

CS 466  
Introduction to Bioinformatics  
Lecture 13

Mohammed El-Kebir

October 9, 2019



# Course Announcements

## Instructor:

- Mohammed El-Kebir (melkebir)
- Office hours: Wednesdays, 3:15-4:15pm

## TA:

- Ramesh Ashwin (aramesh7)
- Office hours: Fridays, 11:00-11:59am in SC 3405

Final on Dec. 16, 8-11am, 2310 Everett Lab

# Outline

- RNA secondary structure

## Reading:

- Topics are not in Jones and Pevzner book but in lecture notes and slides  
[Based on Chapter 10 in “Biological sequence analysis” by Durbin et al.]

# Central Dogma of Molecular Biology

## Three fundamental molecules:

### 1. DNA

Information storage.

### 2. RNA

Old view: Mostly a “messenger”.

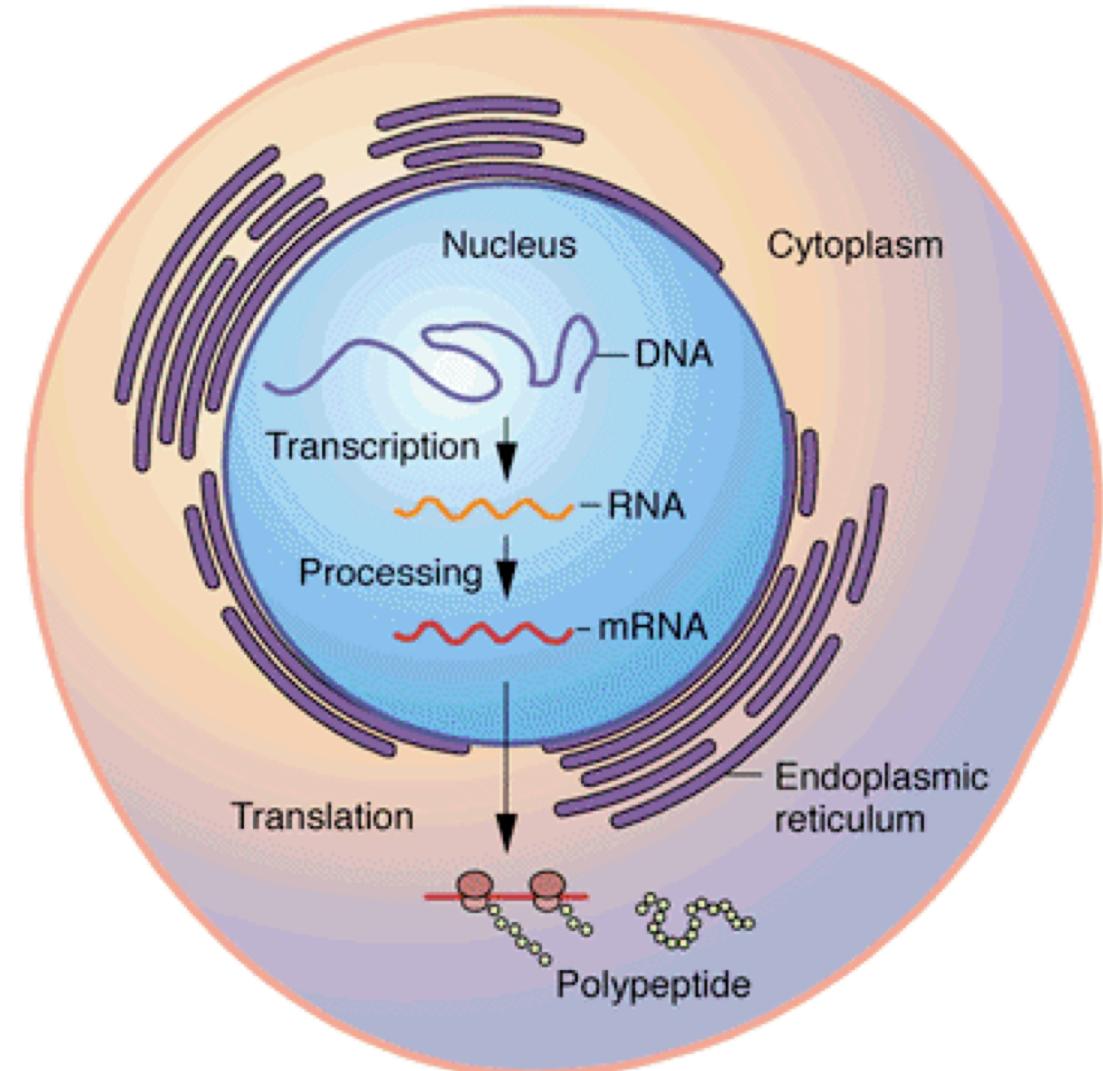
New view: Performs many important functions, through **3-D structure!**

### 3. Protein

Perform most cellular functions  
(biochemistry, signaling, control, etc.)

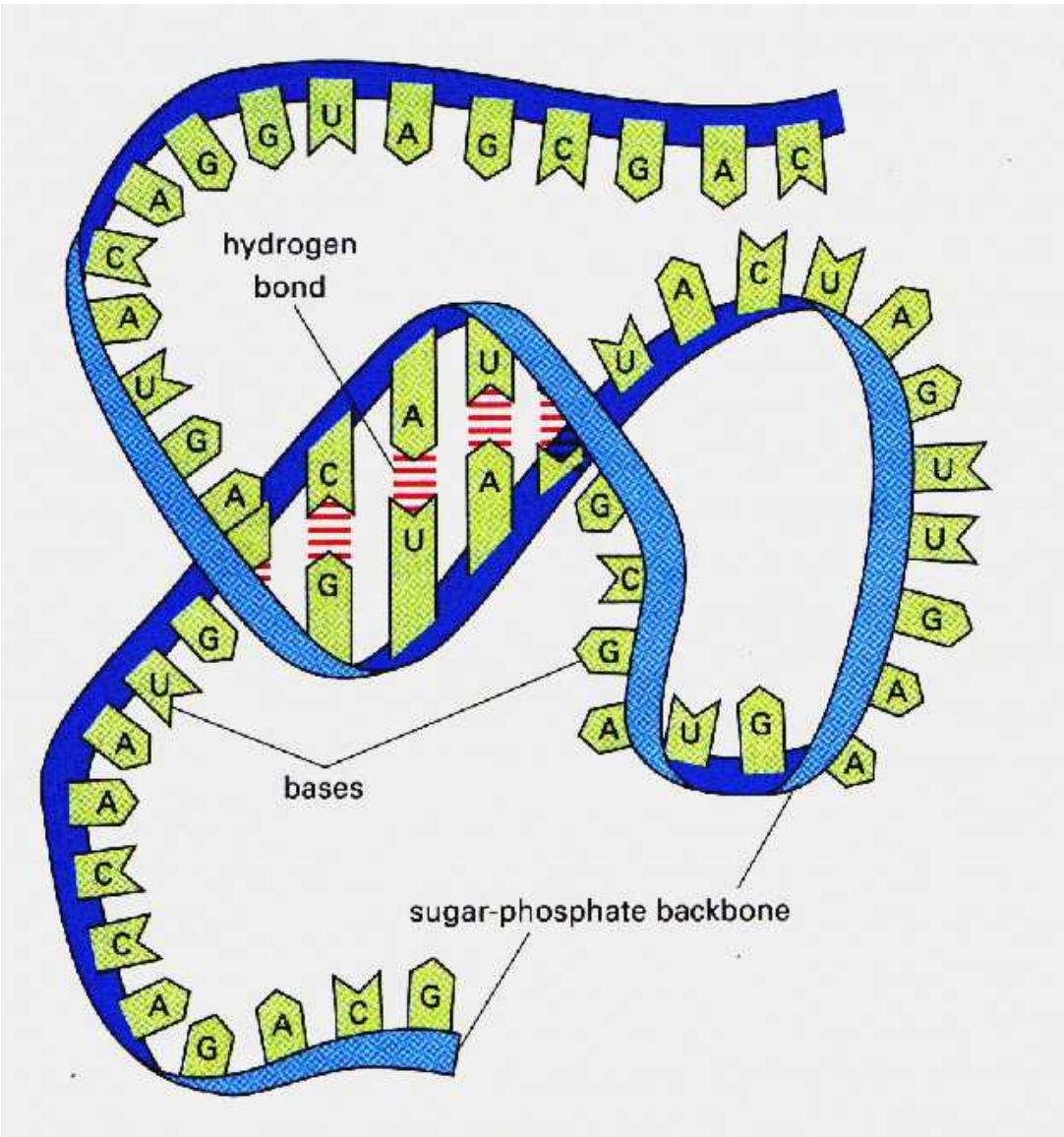
**DNA → RNA → Protein**

*First proposed by Francis Crick in 1956.*



Copyright © 1997, by John Wiley & Sons, Inc. All rights reserved.

