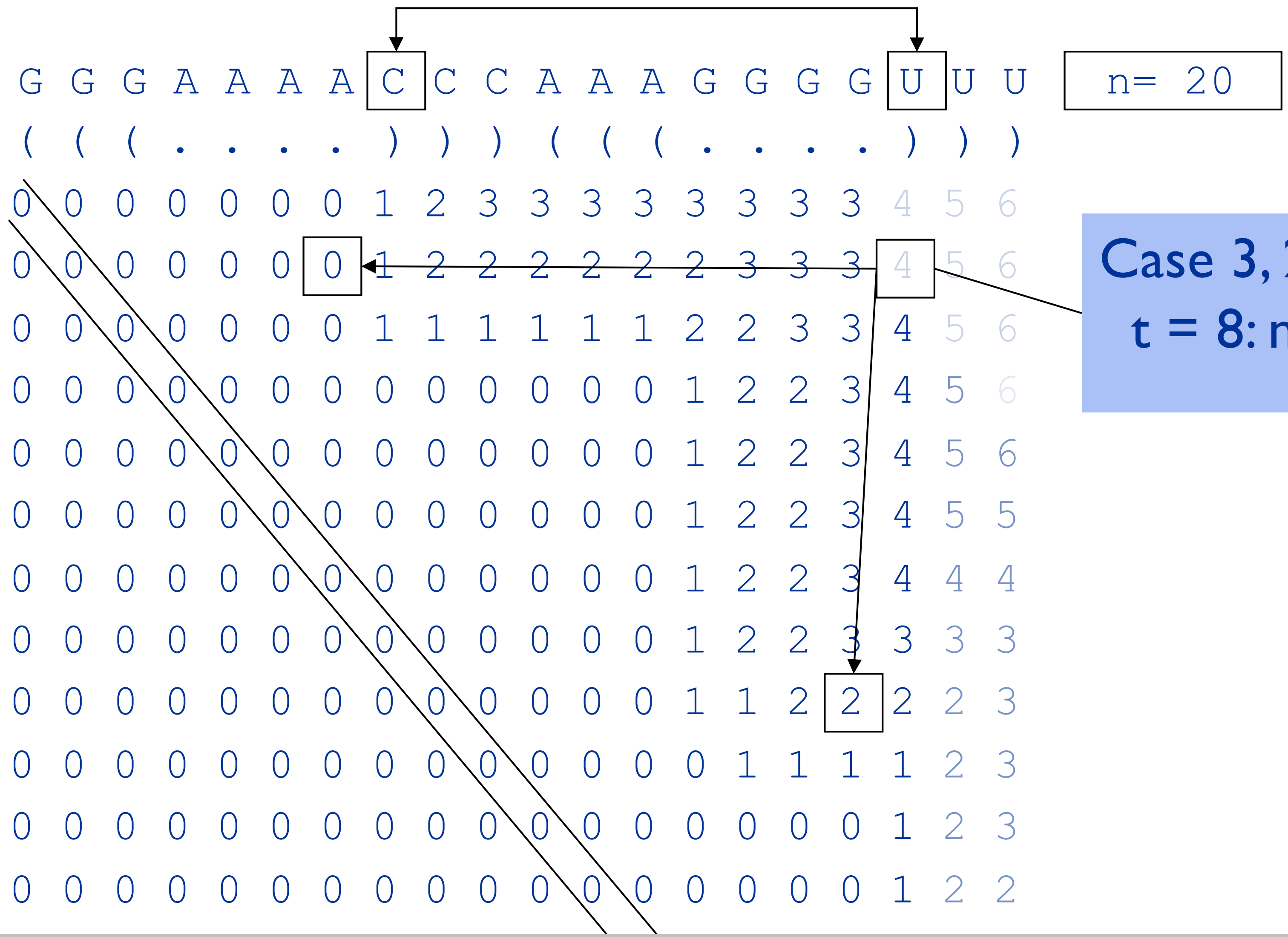
n= 20

Case 3, 2 ≤ t <18-4:
t = 7: yes pair
OPT[2,18]≥1+0+3

GGAAA**A**CCCAAAGGGG**U**
. . . . . **(** (((. . . .))) **)**

$$\mathrm{OPT}(i,j)=\begin{cases}0 & \text{if } i \geq j-4\\ \max\begin{Bmatrix}\mathrm{OPT}[i,j-1]\\ 1+\max_t(\mathrm{OPT}[i,t-1]+\mathrm{OPT}[t+1,j-1])\end{Bmatrix} & \text{otherwise}\end{cases}$$

Computing one cell: OPT[2,18] = ?



