CS481: Bioinformatics Algorithms

Can Alkan
EA224
calkan@cs.bilkent.edu.tr

http://www.cs.bilkent.edu.tr/~calkan/teaching/cs481/

Quiz 4

Given the following distance table, construct its corresponding distance graph, and use the CAST algorithm to find the cliques/clusters

$$\Theta = 8$$

	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11
g1	-	11	4	17	2	13	1	13	13.5	9	2
g2	11	-	9	9.5	13	11	16	1	6.5	2	8.8
g3	4	9	-	11.2	1	13.4	2	8.1	9.2	6.8	1.4
g4	17	9.5	11.2	-	18	14	13.6	7.2	9.9	16.5	9.6
g5	2	13	1	18	-	9	1	6.8	12.1	8.7	3
g6	13	11	13.4	14	9	-	8.9	13.2	1.1	11.5	9.6
g7	1	16	2	13.6	1	8.9	-	9.7	10.5	13.2	11.3
g8	13	1	8.1	7.2	6.8	13.2	9.7	-	9.6	1.6	10.6
g9	13.5	6.5	9.2	9.9	12.1	1.1	10.5	9.6	-	8.7	12.1
g10	9	2	6.8	16.5	8.7	11.5	13.2	1.6	8.7	-	9.9
g11	2	8.8	1.4	9.6	3	9.6	11.3	10.6	12.1	9.9	-

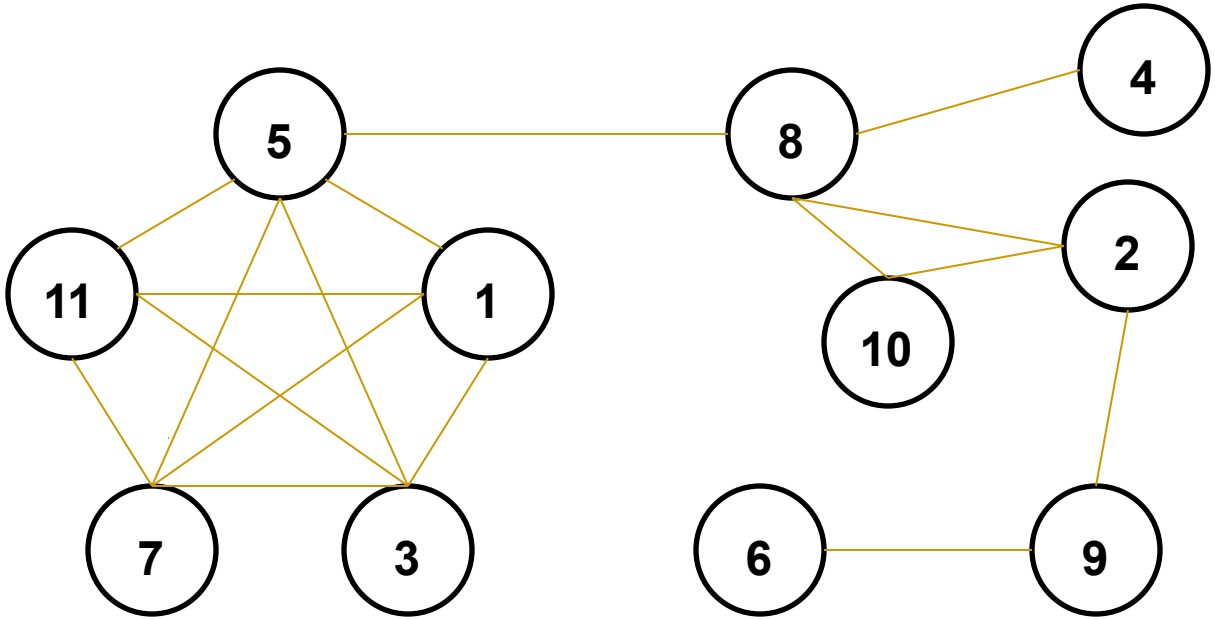
Quiz 4

Given the following distance table, construct its corresponding distance graph, and use the CAST algorithm to find the cliques/clusters

$$\Theta = 8$$

Degree		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11
4	**g1**	-	11	4	17	2	13	1	13	13.5	9	2
3	**g2**	11	-	9	9.5	13	11	16	1	6.5	2	8.8
4	**g3**	4	9	-	11.2	1	13.4	2	8.1	9.2	6.8	1.4
1	**g4**	17	9.5	11.2	-	18	14	13.6	7.2	9.9	16.5	9.6
5	**g5**	2	13	1	18	-	9	1	6.8	12.1	8.7	3
1	**g6**	13	11	13.4	14	9	-	8.9	13.2	1.1	11.5	9.6
4	**g7**	1	16	2	13.6	1	8.9	-	9.7	10.5	13.2	11.3
4	**g8**	13	1	8.1	7.2	6.8	13.2	9.7	-	9.6	1.6	10.6
2	**g9**	13.5	6.5	9.2	9.9	12.1	1.1	10.5	9.6	-	8.7	12.1
2	**g10**	9	2	6.8	16.5	8.7	11.5	13.2	1.6	8.7	-	9.9
4	**g11**	2	8.8	1.4	9.6	3	9.6	11.3	10.6	12.1	9.9	-

Quiz 4



C1={g1,g3,g5,g7,g11} C2={g2,g8,g10} C3={g6,g9} C4={g4}

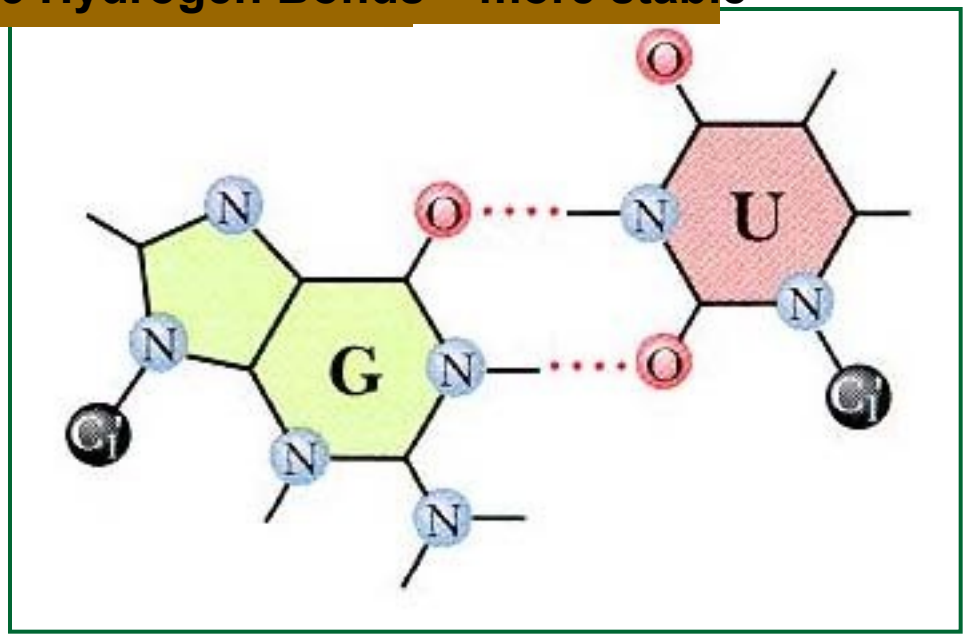
RNA STRUCTURE

RNA Basics

- RNA bases A,C,G,U
- Canonical Base Pairs
  - A-U
  - G-C
  - G-U

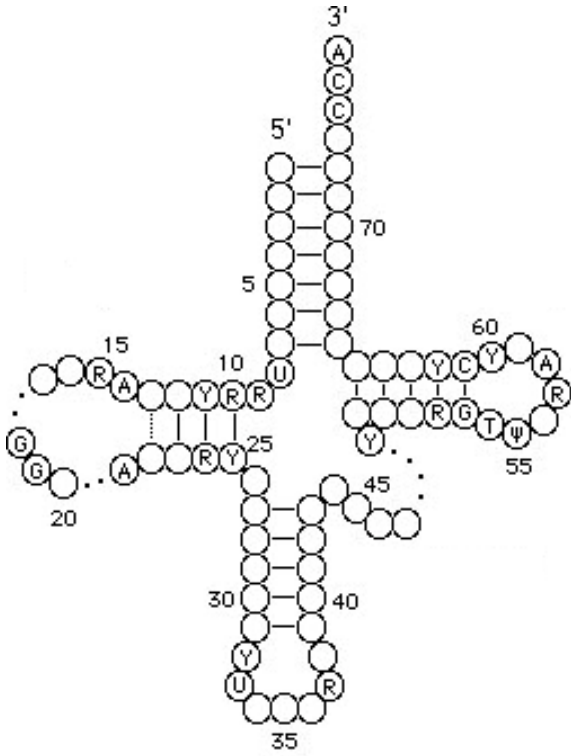
    “wobble” pairing
  - Bases can only pair with **one** other base.

3 Hydrogen Bonds – more stable



RNA Basics

- transfer RNA (tRNA)
- messenger RNA (mRNA)
- ribosomal RNA (rRNA)
- small interfering RNA (siRNA)
- micro RNA (miRNA)
- small nucleolar RNA (snoRNA)



http://www.genetics.wustl.edu/eddy/tRNAscan-SE/

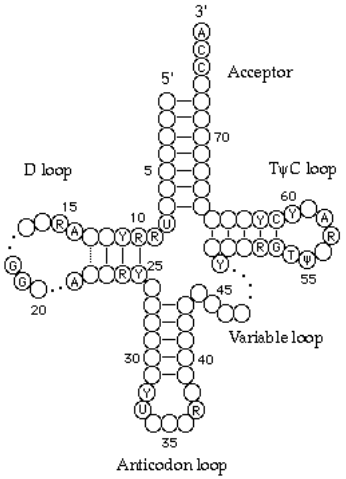
RNA folding

- Prediction of secondary structure of an RNA given its sequence
- General problem is NP-hard due to “difficult” substructures, like pseudoknots
- Most existing algorithms require too much memory ($\geq O(n^2)$), and run time ($\geq O(n^3)$) thus limited to smaller RNA sequences

