

CSE 42I: Intro Algorithms

W. L. Ruzzo

Dynamic Programming:
String alignment and RNA Folding

Outline

A few slides on *applications* of dynamic programming in biology

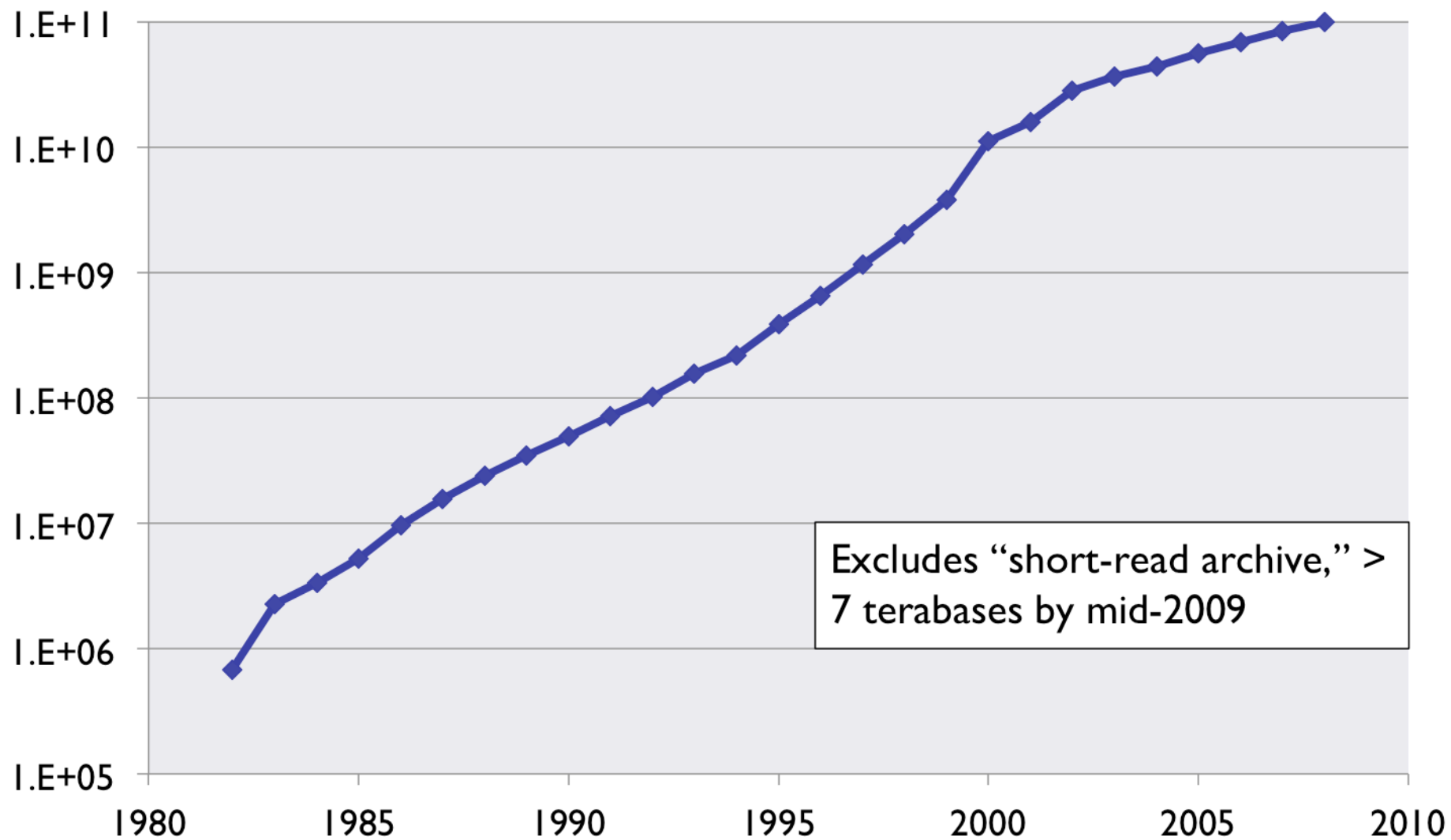
Sequence alignment

RNA structure

Algorithms for RNA structure

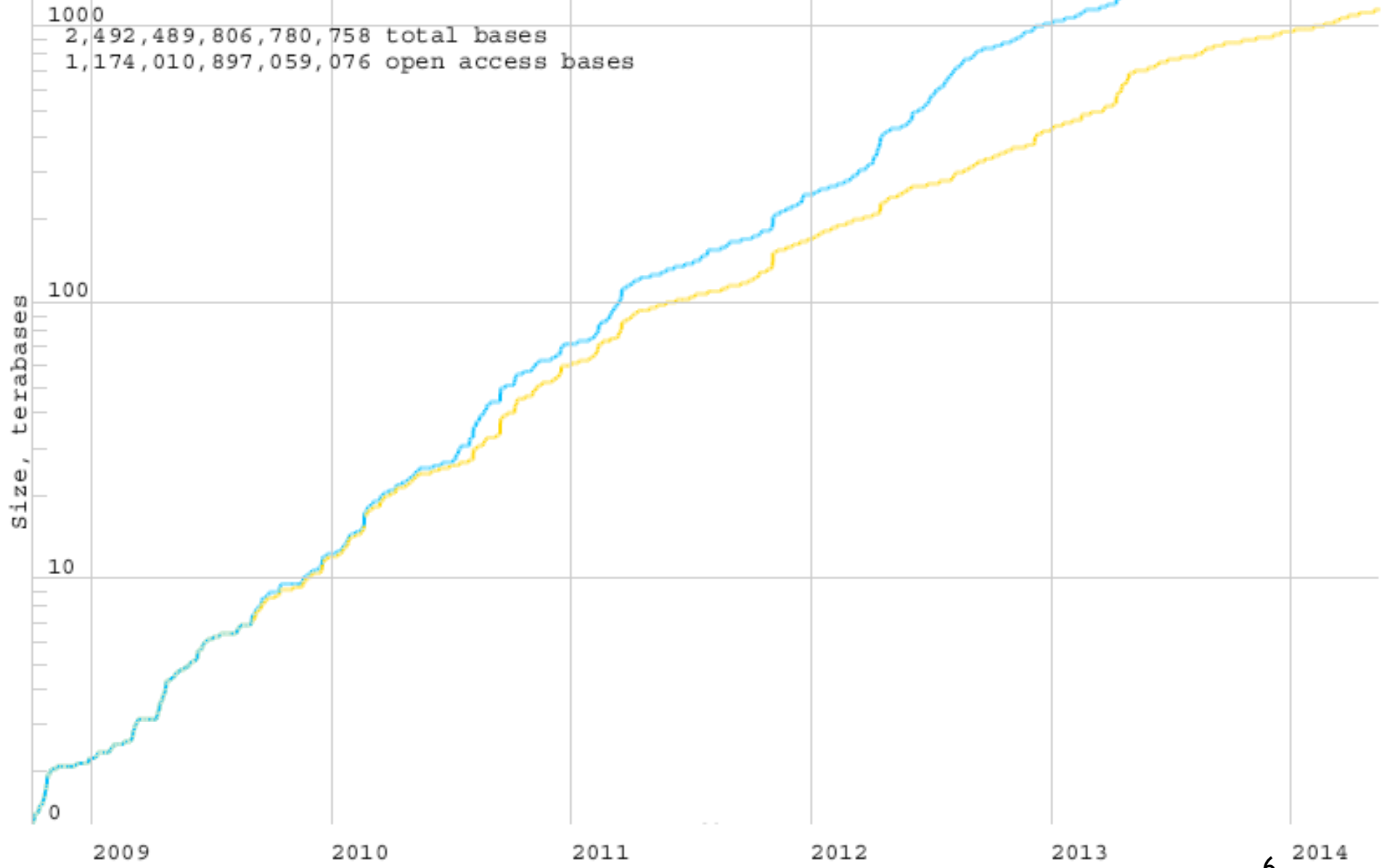
Application: Sequence Search

Growth of GenBank (Base Pairs)



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

SRA database growth



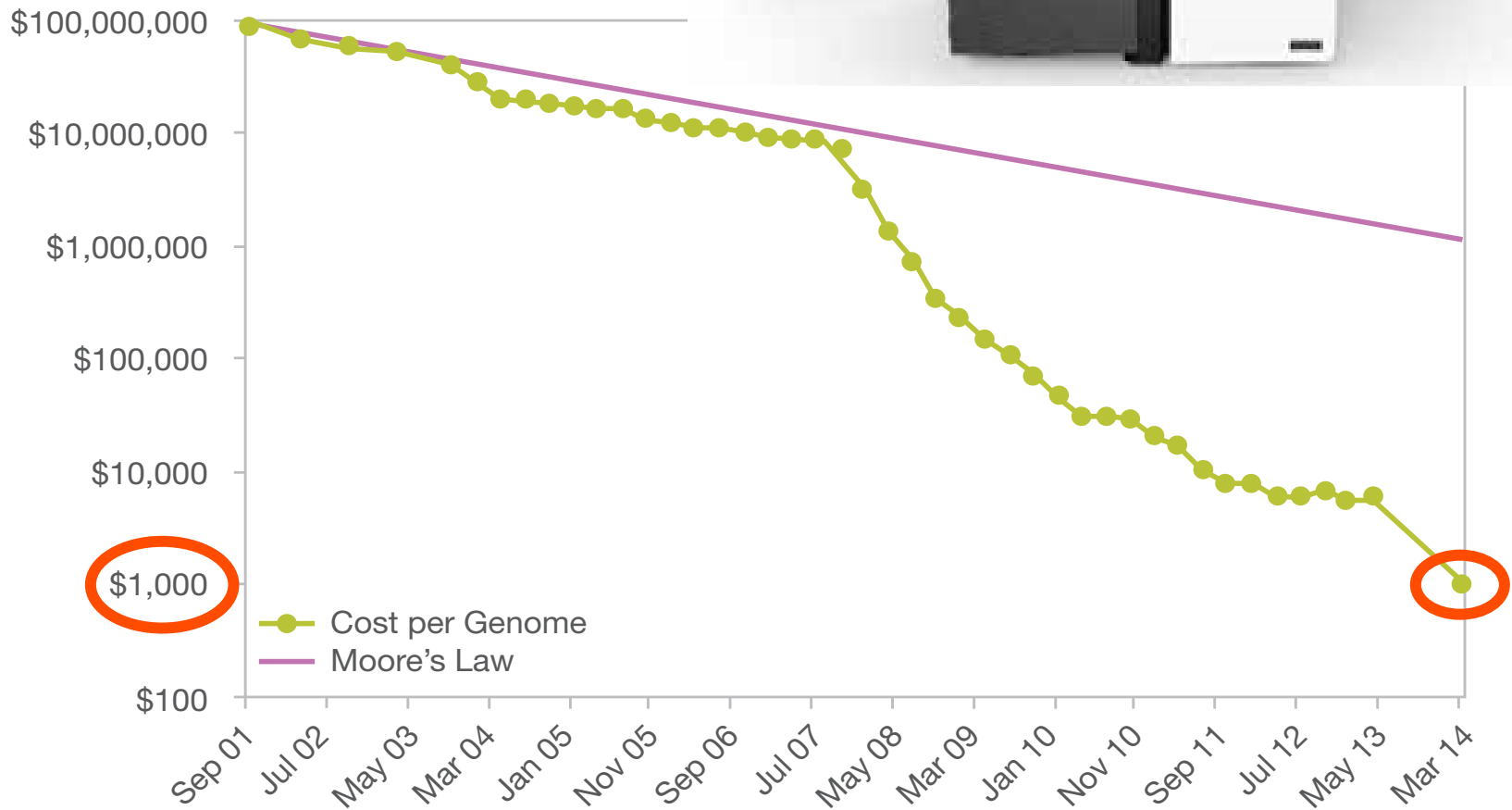
Total bases — (blue line)
Open access bases — (yellow line)

<http://www.ncbi.nlm.nih.gov/Traces/sra/i/g.png>

05/14/2014 12:59pm

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

...And 100'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

Alignment 1 against Q7T109

Score	964	E-value	1.0 ×10 ⁻¹⁰²
Identity	64.0%	Positives	74.0%
Query length	320	Match length	288

Position Q7T109 matches from 1 to 288 (288AA), in the query sequence from 1 to 320 (320AA)

Graphical



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

61	EHSHFPAAVHPAPGAREDEHVRAPSGHHQAGRCLLWACKACKRKT TNADRRKAATMRERR	120	P15172
	+ H EDEHVRAPSGHHQAGRCLLWACKACKRKT TNADRRKAATMRERR		
58	DPHH-----NEDEHVRAPSGHHQAGRCLLWACKACKRKT TNADRRKAATMRERR	106	Q7T109

121	RLSKVNEAFETLKRCTSSNP NQRLPKVEILRN A IRYIEGLQALLRDQDAAPP GAAA AFYA	180	P15172
	RLSKVNEAFETLKRCTS+NP NQRLPKVEILRN A IRYIE LQ+LLR Q+ +FY		
107	RLSKVNEAFETLKRCTSTNP NQRLPKVEILRN A IRYIESLQSLLRGQE-----ESFY-	158	Q7T109

181	PGPLPPGRGGEHYS GDS DASSPRSNCS DGMMDYS GPPSGARRRNCYEGAYYNEAPSEPRP	240	P15172
	P+ EHYS GDS DASSPRSNCS DGM DYS PP G+RRRN Y+ ++Y+++P+ R		
159	--PVL-----EHYS GDS DASSPRSNCS DGMTDYS-PPCGSRRRNSYDSSFYS DSPNGLRL	210	Q7T109

241	GKSAAVSSLDCLSSIVERISTESPAAPALLLADV PSESPPRRQEAAAPSEGES---SGDP	297	P15172
	GKS+ +SSLDCLSSIVERISTESP P + AD SE P +P +GE+ SG		10
211	GKSSVISSLDCLSSIVERISTESPVC PVIPAADSGSEGSP-----CSPLQGETLSESGII	265	Q7T109

Alignments	Entry	Entry name	Status	Protein names	Organism	Length	Identity	Score	E-value	Gene names
	B3LY60	B3LY60_DROAN	★	GF18746	Drosophila ananassae (Fruit fly)	334	42.0%	344	4.0×10 ⁻³⁰	GF18746 Dana\GF18746 Dana_GF18746
	Q4RGJ6	Q4RGJ6_TETNG	★	Chromosome undetermined SCAF15099, whole geno...	Tetraodon nigroviridis (Spotted green pufferfish) (Chelonodon nigroviridis)	126	57.0%	343	5.0×10 ⁻³⁰	GSTENG00034775001
	F1NFS6	F1NFS6_CHICK	★	Uncharacterized protein	Gallus gallus (Chicken)	237	46.0%	343	5.0×10 ⁻³⁰	Gga.378
	Q91151	Q91151_NOTVI	★	Myogenic regulatory factor; transcription fac...	Notophthalmus viridescens (Eastern newt) (Triturus viridescens)	219	44.0%	342	6.0×10 ⁻³⁰	MRF-4
	Q29BN7	Q29BN7_DROPS	★	GA10192	Drosophila pseudoobscura pseudoobscura (Fruit fly)	330	42.0%	342	6.0×10 ⁻³⁰	GA10192 Dpse\GA10192 Dpse_GA10192
	B4GP81	B4GP81_DROPE	★	GL13832	Drosophila persimilis (Fruit fly)	330	42.0%	342	6.0×10 ⁻³⁰	GL13832 Dper\GL13832 Dper_GL13832
	Q92020	MYF6_XENLA	★	Myogenic factor 6	Xenopus laevis (African clawed frog)	240	44.0%	340	1.0×10 ⁻²⁹	myf6 mrf4
	B7ZQB0	B7ZQB0_XENLA	★	MRF4a	Xenopus laevis (African clawed frog)	240	44.0%	340	1.0×10 ⁻²⁹	MRF4
	A7UCI1	A7UCI1_XENLA	★	MRF4a	Xenopus laevis (African clawed frog)	240	44.0%	340	1.0×10 ⁻²⁹	MRF4A MRF4
	F7FIX8	F7FIX8_MONDO	★	Uncharacterized protein	Monodelphis domestica (Gray short-tailed opossum)	243	47.0%	339	1.0×10 ⁻²⁹	MYF6
	D9IV56	D9IV56_EPICO	★	Myogenin	Epinephelus coioides (Orange-spotted grouper) (Epinephelus nebulosus)	250	47.0%	338	2.0×10 ⁻²⁹	
	Q6SYV5	MYF6_TAKRU	★	Myogenic factor 6	Takifugu rubripes (Japanese pufferfish) (Fugu rubripes)	239	46.0%	337	2.0×10 ⁻²⁹	myf6 mrf4
	G3WJP0	G3WJP0_SARHA	★	Uncharacterized protein	Sarcophilus harrisii (Tasmanian devil) (Sarcophilus lanarius)	243	47.0%	337	2.0×10 ⁻²⁹	MYF6
	G7Y452	G7Y452_CLOSI	★	Transcription factor SUM-1	Clonorchis sinensis (Chinese liver fluke)	946	64.0%	337	2.0×10 ⁻²⁹	CLF_100742
	G4LXJ1	G4LXJ1_SCHMA	★	Myogenic factor, putative	Schistosoma mansoni (Blood fluke)	864	68.0%	337	2.0×10 ⁻²⁹	Smp_167400
	G1EN33	G1EN33_DUGJA	★	Myogenic determinant factor	Dugesia japonica (Planarian)	498	42.0%	336	3.0×10 ⁻²⁹	MDF
	Q8MWP6	Q8MWP6_SCHMD	★	MyoD-like protein	Schmidtea mediterranea (Freshwater planarian flatworm)	466	69.0%	335	4.0×10 ⁻²⁹	
	Q6PUV5	MYF6_TETNG	★	Myogenic factor 6	Tetraodon nigroviridis (Spotted green pufferfish) (Chelonodon nigroviridis)	239	45.0%	333	7.0×10 ⁻²⁹	myf6 mrf4 GSTENG00021536001
	Q2PZ12	Q2PZ12_SALSA	★	Myogenin	Salmo salar (Atlantic salmon)	254	55.0%	333	7.0×10 ⁻²⁹	
	Q91207	Q91207_ONCMY	★	TMyogenin protein	Oncorhynchus mykiss (Rainbow trout) (Salmo gairdneri)	254	45.0%	332	9.0×10 ⁻²⁹	TMyogenin

hits at rank ~ 250 still extremely good matches, even though very distantly related organisms