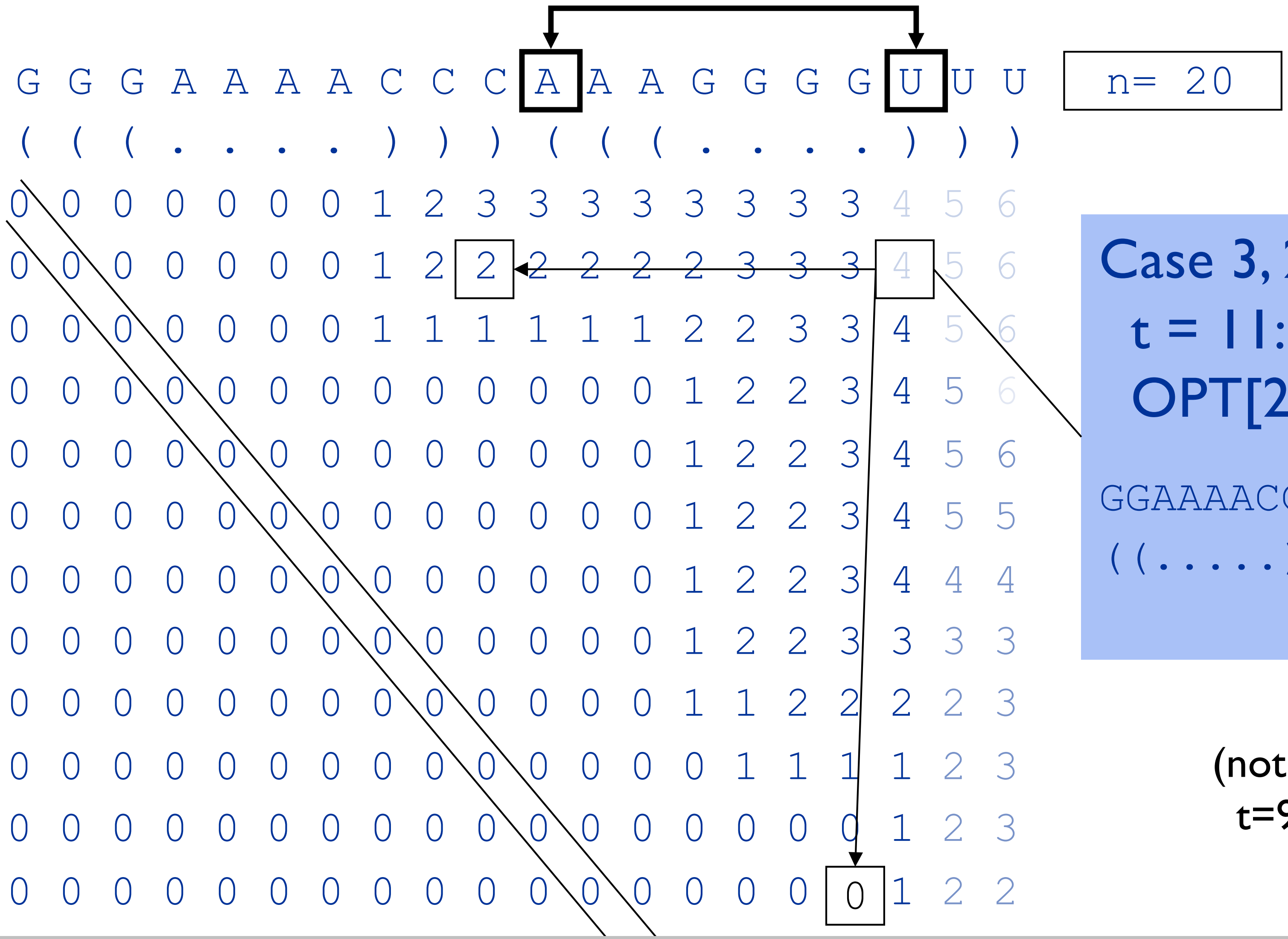
G	G	G	A	A	A	A	$\boxed{C}$	C	C	A	A	A	G	G	G	G	$\boxed{U}$	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	$\boxed{0}$	1	2	2	2	2	2	2	3	3	3	$\boxed{4}$	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	$\boxed{2}$	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

n= 20

Case 3, 2 ≤ t <18-4:
t = 8: no pair

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & \text{if } i \geq j-4 \\ \max\begin{Bmatrix} \mathrm{OPT}[i,j-1] \\ 1+\max_t(\mathrm{OPT}[i,t-1]+\mathrm{OPT}[t+1,j-1]) \end{Bmatrix} & \text{otherwise} \end{cases}$$

Computing one cell: OPT[2,18] = ?



n= 20

G	G	G	A	A	A	A	C	C	C	A	A	A	G	G	G	G	U	U	U
(	(	(	.	.	.	.	)	)	)	(	(	(	.	.	.	.	)	)	)
0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	$\boxed{2}$	2	2	2	2	3	3	3	$\boxed{4}$	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\boxed{0}$	1	2	2