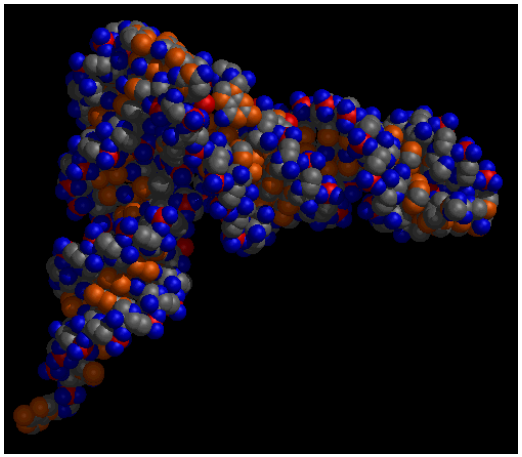
RNA Structural Levels

AAUCG...CUUCUUCCA

Primary



Secondary



Tertiary

RNA families

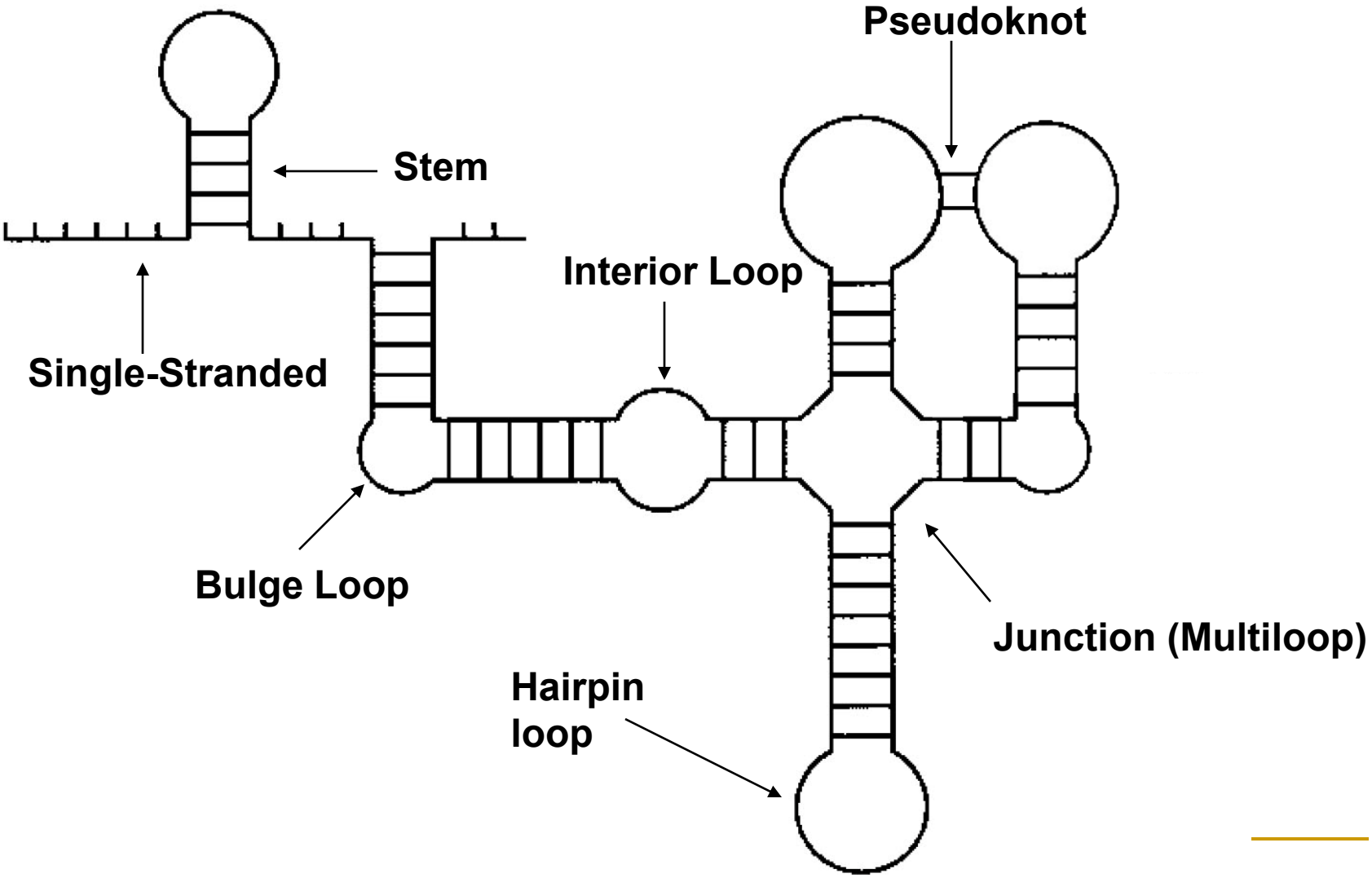
- Rfam : General non-coding RNA database

(most of the data is taken from specific databases)

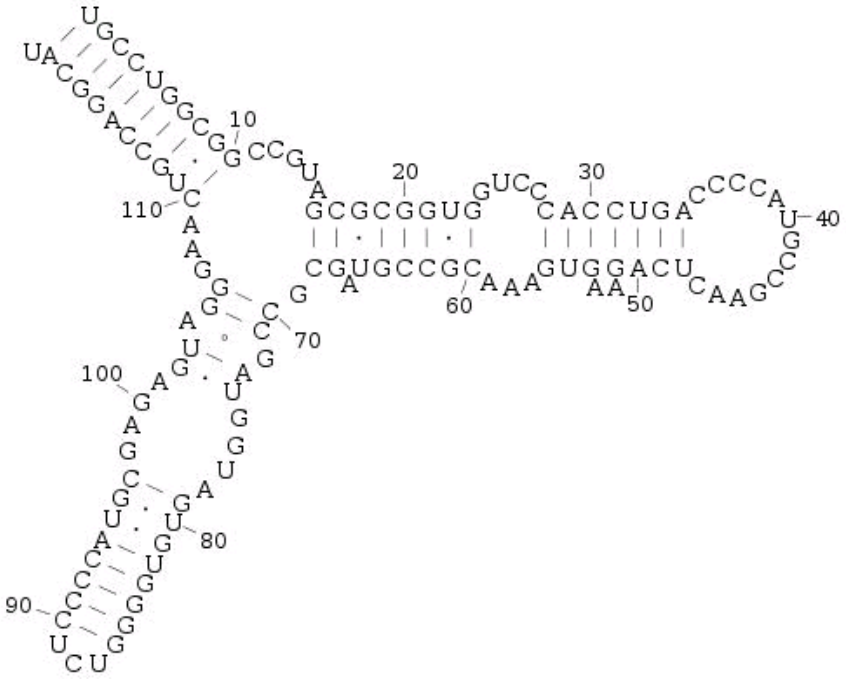
http://www.sanger.ac.uk/Software/Rfam/

Includes many families of non coding RNAs and functional Motifs, as well as their alignement and their secondary structures

