CS 466
Introduction to Bioinformatics
Lecture 13

Mohammed El-Kebir

October 9, 2020

Course Announcements

Instructor:

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

TAs:

- Sarah Christensen (sac2) – Mondays, 3-4pm
- Wesley Wei Qian (weiqian3) – Fridays, 9-10am

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:

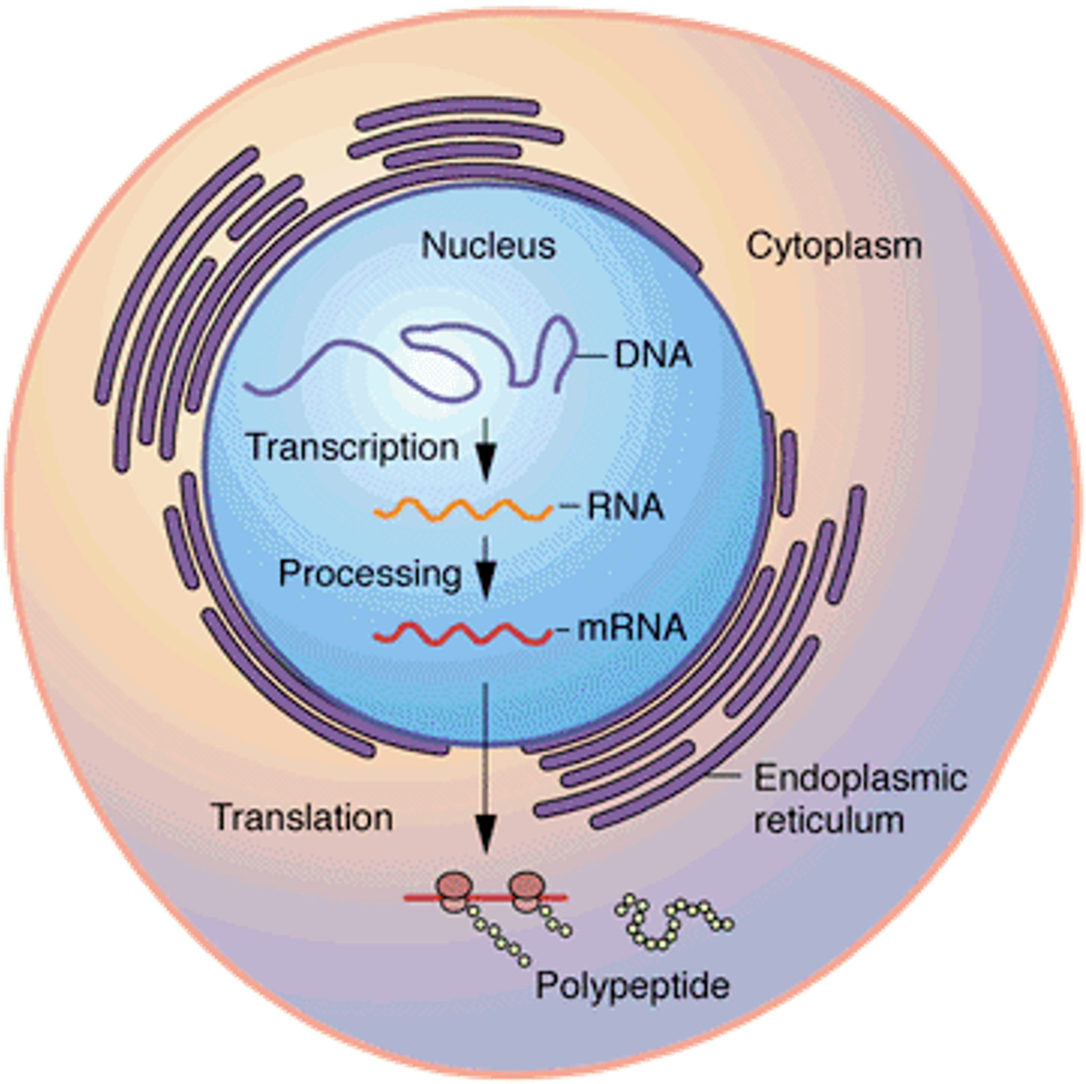
1. **DNA**
   Information storage.

2. **RNA**
   Old view: Mostly a "messenger".
   New view: Performs many important functions, through **3-D structure**!

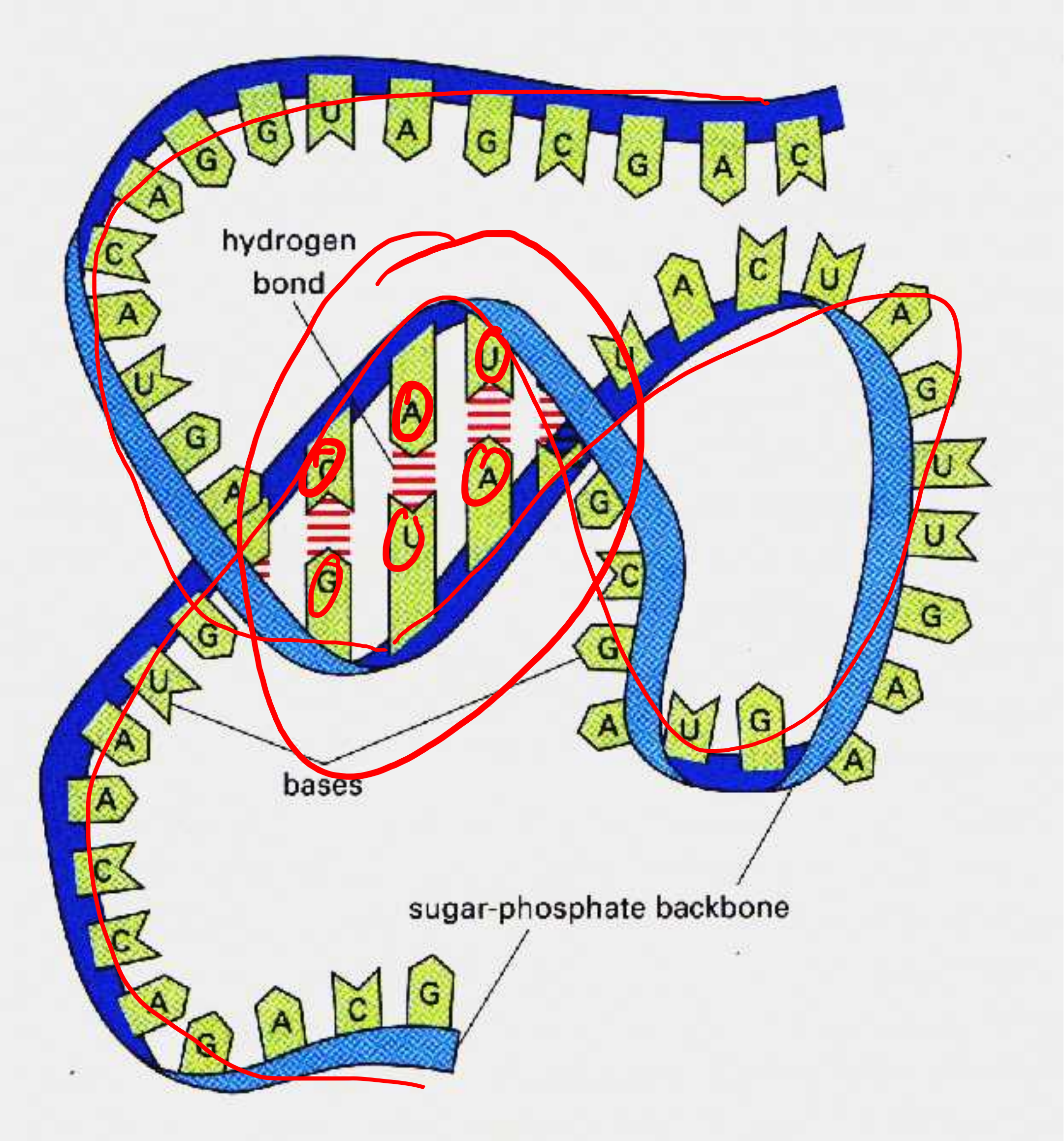
3. **Protein**
   Perform most cellular functions (biochemistry, signaling, control, etc.)

DNA → RNA → Protein

First proposed by Francis Crick in 1956.



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.

