CSE8803/CX4803

Machine Learning in Computational Biology

Lecture 12:
RNA Secondary Structure Prediction

Yunan Luo

Slides adapted from UIUC CS466 by Mohammed El-Kebir

Learning from structure

Date	Topic	Contents
1/10/2022	Introduction	Course intro & how to present papers
1/12/2022	Learning from sequence data	Dynamic programming & sequence alignment I
1/17/2022		No class (MLK Day)
1/19/2022		Sequence alignment II
1/24/2022		HMM & gene/motif finding
1/26/2022		HMM & Profile HMM
1/31/2022		Deep learning for DNA/protein sequence
2/2/2022	Learning from high-dim data	Learn from high-dim data: PCA, autoencoder & VAE
2/7/2022		Learn from high-dim data: MDS, tSNE, UMAP
2/9/2022		Clustering I
2/14/2022		Clustering II
2/16/2022		Clustering III
2/21/2022	Phase 1 presentations	Student presentation 1-3
2/23/2022		Student presentation 4-6
2/28/2022	Learning from structure data	RNA structure prediction
3/2/2022		Deep learning for structures (protein structure prediction)
3/7/2022	Phase 2 presentations	Student presentation 7-9
3/9/2022	Learning from network data	Network basics & traditinal ML for graphs
3/14/2022		Network embeddings
3/16/2022	Phase 3 presentations	Student presentation 10-12
3/21/2022	Spring break	No class (Spring Break)
3/23/2022		No class (Spring Break)
3/28/2022	Learning from network data	Graphical Models
3/30/2022		Deep learning for networks (graph neural networks)

Central Dogma of Molecular Biology

Three fundamental molecules:

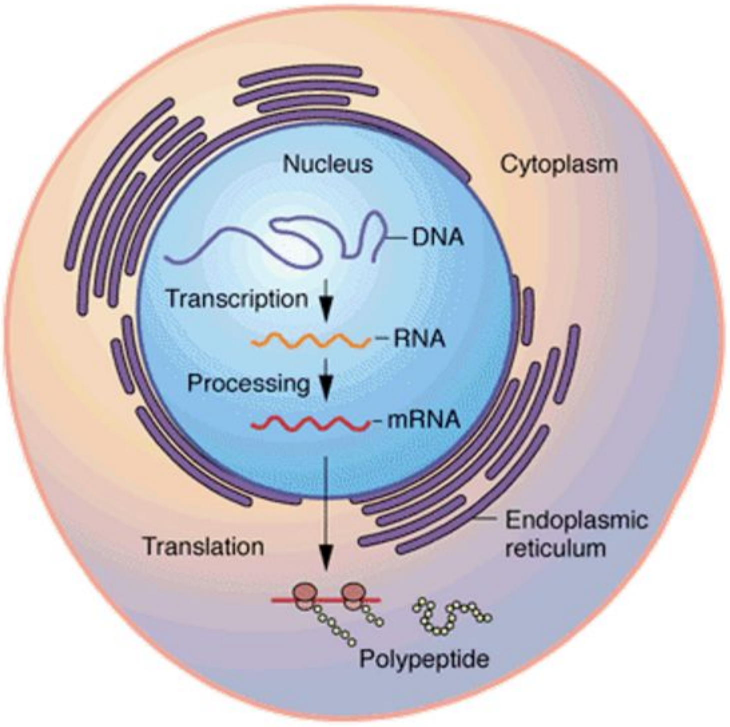
1. **DNA**
   Information storage.
2. **RNA**

   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.)

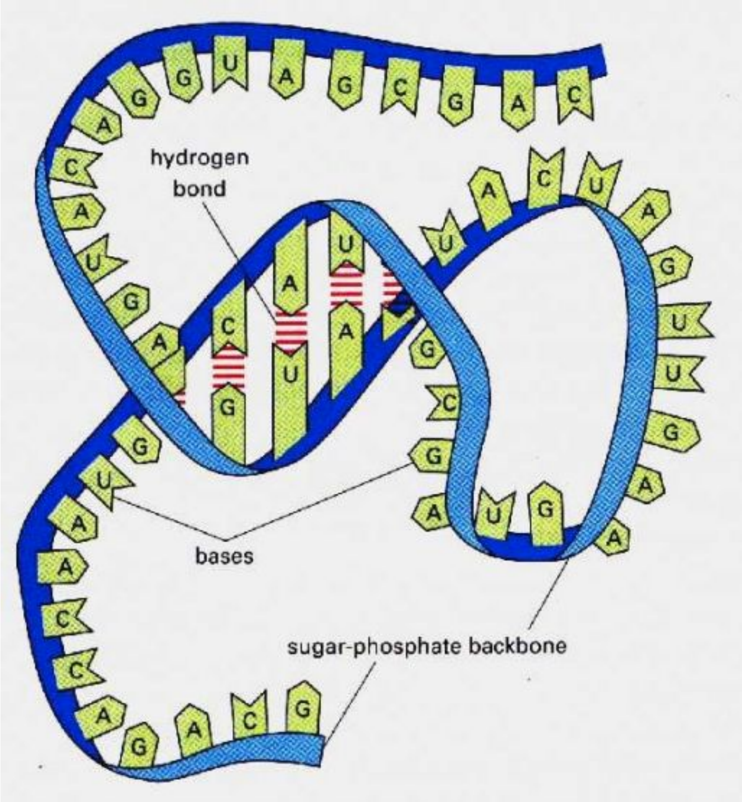
DNA -> RNA -> Protein

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.