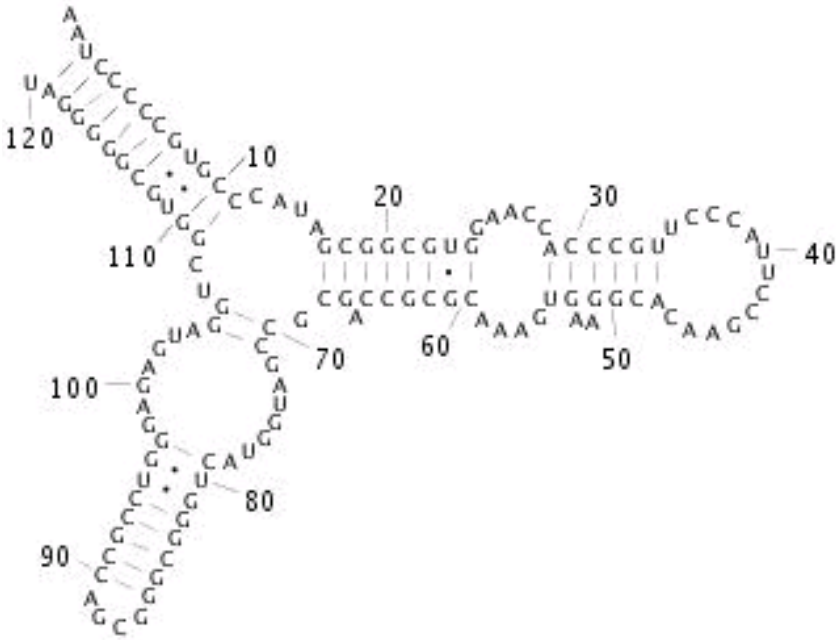
RNA Secondary Structure



Example: 5S rRNA

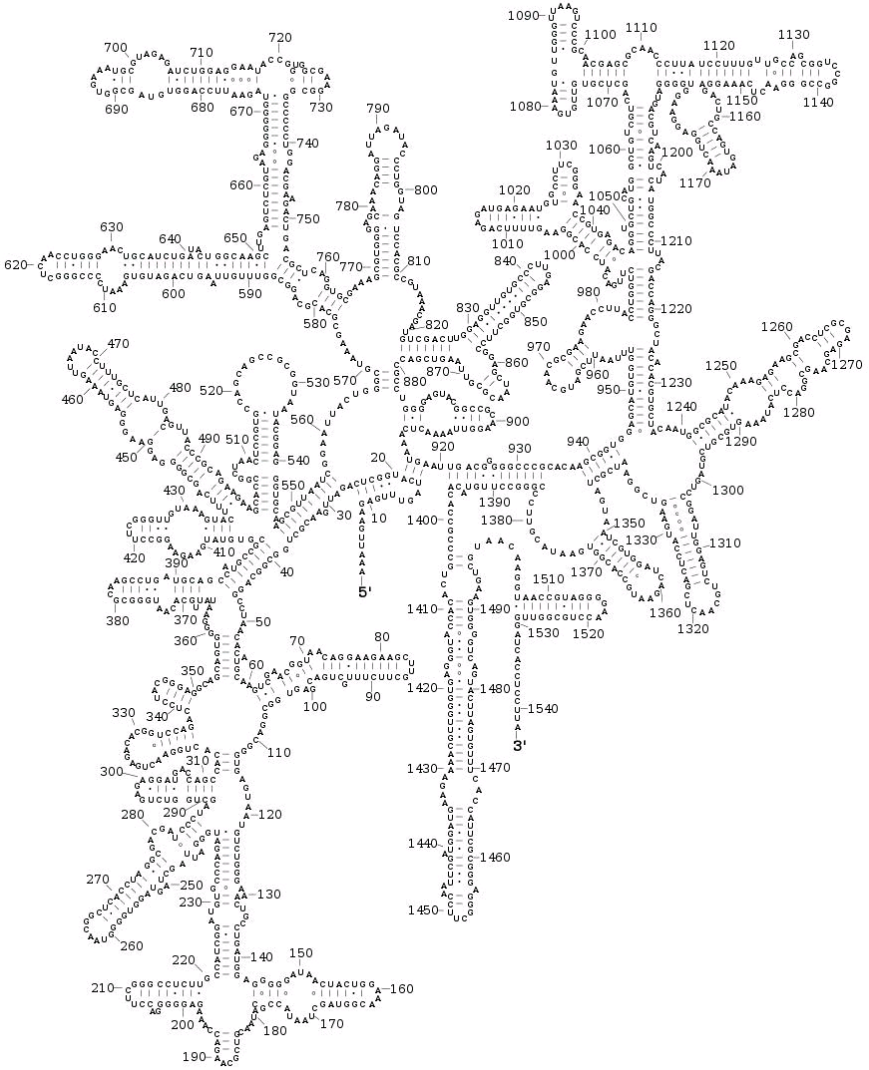


E. coli 5S
120 bases



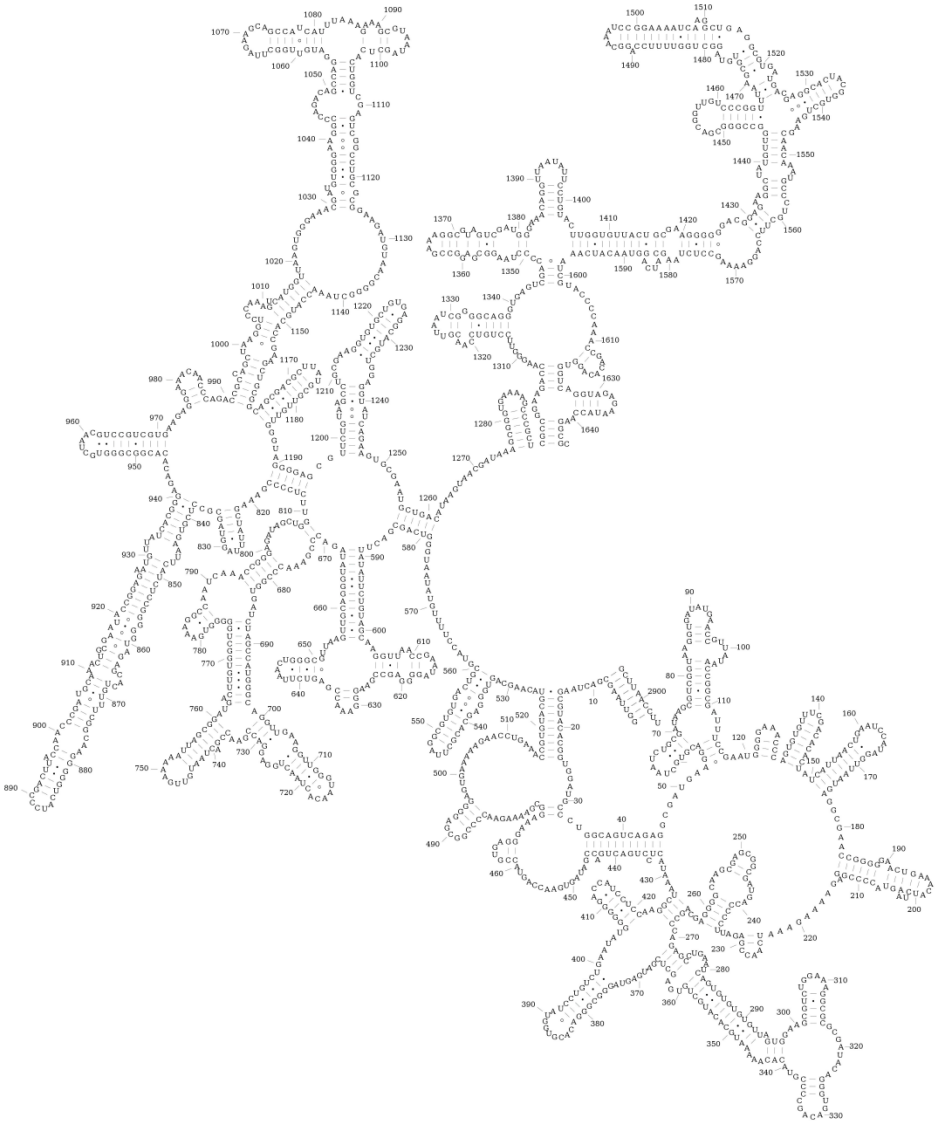
T. thermophilus 5S
120 bases

Example: E. coli 16S rRNA



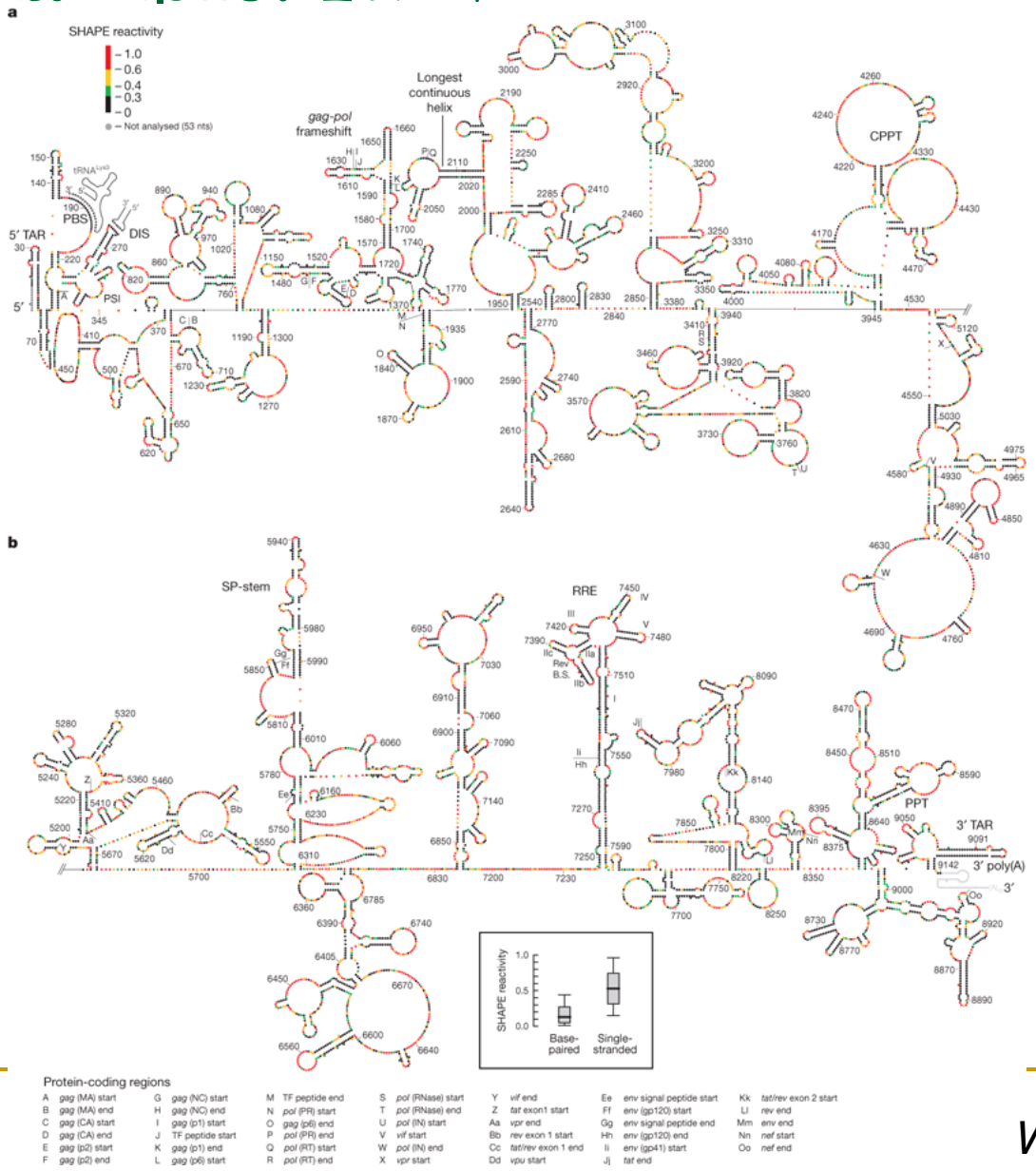
1542 bases

Example: E. coli 23S rRNA



2904 bases

Example: HIV

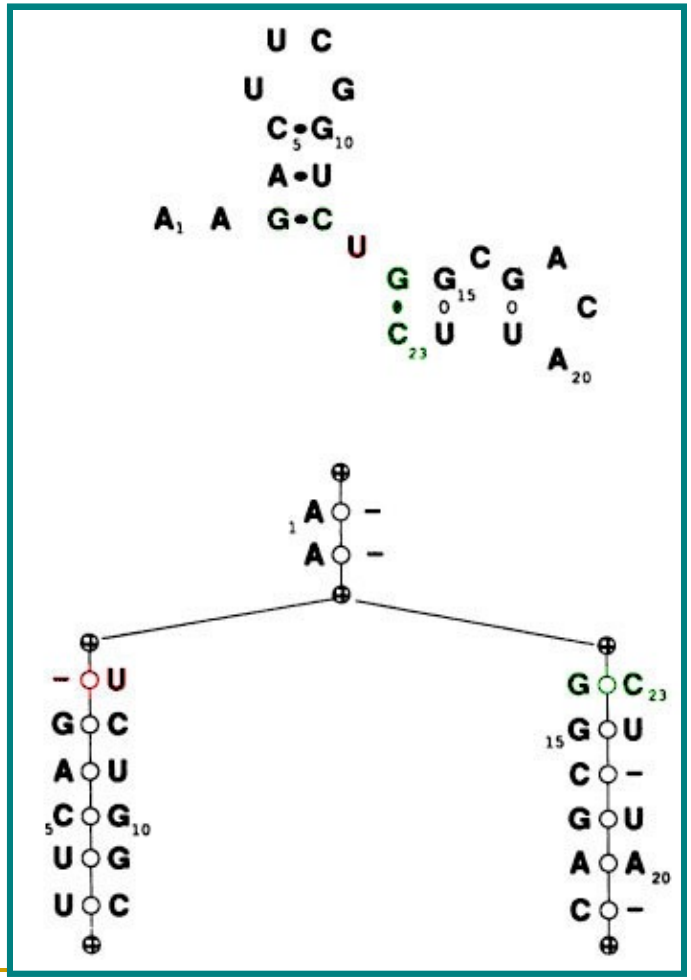


9173 bases

Watts et al., Nature, 2009

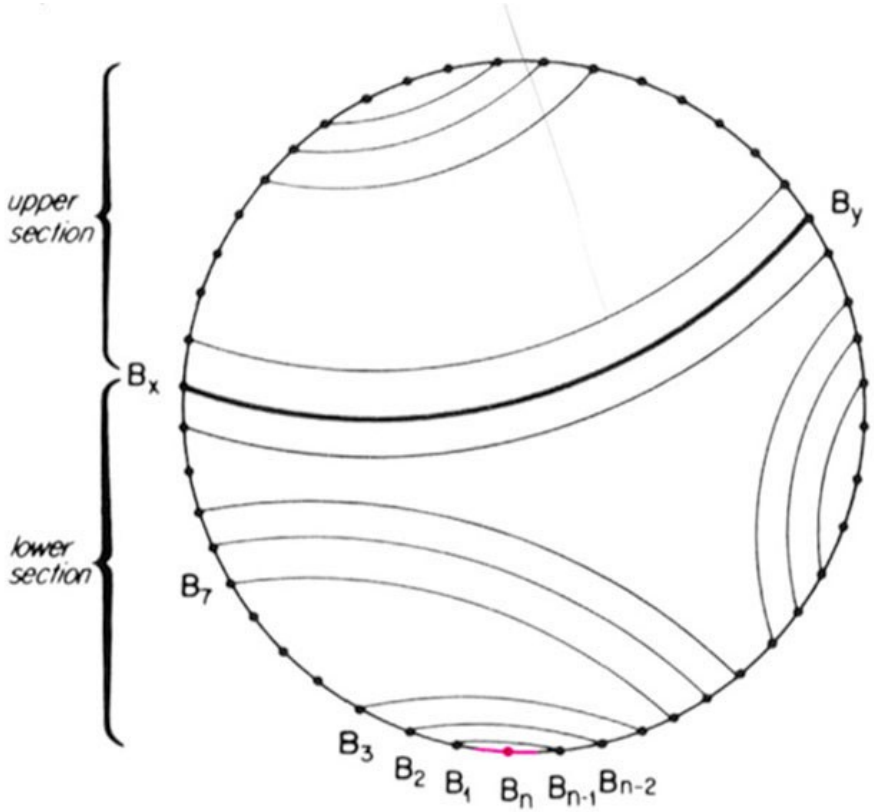
Binary Tree Representation of RNA Secondary Structure

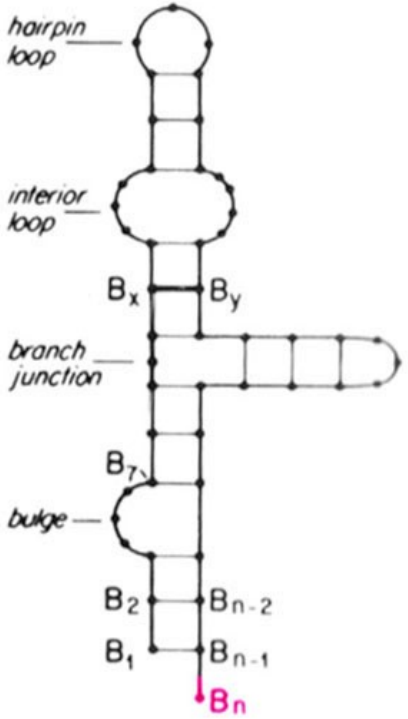
- Representation of RNA structure using Binary tree
- Nodes represent
  - Base pair if two bases are shown
  - Loop if base and "gap" (dash) are shown
- Pseudoknots still not represented
- Tree does not permit varying sequences
  - Mismatches
  - Insertions & Deletions



Images – Eddy et al.

Circular Representation

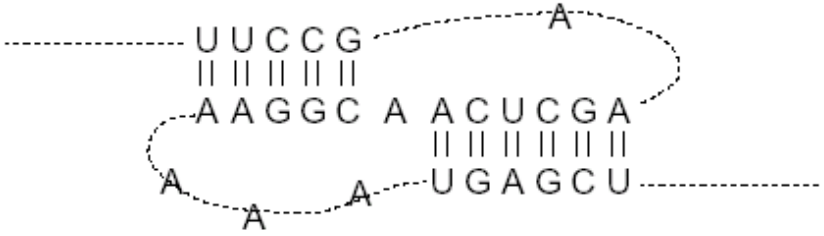




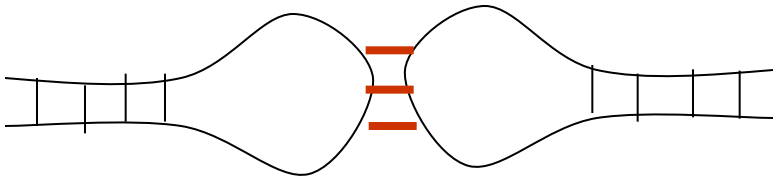
Images – David Mount

Examples of known interactions of RNA secondary structural elements

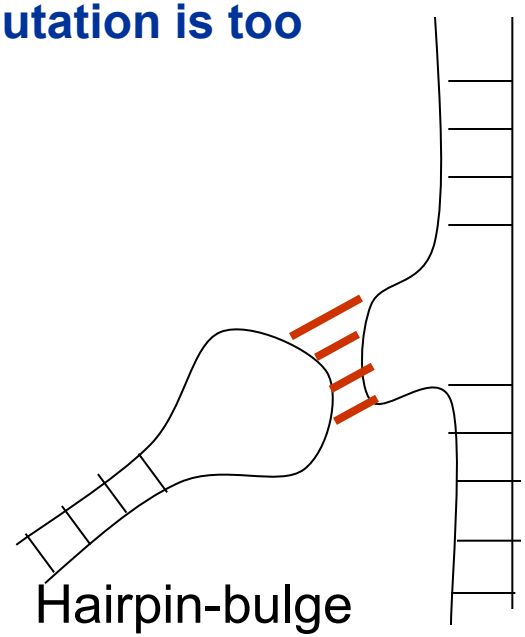
Pseudoknot



These patterns are excluded from the prediction schemes as their computation is too intensive.