Case 3, 2 ≤ t <18-4:
t = 11: yes pair
OPT[2,18]≥1+2+0

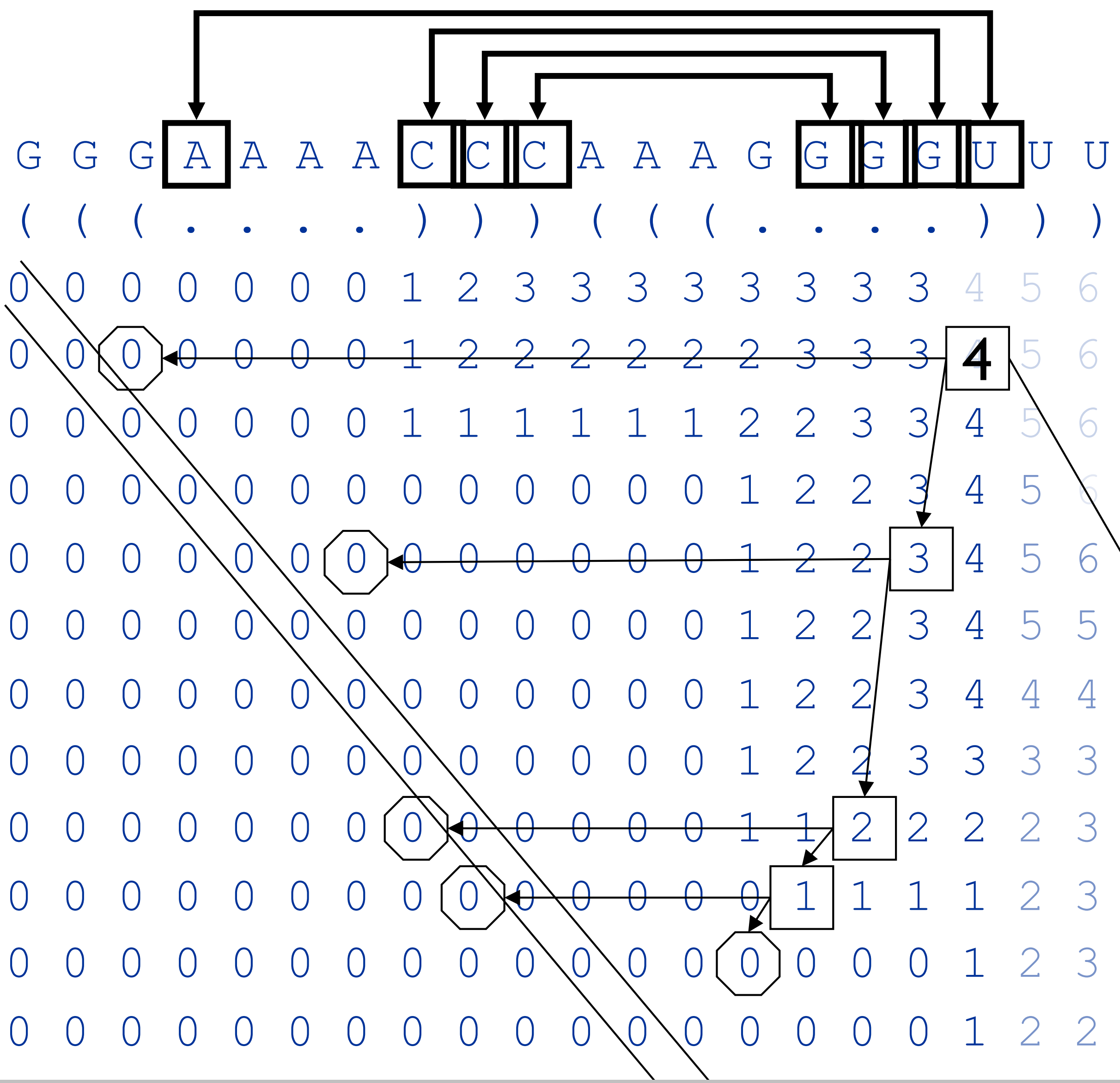
GGAAAACCC**A**AAGGGG**U**
((......))**(**.......**)**

(not shown:
t=9,10, 12,13)

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & \text{if } i \geq j-4 \\ \max\begin{Bmatrix} \mathrm{OPT}[i,j-1] \\ 1+\max_t(\mathrm{OPT}[i,t-1]+\mathrm{OPT}[t+1,j-1]) \end{Bmatrix} & \text{otherwise} \end{cases}$$

Computing one cell: OPT[2,18] = 4

n= 20



```
G G G A A A A C C C A A A G G G G U U U
( ( ( . . . . ) ) ) ( ( ( . . . . ) ) )
```

0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3	4	5	6
0	0	0	0	0	0	0	1	2	2	2	2	2	2	3	3	3	4	5	6
0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2

Overall, Max = 4
several ways, e.g.:

```
GGAAAACCCAAAGGGGU
..(...((((.....))))
```

tree shows trace back:

square = case 3

octagon = case 1

$$\mathrm{OPT}(i,j)=\begin{cases}0 & \text{if } i\ge j-4\\ \max\begin{cases}\mathrm{OPT}[i,j-1]\\ 1+\max_t(\mathrm{OPT}[i,t-1]+\mathrm{OPT}[t+1,j-1])\end{cases} & \text{otherwise}\end{cases}$$