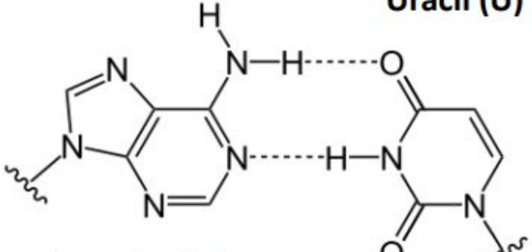
RNA – Nucleotide Complementarity

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

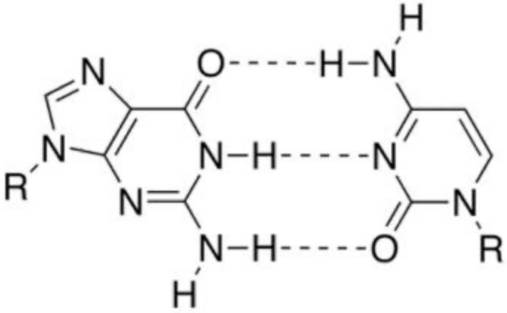
A <--> U (2 hydrogen bonds)
is slightly weaker than
G <--> C (3 hydrogen bonds)

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

Uracil (U)



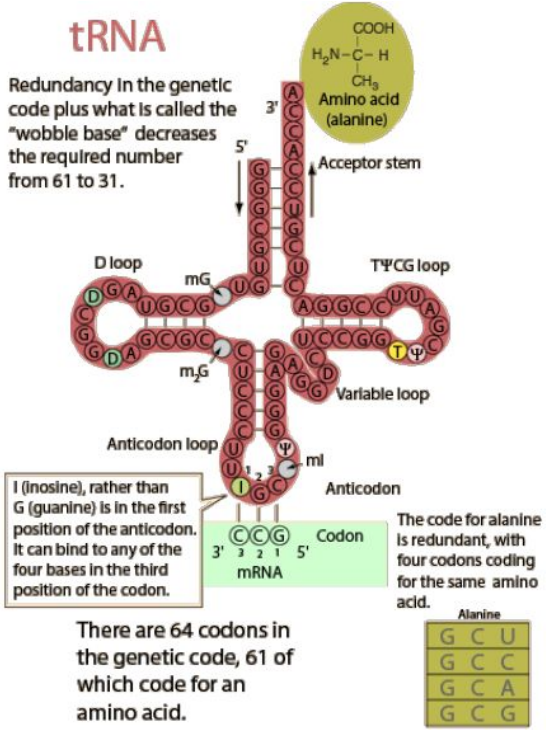
Adenosine (A)



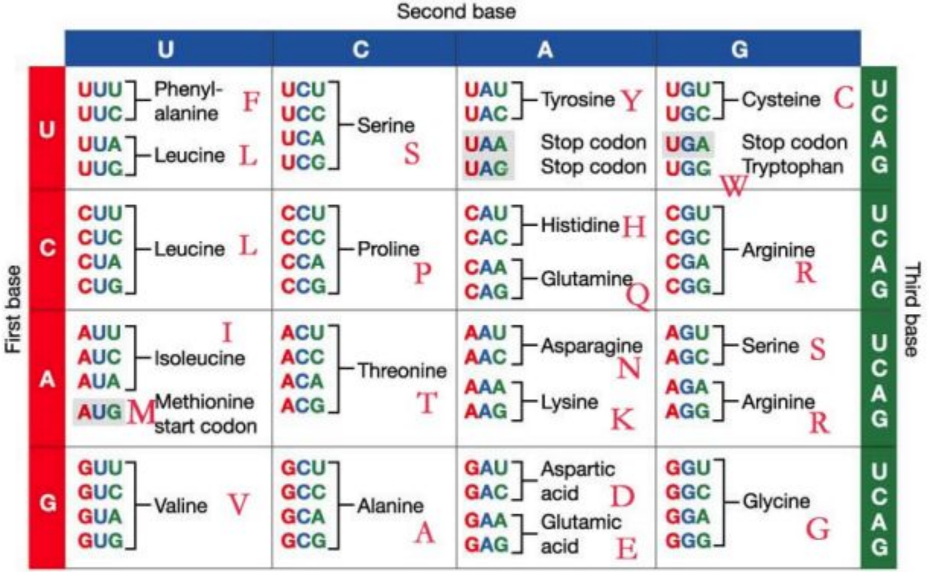
Guanine (G)

Cytosine (C)

