CS 466
Introduction to Bioinformatics
Lecture 6

Mohammed El-Kebir

September 19, 2018

Course Announcements

Instructor:

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

TA:

- Anusri Pampari (pampari2)
- Office hours: Thursdays, 11:00-11:59am in SC 4105

TA office hour canceled on Sept. 20

Solution to HW 1 released on Sept. 21

Midterm on Oct. 10, 7-9pm, 1310 DCL

Outline

- Two open questions on Hirschberg algorithm
- RNA secondary structure

Reading:

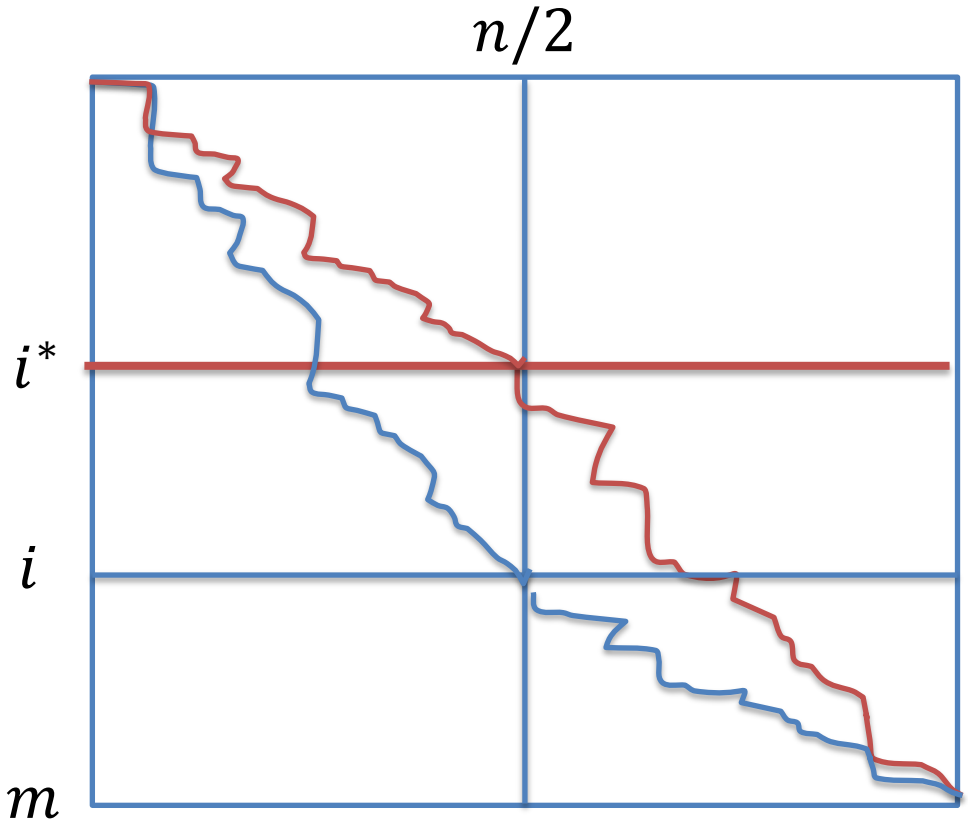
- Topics are not in Jones and Pevzner book but in lecture notes and slides [Based on Chapter 10 in “Biological sequence analysis” by Durbin et al.]

Hirschberg Algorithm: Reversing Edges Necessary?

Max weight path from $(0,0)$ to (m,n) through $(i^*, n/2)$

$$i^* = \arg\max_{0 \le i \le m} \{ \text{prefix}(i) + \text{suffix}(i) \}$$

Compute $\{\text{prefix}(i) \mid 0 \le i \le m\}$ in $O(mn)$ time and $O(m)$ space, by starting from $(0,0)$ to $\left(m, \frac{n}{2}\right)$ keeping only two columns in memory. [**single-source multiple destinations**]



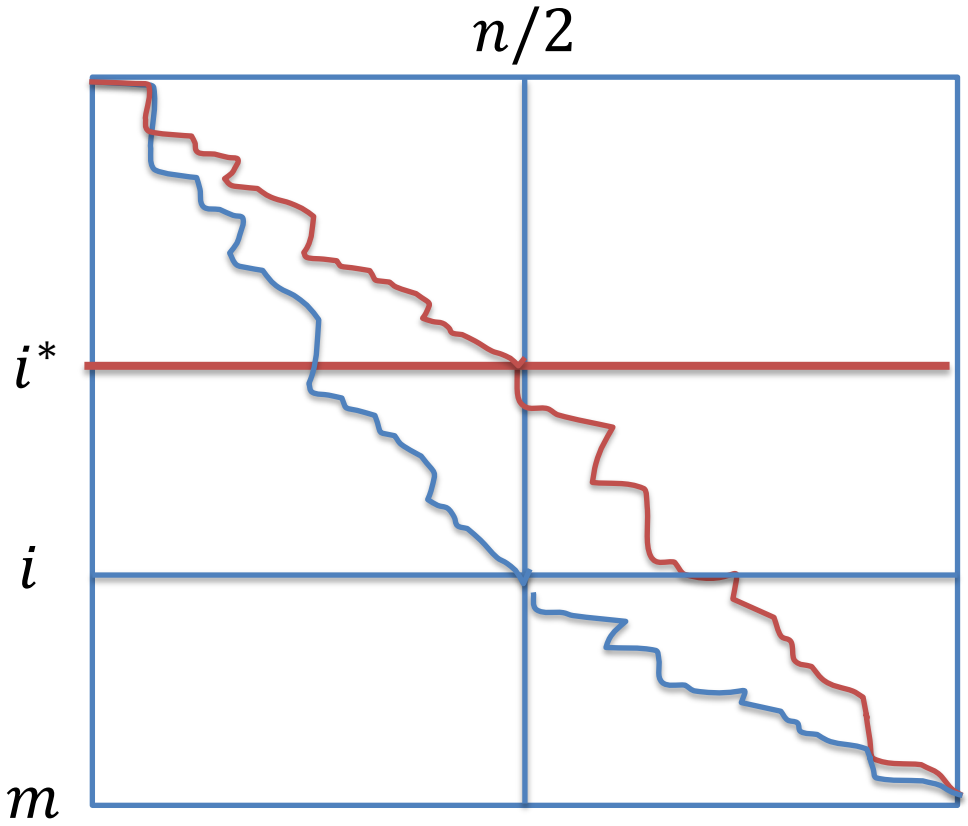
Hirschberg Algorithm: Reversing Edges Necessary?

Max weight path from $(0,0)$ to (m,n) through $(i^*, n/2)$

$$i^* = \arg\max_{0 \le i \le m} \{ \text{prefix}(i) + \text{suffix}(i) \}$$

Compute $\{\text{prefix}(i) \mid 0 \le i \le m\}$ in $O(mn)$ time and $O(m)$ space, by starting from $(0,0)$ to $\left(m, \frac{n}{2}\right)$ keeping only two columns in memory. [**single-source multiple destinations**]

Want: Compute $\{\text{suffix}(i) \mid 0 \le i \le m\}$ in $O(mn)$ time and $O(m)$ space



Doing a shortest path from each $\left(i, \frac{n}{2}\right)$ to (m,n) (for all $0 \le i \le m$) will not achieve desired running time!

Reversing edges enables single-source multiple destination computation in desired time and space bound!

Hirschberg Algorithm: Reconstructing Alignment

		A	T	C	G	C
	V \ W	0	1	2	3	4	5
	0	0	-1	-2	-3	-4	-5
A	1	-1	1	0	-1	-2	-3
T	2	-2	0	2	1	0	-1
G	3	-3	-1	1	1	2	1
T	4	-4	-2	0	0	1	1
C	5	-5	-3	-1	1	0	2

A	T	–	G	T	C
A	T	C	G	–	C

Hirschberg(i, j, i', j')

1. **if** $j' - j > 1$
2. $i^* \leftarrow \arg\max_{0 \le i \le m} \text{wt}(i)$
3. Report $(i^*, j + \frac{j'-j}{2})$
4. Hirschberg$(i, j, i^*, j + \frac{j'-j}{2})$
5. Hirschberg$(i^*, j + \frac{j'-j}{2}, i', j')$

Problem: Given reported vertices and scores $\{(i_0, 0, s_0), \ldots, (i_n, n, s_n)\}$, find intermediary vertices.

