# RNA
# secondary structure
# prediction and analysis

# Resources

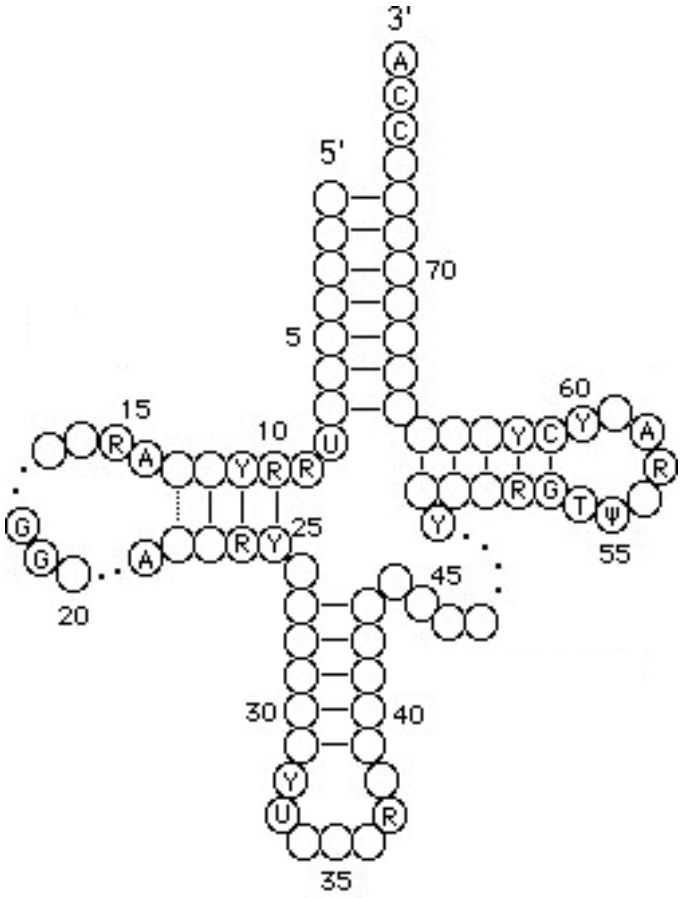
- Lecture Notes from previous years: Takis Benos
- Covariance algorithm: Eddy and Durbin, Nucleic Acids Research, v22: 11, 2079
- Useful lecture slides from  Larry Ruzzo, U of Washington and Phillip Compeau, UCSD

# Outline

- RNA Folding
- Dynamic Programming for RNA secondary structure prediction
  - Nussinov et al and Zucker et al algorithms
- Covariance Model
  - Eddy and Durbin

# Various types of RNA

- messenger RNA (mRNA)
- transfer RNA (tRNA)
- Ribosomal RNA (rRNA)
- small interfering RNA (siRNA)
- micro RNA (miRNA)
- small nuclear RNA (snRNA)
- small nucleolar RNA (snoRNA)
- guide RNA (gRNA)
- efference RNA(eRNA)



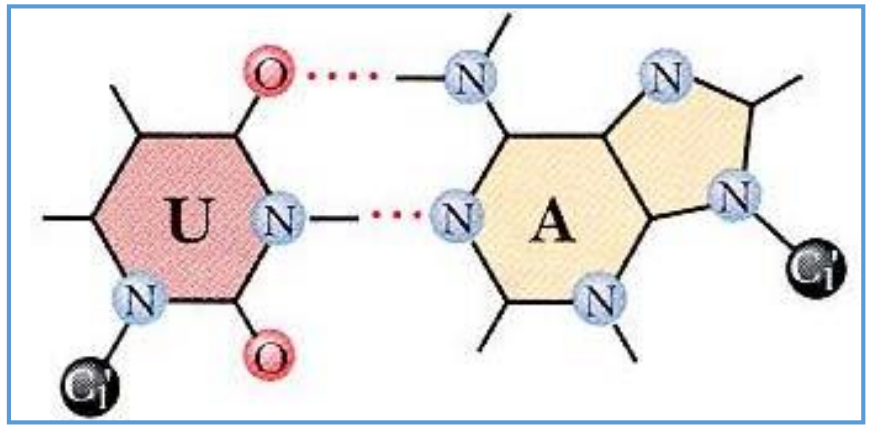
# Non coding RNA (ncRNA)

- RNA that isn’t translated into protein
- Includes:
  - tRNA, rRNA, snRNA, snoRNA, miRNA, gRNA, eRNA, pRNA, tmRNA
- What about mRNA?
  - 5’ methylated cap
  - 5’-UTR (un-translated regions)
  - CDS (coding sequence)
  - 3’-UTR
  - Poly-A tail
- mRNA contrains untranslated regions (5’UTR, 3’UTR), but UTRs are not considered ncRNA

# RNA Basics

- RNA bases: A, C, G, U
- Watson-Crick Pair
  - A-U (~ 2 kcal/mol)
  - G-C (~ 3 kcal/mol)
- Wobble pair
  - G-U (~ 1 kcal/mol)
- Non-Canonical pairs (modified suitably)
- Bases can only pair with one other base

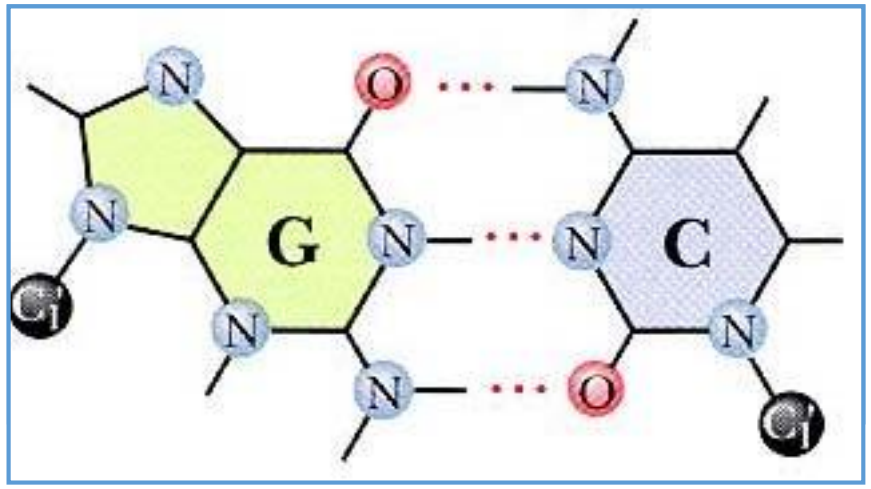
Two hydrogen bonds



# RNA Basics

- RNA bases: A, C, G, U
- Canonical Base Pairs
  - A-U
  - G-C
  - G-U
- Bases can only pair with one other base

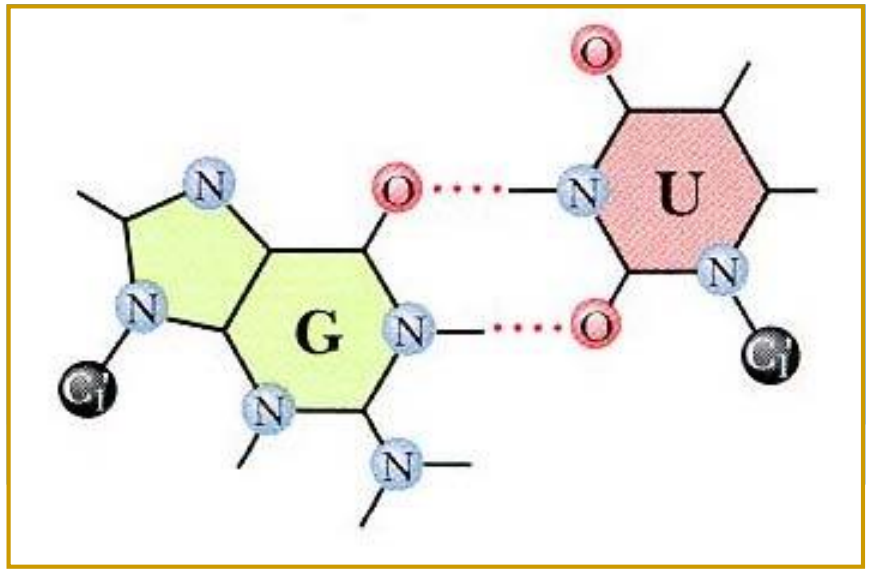
Three hydrogen bonds => more stable



# RNA Basics

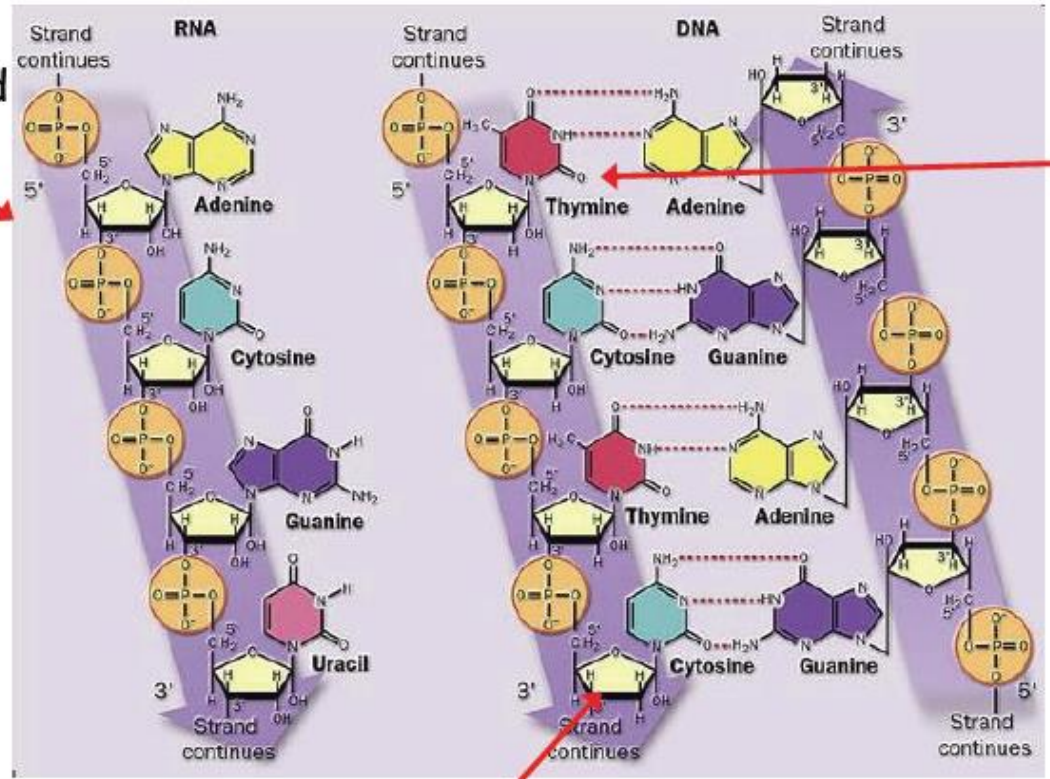
- RNA bases: A, C, G, U
- Canonical Base Pairs
  - A-U
  - G-C
  - G-U
- Bases can only pair with one other base

‘wobble pairing’