The foregoing search capability is a *huge* deal

millions of searches daily

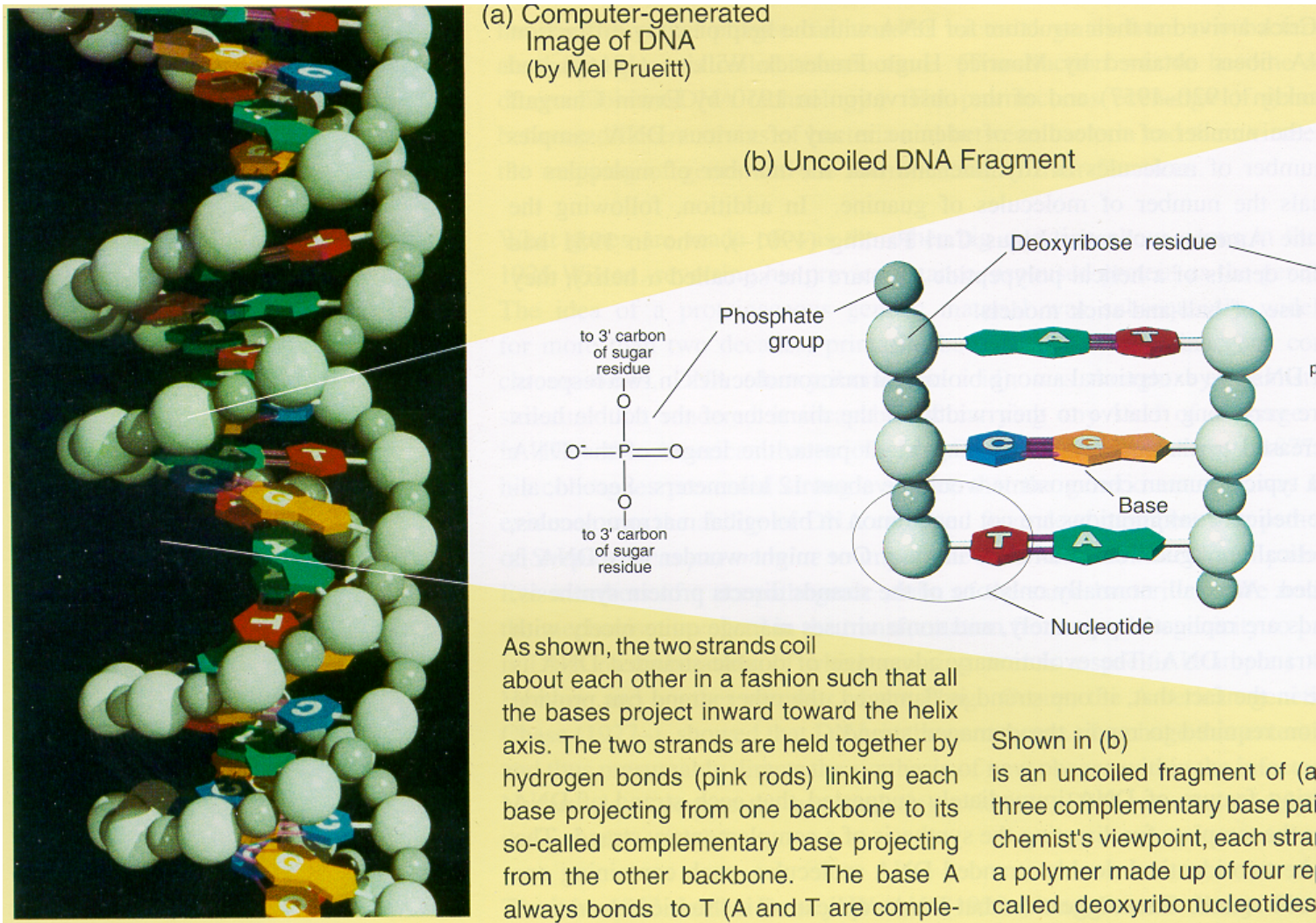
biologists (not just “computational”
biologists) use this routinely

it connects information about *all* living things

More on algorithm later ...

Application: RNA structure

The Double Helix



Central Dogma of Molecular Biology

by

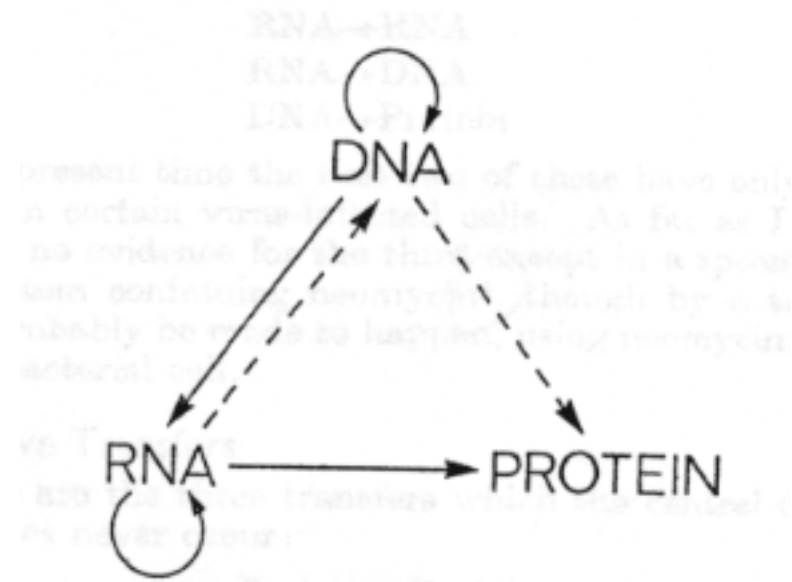
FRANCIS CRICK

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

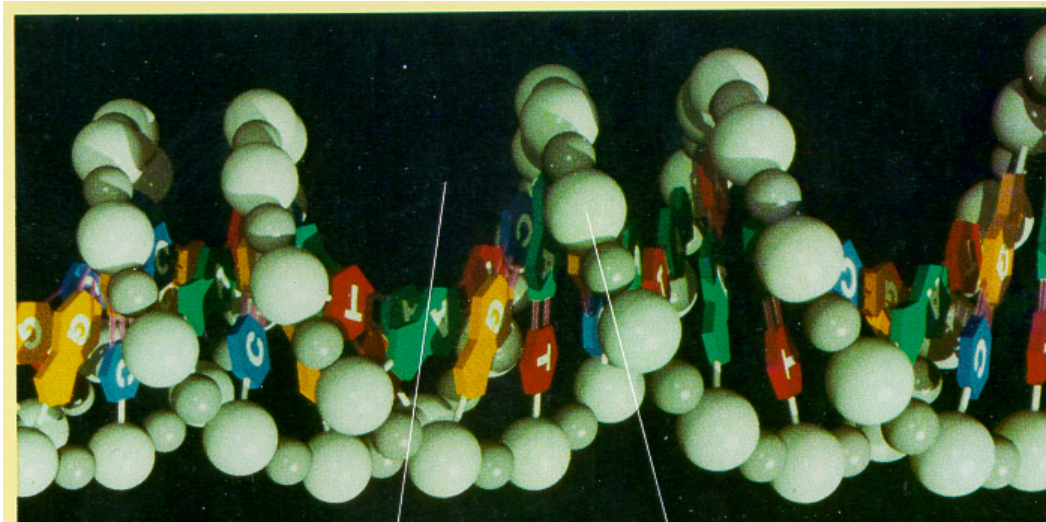
E.g. "miRNA": >1000 in humans; regulate >50% of genes

E.g. "riboswitches": 10000's in bacteria

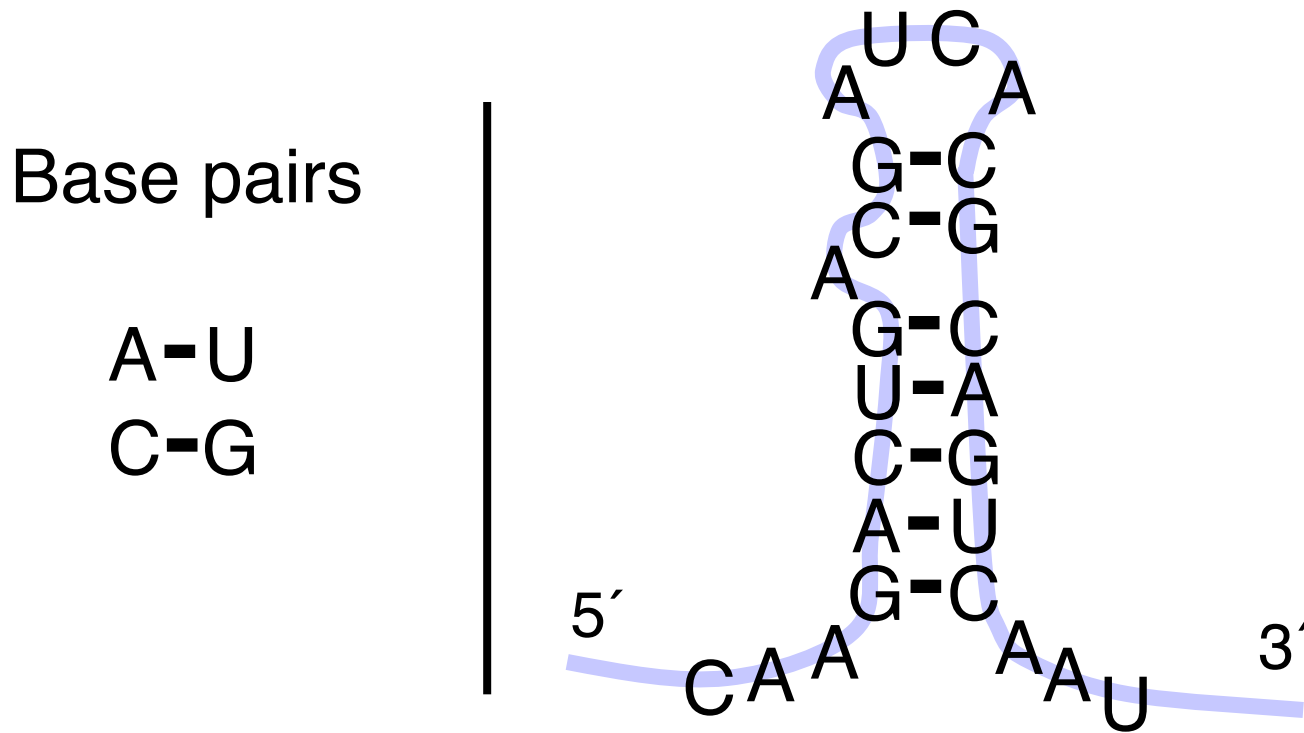
By some estimates, ncRNA >> mRNA

DNA structure: dull

5'...ACCGCTAGATG...3'
| | | | | | | | | |
3'...TGGCGATCTAC...5'



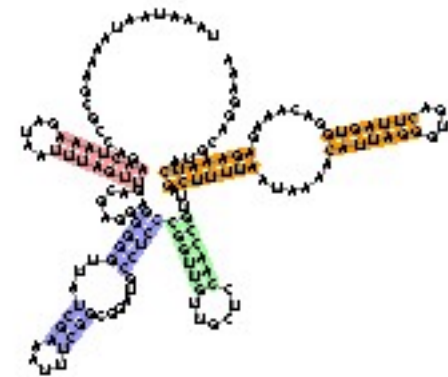
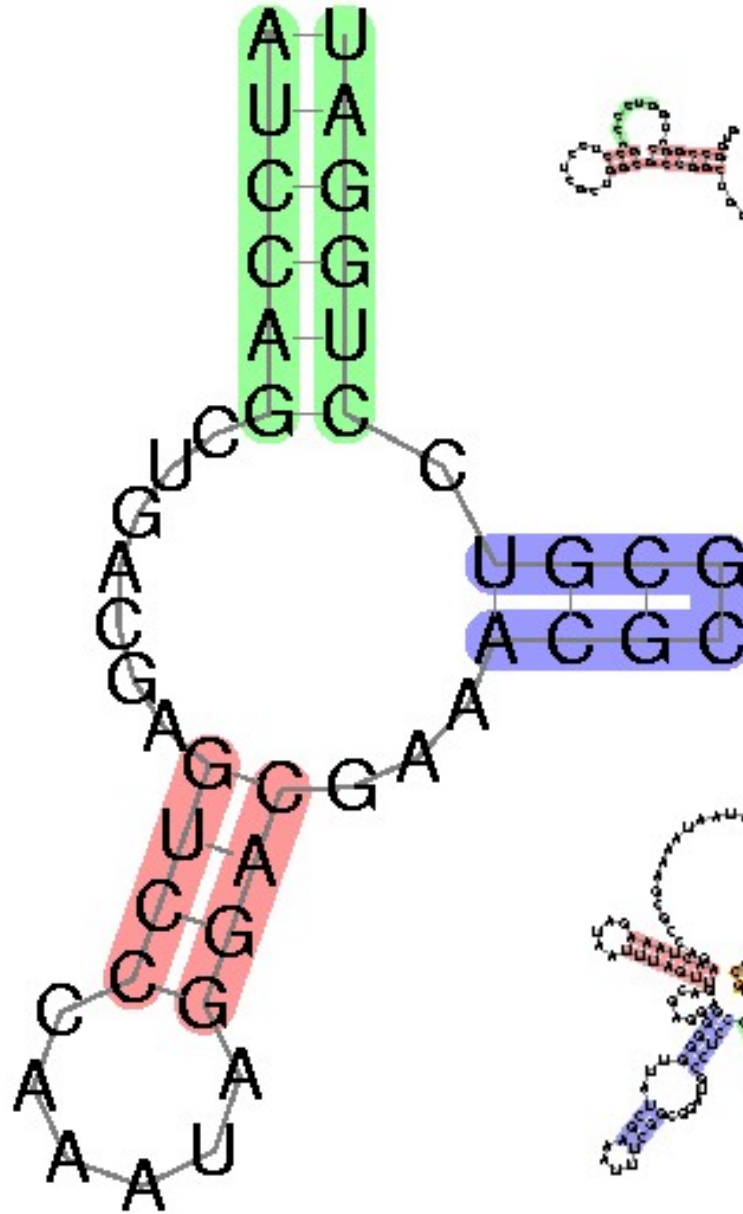
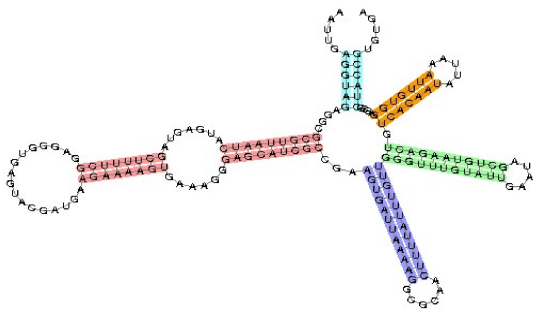
RNA Secondary Structure: RNA makes helices too