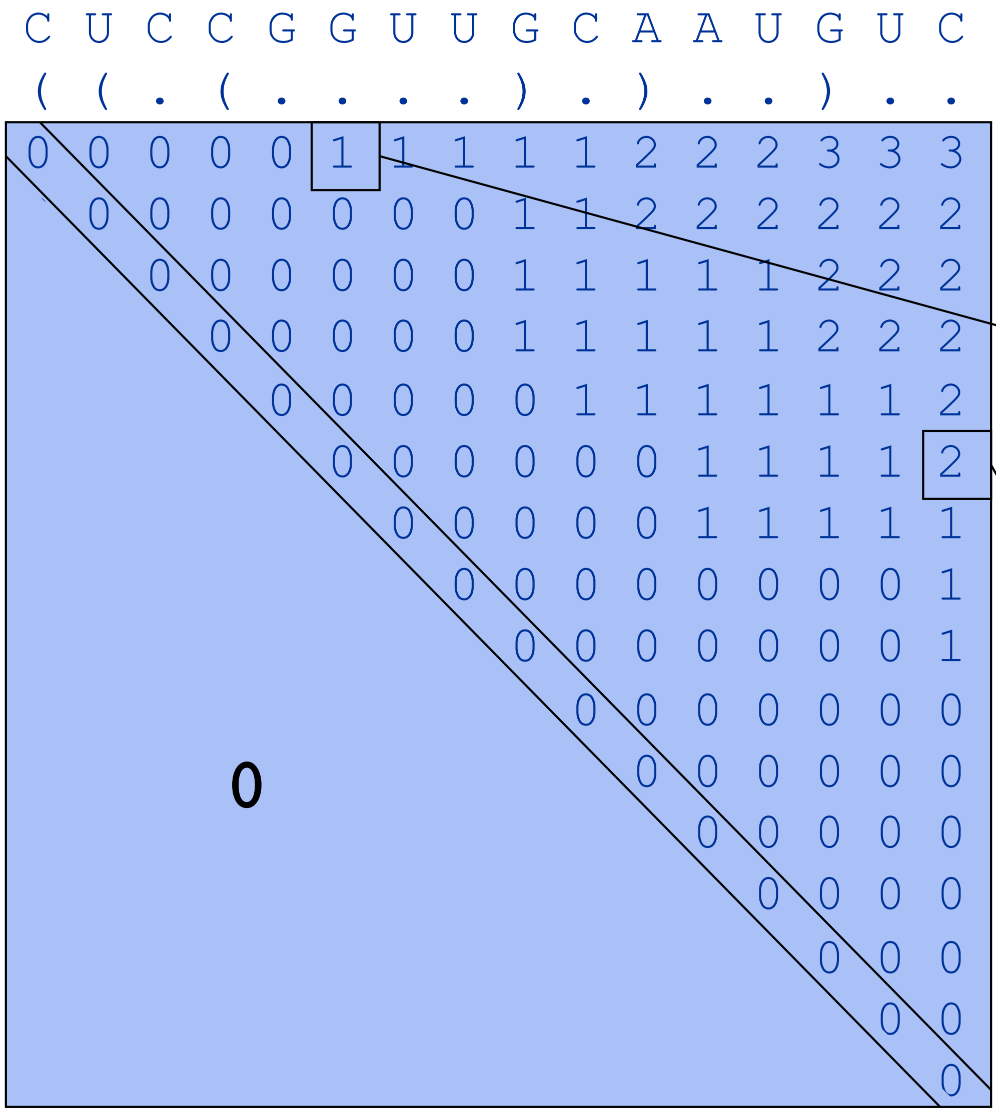
All 5 optimal structures on the above example

```
GGGAAAACCCAAAGGGGUUU
...(((.(((....))))))
...((.((((....))))))
...(.(((((....))))))
....((((((....))))))
(((....)))(((....)))

 0 0 0 0 0 0 0 1 2 3 3 3 3 3 3 3 3 4 5 6
-7 0 0 0 0 0 0 1 2 2 2 2 2 2 3 3 3 4 5 6
-7-7 0 0 0 0 0 1 1 1 1 1 1 2 2 3 3 4 5 6
-7-7-7 0 0 0 0 0 0 0 0 0 0 1 2 2 3 4 5 6
-7-7-7-7 0 0 0 0 0 0 0 0 0 1 2 2 3 4 5 6
-7-7-7-7-7 0 0 0 0 0 0 0 0 1 2 2 3 4 5 5
-7-7-7-7-7-7 0 0 0 0 0 0 0 1 2 2 3 4 4 4
-7-7-7-7-7-7-7 0 0 0 0 0 0 1 2 2 3 3 3 3
-7-7-7-7-7-7-7-7 0 0 0 0 0 1 1 2 2 2 2 3
-7-7-7-7-7-7-7-7-7 0 0 0 0 0 1 1 1 1 2 3
-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0 1 2 3
-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 1 2 2
-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 1 1 1
-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0 0
-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7-7 0
n= 20 Pairs= 6 AltStructs= 5 0.000117 (sec. total)
```