Kissing hairpins



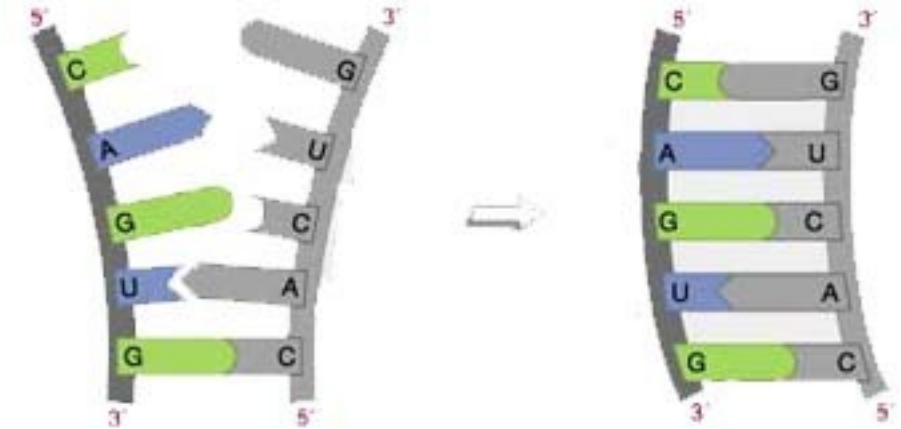
Hairpin-bulge contact

Predicting RNA secondary structure

- Base pair maximization
- Minimum free energy (most common)
  - Fold, Mfold (Zuker & Stiegler)
  - RNAfold (Hofacker)
- Multiple sequence alignment
  - Use known structure of RNA with similar sequence
- Covariance
- Stochastic Context-Free Grammars

Sequence Alignment as a method to determine structure

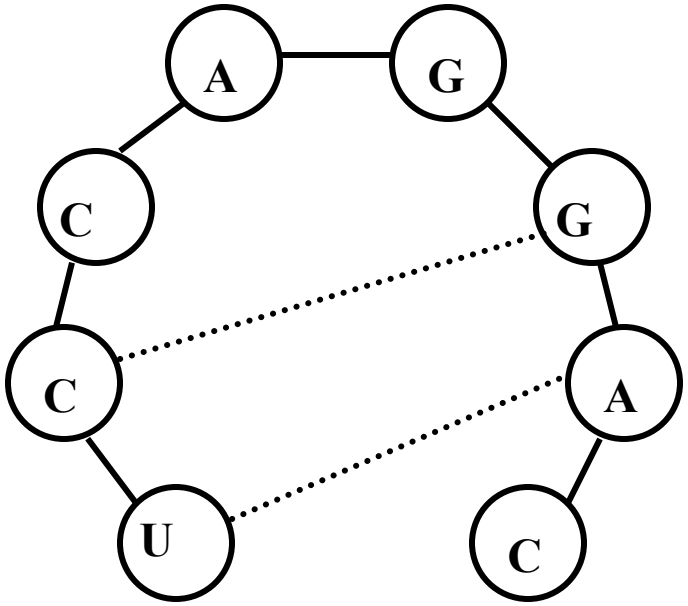
- Bases pair in order to form backbones and determine the secondary structure
- Aligning bases based on their ability to pair with each other gives an algorithmic approach to determining the optimal structure

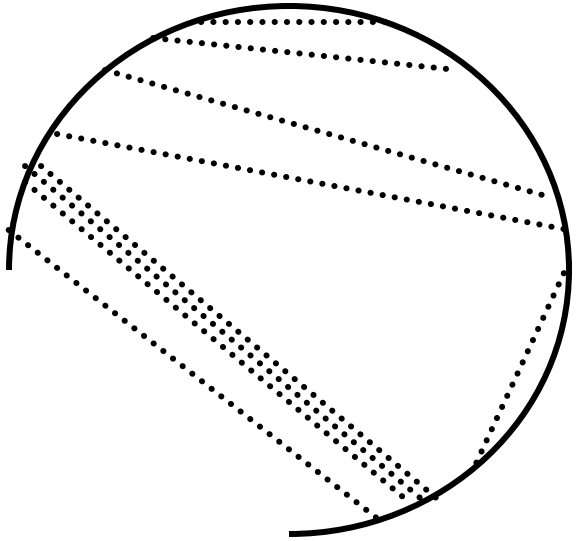


Simplifying Assumptions

- RNA folds into one minimum free-energy structure.
- There are no knots (base pairs never cross).
- The energy of a particular base pair in a double stranded regions is sequence independent
  - Neighbors do not influence the energy.
- Was solved by dynamic programming, Zuker and Stiegler 1981

Base Pair Maximization





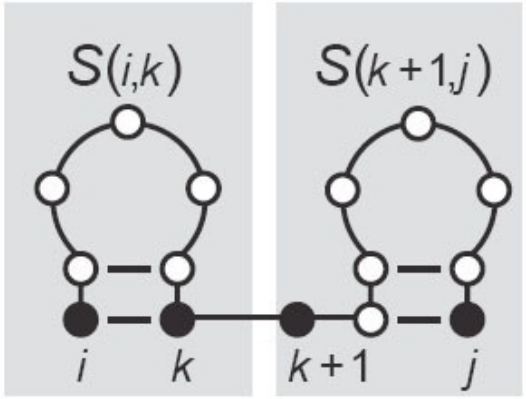
Zuker (1981) *Nucleic Acids Research* 9(1) 133-149

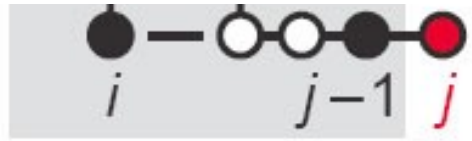
Base Pair Maximization – Dynamic Programming Algorithm

S(i,j) is the folding of the subsequence of the RNA strand from index i to index j which results in the highest number of base pairs

Maximizing Base Pair

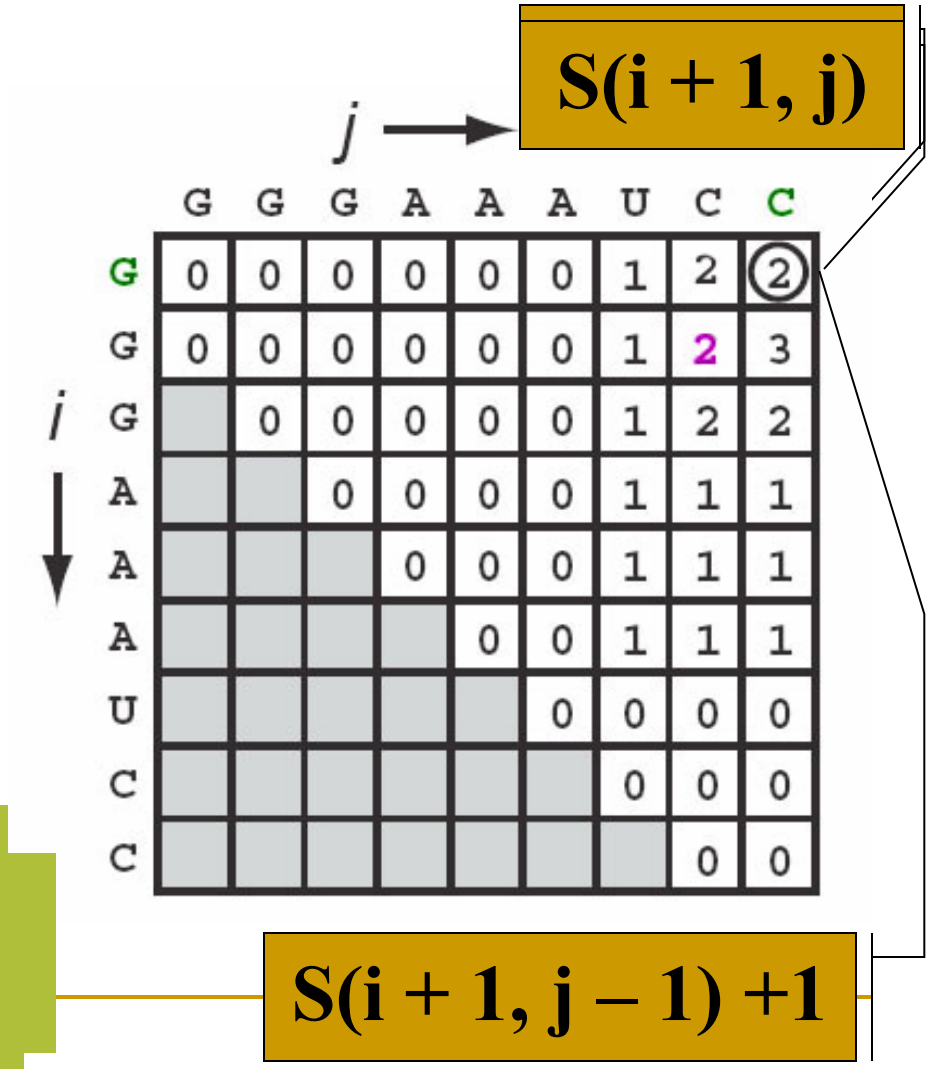
$$S(i,j) = \max \begin{cases} S(i+1,j-1)+1 & [\text{if } i,j \text{ base pair}] \\ S(i+1,j) \\ S(i,j-1) \\ \max_{i<k<j} S(i,k) + S(k+1,j) \end{cases}$$





Base Pair Maximization – Dynamic Programming Algorithm

- **Alignment Method**
  - **Align RNA strand to itself**
  - **Score increases for feasible base pairs**
- **Each score independent of overall structure**
- **Bifurcation adds extra dimension**

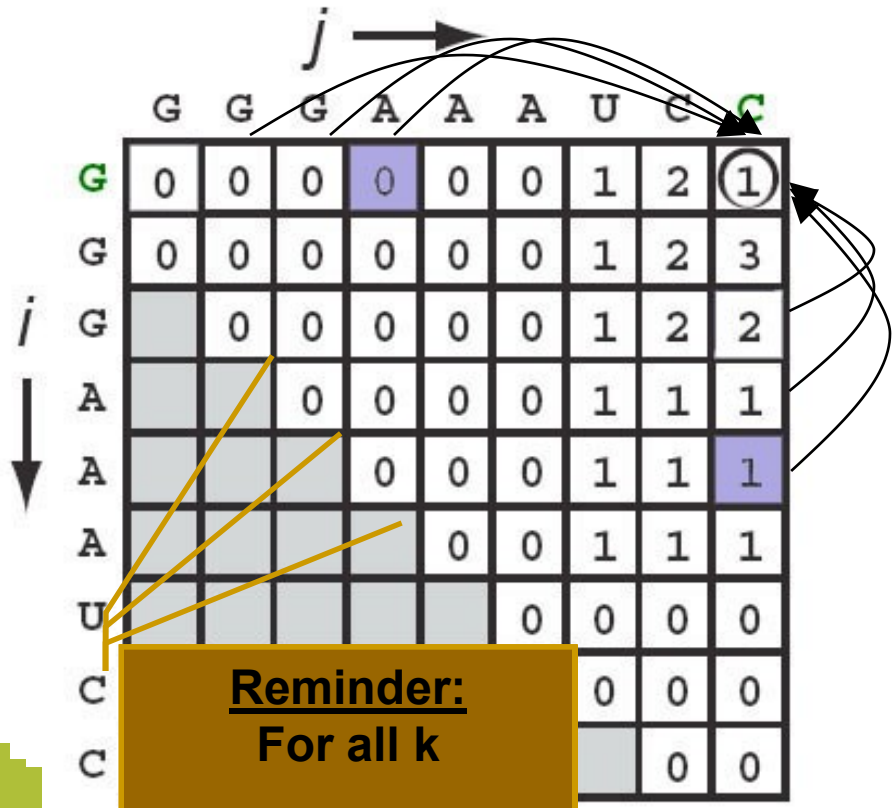


Dynamic Programming – possible paths

Images – Sean Eddy

Base Pair Maximization – Dynamic Programming Algorithm

- **Alignment Method**
  - Align RNA strand to itself
  - Score increases for feasible base pairs
- **Each score independent of overall structure**
- **Bifurcation adds extra dimension**