Usually *single* stranded

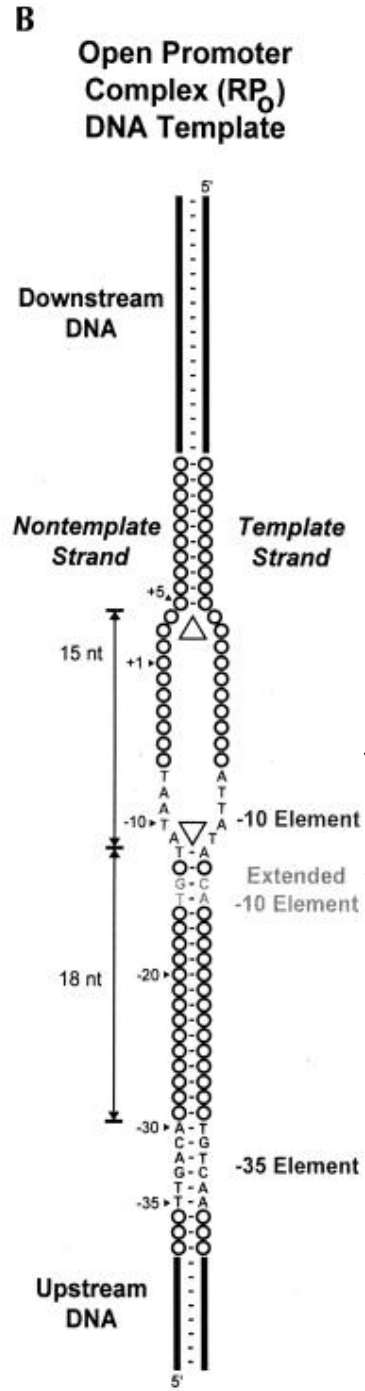
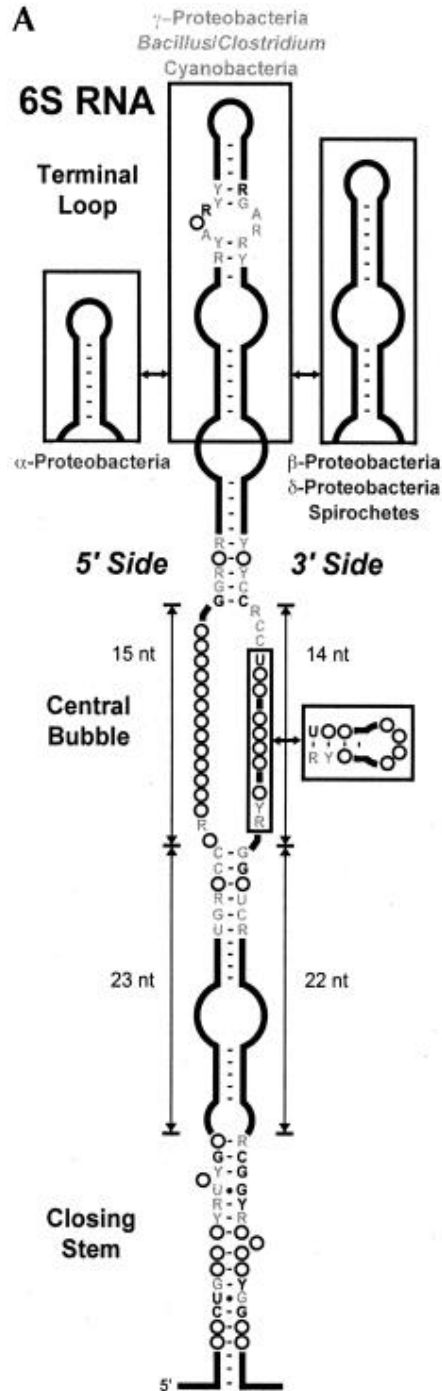
RNA Secondary Structure:

Not everything,
but important,
easier than 3d

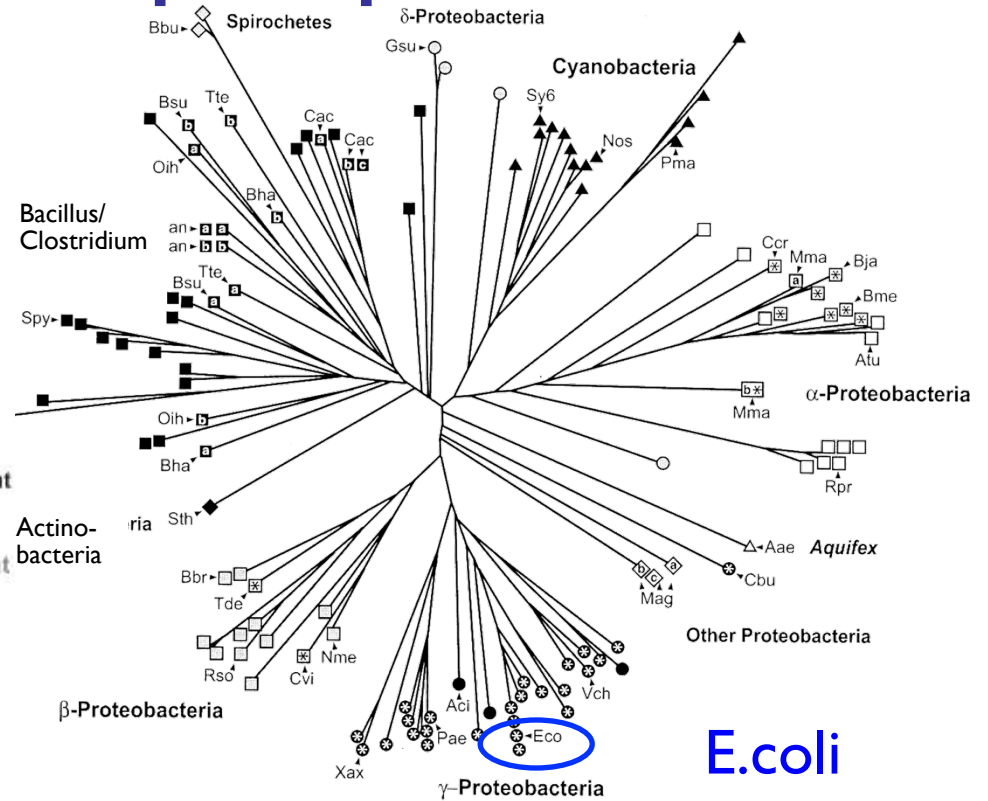


Why is structure important?

- 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



6S mimics an open promoter

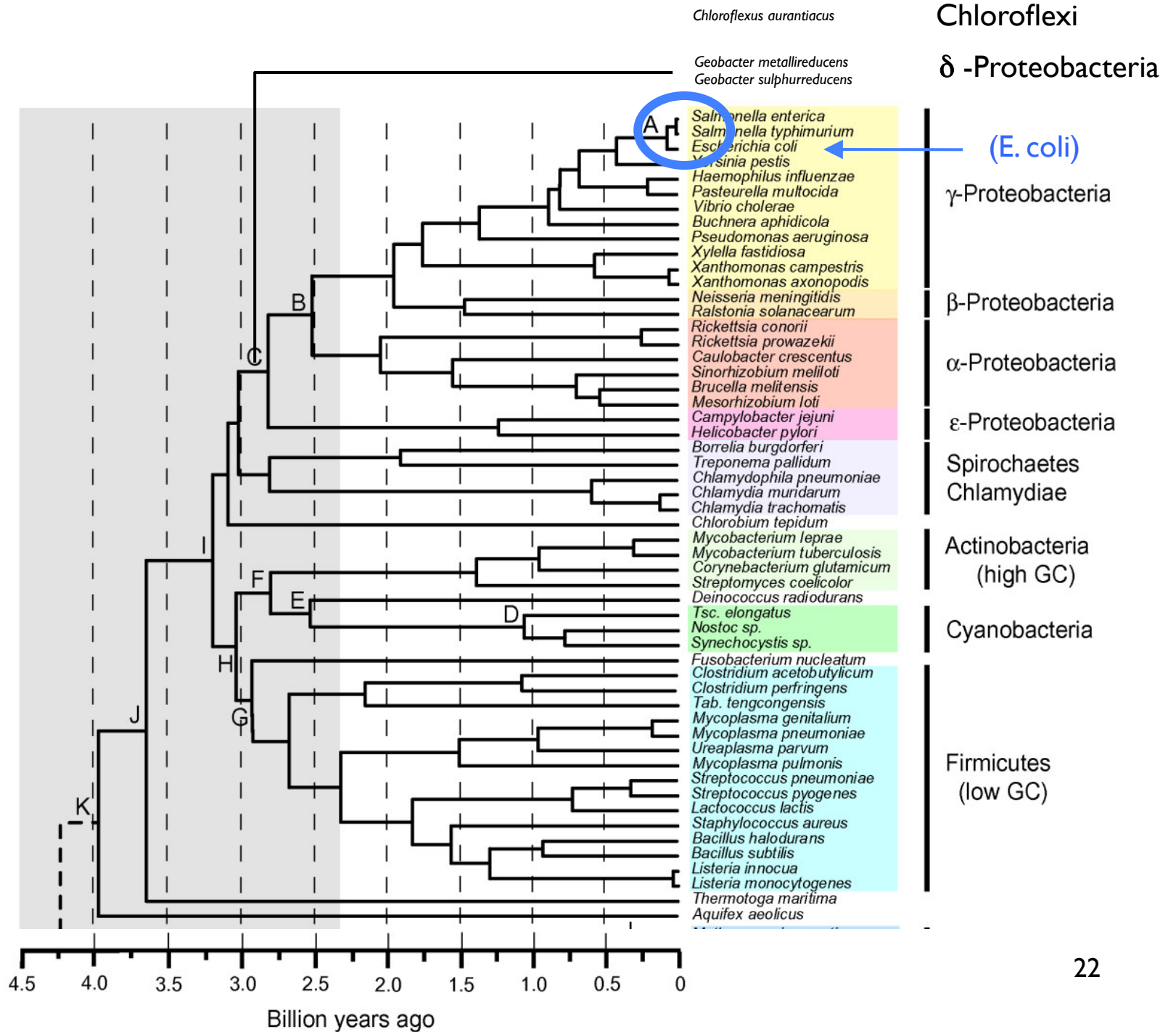


E.coli

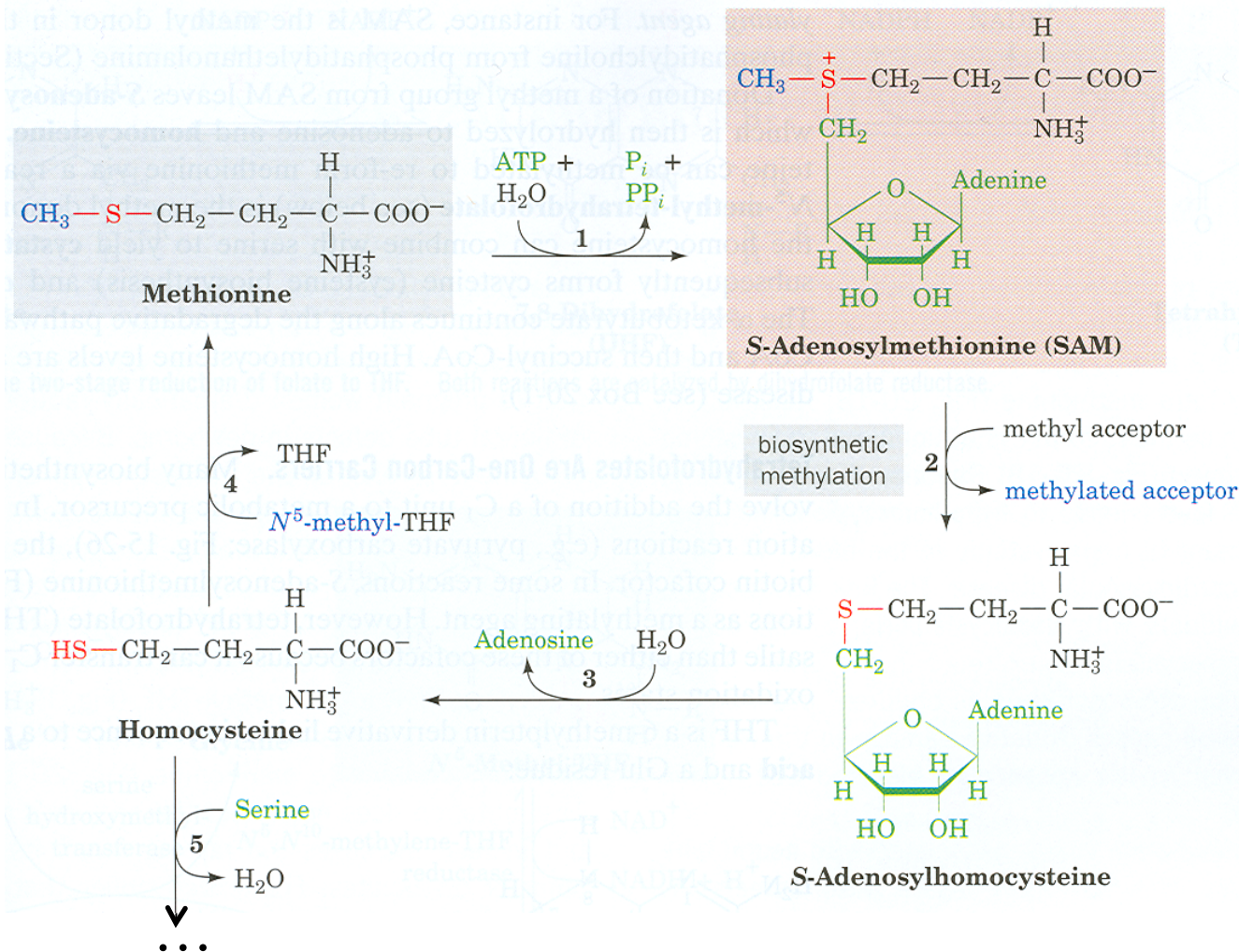
Barrick et al. *RNA* 2005

Trotochaud et al. *NSMB* 2005

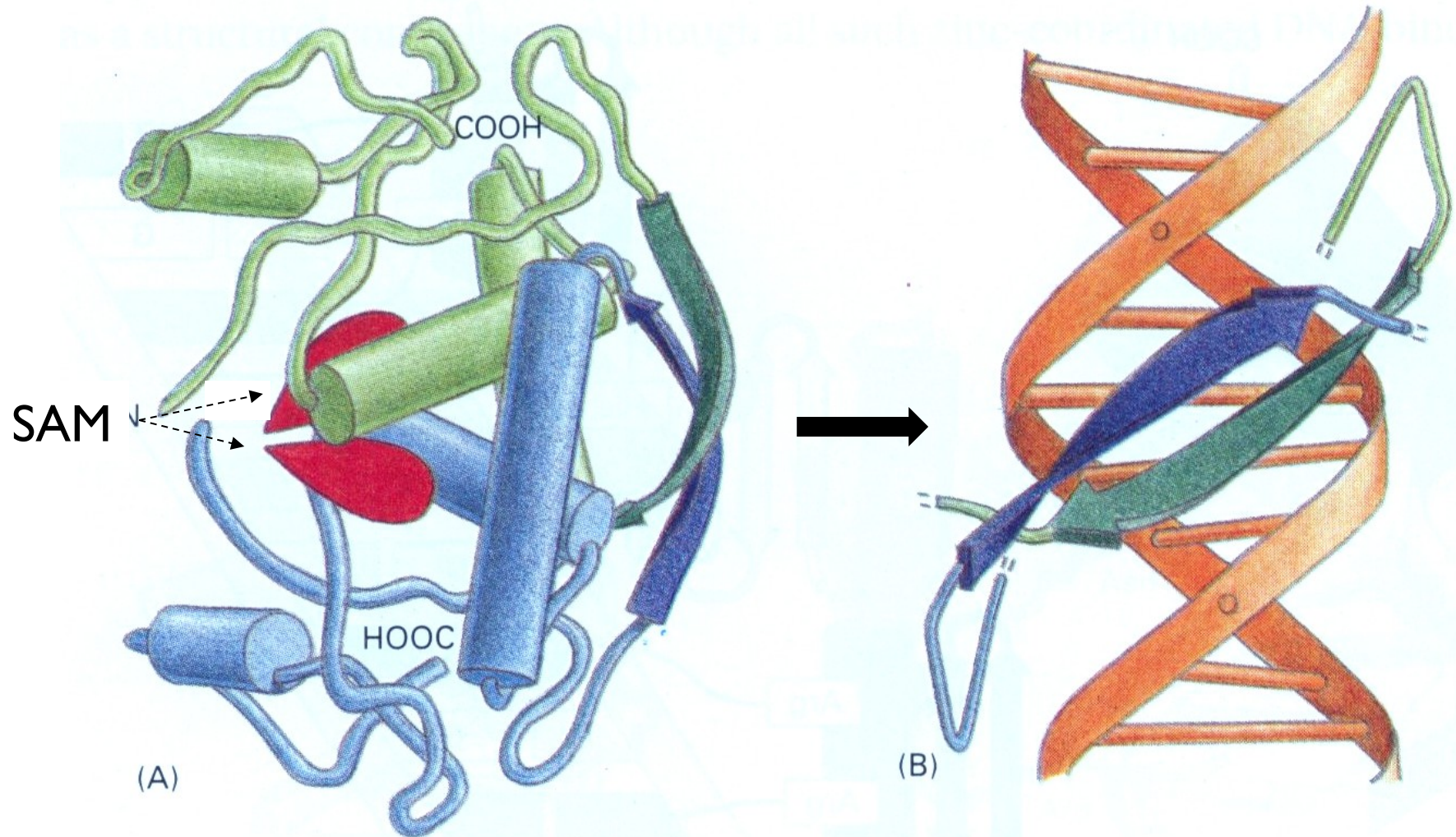
Willkomm et al. *NAR* 2005²¹



In Bacteria: A typical biosynthetic cycle around a critical metabolite (“SAM”)



Gene Regulation: The MET Repressor



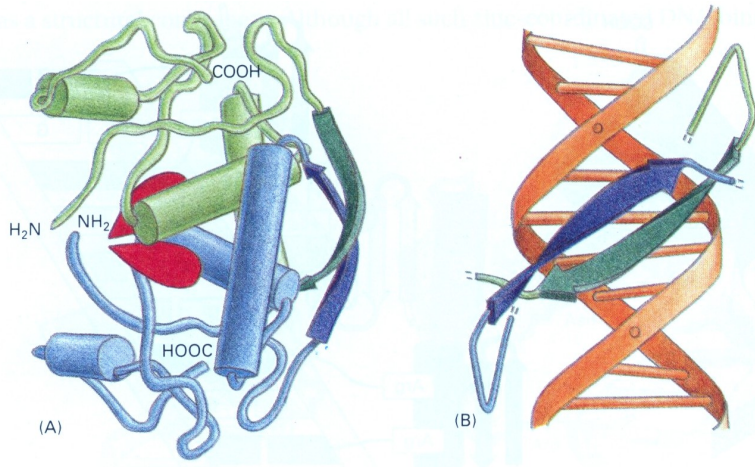
Protein

Alberts, et al, 3e.