Transfer RNA (tRNA) Secondary Structure



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



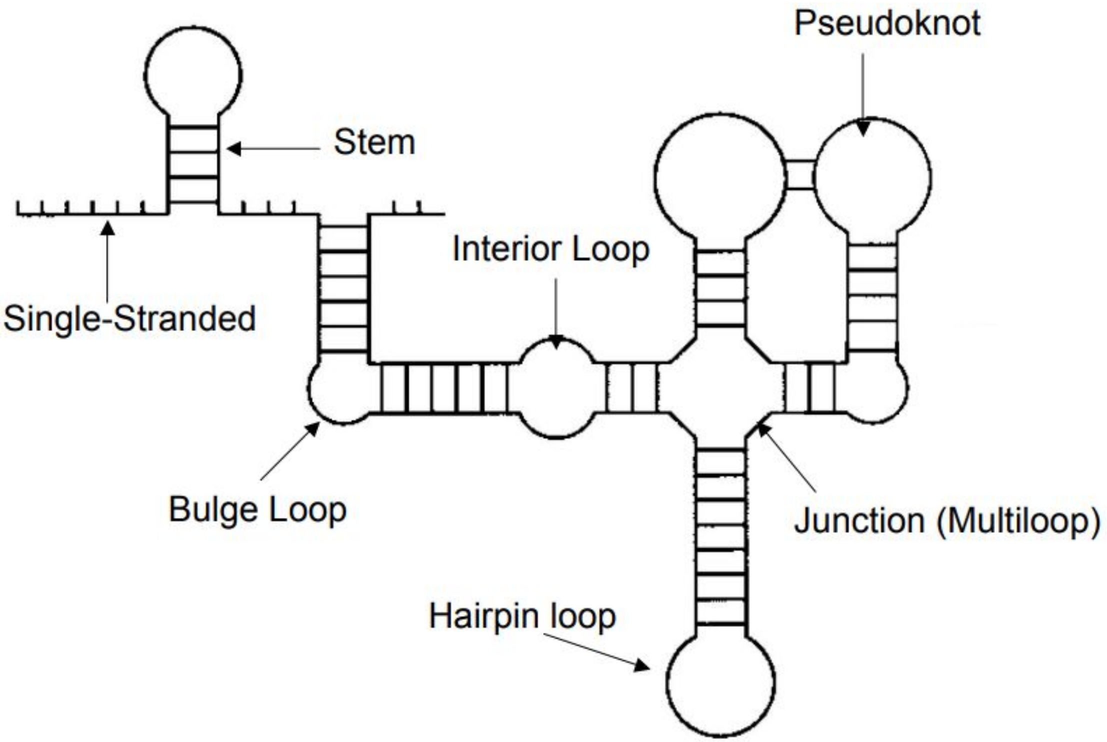
First base	Second base: U	C	A	G	Third base
U	UUU, UUC: Phenyl-alanine F UUA, UUG: Leucine L	UCU, UCC, UCA, UCG: Serine S	UAU, UAC: Tyrosine Y UAA: Stop codon UAG: Stop codon	UGU, UGC: Cysteine C UGA: Stop codon UGG: Tryptophan W	U C A G
C	CUU, CUC, CUA, CUG: Leucine L	CCU, CCC, CCA, CCG: Proline P	CAU, CAC: Histidine H CAA, CAG: Glutamine Q	CGU, CGC, CGA, CGG: Arginine R	U C A G
A	AUU, AUC, AUA: Isoleucine I AUG: Methionine start codon M	ACU, ACC, ACA, ACG: Threonine T	AAU, AAC: Asparagine N AAA, AAG: Lysine K	AGU, AGC: Serine S AGA, AGG: Arginine R	U C A G
G	GUU, GUC, GUA, GUG: Valine V	GCU, GCC, GCA, GCG: Alanine A	GAU, GAC: Aspartic acid D GAA, GAG: Glutamic acid E	GGU, GGC, GGA, GGG: 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

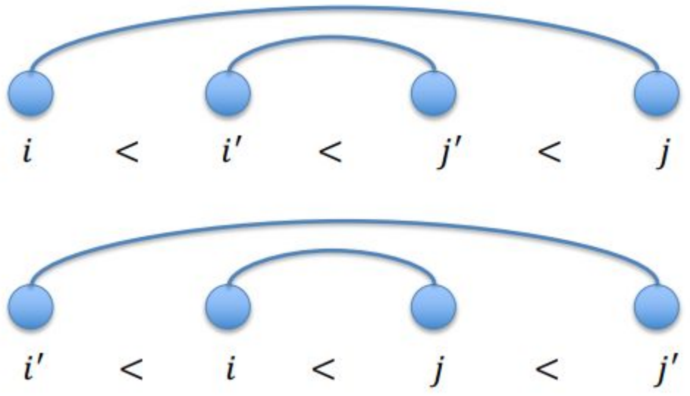
Secondary structure is determined by a set of non-overlapping base/nucleotide pairs

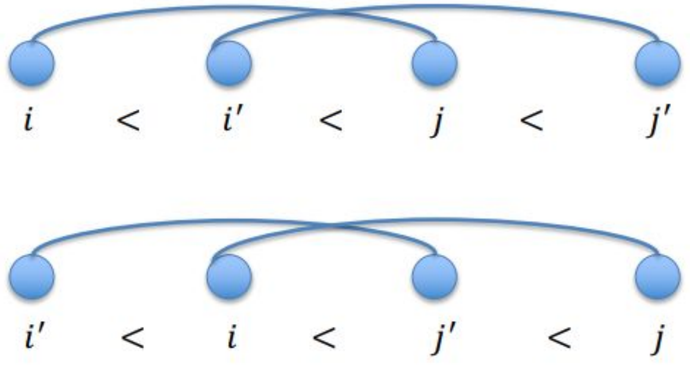


Nesting and Pseudoknot

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

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

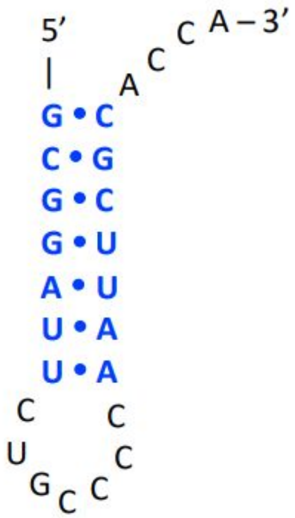


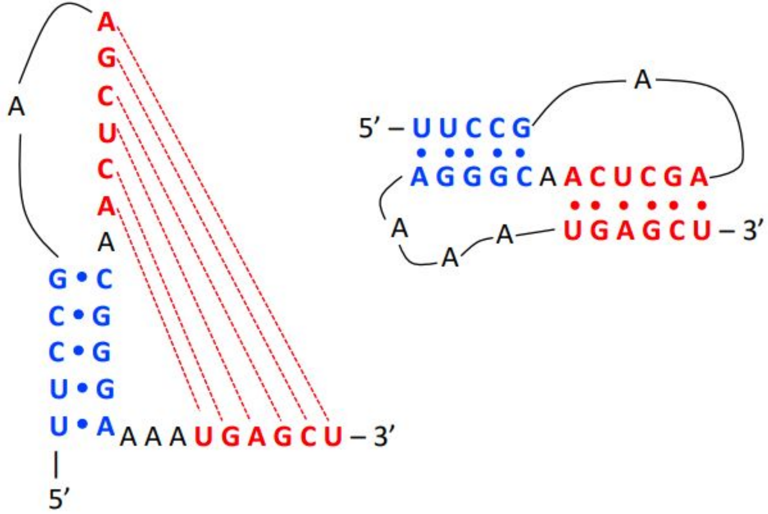


Most RNA molecules consist of nested base pairs

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’