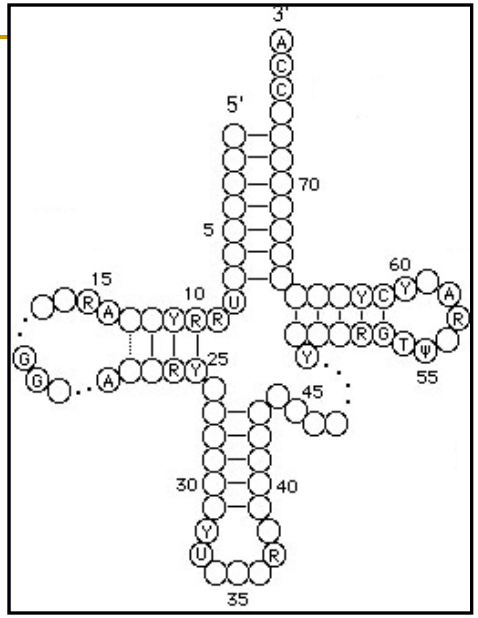
Bifurcation – add values for all k

Images – Sean Ed

Base Pair Maximization - Drawbacks

- Base pair maximization will not necessarily lead to the most stable structure
  - May create structure with many interior loops or hairpins which are energetically unfavorable
- Comparable to aligning sequences with scattered matches – not biologically reasonable

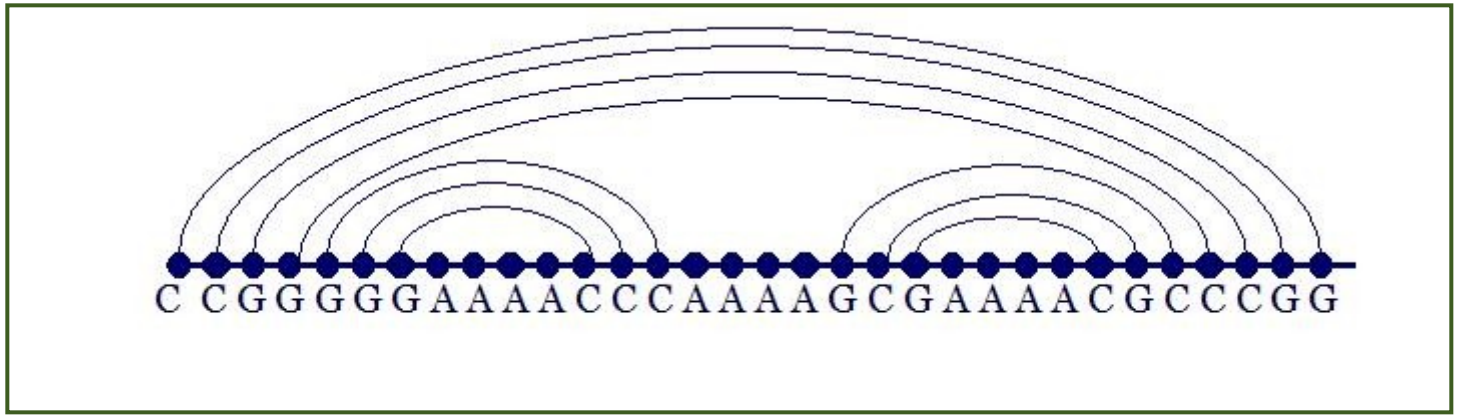
Energy Minimization



- Thermodynamic Stability
  - Estimated using experimental techniques
  - Theory : Most Stable is the Most likely
- No Pseudoknots due to algorithm limitations
- Uses Dynamic Programming alignment technique
- Attempts to maximize the score taking into account thermodynamics
- MFOLD and ViennaRNA

Free energy model

Free energy of a structure is the sum of all interactions energies

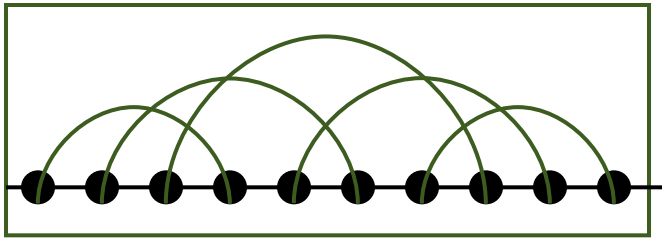


Free Energy(E) = E(CG)+E(CG)+…..

Each interaction energy can be calculated thermodynamically

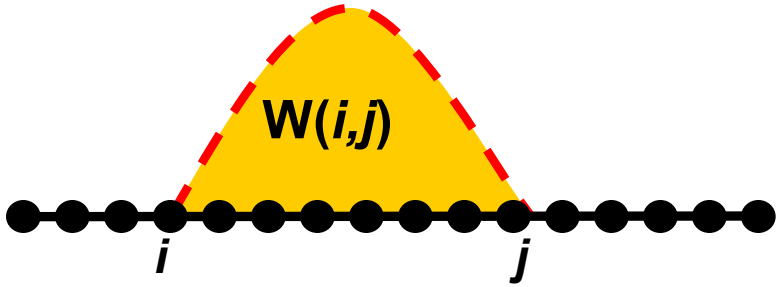
Why is MFE secondary structure prediction hard?

- MFE structure can be found by calculating free energy of all possible structures
- BUT the number of potential structures grows exponentially with the number, n, of bases



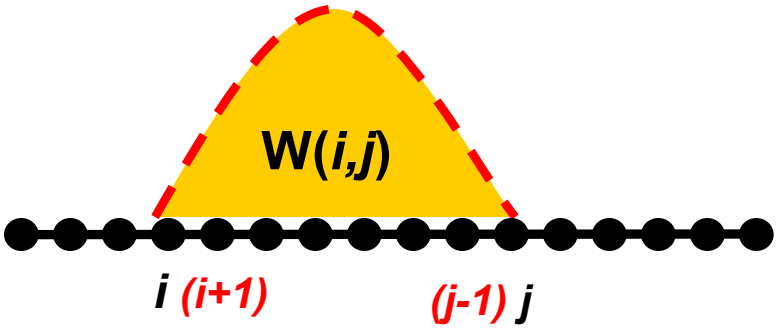
RNA folding with Dynamic programming (Zuker and Stiegler)

- **W(i,j):** MFE structure of substrand from i to j



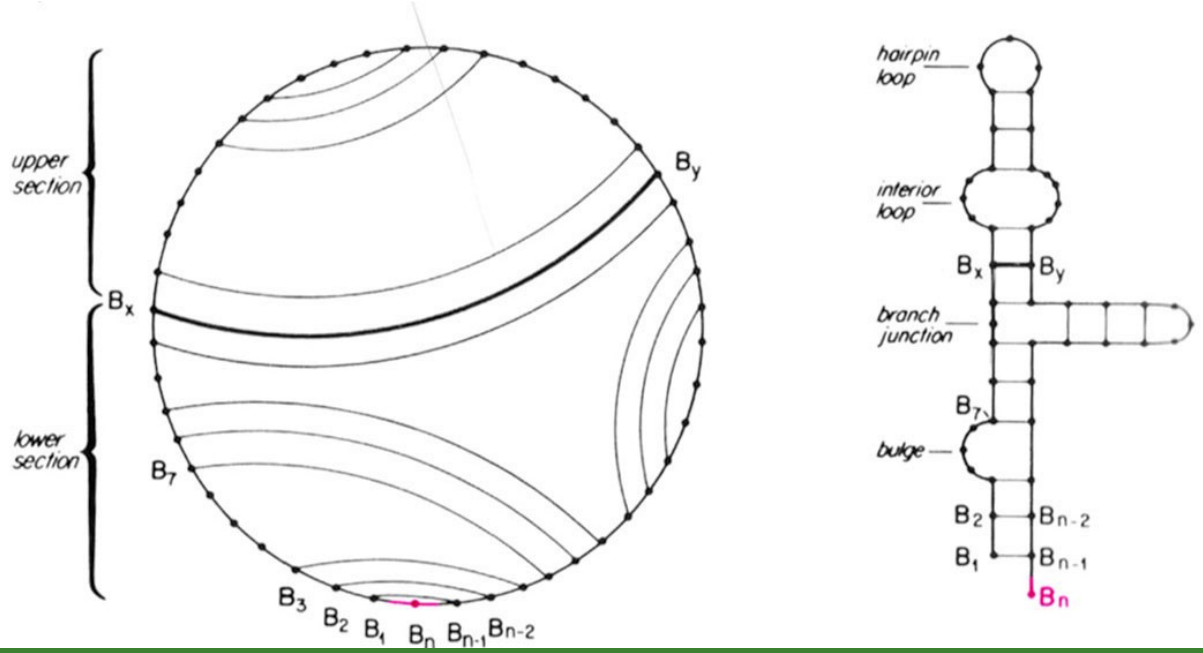
RNA folding with dynamic programming

- Assume a function W(i,j) which is the MFE for the sequence starting at i and ending at j (i<j)



- Define scores, for example base pair (CG) =-1 non-pair(CA)=1 (we want a negative score)
- Consider 4 possibilities:
  - i,j are a base pair, added to the structure for *i+1..j-1*
  - i is unpaired, added to the structure for *i+1..j*
  - j is unpaired, added to the structure for *i..j-1*
  - i,j are paired, but not to each other;
- Choose the minimal energy

Energy Minimization Results

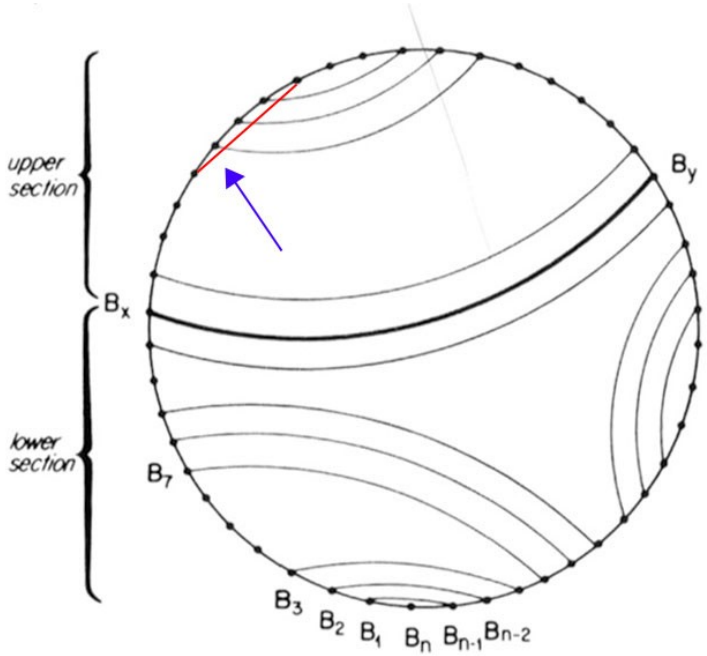


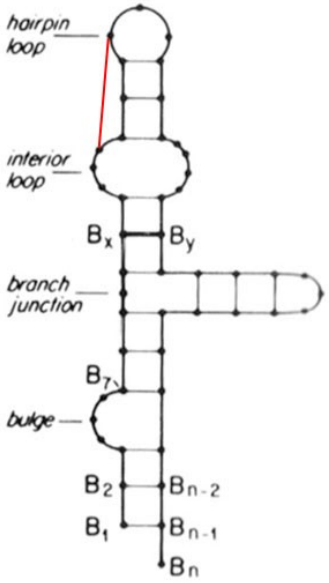
- **All loops must have at least 3 bases in them**
  - **Equivalent to having 3 base pairs between all arcs**

Exception: Location where the beginning and end of RNA come together in circularized representation

