Internal loops in RNA secondary structure prediction

Lyngsø, Zuker, and Pedersen (1999)

Andrew Hendriks

CMPT 889

Selected Topics in Bioinformatics

Overview

- RNA Biochemistry
- RNA roles
- Structure Prediction Overview
- Nussinov's Algorithm

RNA Defined

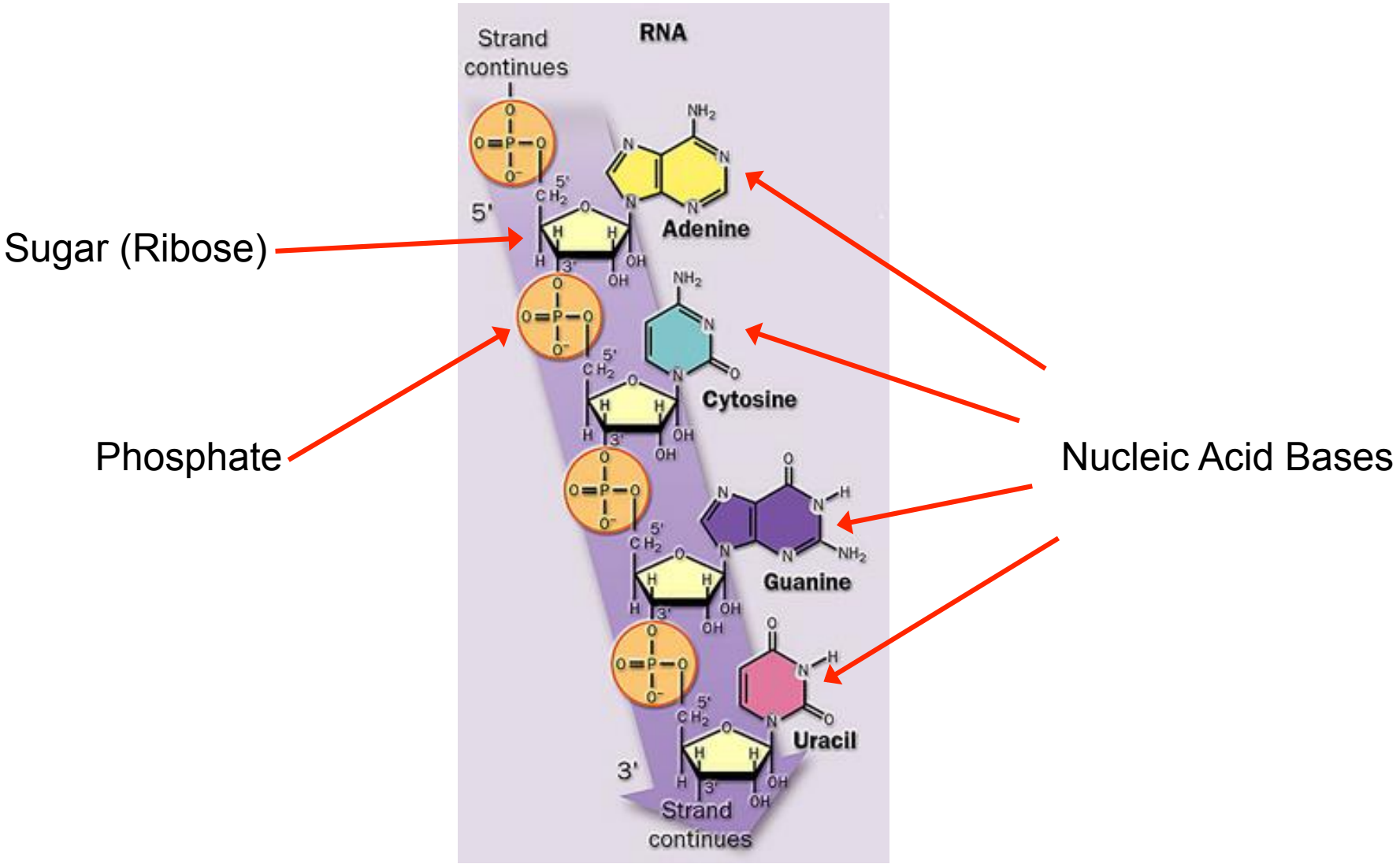
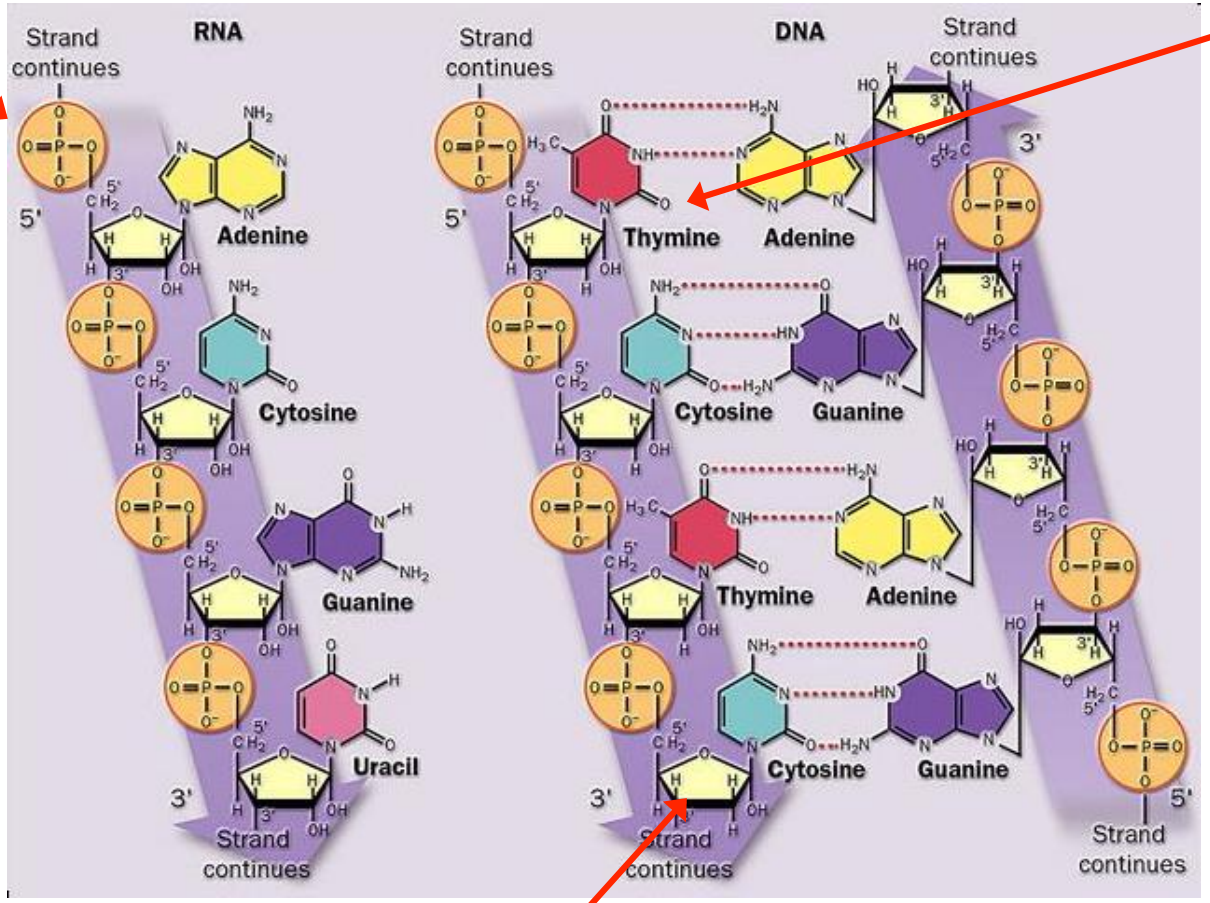


Image Source: Nelson & Cox (2000) “Understand! Biochemistry” Leninger Principles of Biochemistry, Third Edition

How is RNA different from DNA?

Single-stranded

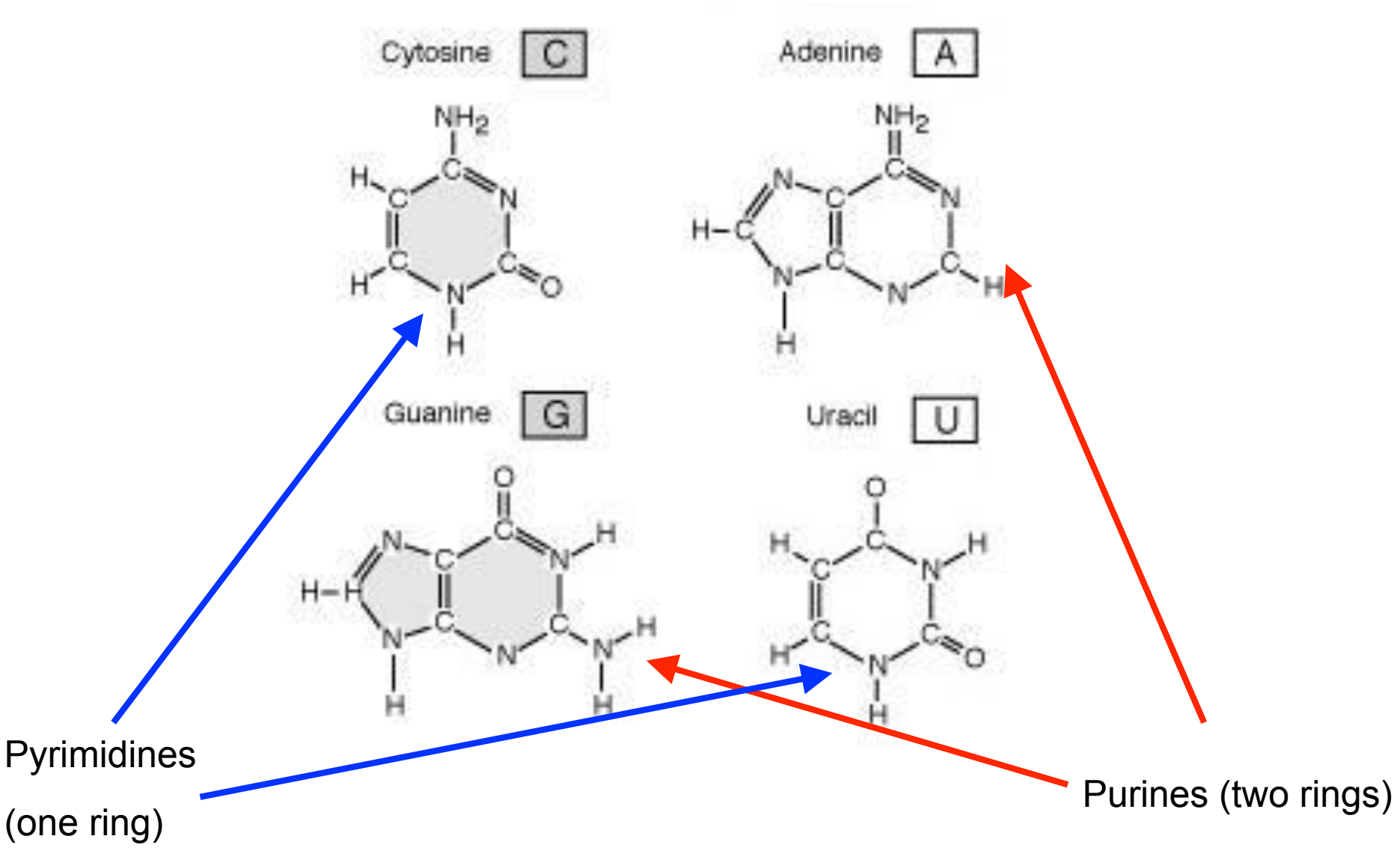
Uracil replaces Thymine



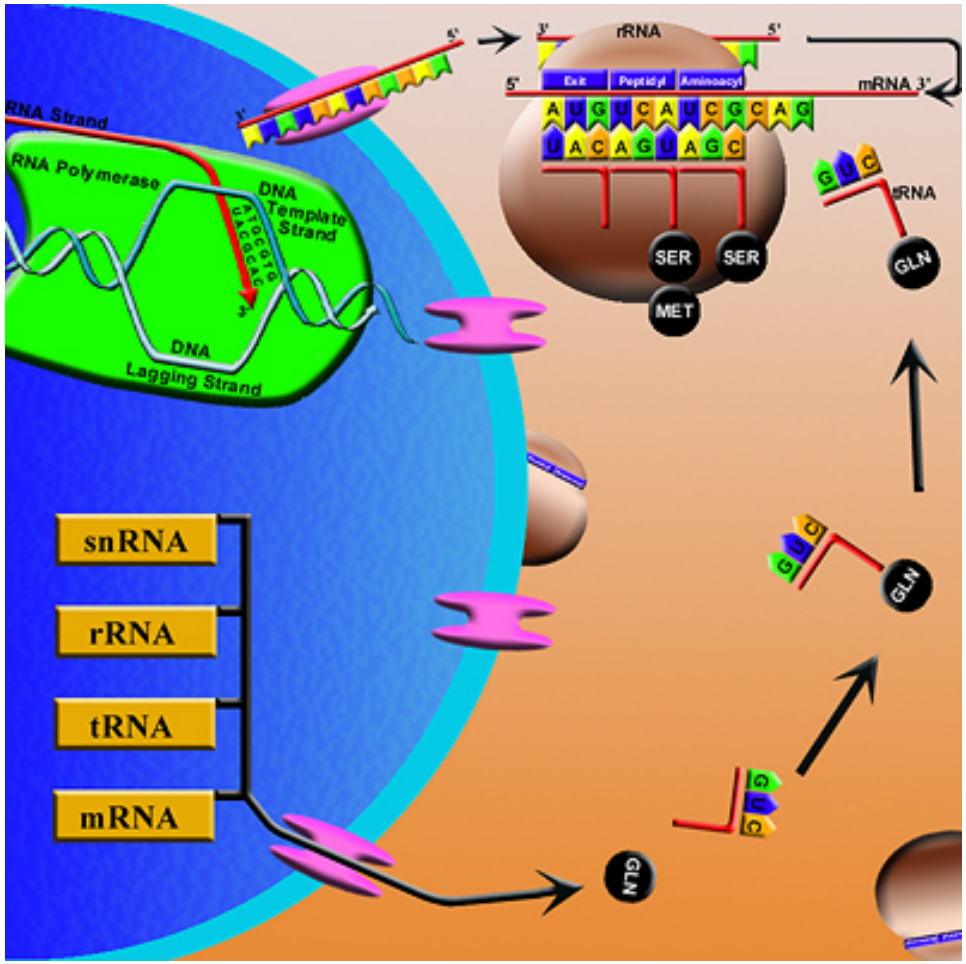
Sugar is Ribose instead of Deoxyribose

Image Source: Nelson & Cox (2000) “Understand! Biochemistry” Leninger Principles of Biochemistry, Third Edition

RNA Bases


Cytosine
C
Adenine
A
Guanine
G
Uracil
U
Pyrimidines
(one ring)
Purines (two rings)

Central Dogma of Molecular Biology



- RNA is central in several stages of protein synthesis

Image source: Regents of New Mexico State Univ./SWBIC (2001), http://www.swbic.org/education/ttexter1.php

Types of RNA

- small nuclear RNA (snRNA)
  - RNA splicing (removal of introns)
- ribosomal RNA (rRNA)
  - combine with proteins to make ribosomes
- transfer RNA (tRNA)
  - combines with amino acids as the first step in protein synthesis
- messenger RNA, (mRNA)
  - transcribed from DNA, encodes proteins

Why ELSE is RNA Important?

- discovery of catalytic RNA by Cech & Bass (1986)
- structural and catalytic RNAs are important in molecular biology of organisms

RNA World Hypothesis

- hypothesis that ancient RNA molecules served as the starting point for life (Gilbert 1986)
- i.e. RNA genomes were replicated by RNA catalysts
- seems to be hotly debated

Why Predict Structure?