# How is RNA different from DNA?

Single-stranded



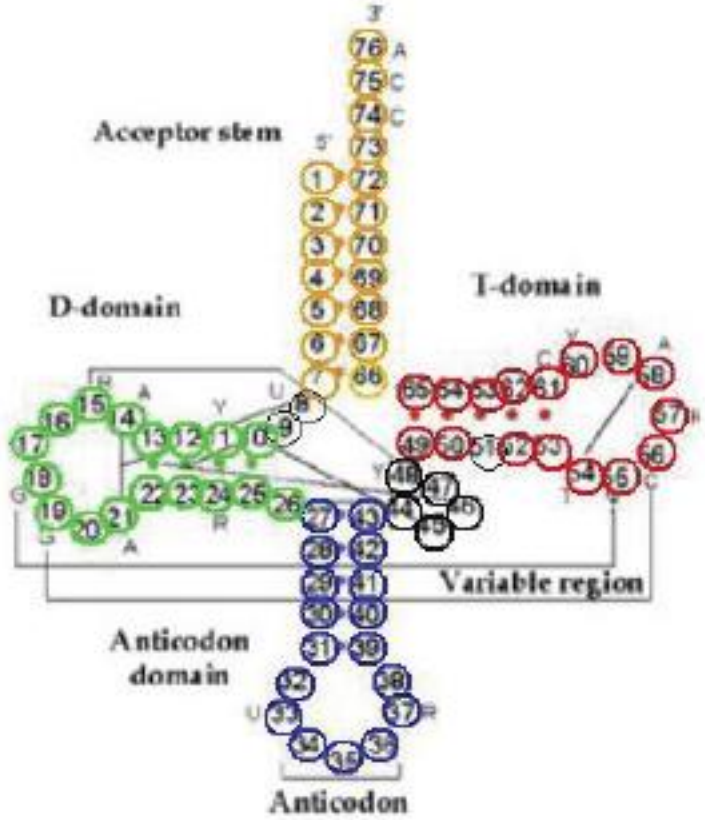

Uracil instead of Thymine

Sugar is Ribose instead of Deoxyribose

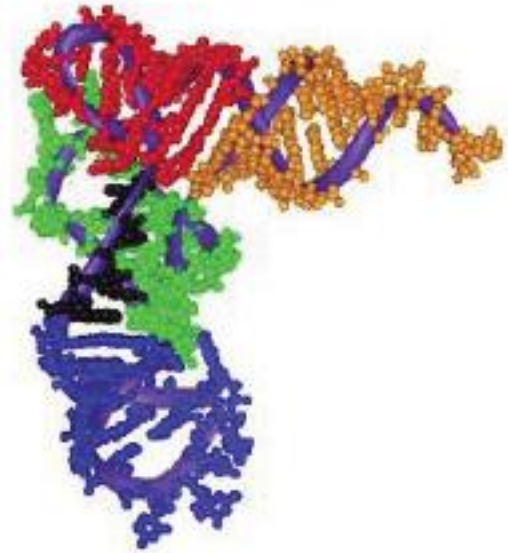
# Secondary and Tertiary Structure

GCGGAUUUAGCUCAGUUGG
GAGAGCGCCAGACUGAAGA
UCUGGAGGUCCUGUGUUCG
AUCCACAGAAUUCGCACCA

**Primary Structure**

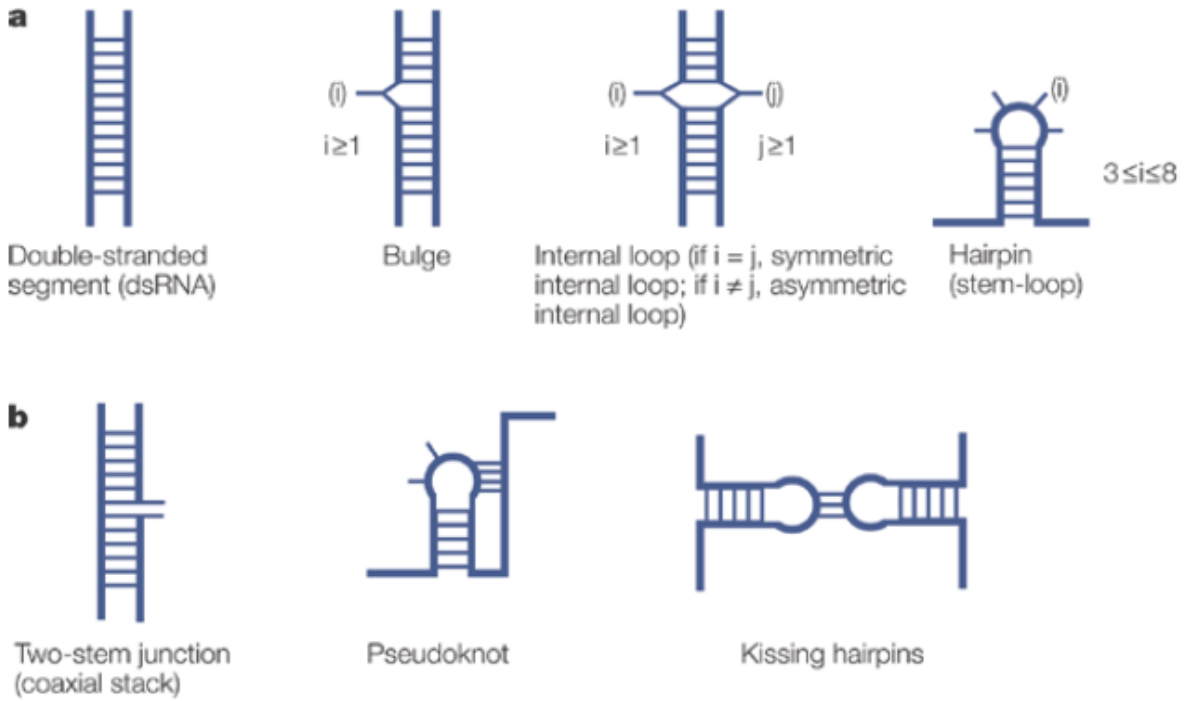



**Secondary Structure**



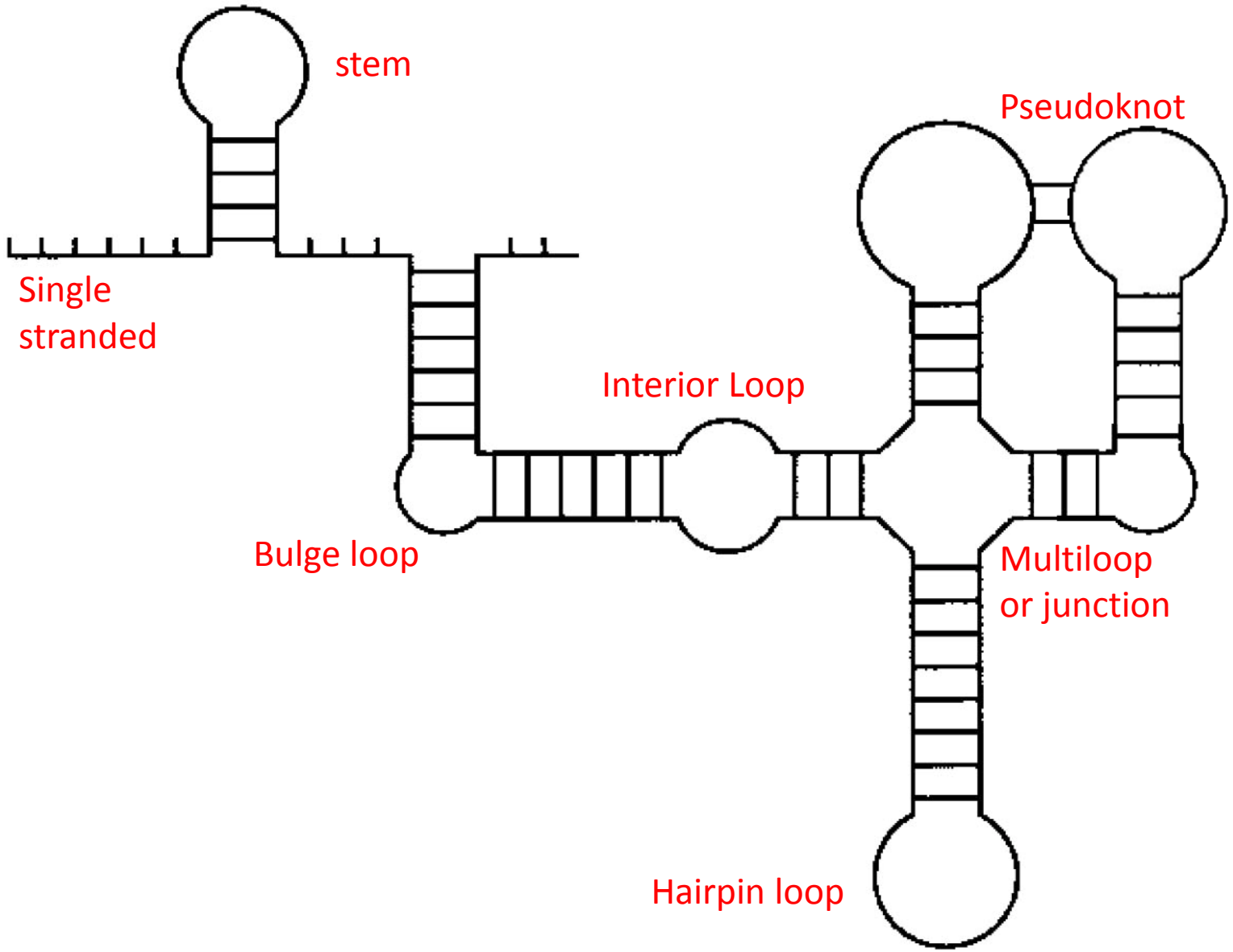
**Tertiary Structure**

# Secondary structural elements

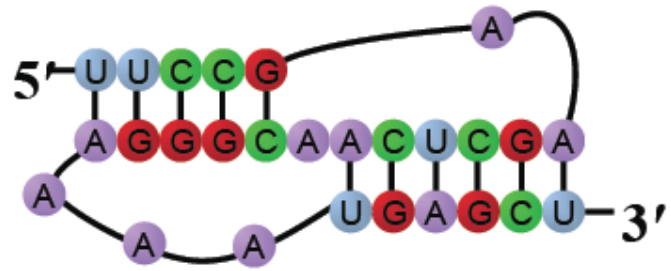



Nature Reviews | Molecular Cell Biology

# RNA Secondary Structure/Motifs



# Pseudoknots




bases pairs between a loop and positions outside the enclosing stem

Challenging to deal with them; however, the total number of pseudoknotted base pairs is relatively small

i.e. in E. coli SSU rRNA, 447 base pairs, only 8 are in pseudoknot structures

# tRNA - Alt. Representations

Anticodon loop

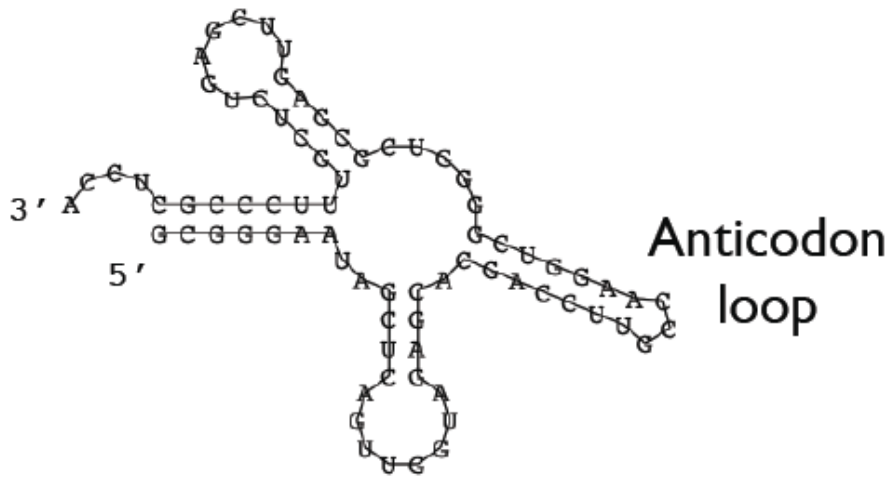
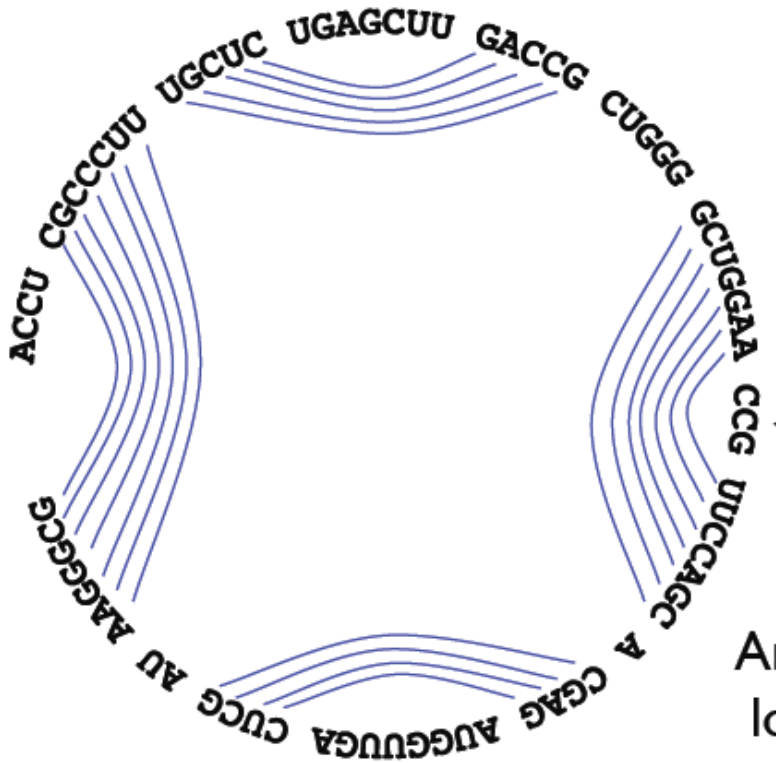


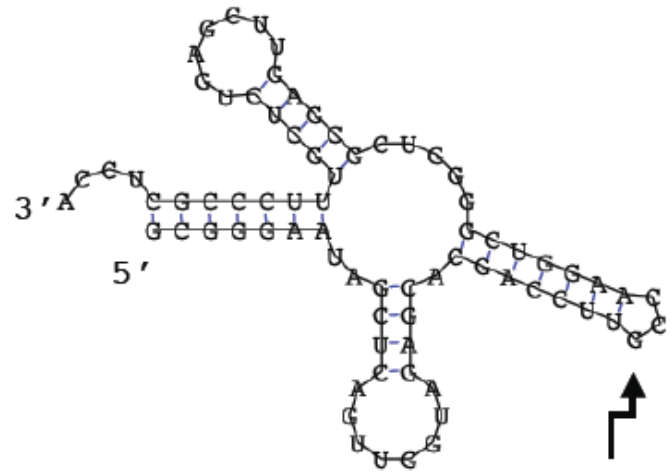

**Figure 1: a)** The spatial structure of the phenylalanine tRNA form yeast

b) The secondary structure extracts the most important information about the structure, namely the pattern of base pairings.

# tRNA - Alt. Representations




Anticodon loop




Anticodon loop

# RNA importance

- Ribozymes (RNA enzymes)
- Retroviruses
- Effects on transcription, translation, splicing..
- Functional RNAs: rRNA, tRNA, snRNA, snoRNA, microRNA, RNAi, riboswitches, regulatory elements in 3’ and 5’ UTR

# RNA motifs

- Function depends on sequence and secondary structure
- Functions include
  - Protein binding
  - Basepairing to another RNA
  - Modifying a nucleic acid bond
- Types:
  - Single-strand regions
  - Helices (or stems)
  - Bulges
  - Hairpin loops
  - Internal loops
  - Junctions

# RNA Motifs Regulatory Effects

- Regulations of translations
- Processing of RNA
- Catalytic modification of other RNAs
- Transport and position in the cell
- Stability of RNA-transcript
- Expression of encoded proteins

# Why predict structures?

- Knowing the shape of a biomolecule is invaluable in drug design and understanding disease mechanisms
- Current physical methods (X-Ray, NMR) are too expensive and time-consuming
- Predict shape from sequence of bases
- Four basic structures: helices, loops, bulges and junctions

# RNA secondary structure

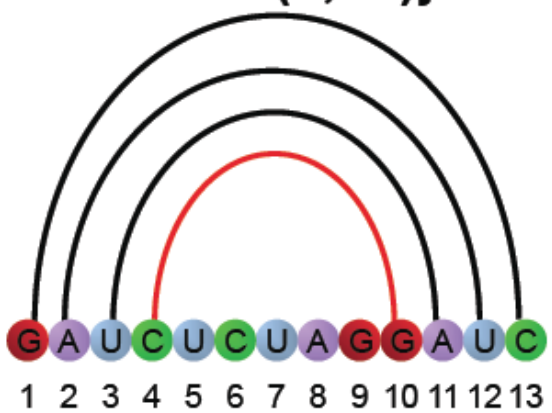
- What makes RNA fold?
- Problem: given an RNA sequence, find the set of base pairs that is "correct" or "optimal"
  - Maximize number of base pairs (Nussinov et al)
  - Minimize energy (Zucker et al)
- Search problem: very high number of possible structures
- Algorithm: dynamic programming
  - Cannot handle pseudoknots

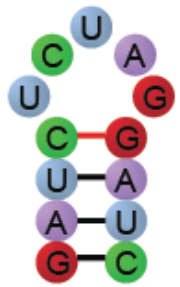
# Structure Representation

- 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)}**




# Definitions

- Sequence $^{5'}\ r_1\ r_2\ r_3 \ldots r_n\ ^{3'}$ in {A, C, G, T}
- A Secondary Structure is a set of pairs i•j s.t.

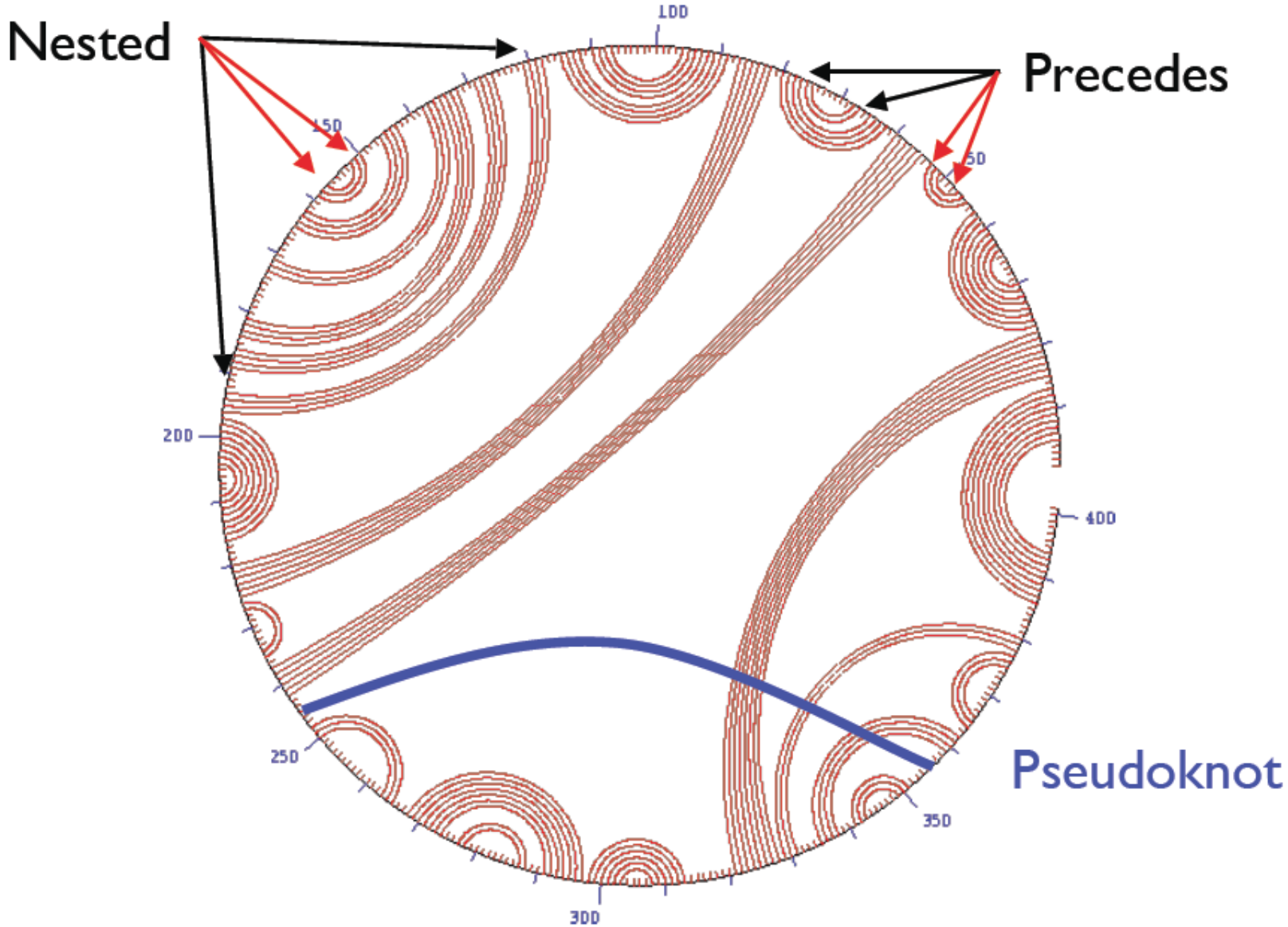
1. $i < j-4$
2. if i•j & i'•j' are two pairs with $i \leq i'$, then