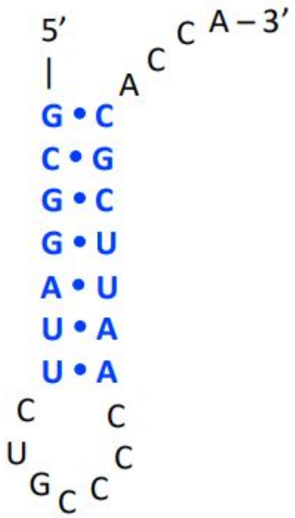


Nesting and Pseudoknot -- Examples

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’

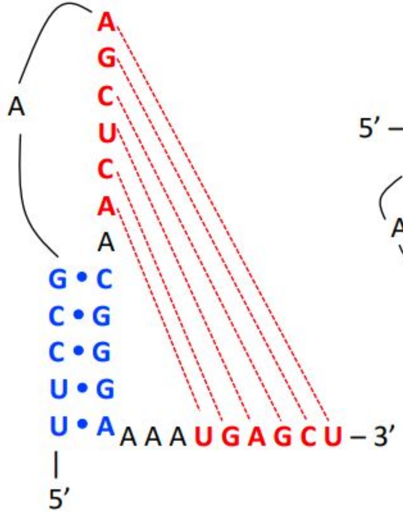
(((((((- - - - - - -))))))) - - - -

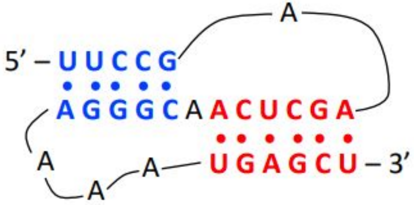


Pseudoknot

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 complementary base pairs

Question: How to find maximum number of such pairs?

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

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

Secondary structure is determined by a set of non-overlapping complementary 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 complementary 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 $\mathbf{v} \in \{\mathrm{A, U, C, G}\}^n$, find a *pseudoknot-free secondary structure* with the maximum number of complementary base pairings

Nussinov Algorithm – Dynamic Programming

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

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

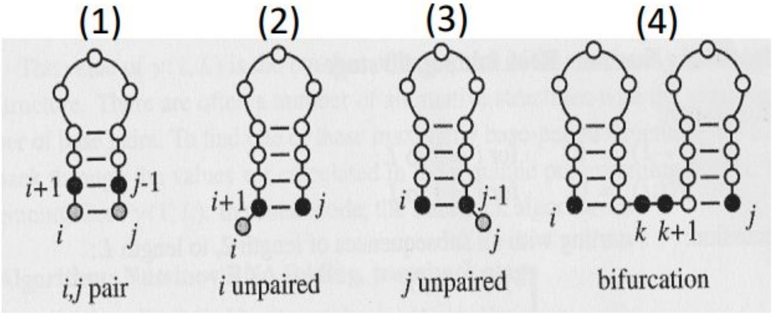
Problem: Given RNA sequence $\mathbf{v} \in \{\mathrm{A, U, C, G}\}^{\mathrm{n}}$, find a *pseudoknot-free secondary structure* with the maximum number of complementary base pairings

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

Nussinov Algorithm – Dynamic Programming

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

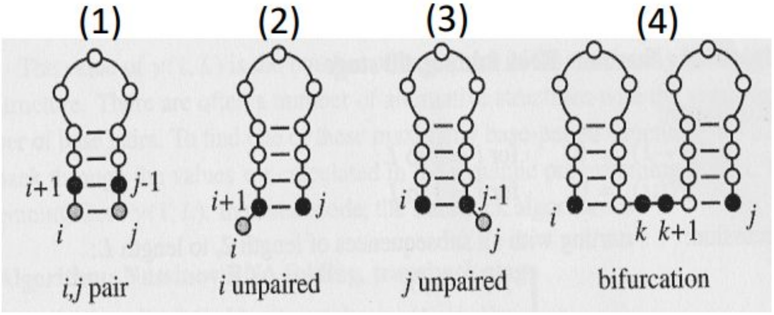
Let $\boldsymbol{s}[\boldsymbol{i},\boldsymbol{j}]$ denote the maximum number of pseudoknot-free complementary base pairings in subsequence $\boldsymbol{v}_{\boldsymbol{i}}, \ldots, \boldsymbol{v}_{\boldsymbol{j}}$



Nussinov Algorithm – Dynamic Programming

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



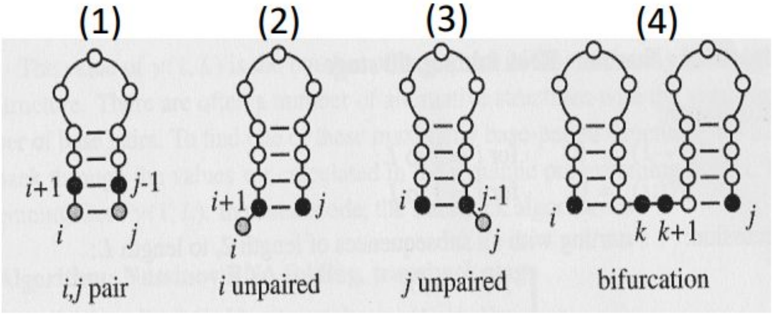
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \quad (1) \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \quad (1^*) \\ s[i+1, j], & \text{if } i < j, \quad (2) \\ s[i, j-1], & \text{if } i 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

Γ = {(G,C), (C,G), (A,U), (U,A)}

Nussinov Algorithm – Dynamic Programming

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



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

Question: Which case is redundant?