- knowing a biomolecule's shape is invaluable in endeavors such as creating new drugs and understanding genetic diseases

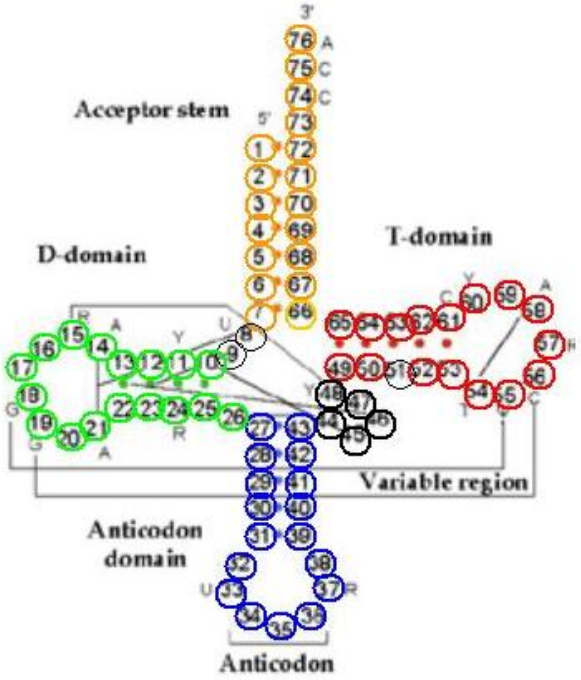
- current physical methods (Nuclear Magnetic Resonance and X-Ray Crystallography) are too expensive and time consuming

- we wish to predict shape of biopolymers from sequence of bases

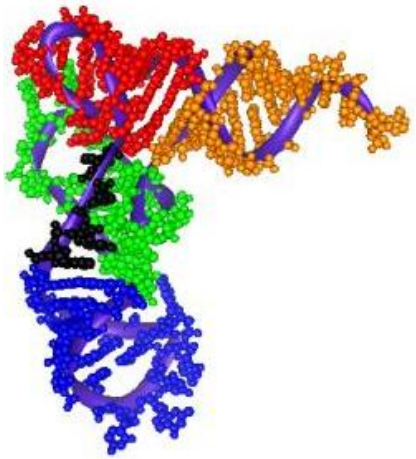
Secondary and Tertiary Structure

GCGGAUUUAGCUCAGUUGG
GAGAGCGCCAGACUGAAGA
UCUGGAGGUCCUGUGUUCG
AUCCACAGAAUUCGCACCA

Primary Structure



Secondary Structure



Tertiary Structure

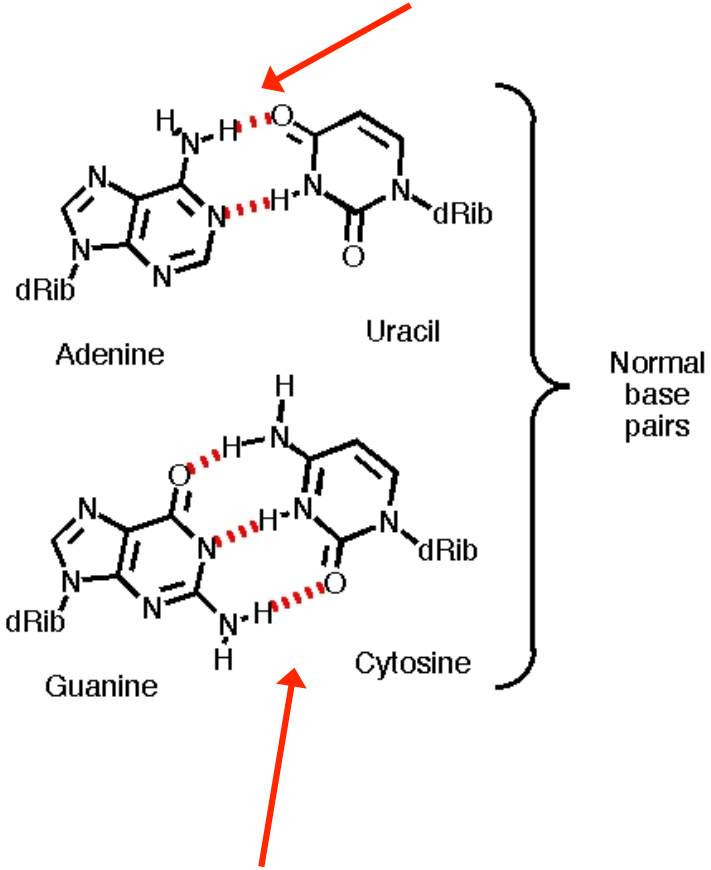
Image Source: Designed Universe http://www.designeduniverse.com/articles/Nobel_Prize/Nobel_DNA2.htm

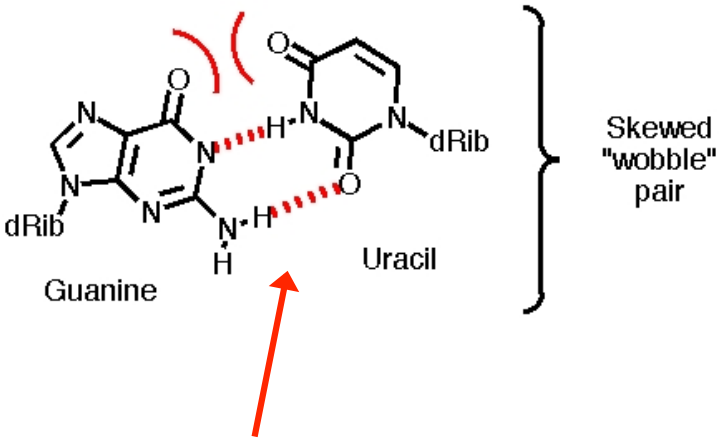
Why RNA Secondary Structure?

- simply put, secondary structure prediction is more straightforward
- four basic structures: helices, loops, bulges and junctions
- energies involved in secondary structures are greater than tertiary, making them more stable (Tinoco & Bustamante, 1999)

Base Pairs in RNA

2 Hydrogen Bonds (less stable)





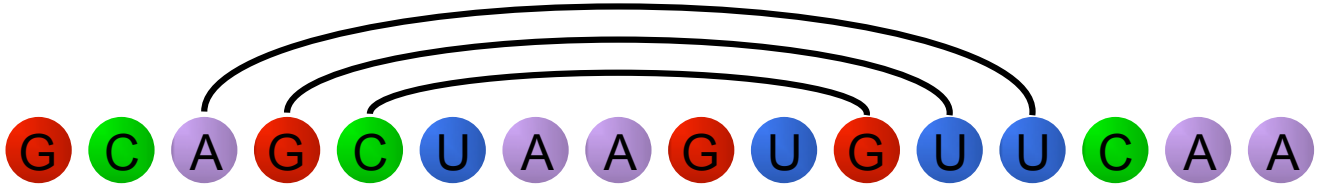
“Non-canonical” base pair

3 Hydrogen Bonds (most stable)

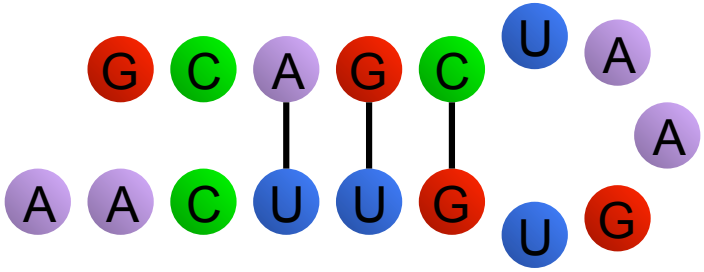
Image Source: “BC 5254/GCS 719, Computer Applications in Biomedical Research” http://www.finchcms.edu/cms/biochem/Walters/rna_folding.html

RNA Folding

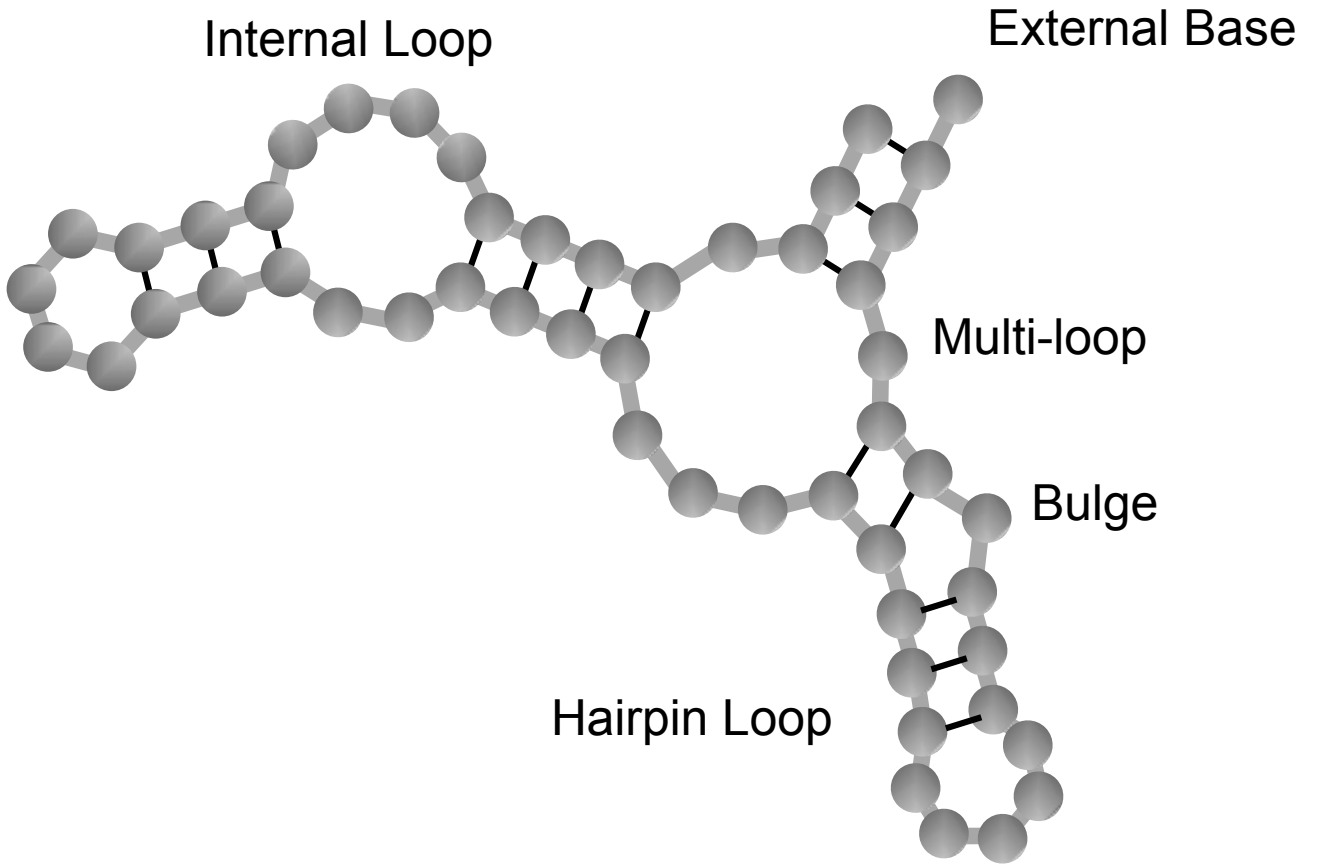
- bonds form between “canonical base pairs” (GC, AU, GU and their mirrors)



- these bonds “fold” the sequence back on itself to form secondary structure (helices)

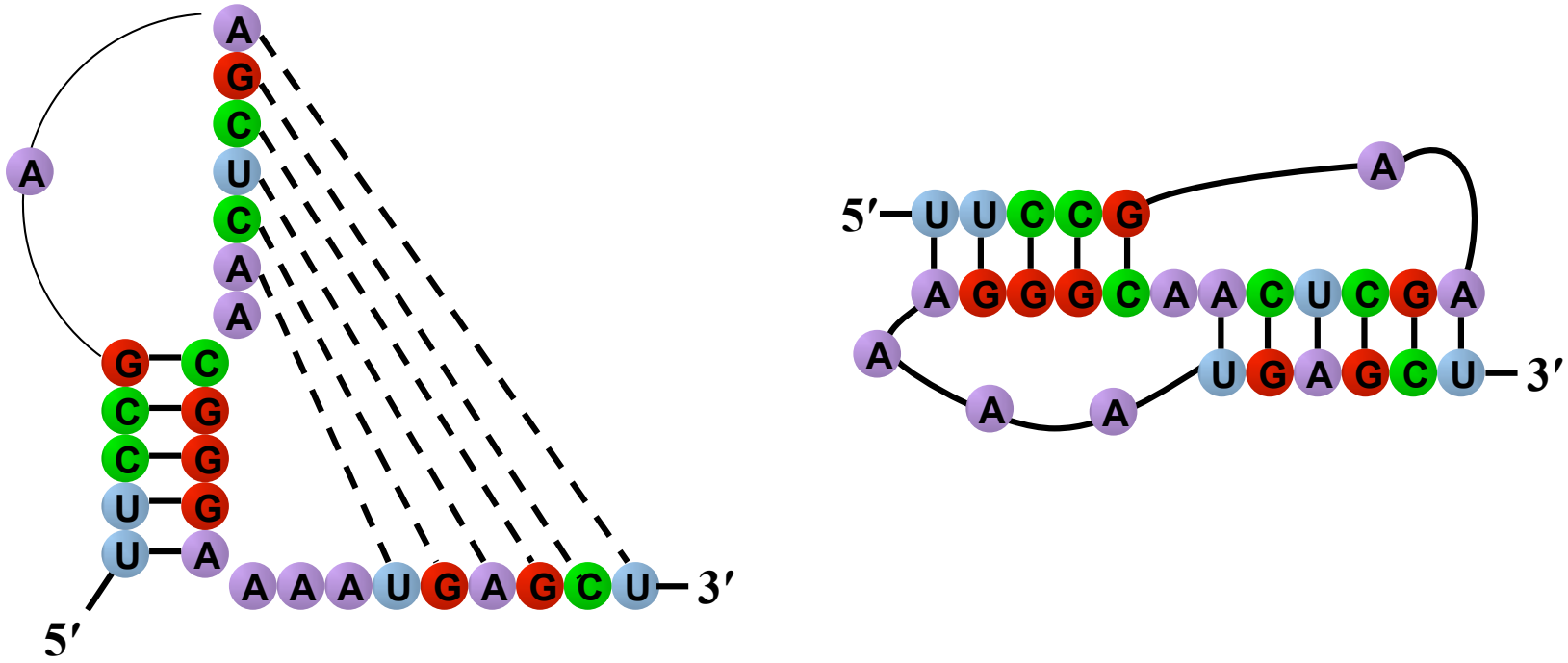


Secondary Structure Elements



Note: the same sequence may produce many different, overlapping helices

Pseudoknots



- bases pairs between a loop and positions outside the enclosing stem
- two stems can stack coaxially and mimic a contiguous A-form helix

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

RNA A-Form Helix

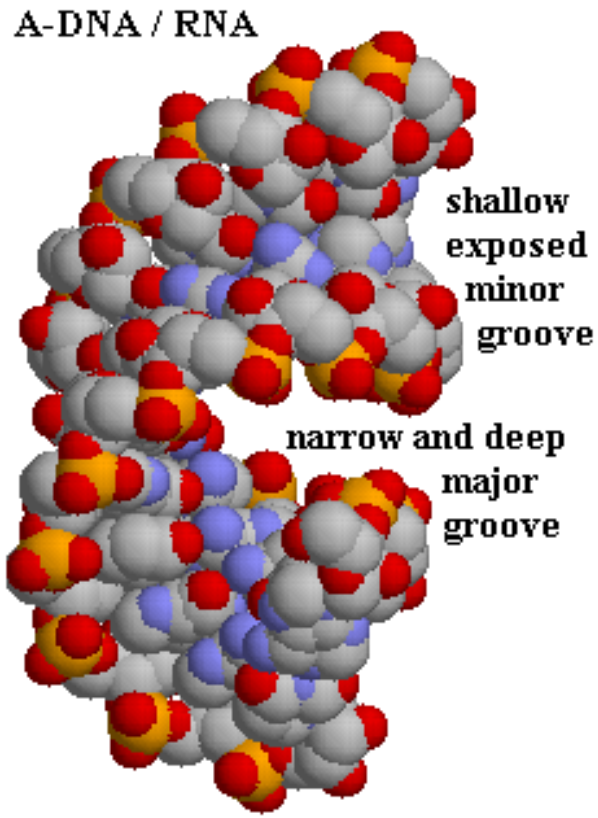


Image source: Oehler, U. (2002) “Chem*730 Proteins and Nucleic acids” http://www.chembio.uoguelph.ca/educmat/chm730/h730.htm