# RNA



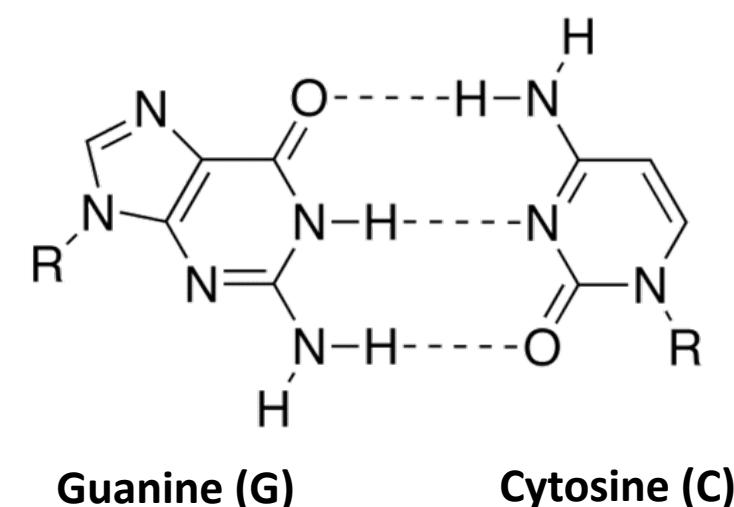
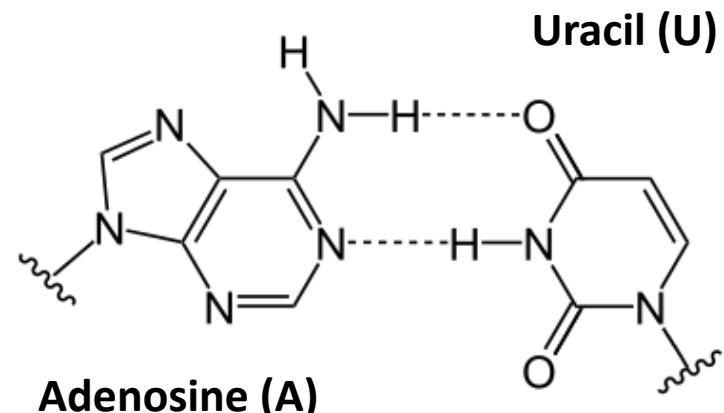
- Single-stranded
  - A (adenine)
  - C (cytosine)
  - U (uracil)
  - G (guanine)
- Can fold into **structures** due to nucleotide complementarity.  
 $A <--> U$ ,  $C <--> G$
- Comes in many flavors:  
mRNA, rRNA, tRNA, tmRNA, snRNA, snoRNA, scaRNA, aRNA, asRNA, piwiRNA, etc.

# RNA – Nucleotide Complementarity

RNA can fold into structures due to nucleotide complementarity:  
 $A \leftrightarrow U$  and  $G \leftrightarrow C$

$A \leftrightarrow U$  (2 hydrogen bonds)  
is slightly weaker than  
 $G \leftrightarrow C$  (3 hydrogen bonds)

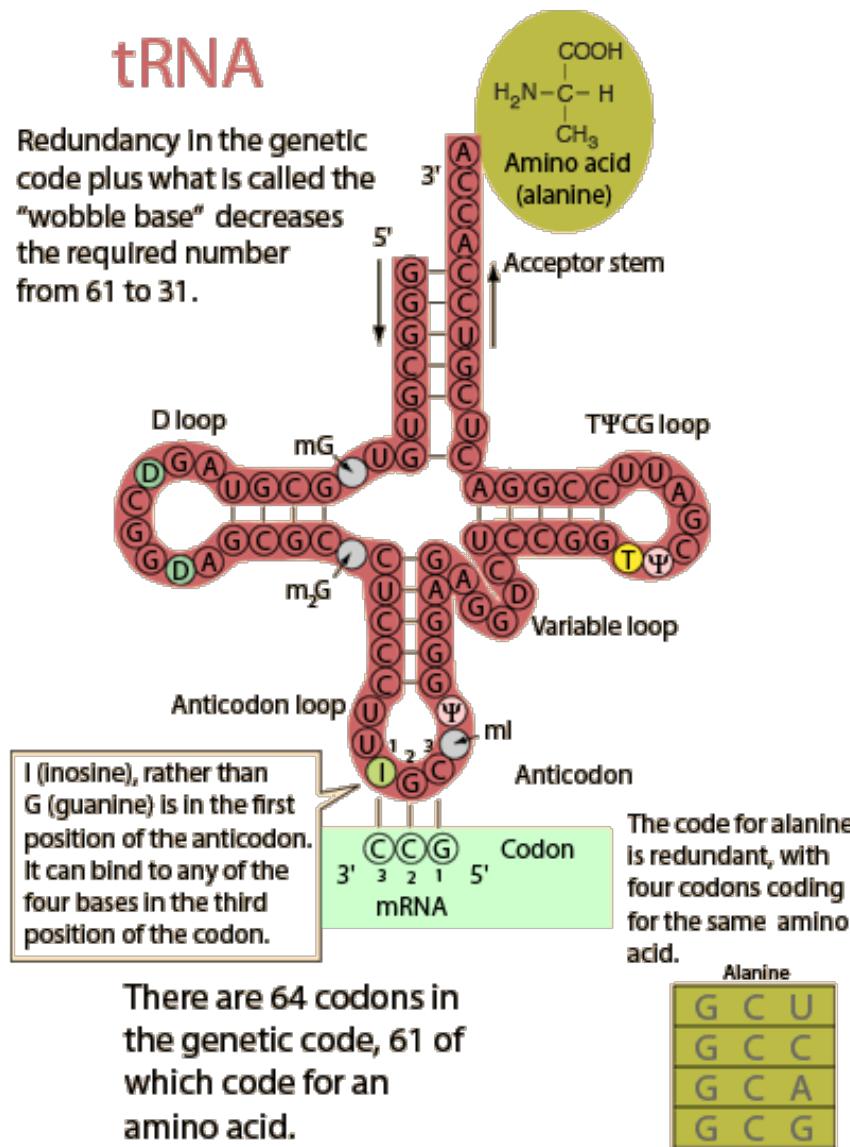
$G \leftrightarrow U$  also observed but not as stable



# transfer RNA (tRNA) Secondary Structure

## tRNA

Redundancy in the genetic code plus what is called the "wobble base" decreases the required number from 61 to 31.



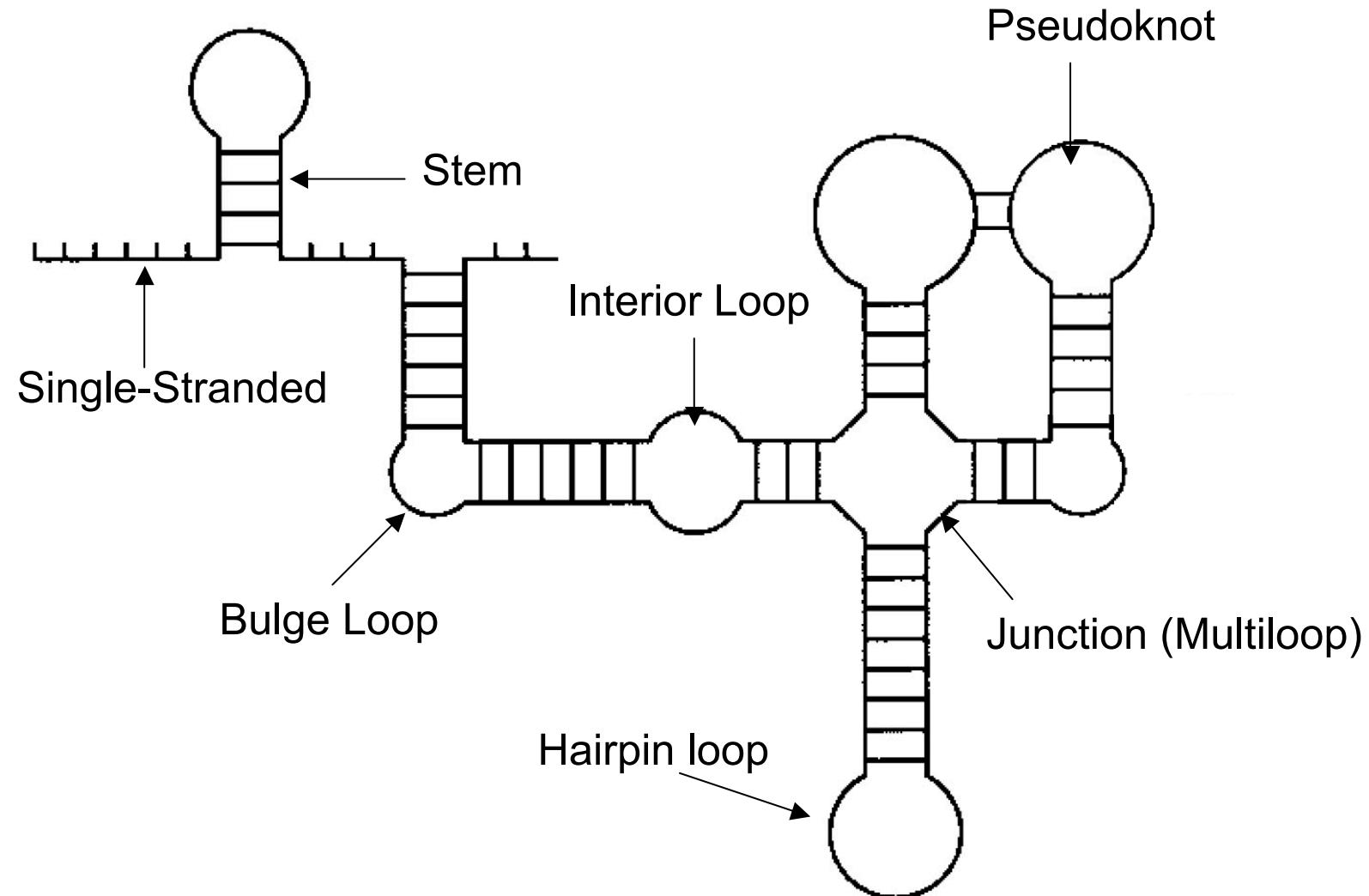
		Second base					
		U	C	A	G		
First base	U	UUU UUC UUA UUG	UCU UCC UCA UCG	UAU UAC	UGU UGC	Cysteine C	
	C	CUU CUC CUA CUG	CCU CCC CCA CCG	CAU CAC CAA CAG	CGU CGC CGA CGG	Stop codon Tryptophan W	
First base	C	CUU CUC CUA CUG	CCU CCC CCA CCG	CAU CAC CAA CAG	CGU CGC CGA CGG	Leucine L Proline P	
	A	AUU AUC AUA AUG M	ACU ACC ACA ACG	AAU AAC AAA AAG	AAU AGC AGA AGG	Histidine H Glutamine Q	
First base	A	AUU AUC AUA AUG M	ACU ACC ACA ACG	AAU AAC AAA AAG	AAU AGC AGA AGG	Isoleucine Methionine start codon	
	G	GUU GUC GUA GUG	GCU GCC GCA GCG	GAU GAC GAA GAG	GAU GAC GAA GAG	Threonine T Lysine K	
		V	A	D	E	Valine V Alanine A	
						Aspartic acid D Glutamic acid E	
						Glycine G	
						U C A G	