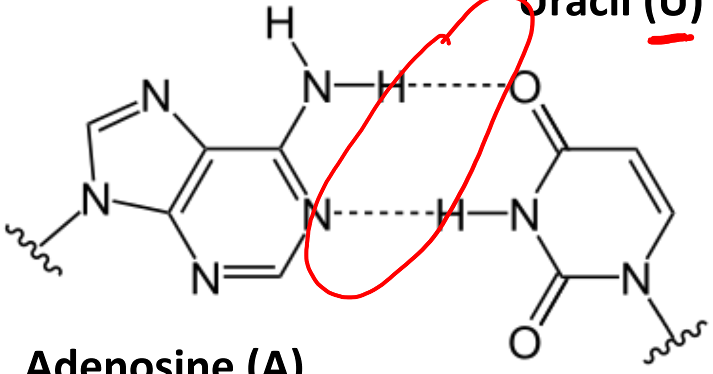
RNA – Nucleotide Complementarity

RNA can fold into structures due to nucleotide complementarity:
A <--> U and G <--> C

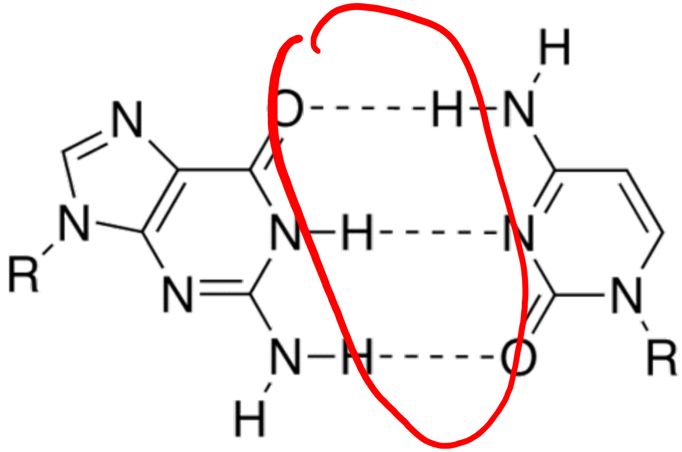
A <--> U (2 hydrogen bonds)
is slightly weaker than
G <--> C (3 hydrogen bonds)

G <--> U also observed but not as stable

Uracil (U)



Adenosine (A)

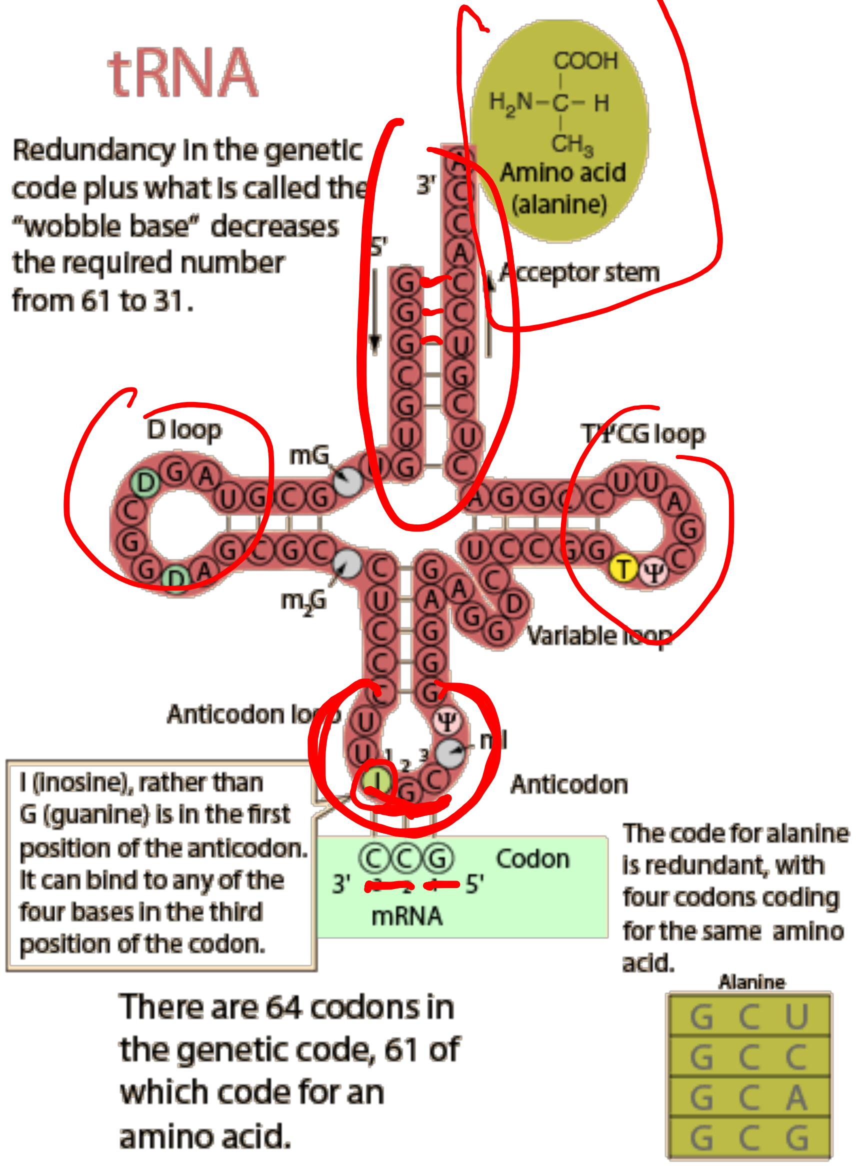


Guanine (G)

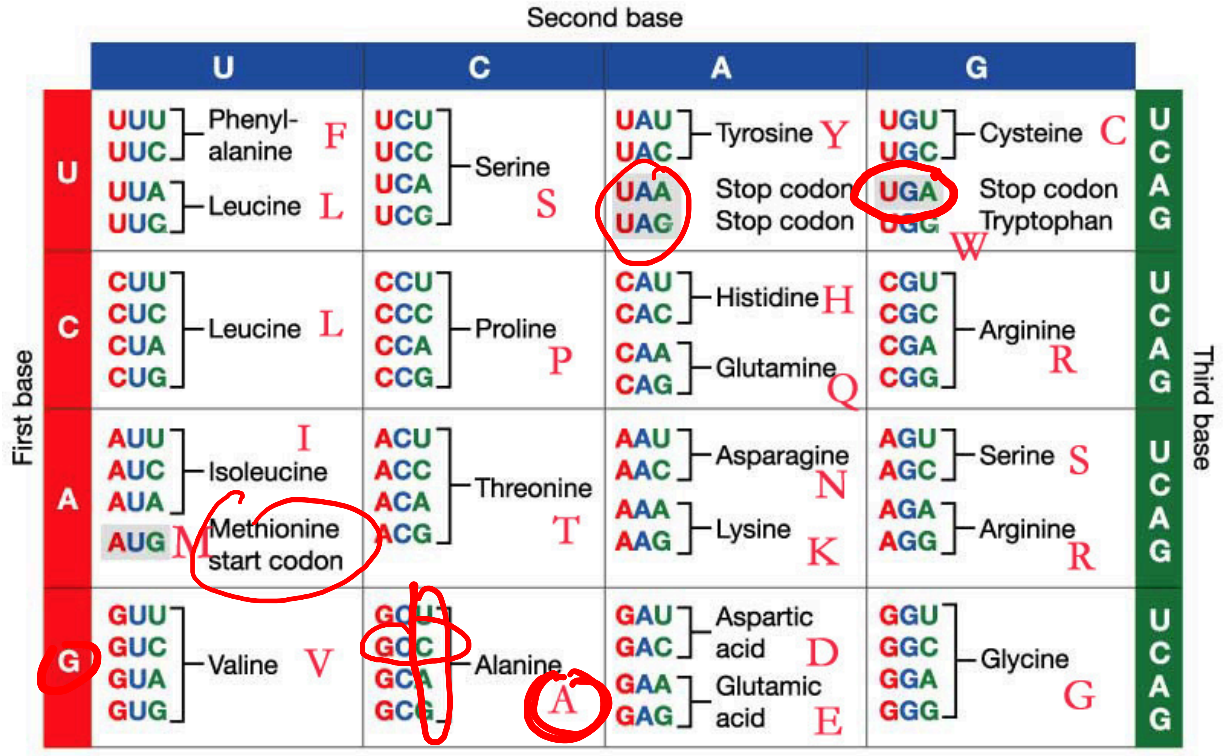
Cytosine (C)