Images – David Mount

Trouble with Pseudoknots

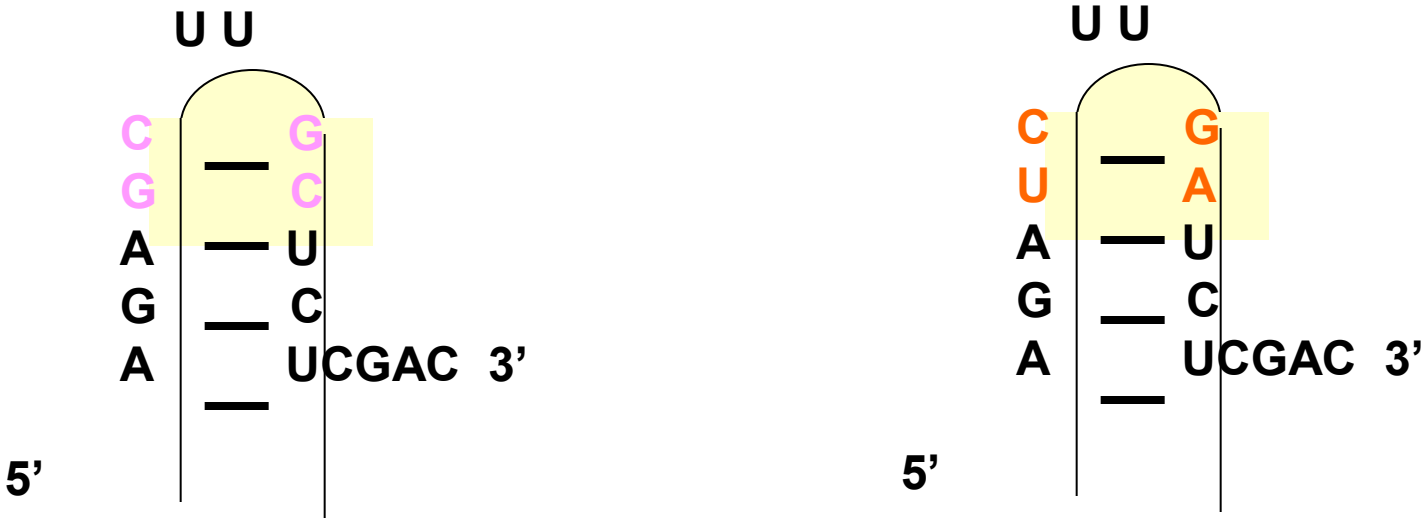




- Pseudoknots cause a breakdown in the Dynamic Programming Algorithm.
- In order to form a pseudoknot, checks must be made to ensure base is not already paired – this breaks down the recurrence relations

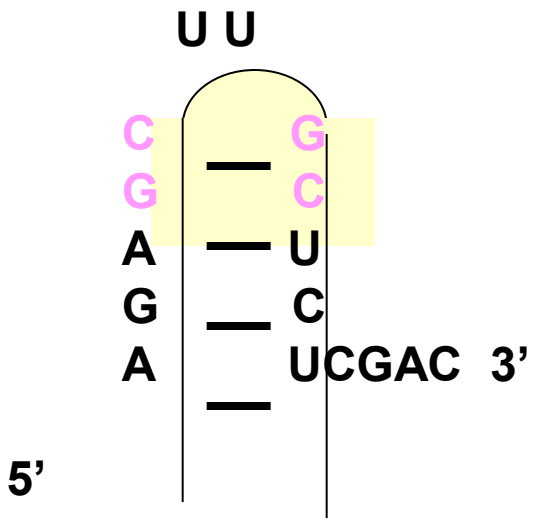
Images – David Mount

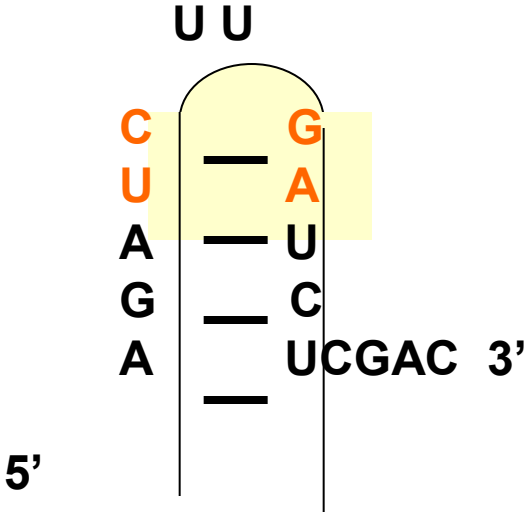
Sequence dependent free-energy
Nearest Neighbor Model



Energy is influenced by the previous base pair (not by the base pairs further down).

Sequence dependent free-energy values of the base pairs





These energies are estimated experimentally from small synthetic RNAs.

Example values:

GC	GC	GC	GC
AU	GC	CG	UA
-2.3	-2.9	-3.4	-2.1

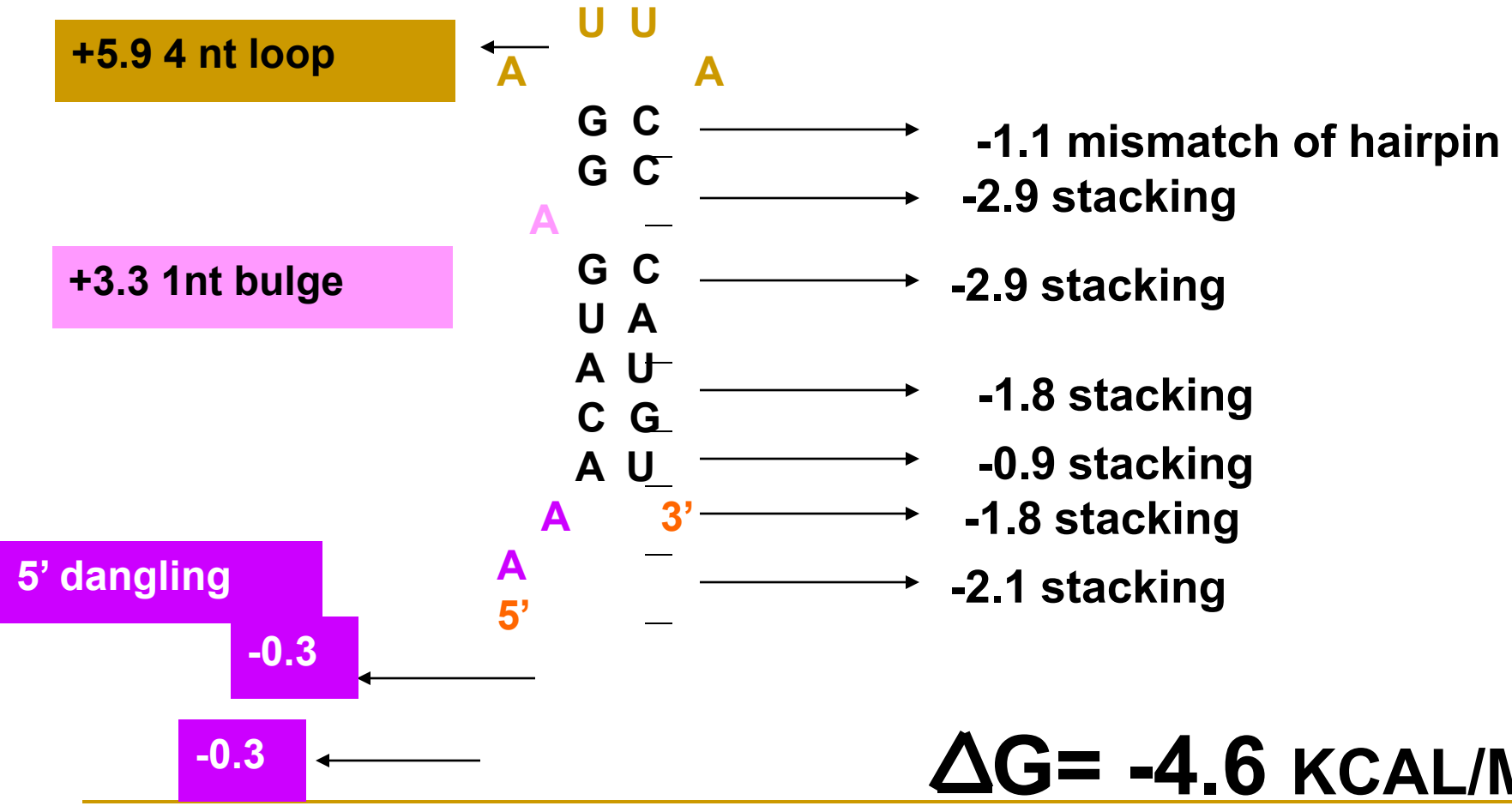
Adding Complexity to Energy Calculations

- Stacking energy - Assign negative energies to these *between base pair* regions.
  - Energy is influenced by the previous base pair (not by the base pairs further down).
  - These energies are estimated experimentally from small synthetic RNAs.
- Positive energy - added for destabilizing regions such as bulges, loops, etc.
- More than one structure can be predicted

Mfold

- Positive energy - added for destabilizing regions such as bulges, loops, etc.
- More than one structure can be predicted

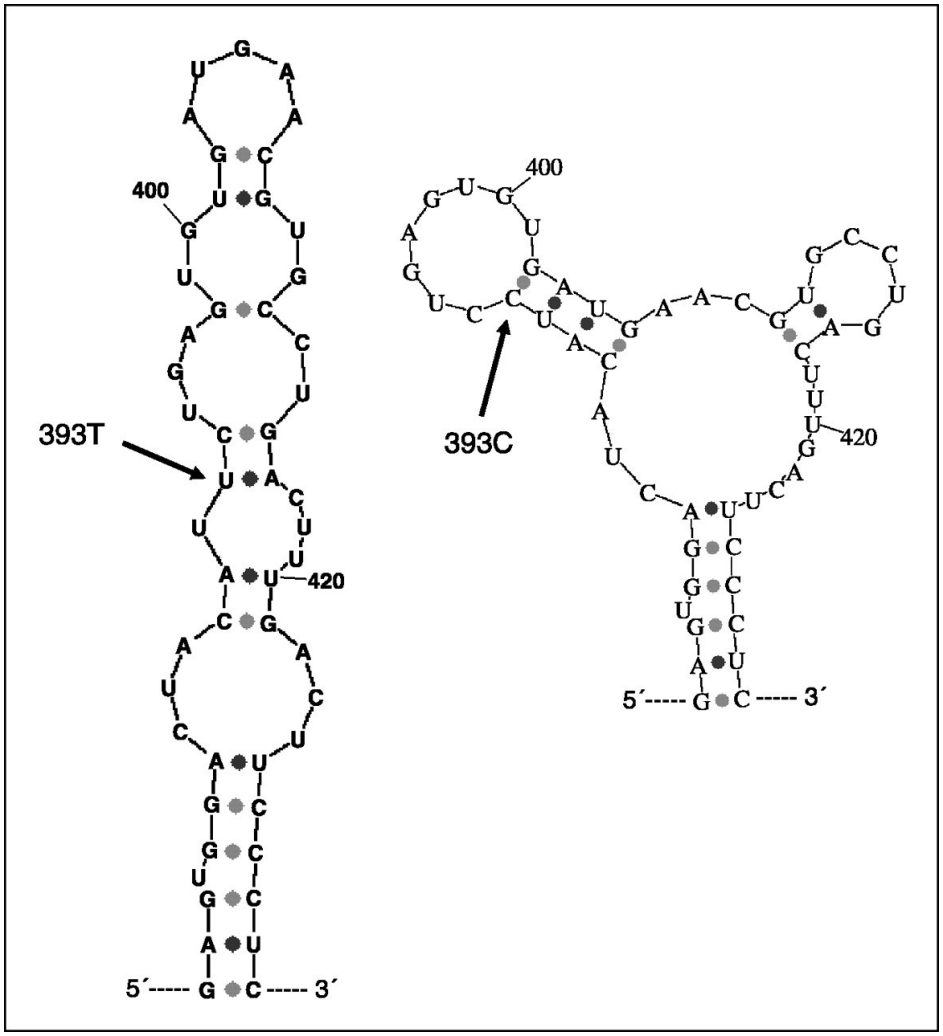
Free energy computation



Mfold

- Positive energy - added for destabilizing regions such as bulges, loops, etc.
- More than one structure can be predicted