Another Example

n = 16



C	U	C	C	G	G	U	U	G	C	A	A	U	G	U	C
(	(	.	(	.	.	.	.	)	.	)	.	.	)	.	.
0	0	0	0	0	1	1	1	1	1	2	2	2	3	3	3
	0	0	0	0	0	0	0	1	1	2	2	2	2	2	2
		0	0	0	0	0	0	1	1	1	1	1	2	2	2
			0	0	0	0	0	1	1	1	1	1	2	2	2
				0	0	0	0	0	1	1	1	1	1	1	2
					0	0	0	0	0	0	1	1	1	1	2
						0	0	0	0	0	1	1	1	1	1
							0	0	0	0	0	0	0	0	1
								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

E.g.:
OPT[1,6] = 1:

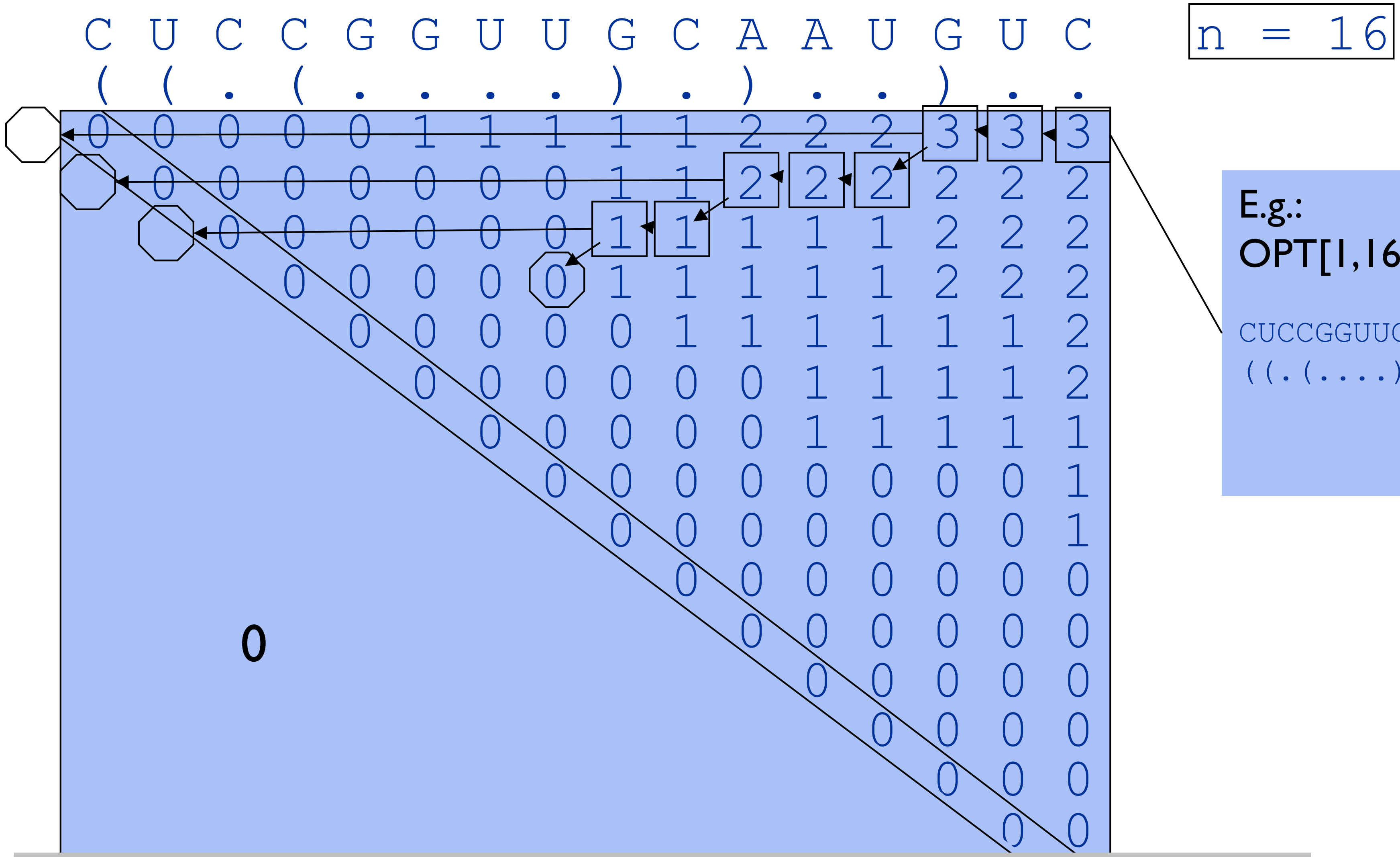
CUCCGG
(....)

E.g.:
OPT[6,16] = 2:

GUUGCAAUGUC
((....)...)