Transposing matrix does not help, because gaps could occur in both input sequences

Outline

- Two open questions on Hirschberg algorithm
- 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:

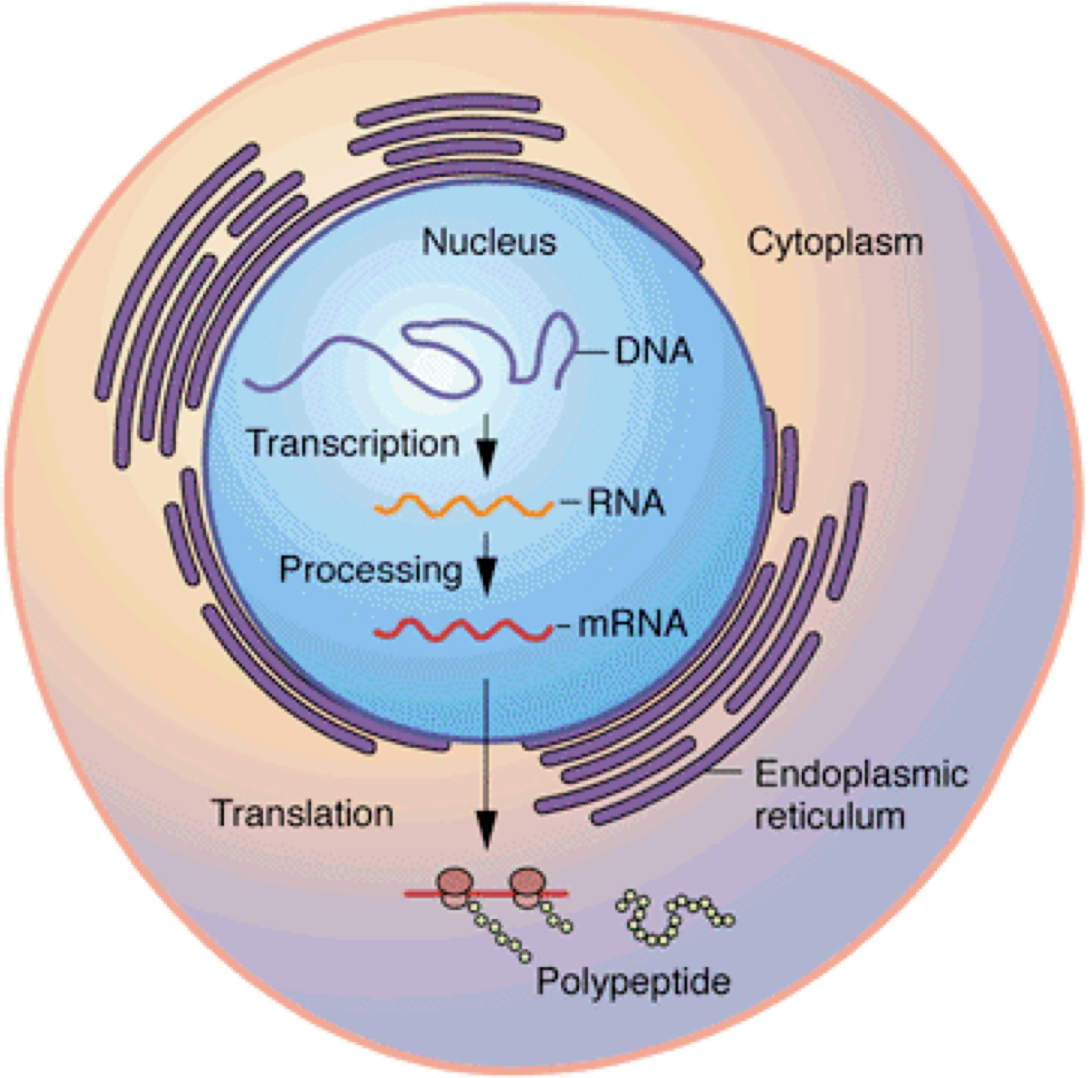
1. **DNA**
   Information storage.

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

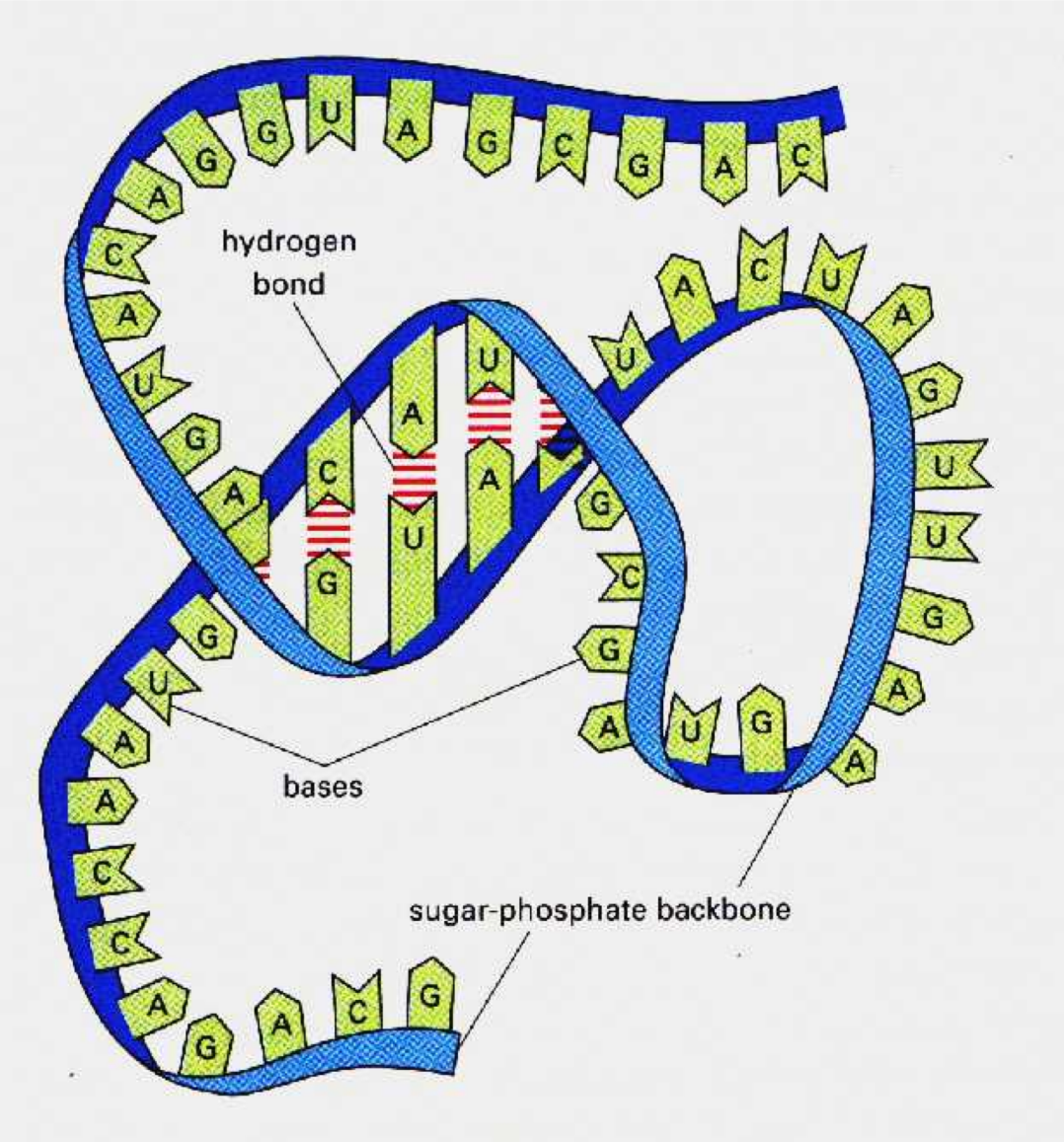
3. **Protein**
   Perform most cellular functions (biochemistry, signaling, control, etc.)