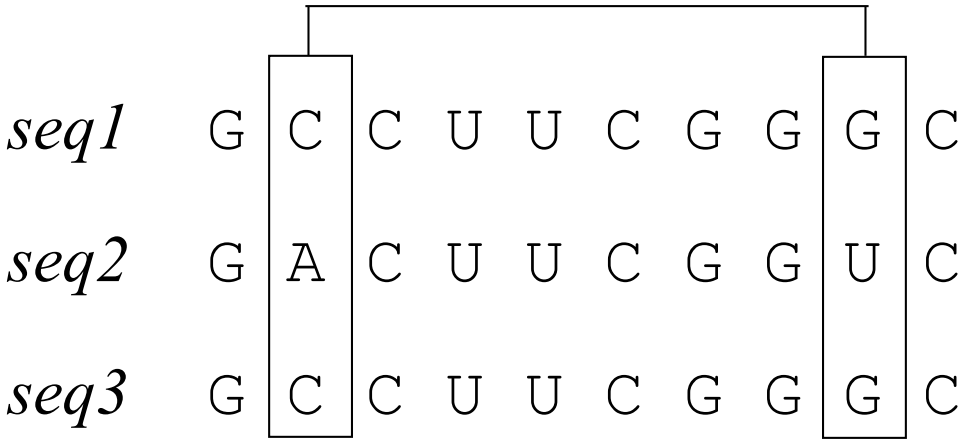
Methods of Secondary Structure Prediction

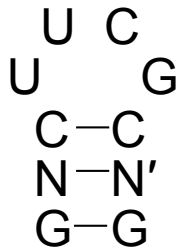
- Comparative Sequence Analysis
- Dynamic Programming

Comparative Sequence Analysis

- during evolution, secondary structure of functional RNA conserved better than primary
- align sets of phylogenetically-ordered homologous sequences
- invariance in certain sections identifies them as being important to structure and function

Comparative Sequence Analysis





- highlighted sections covary, maintaining Watson-Crick complementarity

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Dynamic Programming

- recursive computation
- i.e. maximizes base pairs or minimizes free energy
- focus on algorithms by Nussinov and Zuker

First DP Algorithm: Nussinov

- one possible technique: base pair maximization
- Algorithms for Loop Matching (Nussinov et al., 1978)
- too simple for accurate prediction, but stepping-stone for later algorithms

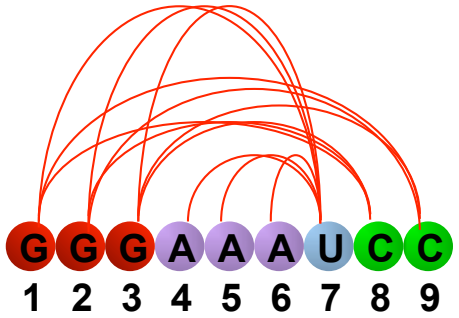
Initial Concepts

- only consider base pairs

- folding of an N nucleotide sequence can be specified by a symmetric $N \times N$ matrix
- $M_{ij}=1$ if bases form a pair
- $M_{ij}=0$ otherwise

Naïve Example 1

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	0	0	0	0	0	0	1	1	1
2	**G**	0	0	0	0	0	0	1	1	1
3	**G**	0	0	0	0	0	0	1	1	1
4	**A**	0	0	0	0	0	0	1	0	0
5	**A**	0	0	0	0	0	0	1	0	0
6	**A**	0	0	0	0	0	0	1	0	0
7	**U**	1	1	1	1	1	1	0	0	0
8	**C**	1	1	1	0	0	0	0	0	0
9	**C**	1	1	1	0	0	0	0	0	0

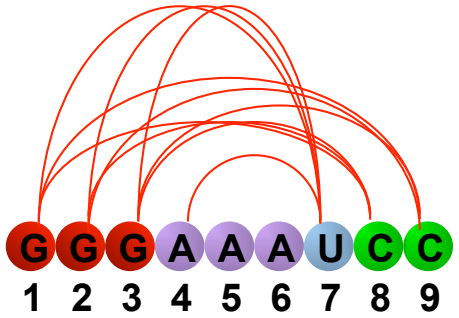


Matching “blocks”

- visually inspect matrices for diagonal lines of 1’s
- manually piece them together into an optimal folded shape

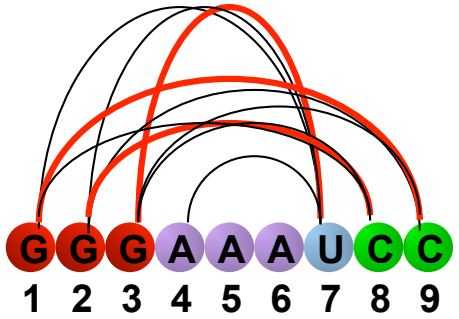
Naïve Example 1

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	0	0	0	0	0	0	**1**	**1**	**1**
2	**G**	0	0	0	0	0	0	**1**	**1**	**1**
3	**G**	0	0	0	0	0	0	**1**	**1**	**1**
4	**A**	0	0	0	0	0	0	**1**	0	0
5	**A**	0	0	0	0	0	0	0	0	0
6	**A**	0	0	0	0	0	0	0	0	0
7	**U**	**1**	**1**	**1**	**1**	0	0	0	0	0
8	**C**	**1**	**1**	**1**	0	0	0	0	0	0
9	**C**	**1**	**1**	**1**	0	0	0	0	0	0



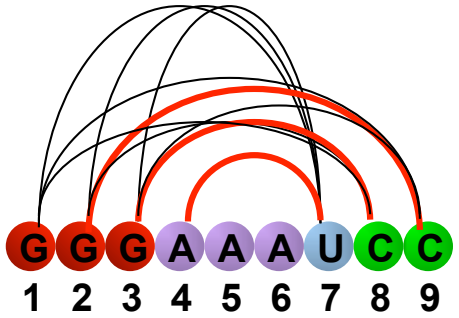
Naïve Example 1

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	0	0	0	0	0	0	**1**	**1**	**1**
2	**G**	0	0	0	0	0	0	**1**	**1**	**1**
3	**G**	0	0	0	0	0	0	**1**	**1**	**1**
4	**A**	0	0	0	0	0	0	**1**	0	0
5	**A**	0	0	0	0	0	0	**1**	0	0
6	**A**	0	0	0	0	0	0	**1**	0	0
7	**U**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0
8	**C**	**1**	**1**	**1**	0	0	0	0	0	0
9	**C**	**1**	**1**	**1**	0	0	0	0	0	0



Naïve Example 1

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	0	0	0	0	0	0	**1**	**1**	**1**
2	**G**	0	0	0	0	0	0	**1**	**1**	**1**
3	**G**	0	0	0	0	0	0	**1**	**1**	**1**
4	**A**	0	0	0	0	0	0	**1**	0	0
5	**A**	0	0	0	0	0	0	**1**	0	0
6	**A**	0	0	0	0	0	0	**1**	0	0
7	**U**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0
8	**C**	**1**	**1**	**1**	0	0	0	0	0	0
9	**C**	**1**	**1**	**1**	0	0	0	0	0	0



Refinement

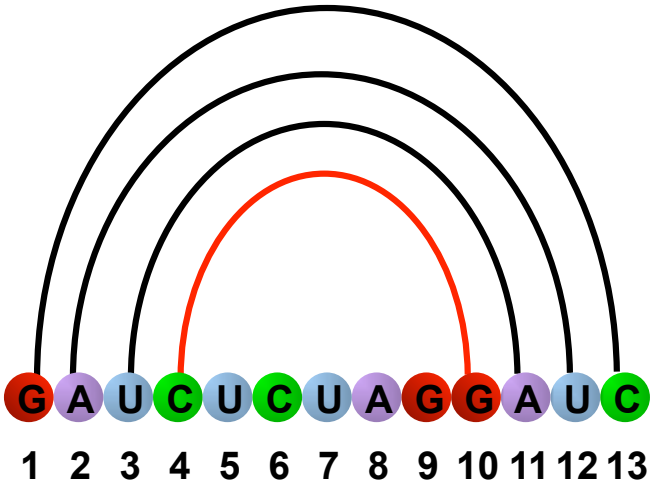
- unfortunately, this finds chemically infeasible structures
- i.e. insufficient space, inflexibility of paired base regions
- next step is to specify better constraints
- solution: a dynamic programming algorithm [Nussinov et al., 1978]

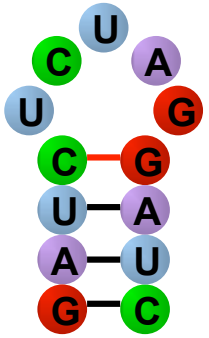
Structure Representation

- secondary structure described as a graph
- base pairs are described via pairs of indices

(i, j), indicating links between base vertices

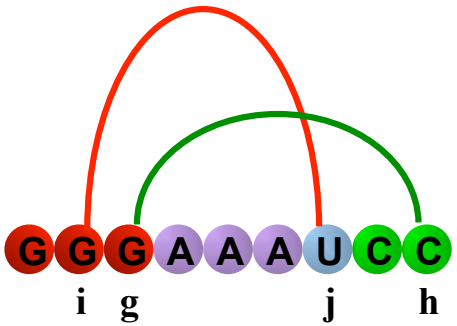
S={(1,13), (2,12), (3,11), (4,10)}





Basic Constraints

1. Each edge contains vertices (bases) linking compatible base pairs
2. No vertex can be in more than one edge
3. Edges must be drawn without crossing

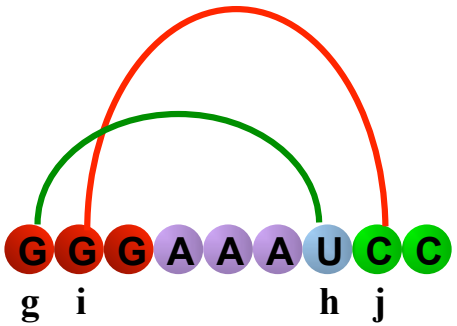


Edges ***(g, h)*** and ***(i, j)***

if $i < g < j < h$ *or* $g < i < h < j$, *both edges cannot belong to the same "matching."*

Basic Constraints

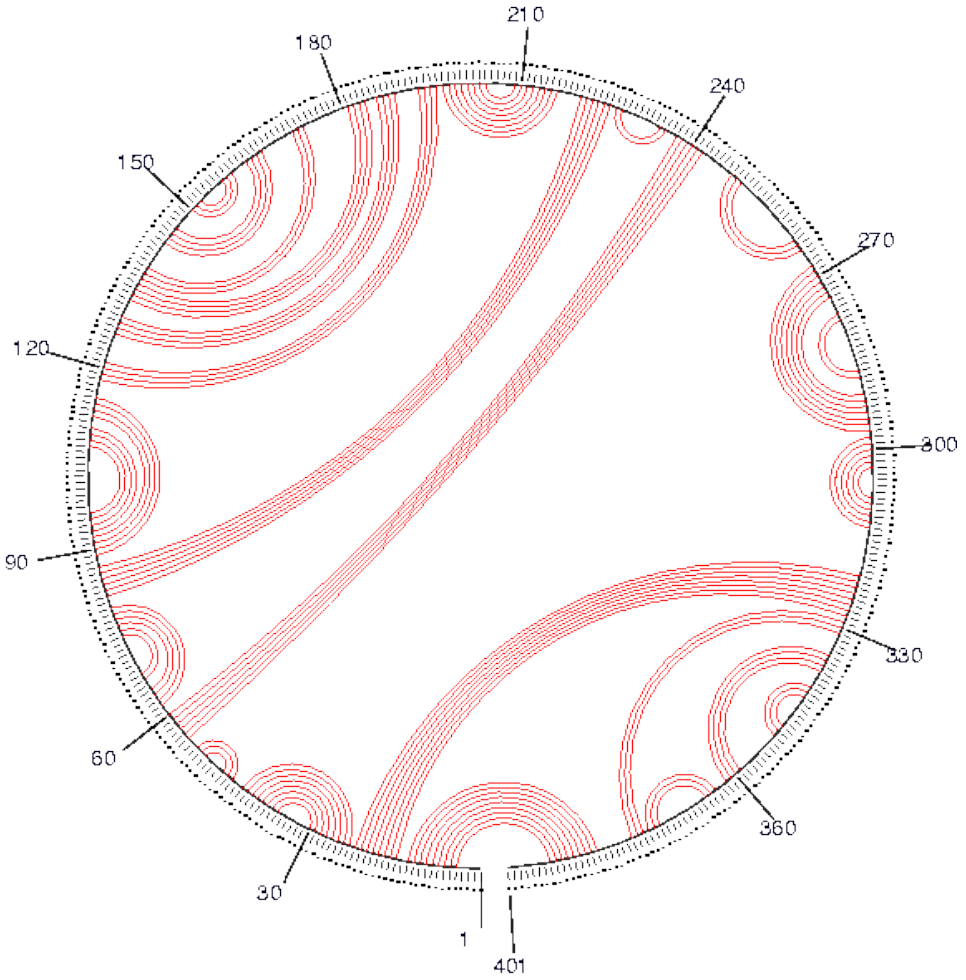
1. Each edge contains vertices (bases) linking compatible base pairs
2. No vertex can be in more than one edge
3. Edges must be drawn without crossing



Edges ***(g, h)*** and ***(i, j)***

if $i < g < j < h$ *or* $\boldsymbol{g < i < h < j}$*, both edges cannot belong to the same "matching."*

Circular Representation



ENERGY = -85.7 Bacillus subtilis RNase P RNA

Image source: Zuker, M. (2002) “Lectures on RNA Secondary Structure Prediction” http://www.bioinfo.rpi.edu/~zukerm/lectures/RNAfold-html/node1.html

Energy Minimization

- objective is a folded shape for a given nucleotide chain such that the energy is minimized

- $E_{ij} = 1$ for each possible compatible base pair, $E_{ij} = 0$ otherwise

Algorithm Behaviour

- recursive computation, finding the best structure for small subsequences
- works outward to larger subsequences
- four possible ways to get the best RNA structure:

Case 1: Adding unpaired base i

- Add unpaired position i onto best structure for subsequence $i+1, j$

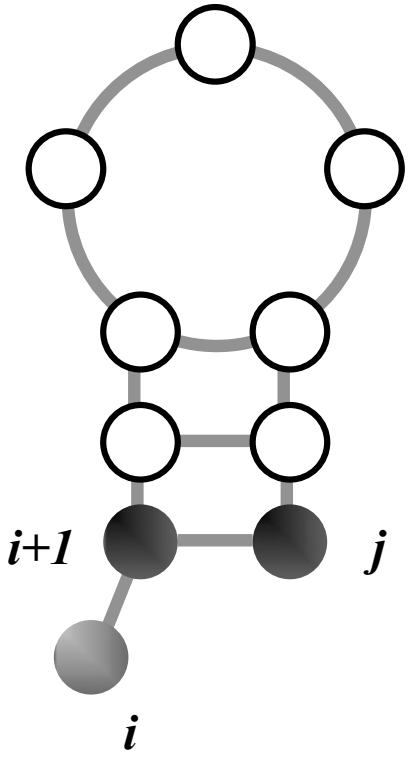


Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Case 2: Adding unpaired base j