transfer RNA (tRNA) Secondary Structure

codon table



http://hyperphysics.phy-astr.gsu.edu/hbase/Organic/trna.html#c3



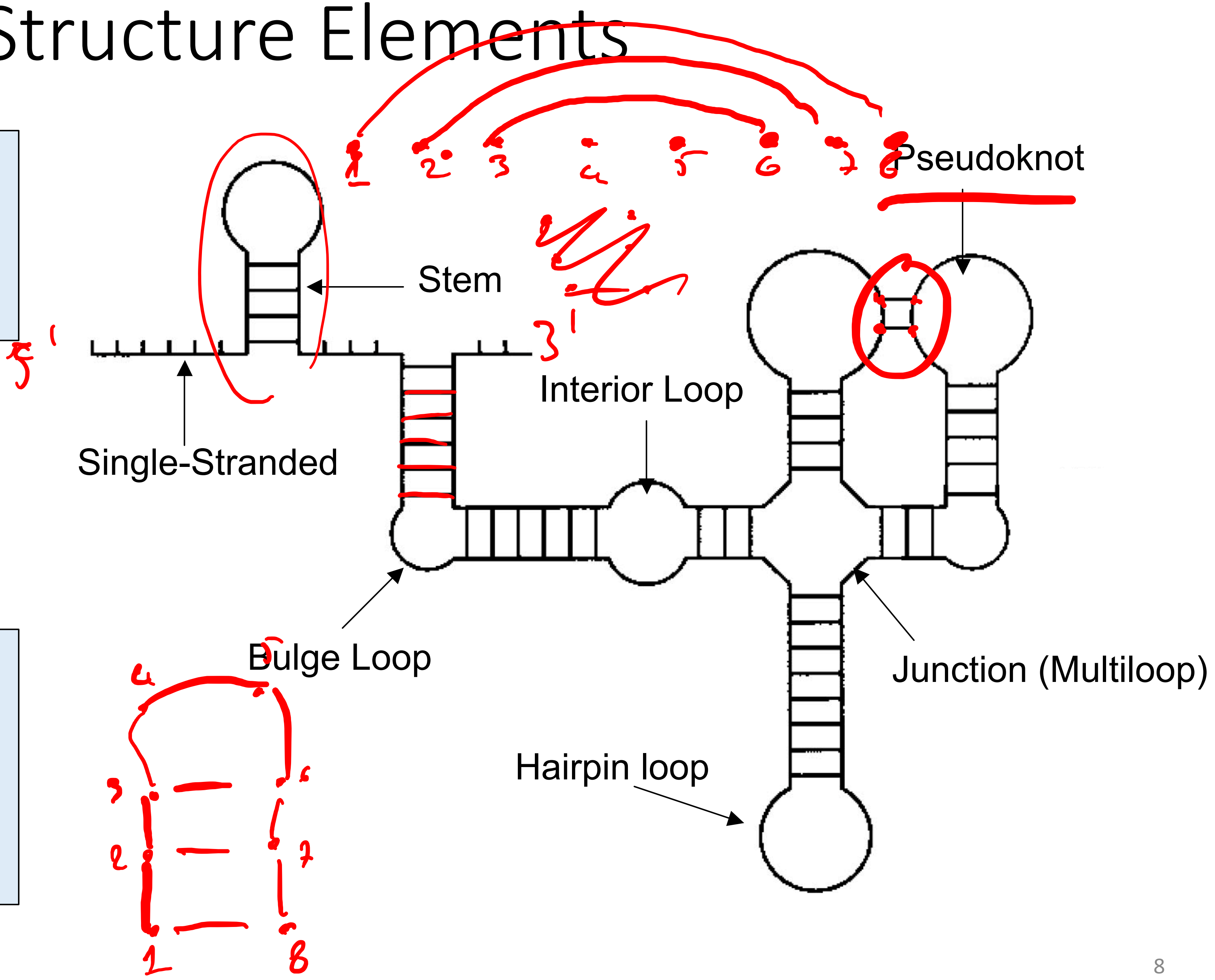
http://bioinfo.bisr.res.in/project/crat/pictures/codon.jpg

RNA

RNA Secondary Structure Elements

Each base/nucleotide participates in at most one pairing

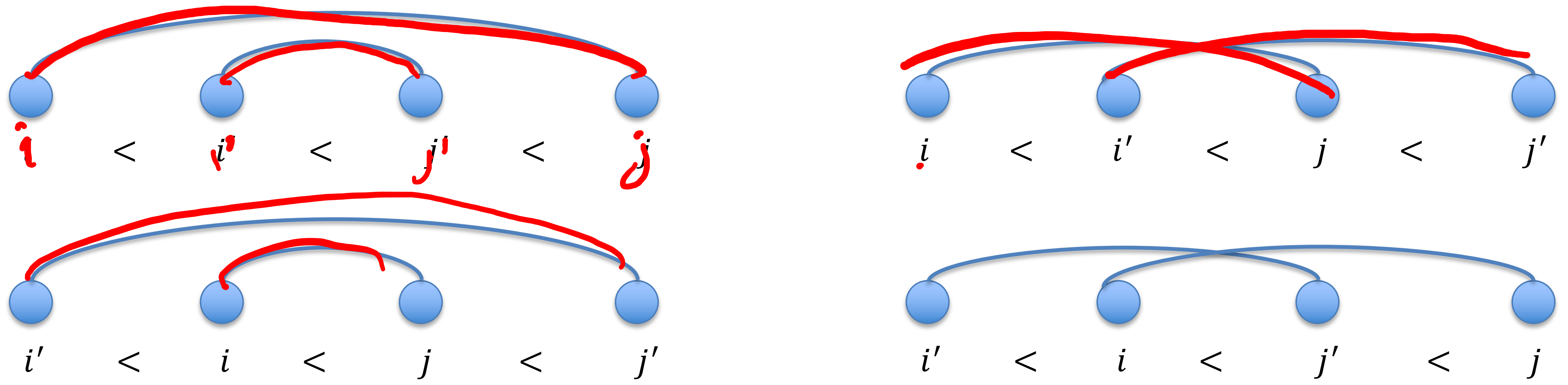
Secondary structure is determined by a set of non-overlapping base/nucleotide pairs



Nesting and Pseudoknot

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

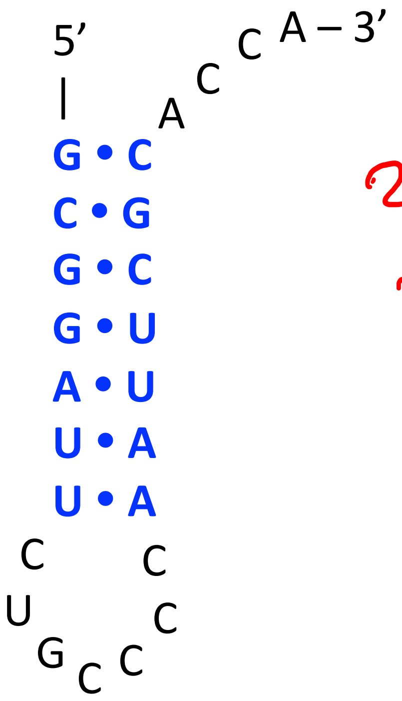
Base pairs (i, j) and (i', j') form a **pseudoknot** provided
$i < i' < j < j'$ or $i' < i < j' < j$



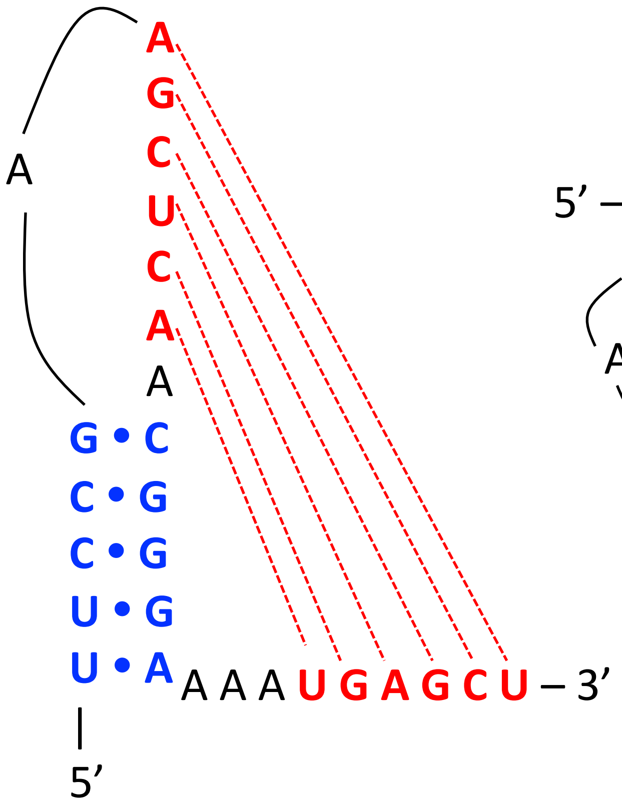
Most RNA molecules consist of nested base pairs

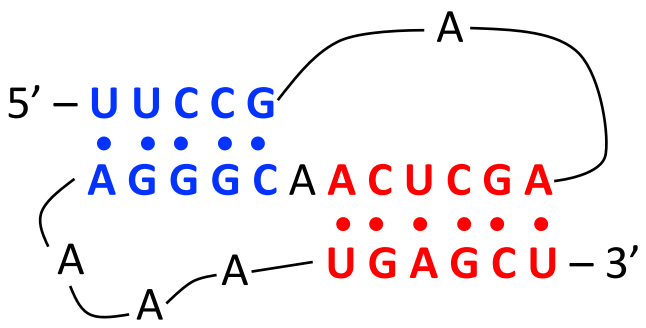
Nesting and Pseudoknot – Examples

Nesting	Pseudoknot
5’ – G C G G A U U C U G C C C C A A U U C G C A C C A – 3’	5’ – U U C C G A A G C U C A A C G G G A A A A U G A G C U – 3’