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:

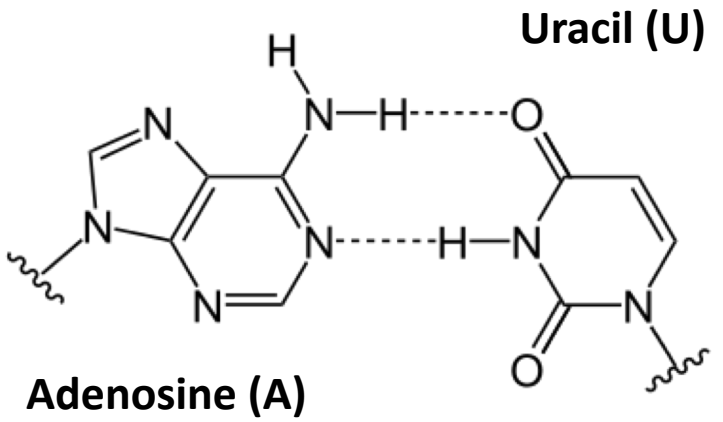
  mRNA, rRNA, tRNA, tmRNA, snRNA, snoRNA, scaRNA, aRNA, asRNA, piwiRNA, etc.

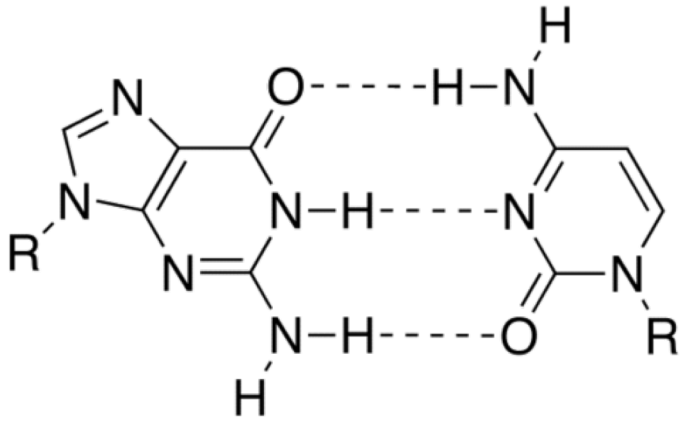
RNA – Nucleotide Complementarity

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

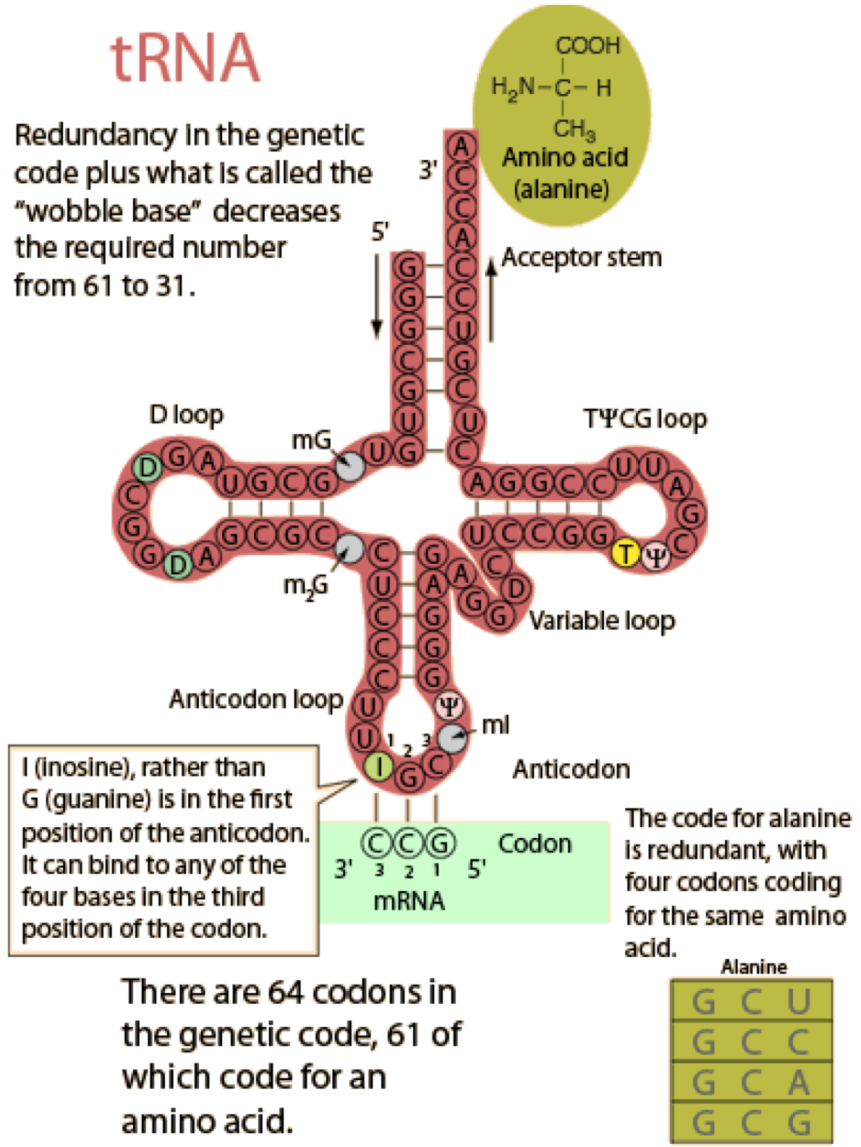
A <--> U (2 hydrogen bonds)
is slightly weaker than
G <--> C (3 hydrogen bonds)