DNA

24

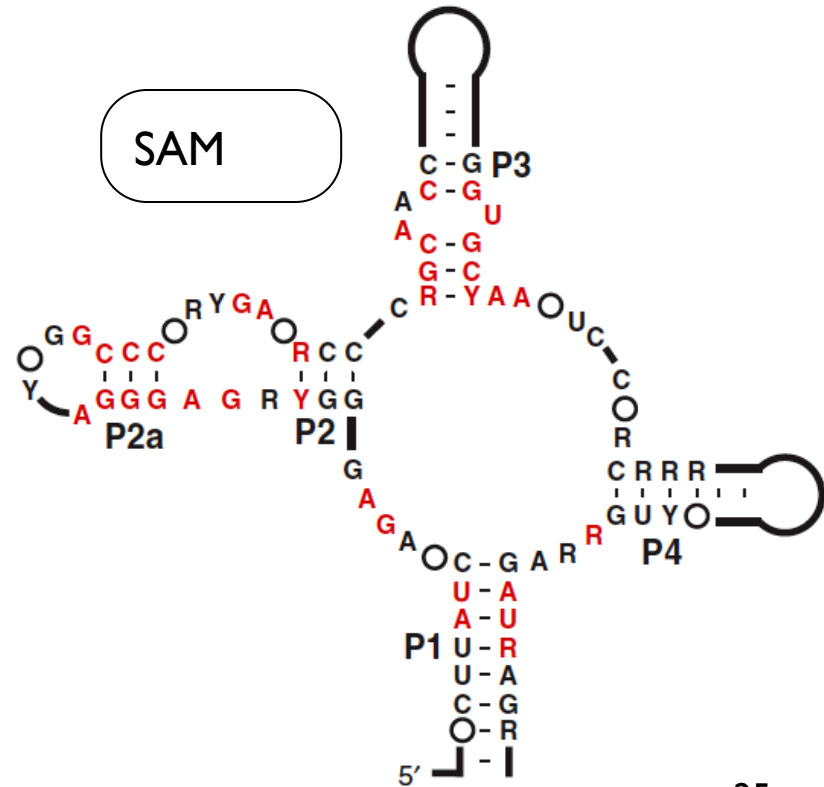
Alberts, et al, 3e.



Not the only way!

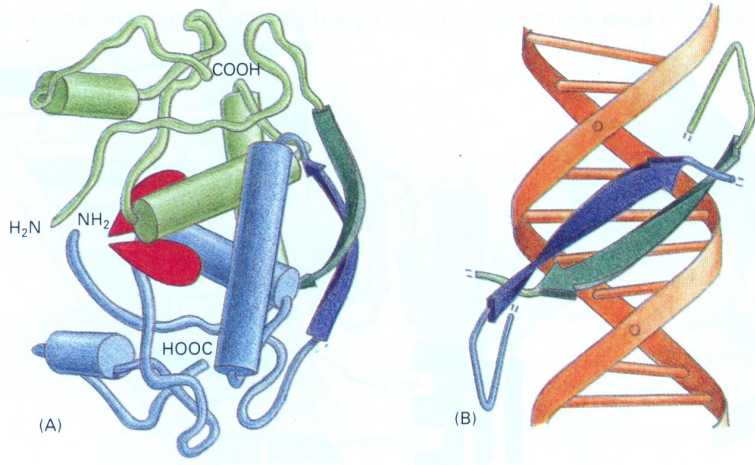
Protein way

Riboswitch alternative



Grundy & Henkin, Mol. Microbiol 1998
Epshtein, et al., PNAS 2003
Winkler et al., Nat. Struct. Biol. 2003

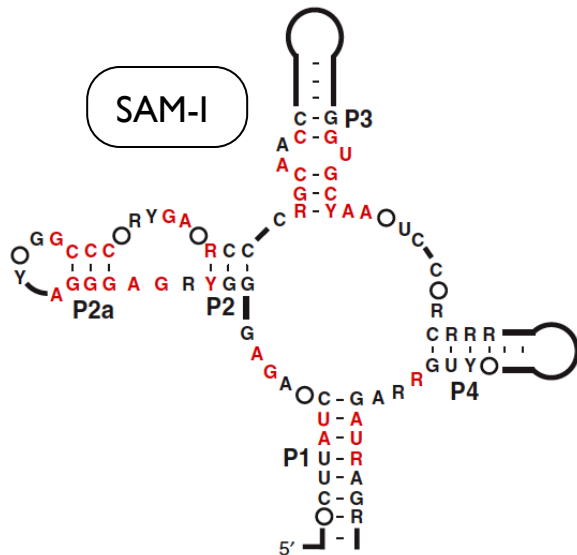
Alberts, et al, 3e.



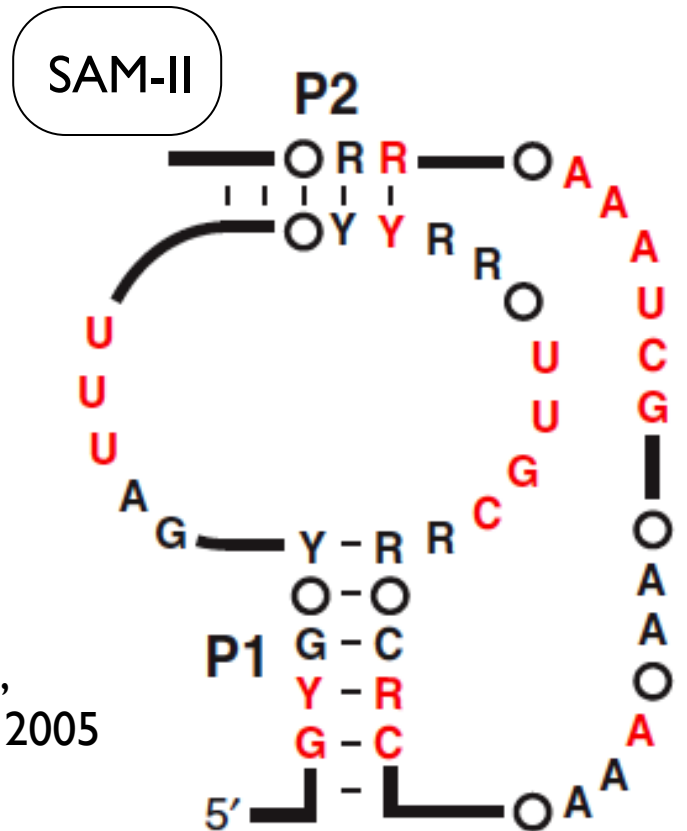
Not the only way!

Protein way

Riboswitch alternatives

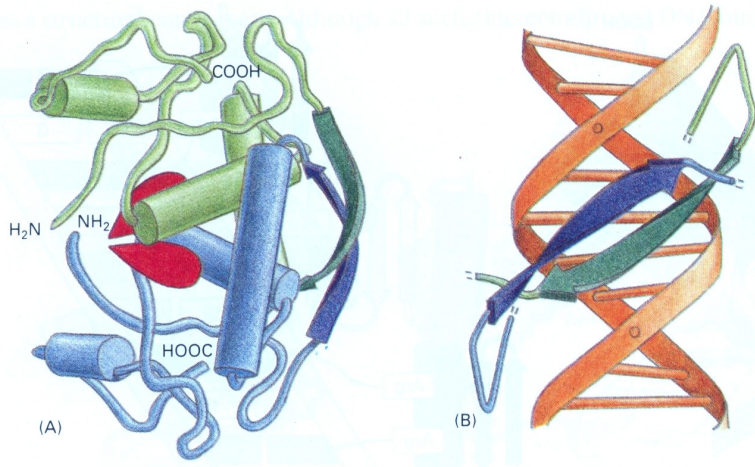


Grundy, Epshtein, Winkler et al., 1998, 2003



Corbino et al.,
Genome Biol. 2005

Alberts, et al, 3e.



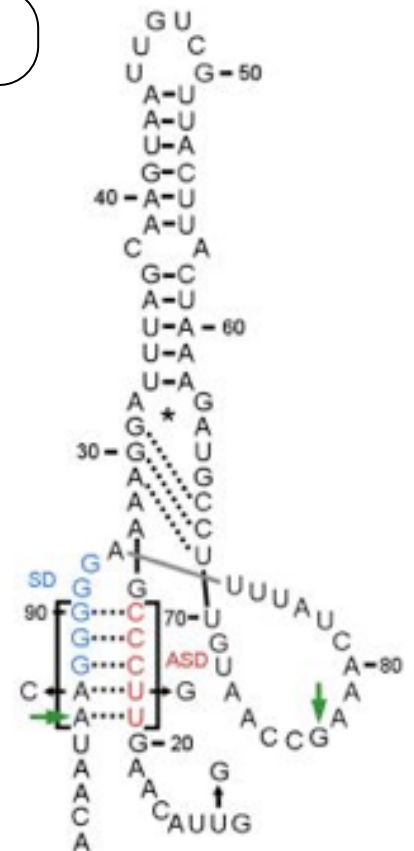
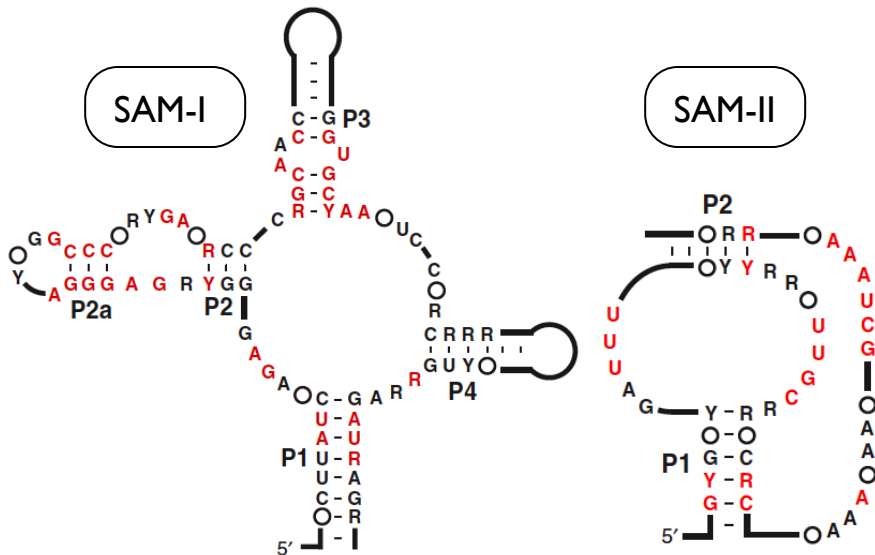
Not the only way!

Protein way

Riboswitch alternatives



SAM-III

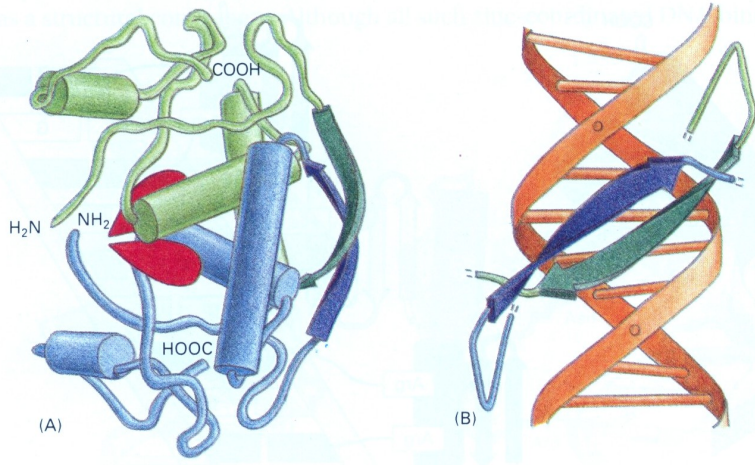


Fuchs et al.,
NSMB 2006

Grundy, Epshtein, Winkler
et al., 1998, 2003

Corbino et al.,
Genome Biol. 2005

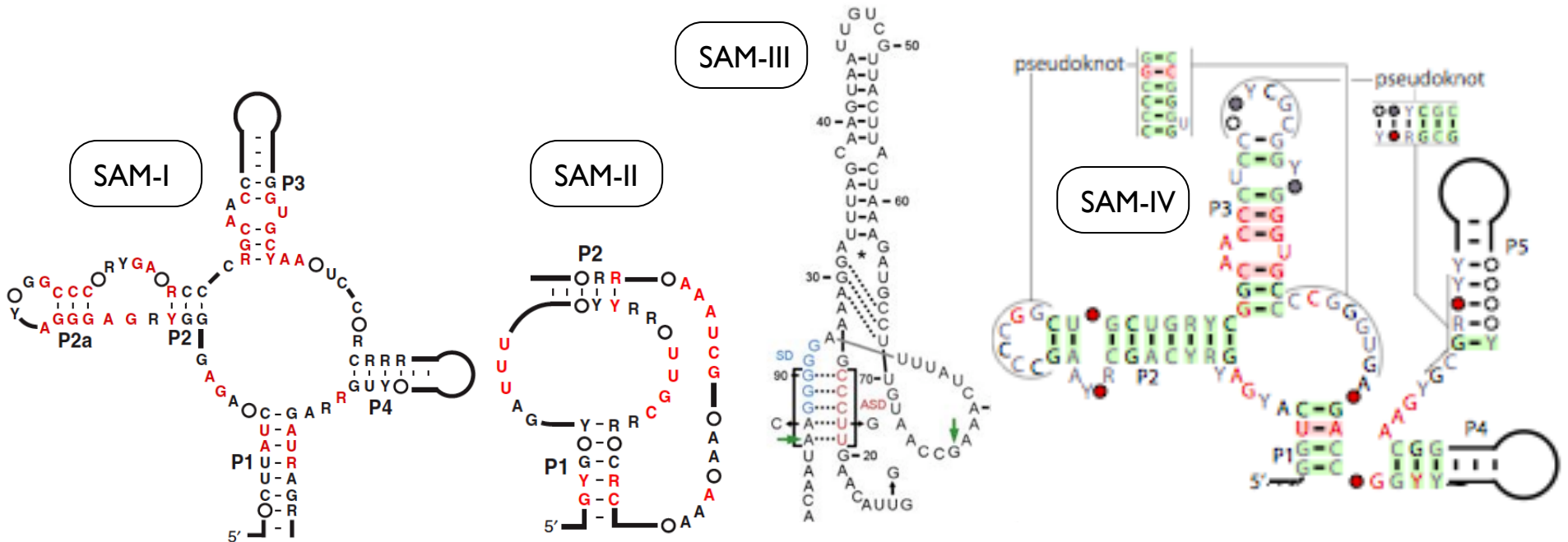
Alberts, et al, 3e.



Not the only way!

Protein way

Riboswitch alternatives



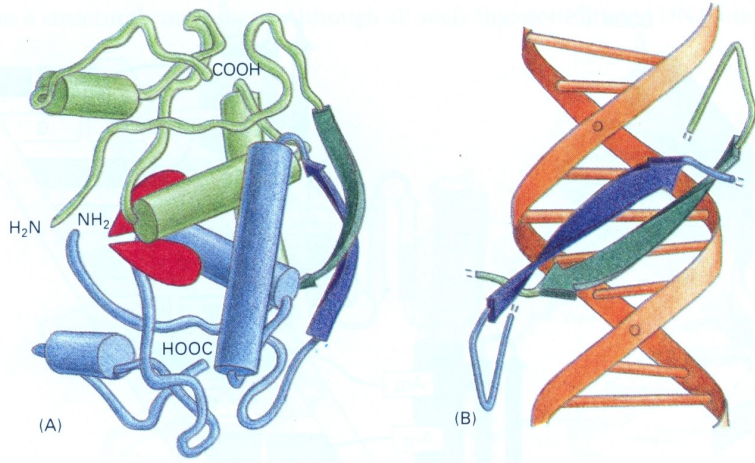
Grundy, Epshtein, Winkler et al., 1998, 2003

Corbino et al., Genome Biol. 2005

Fuchs et al., NSMB 2006

Weinberg et al., RNA 2008

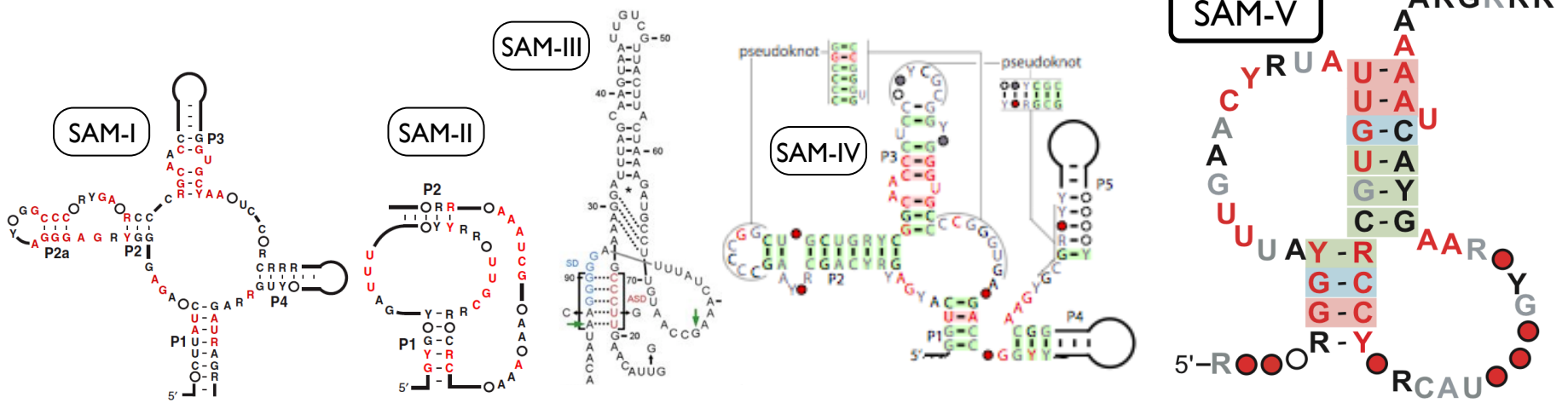
Alberts, et al, 3e.