- Add unpaired position i onto best structure for subsequence $i+1, j$

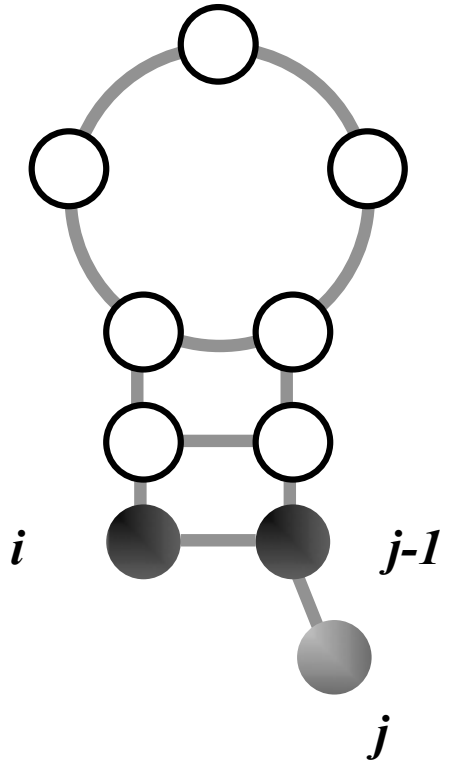


Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Case 3: Adding (i, j) pair

- Add base pair (i, j) onto best structure found for subsequence i+1, j-1

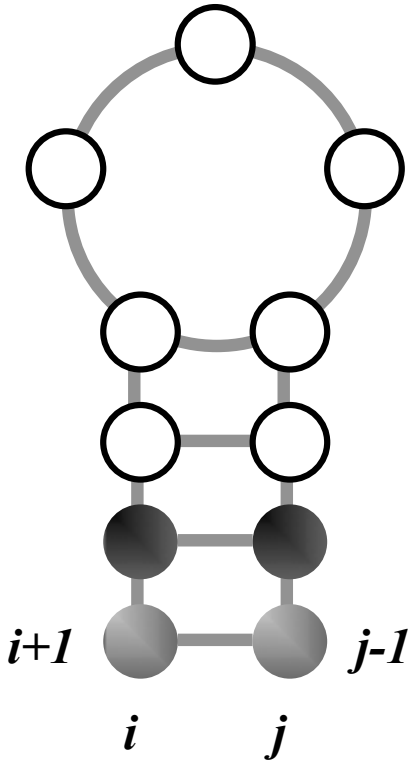


Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Case 4: Bifurcation

- combining two optimal substructures i, k and $k+1, j$

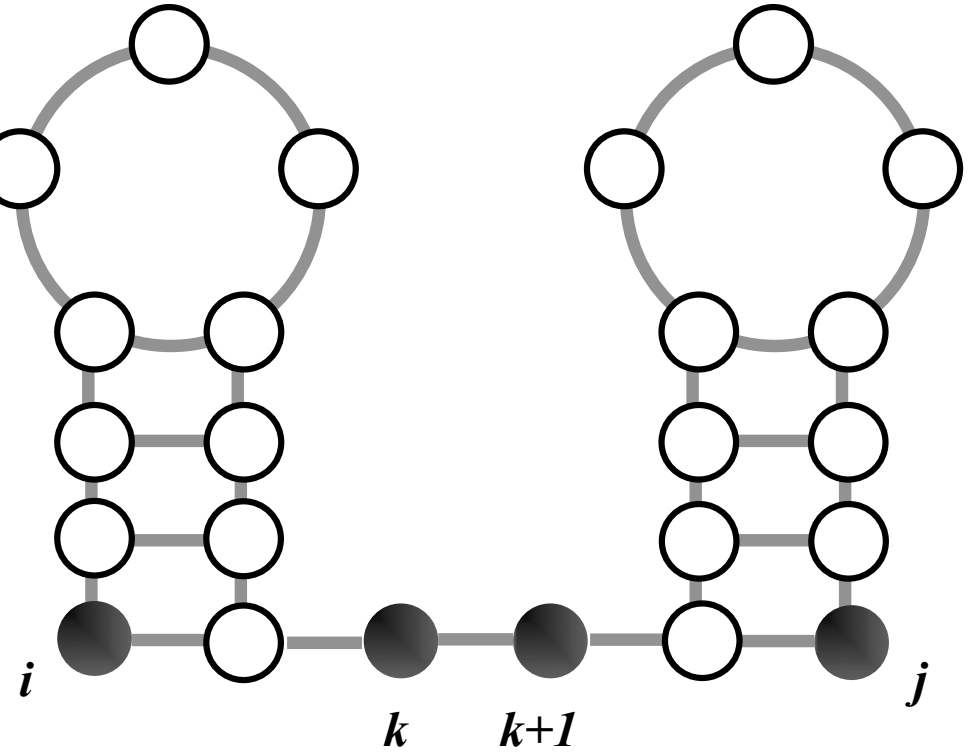


Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Nussinov RNA Folding Algorithm

- Initialization:

$\gamma(i, i\text{-}1) = 0$ *for* $I = 2$ *to* L;

$\gamma(i, i) = 0$ *for* $I = 2$ *to* L.

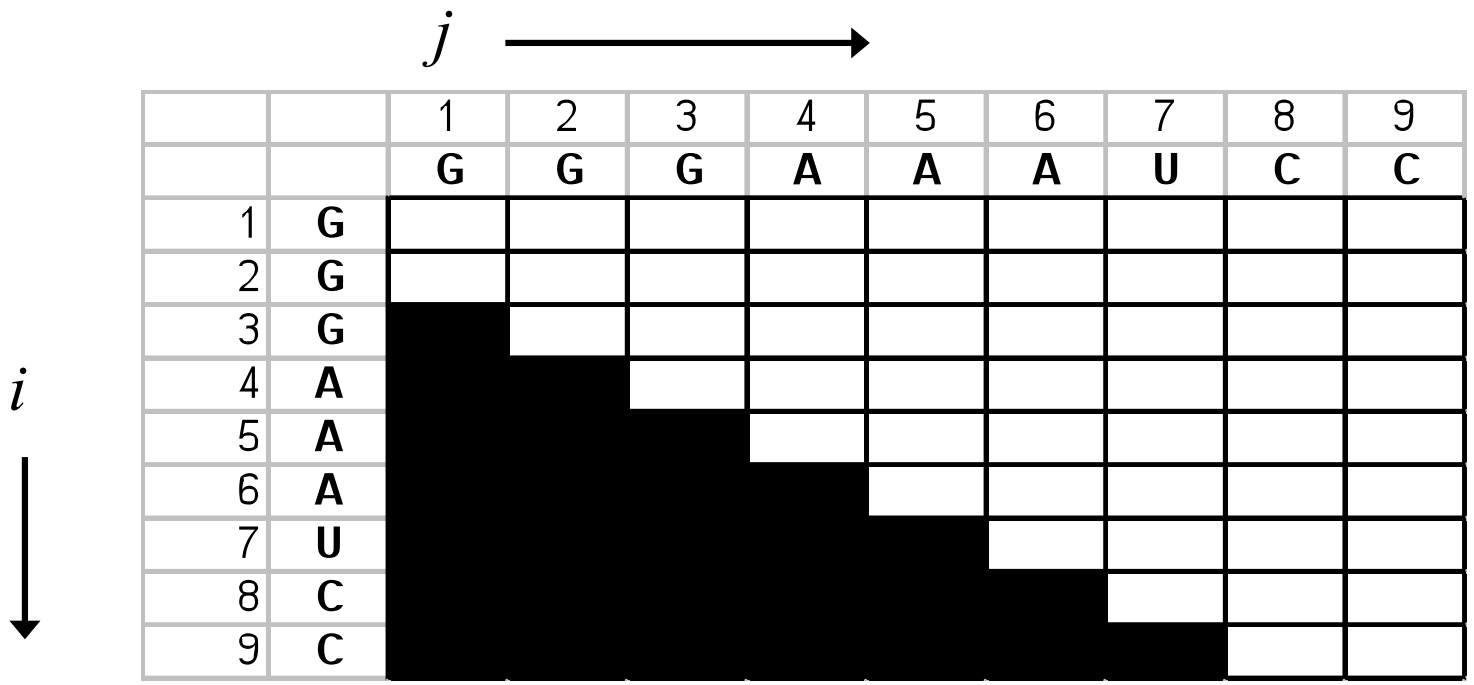


Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Nussinov RNA Folding Algorithm

- Initialization:

$\gamma(i, i\text{-}1) = 0$ *for* $I = 2$ *to* L;

$\gamma(i, i) = 0$ *for* $I = 2$ *to* L.

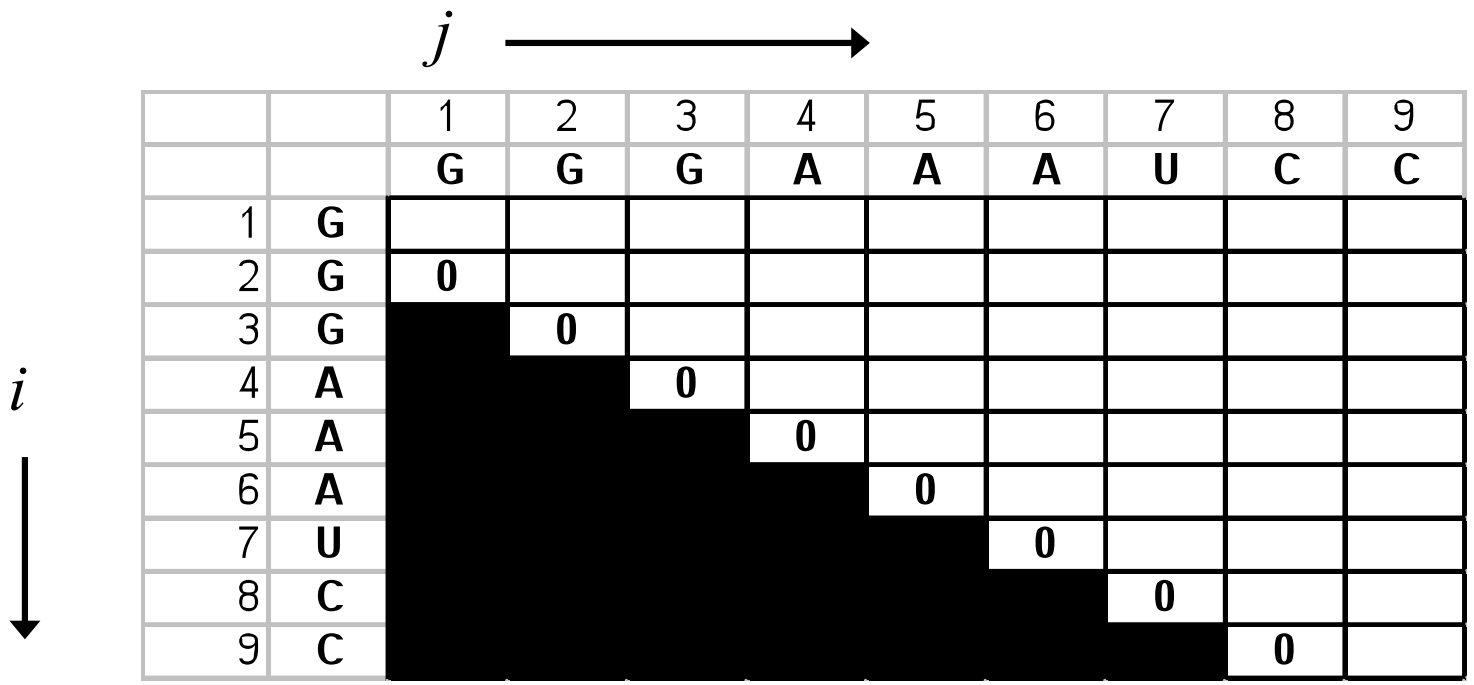


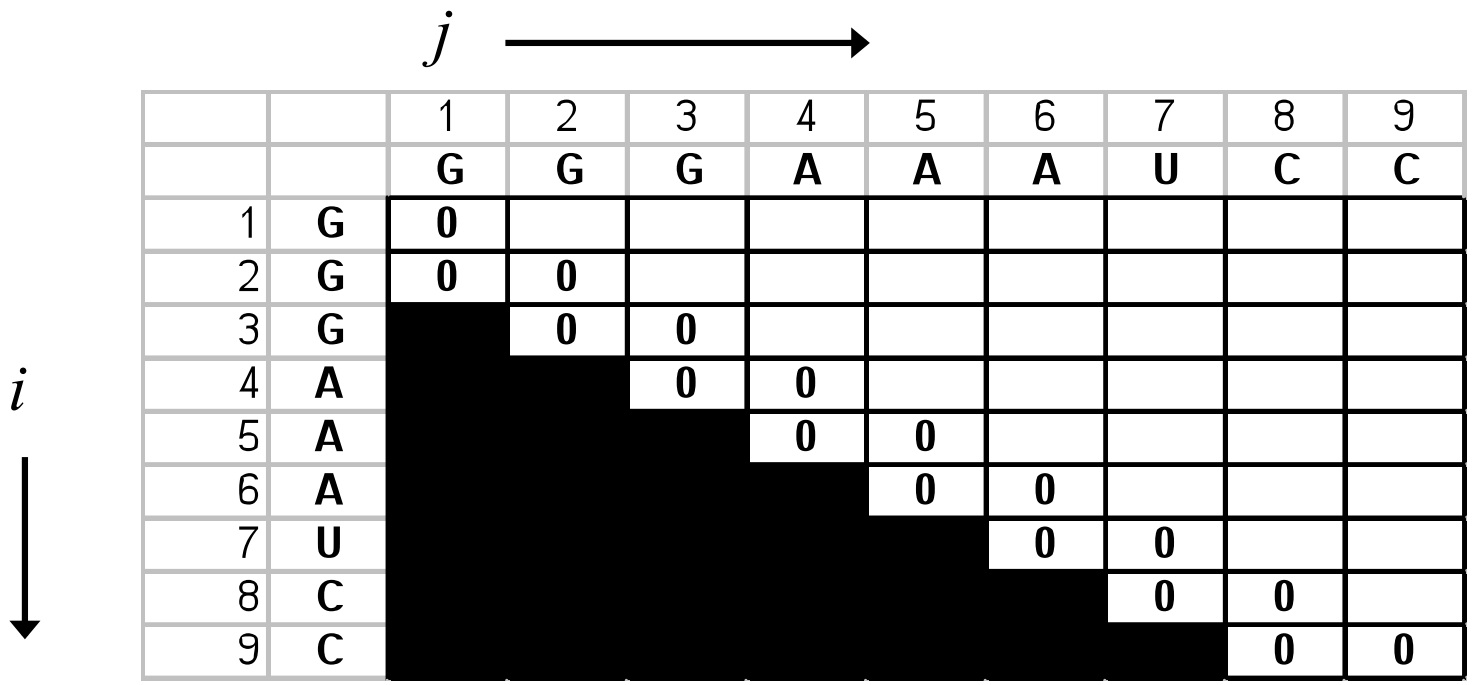
Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Nussinov RNA Folding Algorithm

- Initialization:

$\gamma(i, i\text{-}1) = 0$ *for* $I = 2$ *to* L;

$\gamma(i, i) = 0$ *for* $I = 2$ *to* L.



		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**								
2	**G**	**0**	**0**							
3	**G**		**0**	**0**						
4	**A**			**0**	**0**					
5	**A**				**0**	**0**				
6	**A**					**0**	**0**			
7	**U**						**0**	**0**		
8	**C**							**0**	**0**	
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Nussinov RNA Folding Algorithm

- Recursive Relation:

- For all subsequences from length 2 to length L:

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j) & \text{Case 1} \\ \gamma(i,j-1) & \text{Case 2} \\ \gamma(i+1,j-1)+\delta(i,j) & \text{Case 3} \\ \max_{i<k<j}[\gamma(i,k)+\gamma(k+1,j)] & \text{Case 4} \end{cases}$$

Nussinov RNA Folding Algorithm

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j) \\ \gamma(i,j-1) \\ \gamma(i+1,j-1) + \delta(i,j) \\ \max_{i<k<j}[\gamma(i,k) + \gamma(k+1,j)] \end{cases}$$

$j \longrightarrow$

$i \downarrow$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**							
2	**G**	**0**	**0**	**0**						
3	**G**		**0**	**0**	**0**					
4	**A**			**0**	**0**	**0**				
5	**A**				**0**	**0**	**0**			
6	**A**					**0**	**0**	**1**		
7	**U**						**0**	**0**	**0**	
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Nussinov RNA Folding Algorithm

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j) \\ \gamma(i,j-1) \\ \gamma(i+1,j-1) + \delta(i,j) \\ \max_{i<k<j}[\gamma(i,k) + \gamma(k+1,j)] \end{cases}$$

$j \longrightarrow$

$i \downarrow$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**						
2	**G**	**0**	**0**	**0**	**0**					
3	**G**		**0**	**0**	**0**	**0**				
4	**A**			**0**	**0**	**0**	**0**			
5	**A**				**0**	**0**	**0**	**1**		
6	**A**					**0**	**0**	**1**	**1**	
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Nussinov RNA Folding Algorithm