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

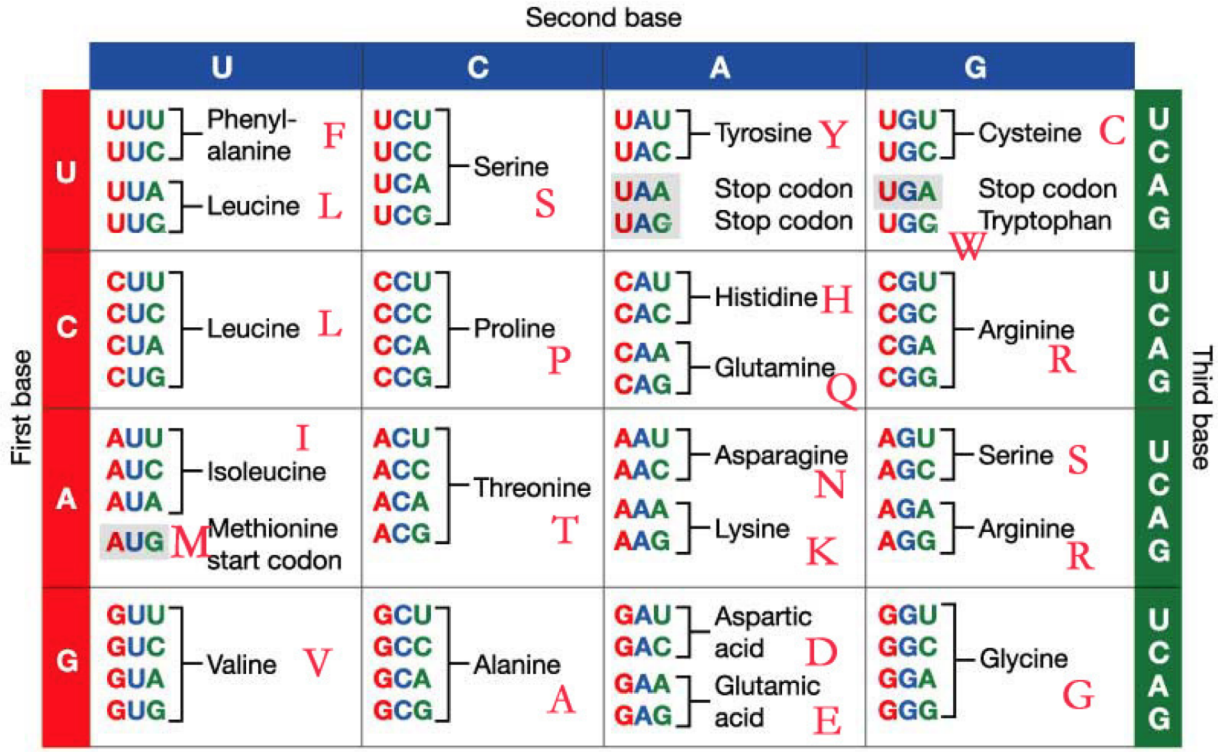




transfer RNA (tRNA) Secondary Structure



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

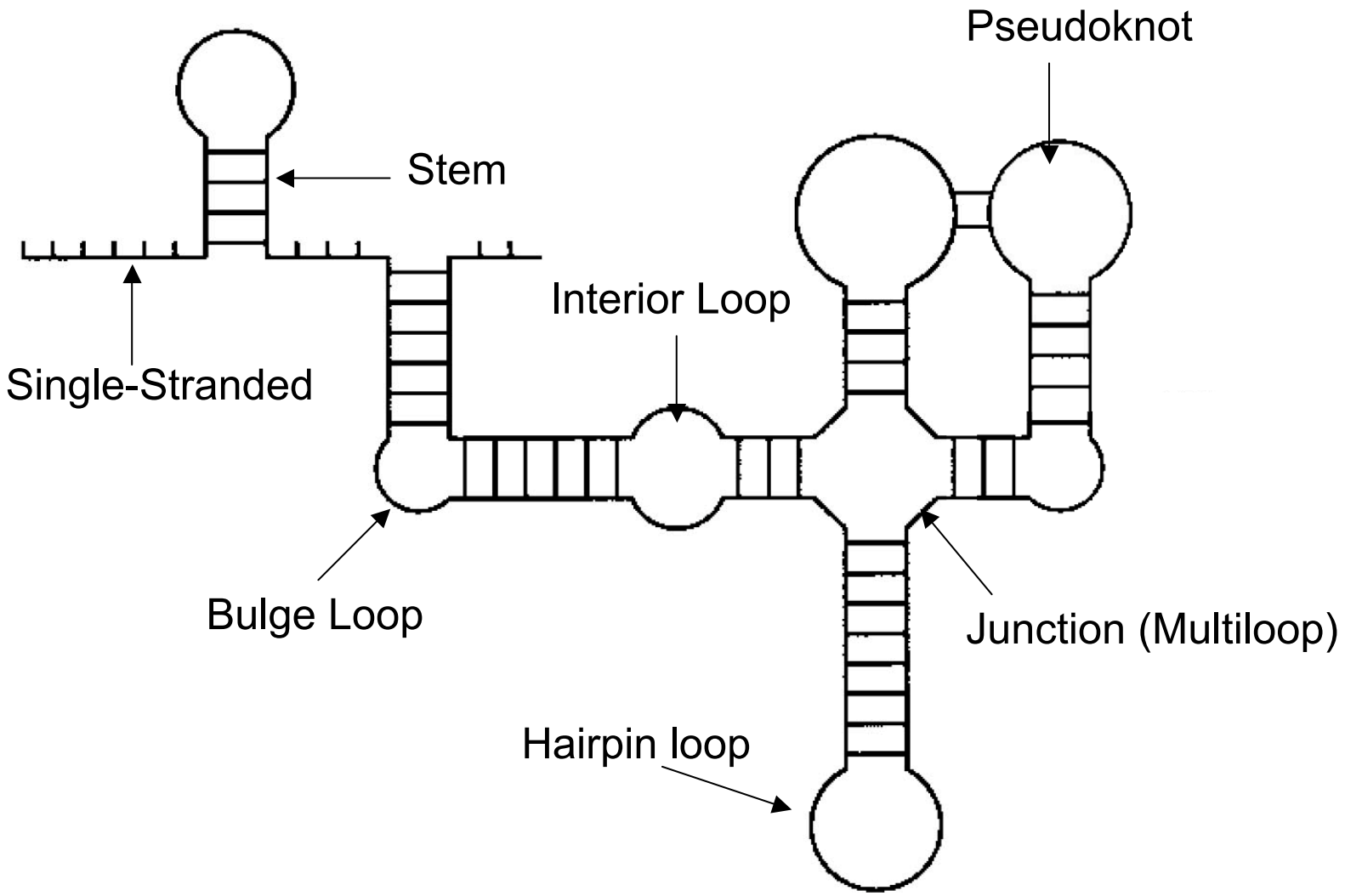


http://bioinfo.bisr.res.in/project/crat/pictures/codon.jpg

RNA Secondary Structure Elements

Each base/nucleotide participates in at most one pairing

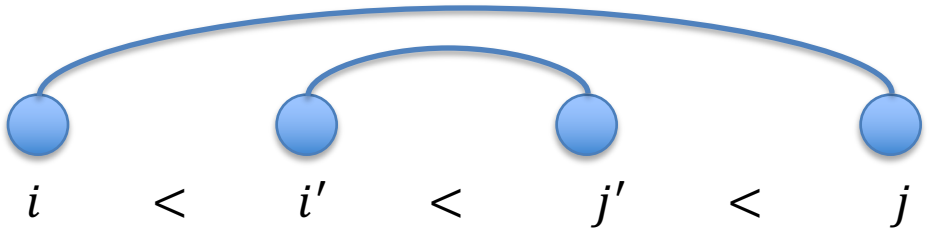
Secondary structure is determined by a set of non-overlapping base/nucleotide pairs

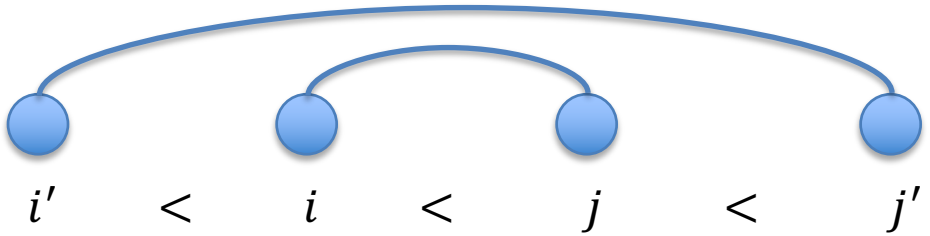


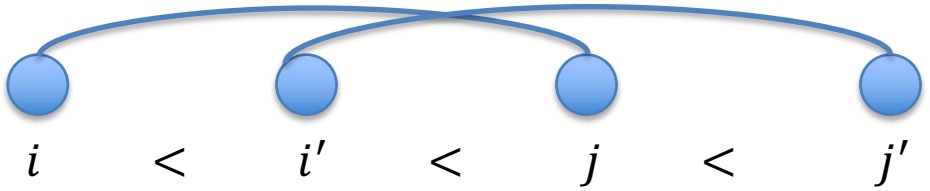
Nesting and Pseudoknot

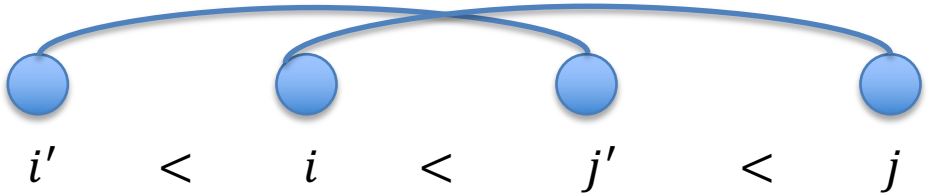
Base pairs (i, j) and (i', j') are **nested** provided
$i < i' < j' < j$ or $i' < i < j < j'$

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









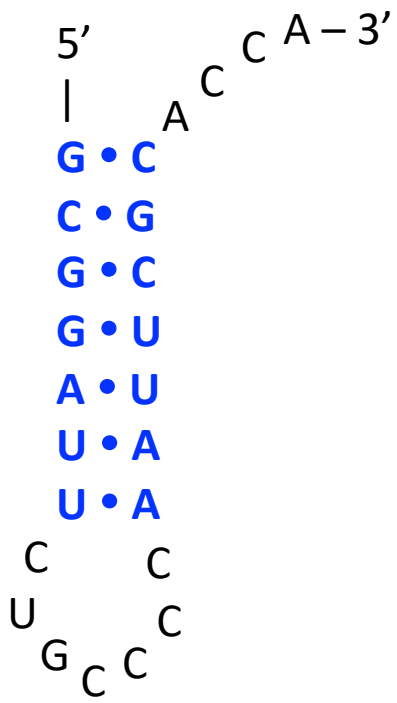
Most RNA molecules consist of nested base pairs