Develop intuition

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

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

Develop intuition

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

(	(	(	(	(	-	-	-	-	-	-	-	-	-	-	-	)	-	-	)	)	)	)		
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					0								C	11
0	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			0								U	13
0	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	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

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

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

Develop intuition

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

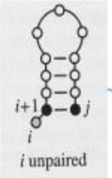
(	(	(	(	(	(	(	-	-	-	)	)	-	(	-	)	)	-	-	)	)	)	)		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
G	**C**	**U**	**C**	**G**	**G**	**G**	**U**	**U**	**C**	**C**	**C**	**U**	**A**	**U**	**U**	**C**	**A**	**A**	**G**	**A**	**G**	**C**		
0																							**G**	1
0	0																			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

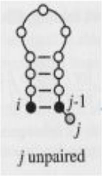
Develop intuition

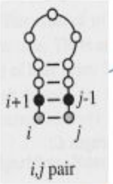
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1,j-1]+1, & \text{if } i < j \text{ and } (v_i,v_j) \in \Gamma, \\ s[i+1,j-1], & \text{if } i < j \text{ and } (v_i,v_j) \notin \Gamma, \\ s[i+1,j], & \text{if } i < j, \\ s[i,j-1], & \text{if } i < j, \\ \max_{i<k<j}\{s[i,k]+s[k+1,j]\},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

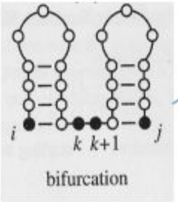
(	(	(	(	(	(	(	-	-	-	)	)	-	(	-	)	)	-	-	)	)	)	)		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
G	C	U	C	G	G	G	U	U	C	C	C	U	A	U	U	C	A	A	G	A	G	C		
0																						8	G	1
0	0																				7		C	2
0	0	0																		6			U	3
0	0	0	0																5				C	4
0	0	0	0	0											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










```
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)
```

Filling in the matrix

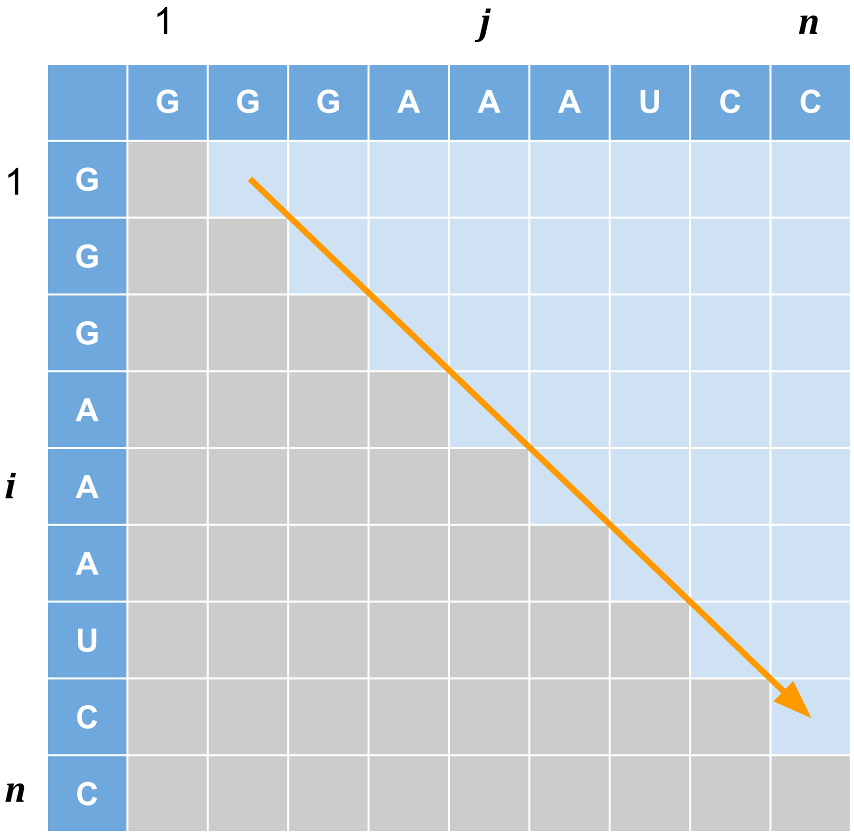
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1, j-1] + 1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \\ s[i+1, j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \\ s[i+1, j], & \text{if } i < j, \\ s[i, j-1], & \text{if } i < j, \\ \max_{i<k<j}\{s[i,k] + s[k+1,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

		1				*j*				*n*
		G	G	G	A	A	A	U	C	C
1	G									
	G									
	G									
	A									
i	A									
	A									
	U									
	C									
n	C									