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j) \\ \gamma(i,j-1) \\ \gamma(i+1,j-1)+\delta(i,j) \\ \max_{i<k<j}[\gamma(i,k)+\gamma(k+1,j)] \end{cases}$$

j →

i ↓

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**	**1**		
5	**A**				**0**	**0**	**0**	**1**	**1**	
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Example Computation

$$\gamma(4,7) = \max \begin{cases} \gamma(5,7) \\ \gamma(4,6) \\ \gamma(5,6) + \delta(4,7) \\ \max_{4<k<7} [\gamma(4,k) + \gamma(k+1,7)] \end{cases}$$

j →

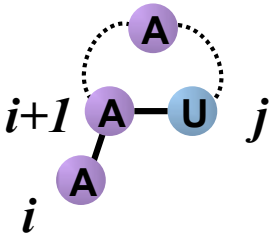
i ↓

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**			
5	**A**				**0**	**0**	**0**	**1**	**1**	
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Example Computation

$$\gamma(4,7) = \max \begin{cases} \gamma(5,7) \\ \gamma(4,6) \\ \gamma(5,6) + \delta(4,7) \\ \max_{4<k<7}[\gamma(4,k) + \gamma(k+1,7)] \end{cases}$$



j →

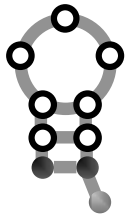
i ↓

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**			
5	**A**				**0**	**0**	**0**	**1**	**1**	
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Example Computation

$$\gamma(4,7) = \max \begin{cases} \gamma(5,7) \\ \gamma(4,6) \\ \gamma(5,6) + \delta(4,7) \\ \max_{4<k<7} [\gamma(4,k) + \gamma(k+1,7)] \end{cases}$$



j →

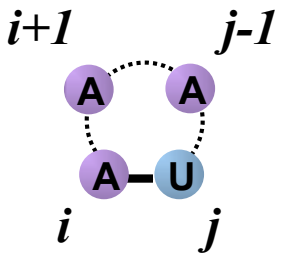
i ↓

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**			
5	**A**				**0**	**0**	**0**	**1**	**1**	
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Example Computation

$$\gamma(4,7) = \max \begin{cases} \gamma(5,7) \\ \gamma(4,6) \\ \gamma(5,6) + \delta(4,7) \\ \max_{4<k<7}[\gamma(4,k) + \gamma(k+1,7)] \end{cases}$$



j →

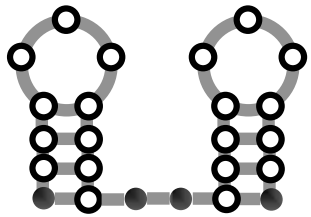
i ↓

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**			
5	**A**				**0**	**0**	**0**	**1**	**1**	
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Example Computation

$$\gamma(4,7) = \max \begin{cases} \gamma(5,7) \\ \gamma(4,6) \\ \gamma(5,6) + \delta(4,7) \\ \max_{4<k<7}[\gamma(4,k) + \gamma(k+1,7)] \end{cases}$$



j →

i ↓

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**			
5	**A**				**0**	**0**	**0**	**1**	**1**	
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Example Computation

$$\gamma(4,7) = \max \begin{cases} \gamma(5,7) \\ \gamma(4,6) \\ \gamma(5,6) + \delta(4,7) \\ \max_{4<k<7} [\gamma(4,k) + \gamma(k+1,7)] \end{cases}$$

$j \longrightarrow$

$i \downarrow$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**	**1**		
5	**A**				**0**	**0**	**0**	**1**	**1**	
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Completed Matrix

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j) \\ \gamma(i,j-1) \\ \gamma(i+1,j-1)+\delta(i,j) \\ \max_{i<k<j}[\gamma(i,k)+\gamma(k+1,j)] \end{cases}$$

j →

i ↓

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**2**	**3**
2	**G**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**2**	**3**
3	**G**		**0**	**0**	**0**	**0**	**0**	**1**	**2**	**2**
4	**A**			**0**	**0**	**0**	**0**	**1**	**1**	**1**
5	**A**				**0**	**0**	**0**	**1**	**1**	**1**
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Traceback

- value at $\gamma(1, L)$ is the total base pair count in the maximally base-paired structure
- as in other DP, traceback from $\gamma(1, L)$ is necessary to recover the final secondary structure
- pushdown stack is used to deal with bifurcated structures

Traceback Pseudocode

Initialization: Push $(1,L)$ onto stack

Recursion: Repeat until stack is empty:

- pop (i, j).
- If $i >= j$ continue; // hit diagonal

  else if $\gamma(i+1,j) = \gamma(i, j)$ push $(i+1,j)$; // case 1

  else if $\gamma(i, j-1) = \gamma(i, j)$ push $(i,j-1)$; // case 2

  else if $\gamma(i+1,j-1)+\delta_{i,j} = \gamma(i, j)$: // case 3

  record i, j base pair

  push $(i+1,j-1)$;

  else for $k=i+1$ to $j-1$: if $\gamma(i, k)+\gamma(k+1,j)=\gamma(i, j)$: // case 4

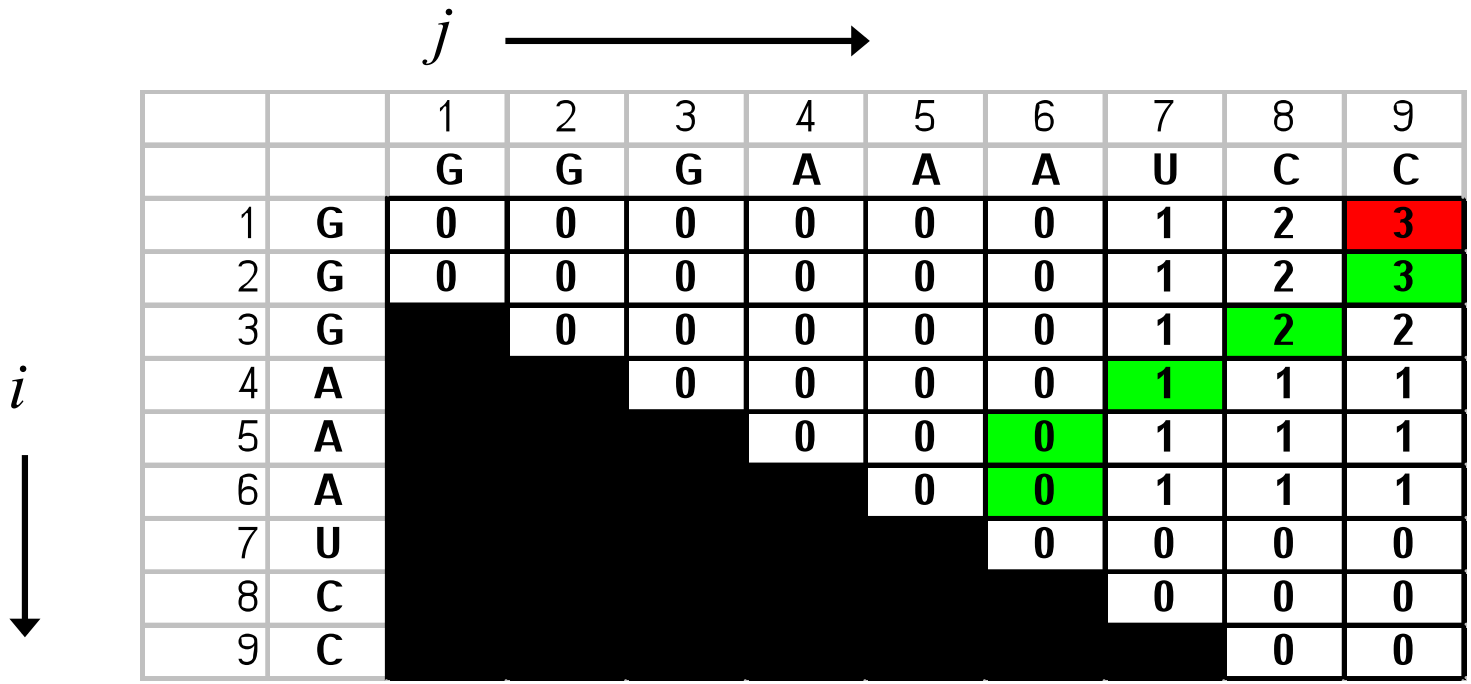
  push $(k+1, j)$.

  push (i, k).

  break

Retrieving the Structure

PAIRS	STACK	CURRENT
	(1,9)	

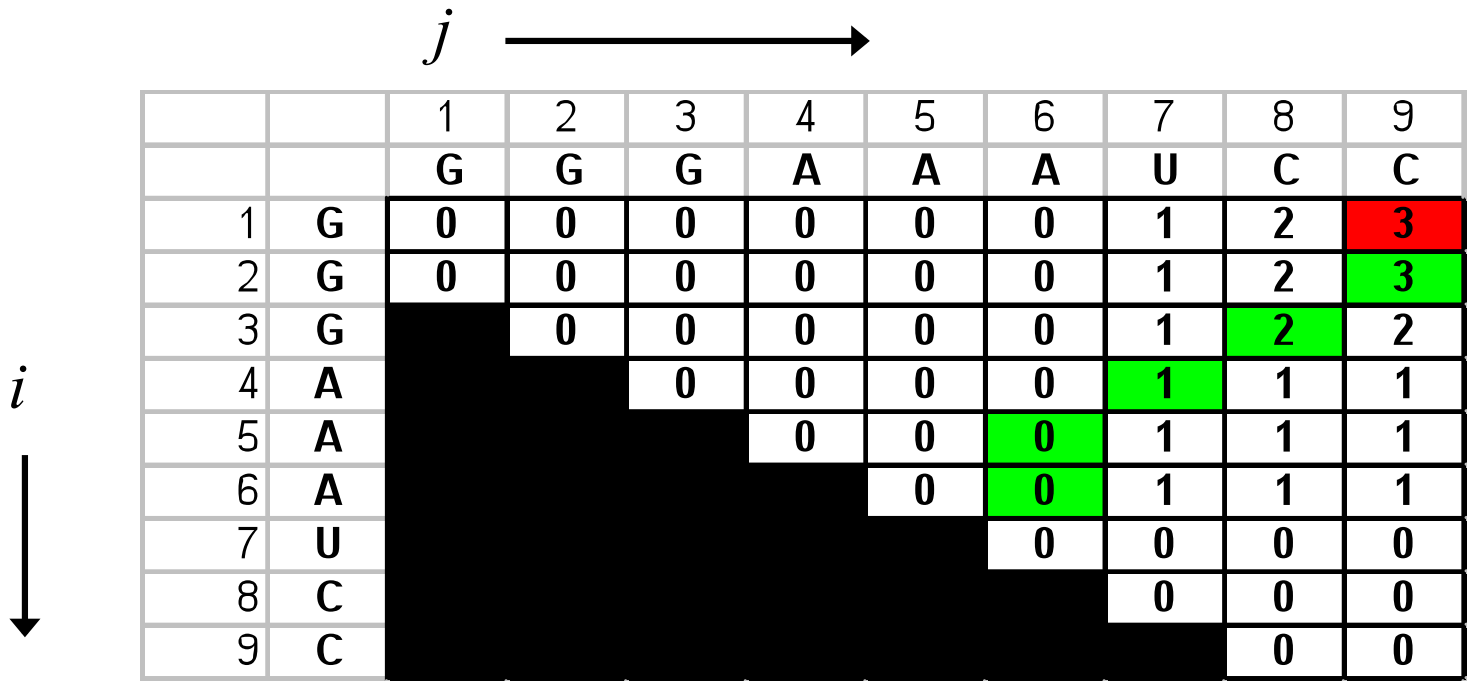


		1	2	3	4	5	6	7	8	9
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
2	G	0	0	0	0	0	0	1	2	3
3	G		0	0	0	0	0	1	2	2
4	A			0	0	0	0	1	1	1
5	A				0	0	0	1	1	1
6	A					0	0	1	1	1
7	U						0	0	0	0
8	C							0	0	0
9	C								0	0

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Retrieving the Structure

PAIRS	STACK	CURRENT
	(2,9)	(1,9)

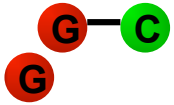


		1	2	3	4	5	6	7	8	9
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
2	G	0	0	0	0	0	0	1	2	3
3	G		0	0	0	0	0	1	2	2
4	A			0	0	0	0	1	1	1
5	A				0	0	0	1	1	1
6	A					0	0	1	1	1
7	U						0	0	0	0
8	C							0	0	0
9	C								0	0

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Retrieving the Structure

PAIRS	STACK	CURRENT
(2,9)	(3,8)	(2,9)



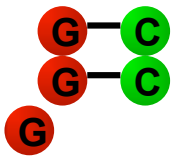
j →

		1	2	3	4	5	6	7	8	9
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
2	G	0	0	0	0	0	0	1	2	3
3	G		0	0	0	0	0	1	2	2
4	A			0	0	0	0	1	1	1
5	A				0	0	0	1	1	1
6	A					0	0	1	1	1
7	U						0	0	0	0
8	C							0	0	0
9	C								0	0

i ↓

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Retrieving the Structure



PAIRS	STACK	CURRENT
(2,9)	(4,7)	(3,8)
(3,8)		

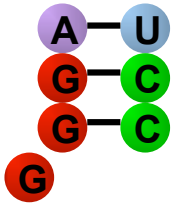
j →

i ↓

		1	2	3	4	5	6	7	8	9
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
2	G	0	0	0	0	0	0	1	2	3
3	G		0	0	0	0	0	1	2	2
4	A			0	0	0	0	1	1	1
5	A				0	0	0	1	1	1
6	A					0	0	1	1	1
7	U						0	0	0	0
8	C							0	0	0
9	C								0	0

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Retrieving the Structure



PAIRS	STACK	CURRENT
(2,9)	(5,6)	(4,7)
(3,8)		
(4,7)		

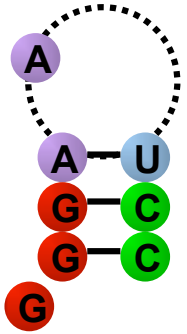
j →

i ↓

		1	2	3	4	5	6	7	8	9
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
2	G	0	0	0	0	0	0	1	2	3
3	G		0	0	0	0	0	1	2	2
4	A			0	0	0	0	1	1	1
5	A				0	0	0	1	1	1
6	A					0	0	1	1	1
7	U						0	0	0	0
8	C							0	0	0
9	C								0	0

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Retrieving the Structure



PAIRS	STACK	CURRENT
(2,9)	(6,6)	(5,6)
(3,8)		
(4,7)		

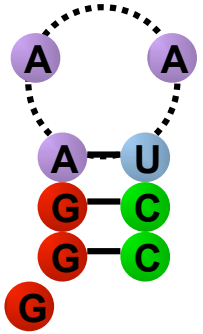
j →

i ↓

		1	2	3	4	5	6	7	8	9
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
2	G	0	0	0	0	0	0	1	2	3
3	G		0	0	0	0	0	1	2	2
4	A			0	0	0	0	1	1	1
5	A				0	0	0	1	1	1
6	A					0	0	1	1	1
7	U						0	0	0	0
8	C							0	0	0
9	C								0	0

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Retrieving the Structure



PAIRS	STACK	CURRENT
(2,9)	-	(6,6)
(3,8)		
(4,7)		

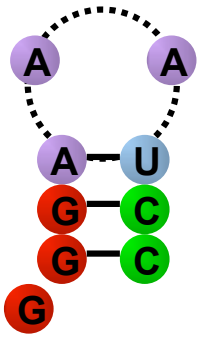
j →

i ↓

		1	2	3	4	5	6	7	8	9
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
2	G	0	0	0	0	0	0	1	2	3
3	G		0	0	0	0	0	1	2	2
4	A			0	0	0	0	1	1	1
5	A				0	0	0	1	1	1
6	A					0	0	1	1	1
7	U						0	0	0	0
8	C							0	0	0
9	C								0	0

Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

Retrieving the Structure



j →

		1	2	3	4	5	6	7	8	9
		G	G	G	A	A	A	U	C	C
1	G	0	0	0	0	0	0	1	2	3
2	G	0	0	0	0	0	0	1	2	3
3	G		0	0	0	0	0	1	2	2
4	A			0	0	0	0	1	1	1
5	A				0	0	0	1	1	1
6	A					0	0	1	1	1
7	U						0	0	0	0
8	C							0	0	0
9	C								0	0

i ↓

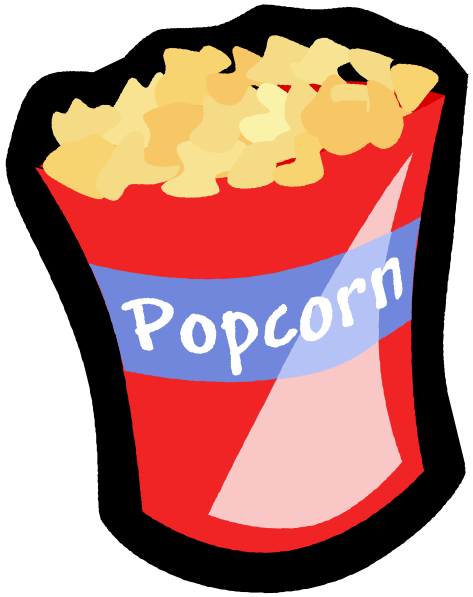
Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

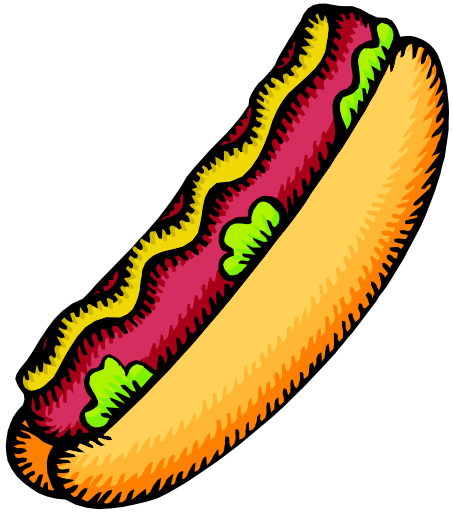
Evaluation of Nussinov

- unfortunately, while this does maximize the base pairs, it does not create viable secondary structures

- in Zuker's algorithm, the correct structure is assumed to have the lowest equilibrium free energy (ΔG) (Zuker and Stiegler, 1981; Zuker 1989a)

Break Time!