<http://bioinfo.bisr.res.in/project/crat/pictures/codon.jpg>

# RNA Secondary Structure Elements

Each base/nucleotide participates in at most one pairing

Secondary structure is determined by a set of non-overlapping base/nucleotide pairs



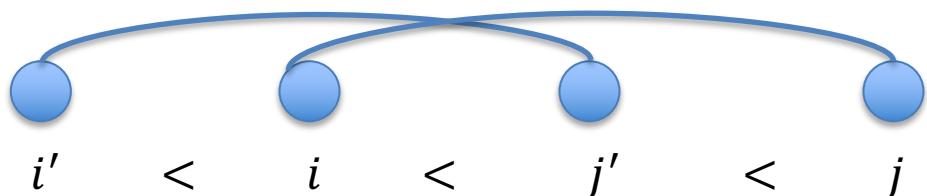
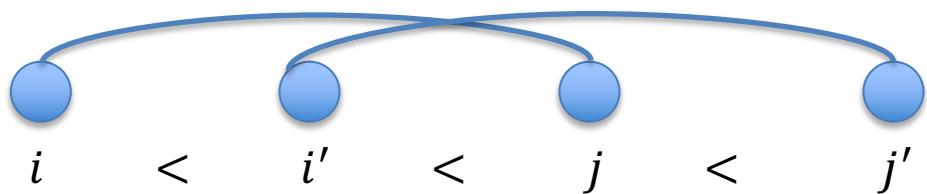
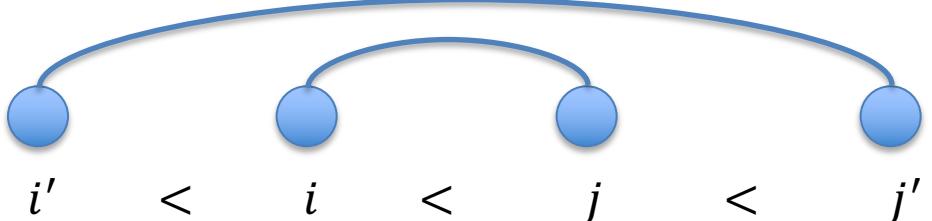
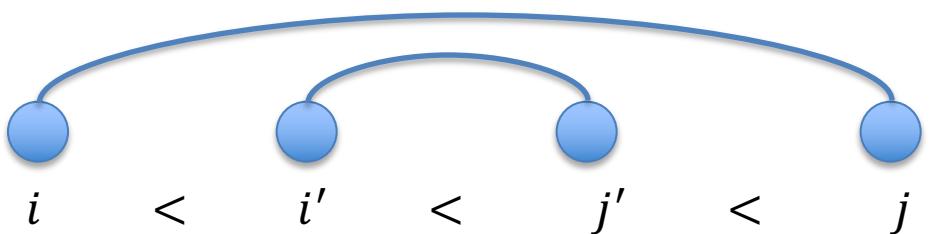
# Nesting and Pseudoknot

Base pairs  $(i, j)$  and  $(i', j')$  are **nested** provided

$$i < i' < j' < j \quad \text{or} \quad i' < i < j < j'$$

Base pairs  $(i, j)$  and  $(i', j')$  form a **pseudoknot** provided

$$i < i' < j < j' \quad \text{or} \quad i' < i < j' < j$$

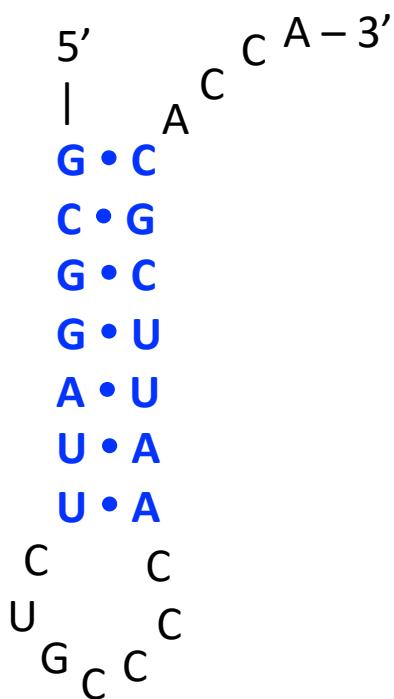


Most RNA molecules consist of nested base pairs

# Nesting and Pseudoknot – Examples

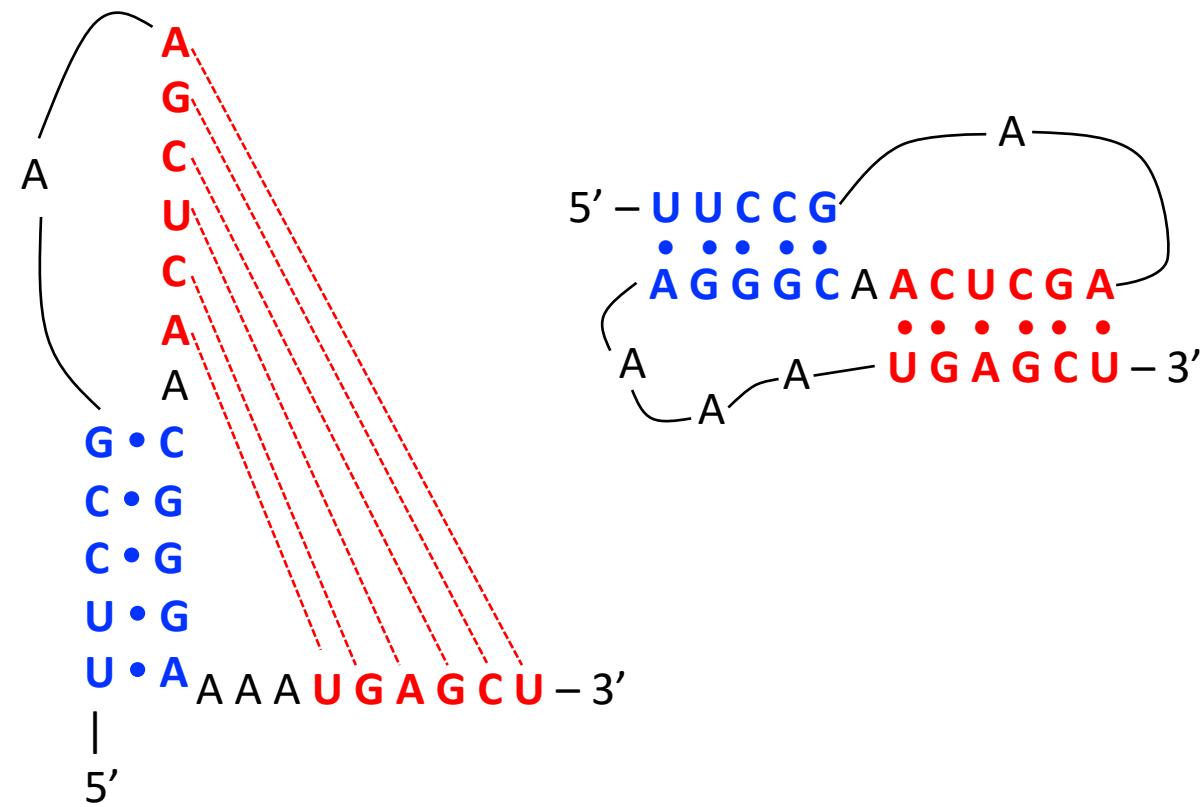
Nesting

5' – GCGGAUUCUGCCCCAA UUCGCACCA – 3'