Not the only way!

Protein way

Riboswitch alternatives



Grundy, Epshtein, Winkler et al., 1998, 2003

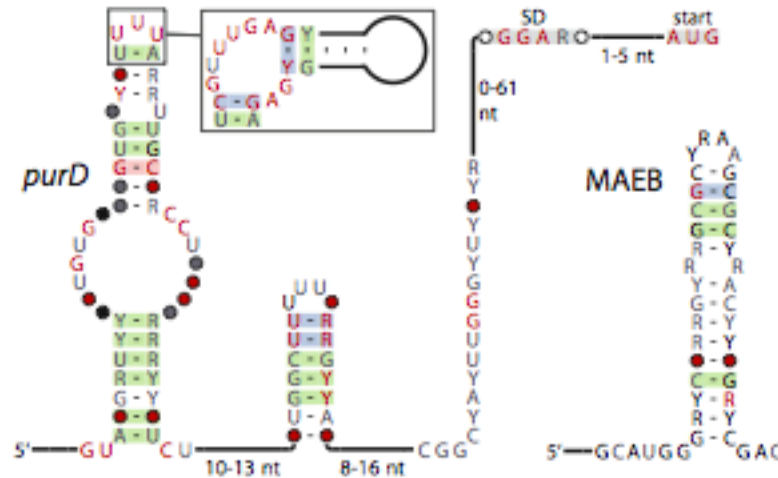
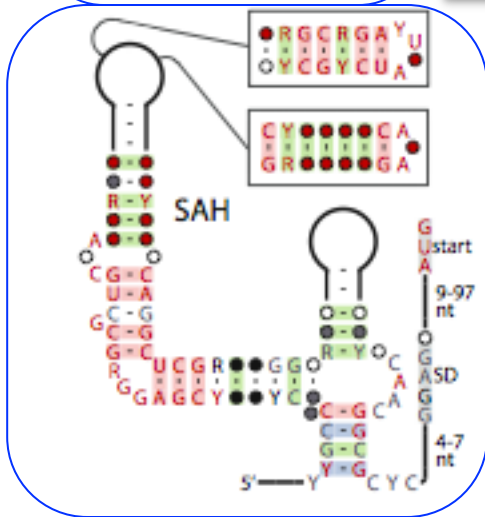
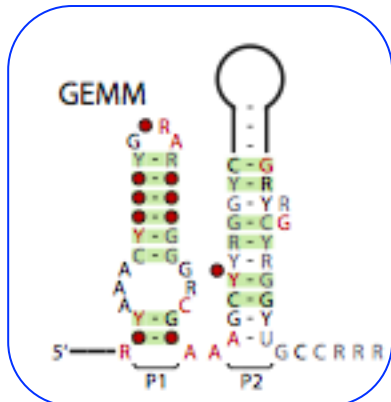
Corbino et al., Genome Biol. 2005

Fuchs et al., NSMB 2006

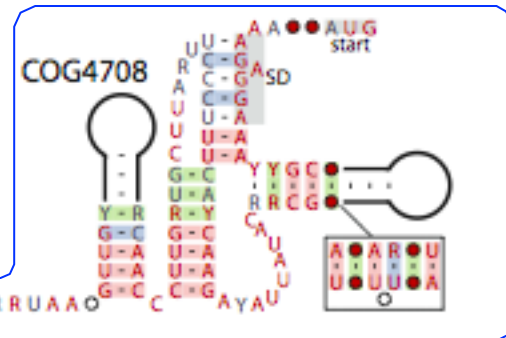
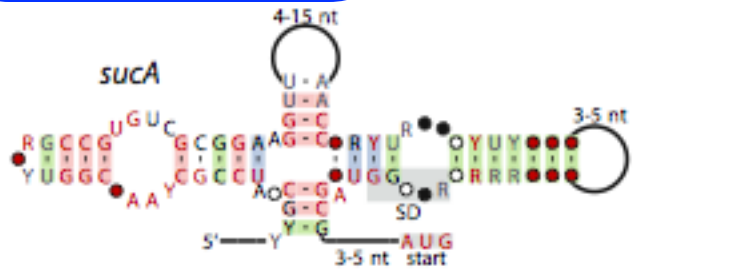
Weinberg et al., RNA 2008

Meyer, et al., BMC Genomics 2009

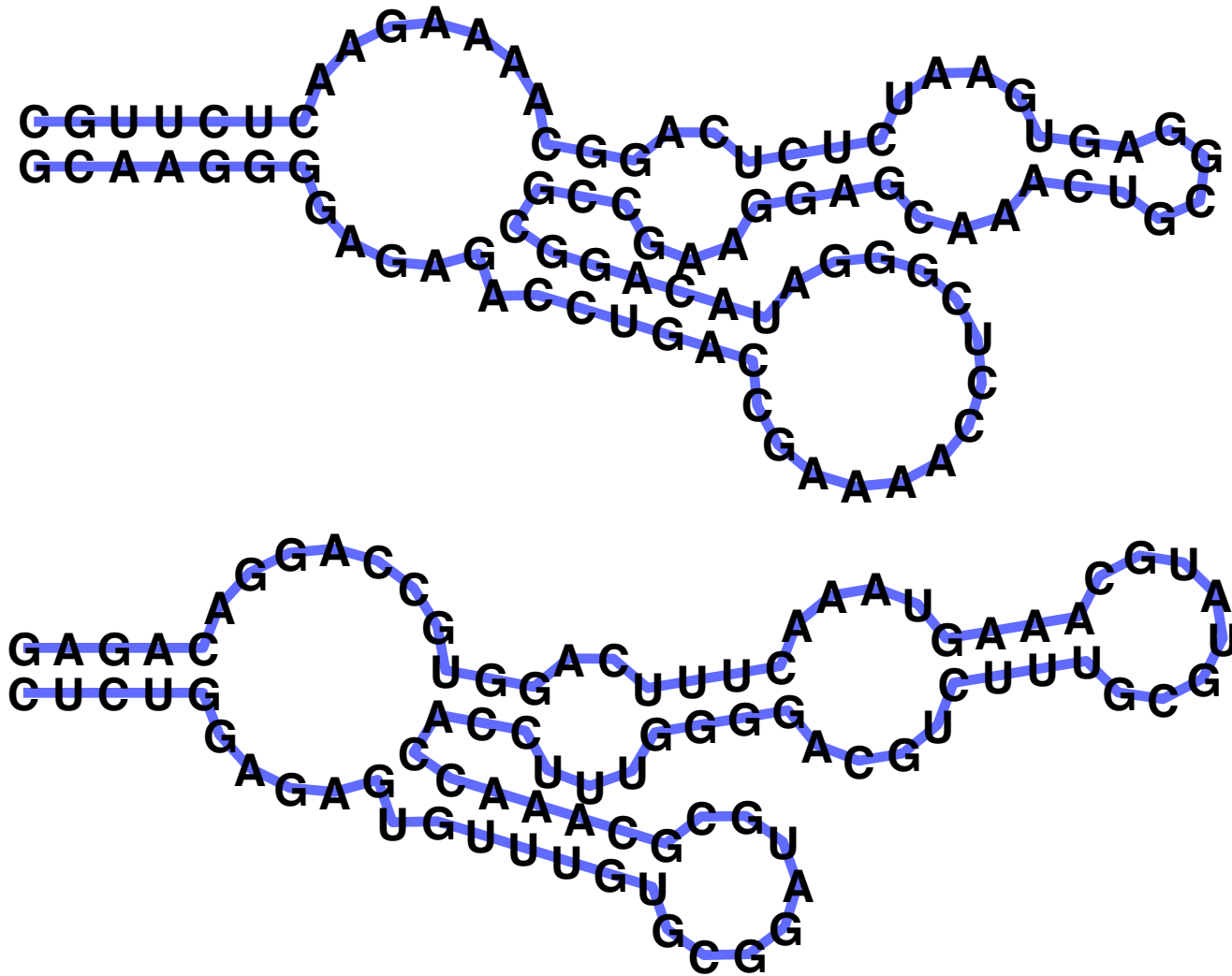
And many other examples. Widespread, deeply conserved, structurally sophisticated, functionally diverse, biologically important uses for ncRNA throughout prokaryotic world.



boxed = confirmed riboswitch (+2 more)



Why is RNA hard to deal with?



A: Structure often more important than sequence³¹

Origin of Life?

Life needs

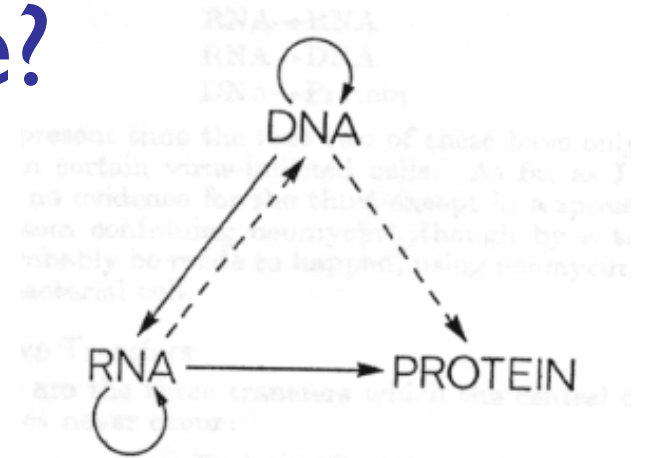
information carrier: DNA

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

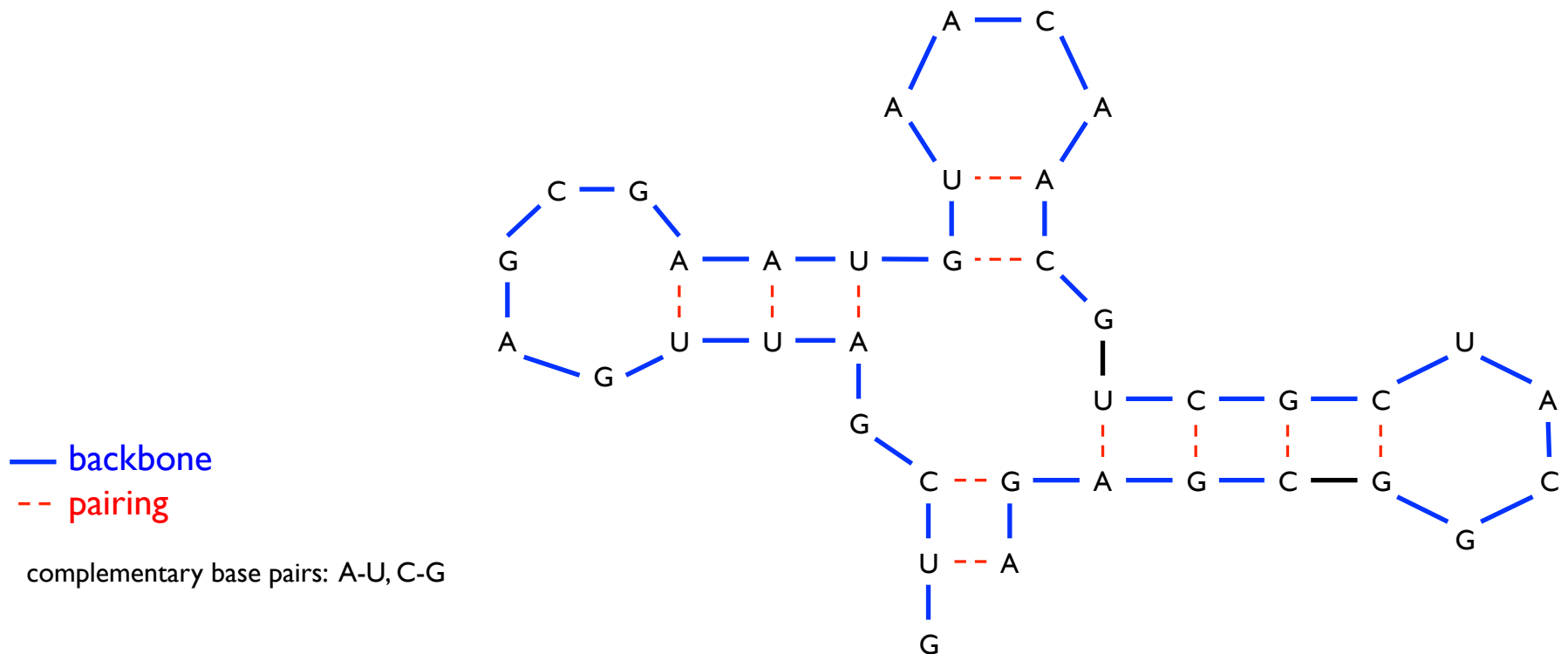
6.5 RNA Secondary Structure

Nussinov's Algorithm – core technology
for RNA structure prediction

RNA Secondary Structure

RNA. String $B = b_1b_2\dots b_n$ 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.



Ex: GUCGAUUGAGCGAAUGUAACAACGUGGCUACGGCGAGA

RNA Secondary Structure (somewhat oversimplified)

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

- [Watson-Crick.]
 - S is a *matching*, i.e. each base pairs with at most one other, 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 $i < k < j < l$. (Violation of this is called a *pseudoknot*.)

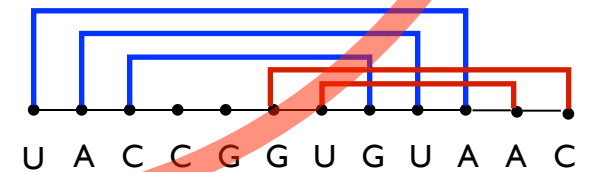
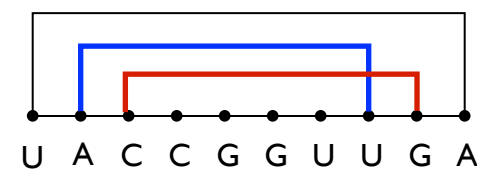
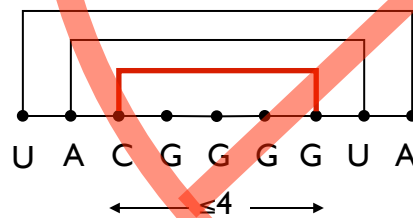
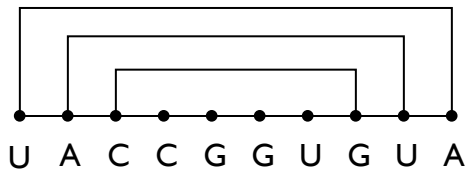
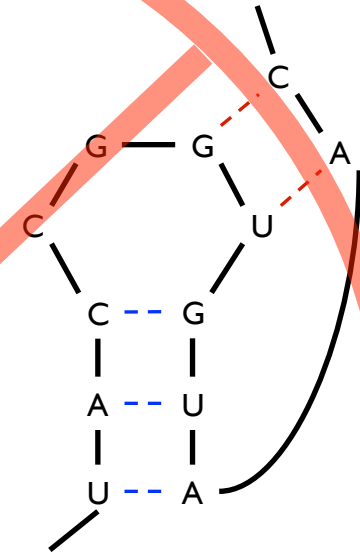
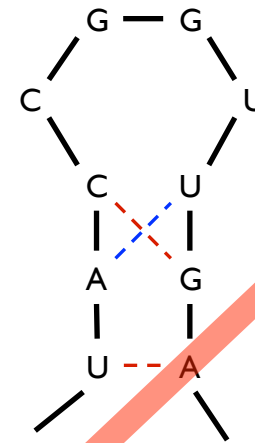
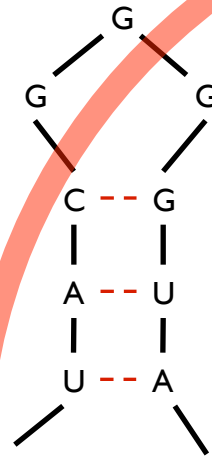
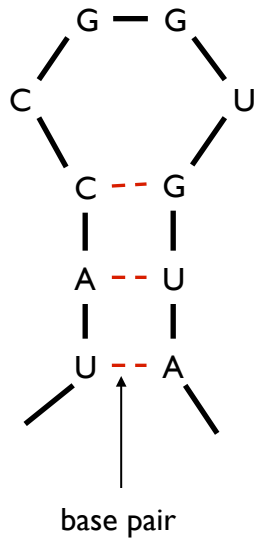
Free energy. Usual hypothesis is that an RNA molecule will form the structure with the optimum total free energy.

← approximated 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.



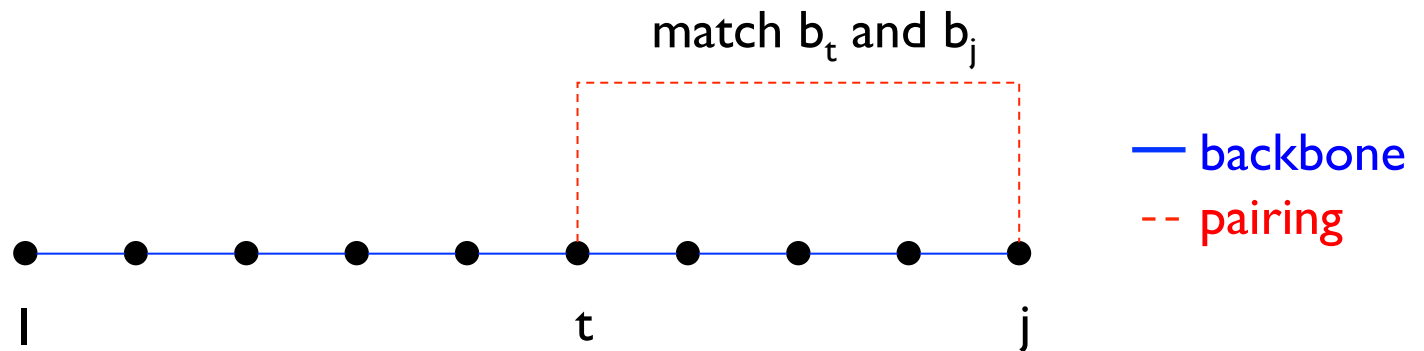
ok

sharp turn

crossing

RNA Secondary Structure: Subproblems

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



Difficulty. Results in two sub-problems.