Free Energy (ΔG)

- ΔG approximated as the sum of contributions from loops, base pairs and other secondary structures

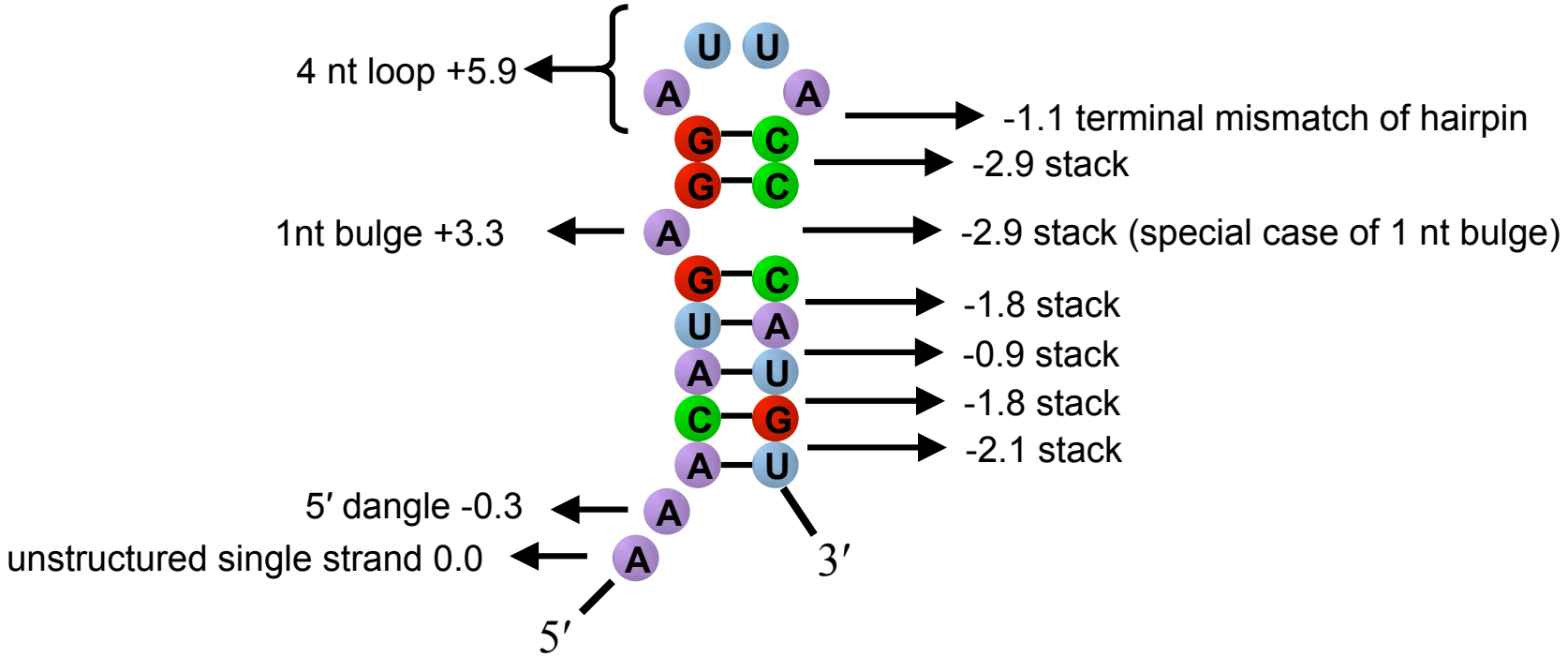


Image Source: Durbin et al. (2002) “Biological Sequence Analysis”

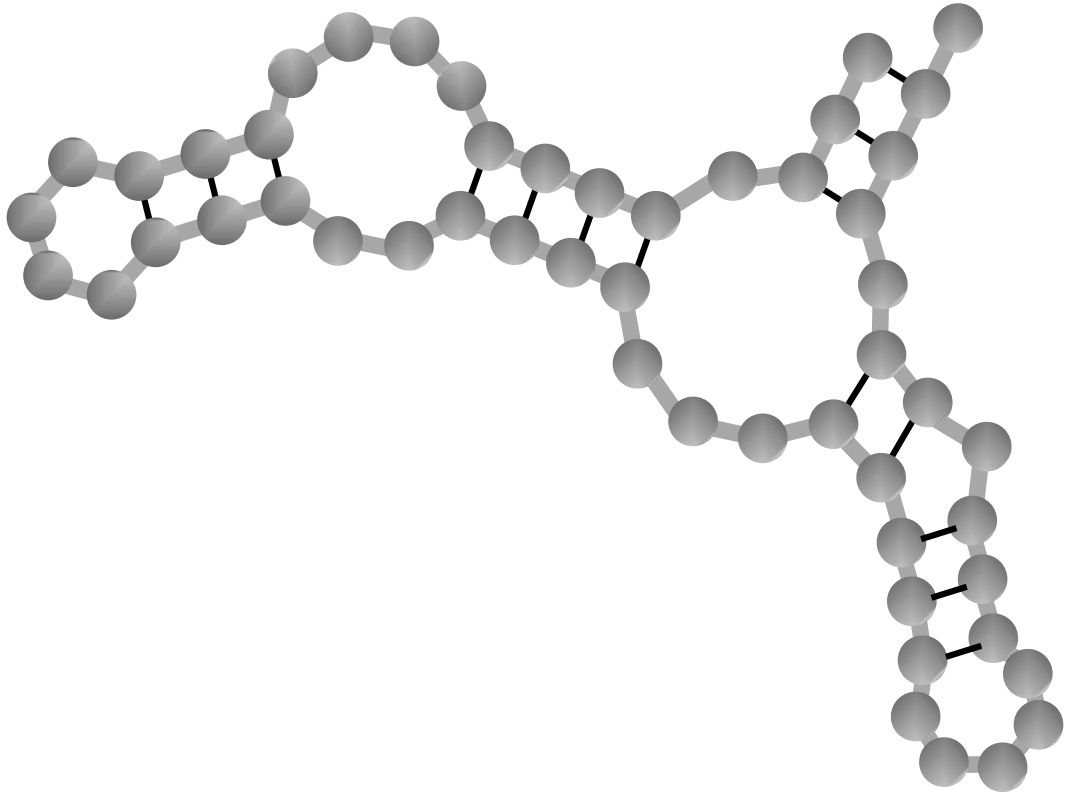
Basic Notation

- secondary structure of sequence s is a set S of base pairs $i \bullet j,\ 1 \leq i < j \leq |s|$

- we assume:
  - each base is only in one base pair
  - no pseudoknots
  - sharp “U-turns” prohibited; a hairpin loop must contain at least 3 bases

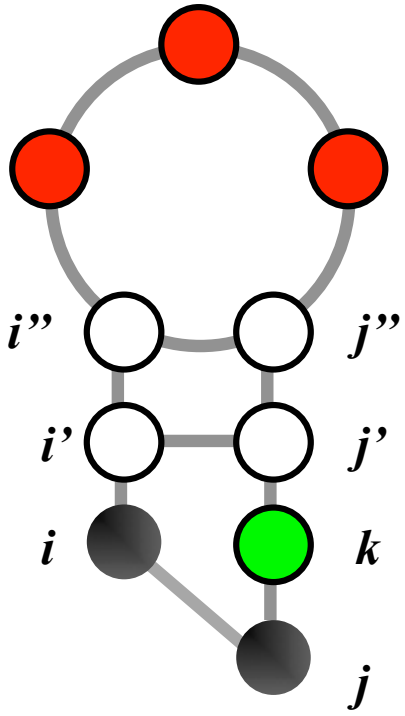
Secondary Structure Representation

- can view a structure S as a collection of loops together with some external unpaired bases



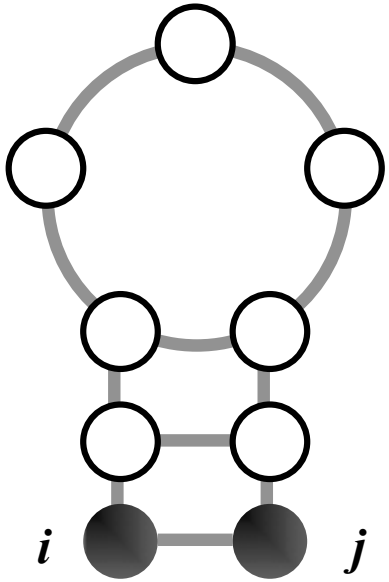
Accessible Bases

- Let $i < k < j$ with $i \bullet j \in S$
- k is *accessible* from $i \bullet j$ *if for all* $i' \bullet j' \in S$ if it is not the case that $i < i' < k < j' < j$



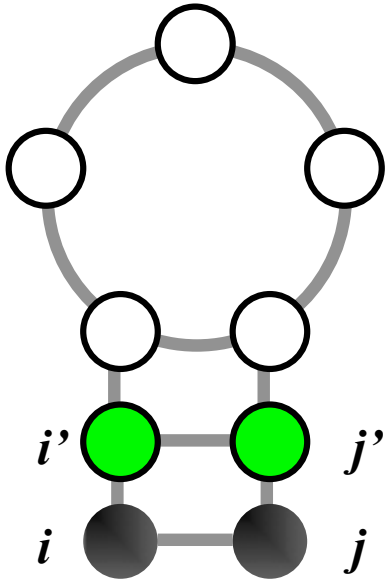
Exterior Base Pairs

- base pair $i \bullet j$ is the exterior base pair of (or closing) the loop consisting of $i \bullet j$ and all bases *accessible* from it



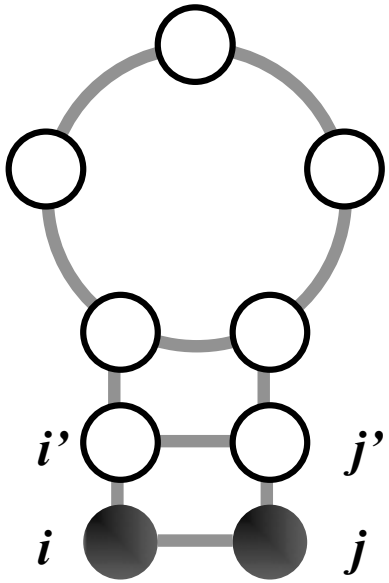
Interior Base Pairs

- if i' and j' are accessible from $i \bullet j$
- and $i' \bullet j' \in S$
- then $i' \bullet j'$ is an interior base pair, and is accessible from $i \bullet j$



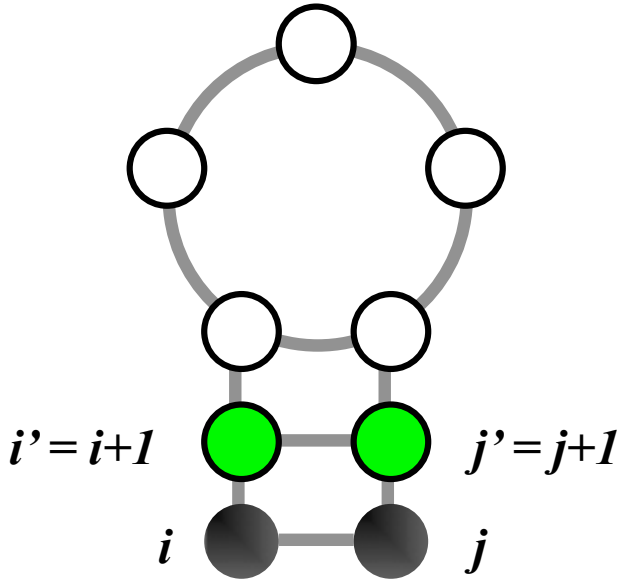
Hairpin Loop

- if there are no interior base pairs in a loop, it is a hairpin loop



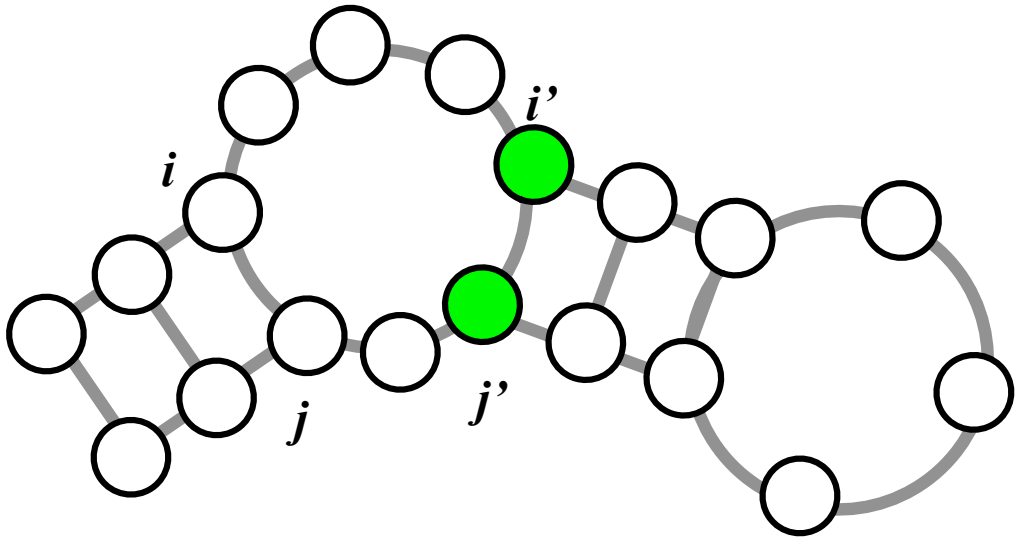
Stacked Pair

- a loop with one interior base pair is a stacked pair if $i' = i+1$ and $j' = j-1$



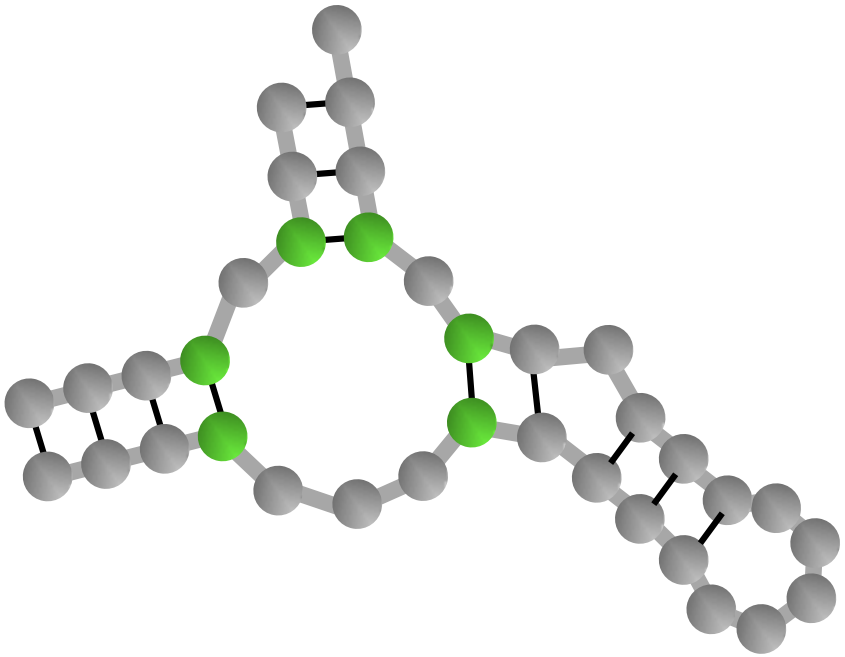
Internal Loop

- if it is not true that the interior base pair $i \bullet j$ that $i' = i+1$ and $j' = j-1$, it is an internal loop