More than one structure can be predicted for the same RNA



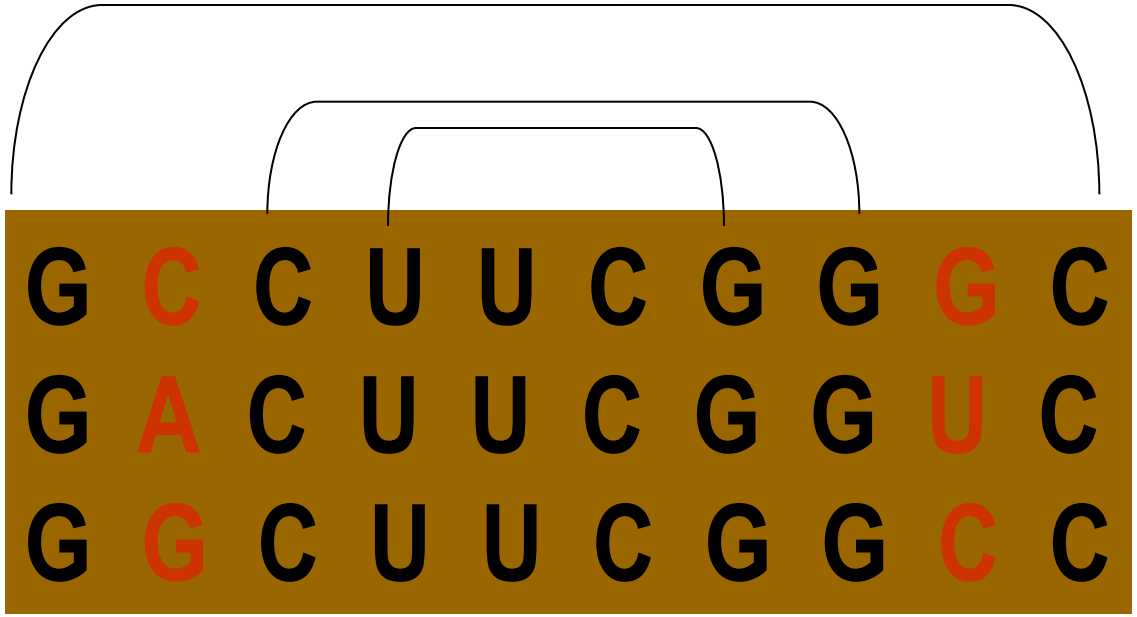
Frey U H et al. Clin Cancer Res 2005;11:5071-5077

Energy Minimization Drawbacks

- Compute only one optimal structure
- Usual drawbacks of purely mathematical approaches
  - Similar difficulties in other algorithms
    - Protein structure
    - Exon finding

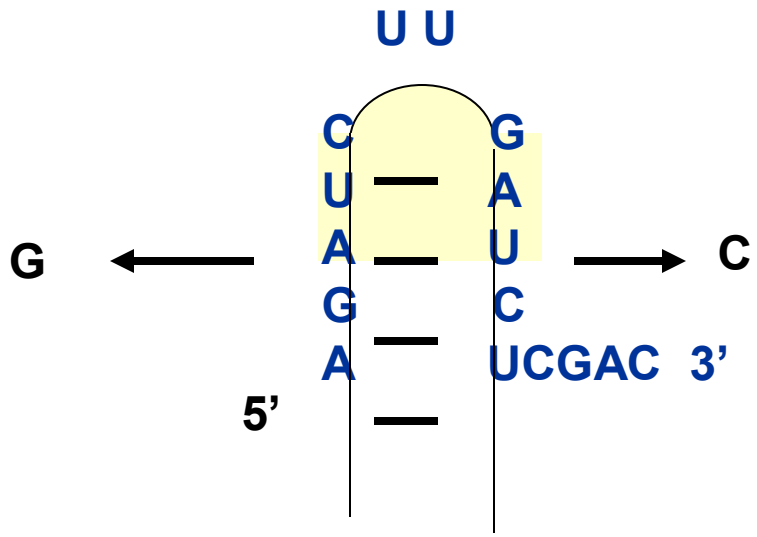
RNA fold prediction based on Multiple Alignment

Information from multiple sequence alignment (MSA) can help to predict the probability of positions i,j to be base-paired.

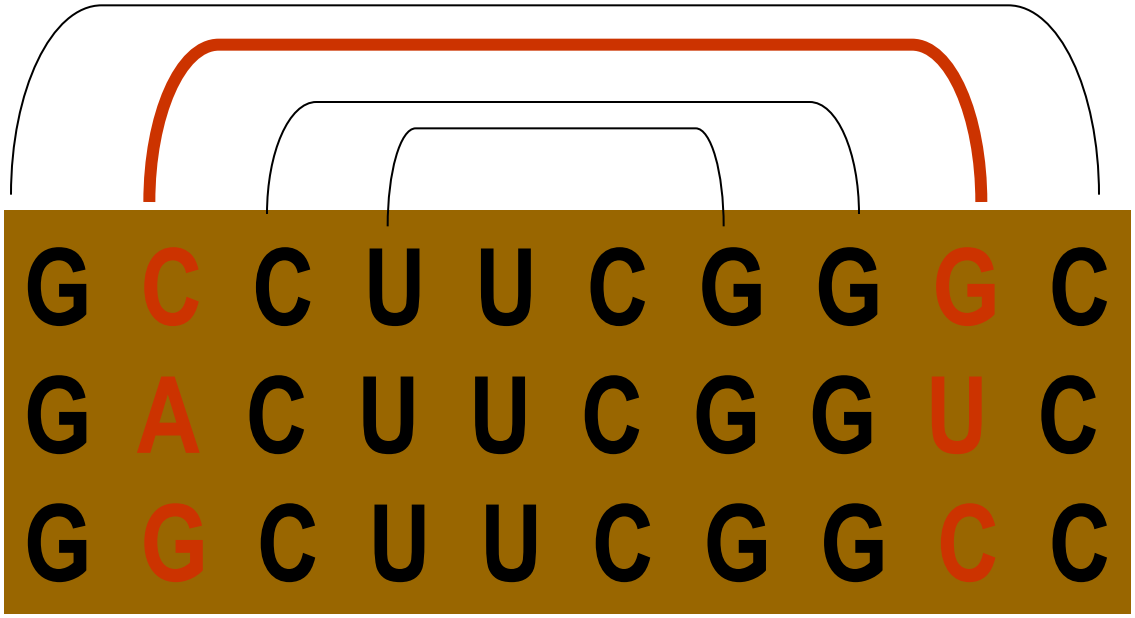


Compensatory Substitutions

Mutations that maintain the secondary structure can help predict the fold



RNA secondary structure can be revealed by identification of compensatory mutations



U C
U G
C G
N N'
G C

Insight from Multiple Alignment

Information from multiple sequence alignment (MSA) can help to predict the probability of positions i,j to be base-paired.

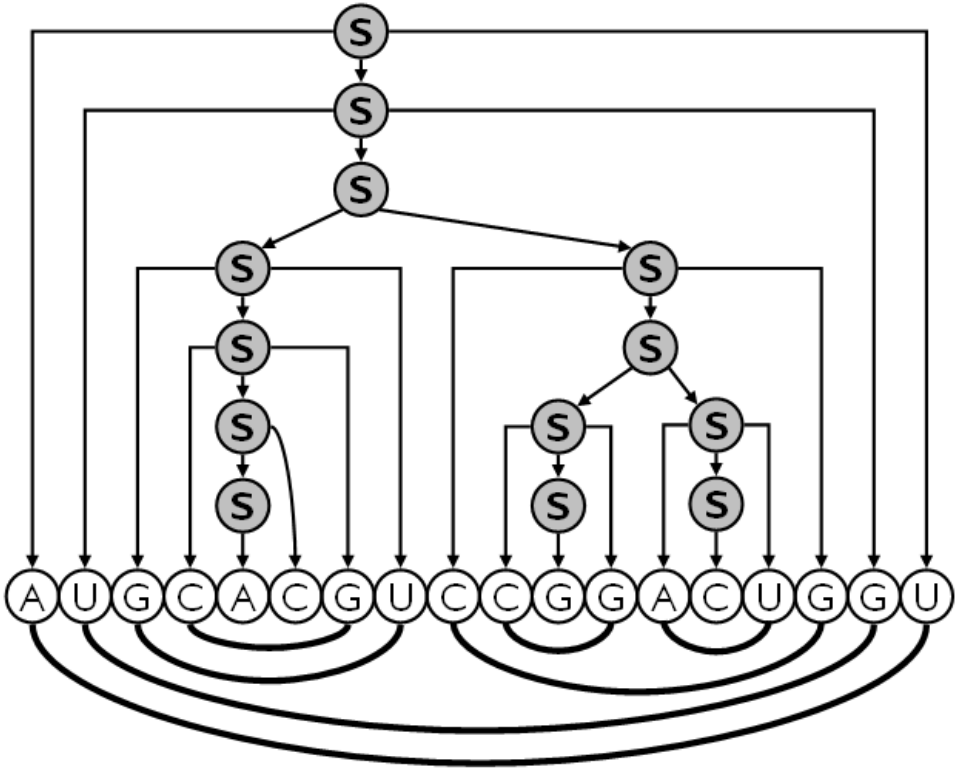
- Conservation – no additional information
- Consistent mutations (GC→ GU) – support stem
- Inconsistent mutations – does not support stem.
- Compensatory mutations – support stem.

RNAalifold

- Predicts the consensus secondary structure for a set of aligned RNA sequences by using modified dynamic programming algorithm that add alignment information to the standard energy model
- Improvement in prediction accuracy

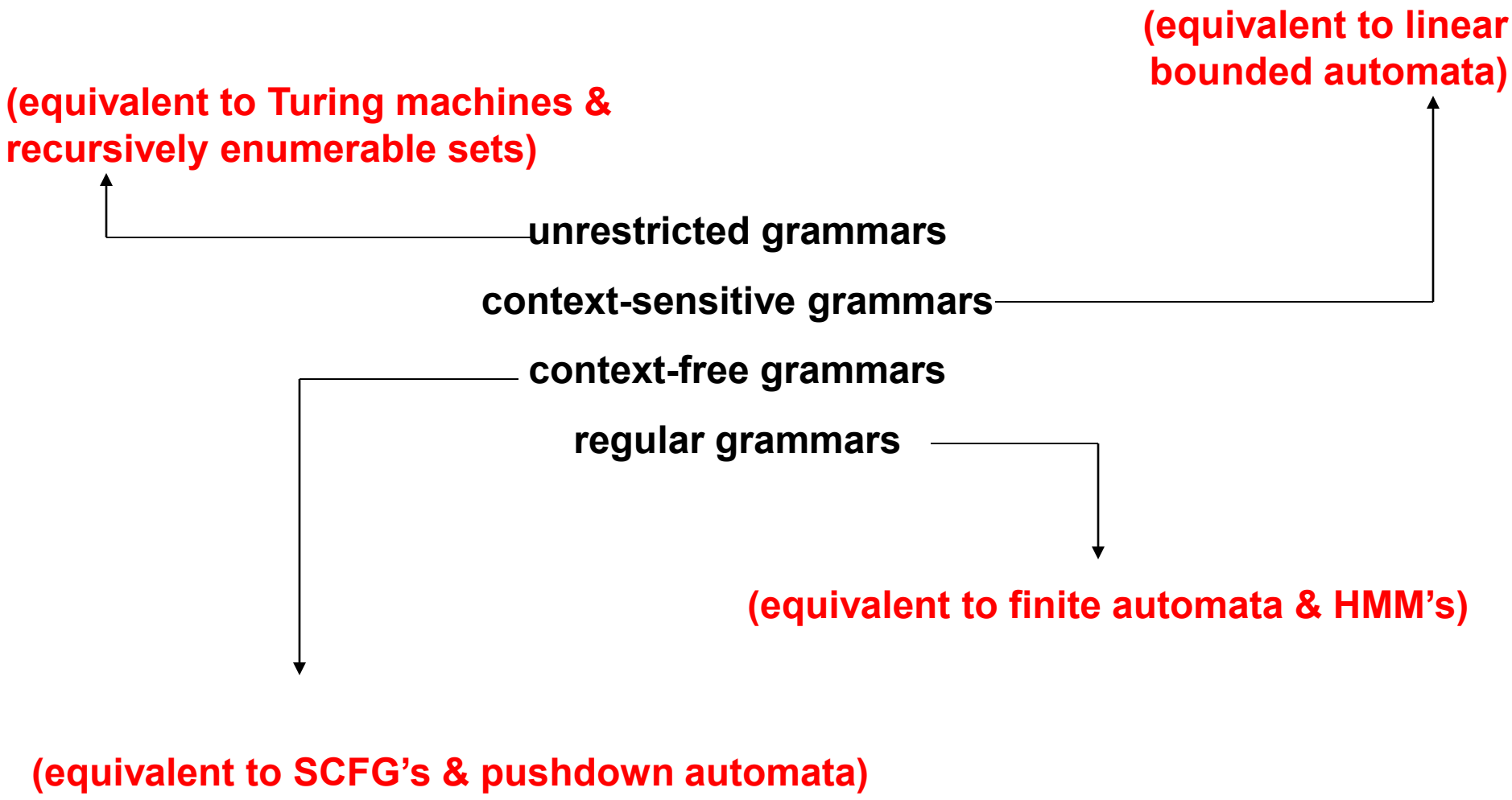
STOCHASTIC CONTEXT-FREE GRAMMARS

SCFG



- RNA folding can be represented as context-free grammars

Chomsky hierarchy



Context-free grammars

A *context-free grammar* is a generative model denoted by a 4-tuple:

$$G = (V, \alpha, S, R)$$

where:

α **is a *terminal alphabet*, *(e.g., {a, c, g, u})***
V **is a *nonterminal alphabet*, *(e.g., {A, B, C, D, E, ...})***
$S \in V$ **is a special *start symbol*, and**
R **is a set of rewriting rules called *productions*.**

Productions in R are rules of the form:

$$X \rightarrow \lambda$$

where $X \in V$, $\lambda \in (V \cup \alpha)^*$

Context "freeness"

The "***context-freeness***" is imposed by the requirement that the l.h.s of each production rule may contain only a **single** symbol, and that symbol must be a **nonterminal:**

$$X \rightarrow \lambda$$

Thus, a CFG **cannot** specify ***context-sensitive*** rules such as:

$$wXz \rightarrow w\lambda z$$

Derivations

Suppose a CFG G has generated a *terminal string* $x \in \alpha^*$. A *derivation* $\mathbf{S} \Rightarrow^* \mathbf{x}$ denotes a possible derivation for generating x.

A *derivation* (or *parse*) consists of a series of applications of productions from R, beginning with the *start symbol* S and ending with the *terminal string* x:

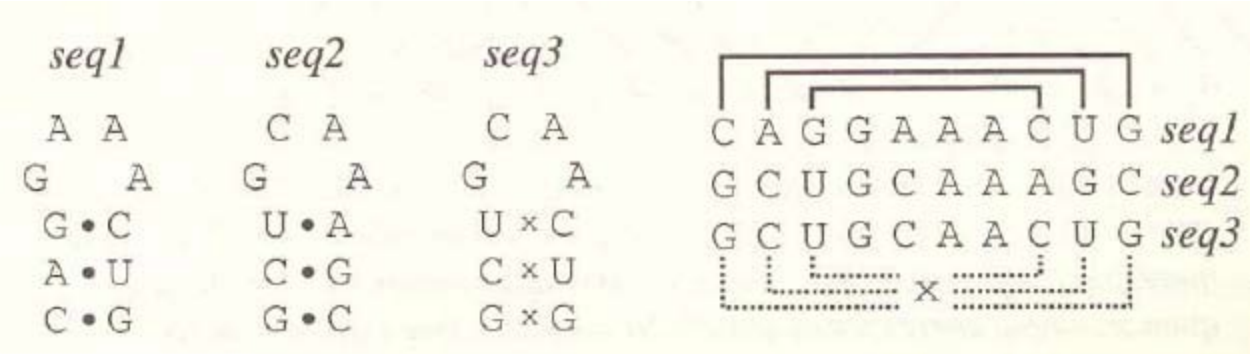
$$S \Rightarrow s_1 \Rightarrow s_2 \Rightarrow s_3 \Rightarrow \cdots \Rightarrow x$$

where $s_i \in (V \cup \alpha)^*$.

We'll concentrate of leftmost derivations where the leftmost nonterminal is always replaced first.

A CFG for an RNA

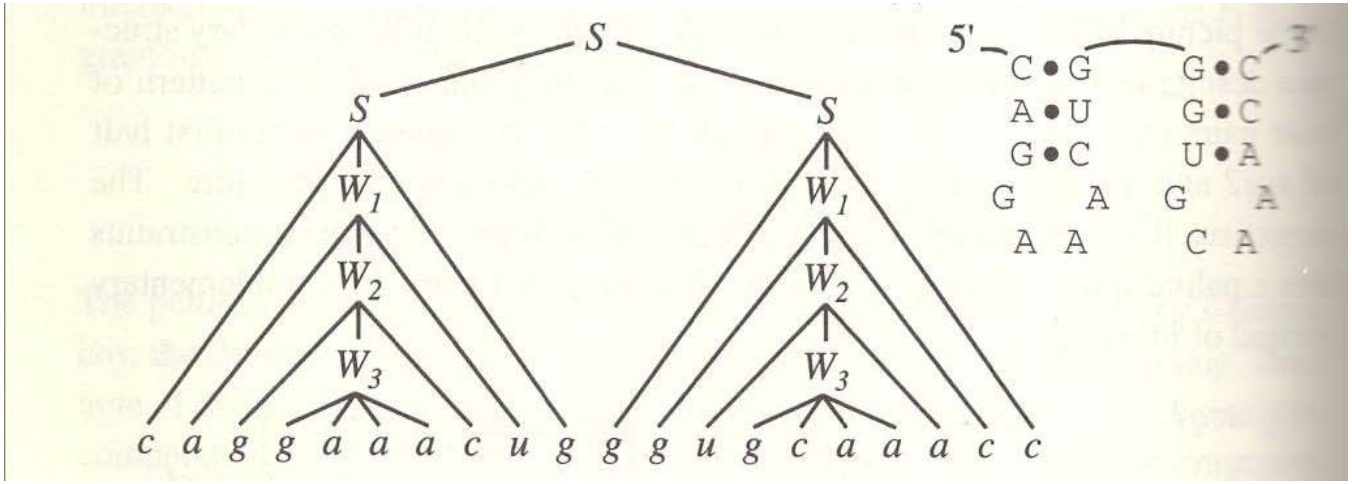
- RNA hairpin with 3 bp stem and a 4-base loop (GAAA or GCAA)



S-> aXu | cXg | gXc | uXa
X-> aYu | cYg | gYc | uYa
Y-> aZu | cZg | gZc | uZa
Z->gaaa | gcaa

Parse trees

- A representation of a parse of a string by a CFG
- **Root –** start nonterminal S
- **Leaves –** terminal symbols in the given string
- **Internal nodes -** nonterminals
- The children of an internal node are the productions of that nonterminal (left-to-right order



Stochastic CFG

A *stochastic context-free grammar* (*SCFG*) is a CFG plus a probability distribution on productions:

$$G = (V, \alpha, S, R, P_p)$$

where $P_p : R$ a ¡, and probabilities are normalized at the level of each l.h.s. symbol X:

$$\forall \left[\sum_{X \in V \quad X \to \lambda} P_p(X \to \lambda) = 1 \right]$$

Thus, we can compute the probability of a single derivation $S \Rightarrow^* x$ by multiplying the probabilities for all productions used in the derivation:

$$\prod_i P(X_i \to \lambda_i)$$

We can sum over all possible (leftmost) derivations of a given string x to get the probability that G will generate x at random:

$$P(x \mid G) = \sum_j P(S \Rightarrow_j^* x \mid G).$$

An example

As an example, consider G=(V_G, α, S, R_G, P_G), for V_G={S, L, N}, α={a,c,g,t}, and R_G the set consisting of:

$$S \rightarrow \mathtt{a}\,S\,\mathtt{u} \mid \mathtt{u}\,S\,\mathtt{a} \mid \mathtt{c}\,S\,\mathtt{g} \mid \mathtt{g}\,S\,\mathtt{c} \mid L \qquad \textbf{(P=0.2)}$$

$$L \rightarrow N\,N\,N\,N \qquad \textbf{(P=1.0)}$$

$$N \rightarrow \mathtt{a} \mid \mathtt{c} \mid \mathtt{g} \mid \mathtt{u} \qquad \textbf{(P=0.25)}$$

Then the probability of the sequence acguacguacgu is given by:

$$P(\mathtt{acguacguacgu}) =$$

$$P(\, S \Rightarrow \mathtt{a}S\mathtt{u} \Rightarrow \mathtt{ac}S\mathtt{gu} \Rightarrow \mathtt{acg}S\mathtt{cgu} \Rightarrow \mathtt{acgu}S\mathtt{acgu} \Rightarrow$$
$$\mathtt{acgu}L\mathtt{acgu} \Rightarrow \mathtt{acgu}NNNN\mathtt{acgu} \Rightarrow \mathtt{acgua}NNN\mathtt{acgu} \Rightarrow$$
$$\mathtt{acguac}NN\mathtt{acgu} \Rightarrow \mathtt{acguacg}N\mathtt{acgu} \Rightarrow \mathtt{acguacguacgu}) =$$

$$0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 1 \quad 0.25 \quad 0.25 \quad 0.25 \quad 0.25 = 1.25\ 10^{-6}$$

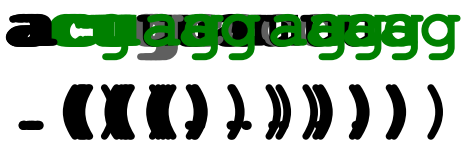
because this sequence has only one possible (leftmost) derivation under grammar G.

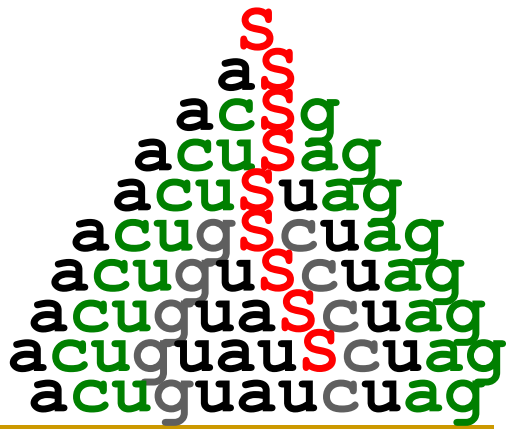
Structure using SFCG

- Grammar rules with associated probabilities

S →	aSu	cSg	aS	uS	…	Su	SS	ε
P	.21	.15	.11	.08		.03	.22	.02

- We select the set of transformations that highest probability of generating the input sequence. This set gives us our structure.
- Let's generate a structure for the sequence acuguaucuag





Chomsky Normal Form

A CNF grammar is one in which all productions are of the form:

$X \rightarrow Y\,Z$

or:

$X \rightarrow \mathtt{a}$

Non-CNF:

$S \rightarrow \mathtt{a}\,S\,\mathtt{t} \mid \mathtt{t}\,S\,\mathtt{a} \mid \mathtt{c}\,S\,\mathtt{g} \mid \mathtt{g}\,S\,\mathtt{c} \mid L$

$L \rightarrow N\,N\,N\,N$

$N \rightarrow \mathtt{a} \mid \mathtt{c} \mid \mathtt{g} \mid \mathtt{u}$

CNF:

$S \rightarrow A\,S_T \mid T\,S_A \mid C\,S_G \mid G\,S_C \mid N\,L_1$

$S_A \rightarrow S\,A$

$S_T \rightarrow S\,T$

$S_C \rightarrow S\,C$

$S_G \rightarrow S\,G$

$L_1 \rightarrow N\,L_2$

$L_2 \rightarrow N\,N$

$N \rightarrow \mathtt{a} \mid \mathtt{c} \mid \mathtt{g} \mid \mathtt{u}$

$A \rightarrow \mathtt{a}$

$C \rightarrow \mathtt{c}$

$G \rightarrow \mathtt{g}$

$T \rightarrow \mathtt{u}$

Parsing CFG

Two questions for a CFG:

1) Can a grammar *G* derive string *x*?
2) If so, what series of productions would be used during the derivation? *(there may be multiple answers!)*

Additional questions for an SCFG:

1) What is the *probability* that *G* derives string *x*?
2) What is the *most probable* derivation of *x* via *G*?

Parsing CFG

- CYK Algorithm (Cocke-Younger-Kasami)
  - Dynamic Programming method
- Modified CYK for SCFG
  - “Inside algorithm”
  - Training similar to HMM
    - If parses are known for training data sequences, simply count the number of times for each production, calculate probabilities (labeled sequence training for HMM)
    - If parses are not known, apply an EM algorithm called “Inside-Outside” (“forward-backward” for HMM)