- Finding secondary structure in: $b_1 b_2 \dots b_{t-1}$. ← $OPT(t-1)$
- Finding secondary structure in: $b_{t+1} b_{t+2} \dots b_{j-1}$. ← not “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 $b_i b_{i+1} \dots b_j$.

- Case 1. If $i \geq j - 4$.

$$OPT[i, j] = 0 \text{ by no-sharp turns condition.}$$

- 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

Key point:
Either last base
is unpaired
(case 1,2) or
paired (case 3)

Remark. Core idea in CKY algorithm for context-free parsing

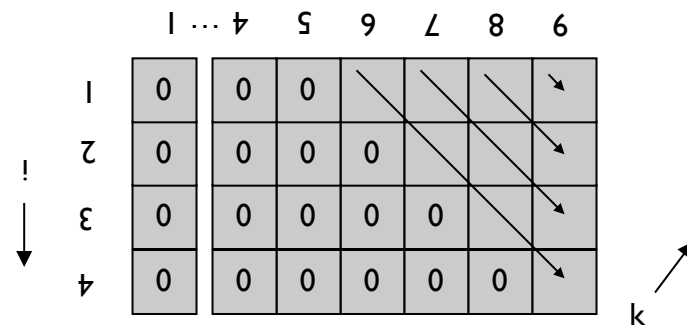
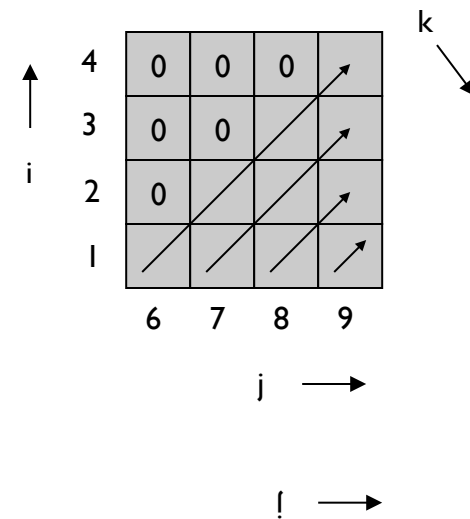
Bottom Up Dynamic Programming Over Intervals

Q. What order to solve the sub-problems?

A. One way—do shortest intervals first:

```

RNA ( $b_1, \dots, b_n$ ) {
Interval length → for  $k = 5, 6, \dots, n-1$ 
Start position →   for  $i = 1, 2, \dots, n-k$ 
End position   →    $j = i + k$ 
                  Compute  $OPT[i, j]$ 
                  using recurrence
                  return  $OPT[1, n]$ 
}
    
```



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

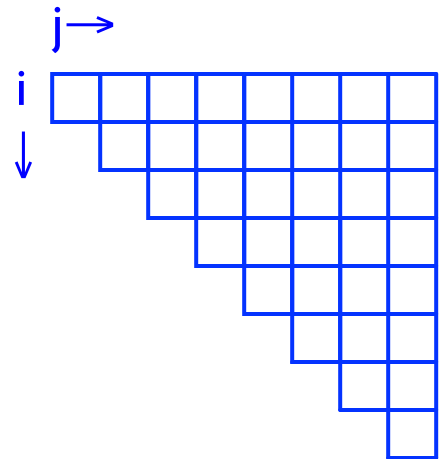
Nussinov: Max Pairing

$\text{opt}(i,j) = \#$ pairs in optimal pairing of $b_i \dots b_j$

$\text{opt}(i,j) = 0$ for all i, j with $i \geq j-4$; otherwise

$\text{opt}(i,j) = \max$ of:

$$\left\{ \begin{array}{l} \text{opt}(i,j-1) \\ \max \{ \text{opt}(i,t-1) + 1 + \text{opt}(t+1,j-1) \mid \\ \quad i \leq t < j-4 \text{ and } b_t - b_j \text{ may pair} \} \end{array} \right.$$

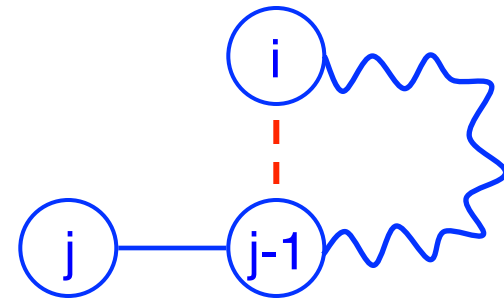


“Optimal pairing of $b_i \dots b_j$ ”

Two possibilities:

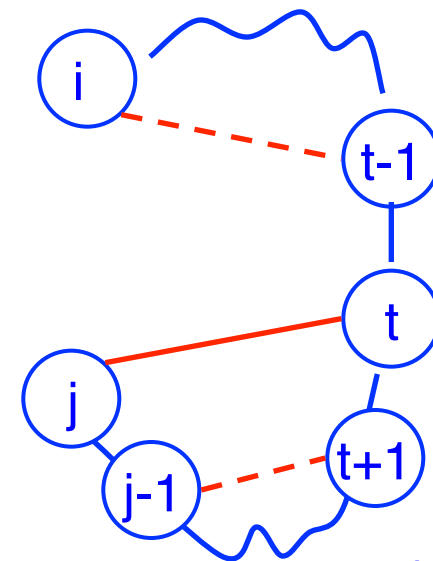
j Unpaired:

Find best pairing of $b_i \dots b_{j-1}$



j Paired (with some t):

Find best $b_i \dots b_{t-1}$ +
best $b_{t+1} \dots b_{j-1}$ **plus 1**



Why is it slow?

Why do pseudoknots matter?

— backbone
— pair
-- pair, maybe?

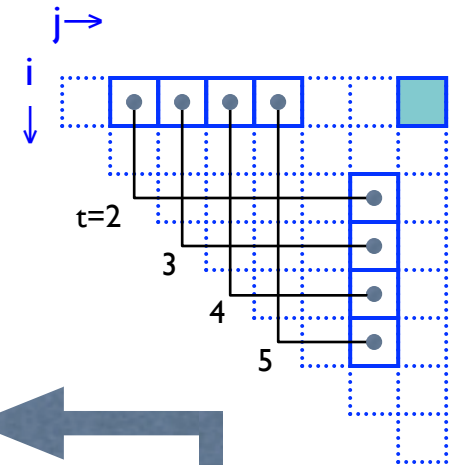
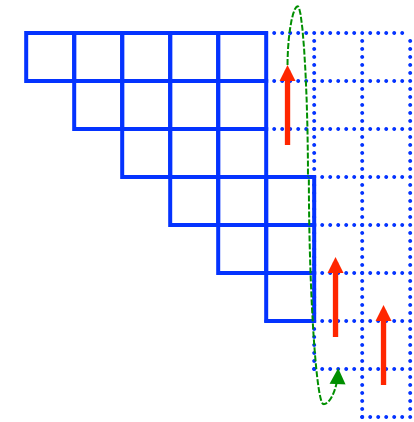
Another Computation Order

$\text{opt}(i,j)$ = # pairs in optimal pairing of $b_i \dots b_j$

$\text{opt}(i,j) = 0$ for all i, j with $i \geq j-4$; otherwise:

$\text{opt}(i,j) = \max$ of:

$$\left\{ \begin{array}{l} \text{opt}(i,j-1) \\ \max \{ \text{opt}(i,t-1) + 1 + \text{opt}(t+1,j-1) \mid \\ \quad i \leq t < j-4 \text{ and } b_t - b_j \text{ may pair} \} \end{array} \right.$$



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

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

$n = 16$

0	0	0	0	0	1	1	1	1	1	2	2	2	3	3	3
	0	0	0	0	0	0	0	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	1	1	1	1	1	2
						0	0	0	0	0	1	1	1	1	1
							0	0	0	0	0	1	1	1	1
								0	0	0	0	0	0	0	1
									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

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

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

(Examples here and below assume 1-based indexing)

Computing one cell: $\text{OPT}[2, 18] = ?$

G	G	G	A	A	A	A	C	C	C	A	A	A	G	G	G	G	U	U	U	n= 20
((.	.	.	.))	((.	.	.	.))					
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	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	

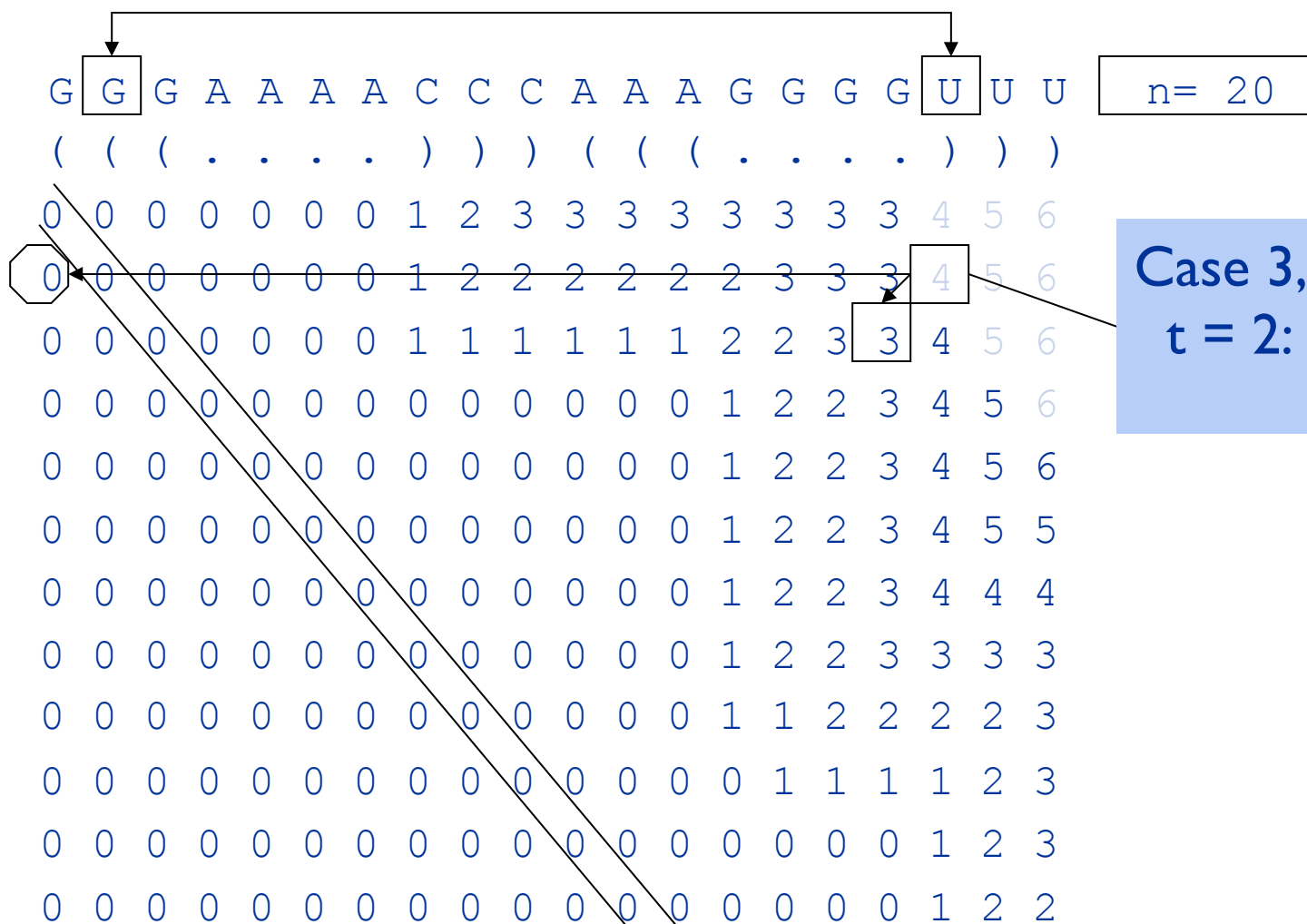
Case 1:
 $2 \geq 18-4?$ no.

Case 2:
 B_{18} unpaired?
 Always a possibility;
 then $\text{OPT}[2, 18] \geq 3$

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

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

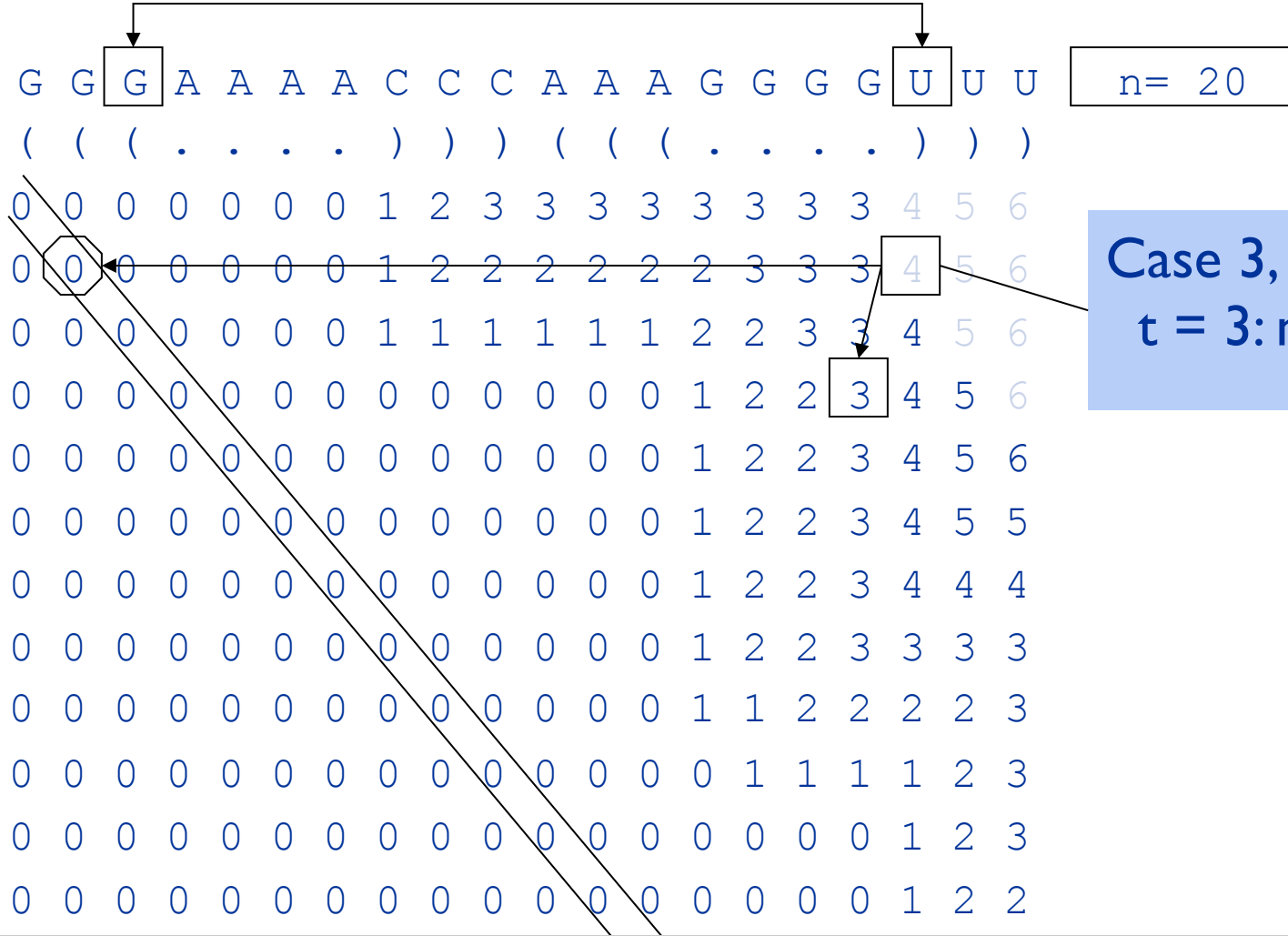
Computing one cell: OPT[2,18] = ?