Nesting and Pseudoknot – Examples

Nesting

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’

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

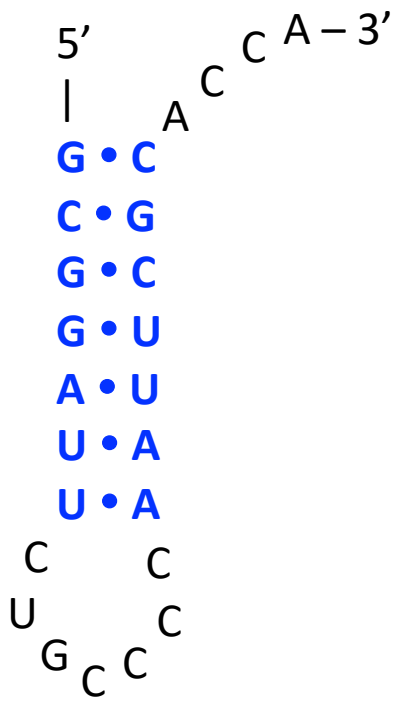


Nesting and Pseudoknot – Examples

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’

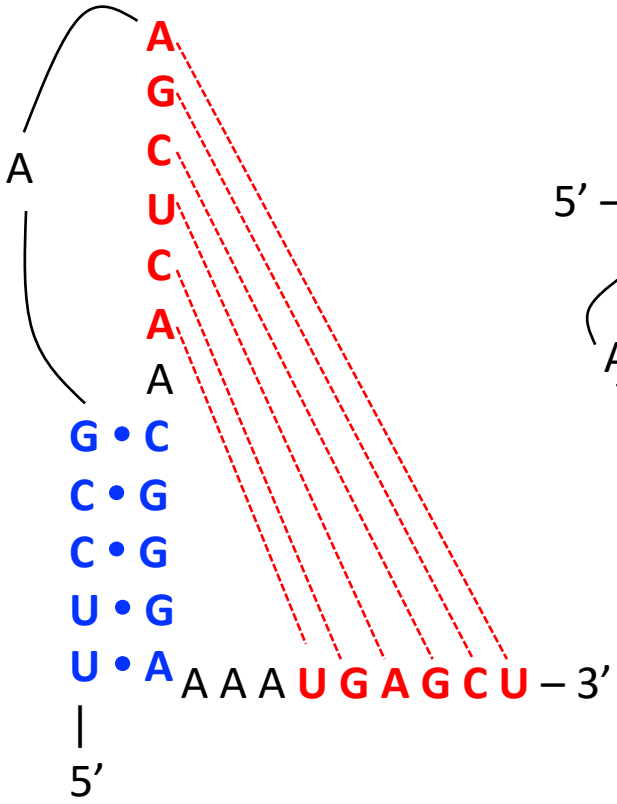
(((((((- - - - - - - -))))))) - - - -

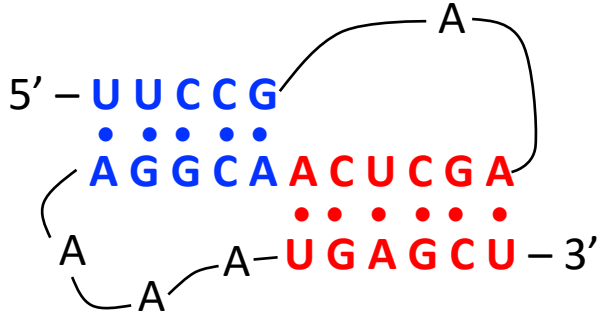


Pseudoknot

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?

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

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

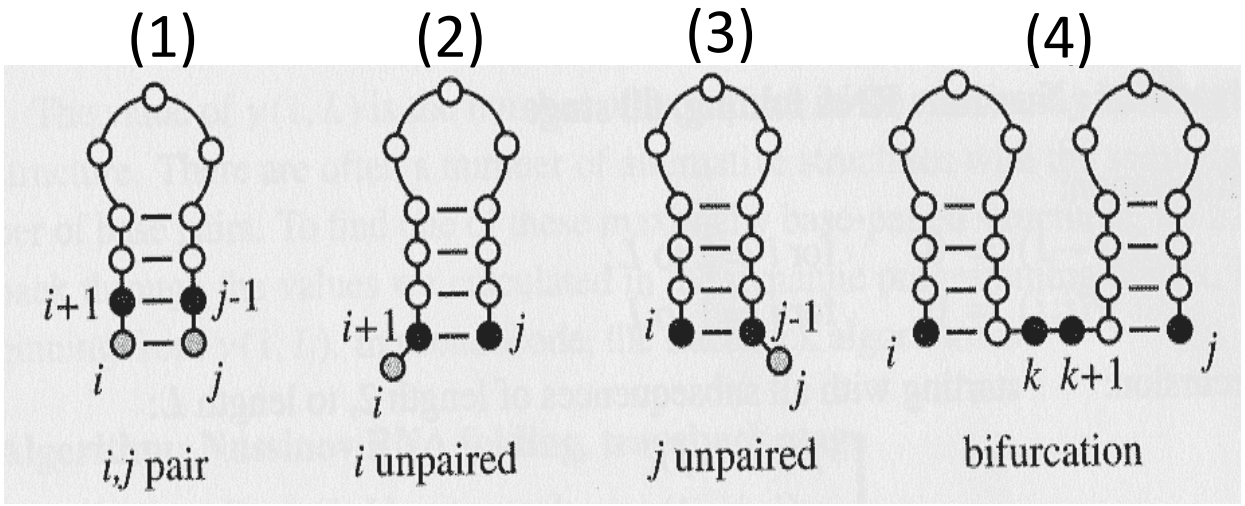
Problem: Given RNA sequence $\mathbf{v} \in \{\mathrm{A}, \mathrm{U}, \mathrm{C}, \mathrm{G}\}^n$, find a *pseudoknot-free secondary structure* with the maximum number of complementary base pairings

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

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

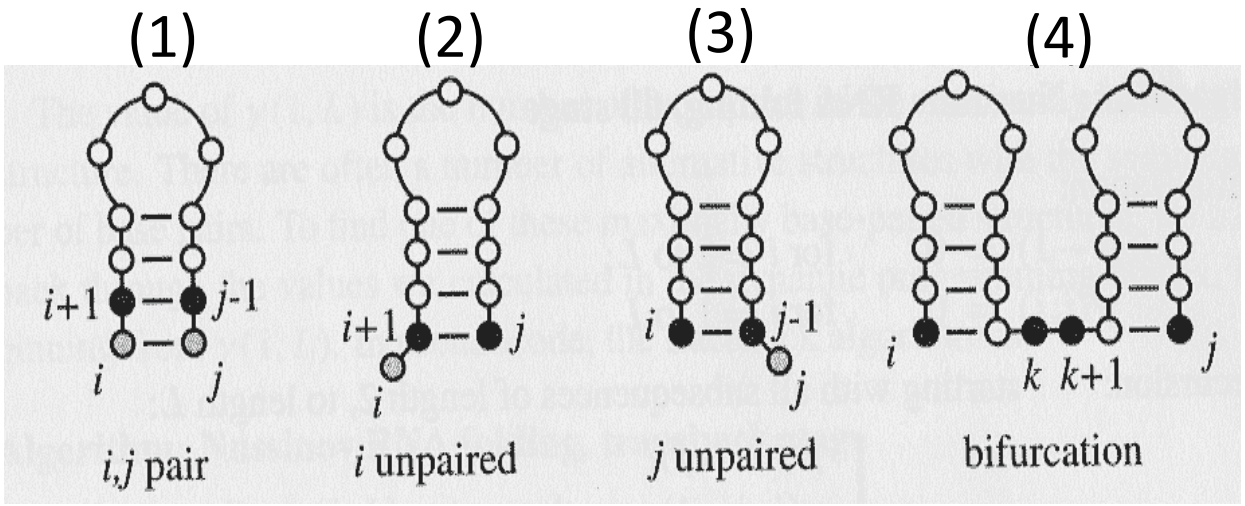
Let $s[i, j]$ denote the maximum number of pseudoknot-free complementary base pairings in subsequence $v_i, \ldots, v_j$