2D representation



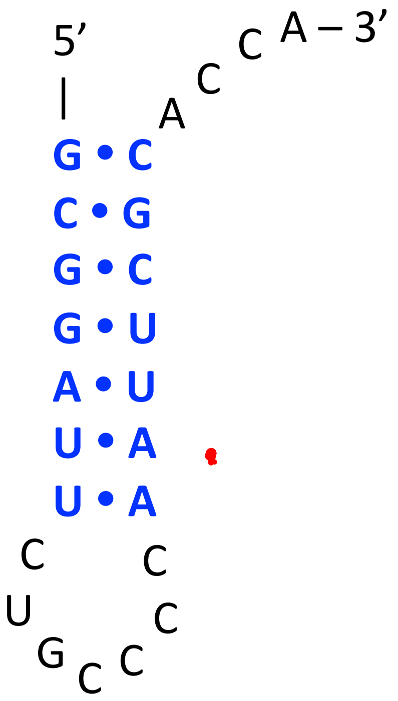


Nesting and Pseudoknot – Examples

Nesting

5’ – G C G G A U U C U G C C C C A A U U C G C A C C A – 3’

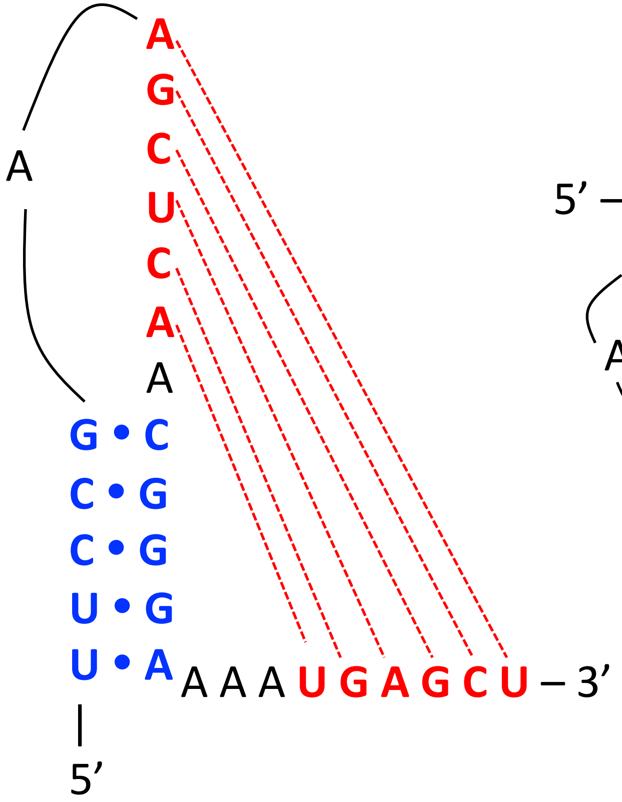
(((((((- - - - - - -))))))) - - - -

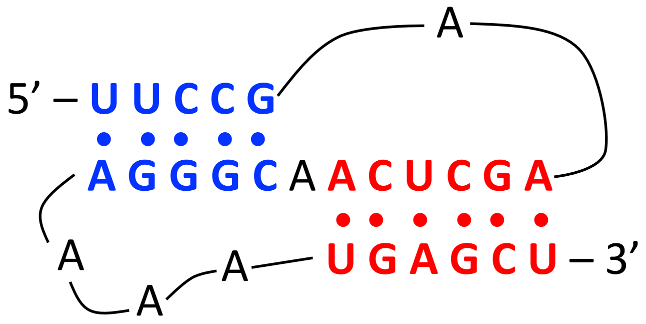


Pseudoknot

5’ – U U C C G A A G C U C A A C G G G A A A A U G A G C U – 3’

(((((- ((((((-))))) - - -))))))





Nussinov Algorithm

RNA can fold into structures due to nucleotide complementarity:
A <--> U and G <--> 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 <--> U and G <--> C

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

Question: How to find maximum number of such pairs?

matching is a subset of edges that are pairwise disjoint

5' – G C G G A U U C U G C C C C A A U U C G C A C C A – 3'

max cardinality matching

Nussinov Algorithm

RNA can fold into structures due to nucleotide complementarity:
A <--> U and G <--> 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 <--> U and G <--> 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,† GEORGE PIECZENIK,‡ JERROLD R. GRIGGS¶
AND DANIEL J. KLEITMAN§

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

Nussinov Algorithm – Dynamic Programming

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

1 2 3 4 ... n

$\bar{v}$ = 5’ – G C G G A U U C U G C C C C A A U U C G C A C C A – 3’

Nussinov Algorithm – Dynamic Programming

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

P^* set of pairings

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

$i \quad j$

$(i,j) \notin P^*$

$s[1,n]$ a solution

$s[i,j]$

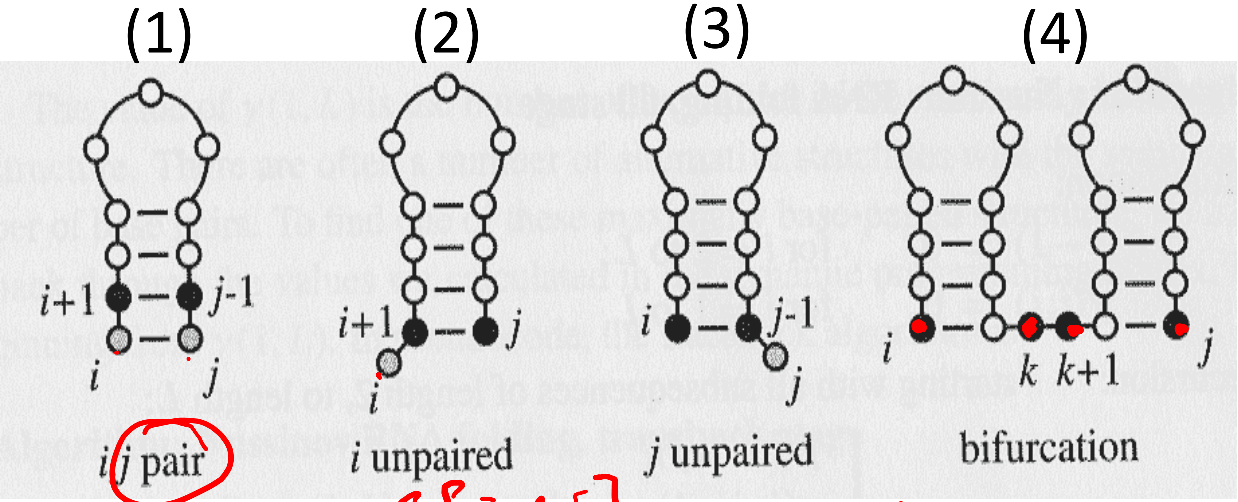
$\boxed{(i,j)} \in P^*$

$s[i+1, j-1] + 1$

Nussinov Algorithm – Dynamic Programming

Problem: Given RNA sequence $\mathbf{v} \in \{\mathrm{A}, \mathrm{U}, \mathrm{C}, \mathrm{G}\}^n$, 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 $v_i, \dots, v_j$