   A. $i = i'$ & $j = j'$, or

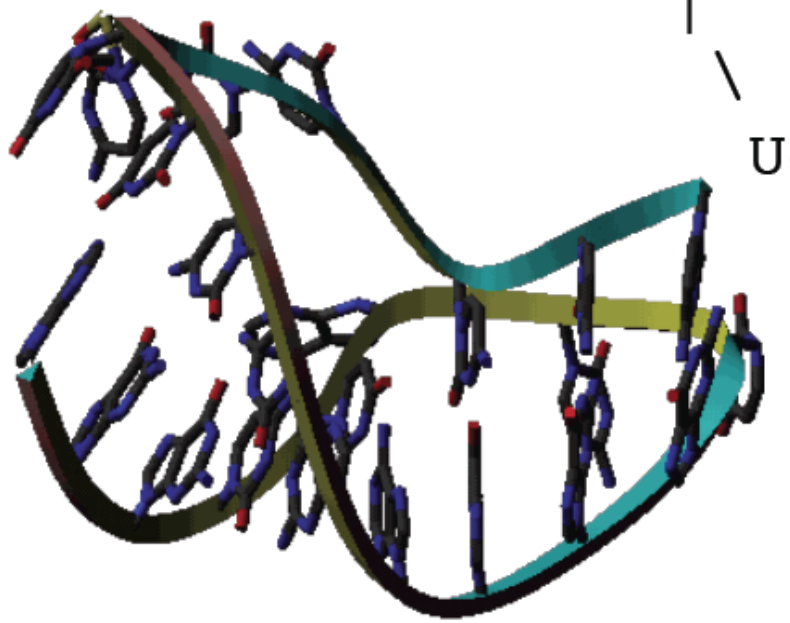
   B. $j < i'$, or

   C. $i < i' < j' < j$

} First pair precedes 2nd, or is nested within it. No "pseudoknots."


Nested
Precedes
Pseudoknot
100
150
50
200
400
250
350
300

# A Pseudoknot



```
                    A–C
                   /   \
3′ – A–G–G–C–U      U
     U–C–C–G–A–G–G–G
     |           C–C–C – 5′
      \         /
       U–C–U–C
```

# Approaches to Structure Prediction

- Maximum Pairing
  + works on single sequences
  + simple
  - too inaccurate
- Minimum Energy
  + works on single sequences
  - ignores pseudoknots
  - only finds "optimal" fold
- Partition Function
  + finds all folds
  - ignores pseudoknots

# Approaches, II

- Comparative sequence analysis
  + handles all pairings (incl. pseudoknots)
  - requires several (many?) aligned, appropriately diverged sequences
- Stochastic Context-free Grammars
  Roughly combines min energy & comparative, but no pseudoknots
- Physical experiments (x-ray crystalography, NMR)

# Base Pair Maximization: Dynamic Programming

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

- Recurrence:

Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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

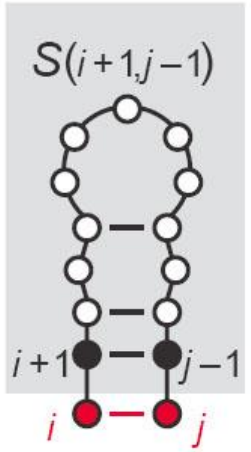
- Recurrence:

$$S(i, j) = \max \begin{cases} \\ \\ \\ \\ \end{cases}$$

Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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




Base pair at $i$ and $j$

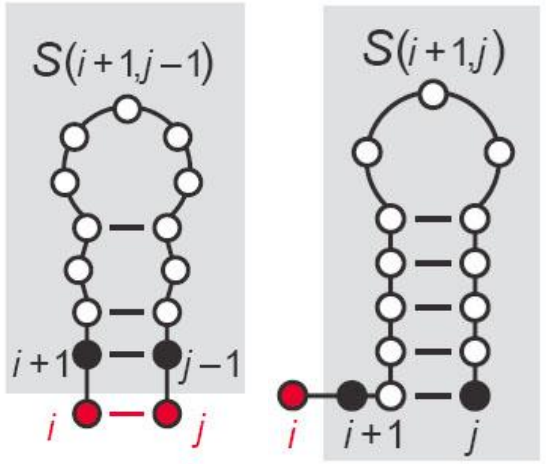
- Recurrence:

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

Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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




Base pair at $i$ and $j$ Unmatched at $i$

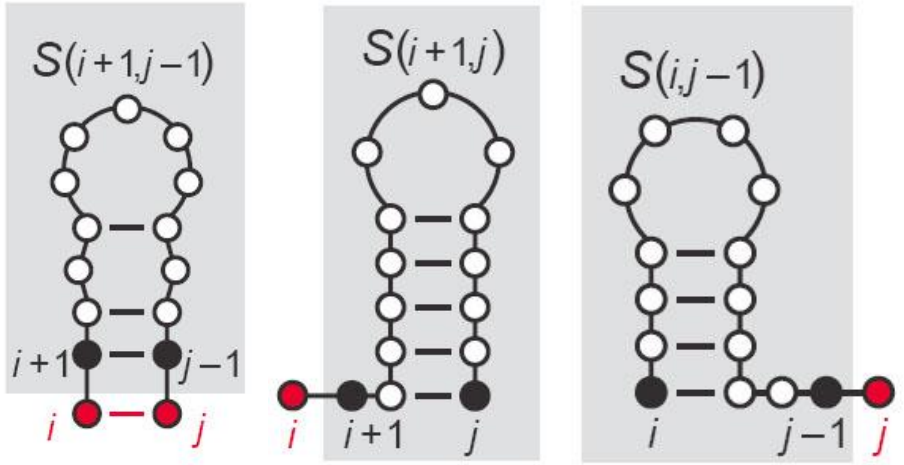
- Recurrence:

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

Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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




Base pair at $i$ and $j$ | Unmatched at $i$ | Unmatched at $j$

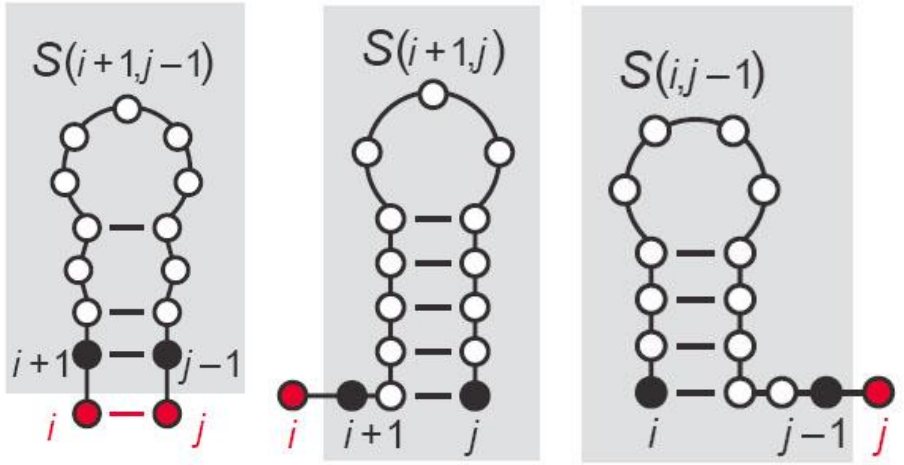
- Recurrence:

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

Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

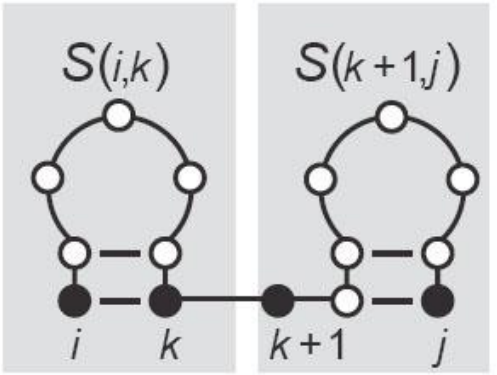
- $S(i, j)$ is the folding of the RNA subsequence of the strand from index $i$ to index $j$ which results in the highest number of base pairs.




Base pair at $i$ and $j$ | Unmatched at $i$ | Unmatched at $j$

- Recurrence:

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

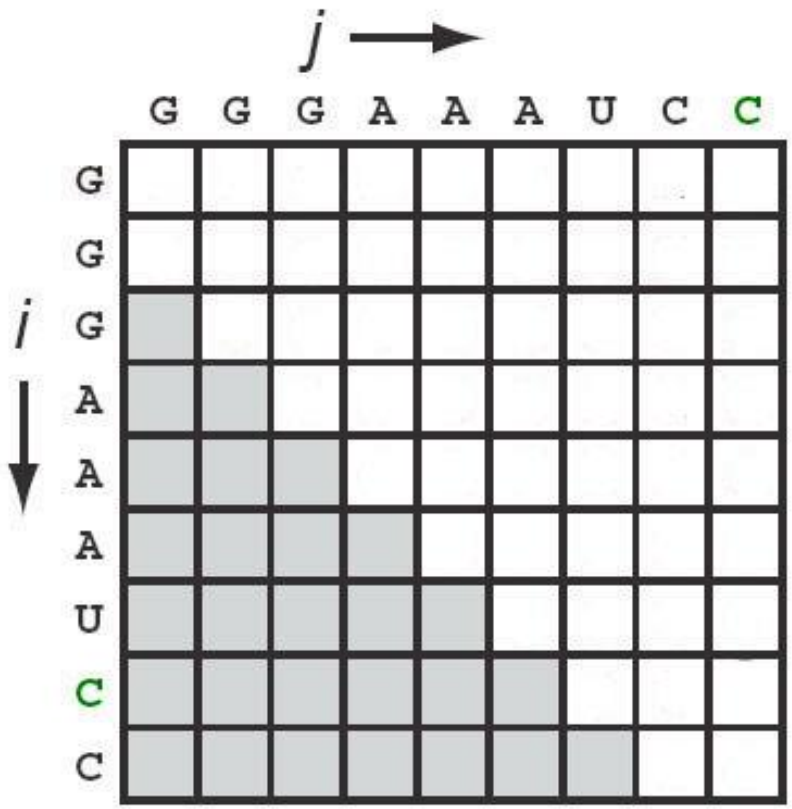



Bifurcation

Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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



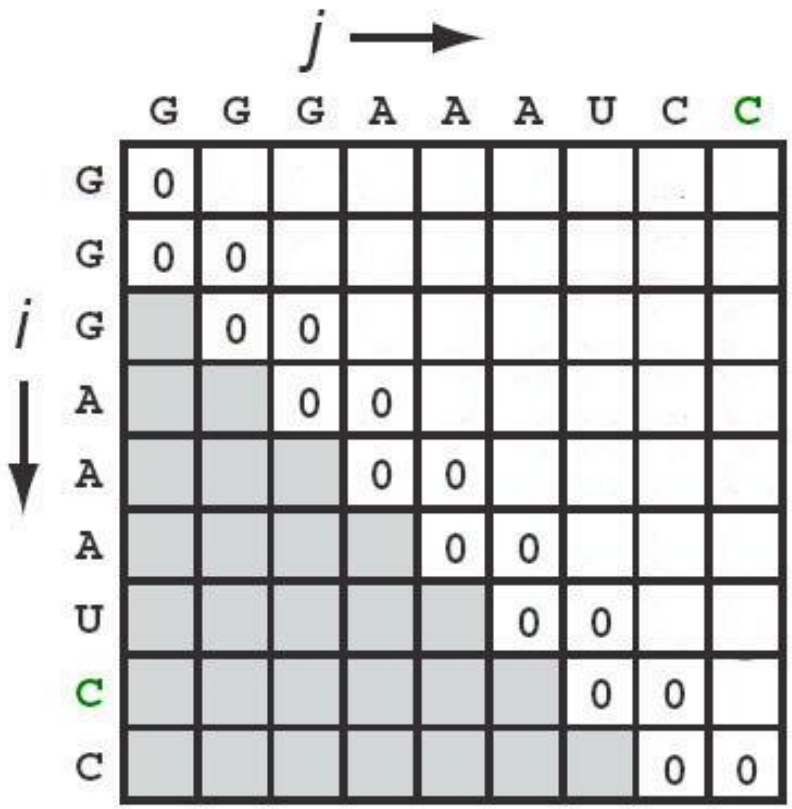

Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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




Initialize first two diagonals to 0

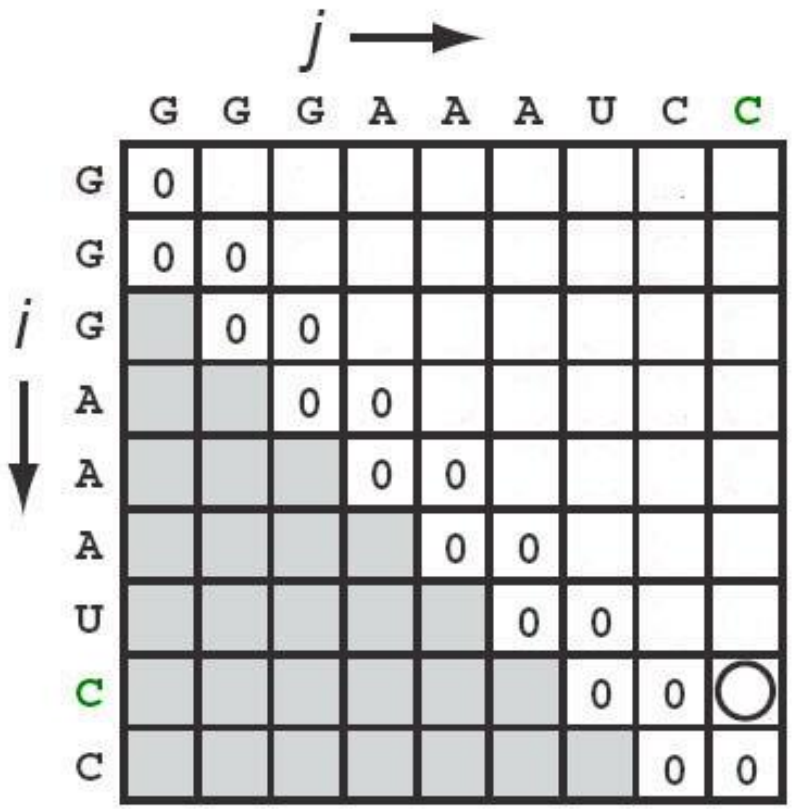
Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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