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

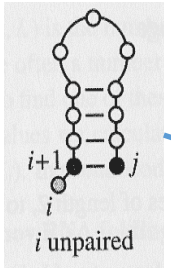
Let $s[i,j]$ denote the maximum number of pseudoknot-free complementary base pairings in subsequence $v_i, \ldots, v_j$

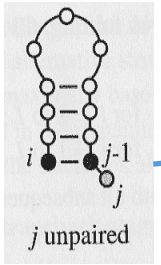


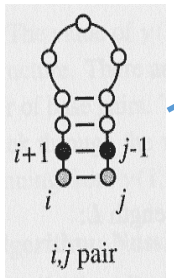
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \quad (1) \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \quad (1^*) \\ s[i+1, j], & \text{if } i < j, \quad (2) \\ s[i, j-1], & \text{if } i 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

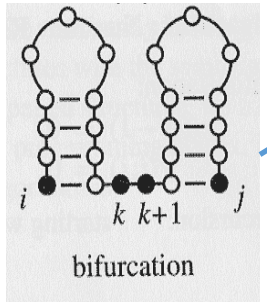
Develop Intuition [Spreadsheet/Whiteboard]

Nussinov Algorithm – Traceback Step









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)

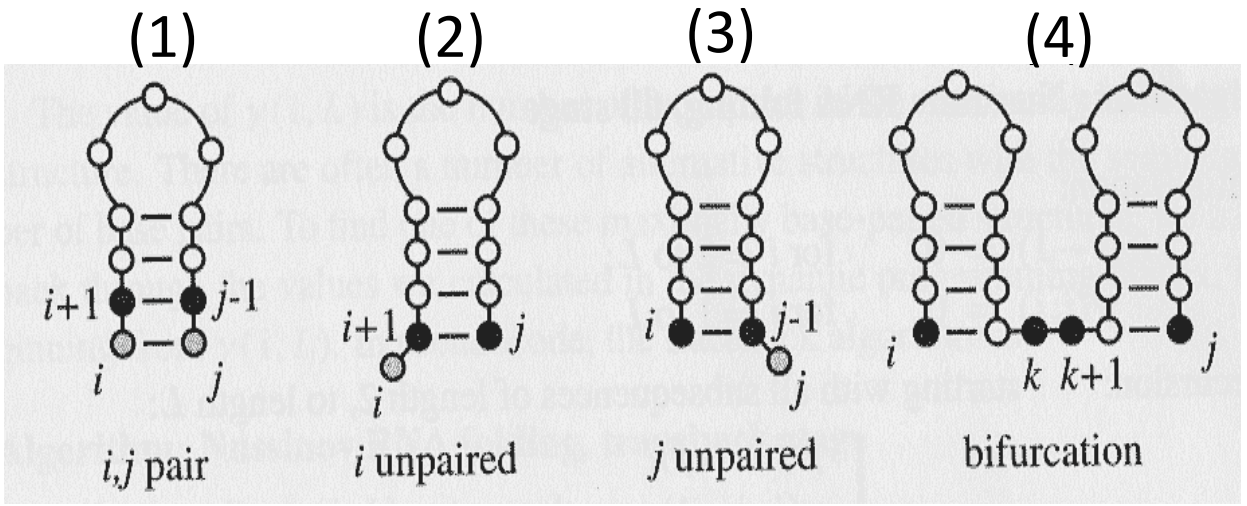
Question: Will this return one alignment? Or all alignments?

Question: Can we do this recursively?

We only need to know matches.

Nussinov Algorithm – Example

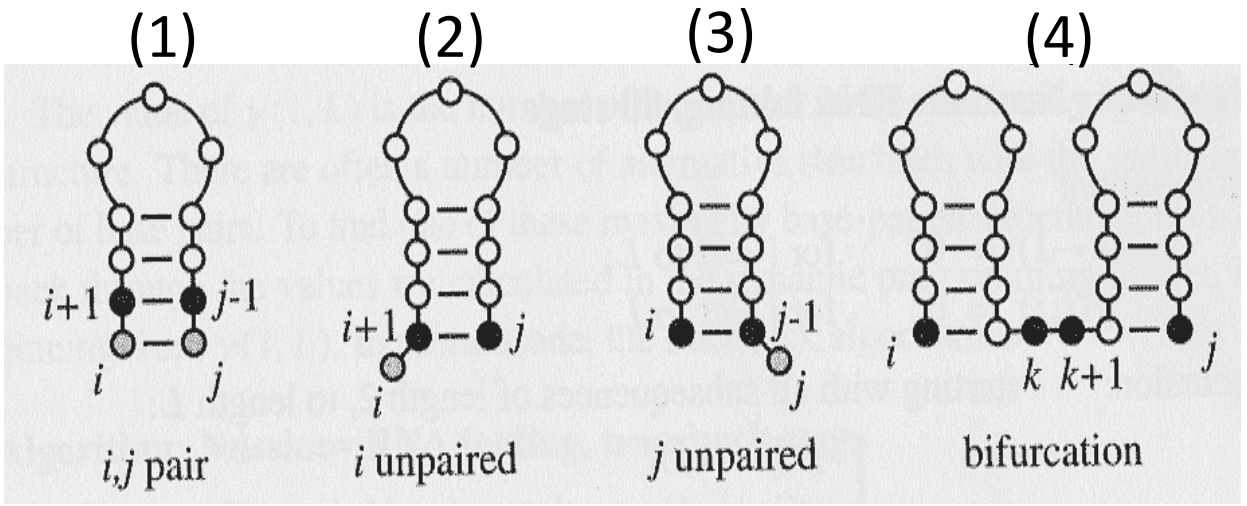
	G	G	G	A	A	A	U	C	C
G	0								
G	0	0							
G	0	0	0						
A	0	0	0	0					
A	0	0	0	0	0				
A	0	0	0	0	0	0			
U	0	0	0	0	0	0	0		
C	0	0	0	0	0	0	0	0	
C	0	0	0	0	0	0	0	0	0



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

Nussinov Algorithm – Example

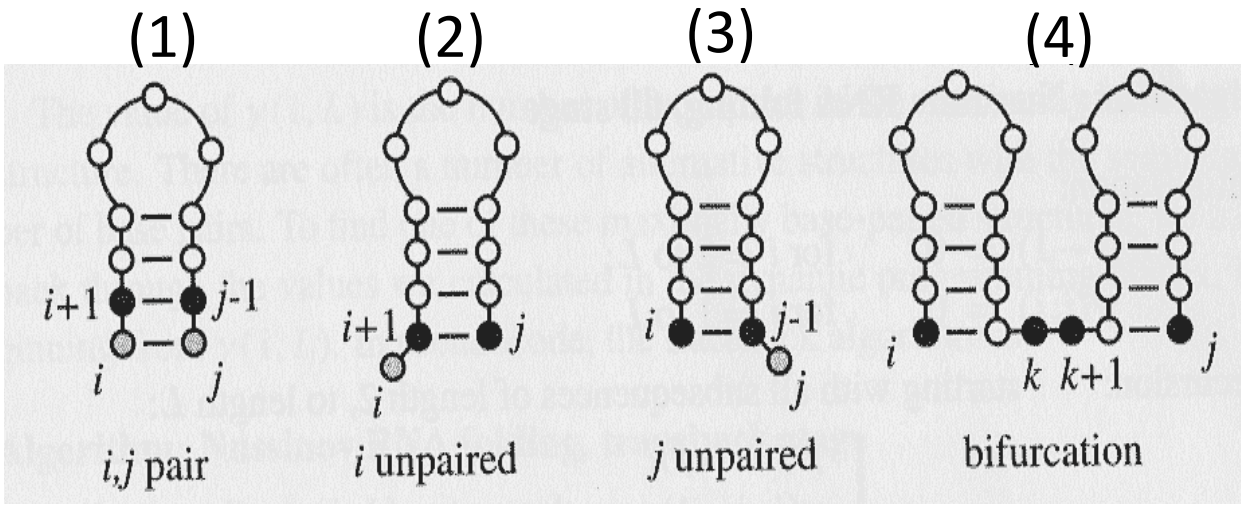
	G	G	G	A	A	A	U	C	C
G	0	0							
G	0	0	0						
G	0	0	0	0					
A	0	0	0	0	0				
A	0	0	0	0	0	0			
A	0	0	0	0	0	0	1		
U	0	0	0	0	0	0	0	0	
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