$1 + s[i+1, j-1]$

$s[i+1, j]$

$s[i, j-1]$

$s[i,k]$

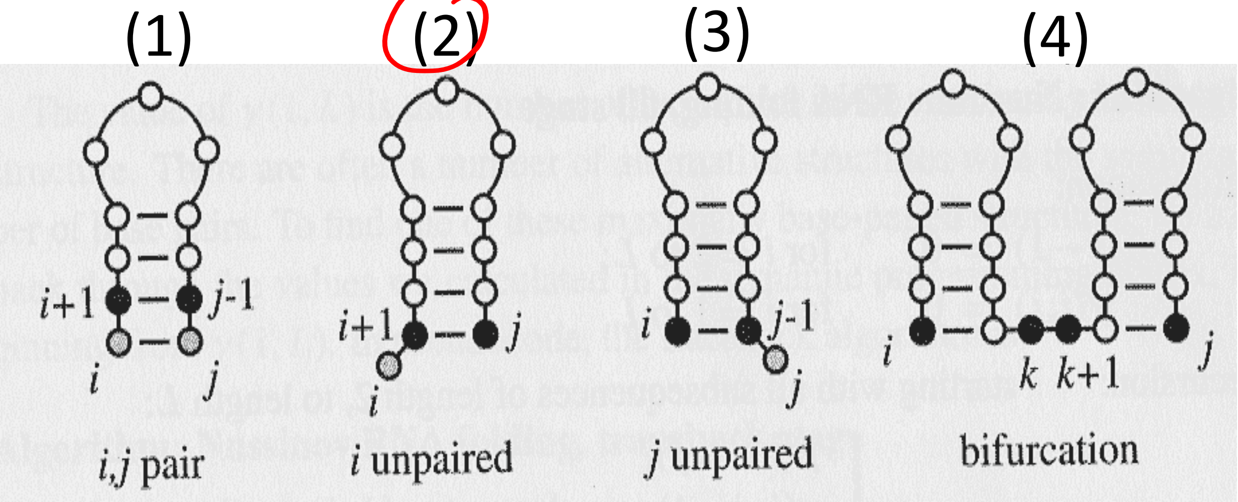
$s[k+1, j]$

Nussinov Algorithm – Dynamic Programming

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

$\Gamma = \{(A,U),(U,A),(C,G),(G,C)\}$

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



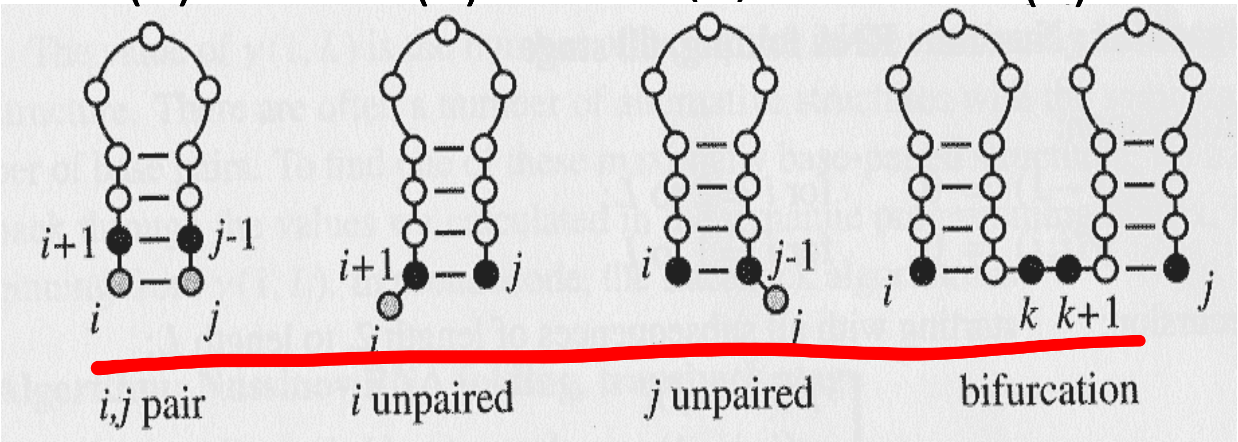
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \quad \text{base case} \\ s[i+1,j-1]+1, & \text{if } i<j \text{ and } (v_i,v_j) \in \Gamma, \quad (1) \\ s[i+1,j-1], & \text{if } i<j \text{ and } (v_i,v_j) \notin \Gamma, \quad (1^*) \\ s[i+1,j], & \text{if } i<j, \quad (2) \\ s[i,j-1], & \text{if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

Nussinov Algorithm – Dynamic Programming

Problem: Given RNA sequence $\mathbf{v} \in \{\mathrm{A}, \mathrm{U}, \mathrm{C}, \mathrm{G}\}^n$, 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 $v_i, \ldots, v_j$

(1) (2) (3) (4)



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

Question: Which case is redundant?

Develop Intuition

$j \rightarrow$

$n = 23$

solution

$s[1, n]$

i

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

$s[1, 23] = 5$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
G	C	U	C	G	G	G	U	U	C	C	C	U	A	U	U	C	A	A	G	A	G	C		
0																						5	G	1
0	0																				4		C	2
0	0	0																		3			U	3
0	0	0	0																2				C	4
0	0	0	0	0												1	1	1					G	5
0	0	0	0	0	0										0								G	6
0	0	0	0	0	0	0								0									G	7
0	0	0	0	0	0	0	0						0										U	8
0	0	0	0	0	0	0	0	0				0											U	9
0	0	0	0	0	0	0	0	0	0		0												C	10
0	0	0	0	0	0	0	0	0	0	0													C	11
0	0	0	0	0	0	0	0	0	0	0	0												C	12
0	0	0	0	0	0	0	0	0	0	0	0	0											U	13
0	0	0	0	0	0	0	0	0	0	0	0	0	0										A	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									U	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								U	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							C	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						A	18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					A	19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				G	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			A	21
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		G	22
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C	23

Develop Intuition

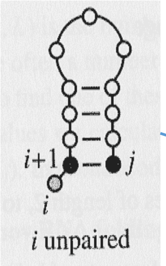
$$
s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \\ s[i+1, j], & \text{if } i < j, \\ s[i, j-1], & \text{if } i < j, \\ \boxed{\max_{i<k<j}\{s[i,k]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

(i, k, j)

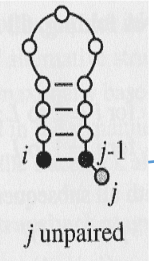
$O(n^3)$

(	(	(	(	(	(	(	-	-	-	)	)	-	(	-	)	)	-	-	)	)	)	)		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
G	**C**	**U**	**C**	**G**	**G**	**G**	**U**	**U**	**C**	**C**	**C**	**U**	**A**	**U**	**U**	**C**	**A**	**A**	**G**	**A**	**G**	**C**		
0																							**G**	1
0	0																						**C**	2
0	0	0																					**U**	3
0	0	0	0																				**C**	4
0	0	0	0	0																			**G**	5
0	0	0	0	0	0																		**G**	6
0	0	0	0	0	0	0																	**G**	7
0	0	0	0	0	0	0	0																**U**	8
0	0	0	0	0	0	0	0	0															**U**	9
0	0	0	0	0	0	0	0	0	0														**C**	10
0	0	0	0	0	0	0	0	0	0	0													**C**	11
0	0	0	0	0	0	0	0	0	0	0	0												**C**	12
0	0	0	0	0	0	0	0	0	0	0	0	0											**U**	13
0	0	0	0	0	0	0	0	0	0	0	0	0	0										**A**	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									**U**	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								**U**	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							**C**	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						**A**	18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					**A**	19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				**G**	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			**A**	21
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		**G**	22
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**C**	23