Fill in squares sweeping diagonally

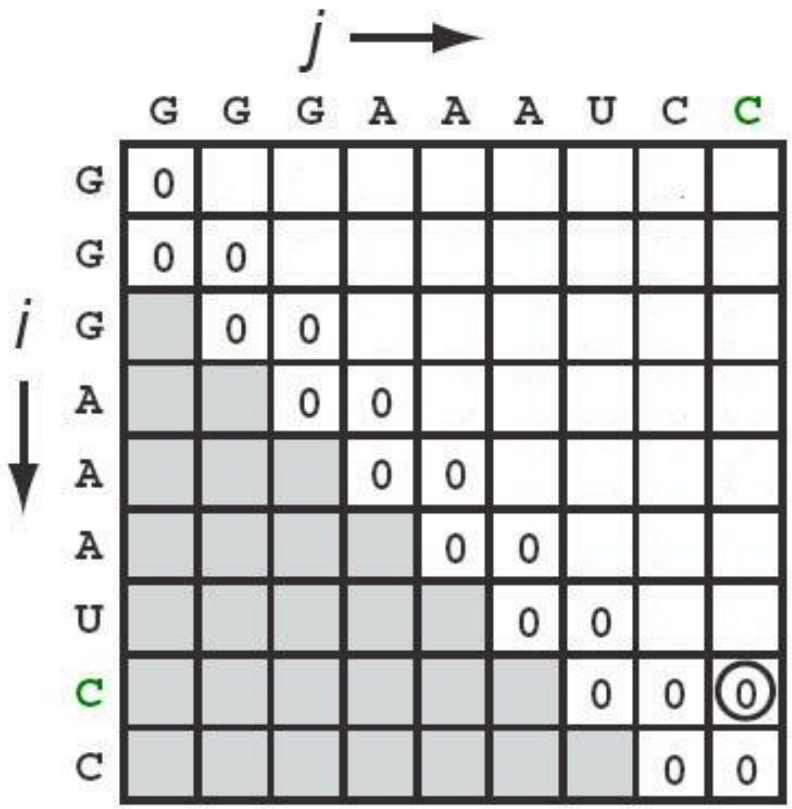
Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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




Fill in squares sweeping diagonally

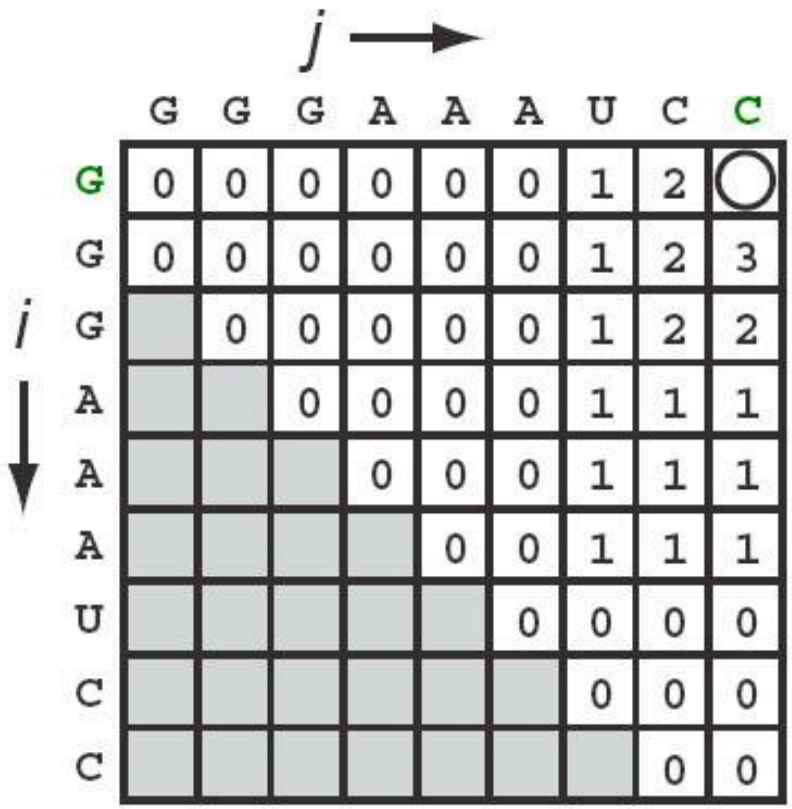
Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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




Bases cannot pair

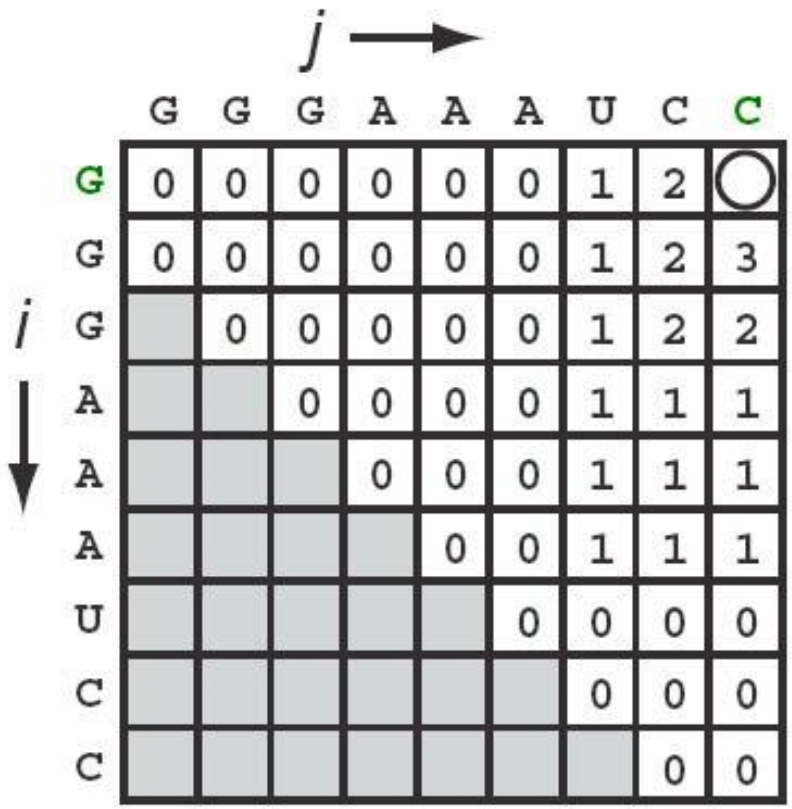
Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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




Bases can pair, similar to matched alignment

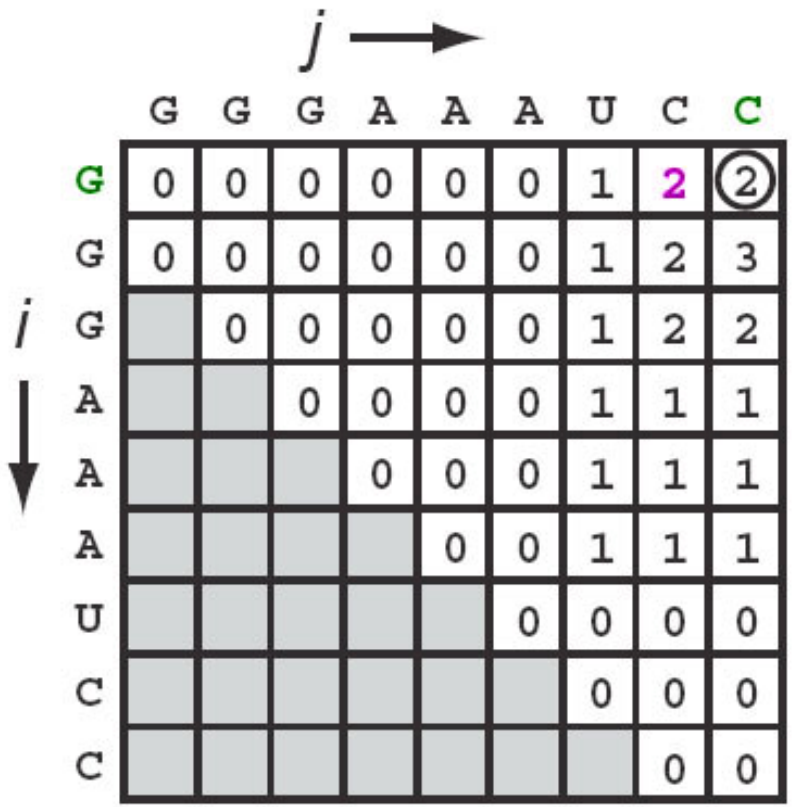
Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

- 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

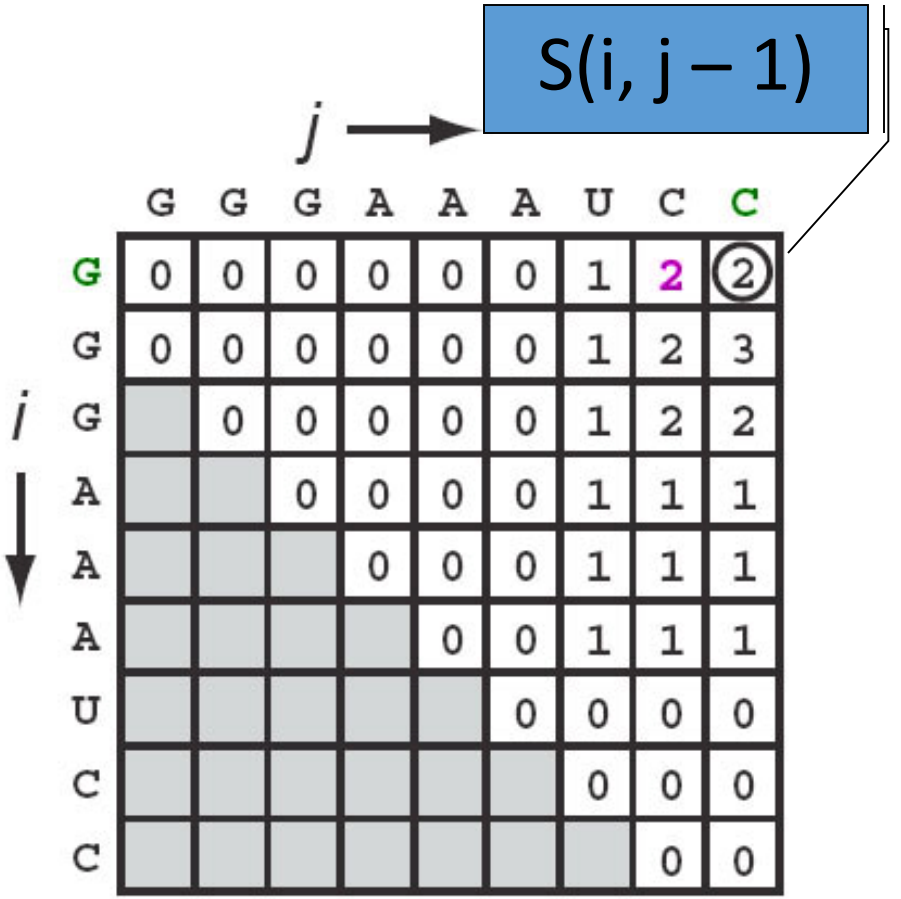
Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

- 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

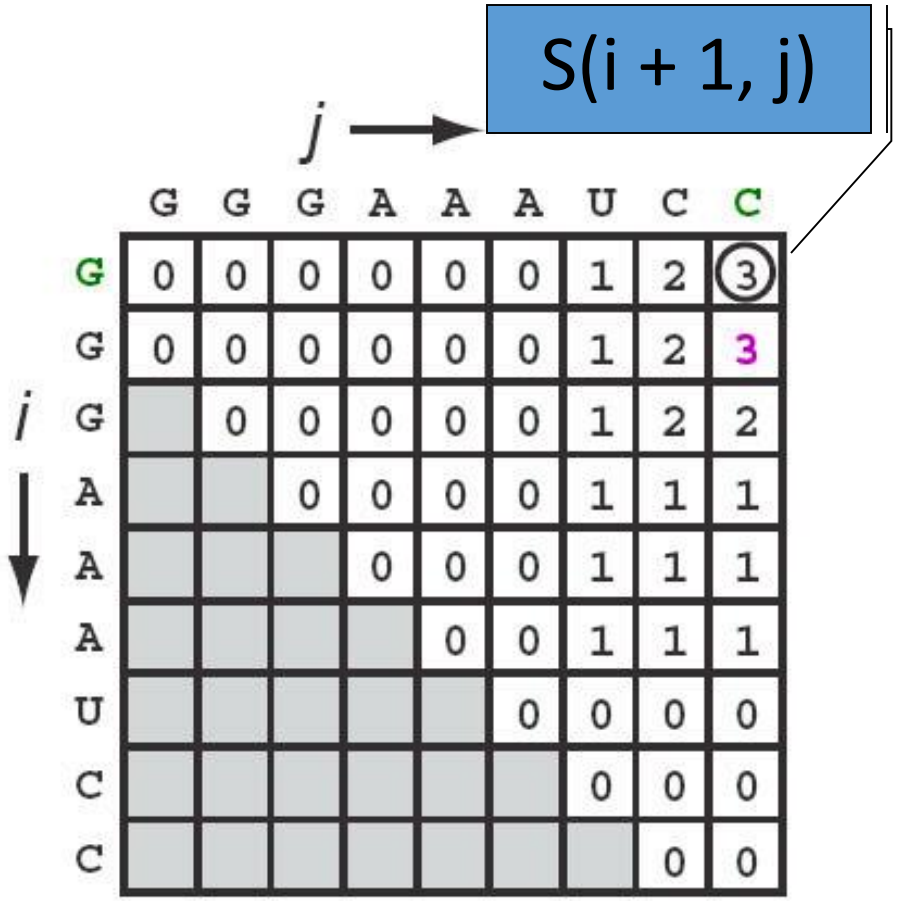
Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