Pseudoknot

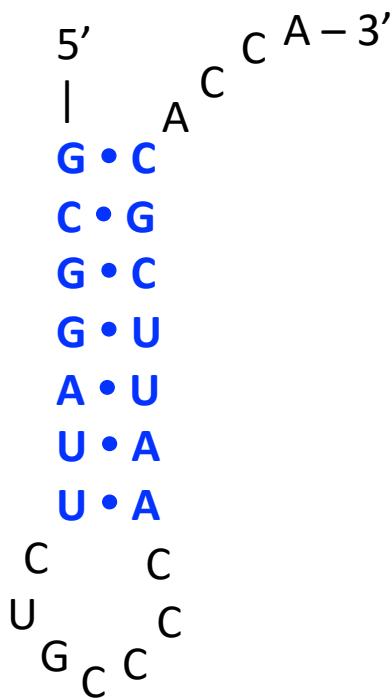
5' – UUCCGAAGCUCAACGGGAAAAUUGAGCU – 3'



# Nesting and Pseudoknot – Examples

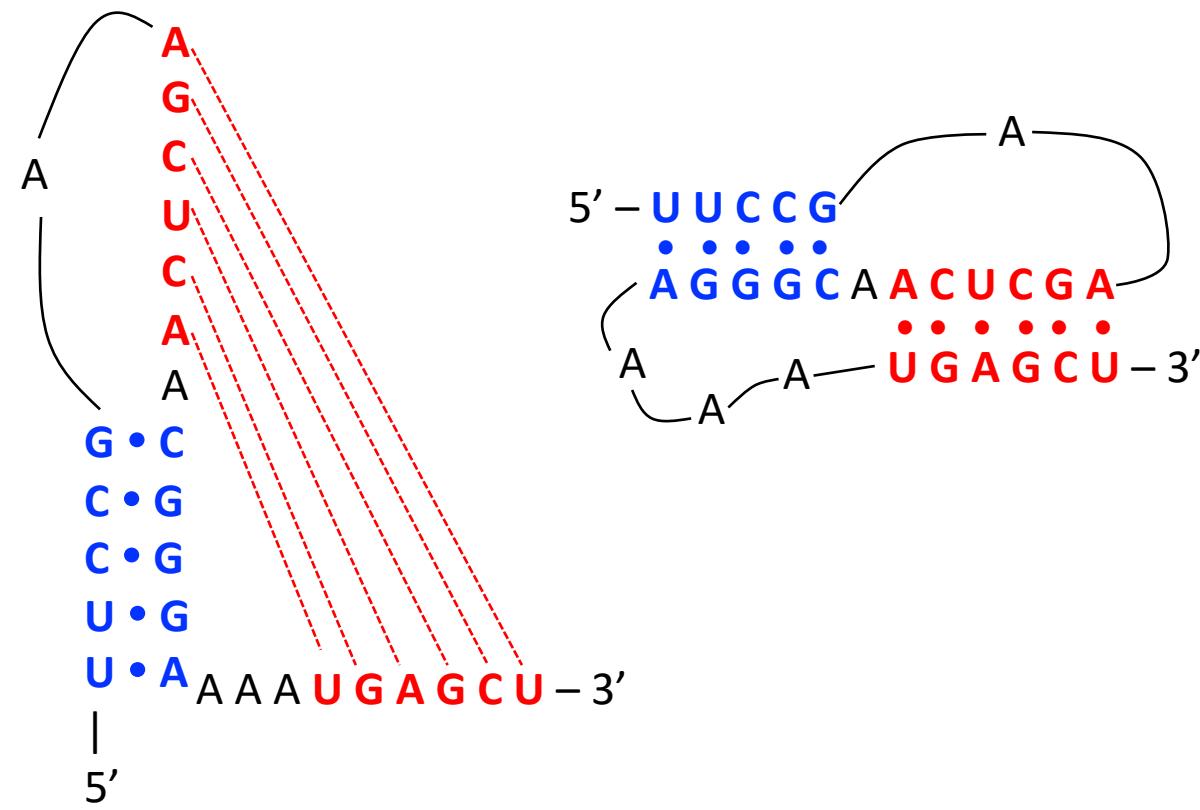
Nesting

5' – GCGGAUUCUGCCCCAA UUCGCACCA – 3'  
(( (( (( ( - - - - - ) ) ) ) ) ) ) - - -



Pseudoknot

5' – UUCCGAAGCUCAACGGGAAAAUUGAGCU – 3'  
(( (( (( - ( ( ( ( ( - ) ) ) ) ) ) ) ) ) ) - - - ) ) ) ) )



# Nussinov Algorithm

RNA can fold into structures due to nucleotide complementarity:  
 $A \leftrightarrow U$  and  $G \leftrightarrow C$

Secondary structure is determined by a set of non-overlapping complimentary base pairs

# Nussinov Algorithm

RNA can fold into structures due to nucleotide complementarity:  
 $A \leftrightarrow U$  and  $G \leftrightarrow C$

Secondary structure is determined by a set of non-overlapping complimentary base pairs

**Question:** How to find maximum number of such pairs?

# Nussinov Algorithm

RNA can fold into structures due to nucleotide complementarity:  
 $A \leftrightarrow U$  and  $G \leftrightarrow C$

Secondary structure is determined by a set of non-overlapping complimentary base pairs

**Question:** How to find maximum number of such pairs?

Need to constrain space of feasible solutions!

# Nussinov Algorithm

RNA can fold into structures due to nucleotide complementarity:  
 $A \leftrightarrow U$  and  $G \leftrightarrow C$

Secondary structure is determined by a set of non-overlapping complimentary base pairs

**Question:** How to find maximum number of such pairs?

Need to constrain space of feasible solutions!

SIAM J. APPL. MATH.  
Vol. 35, No. 1, July 1978

© Society for Industrial and Applied Mathematics  
0036-1399/78/3501-0006 \$01.00/0

## ALGORITHMS FOR LOOP MATCHINGS\*

RUTH NUSSINOV,<sup>†</sup> GEORGE PIECZENIK,<sup>‡</sup> JERROLD R. GRIGGS<sup>¶</sup>  
AND DANIEL J. KLEITMAN<sup>§</sup>



**Problem:** Given RNA sequence  $v \in \{A, U, C, G\}^n$ , find a *pseudoknot-free secondary structure* with the maximum number of complementary base pairings

# Nussinov Algorithm – Dynamic Programming

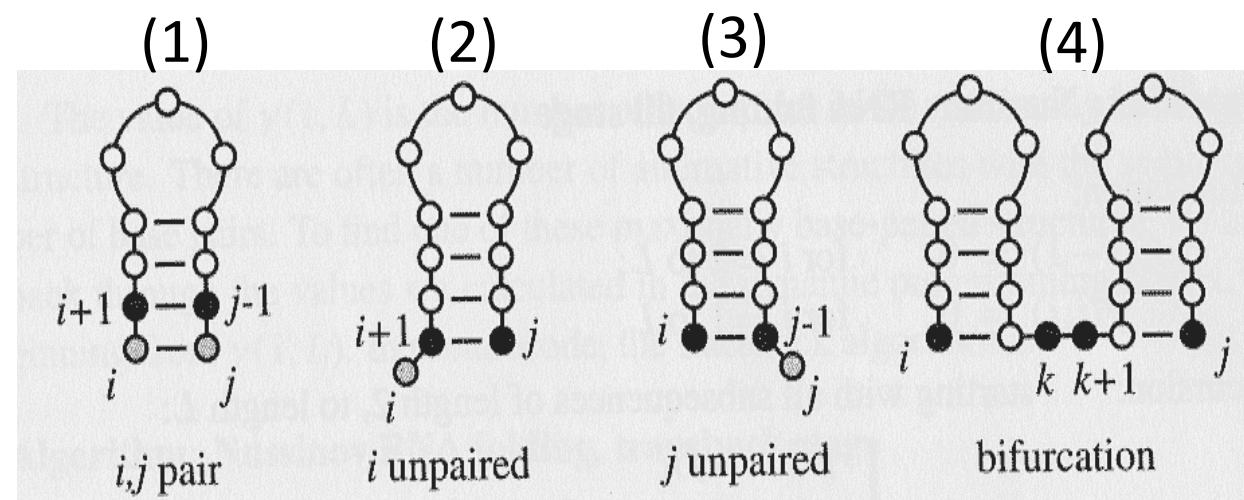
**Problem:** Given RNA sequence  $v \in \{A, U, C, G\}^n$ , find a *pseudoknot-free secondary structure* with the maximum number of complementary base pairings

Let  $s[i, j]$  denote the maximum number of pseudoknot-free complementary base pairings in subsequence  $v_i, \dots, v_j$

# Nussinov Algorithm – Dynamic Programming

**Problem:** Given RNA sequence  $v \in \{A, U, C, G\}^n$ , find a *pseudoknot-free secondary structure* with the maximum number of complementary base pairings