Filling in the matrix

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

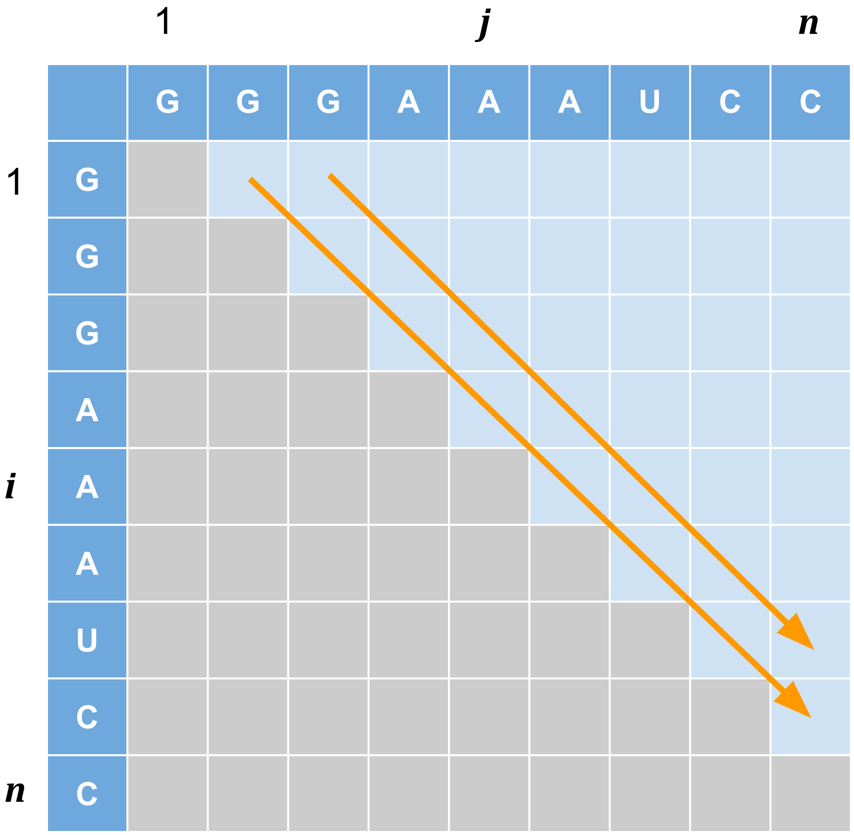
In order of increasing ***j* - *i***



Filling in the matrix

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

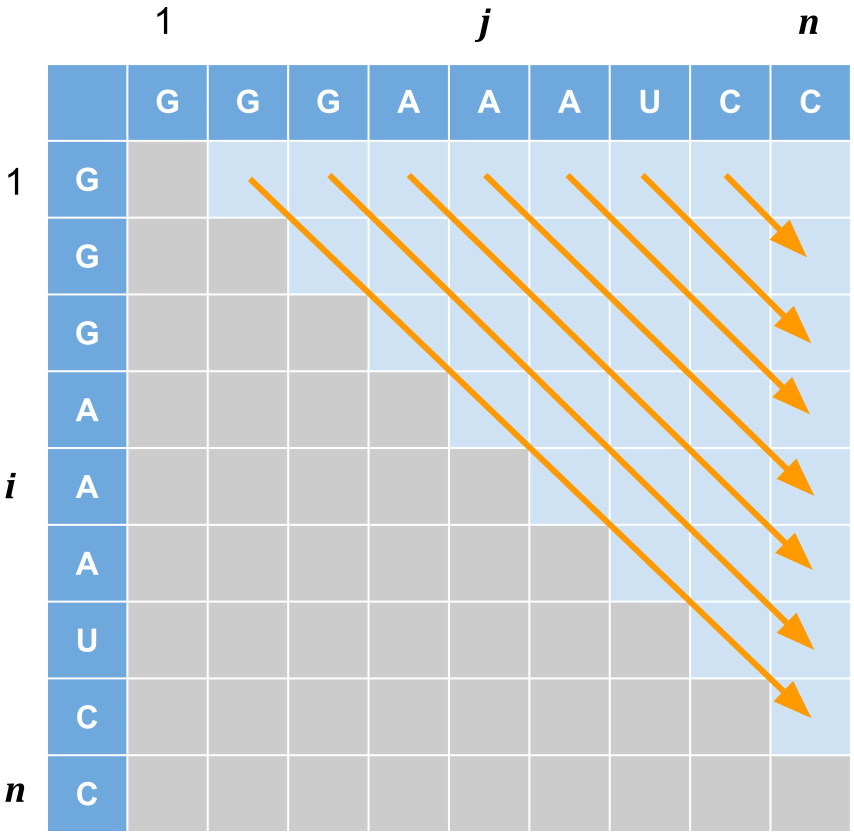
In order of increasing ***j* - *i***



Filling in the matrix

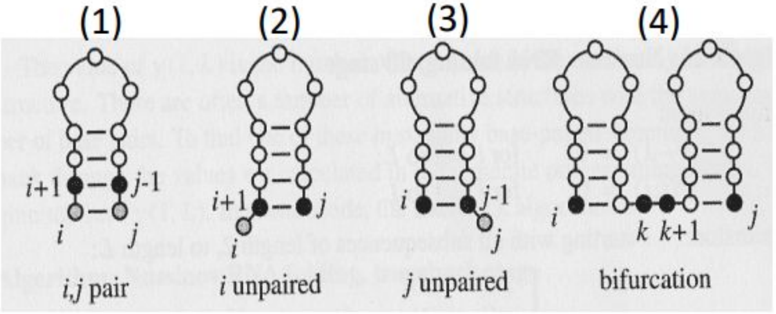
$$s[i,j] = \max \begin{cases} 0, & \text{if } i \geq j, \\ s[i+1,j-1]+1, & \text{if } i < j \text{ and } (v_i, v_j) \in \Gamma, \\ s[i+1,j-1], & \text{if } i < j \text{ and } (v_i, v_j) \notin \Gamma, \\ s[i+1,j], & \text{if } i < j, \\ s[i,j-1], & \text{if } i < j, \\ \max_{i<k<j}\{s[i,k]+s[k+1,j]\},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

In order of increasing ***j* - *i***



Nussinov Algorithm – Example

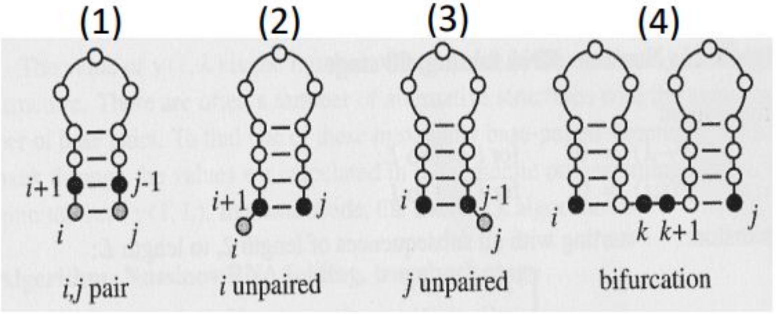
		1				*j*				*n*
		G	G	G	A	A	A	U	C	C
1	G	0								
	G	0	0							
	G	0	0	0						
	A	0	0	0	0					
i	A	0	0	0	0	0				
	A	0	0	0	0	0	0			
	U	0	0	0	0	0	0	0		
	C	0	0	0	0	0	0	0	0	
n	C	0	0	0	0	0	0	0	0	0