- 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

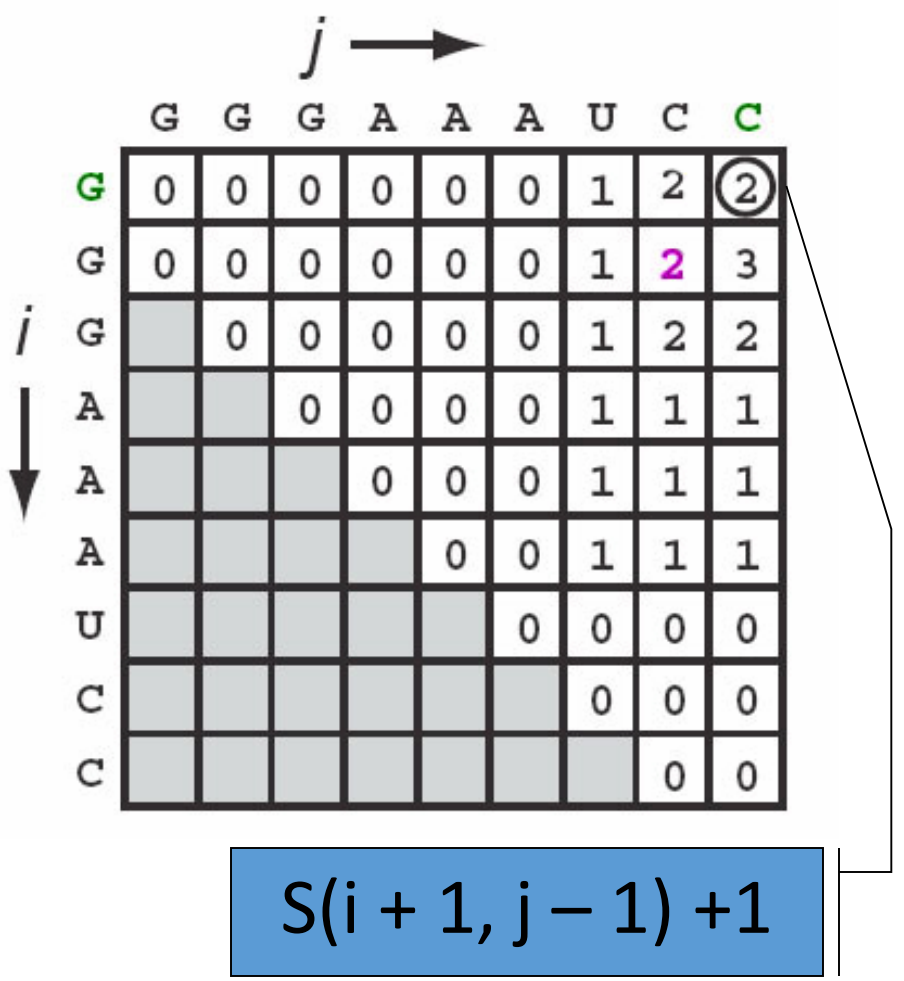
Images—Sean Eddy



Images – Sean Eddy

# Base Pair Maximization: Dynamic Programming

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

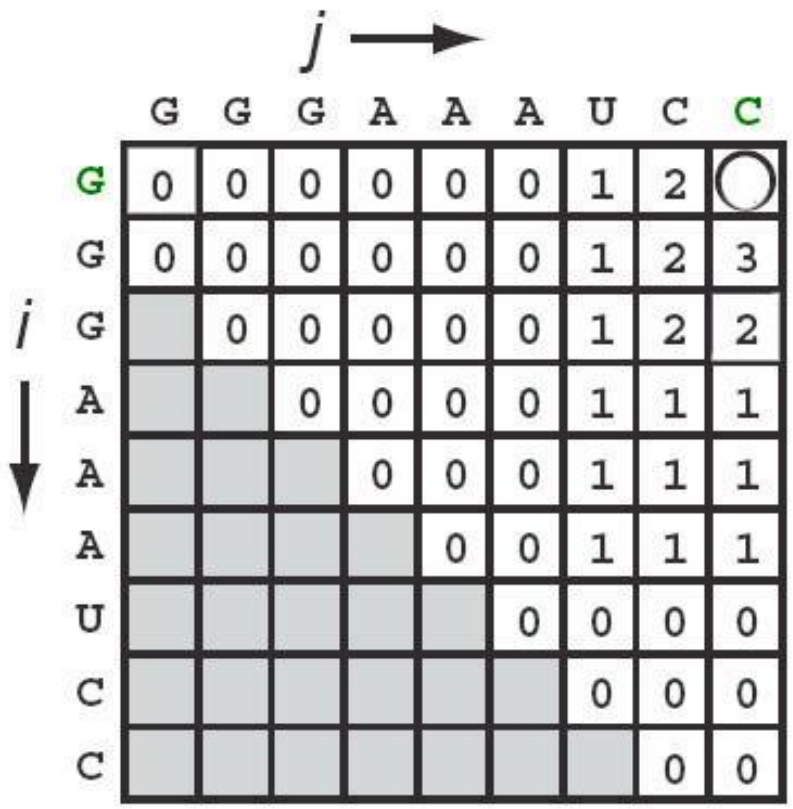



Dynamic Programming—possible paths

Images—Sean Eddy

# Base Pair Maximization: Dynamic Programming

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

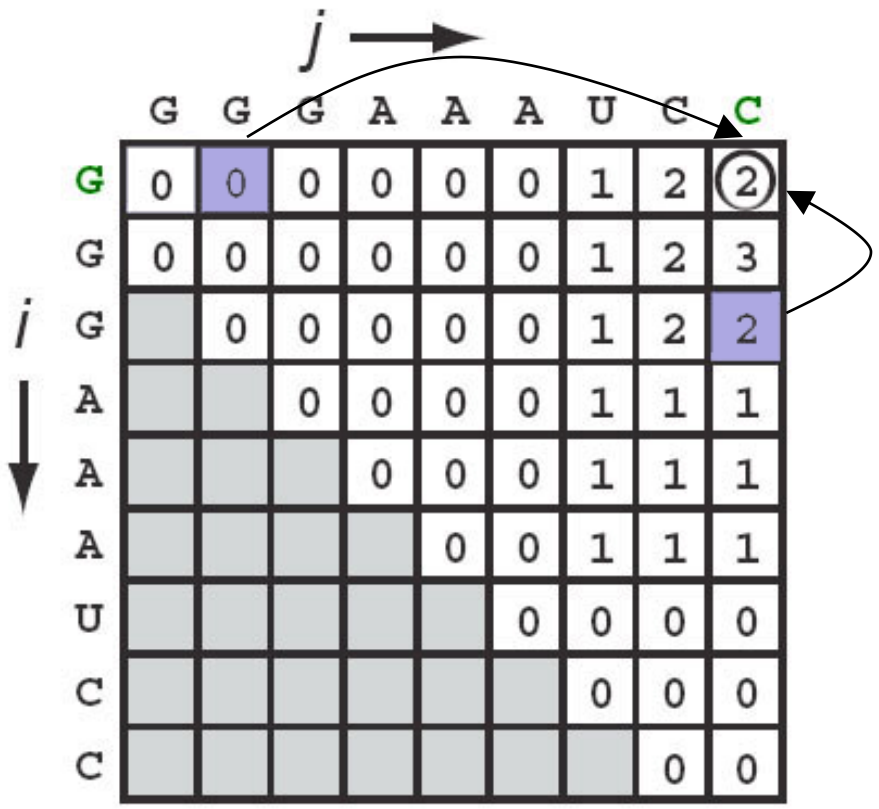



Bifurcation—add values for all *k*

Images—Sean Eddy

# Base Pair Maximization: Dynamic Programming

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

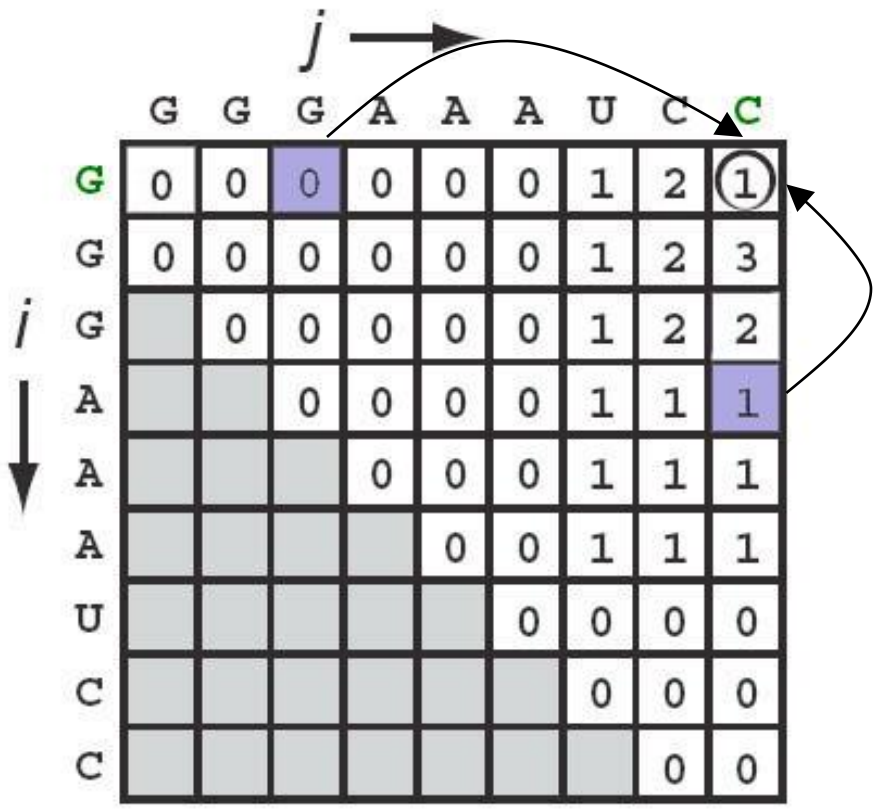



Bifurcation—add values for all *k*

Images—Sean Eddy

# Base Pair Maximization: Dynamic Programming

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

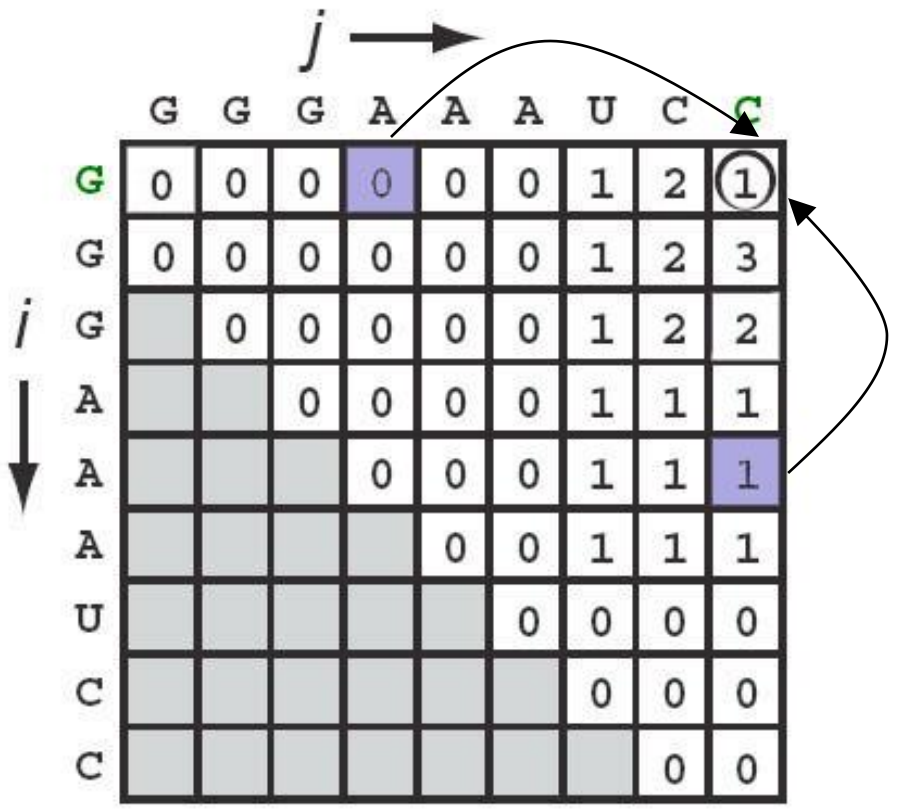



Bifurcation—add values for all *k*

Images—Sean Eddy

# Base Pair Maximization: Dynamic Programming

- 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 Eddy

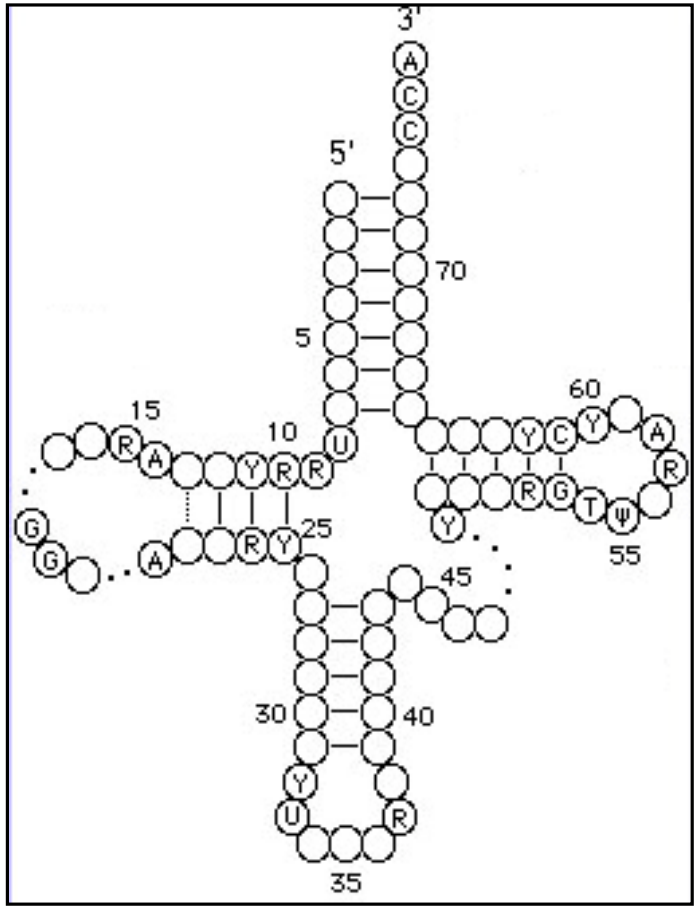
# Base Pair Maximization: Drawbacks

- Base pair maximization will not necessarily lead to the most stable structure.
  - It may create structure with many interior loops or hairpins which are energetically unfavorable.

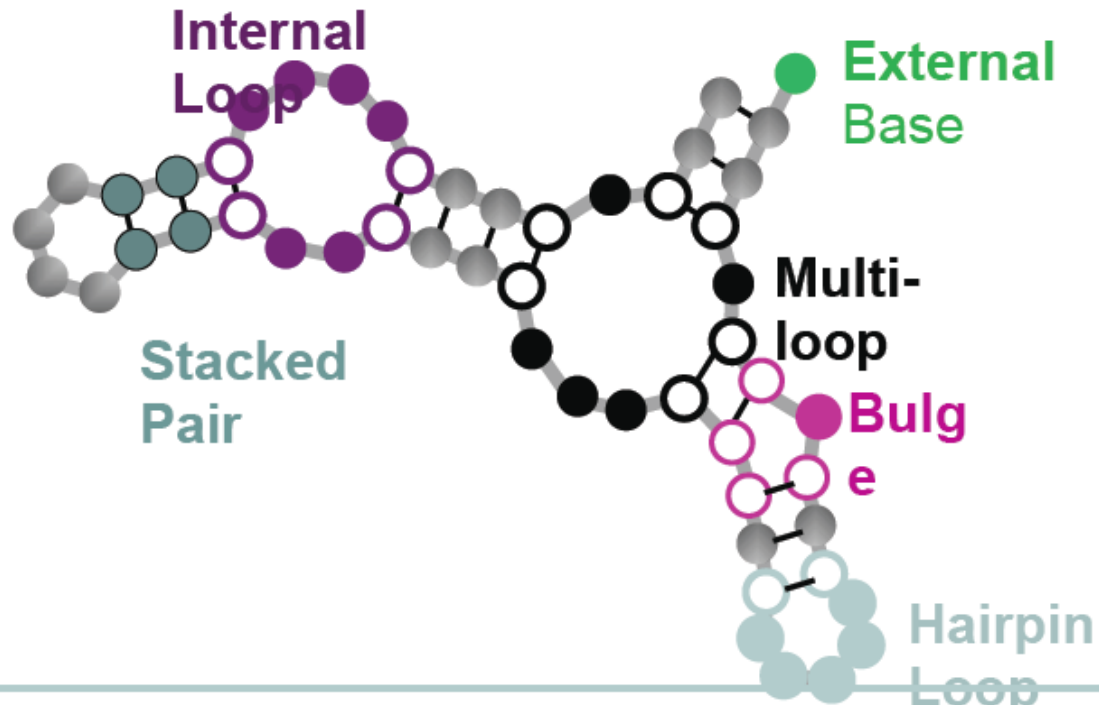
- Results comparable to aligning sequences with scattered matches—not biologically reasonable.