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

Nussinov Algorithm – Example

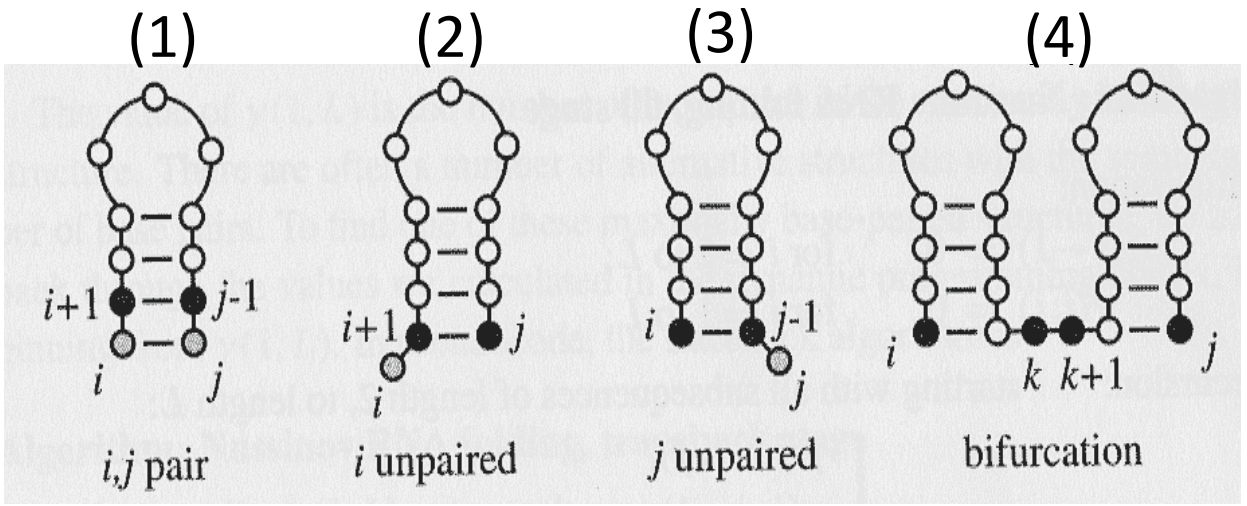
	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0



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

Nussinov Algorithm – Example

	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 – 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
(		(			(	)	)	)

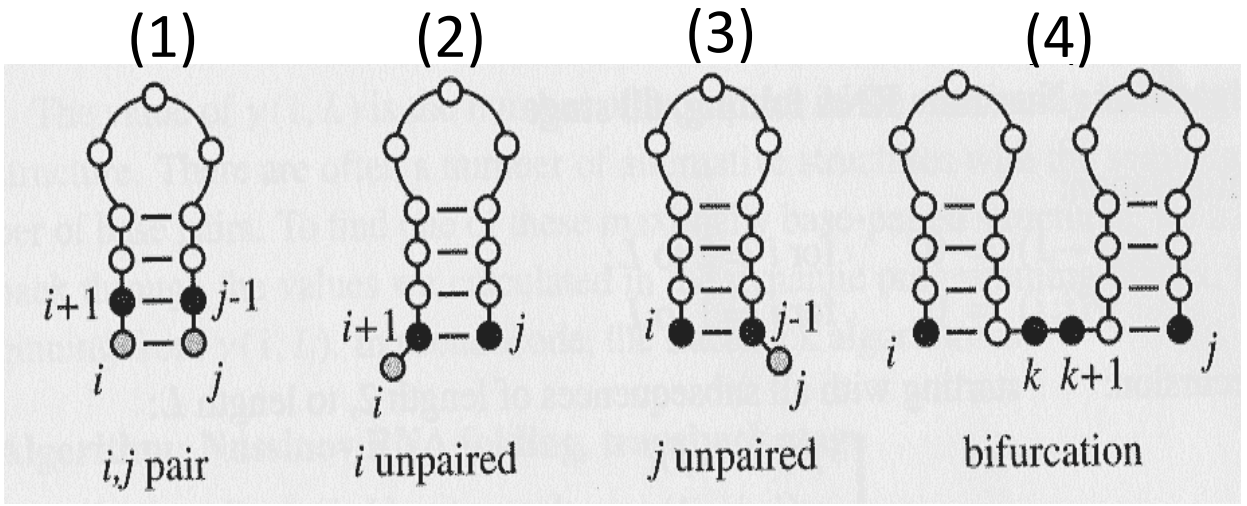
Nussinov Algorithm – Example With Bifurcation

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

Where did we come from to get here?

GCACGACG

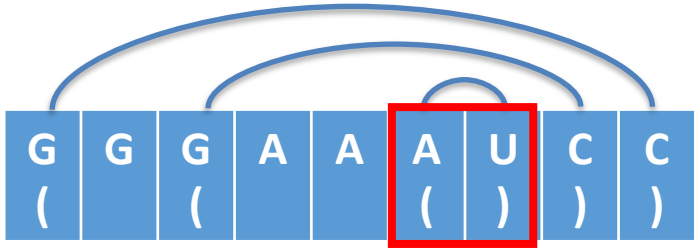
( ) . ((.))

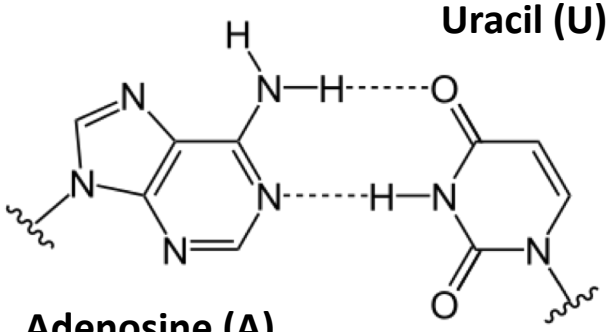


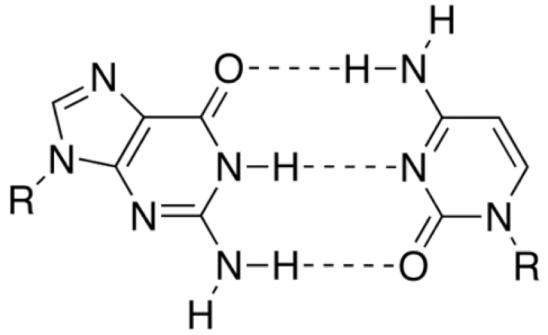
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \quad (1) \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \quad (1^*) \\ s[i+1, j], & \text{if } i < j, \quad (2) \\ s[i, j-1], & \text{if } i 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