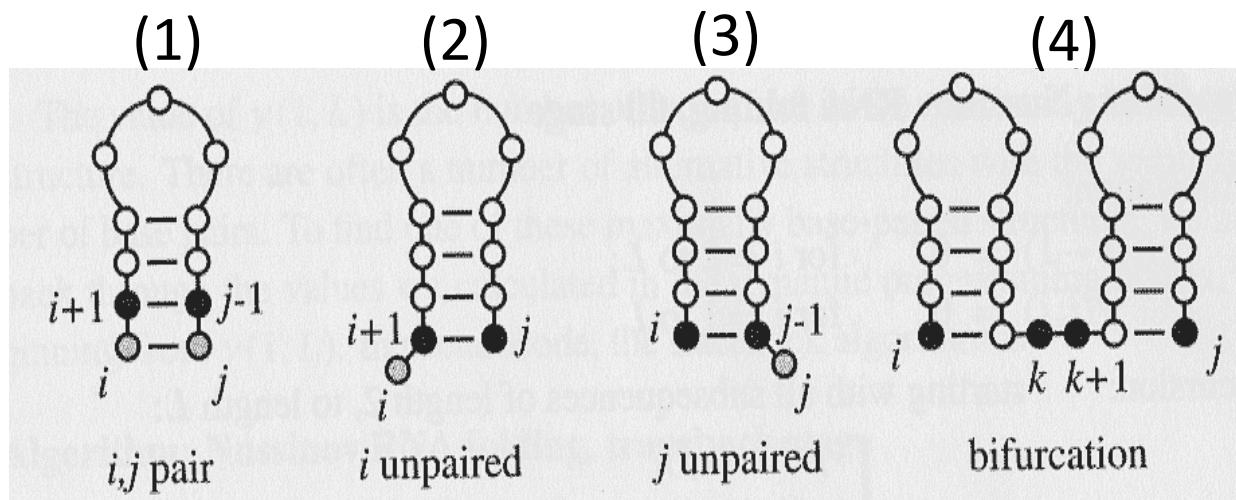
Let  $s[i, j]$  denote the maximum number of pseudoknot-free complementary base pairings in subsequence  $v_i, \dots, v_j$



# Nussinov Algorithm – Dynamic Programming

**Problem:** Given RNA sequence  $v \in \{A, U, C, G\}^n$ , find a *pseudoknot-free secondary structure* with the maximum number of complementary base pairings

Let  $s[i, j]$  denote the maximum number of pseudoknot-free complementary base pairings in subsequence  $v_i, \dots, v_j$



$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \text{ (1)} \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \text{ (1*)} \\ s[i + 1, j], & \text{if } i < j, \text{ (2)} \\ s[i, j - 1], & \text{if } i < j, \text{ (3)} \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \text{ (4)} \end{cases}$$

# Develop Intuition

$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \\ s[i + 1, j], & \text{if } i < j, \\ s[i, j - 1], & \text{if } i < j, \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \end{cases}$$

Sequence: ( ( ( ( ( - - - - - - - - - ) - - ) ) ) )

Index: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

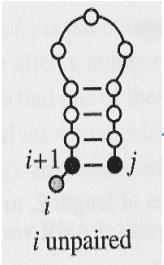
Base: G C U C G G U U C C C U A U U C A A A G A G C

DP Table:

	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
G	1																								
C		2																							
U			3																						
C				4																					
G					5																				
G						6																			
G							7																		
U								8																	
U									9																
C										10															
C											11														
C												12													
U													13												
A														14											
U															15										
U																16									
C																	17								
A																		18							
A																			19						
G																				20					
A																					21				
G																						22			
C																							23		

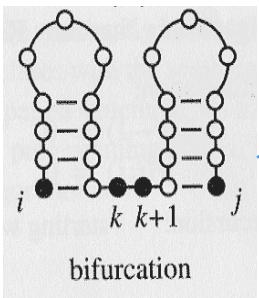
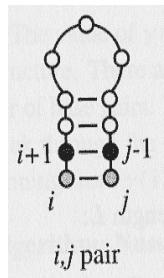
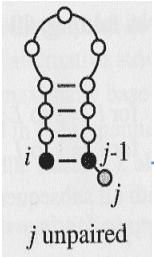
Green arrows indicate the path from the bottom-right cell to the top-left cell, representing the local alignment between the sequence and its reverse complement.

# Nussinov Algorithm – Traceback Step



Push  $(1, n)$  onto stack  
Repeat until stack is empty:

pop  $(i, j)$   
if  $i \geq j$  continue  
else if  $s[i+1, j] = s[i, j]$   
    push  $(i+1, j)$   
else if  $s[i, j-1] = S[i, j]$   
    push  $(i, j-1)$   
else if  $s[i+1, j-1] + 1 = s[i, j]$   
    **record  $(i, j)$  base pair**  
    push  $(i+1, j-1)$   
else for  $k = i+1$  to  $j-1$   
    if  $s[i, k] + s[k+1, j] = s[i, j]$   
        push  $(k+1, j)$   
        push  $(i, k)$   
        break (for loop)



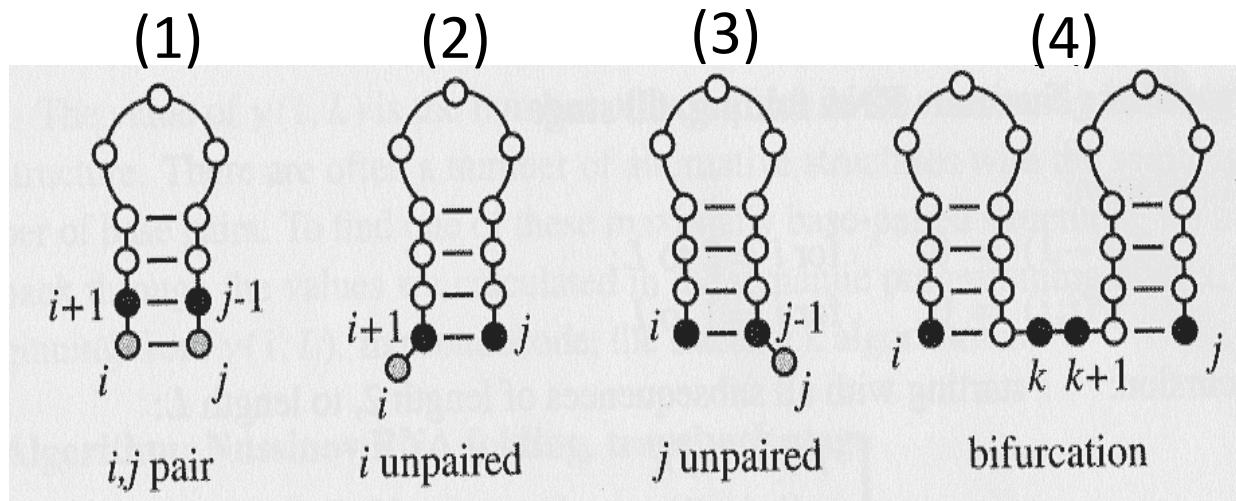
**Question:** Will this return one alignment? Or all alignments?

**Question:** Can we do this recursively?

We only need to know matches.

# Nussinov Algorithm – Example

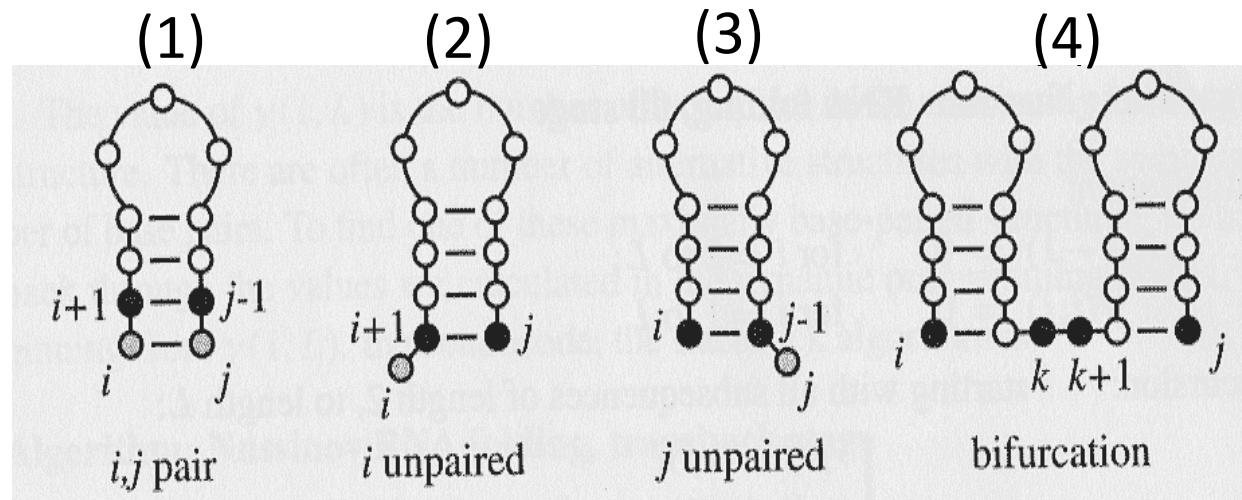
	G	G	G	A	A	A	U	C	C
G	0								
G	0	0							
G	0	0	0						
A	0	0	0	0					
A	0	0	0	0	0				
A	0	0	0	0	0	0			
U	0	0	0	0	0	0	0		
C	0	0	0	0	0	0	0	0	
C	0	0	0	0	0	0	0	0	0



$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \text{ (1)} \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \text{ (1*)} \\ s[i + 1, j], & \text{if } i < j, \text{ (2)} \\ s[i, j - 1], & \text{if } i < j, \text{ (3)} \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \text{ (4)} \end{cases}$$