(Examples here and below assume 1-based indexing)

Another Trace Back Example



n = 16

C	U	C	C	G	G	U	U	G	C	A	A	U	G	U	C
(	(	.	(	.	.	.	.	)	.	)	.	.	)	.	.
0	0	0	0	0	1	1	1	1	1	2	2	2	3	3	3
	0	0	0	0	0	0	0	1	1	2	2	2	2	2	2
		0	0	0	0	0	0	1	1	1	1	1	2	2	2
			0	0	0	0	0	1	1	1	1	1	2	2	2
				0	0	0	0	0	1	1	1	1	1	1	2
					0	0	0	0	0	0	1	1	1	1	2
						0	0	0	0	0	1	1	1	1	1
							0	0	0	0	0	0	0	0	1
								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

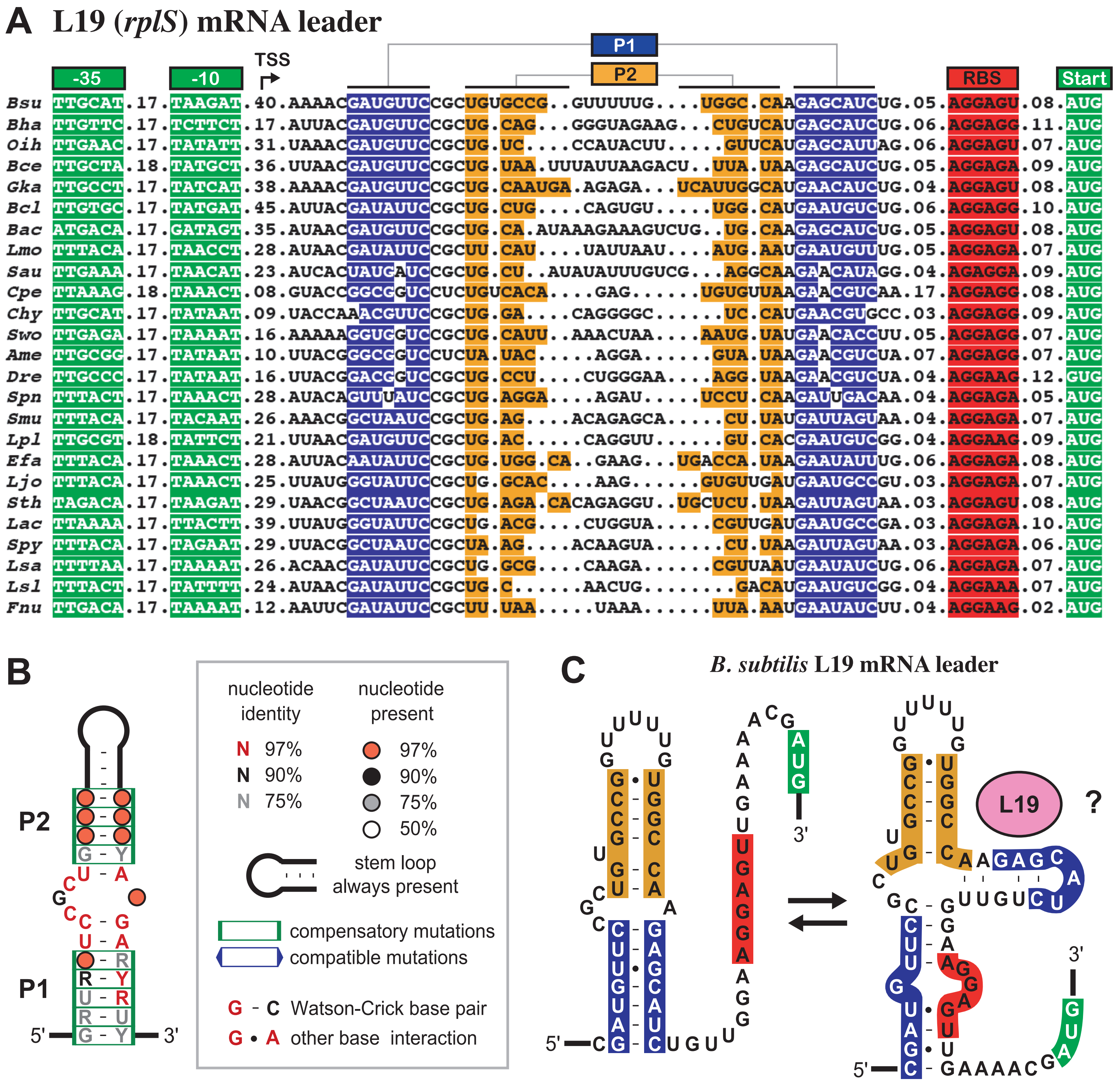
E.g.:
OPT[1,16] = 3:

```
CUCCGGUUGCAAUGUC
((.(....).)..)..
```

$$\mathrm{OPT}(i,j) = \begin{cases} 0 & \text{if } i \geq j-4 \\ \max\begin{Bmatrix} \mathrm{OPT}[i,j-1] \\ 1+\max_t(\mathrm{OPT}[i,t-1]+\mathrm{OPT}[t+1,j-1]) \end{Bmatrix} & \text{otherwise} \end{cases}$$