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

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

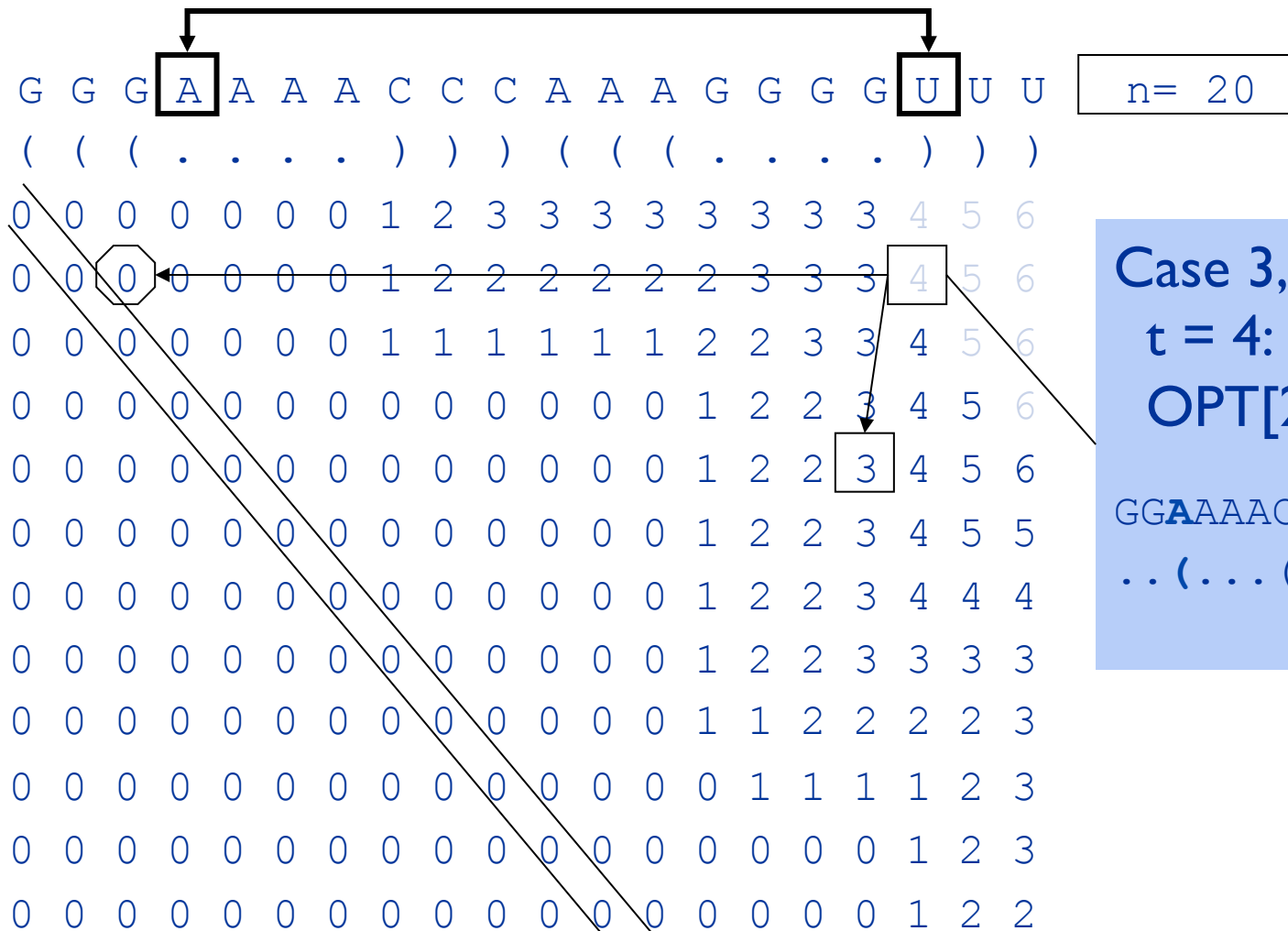
Computing one cell: OPT[2,18] = ?



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

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

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

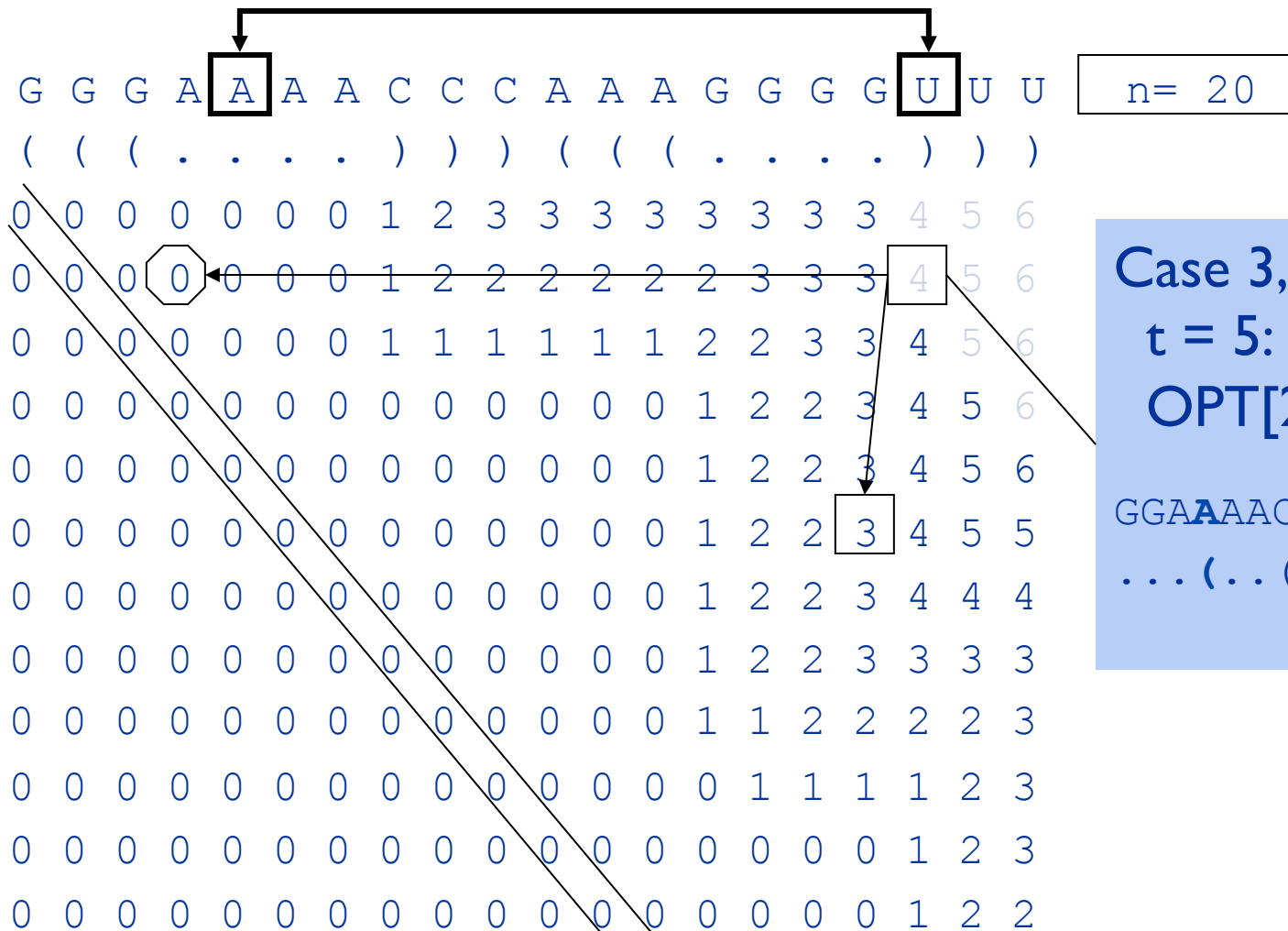


Case 3, $2 \leq t < 18-4$:
 $t = 4$: yes pair
 $OPT[2,18] \geq 1+0+3$

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

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

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

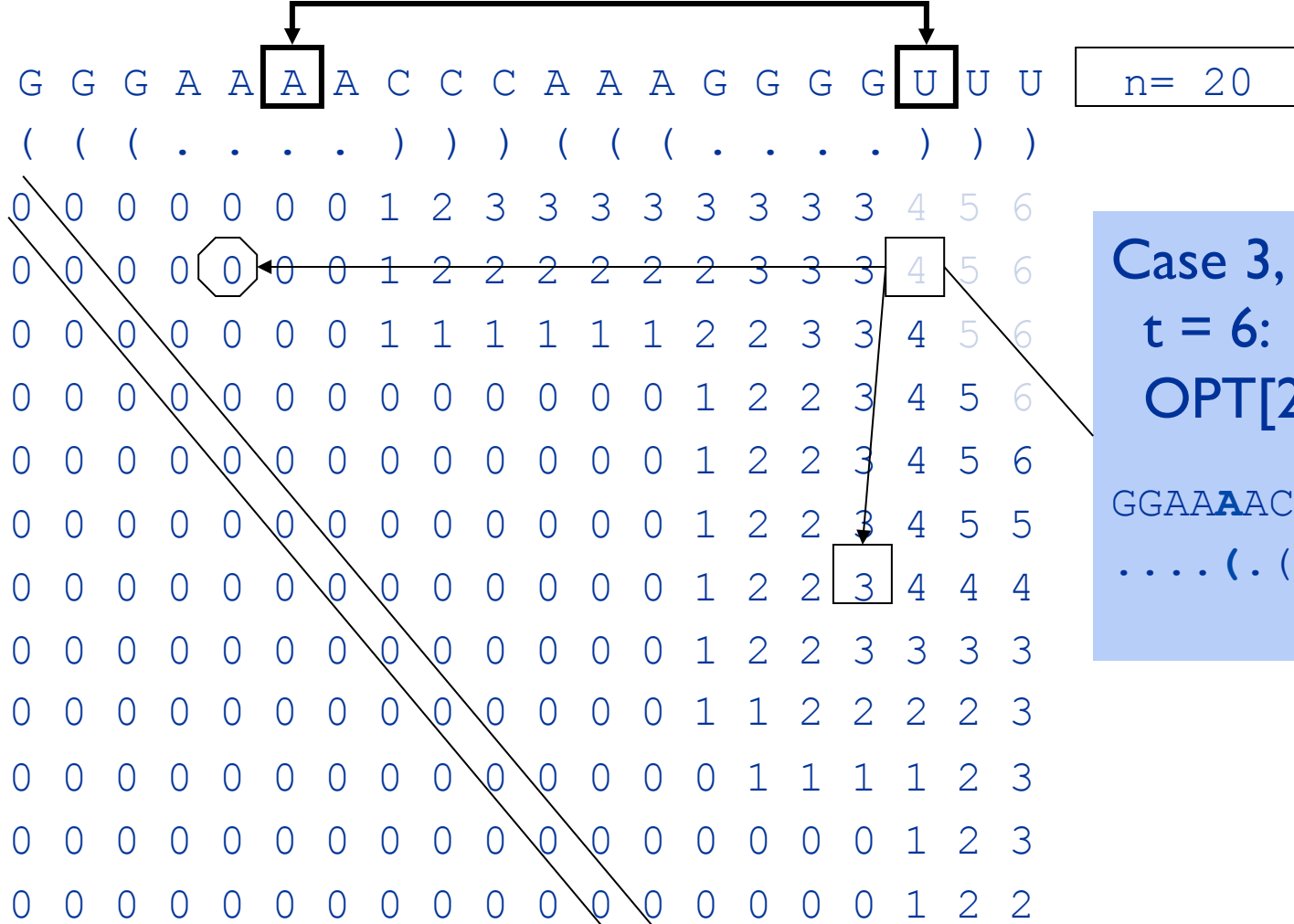


Case 3, $2 \leq t < 18-4$:
 $t = 5$: yes pair
 $OPT[2,18] \geq 1+0+3$

GGAA(A)AACCC(A)AAGGGG(U)
 $\dots (\dots (((\dots))))$

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

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

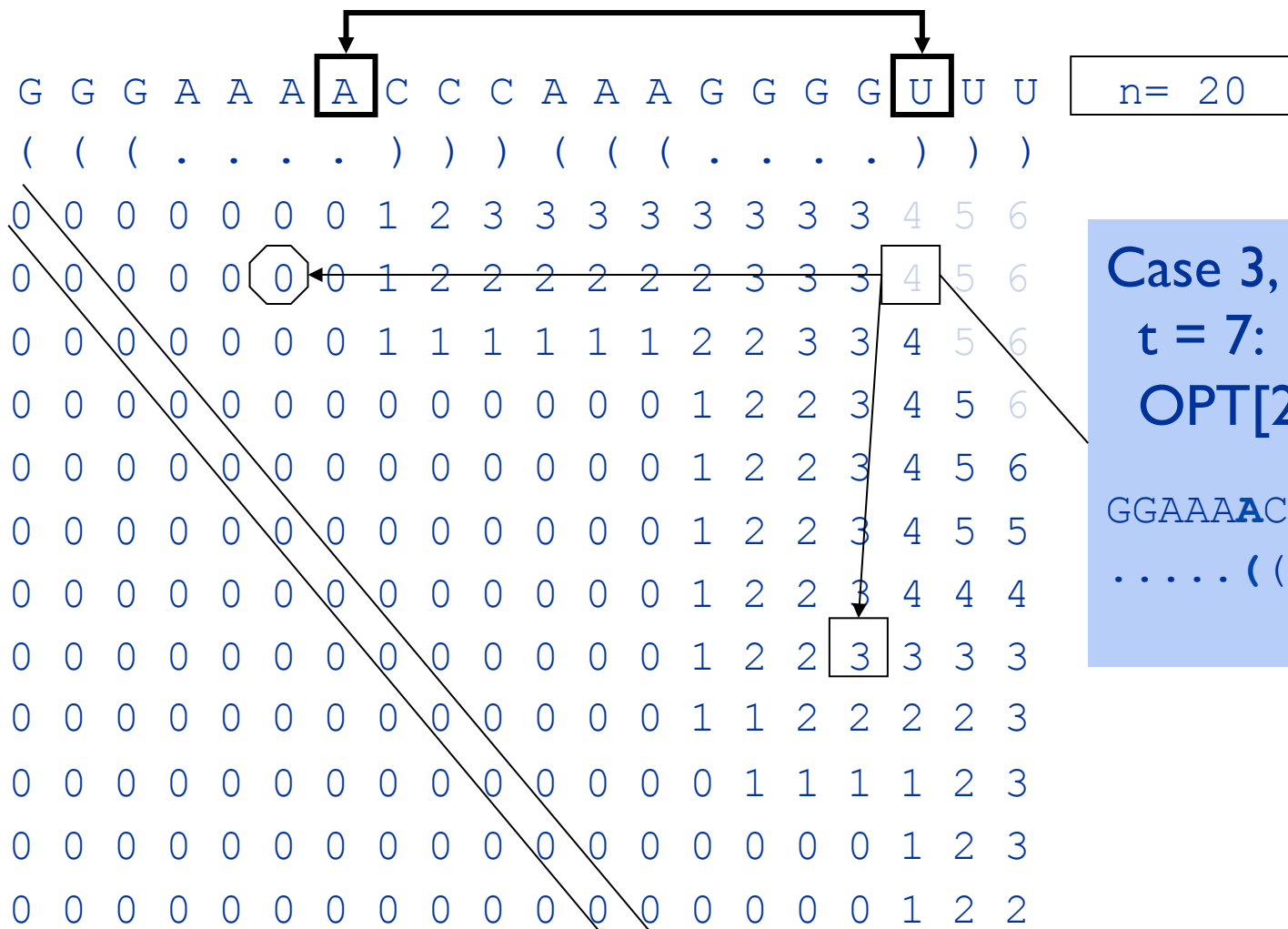


Case 3, $2 \leq t < 18-4$:
 $t = 6$: yes pair
 $OPT[2,18] \geq 1+0+3$

GGAA**A**ACCCAAAGGGGU
 (. (((.....))))

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

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



Case 3, $2 \leq t < 18-4$:
 $t = 7$: yes pair
 $OPT[2,18] \geq 1+0+3$

GGAAA**A**CCCAAAGGGGU
((((.....)))

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

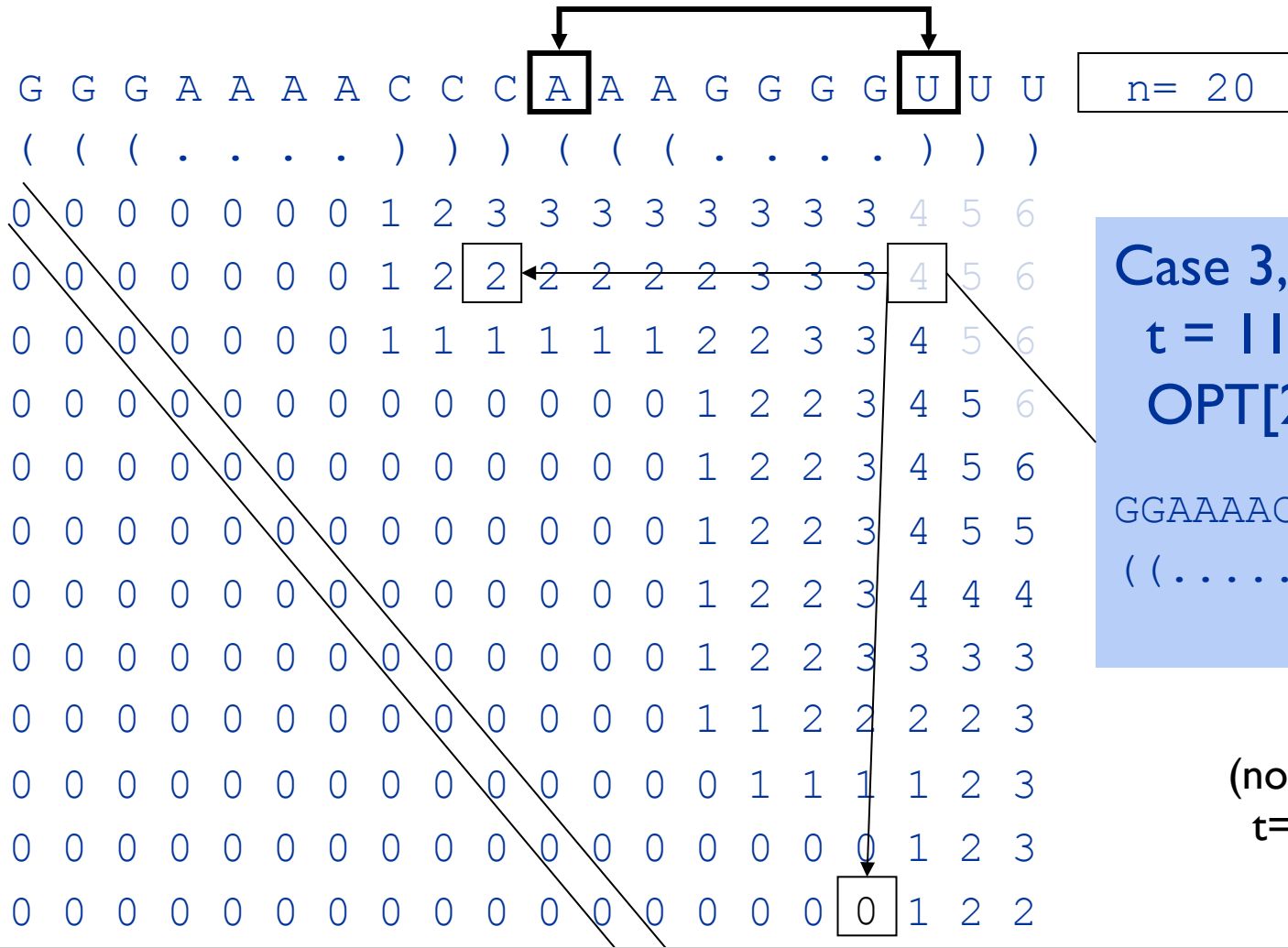
Computing one cell: OPT[2,18] = ?