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

Nussinov Algorithm – Example

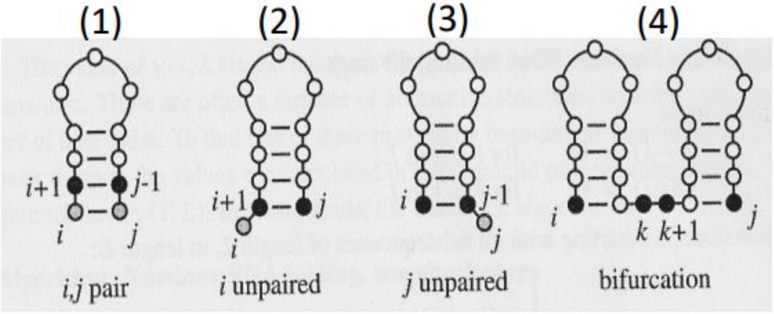
		1				*j*				*n*
		G	G	G	A	A	A	U	C	C
1	G	0	0							
	G	0	0	0						
	G	0	0	0	0					
	A	0	0	0	0	0				
i	A	0	0	0	0	0	0			
	A	0	0	0	0	0	0	1		
	U	0	0	0	0	0	0	0	0	
	C	0	0	0	0	0	0	0	0	0
n	C	0	0	0	0	0	0	0	0	0



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

Nussinov Algorithm – Example

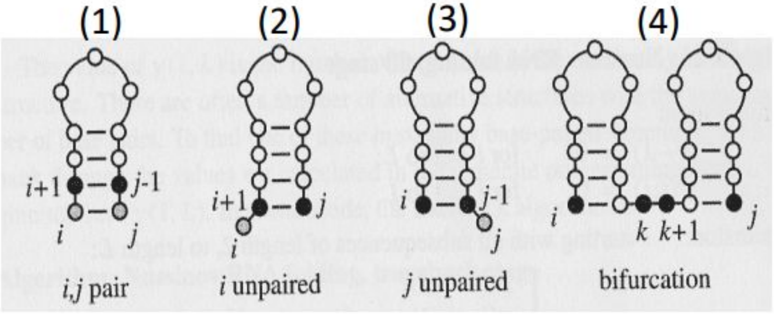
		1			j				n
		G	G	G	A	A	A	U	C	C
1	G	0	0	0						
	G	0	0	0	0					
	G	0	0	0	0	0				
	A	0	0	0	0	0	0			
i	A	0	0	0	0	0	0	1		
	A	0	0	0	0	0	0	1	1	
	U	0	0	0	0	0	0	0	0	0
	C	0	0	0	0	0	0	0	0	0
n	C	0	0	0	0	0	0	0	0	0



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

Nussinov Algorithm – Example

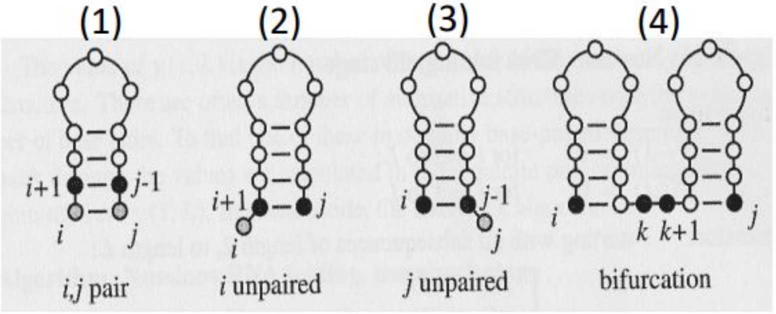
		1				j				n
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0					
	G	0	0	0	0	0				
	G	0	0	0	0	0	0			
	A	0	0	0	0	0	0	1		
i	A	0	0	0	0	0	0	1	1	
	A	0	0	0	0	0	0	1	1	1
	U	0	0	0	0	0	0	0	0	0
	C	0	0	0	0	0	0	0	0	0
n	C	0	0	0	0	0	0	0	0	0



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

Nussinov Algorithm – Example

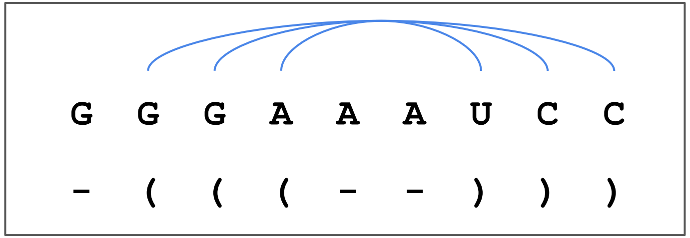
		1				*j*				*n*
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
	G	0	0	0	0	0	0	1	2	3
	G	0	0	0	0	0	0	1	2	2
	A	0	0	0	0	0	0	1	1	1
i	A	0	0	0	0	0	0	1	1	1
	A	0	0	0	0	0	0	1	1	1
	U	0	0	0	0	0	0	0	0	0
	C	0	0	0	0	0	0	0	0	0
n	C	0	0	0	0	0	0	0	0	0



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

Nussinov Algorithm – Example

		1				j				n
		G	G	G	A	A	A	U	C	C
1	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
i	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
n	C	0	0	0	0	0	0	0	0	0