# Energy Minimization

- Thermodynamic Stability
  - Estimated using experimental techniques.
  - Theory : Most Stable = Most likely
- No pseudoknots due to algorithm limitations.
- Attempts to maximize the score, taking thermodynamics into account.
- MFOLD and ViennaRNA



# Energy minimization for RNA prediction




Every element is a "loop" of the 5 categories. and the total energy is the sum of all loop energies

energy is assigned to substructures, not to base pairs

# How to find the minimum energy secondary structure? (Zucker Algo)

Similar to Nussinov, we use DP

***W(j):*** energy of the optimal secondary structure for S[1..j]

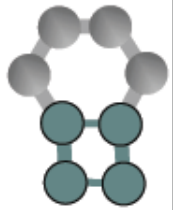
***V(i, j):*** optimal energy for S[i..j] with (i, j) forming base pair

***VBI(i, j):*** optimal energy for S[i..j] with (i, j) closing an internal loop

***VM(i, j):*** optimal energy for S[i..j] with (i, j) closing a multi-loop

# Loop energy

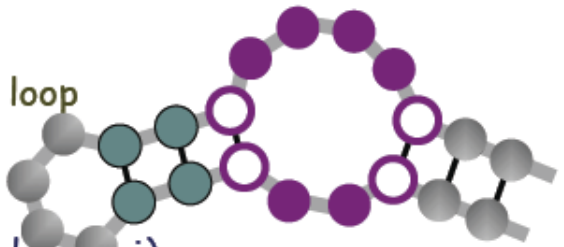
Stacked Pair



- *eS(i, j):* free energy of the stacking pair consists of base pairs (i, j) and (i+1,j-1). Stacking pair stabilizes the structure and has a negative energy
- *eS depends on all the four bases involved in the stack*

Hairpin loop

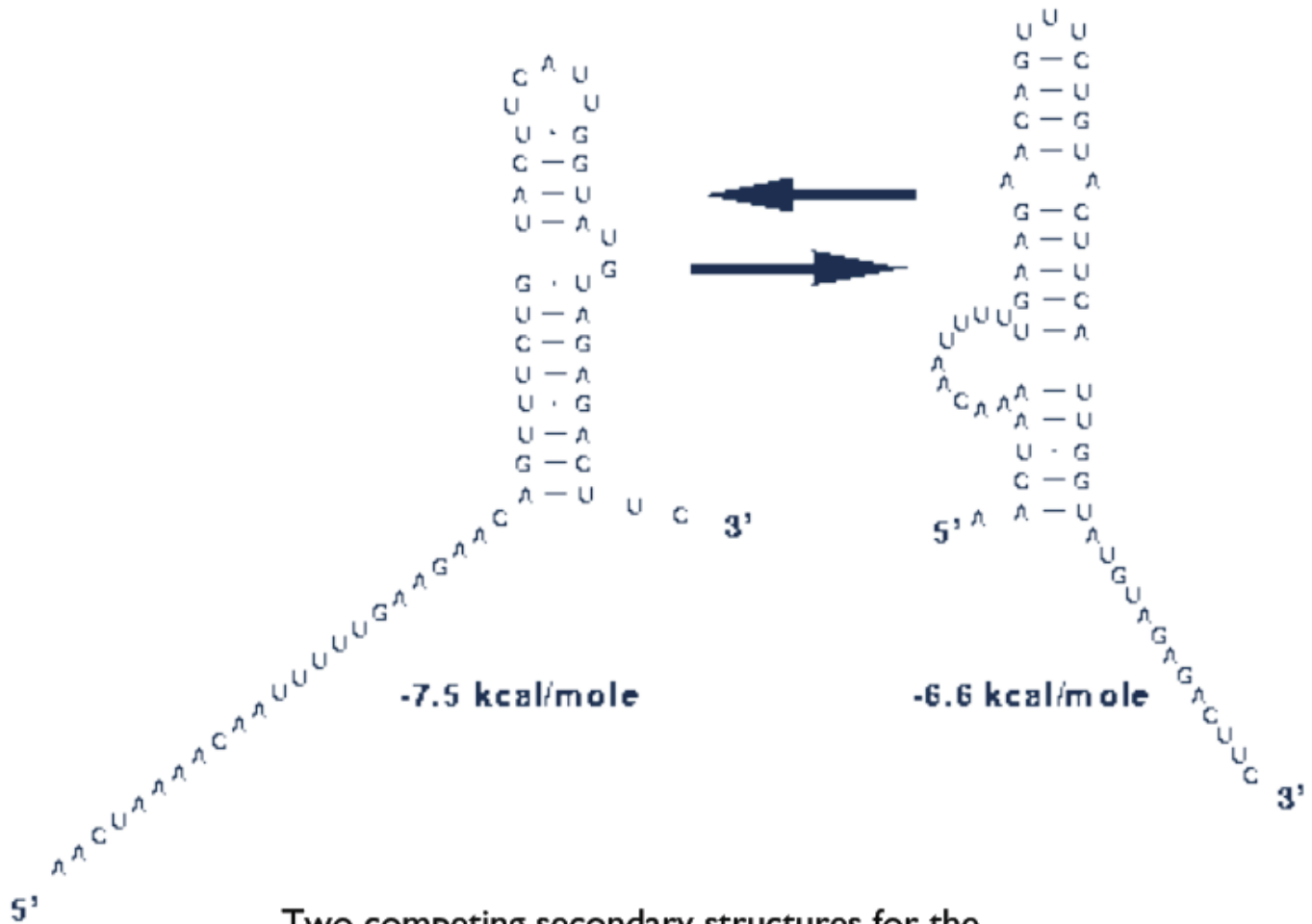
- *eH(i, j):* free energy of the hairpin closed by base pair i, j
- Depends on loopsize; the unpaired bases adjacent to *i,j* in loop



- eL(i,j,i',j'): free energy of an internal loop or bulge enclosed by (i, j) and consists of 2 base pairs.
- *Similar to eH; depends on four paired bases (I,j,i',j'); loopsize;*
- *Unpaired bases next to the paired bases*

**Internal Loop**

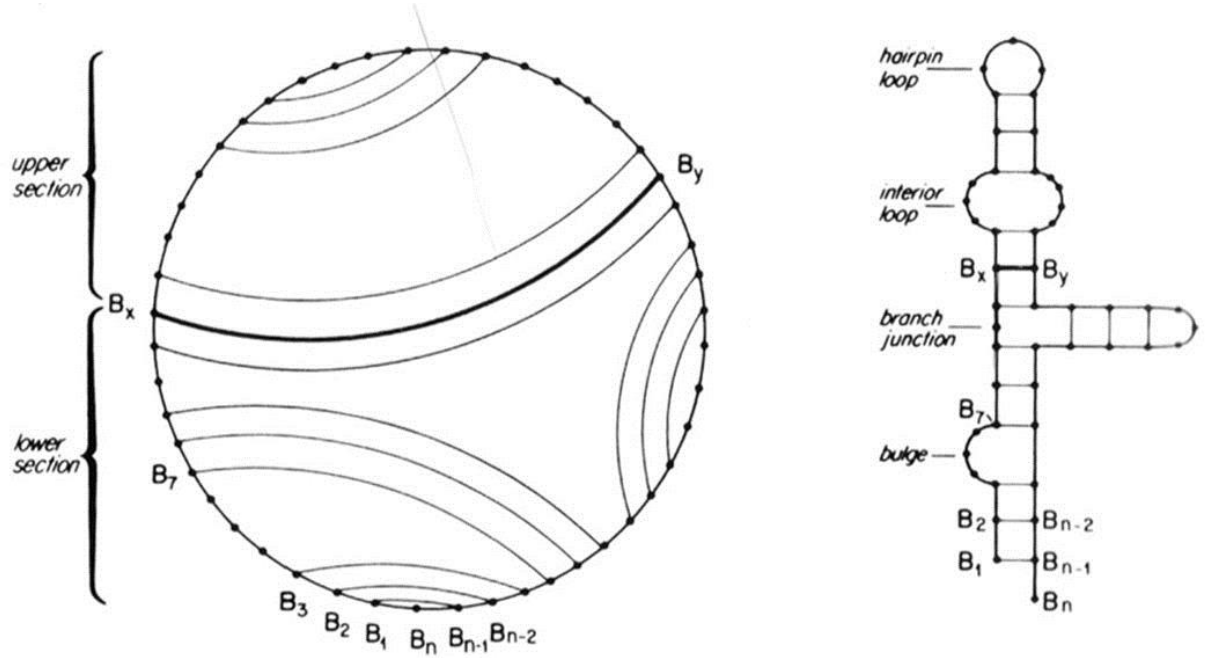
- eM(i,j,$i_1$,$j_1$,…,$i_k$,$j_k$): free energy of a multi-loop enclosed by (i, j) and consists of k+1 base pairs.
- *Similar to eL, but many approximations used.*




Two competing secondary structures for the Leptomonas collosoma spliced leader mRNA.

# Energy Minimization Results

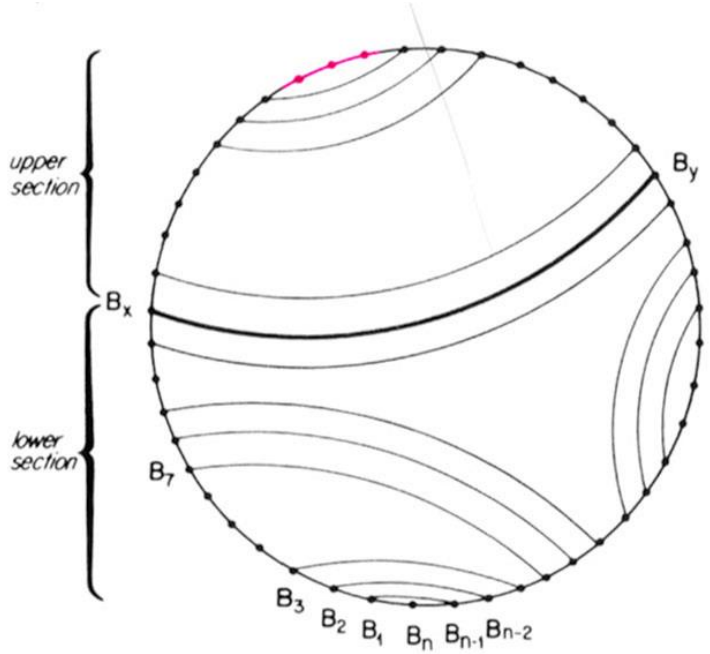
- Linear RNA strand folded back on itself to create secondary structure
- Circularized representation uses this requirement
  - Arcs represent base pairing

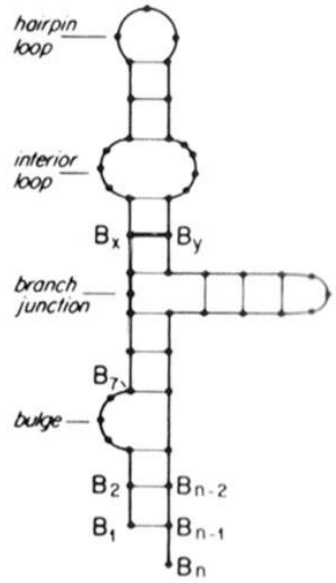



Images – David Mount

# Energy Minimization Results

- All loops must have exactly three bases in them.
  - Equivalent to having at least three base pairs between arc endpoints.

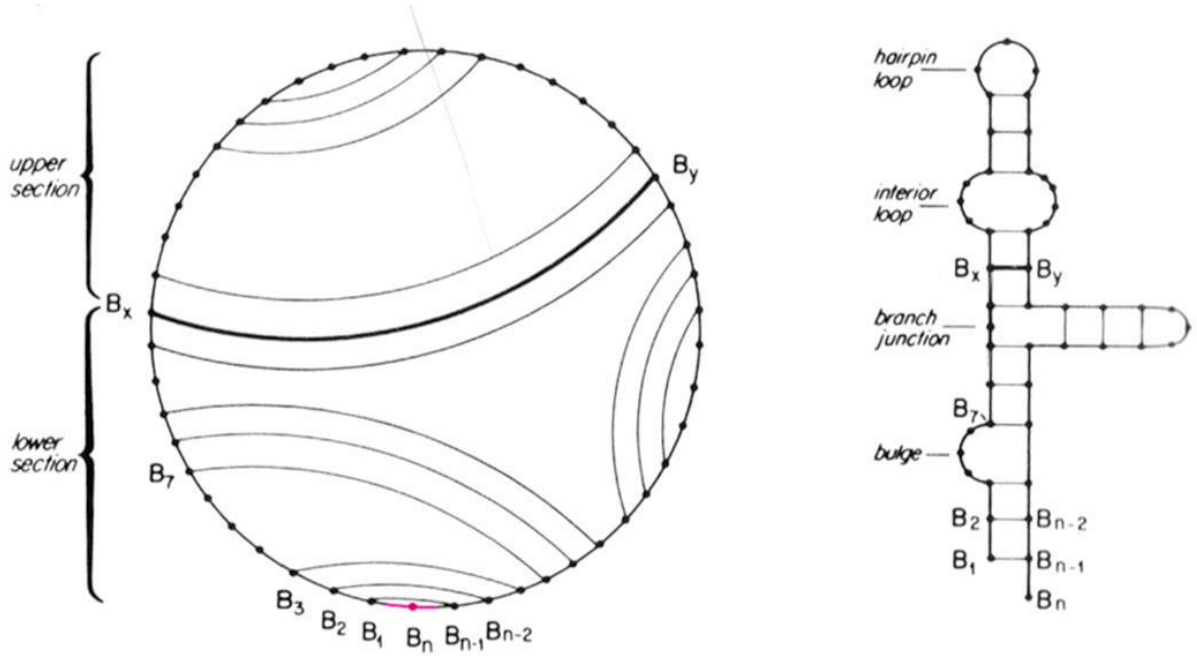



Images – David Mount

# Energy Minimization Results

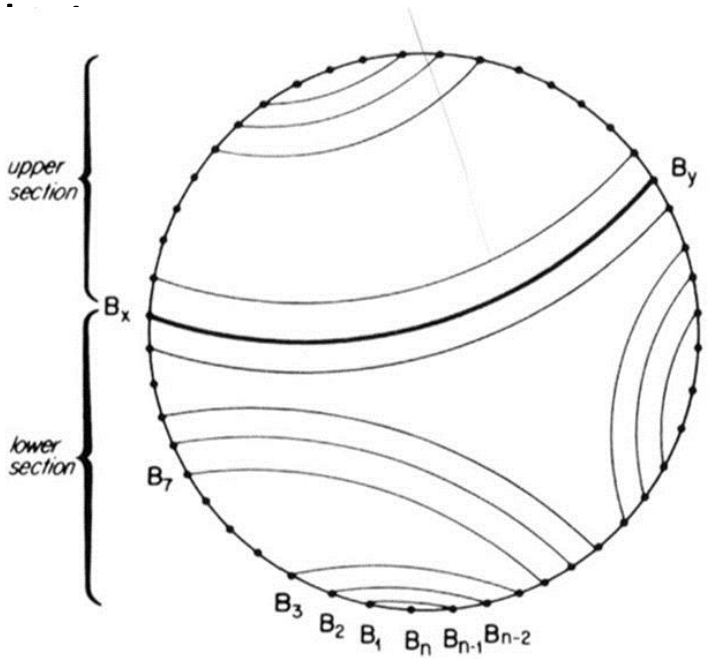
- All loops must have exactly three bases in them.
  - **Exception**: Location where beginning and end of RNA come together in circularized representation.