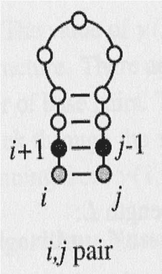
Nussinov Algorithm – Traceback Step



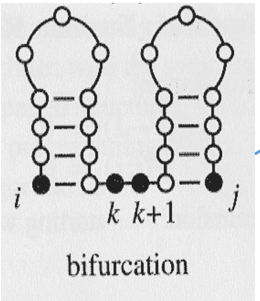
i unpaired



j unpaired



i,j pair



bifurcation

```
Push (1, n) onto stack
Repeat until stack is empty:
        pop (i,j)
        if i ≥ 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?

recursive

functional

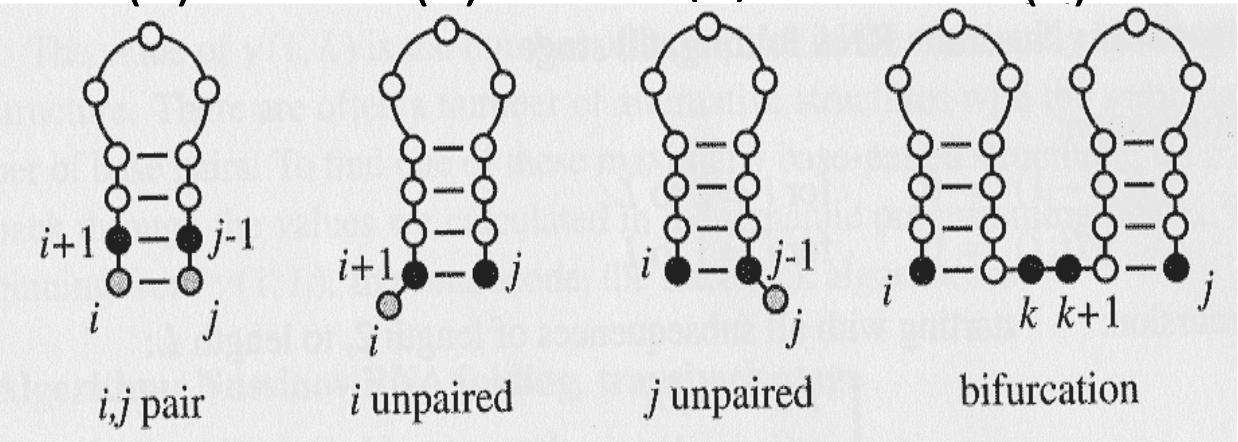
Question: Can we do this recursively?

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

(1) (2) (3) (4)

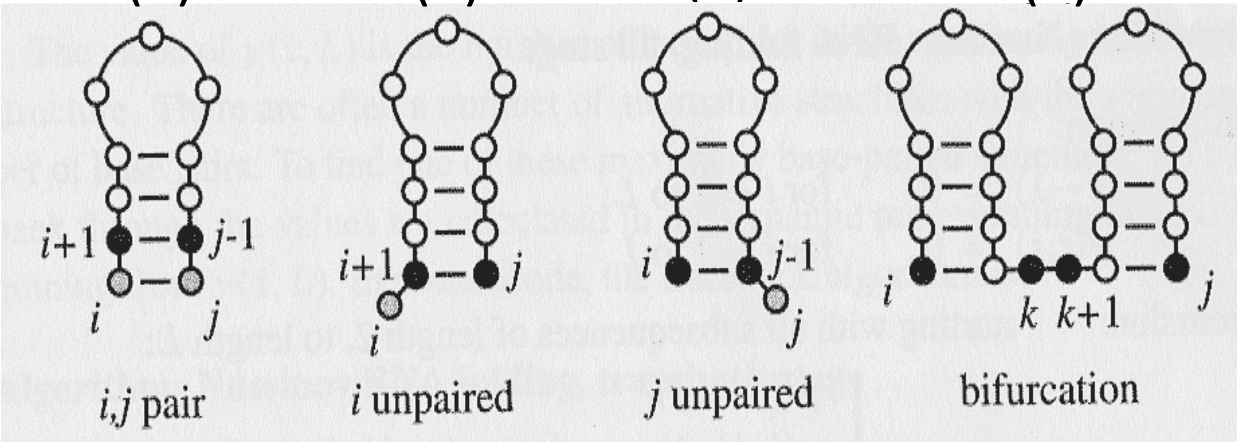


$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1,j-1]+1, & \text{if } i<j \text{ and } (v_i,v_j)\in\Gamma, \quad (1) \\ s[i+1,j-1], & \text{if } i<j \text{ and } (v_i,v_j)\notin\Gamma, \quad (1^*) \\ s[i+1,j], & \text{if } i<j, \quad (2) \\ s[i,j-1], & \text{if } i<j, \quad (3) \\ \max_{i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

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

(1) (2) (3) (4)

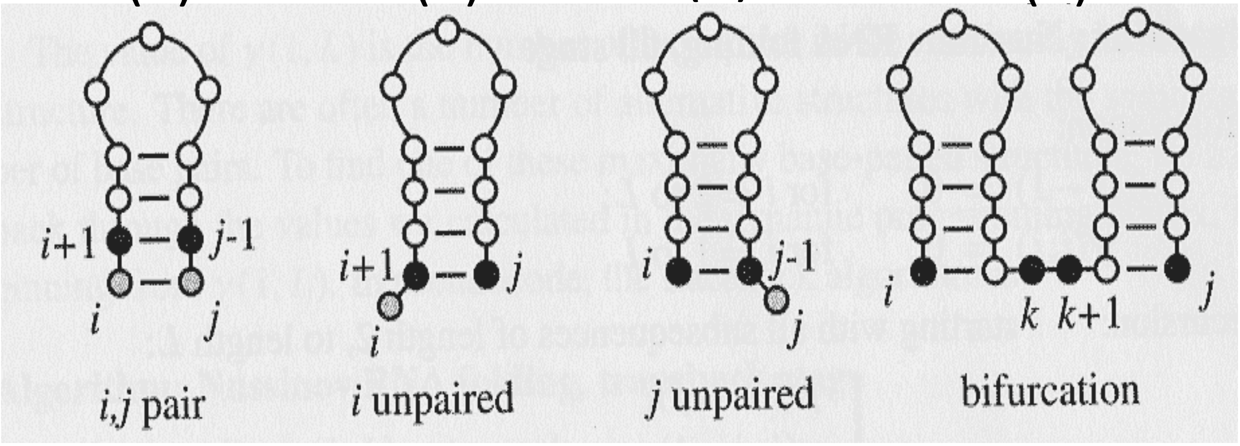


$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \quad (1) \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \quad (1^*) \\ s[i+1, j], & \text{if } i < j, \quad (2) \\ s[i, j-1], & \text{if } i 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

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

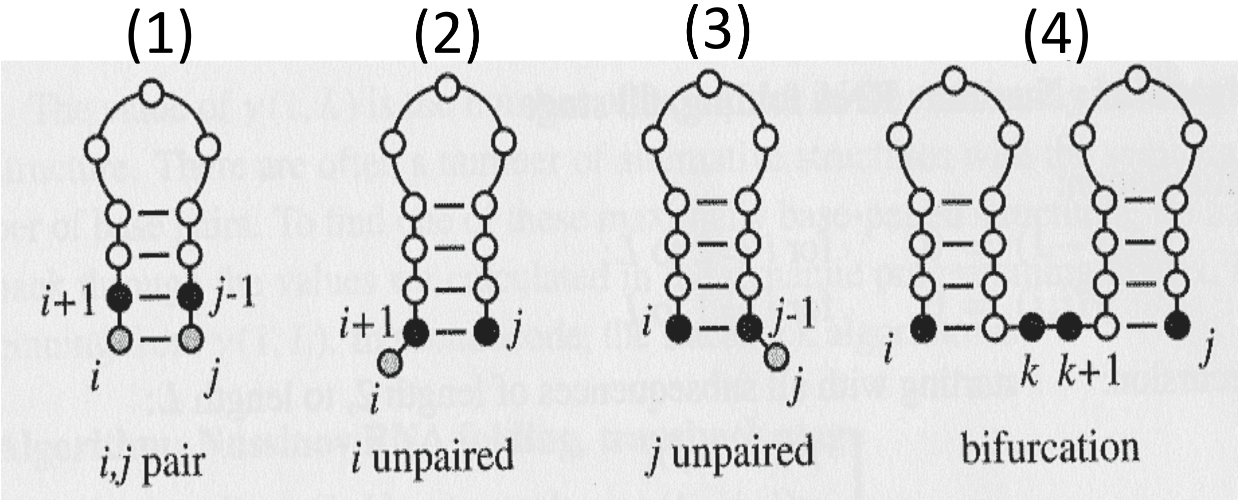
(1) (2) (3) (4)



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

Nussinov Algorithm – Example

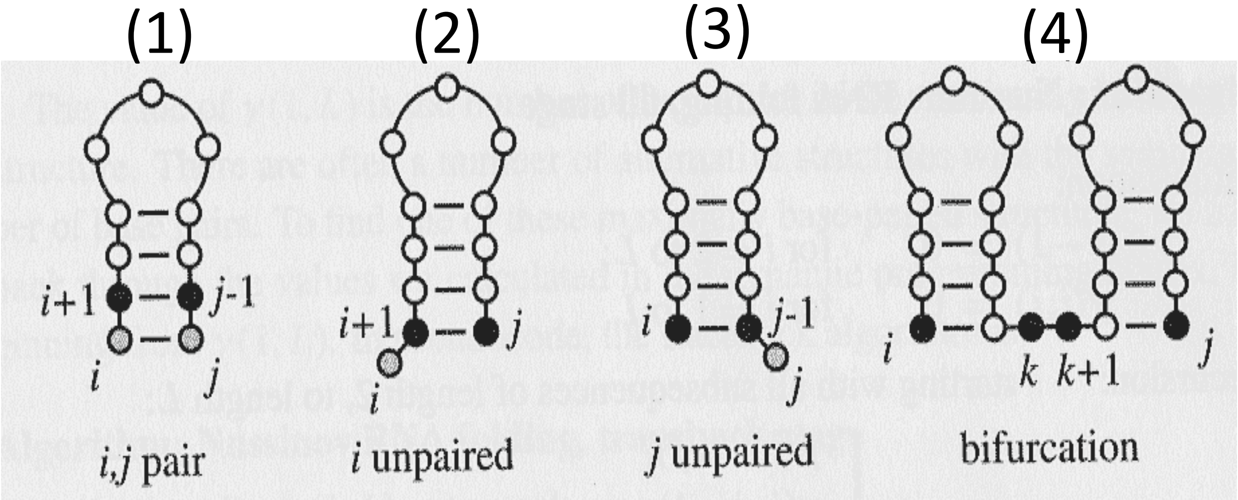
	G	G	G	A	A	A	U	C	C
G	0	0	0	0					
G	0	0	0	0	0				