# Nussinov Algorithm – Example

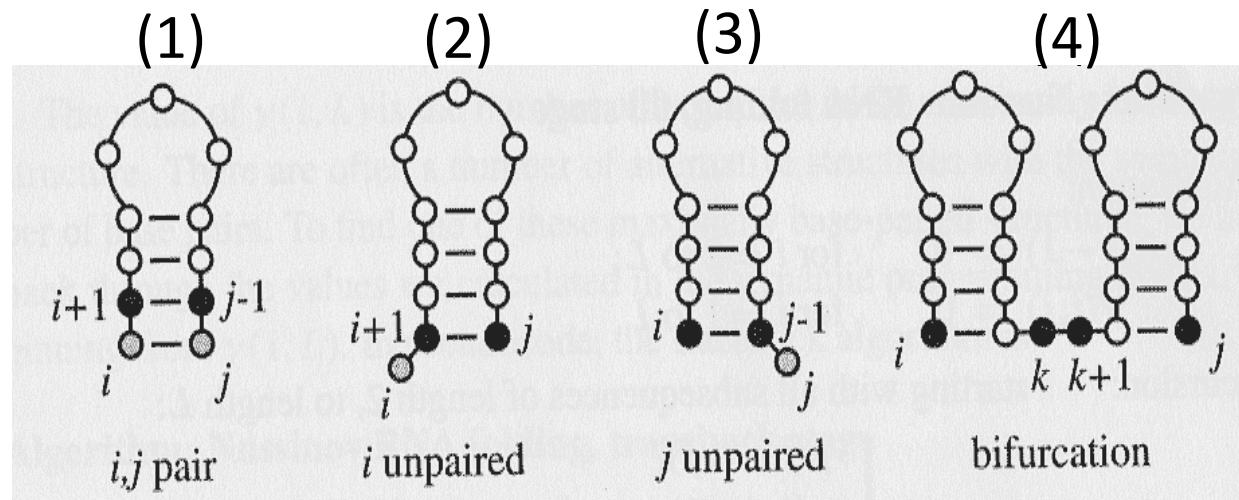
	G	G	G	A	A	A	U	C	C
G	0	0							
G	0	0	0						
G	0	0	0	0					
A	0	0	0	0	0				
A	0	0	0	0	0	0			
A	0	0	0	0	0	0	1		
U	0	0	0	0	0	0	0	0	
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \text{ (1)} \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \text{ (1*)} \\ s[i + 1, j], & \text{if } i < j, \text{ (2)} \\ s[i, j - 1], & \text{if } i < j, \text{ (3)} \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \text{ (4)} \end{cases}$$

# Nussinov Algorithm – Example

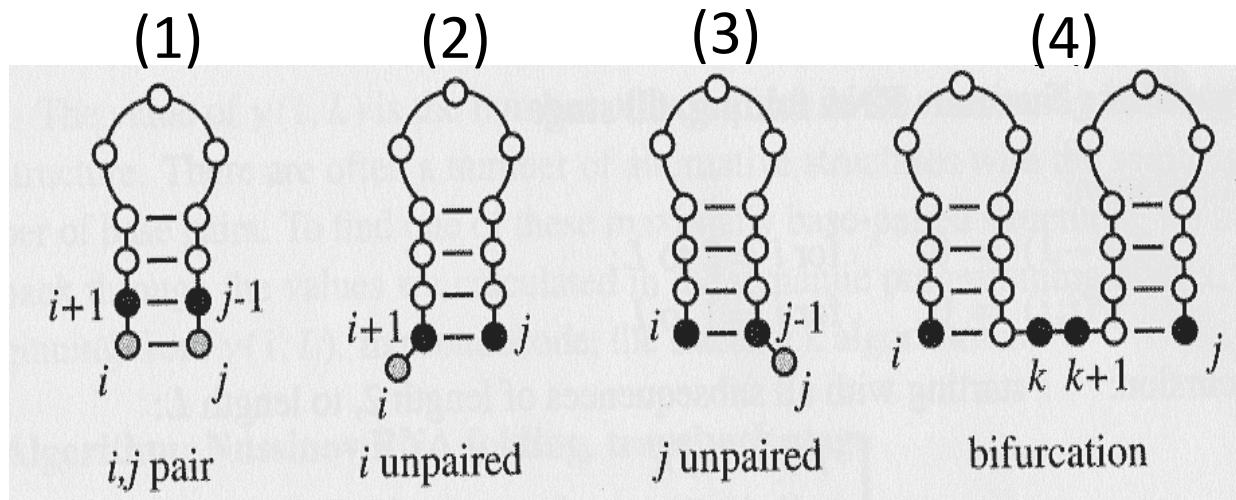
	G	G	G	A	A	A	U	C	C
G	0	0	0						
G	0	0	0	0					
G	0	0	0	0	0				
A	0	0	0	0	0	0			
A	0	0	0	0	0	0	1		
A	0	0	0	0	0	0	1	1	
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \text{ (1)} \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \text{ (1*)} \\ s[i + 1, j], & \text{if } i < j, \text{ (2)} \\ s[i, j - 1], & \text{if } i < j, \text{ (3)} \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \text{ (4)} \end{cases}$$

# Nussinov Algorithm – Example

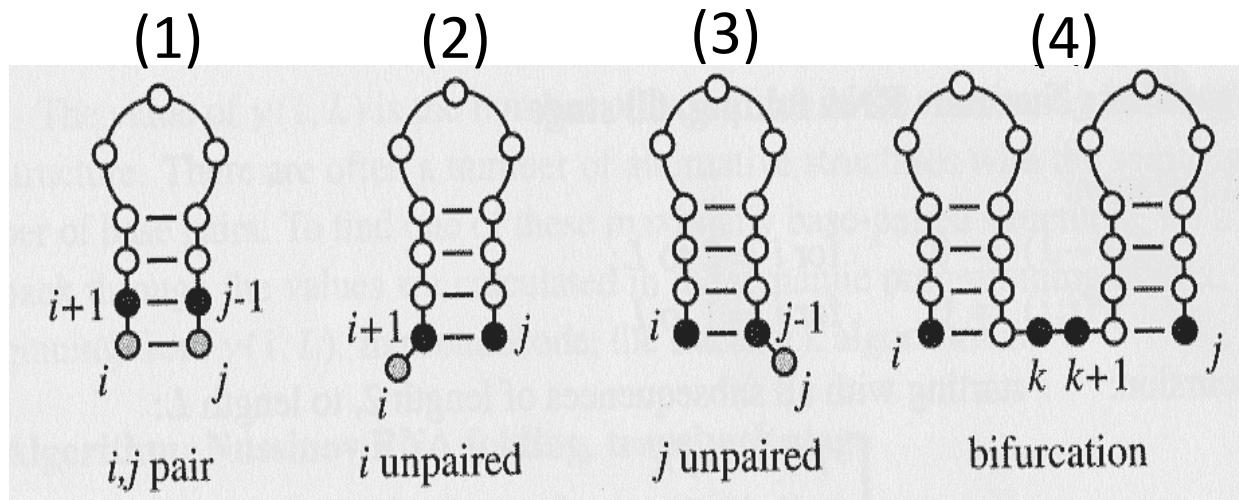
	G	G	G	A	A	A	U	C	C
G	0	0	0	0					
G	0	0	0	0	0				
G	0	0	0	0	0	0			
A	0	0	0	0	0	0	1		
A	0	0	0	0	0	0	1	1	
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \text{ (1)} \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \text{ (1*)} \\ s[i + 1, j], & \text{if } i < j, \text{ (2)} \\ s[i, j - 1], & \text{if } i < j, \text{ (3)} \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \text{ (4)} \end{cases}$$

# Nussinov Algorithm – Example

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0

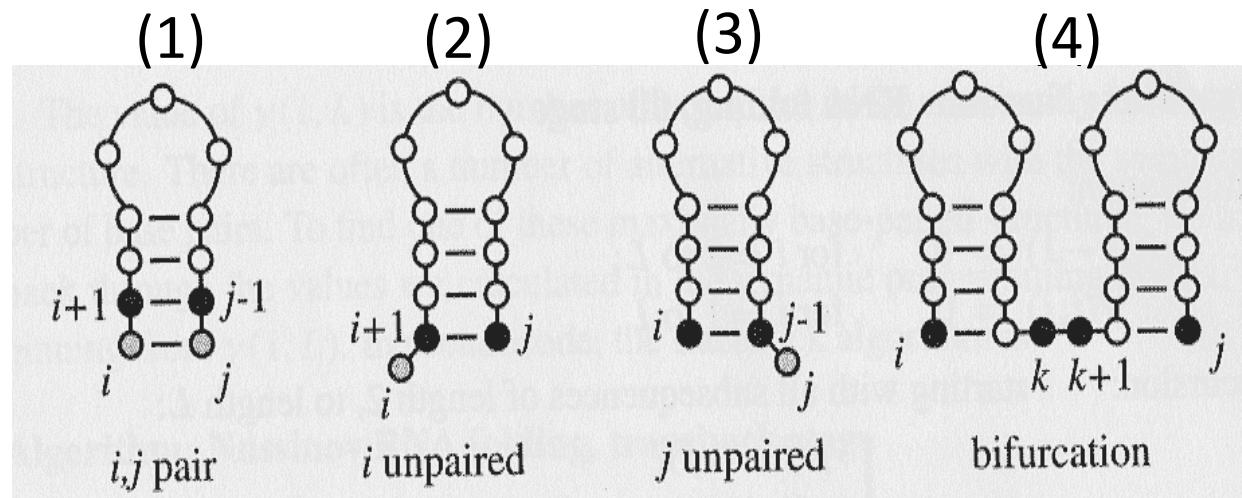


$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \text{ (1)} \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \text{ (1*)} \\ s[i + 1, j], & \text{if } i < j, \text{ (2)} \\ s[i, j - 1], & \text{if } i < j, \text{ (3)} \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \text{ (4)} \end{cases}$$