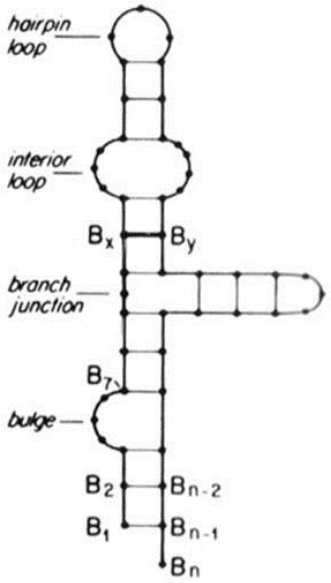



Images – David Mount

# 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

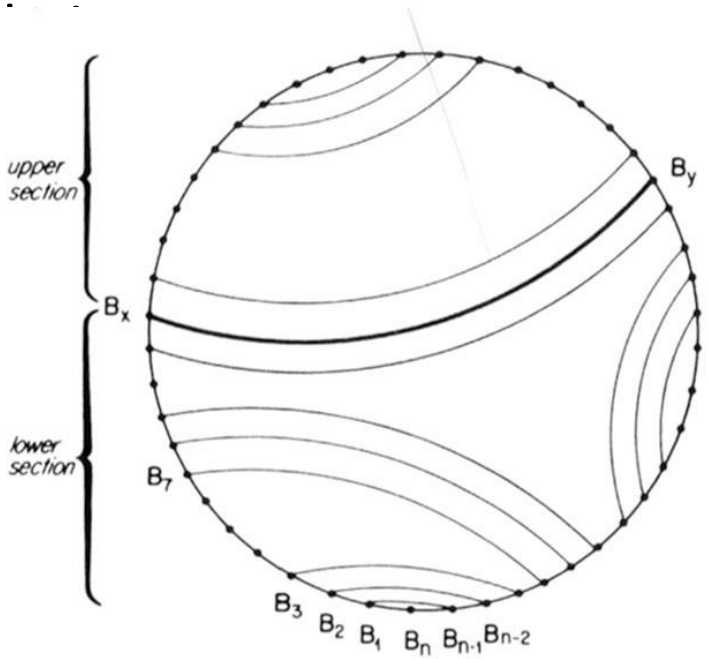


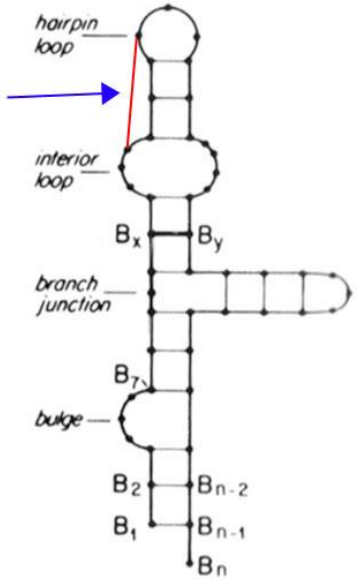




Images – David Mount

# 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

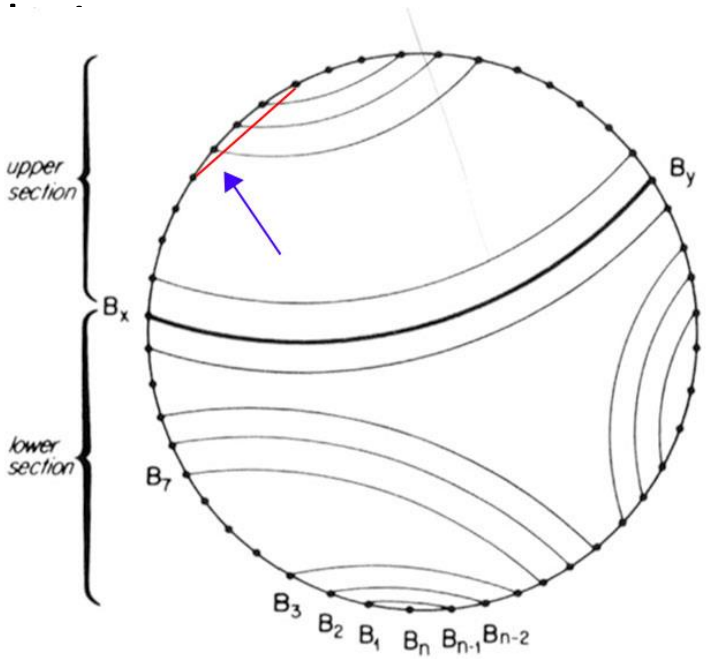


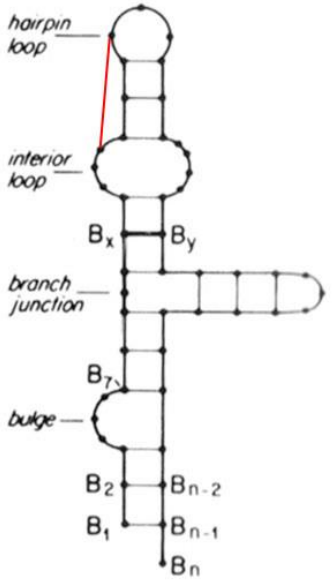




Images – David Mount

# 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



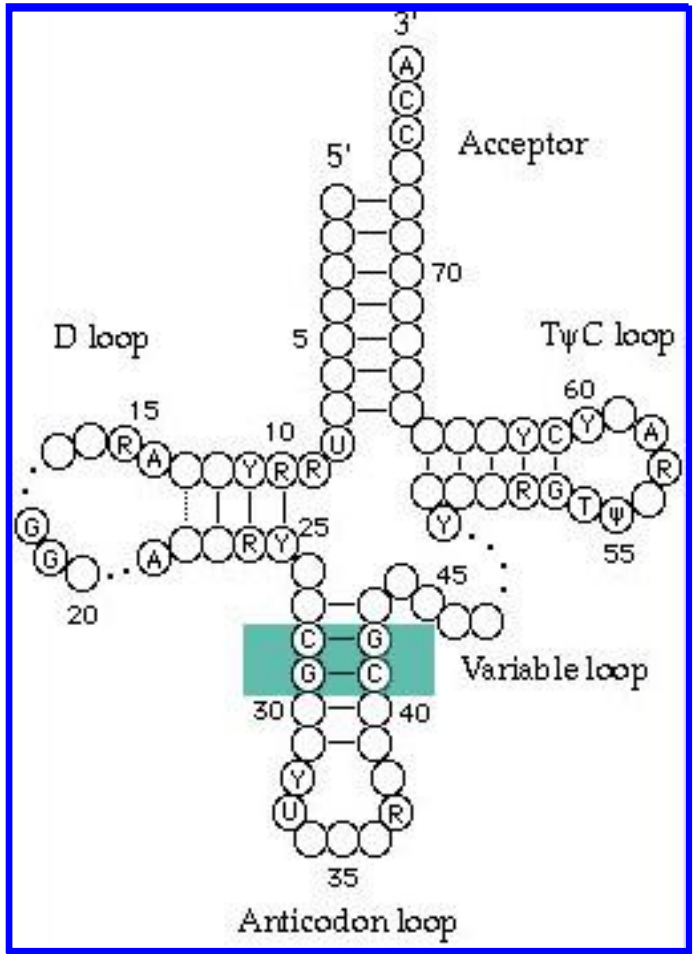

Images – David Mount

# Energy Minimization: Drawbacks

- Computes only one optimal structure.
- Optimal solution may not represent the biologically correct solution.

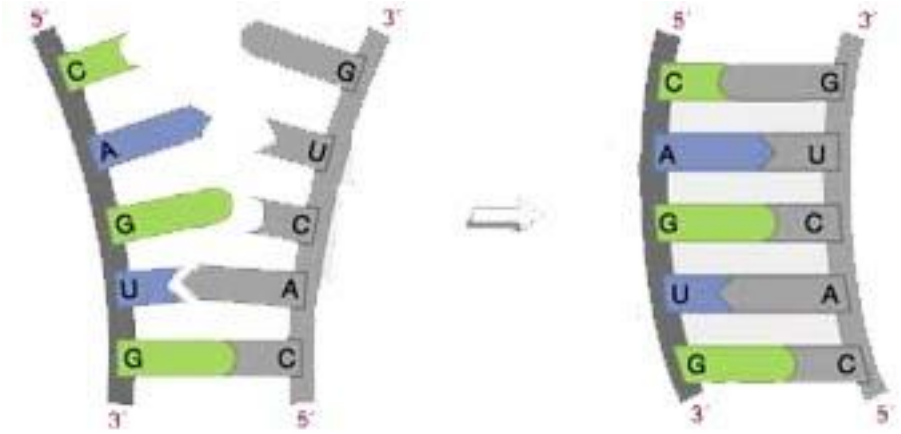
# Alternative Algorithms - Covariaton

- Expect areas of base pairing in tRNA to be covarying between various species. Find this by sequence alignment.
- Base pairing creates same stable tRNA structure in organisms.



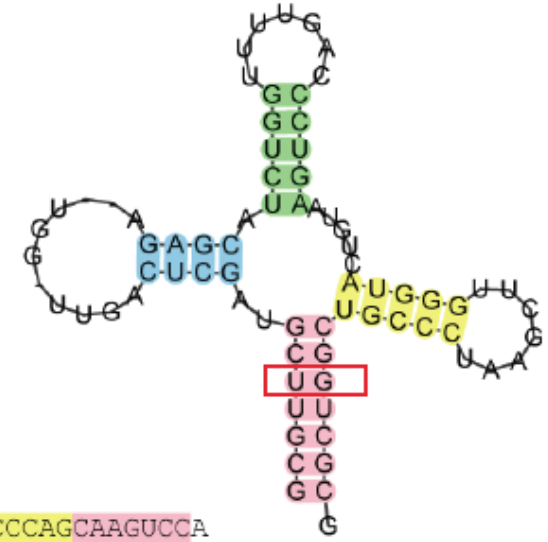
# Sequence Alignment to Determine Structure

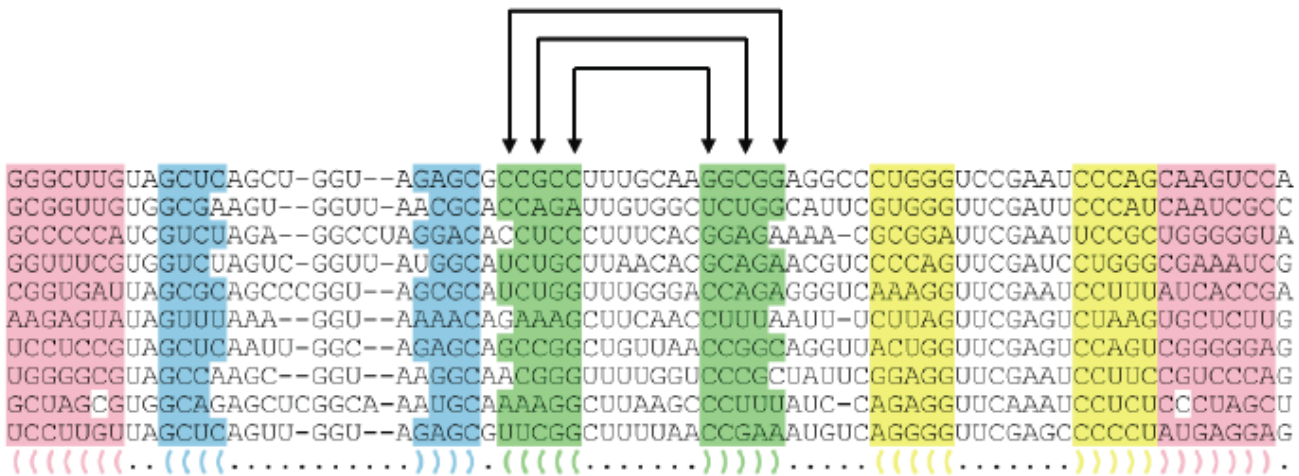
- Bases pair in order to form backbones and determine the secondary structure.
- Aligning bases based on their ability to pair with each other gives an algorithmic approach to determining the optimal structure.



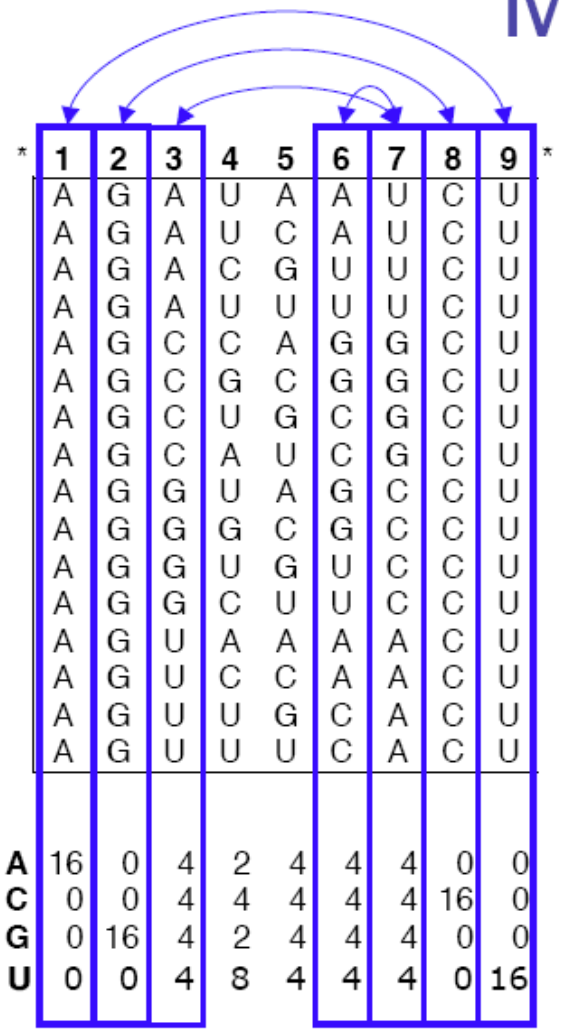
# Mutual Information in RNA structure prediction

$$I(X,Y) = \sum_i \sum_j f_{X,Y}(x_i, y_j) \cdot \log \frac{f_{X,Y}(x_i, y_j)}{f_X(x_i) \cdot f_Y(y_j)}$$






# M.I. Example



| * | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | * |
|---|---|---|---|---|---|---|---|---|---|---|
| | A | G | A | U | A | A | U | C | U | |
| | A | G | A | U | C | A | U | C | U | |
| | A | G | A | C | G | U | U | C | U | |
| | A | G | A | U | U | U | U | C | U | |
| | A | G | C | C | A | G | G | C | U | |
| | A | G | C | G | C | G | G | C | U | |
| | A | G | C | U | G | C | G | C | U | |
| | A | G | C | A | U | C | G | C | U | |
| | A | G | G | U | A | G | C | C | U | |
| | A | G | G | G | C | G | C | C | U | |
| | A | G | G | U | G | U | C | C | U | |
| | A | G | G | C | U | U | C | C | U | |
| | A | G | U | A | A | A | A | C | U | |
| | A | G | U | C | C | A | A | C | U | |
| | A | G | U | U | G | C | A | C | U | |
| | A | G | U | U | U | C | A | C | U | |
| **A** | 16 | 0 | 4 | 2 | 4 | 4 | 4 | 0 | 0 | |
| **C** | 0 | 0 | 4 | 4 | 4 | 4 | 4 | 16 | 0 | |
| **G** | 0 | 16 | 4 | 2 | 4 | 4 | 4 | 0 | 0 | |
| **U** | 0 | 0 | 4 | 8 | 4 | 4 | 4 | 0 | 16 | |

| MI: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| **9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| **7** | 0 | 0 | **2** | **0.30** | **0** | **1** | | | |
| **6** | 0 | 0 | **1** | **0.55** | **1** | | | | |
| **5** | 0 | 0 | **0** | **0.42** | | | | | |
| **4** | 0 | 0 | **0.30** | | | | | | |
| **3** | 0 | 0 | | | | | | | |
| **2** | 0 | | | | | | | | |
| **1** | | | | | | | | | |

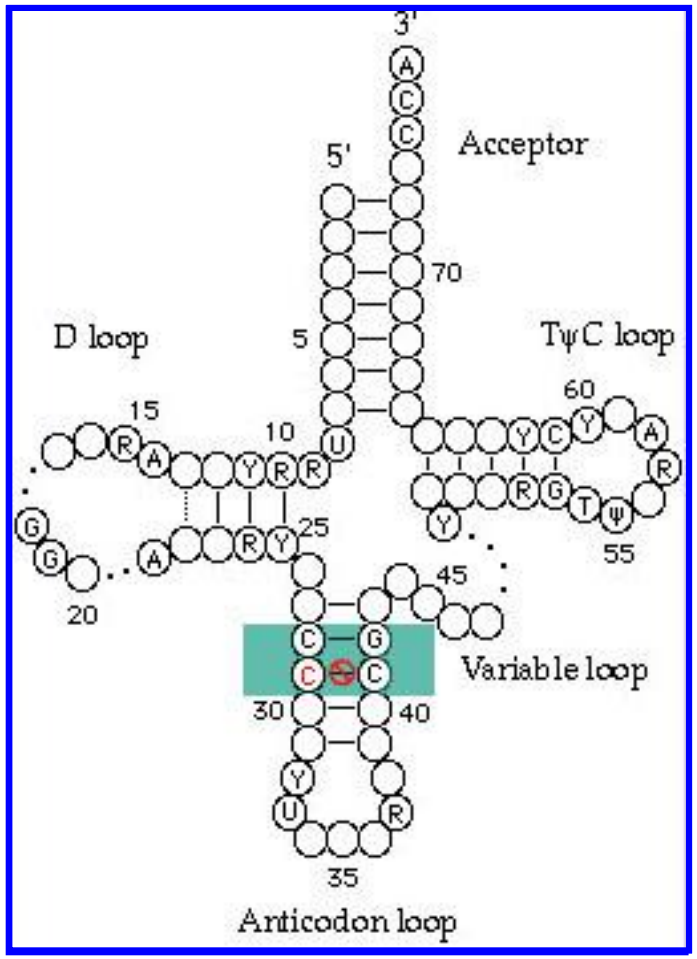
Cols 1 & 9, 2 & 8: perfect conservation and *might* be base-paired, but unclear whether they are. M.I. = 0

Cols 3 & 7: completely unconserved, but always W-C pairs, so seems likely that they do base-pair. M.I. = 2 bits.