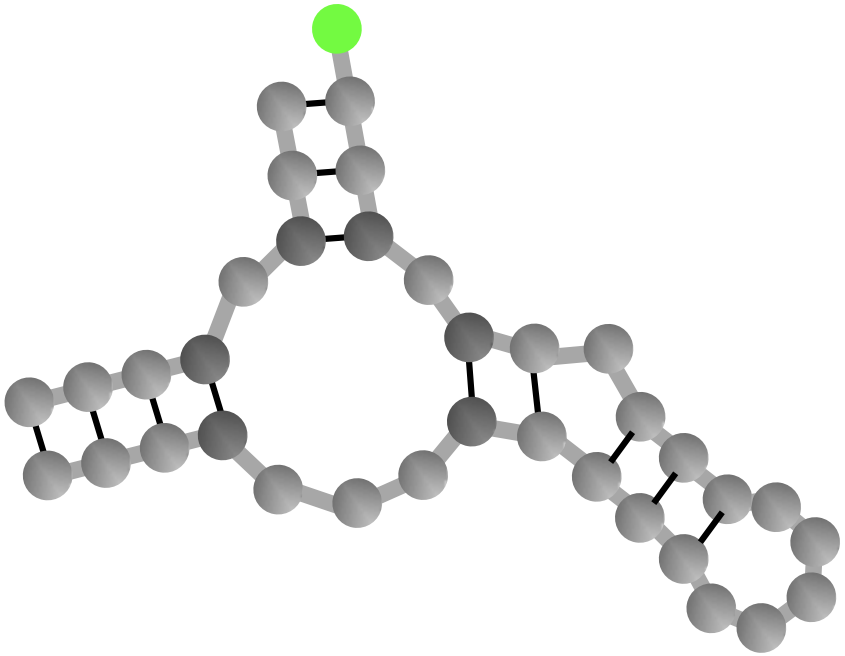
Multibranch Loops

- loops with more than one interior base pair are multibranched loops



External Bases and Base Pairs

- any bases or base pairs not accessible from any base pair are called external



Assumptions

- structure prediction determines the most stable structure for a given sequence
- stability of a structure is based on free energy
- energy of secondary structures is the sum of independent loop energies

Recursion Relation

- four arrays are used to hold the minimal free energy of specific structures of subsequences of s
- arrays are computed interdependently
- calculated recursively using pre-specified free energy functions for each type of loop

$W(i)$

- energy of an optimal structure of subsequence 1 through i:

$$W(i) = \min \begin{cases} W(i-1) \\ \min\limits_{i<j\leq i} \{W(j-1) + V(j,i)\} \end{cases}$$

$V(i,j)$

- energy of an optimal structure of subsequence i through j closed by $i \bullet j$:

$$V(i,j) = \min \begin{cases} eH(i,j) \\ eS(i,j) + V(i+1, j-1) \\ VBI(i,j) \\ VM(i,j) \end{cases}$$

eH(i,j)

- energy of hairpin loop closed by $i \bullet j$
- computed with: $\delta\delta G = 1.75 \times RT \times \ln(l_s)$,
- R = universal gas constant (1.9872 cal/mol/K).
- T = absolute temperature
- l_s = total single-stranded (unpaired) bases in loop

Loop Energy Table

DESTABILIZING ENERGIES BY SIZE OF LOOP

SIZE	INTERNAL	BULGE	HAIRPIN
1	.	3.8	.
2	.	2.8	.
3	.	3.2	5.6
4	1.7	3.6	5.5
5	1.8	4.0	5.6
6	2.0	4.4	5.3
7	2.2	4.6	5.8
8	2.3	4.7	5.4
	...		
30	3.7	6.1	7.7

eS(i,j)

- energy of stacking base pair *i•j* with *i+1•j-1*

```
          5' --> 3'
             CX
             GY
          3' <-- 5'
    Y:   A     C     G     U
   ---------------------------
X: A |   .     .     .   -2.1
   C |   .     .   -3.3   .
   G |   .   -2.4   .   -1.4
   U | -2.1   .   -2.1   .
```

- sample free energies in kcal/mole for CG base pairs stacked over all possible base pairs, XY
- '.' entries are undefined, and can be assumed as ∞

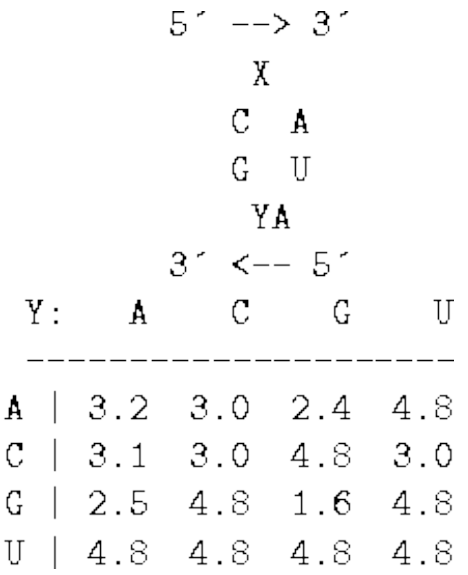
VBI(i,j)

- energy of an optimal structure of the subsequence from i through j, where $i \bullet j$ closes a bulge or an internal loop

$$VBI(i,j) = \min_{\substack{i<i'<j'<j \\ i'-i+j-j'>2}} \{eL(i,j,i',j') + V(i',j')\}$$

$eL(i,j,i',j')$

- energy of a bulge or internal loop with exterior base pair $i \bullet j$ and interior base pair $i' \bullet j'$



Y:	A	C	G	U
A	3.2	3.0	2.4	4.8
C	3.1	3.0	4.8	3.0
G	2.5	4.8	1.6	4.8
U	4.8	4.8	4.8	4.8

- free energies for all 1 x 2 interior loops in RNA closed by a CG and an AU base pair, with a single stranded U 3' to the double stranded U.

VM(i,j)

- energy of an optimal structure of the subsequence from i through j, where $i \bullet j$ closes a multibranched loop

$$VM(i,j) = \min_{\substack{i<i_1<j_1< \\ \dots \\ <i_k<j_k<j}} \left\{ eM(i,j,i_1,j_1,\dots,i_k,j_k) + \sum_{l=1}^{k} V(i_l,j_l) \right\}$$

$eM(i,j,i_1,j_1,\ldots,i_k,j_k)$

- energy of a multibranched loop with exterior base pair $i \bullet j$ and interior base pairs $i_1 \bullet j_1, \ldots, i_k \bullet j_k$
- simplification: linear contributions from number of unpaired bases in loop, number of branches and a constant

$$eM(i,j,i_1,j_1,\ldots,i_k,j_k)$$

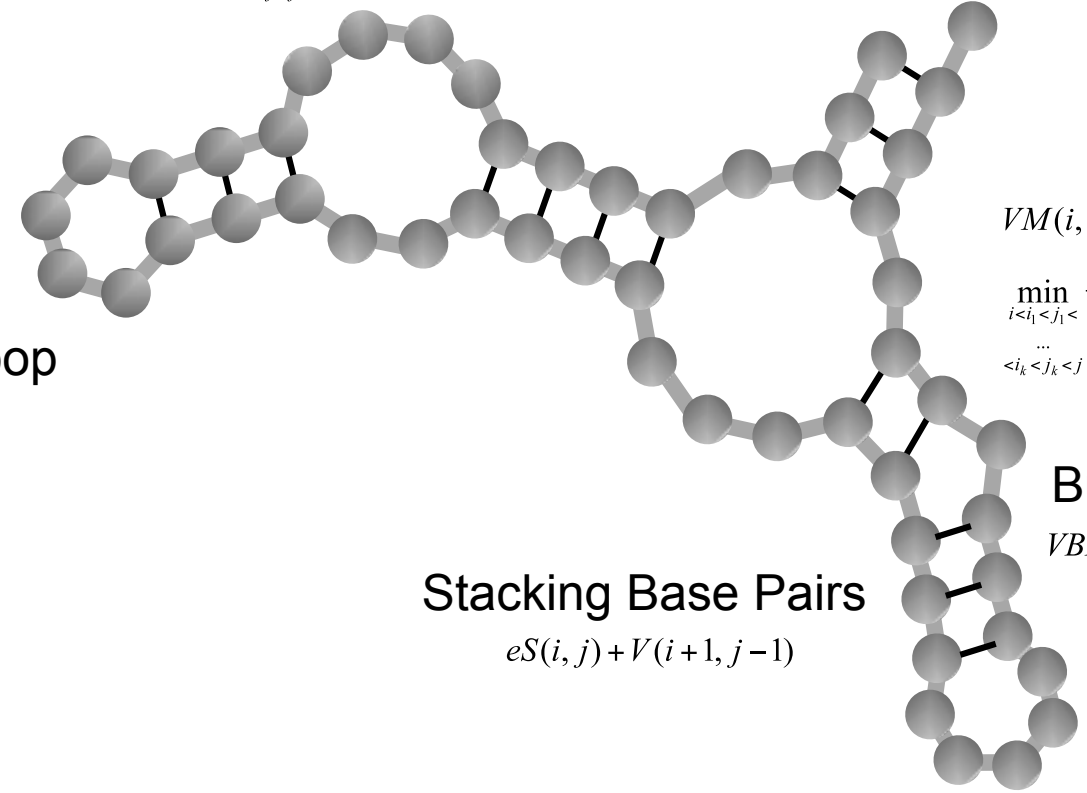
$$= a + bk + c(i_1 - i - 1 + j - j_k - 1 + \sum_{l=1}^{k-1}(i_l + 1 - j_l + 1))$$

eM refactored as $VM(i,j)$

- energy of an optimal structure of subsequence i – j constituting part of a multibranched loop structure
- unpaired bases and external base pairs are penalized as per the previous equation:

$$WM(i,j) = \min\left\{\begin{array}{c} V(i,j)+b \\ WM(i,j-1)+c \\ WM(i+1,j)+c \\ \min\limits_{i<k\le j}\{WM(i,k-1)+WM(k,j)\} \end{array}\right\}$$

Assembling the Pieces



Internal Loop

$$VBI(i,j) = \min_{\substack{i<i'<j'<j \\ i'-i+j-j'>2}} \{eL(i,j,i',j') + V(i',j')\}$$

External Base

Multi-loop

$$VM(i,j) = \min_{\substack{i<i_1<j_1< \\ \cdots \\ <i_k<j_k<j}} \{eM(i,j,i_1,j_1,\ldots,i_k,j_k) + \sum_{l=1}^{k} V(i_l,j_l)\}$$

Hairpin Loop

$$eH(i,j)$$

Bulge

$$VBI(i,j) = \min_{\substack{i<i'<j'<j \\ i'-i+j-j'>2}} \{eL(i,j,i',j') + V(i',j')\}$$

Stacking Base Pairs

$$eS(i,j) + V(i+1,j-1)$$

The Trouble with Internal Loops

- objective of this paper is to reduce the computational complexity from $O(|s|^4)$ to $O(|s|^3)$

- the most computationally complex element of the four different secondary structure types is $VBI(i,j)$, or bulge or internal loops

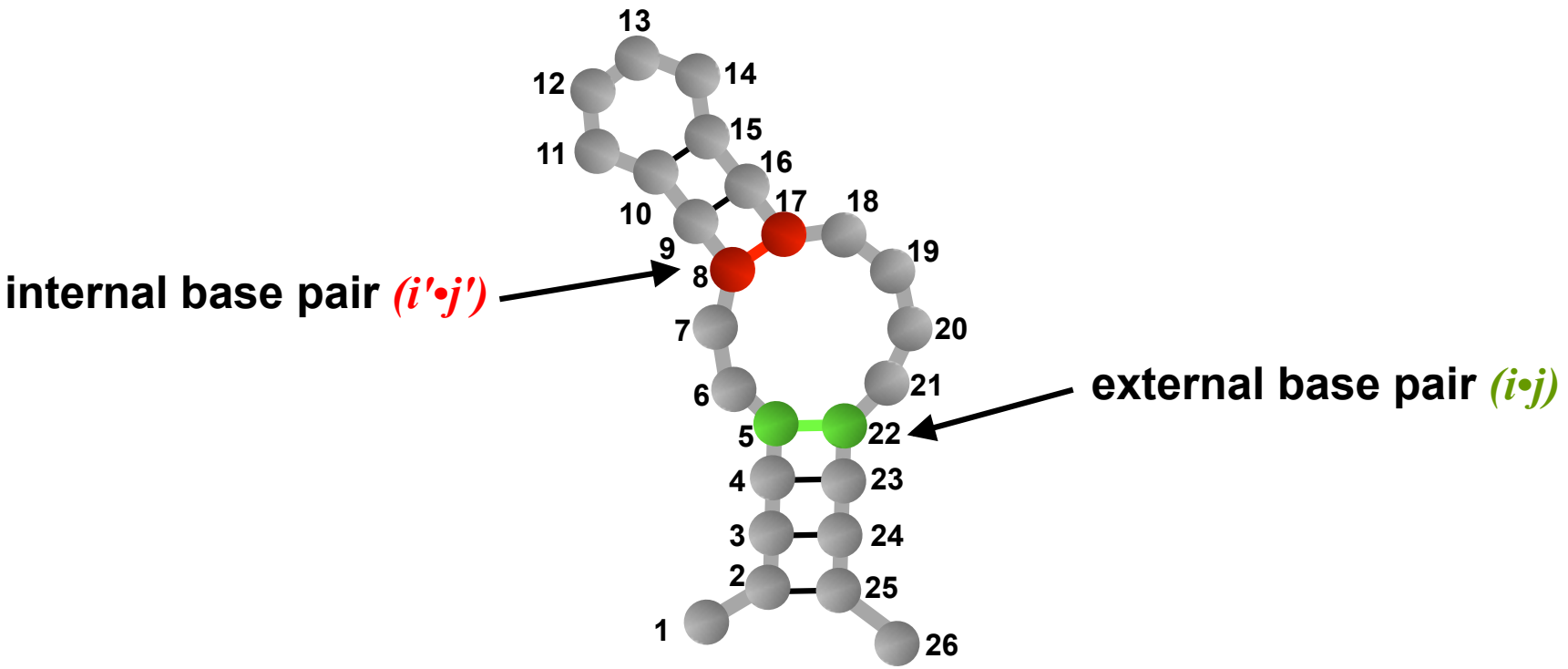
Internal Loops Revisited

- computational complexity: all possible base pairs accessible to i and j are considered for all i and j computed in VBI

$$VBI(i,j) = \min_{\substack{i<i'<j'<j \\ i'-i+j-j'>2}} \{eL(i,j,i',j') + V(i',j')\}$$

- also add destabilizing loop energy and energy of optimal substructure closed by $(i' \bullet j')$, the complexity is $O(|s|^4)$

Example Internal Loop



$$VBI(5,22) = \min_{\substack{5<i'<j'<22 \\ i'-5+22-j'>2}} \{eL(5,22,i',j') + V(i',j')\}$$

Simplifying the Energy Computation

- the energy function eL for internal loops can be split into three components:

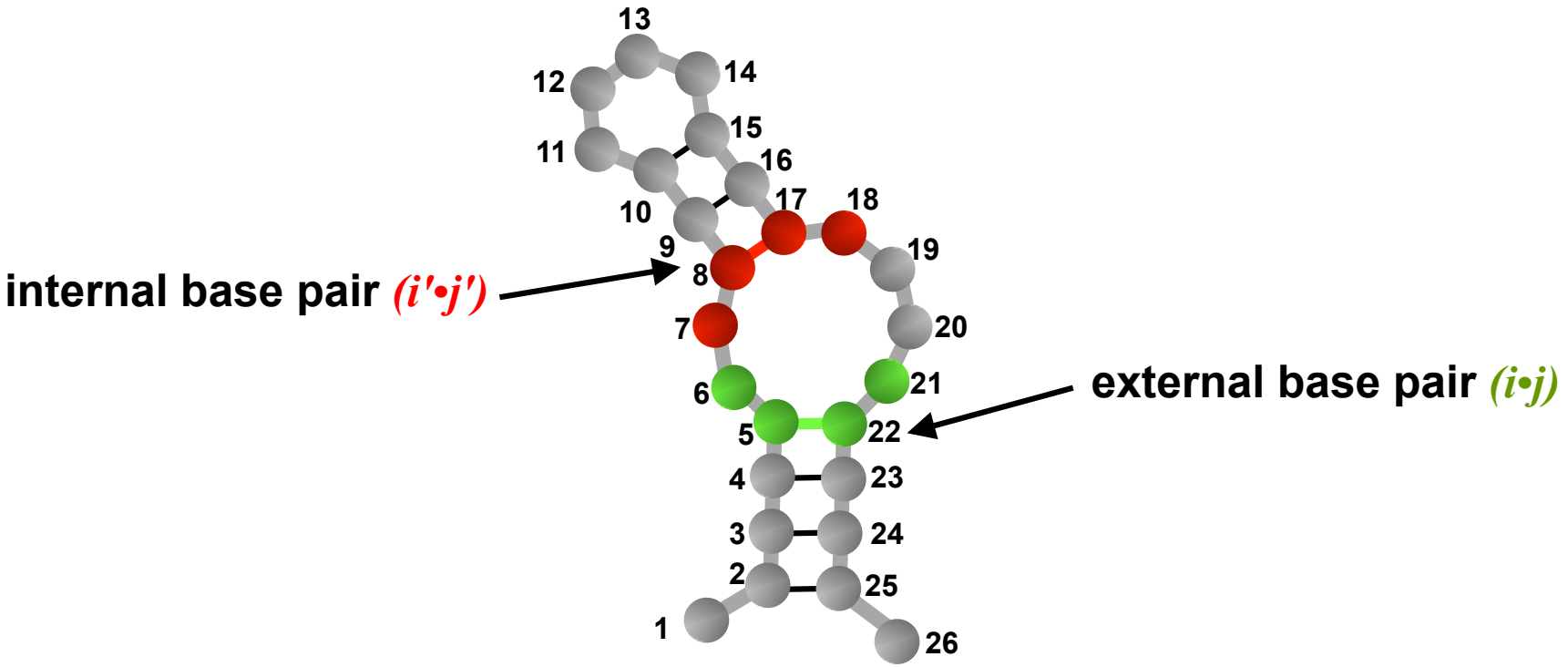
$$eL(i, j, i', j') = size(i' - i + j - j' - 2) + \quad \textbf{(1)}$$

$$asymmetry(i' - i - 1, j - j' - 1) + \quad \textbf{(2)}$$

$$stacking(i \bullet j) + stacking(i' \bullet j') \quad \textbf{(3)}$$

1. entropic term depending on size of the loop
2. asymmetric penalty for asymmetric loops
3. stacking energies of interior and exterior base pairs with the nearest unpaired bases

Example *eL(i,j,i',j')* Computation



$$eL(5,22,8,17) = size((8)-(5)+(22)-(17)-2)+ \quad size(6)+$$
$$asymmetry((8)-(5)-1,(5)-(22)-1)+ \quad asymmetry(2,4)+$$
$$stacking(5\bullet 22)+stacking(8\bullet 17) \quad stacking(5\bullet 22)+stacking(8\bullet 17)$$

Dealing with Asymmetry Penalty

- we assume that lopsidedness and size dependence of asymmetry can be separated out:

$$asymmetry(n_1, n_2) = \min\{E_{\max}, n \bullet f(m)\}$$

$$n = |n_1 - n_2|, m = \min\{n_1, n_2, c\}$$

- main idea: if we fix lopsidedness, asymmetry penalty doesn't change with size

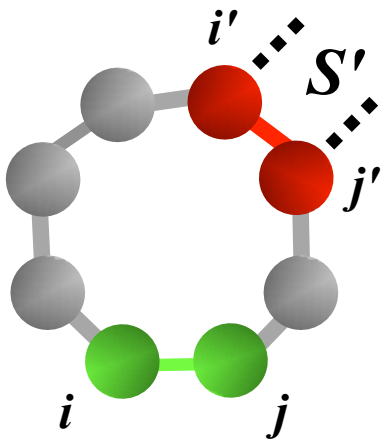
$$asymmetry(n_1, n_2) = asymmetry(n_1 + 1, n_2 + 1)$$

The Payoff

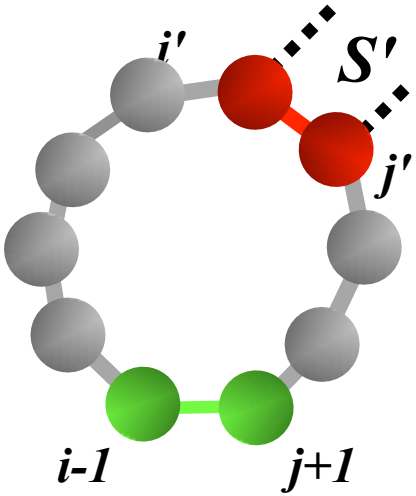
- for internal loops of size l and shortest length of unpaired bases c, if we know:
  - the optimal interior base pair $(i' \bullet j')$
  - the exterior base pair $(i \bullet j)$

- we can find the optimal interior base pair for loop size l+2 with exterior base pair $(i+1 \bullet j+1)$ in constant time

Lopsided Illustration



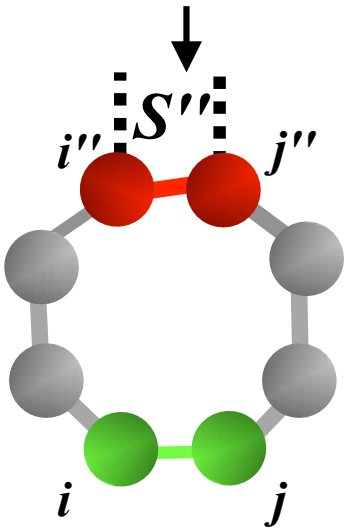
shift closing pair from (*i, j*) to (*i',j'*)

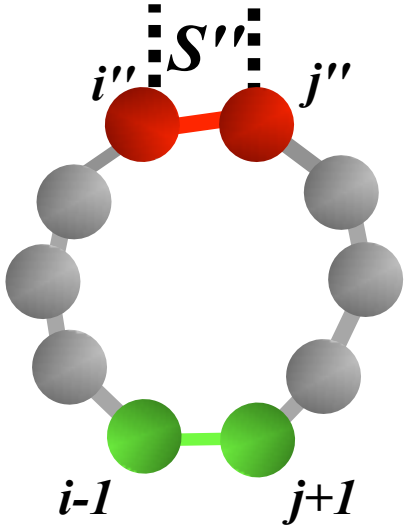


lopsided to straight

Change in size +

stacking(*i-1, j-1*) - stacking(*i, j*)





The Algorithm

- compare structure with interior base pair $(i' \bullet j')$ with the two structures with an interior base pair that gives a shortest length of c unpaired bases

- algorithm evaluates internal loops of size $2l + a$ with exterior base pair $i\text{-}l \bullet j+l+a$ and shortest length of at least c unpaired bases

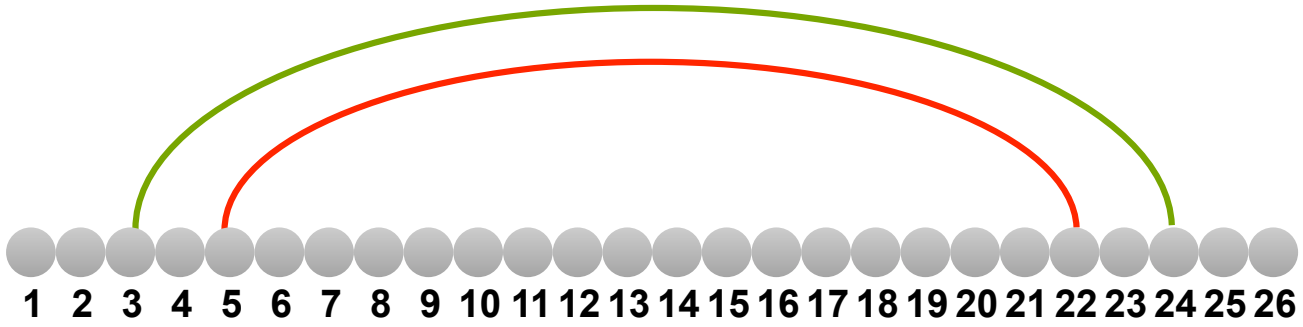
Algorithm Pseudocode

```
Require: i, j with i < j
    For a = 0 to 1 do          // a=0 for even, a=1 for odd sized loops
        E=∞   // energy of optimal loop excepting size and external stacking
        For l = c + 1 to min{i-1,|s|-j-a} do
            E = min {E,
                V(i-l+c+1,j-l+c+1)+
                  asymmetry(c,2l+a-c-2)+
                  stacking(i-l+c+1,j-l+c+1),   // Examine two new
                V(i+a+l-c-1,j+a+l-c-1)+        // candidate base pairs
                  asymmetry(2l+a-c-2,c)+       // i.e. interior base pairs next to
                  stacking(i-l+c+1,j-l+c+1)}  // current exterior base pair
                VBI(i-l,j+a+l)=
                  min{VBI(i-l,j+a+l),
                  E+size(2l+a-2)+stacking(i-l,j+a+l)}  // update VBI for current
        end for                                          // exterior base pair
    end for
```

Algorithm Walkthrough (5,22)

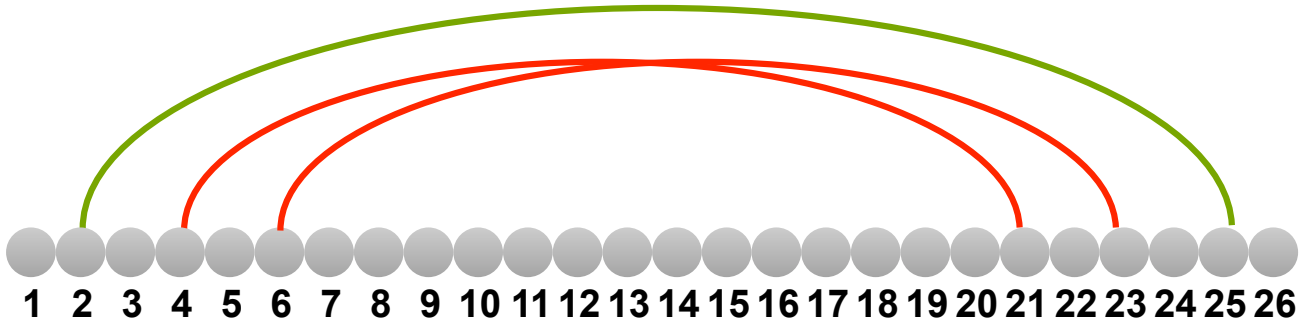
V(5,22) + asymmetry(1,1) + stacking(5,22)

VBI(3,24)



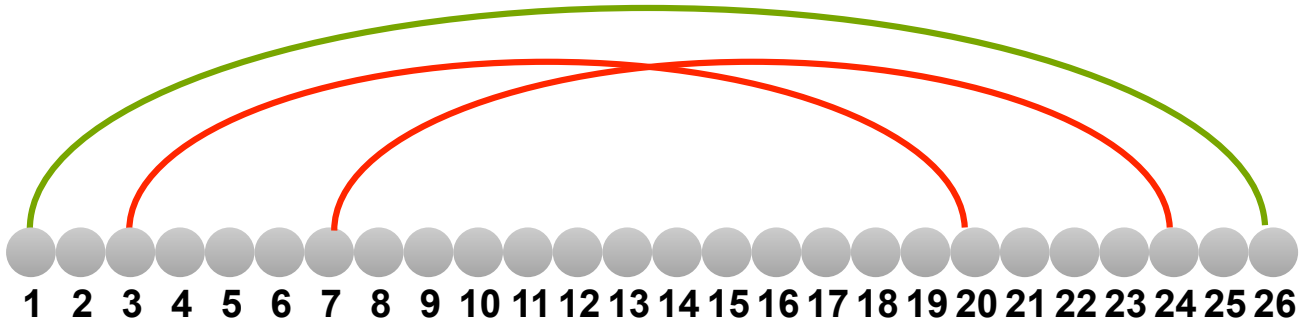
Algorithm Walkthrough (5,22)

V(4,21) + asymmetry(1,3) + stacking(4,21)
V(6,23) + asymmetry(3,1) + stacking(6,23)
VBI(2,25)



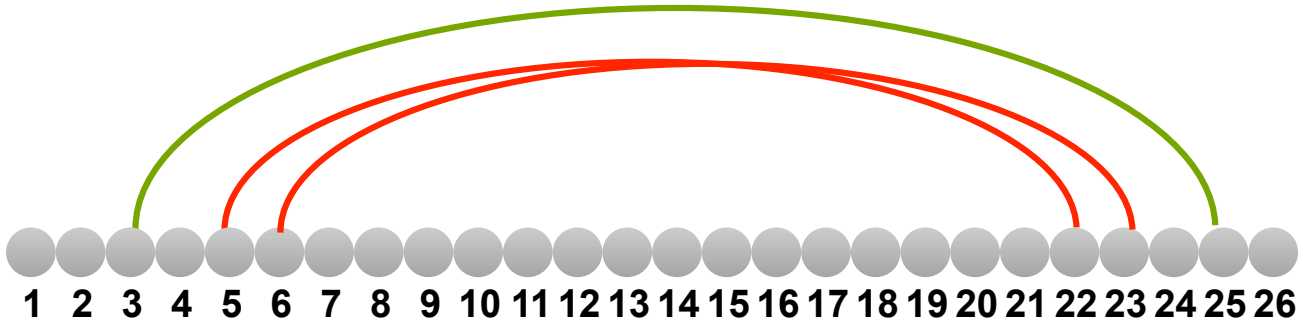
Algorithm Walkthrough (5,22)

V(3,20) + asymmetry(1,5) + stacking(3,20)
V(7,24) + asymmetry(5,1) + stacking(7,24)
VBI(1,26)



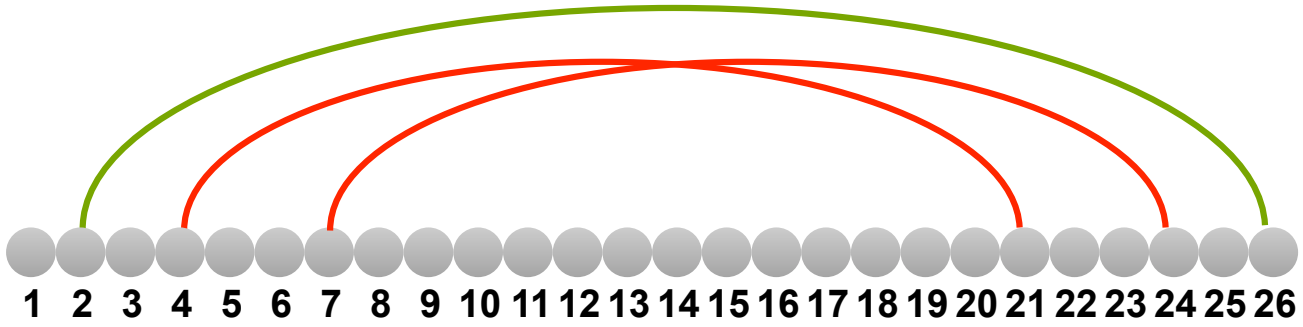
Algorithm Walkthrough (5,22)

V(5,22) + asymmetry(1,2) + stacking(5,22)
V(6,23) + asymmetry(2,1) + stacking(6,23)
VBI(3,25)



Algorithm Walkthrough (5,22)

V(4,21) + asymmetry(1,4) + stacking(4,21)
V(7,24) + asymmetry(4,1) + stacking(7,24)
VBI(2,26)



End Result

- $O(|s|^3)$ algorithm for internal loops with shortest stretch of unpaired bases c
- $O(c|s|^3)$ needed to consider all internal loops (evaluate these individually)
- experiments performed on artificial sequence, Qβ, and Thermococcus celer

Experimental Results

1. artificial sequence: resolves double-bulge problem
2. Coliphage Qβ RNA: unable to find any structures found by Jacobson (1991)
3. Thermococcus celer: found some key elements

Conclusion

- tried predicting structures at high temperatures to generate large (~30) loops
- energy parameters extrapolated for high temperatures do not support long range base pairing

References

- Durbin, R., Eddy, S., Krogh, A, & Mitchison, G. (1998) *Biological Sequence Analysis* (Cambridge University Press, Cambridge).
- R. B. Lyngsø, M. Zuker, and C. N. S. Pedersen. (1999) *Internal loops in RNA secondary structure prediction.* In Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB),
- R. Nussinov, G. Piecznik, J. R. Grigg and D. J. Kleitman, (1978) *Algorithms for loop matchings*, SIAM Journal on Applied Mathematics 35, 68-82.
- M. Zuker and P. Stiegler, (1981) *Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information*, Nucleic Acid Res. 9, 133-148. 12
- R.B. Lyngsø, M. Zuker, and C.N.S. Pedersen. (1999) *An Improved Algorithm for RNA Secondary Structure Prediction.* Tech-report BRICS RS-99-15.