G	0	0	0	0	0	0			
A	0	0	0	0	0	0	1		
A	0	0	0	0	0	0	1	1	
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



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

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



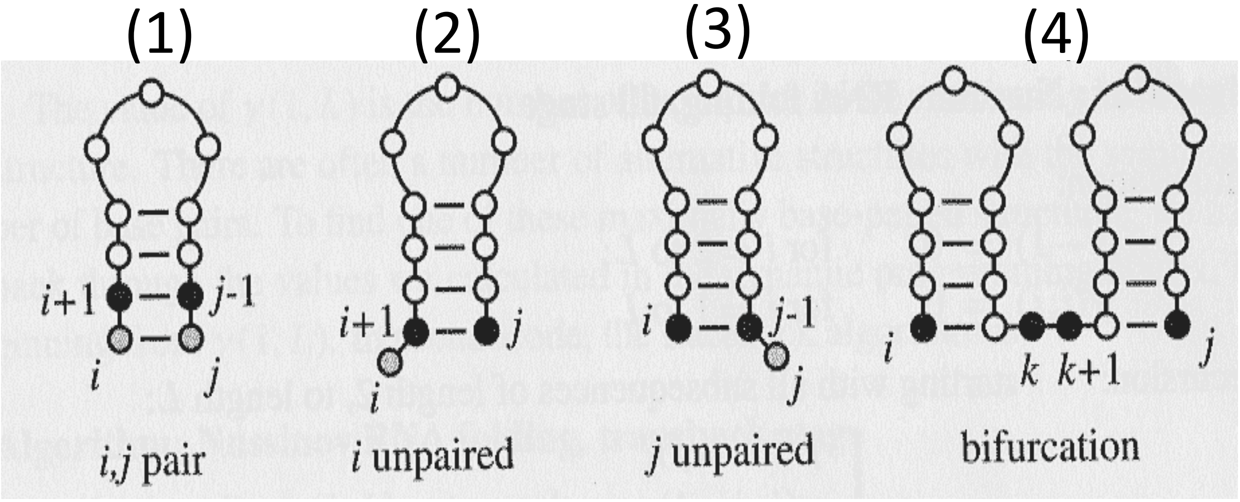
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \quad (1) \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \quad (1^*) \\ s[i+1, j], & \text{if } i < j, \quad (2) \\ s[i, j-1], & \text{if } i 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

Nussinov Algorithm – Example

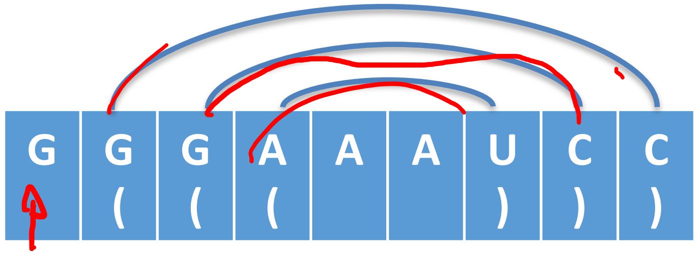
j

i

	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



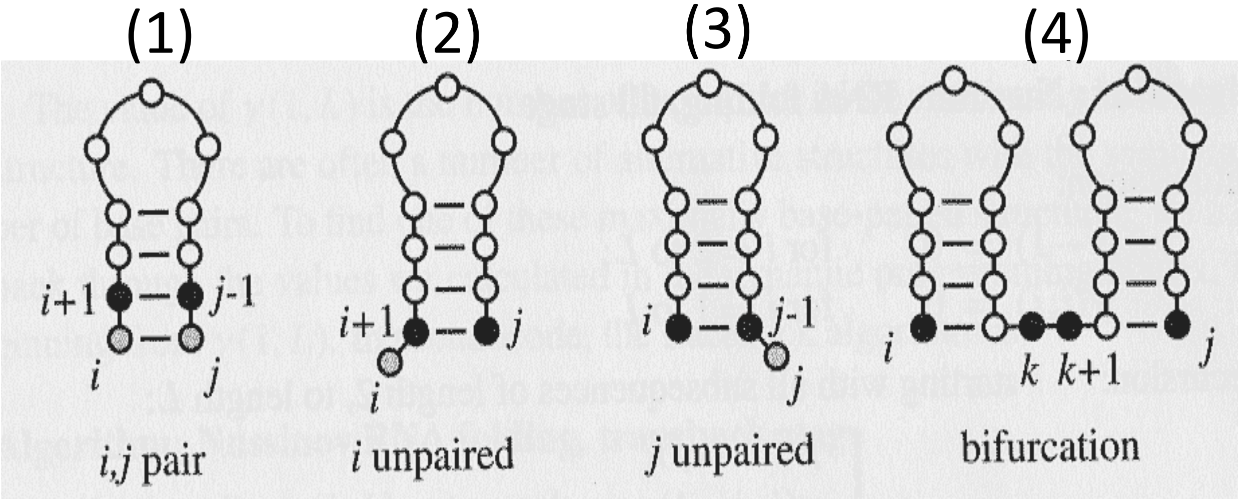
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1,j-1]+1, & \text{if } i<j \text{ and } (v_i,v_j)\in\Gamma, \quad (1) \\ s[i+1,j-1], & \text{if } i<j \text{ and } (v_i,v_j)\notin\Gamma, \quad (1^*) \\ s[i+1,j], & \text{if } i<j, \quad (2) \\ s[i,j-1], & \text{if } i<j, \quad (3) \\ \max_{i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



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



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

GCACGACG

( ) . ((.))

$s[1,8] = s[1,3] + s[4,8]$

k

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	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

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

	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

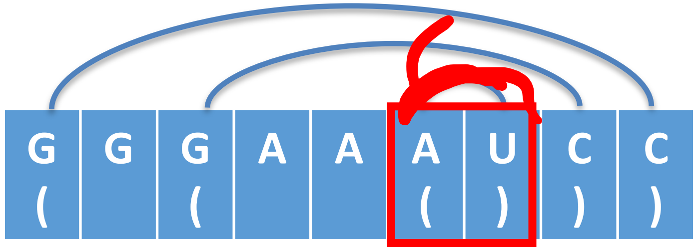
G	G	G	A	A	A	U	C	C
(		(	(			)	)	)

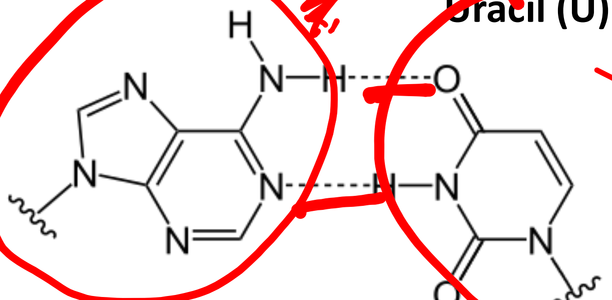