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





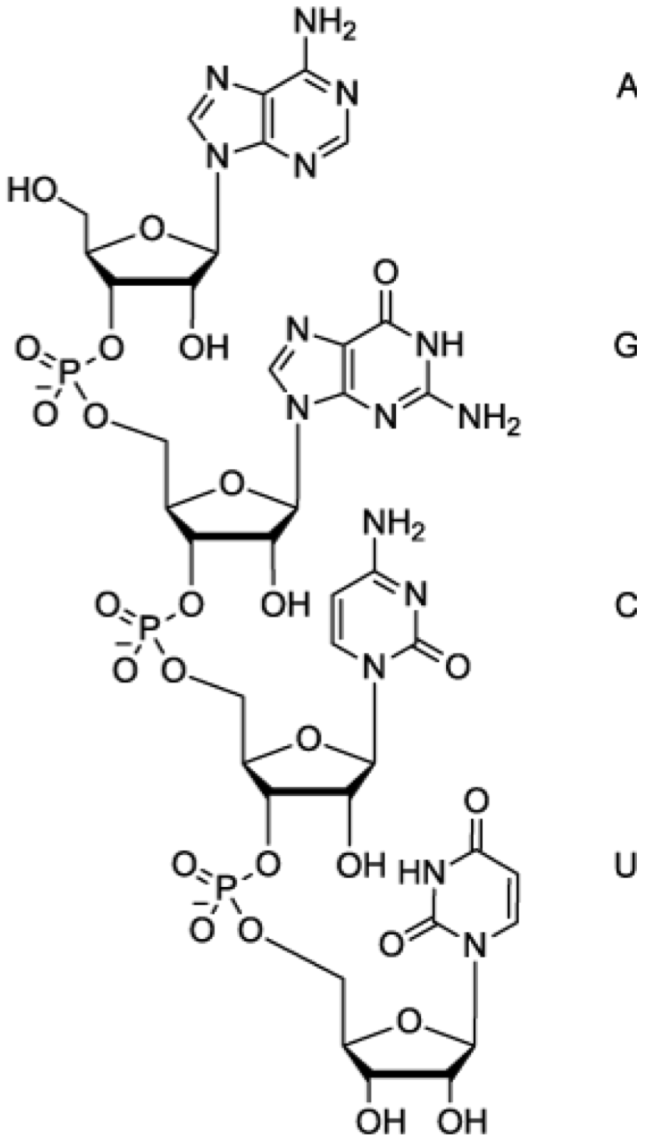


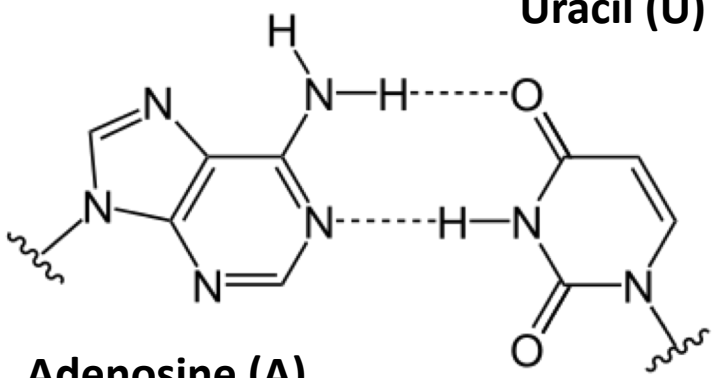
	1	2	3	4	5	6	7	8	
	G	C	A	C	G	A	C	G	
G	0	1	1	1	2	2	2	3	G 1
C	0	0	0	0	1	1	1	2	C 2
A		0	0	0	1	1	1	2	A 3
C			0	0	1	1	1	2	C 4
G				0	0	0	1	1	G 5
A					0	0	0	1	A 6
C						0	0	1	C 7
G							0	0	G 8

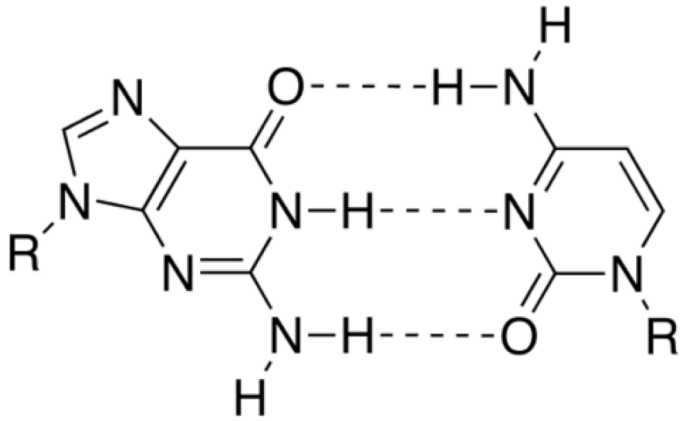
GCACGACG

( ) . ((.))

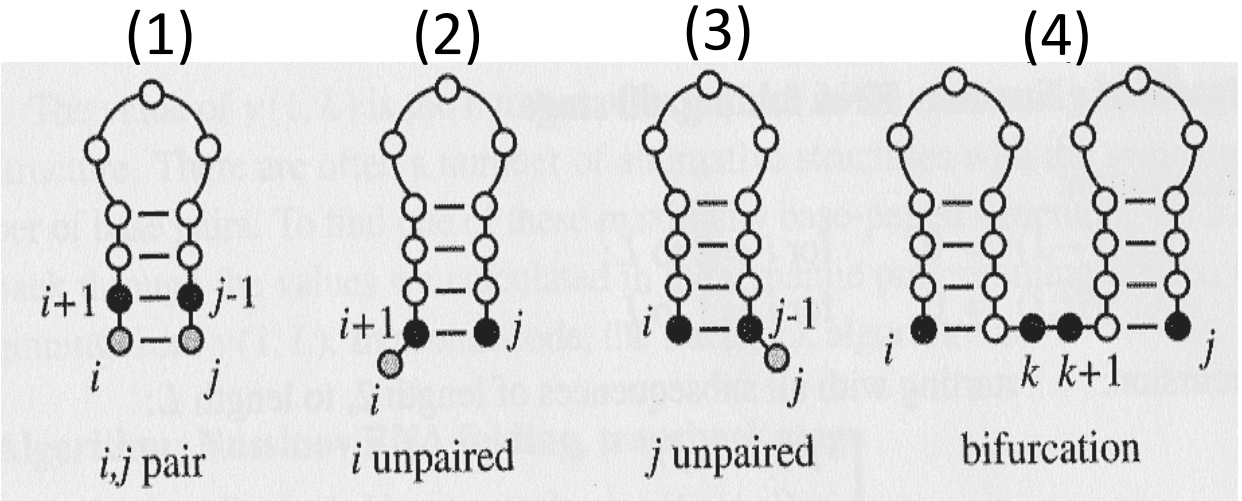
Extension: Hairpin Loops with Minimum Length ℓ







Extension: Hairpin Loops with Minimum Length ℓ



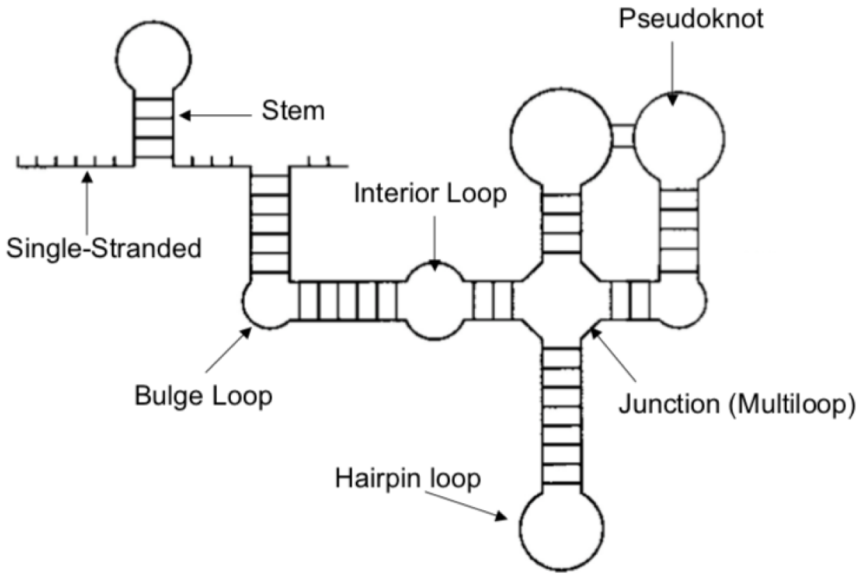
$$s[i,j] = \max \begin{cases} 0, & \text{if } i+\ell \geq j, \\ s[i+1,j-1]+1, & \text{if } i+\ell < j \text{ and } (v_i, v_j) \in \Gamma, \quad (1) \\ s[i+1,j-1], & \text{if } i+\ell < j \text{ and } (v_i, v_j) \notin \Gamma, \quad (1^*) \\ s[i+1,j], & \text{if } i+\ell < j, \quad (2) \\ s[i,j-1], &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

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
- 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.]