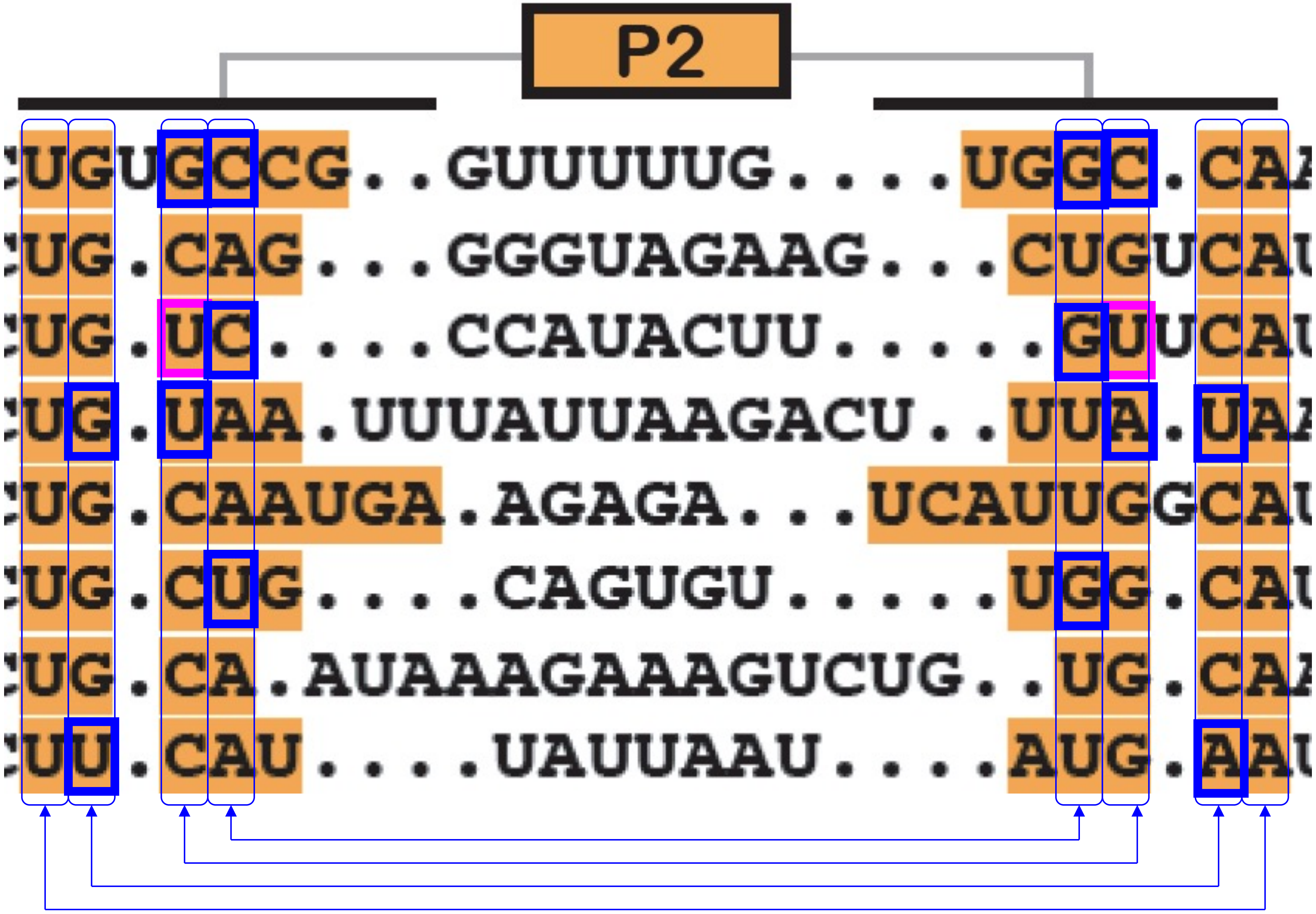
Example: Ribosomal Autoregulation:

Excess L19 represses L19 (RF00556; 555-559 similar)