	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0

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

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

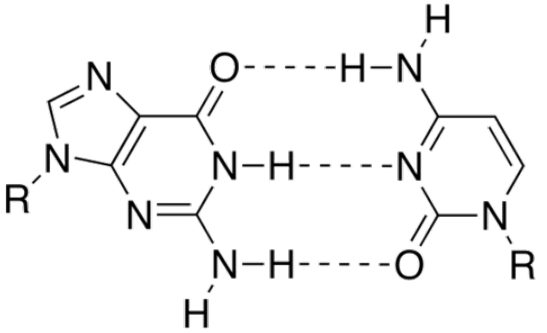




Uracil (U)

Adenosine (A)

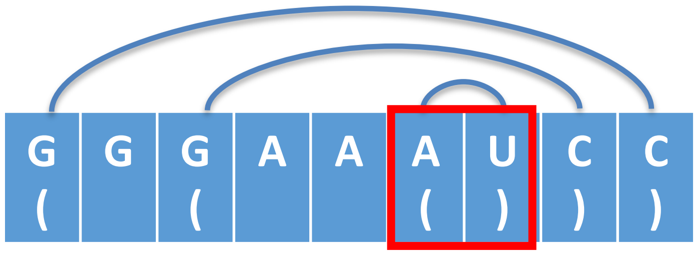
A U

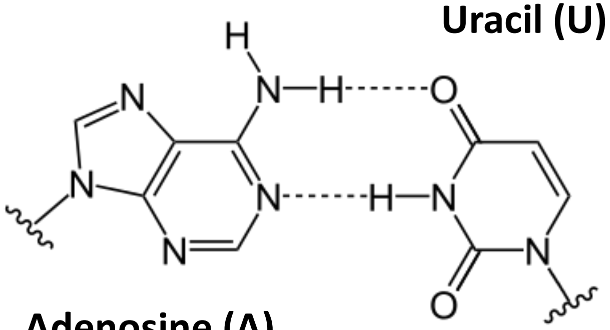


Guanine (G) Cytosine (C)

Does this make sense?

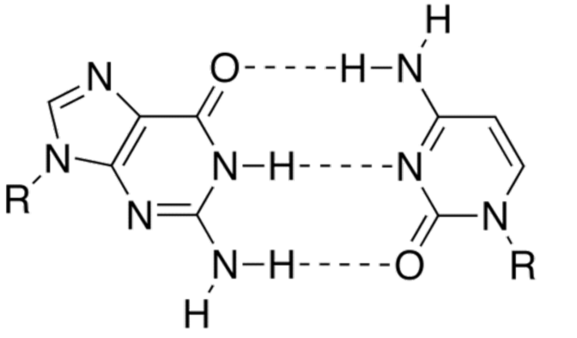
	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0





Adenosine (A)

Uracil (U)



Guanine (G)

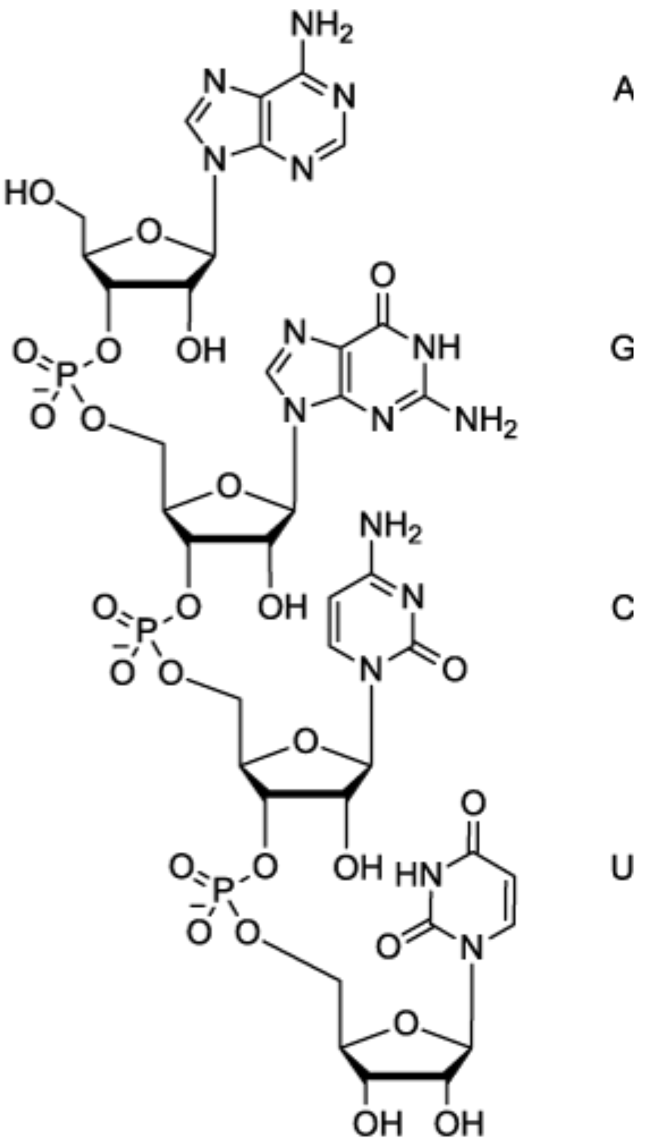
Cytosine (C)

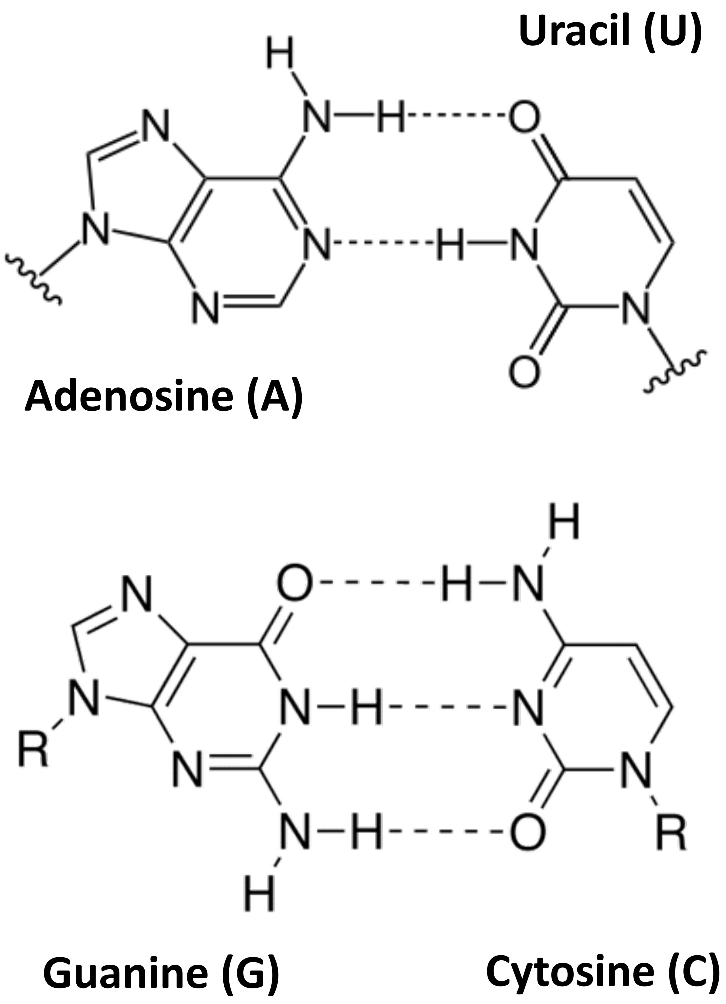
1	2	3	4	5	6	7	8		
G	C	A	C	G	A	C	G		
0	1	1	1	2	2	2	3	G	1
0	0	0	0	1	1	1	2	C	2
	0	0	0	1	1	1	2	A	3
		0	0	1	1	1	2	C	4
			0	0	0	1	1	G	5
				0	0	0	1	A	6
					0	0	1	C	7
						0	0	G	8

GCACGACG

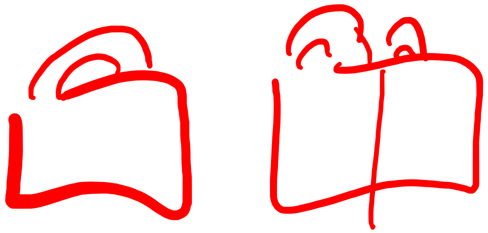
( ) . ((.))

Extension: Hairpin Loops with Minimum Length ℓ



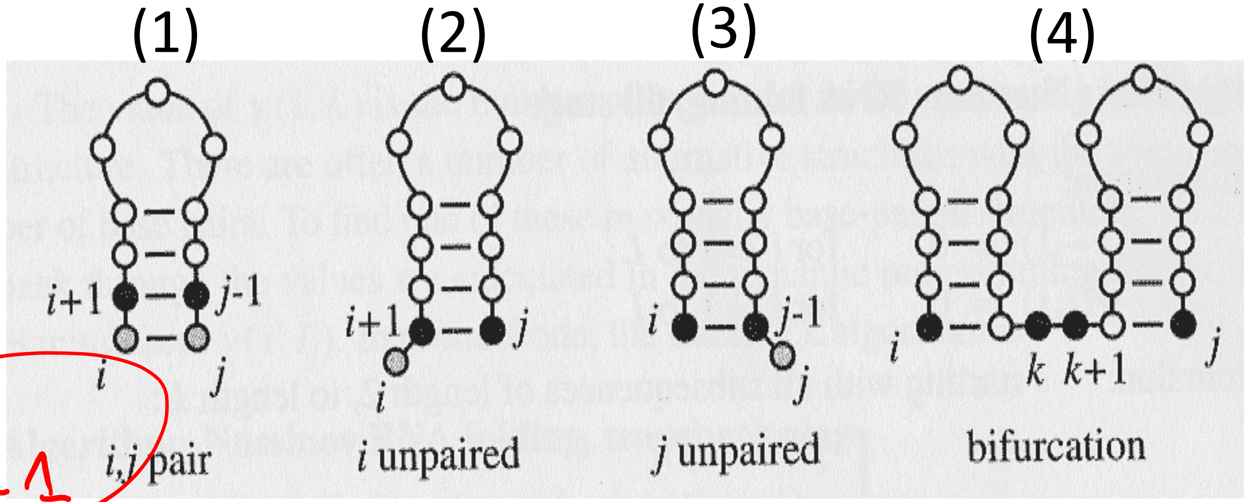


Extension: Hairpin Loops with Minimum Length ℓ



i, j

if $i = j - 1$



$\ell = 1$

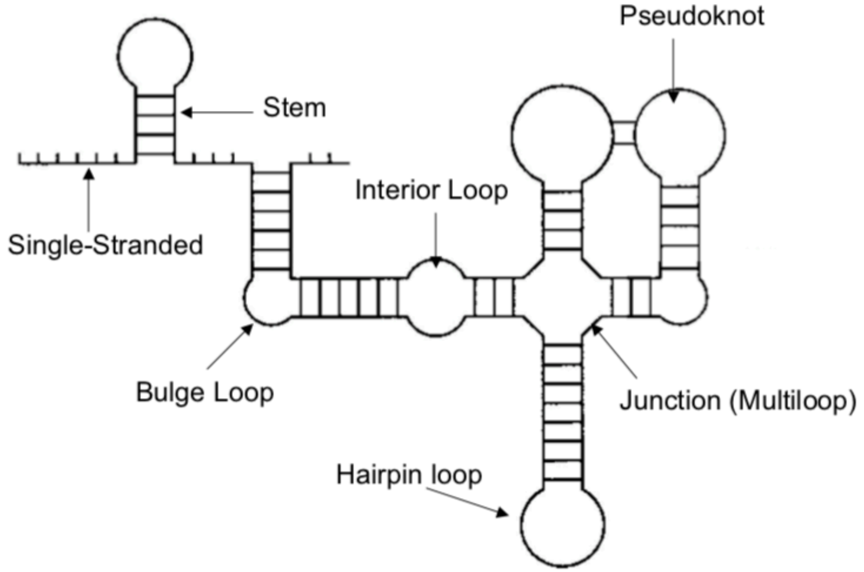
$$s[i,j] = \max \begin{cases} 0, & \text{if } i + \ell \geq j, \\ s[i+1, j-1] + 1, & \text{if } i + \ell < j \text{ and } (v_i, v_j) \in \Gamma, & (1) \\ s[i+1, j-1], & \text{if } i + \ell < j \text{ and } (v_i, v_j) \notin \Gamma, & (1^*) \\ s[i+1, j], & \text{if } i + \ell < j, & (2) \\ s[i, j-1],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

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 Matching
- 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.]