Nussinov Algorithm – Example With Bifurcation

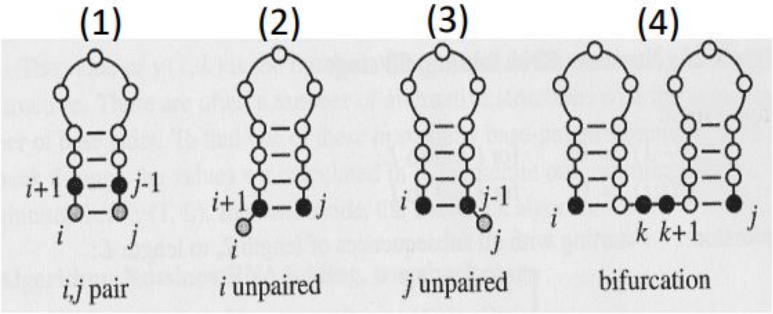
1 *j* *n*

		G	C	A	C	G	A	C	G
1	G	0	1	1	1	2	2	2	3
	C	0	0	0	0	1	1	1	2
	A	0	0	0	0	1	1	1	2
	C	0	0	0	0	1	1	1	2
i	G	0	0	0	0	0	0	1	1
	A	0	0	0	0	0	0	0	1
	G	0	0	0	0	0	0	0	1
n	C	0	0	0	0	0	0	0	0

GCACGACG

().((.))

Where did we come from to get here?



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

Nussinov Algorithm – Example With Bifurcation

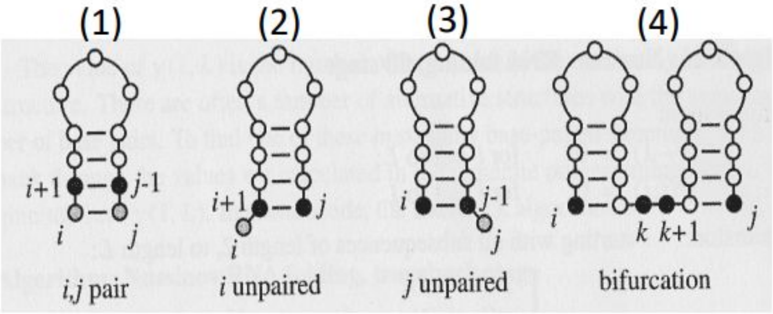
1 *j* *n*

1 *i* *n*

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

GCACGACG

().((.))



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

Nussinov Algorithm – Alternative Solutions

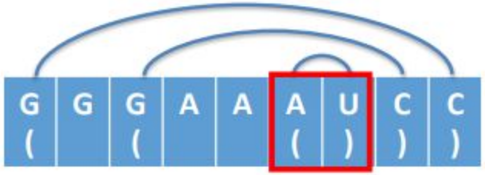
	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0

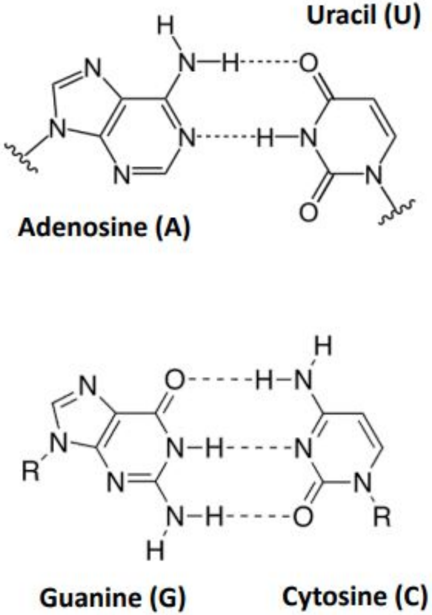
	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0

	G	G	G	A	A	A	U	C	C
G	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?

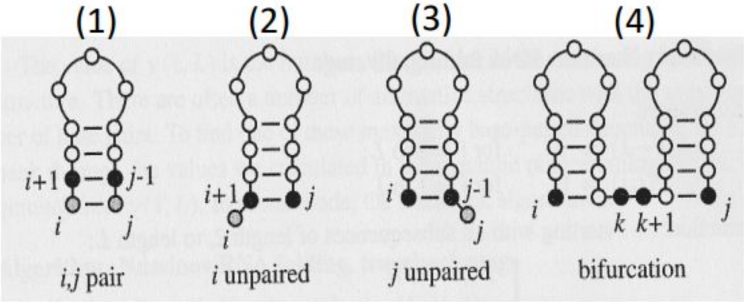
	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	3
G	0	0	0	0	0	0	1	2	2
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
A	0	0	0	0	0	0	1	1	1
U	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0





Sharp loops are not preferred

Hairpin Loops with Minimum Length



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

A typical value of minimum loop length is 4

Time and space complexity

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

- We have a subproblem for every interval (*i*,*j*)
- How many subproblems are there?

$$\binom{n}{2} = O(n^2)$$

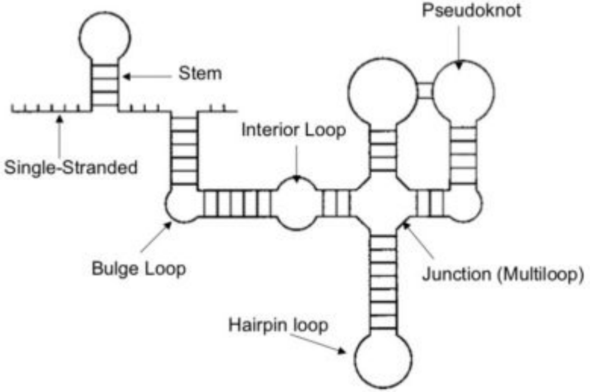
- Each takes O(*n*) time to solve
  - have to search over all possible choices of *k*
- Total running time is $O(n^3)$
- Space complexity $O(n^2)$

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
- RNA folding rules:
  - If two bases are closer than 4 bases apart, they cannot pair (no sharp turns)
  - Each base is matched to at most one other base
  - The allowable pairs are {U, A} and {C, G}
  - Pairs cannot “cross.”
- Nussinov Algorithm: Dynamic programming to find pseudoknot-free structure with maximum number of complementary nucleotide pairs