A L19 (*rplS*) mRNA leader

```
      -35        -10     TSS       P1              P2                               P1              RBS        Start
Bsu TTGCAT.17.TAAGAT.40.AAAACGAUGUUCCGCUGUGCCG..GUUUUUG....UGGC.CAAGAGCAUCUG.05.AGGAGU.08.AUG
Bha TTGTTC.17.TCTTCT.17.AUUACGAUGUUCCGCUG.CAG...GGGUAGAAG...CUGUCAUGAGCAUCUG.06.AGGAGG.11.AUG
Oih TTGAAC.17.TATATT.31.UAAACGAUGUUCCGCUG.UC....CCAUACUU.....GUUCAUGAGCAUUAG.06.AGGAGU.07.AUG
Bce TTGCTA.18.TATGCT.36.UUAACGAUGUUCCGCUG.UAA.UUUAUUAAGACU..UUA.UAAGAGCAUCUG.05.AGGAGA.09.AUG
Gka TTGCCT.17.TATCAT.38.AAAACGAUGUUCCGCUG.CAAUGA.AGAGA...UCAUUGGCAUGAACAUCUG.04.AGGAGU.08.AUG
Bcl TTGTGC.17.TATGAT.45.AUUACGAUAUUCCGCUG.CUG....CAGUGU.....UGG.CAUGAAUGUCUG.06.AGGAGG.10.AUG
Bac ATGACA.17.GATAGT.35.AUAACGAUGUUCCGCUG.CA.AUAAAGAAAGUCUG..UG.CAAGAGCAUCUG.05.AGGAGU.08.AUG
Lmo TTTACA.17.TAACCT.28.AUAACGAUAUUCCGCUU.CAU....UAUUAAU....AUG.AAUGAAUGUUUG.05.AGGAGA.07.AUG
Sau TTGAAA.17.TAACAT.23.AUCACUAUGAUCCGCUG.CU..AUAUAUUUGUCG...AGGCAAGAACAUAGG.04.AGAGGA.09.AUG
Cpe TTAAAG.18.TAAACT.08.GUACCGGCGGUCCUCUGUCACA....GAG......UGUGUUAAGAACGUCAA.17.AGGAGG.08.AUG
Chy TTGCAT.17.TATAAT.09.UACCAAACGUUCCGCUG.GA....CAGGGGC......UC.CAUGAACGUGCC.03.AGGAGG.09.AUG
Swo TTGAGA.17.TAAAAT.16.AAAAAGGUGGUCCGCUG.CAUU..AAACUAA....AAUG.UAUGAACACCUU.05.AGGAGG.07.AUG
Ame TTGCGG.17.TATAAT.10.UUACGGGCGGUCCUCUA.UAC.....AGGA......GUA.UAAGAACGUCUA.07.AGGAGG.07.AUG
Dre TTGCCC.17.TATAAT.16.UUACGGACGGUCCGCUG.CCU....CUGGGAA....AGG.UAAGAACGUCUA.04.AGGAAG.12.GUG
Spn TTTACT.17.TAAACT.28.AUACAGUUUAUCCGCUG.AGGA....AGAU.....UCCU.CAAGAUUGACAA.04.AGGAGA.05.AUG
Smu TTTACA.17.TACAAT.26.AAACGGCUAAUCCGCUG.AG....ACAGAGCA.....CU.UAUGAUUAGUAA.04.AGGAGA.07.AUG
Lpl TTGCGT.18.TATTCT.21.UUAACGAUGUUCCGCUG.AC.....CAGGUU......GU.CACGAAUGUCGG.04.AGGAAG.09.AUG
Efa TTTACA.17.TAAACT.28.AUUACAAUAUUCCGCUG.UGG.CA..GAAG...UGACCA.UAAGAAUAUUUG.06.AGGAGA.08.AUG
Ljo TTTACA.17.TAAACT.25.UUAUGGGUAUUCCGCUG.GCAC....AAG......GUGUUGAUGAAUGCCGU.03.AGGAGA.07.AUG
Sth TAGACA.17.TAAGAT.29.UAACGGCUAAUCCGCUG.AGA.CACAGAGGU..UGCUCU.UAAGAUUAGUAA.03.AGGAGU.08.AUG
Lac TTAAAA.17.TTACTT.39.UUAUGGGUAUUCCGCUG.ACG....CUGGUA.....CGUUGAUGAAUGCCGA.03.AGGAGA.10.AUG
Spy TTTACA.17.TAGAAT.29.UUACGGCUAAUCCGCUA.AG....ACAAGUA......CU.UAAGAUUAGUAA.03.AGGAGA.06.AUG
Lsa TTTTAA.17.TAAAAT.26.ACAACGAUAUUCCGCUG.GCG....CAAGA......CGUUAAUGAAUAUCUG.06.AGGAGA.07.AUG
Lsl TTTACT.17.TATTTT.24.AUAACGAUAUUCCGCUG.C......AACUG........GACAUGAAUGUCGG.04.AGGAAA.07.AUG
Fnu TTGACA.17.TAAAAT.12.AAUUCGAUAUUCCGCUU.UAA.....UAAA......UUA.AAUGAAUAUCUU.04.AGGAAG.02.AUG
```


P2
UGUGCCG..GUUUUUG....UGGC.CA
UG.CAG...GGGUAGAAG...CUGUCA
UG.UC....CCAUACUU.....GUUCA
UG.UAA.UUUAUUAAGACU..UUA.UA
UG.CAAUGA.AGAGA...UCAUUGGCA
UG.CUG....CAGUGU.....UGG.CA
UG.CA.AUAAAGAAAGUCUG..UG.CA
UU.CAU....UAUUAAU....AUG.AA

Covariation is strong evidence for base pairing

Summary

- RNA has important roles
  - Beyond mRNA; many unexpected recent discoveries
- Structure is critical to function
  - True of other molecules, too
- RNA secondary structure prediction is a key tool
- Dynamic programming–useful accuracy, $O(n^3)$ time:
  - Binary choice again: last base is paired or not
  - Optimal substructure again: given last pair, optimally fold inside & outside separately
  - Tabulate again: best folding of all substrings.