# Nussinov Algorithm – Example

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	0	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0

G | G | G | A | A | A | U | C | C  
 ( | ( | ( | ) | ) | ) | ) | )



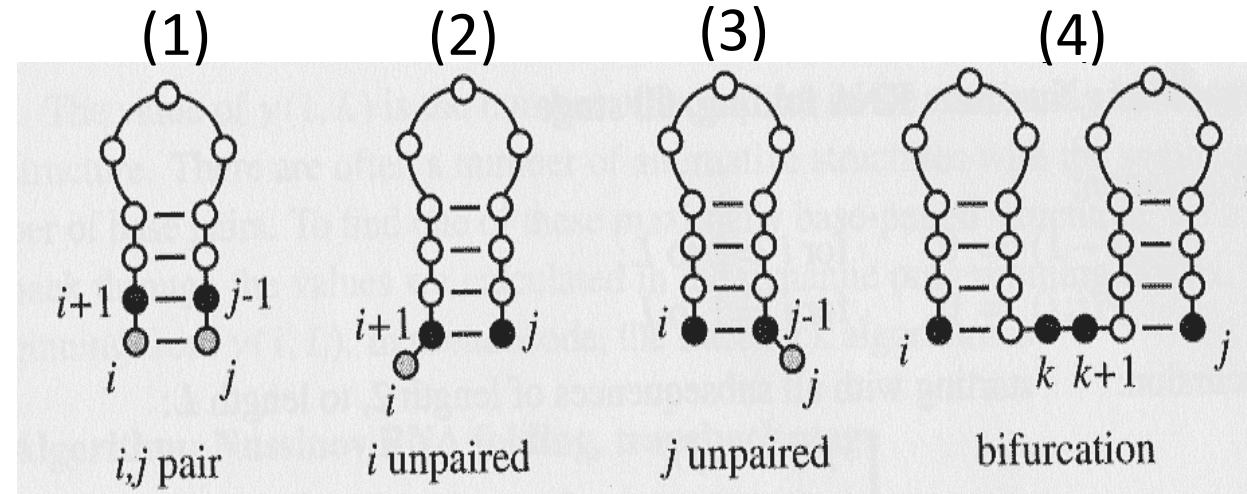
$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \text{ (1)} \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \text{ (1*)} \\ s[i + 1, j], & \text{if } i < j, \text{ (2)} \\ s[i, j - 1], & \text{if } i < j, \text{ (3)} \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \text{ (4)} \end{cases}$$

# Nussinov Algorithm – Example With Bifurcation

Where did we come from to get here?

1	2	3	4	5	6	7	8
G	C	A	C	G	A	C	G
0 1	1 1	2 2	2 2	3			
0 0	0 0	1 1	1 1	1 1	2		
0 0 0	1 1 1	1 1 1	2				
0 0 1	1 1 1	1 1 2					
0 0 0	1 1 1	1 1 1	2				
0 0 0	0 0 1	1 1 1					
0 0	0 0 1	1 1 1					
0 0	0 0	0 0 1					

G 1  
C 2  
A 3  
C 4  
G 5  
A 6  
C 7  
G 8



$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i + 1, j - 1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \text{ (1)} \\ s[i + 1, j - 1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \text{ (1*)} \\ s[i + 1, j], & \text{if } i < j, \text{ (2)} \\ s[i, j - 1], & \text{if } i < j, \text{ (3)} \\ \max_{i < k < j} \{s[i, k] + s[k + 1, j]\}, & \text{if } i < j, \text{ (4)} \end{cases}$$

GCACGACG  
( ) . (( . ))

# Develop Intuition

$$s[i, j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \\ s[i+1, j], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \\ s[i, j-1], & \text{if } i < j, \\ \max_{i < k < j} \{s[i, k] + s[k+1, j]\}, & \text{if } i < j, \end{cases}$$

# Nussinov Algorithm – Alternative Solutions

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	1	2	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0

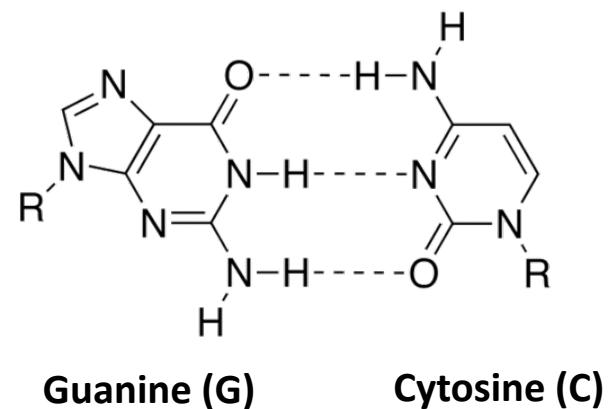
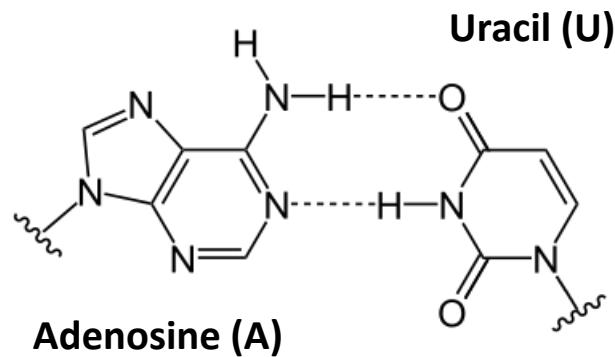
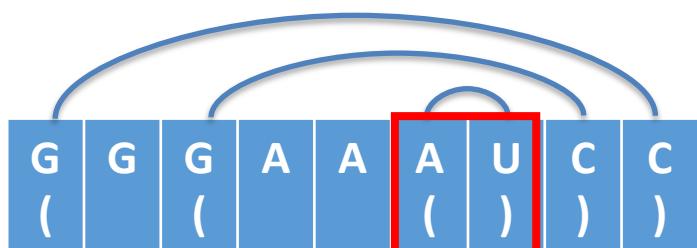


	G	G	G	A	A	A	U	C	C	
G	0	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	0	0	1	1
A	0	0	0	0	0	0	0	0	1	1
A	0	0	0	0	0	0	0	0	1	1
A	0	0	0	0	0	0	0	0	1	1
U	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0



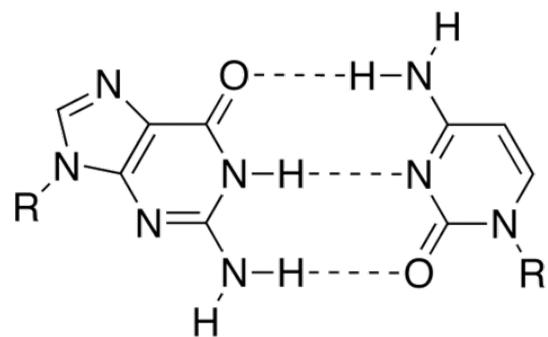
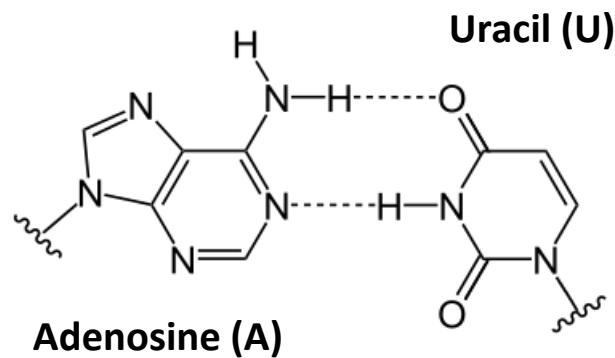
# Does this make sense?