Case 3, $2 \leq t < 18-4$:
 $t = 8$: no pair

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

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



Case 3, $2 \leq t < 18-4$:
 $t = 11$: yes pair
 $OPT[2,18] \geq 1+2+0$

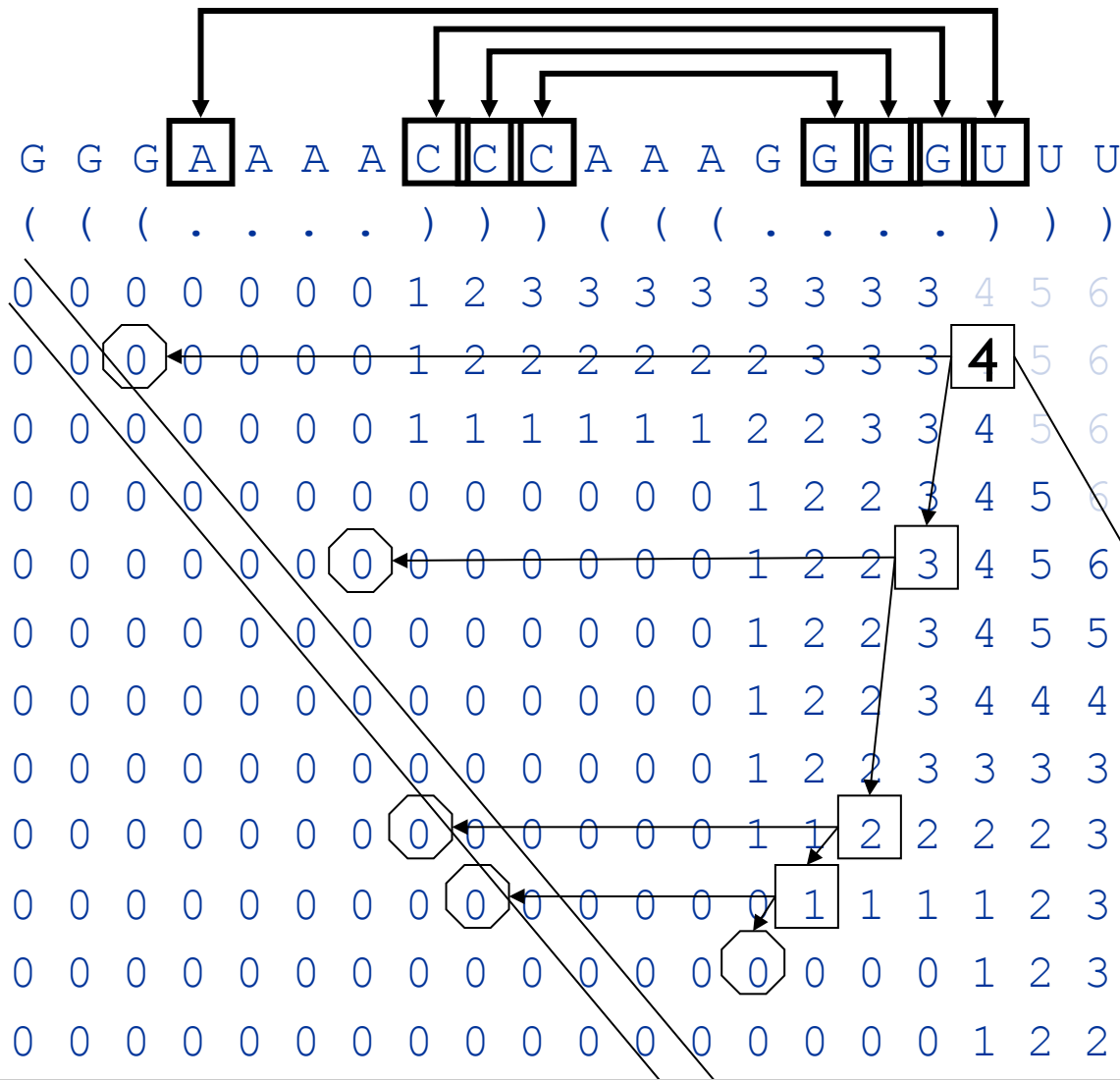
GGAAAACCC**AA**AGGGGU
 ((.....)) (.....)

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

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

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

$n = 20$



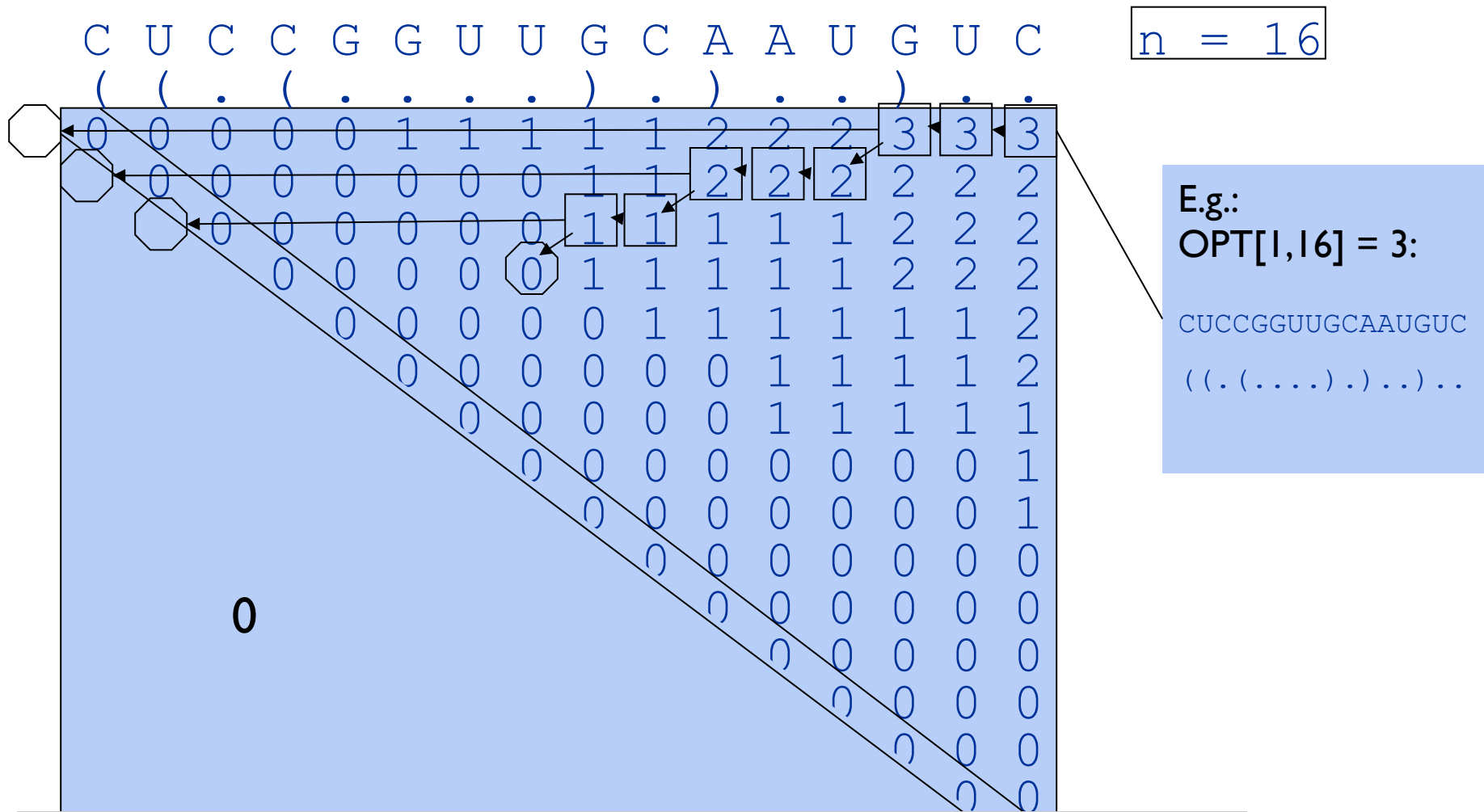
Overall, Max = 4
 several ways, e.g.:

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

tree shows trace back:
 square = case 3
 octagon = case 1

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

Another Trace Back Example



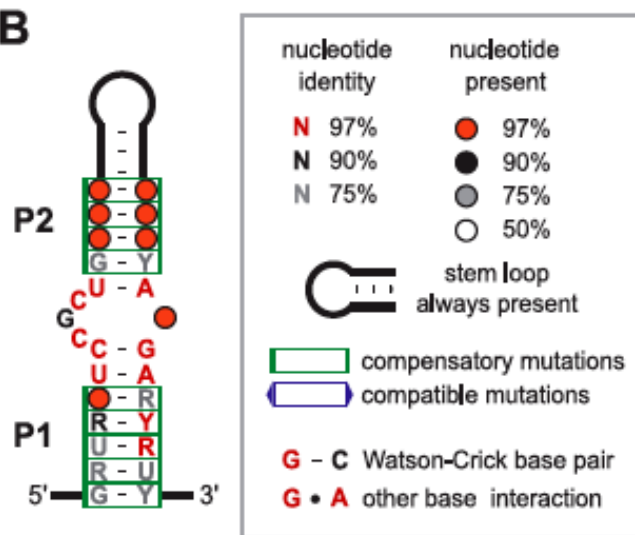
E.g.:
OPT[1,16] = 3:
 CUCCGGUUGCAAUGUC
 ((.(.....).)...)..

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

A L19 (*rplS*) mRNA leader

	-35	-10	TSS	P1	P2	RBS	Start			
<i>Bsu</i>	TTGCAT	TAAGAT	40. AAAAC	GAUGUUC	CGCUG	UGGCC	GUUUUUG... UGGC	CAAGAGCAUCUG	05. AGGAGU	08. AUG
<i>Bha</i>	TTGTTC	TCTTCT	17. AUUAC	GAUGUUC	CGCUG	CAG...	GGUAGAAG... CUGUCAU	GAGCAUCUG	06. AGGAGG	11. AUG
<i>Oih</i>	TTGAAC	TATATT	31. UAAAC	GAUGUUC	CGCUG	UC...	CCAUACUU... GUUCAU	GAGCAUAG	06. AGGAGU	07. AUG
<i>Bce</i>	TTGCTA	TATGCT	36. UUAAC	GAUGUUC	CGCUG	UAA	UUUAUAAGACU... UUA	UAA GAGCAUCUG	05. AGGAGA	09. AUG
<i>Gka</i>	TTGCCA	TATCAT	38. AAAAC	GAUGUUC	CGCUG	CAAUGA	AGAGA... UCAUUGG	CAUGAACUCUG	04. AGGAGU	08. AUG
<i>Bcl</i>	TTGTGC	TATGAT	45. AUUAC	GAUUAUC	CGCUG	CUG...	CAGUGU... UGG	CAUGAAUGUCUG	06. AGGAGG	10. AUG
<i>Bac</i>	ATGACA	GATAGT	35. AUAAC	GAUGUUC	CGCUG	CA	AUAAAAGAAAGUCUG... UG	CAAGAGCAUCUG	05. AGGAGU	08. AUG
<i>Lmo</i>	TTTACA	TAACCT	28. AUAAC	GAUUAUC	CGCUU	CAU...	UAUUAU... AUG	AAUGAAUUGUUG	05. AGGAGA	07. AUG
<i>Sau</i>	TTGAA	TAACCT	23. AUCAC	UAUGAUC	CGCUG	CU	AUAUAUUUGUCG... AGGCA	GAACAUAAGG	04. AGAGGA	09. AUG
<i>Cpe</i>	TTAAAC	TAAACT	08. GUAAC	GGCGU	CUCUG	CACA...	GAG... UGUGU	UAA GAACGUCAA	17. AGGAGG	08. AUG
<i>Chy</i>	TTGCAT	TATAAT	09. UACCAA	ACGUUC	CGCUG	GA...	CAGGGGC... UC	CAUGAACGUGCC	03. AGGAGG	09. AUG
<i>Swo</i>	TTGAGA	TAAAAT	16. AAAAA	GGUGU	CGCUG	CAUU...	AAACUAA... AAUG	UAUGAACACCUU	05. AGGAGG	07. AUG
<i>Ame</i>	TTGCGG	TATAAT	10. UUACG	GGCGU	CUCUA	UAC...	AGGA... GUA	UAA GAACGUCUA	07. AGGAGG	07. AUG
<i>Dre</i>	TTGCC	TATAAT	16. UUACG	GACGU	CGCUG	CCU...	CUGGGAA... AGG	UAA GAACGUCUA	04. AGGAAG	12. GUG
<i>Spn</i>	TTTACT	TAAACT	28. AUACA	GUUAUC	CGCUG	AGGA...	AGAU... UCCU	CAAGAUUGACAA	04. AGGAGA	05. AUG
<i>Smu</i>	TTTACA	TACAAT	26. AAACG	GCUAUC	CGCUG	AG...	ACAGAGCA... CU	UAUGAUUAGUAA	04. AGGAGA	07. AUG
<i>Lpl</i>	TTGCAT	TATTCT	21. UUAAC	GAUGUUC	CGCUG	AC...	CAGGU... GU	CACGAAUGUCGG	04. AGGAAG	09. AUG
<i>Efa</i>	TTTACA	TAAACT	28. AUUAC	AAUAUC	CGCUG	UGG	CA... GAAG... UGACCA	UAA GAAUAUU	06. AGGAGA	08. AUG
<i>Ljo</i>	TTTACA	TAAACT	25. UUAUG	GGUAUC	CGCUG	GCAC...	AAG... GUGUUGAU	GAAUGCCGU	03. AGGAGA	07. AUG
<i>Sth</i>	TAGACA	TAAGAT	29. UAACG	GCUAUC	CGCUG	AGA	CACAGAGGU... UGCUCU	UAA GAUUAUAA	03. AGGAGU	08. AUG
<i>Lac</i>	TTAAA	TACTT	39. UUAUG	GGUAUC	CGCUG	ACG...	CUGGUA... CGUUGAU	GAAUGCCGA	03. AGGAGA	10. AUG
<i>Spy</i>	TTTACA	TAGAAT	29. UUACG	GCUAUC	CGCUA	AG...	ACAAGUA... CU	UAA GAUUAUAA	03. AGGAGA	06. AUG
<i>Lsa</i>	TTTTAA	TAAAAT	26. ACAAC	GAUUAUC	CGCUG	GCG...	CAAGA... CGUUAU	GAAUAUCUG	06. AGGAGA	07. AUG
<i>Lsl</i>	TTTACT	TATTTT	24. AUAAC	GAUUAUC	CGCUU	C...	AACUG... GACAU	GAAUGCCG	04. AGGAAA	07. AUG
<i>Fnu</i>	TTGACA	TAAAAT	12. AAUUC	GAUUAUC	CGCUU	UAA...	UAAA... UUA	AAUGAAUAUCU	04. AGGAAG	02. AUG

B



C

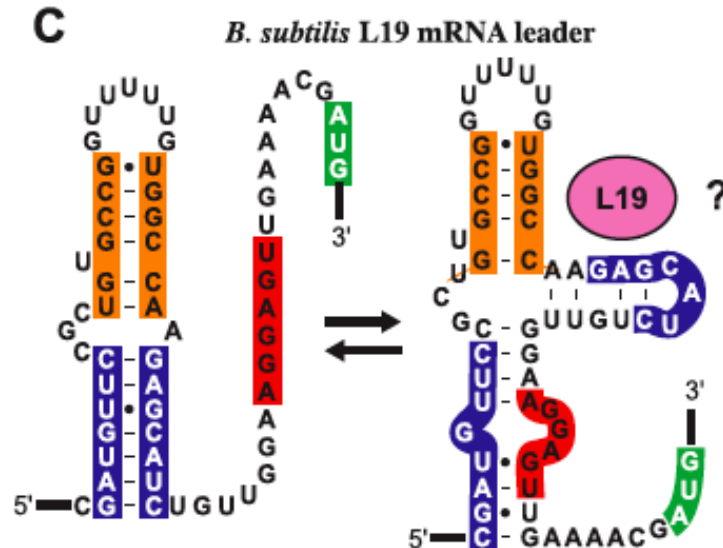


Figure 3. Putative Autoregulatory Structure in L19 mRNA Leaders