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



# Does this make sense?

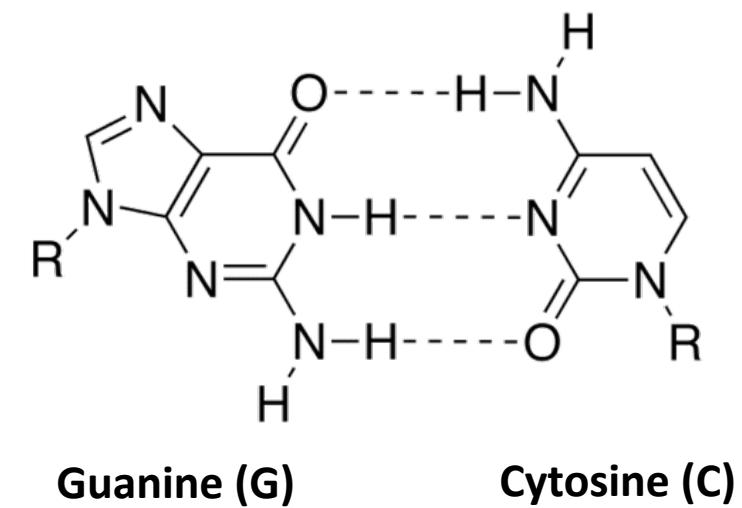
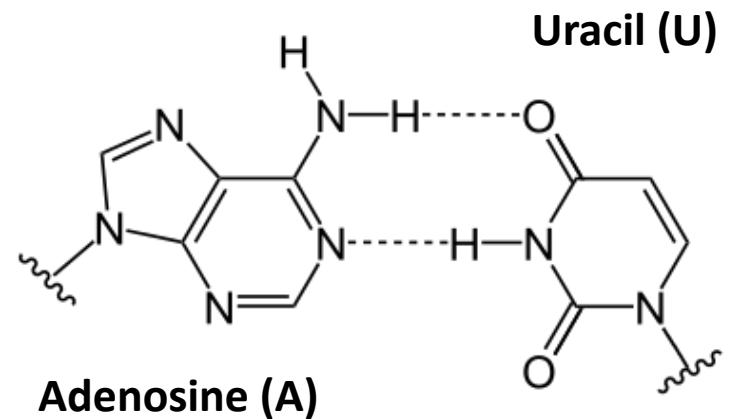
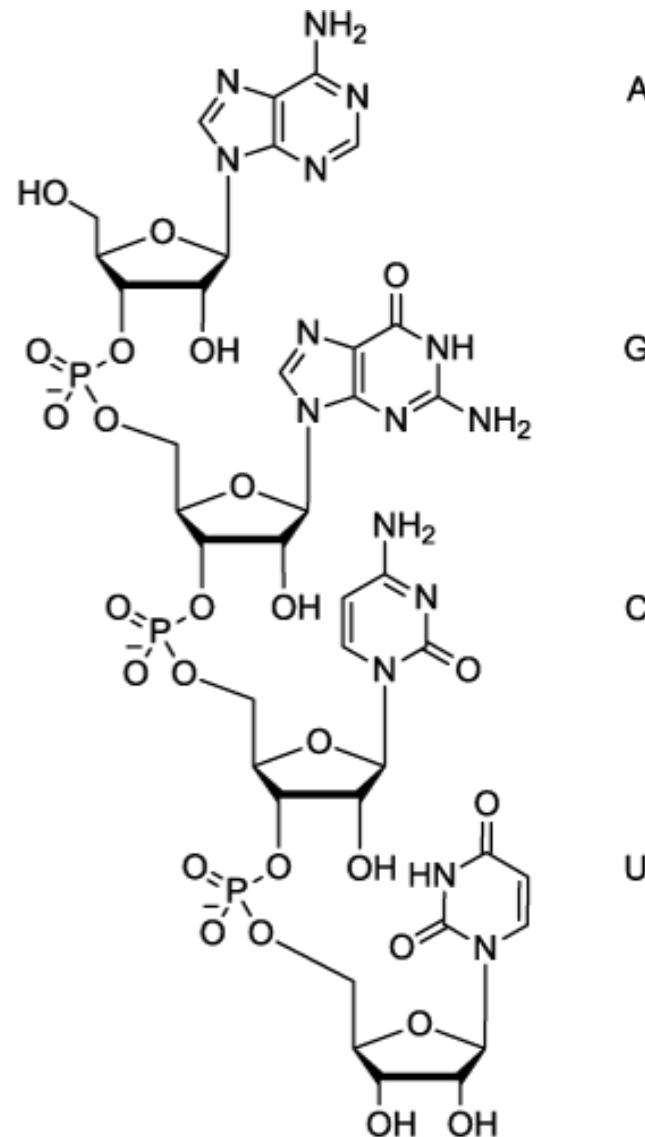
	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



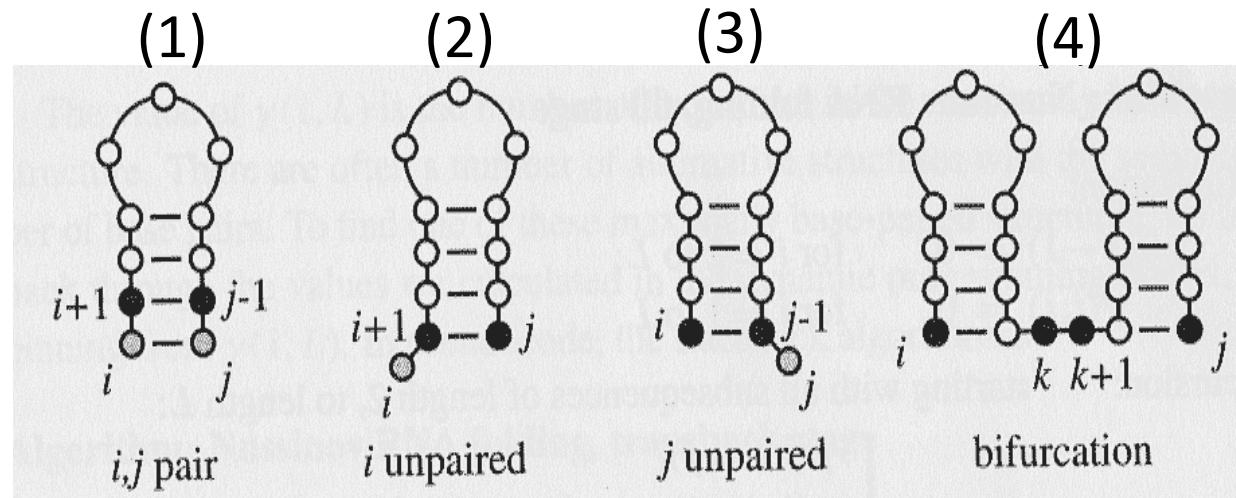
1	2	3	4	5	6	7	8	
G	C	A	C	G	A	C	G	
0	1	1	1	2	2	2	3	G 1
0	0	0	0	1	1	1	2	C 2
0	0	0	1	1	1	1	2	A 3
0	0	1	1	1	1	2		C 4
0	0	0	1	1	1	1		G 5
0	0	0	0	1	1			A 6
0	0	0	0	0	1			C 7
0	0	0	0	0	0			G 8

GCACGACG  
( ) • (( . ))

# Extension: Hairpin Loops with Minimum Length $\ell$



# Extension: Hairpin Loops with Minimum Length $\ell$



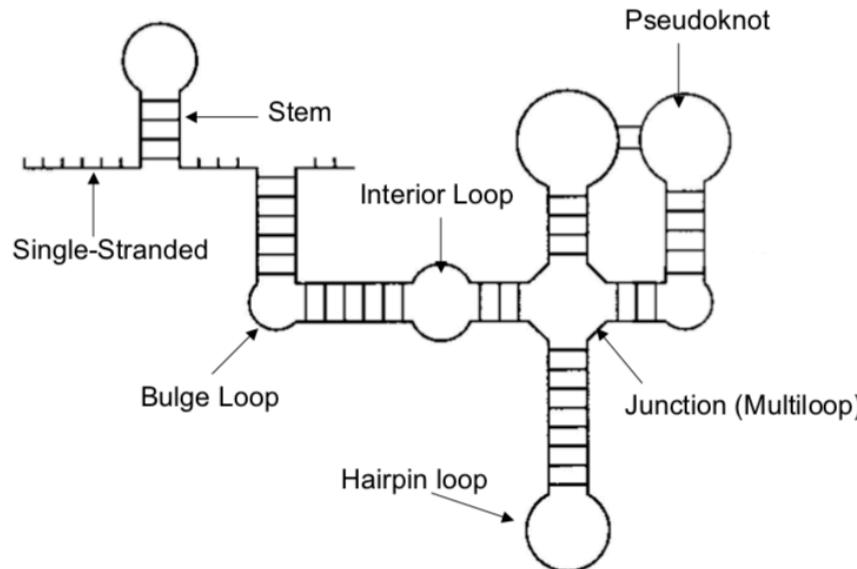
$$s[i, j] = \max \begin{cases} 0, & \text{if } i + \ell \geq j, \\ s[i+1, j-1] + 1, & \text{if } i + \ell < j \text{ and } (v_i, v_j) \in \Gamma, \\ s[i+1, j], & \text{if } i + \ell < j \text{ and } (v_i, v_j) \notin \Gamma, \\ s[i, j-1], & \text{if } i + \ell < j, \\ \max_{i+\ell < k < j} \{s[i, k] + s[k+1, j]\}, & \text{if } i + \ell < j, \end{cases}$$

# RNA Secondary Structure Prediction in Practice

Rather than maximize number of compl. base pairs, minimize free energy (FE)

Zuker's algorithm: Dynamic programming w/ three matrices similar to affine gap penalties

- $V(i,j)$ : FE of optimal structure of  $s[i..j]$  assuming  $i,j$  form a base pair
- $VBI(i,j)$ : FE of optimal structure of  $s[i..j]$  assuming  $i,j$  closes a bulge or internal loop
- $VM(i,j)$ : FE of optimal structure of  $s[i..j]$  assuming  $i,j$  closes a multibranch loop



FE minimization with pseudoknots is NP-hard  
[Lyngso and Pedersen, RECOMB 2000]

# Summary

- RNA is a sequence of four bases/nucleotides {A, U, C, G}
- RNA folds into structures due to base/nucleotide complementarity
  - A <--> U and C <--> G
- RNA secondary structure is defined by a set of non-overlapping complementary nucleotide pairs
- Pseudoknot-free structures have no “crossing” pairs
- Nussinov Algorithm: Dynamic programming to find pseudoknot-free structure with maximum number of complementary nucleotide pairs

## Reading:

- Topics are not in Jones and Pevzner book but in lecture notes and slides  
[Based on Chapter 10 in “Biological sequence analysis” by Durbin et al.]