Cols 7->6: unconserved, but each letter in 7 has only 2 possible mates in 6. M.I. = 1 bit.

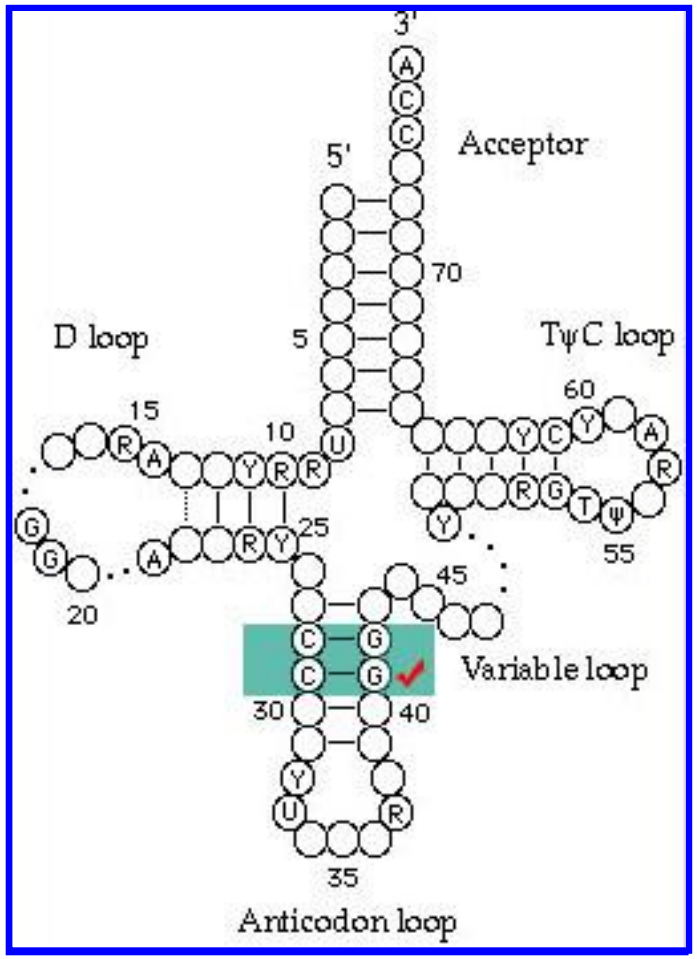
# Alternative Algorithms - Covariaton

- Expect areas of base pairing in tRNA to be covarying between various species.
- Base pairing creates same stable tRNA structure in organisms.
- Mutation in one base yields pairing impossible and breaks down structure.



# Alternative Algorithms - Covariaton

- Expect areas of base pairing in tRNA to be covarying between various species.
- Base pairing creates same stable tRNA structure in organisms.
- Mutation in one base yields pairing impossible and breaks down structure.
- Covariation ensures ability to base pair is maintained and RNA structure is conserved.



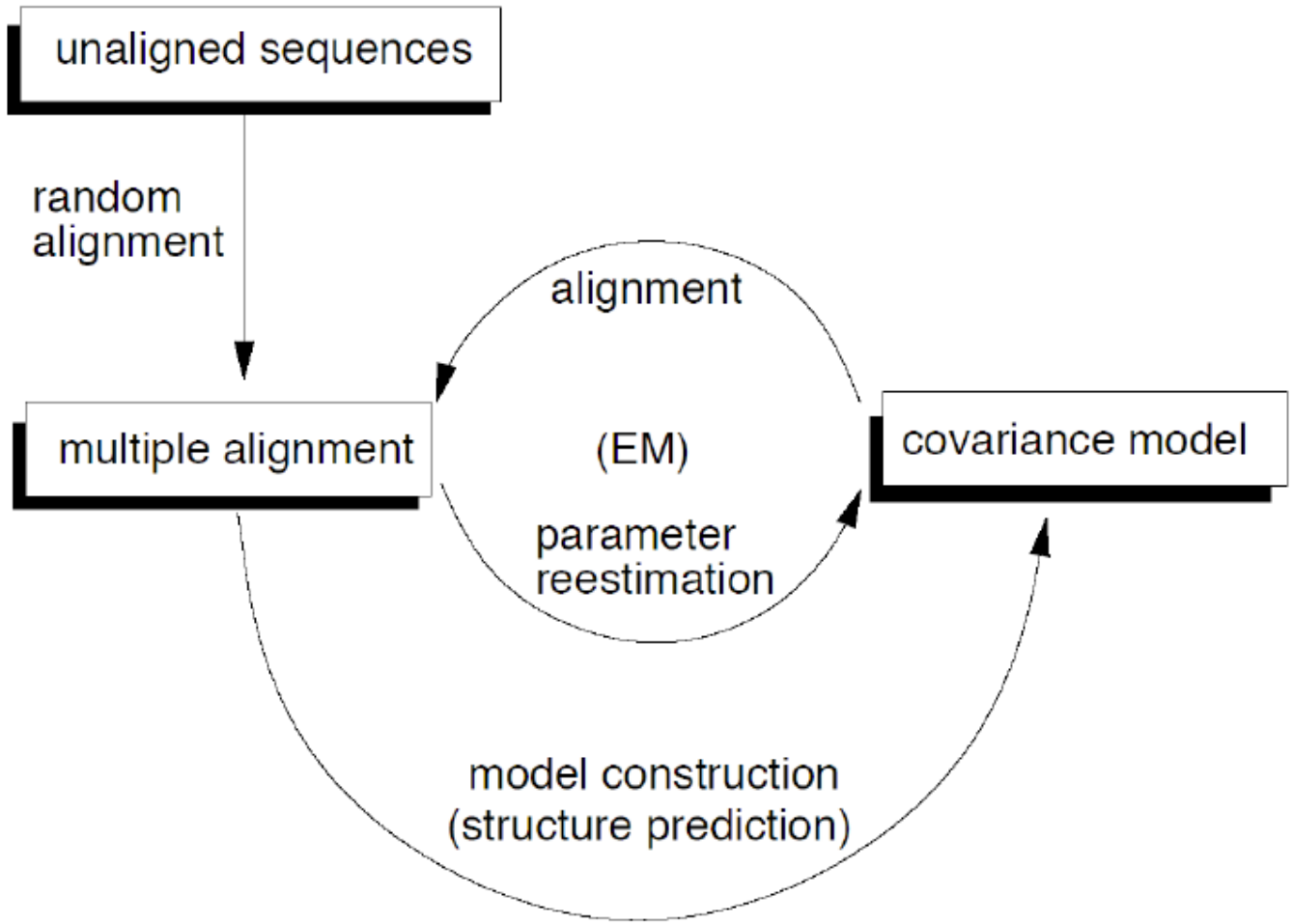
# MI-Based Structure-Learning

- find best (max total MI) subset of column pairs among i…j, subject to absence of pseudo-knots

$$S_{i,j} = \max \begin{cases} S_{i+1,j} \\ S_{i,j-1} \\ S_{i+1,j-1} + M_{i,j} \\ \max_{i<j<k} S_{i,k} + S_{k+1,j} \end{cases}$$

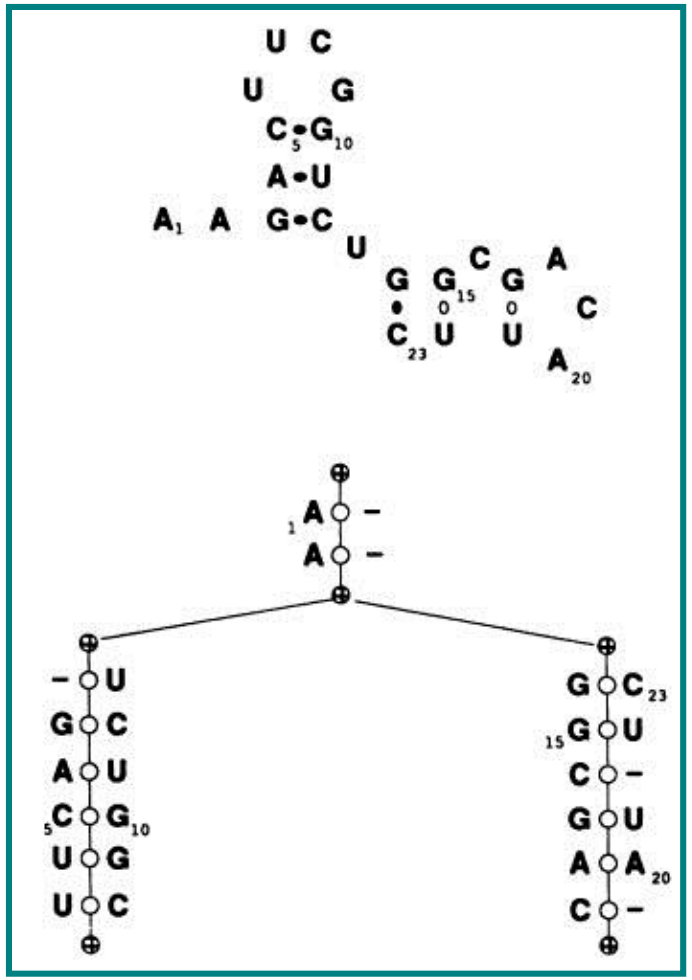
- "just like Nussinov/Zucker folding"
- BUT, need enough data---enough sequences at right phylogenetic distance

# Model Training



# Binary Tree Representation of RNA Secondary Structure

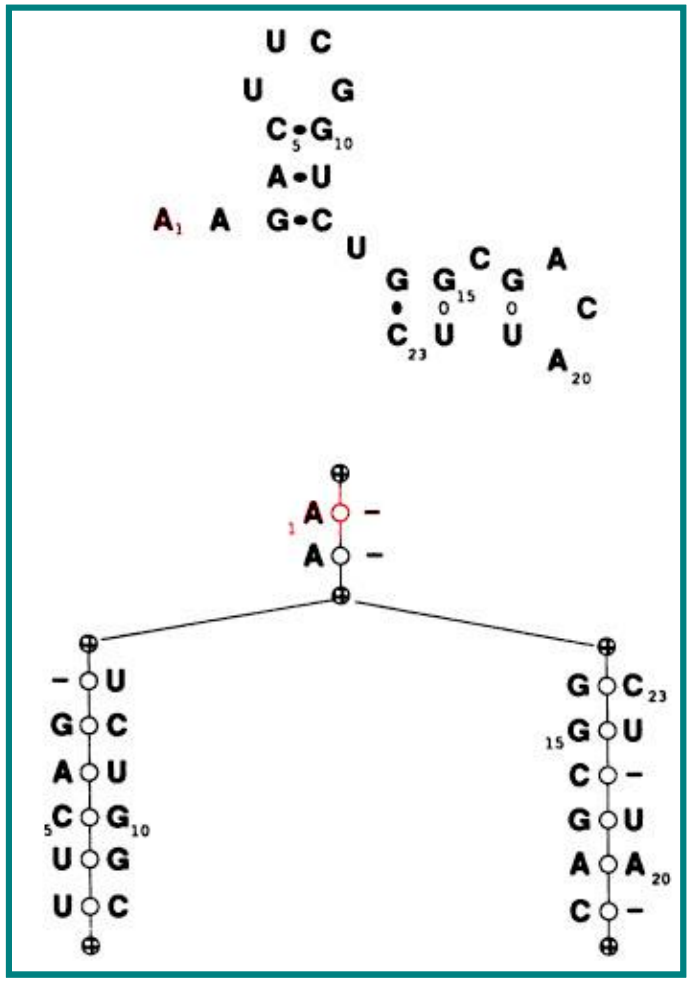
- Representation of RNA structure using Binary tree
- Nodes represent
  - Base pair if two bases are shown
  - Loop if base and “gap” (dash) are shown
- Pseudoknots still not represented
- Tree does not permit varying sequences
  - Mismatches
  - Insertions & Deletions



Images – Eddy et al.

# Binary Tree Representation of RNA Secondary Structure

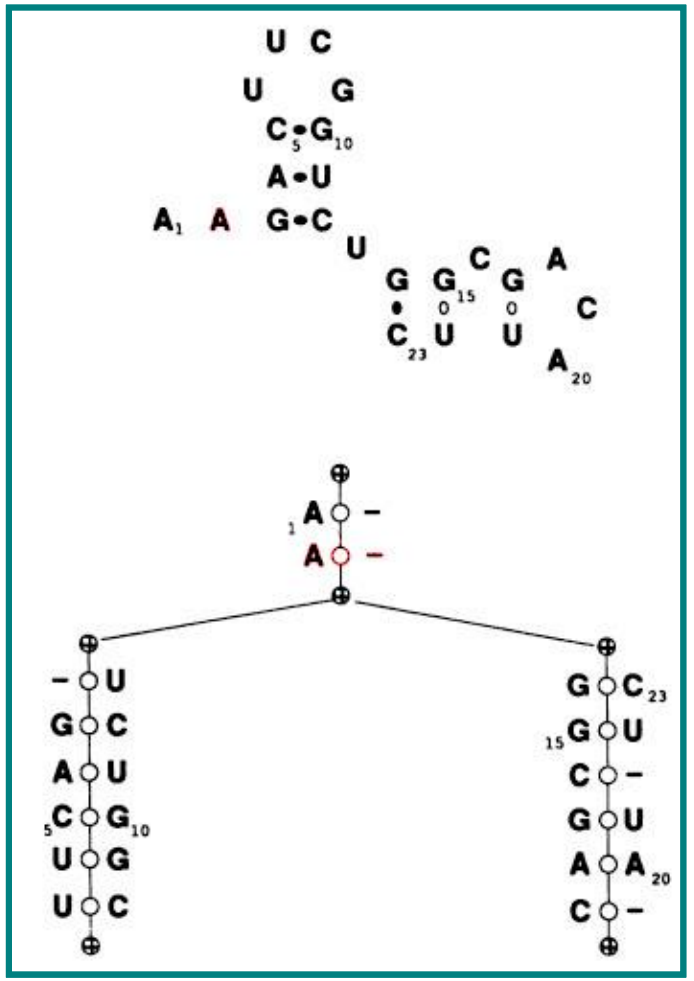
- Representation of RNA structure using Binary tree
- Nodes represent
  - Base pair if two bases are shown
  - Loop if base and “gap” (dash) are shown
- Pseudoknots still not represented
- Tree does not permit varying sequences
  - Mismatches
  - Insertions & Deletions



Images – Eddy et al.

# Binary Tree Representation of RNA Secondary Structure

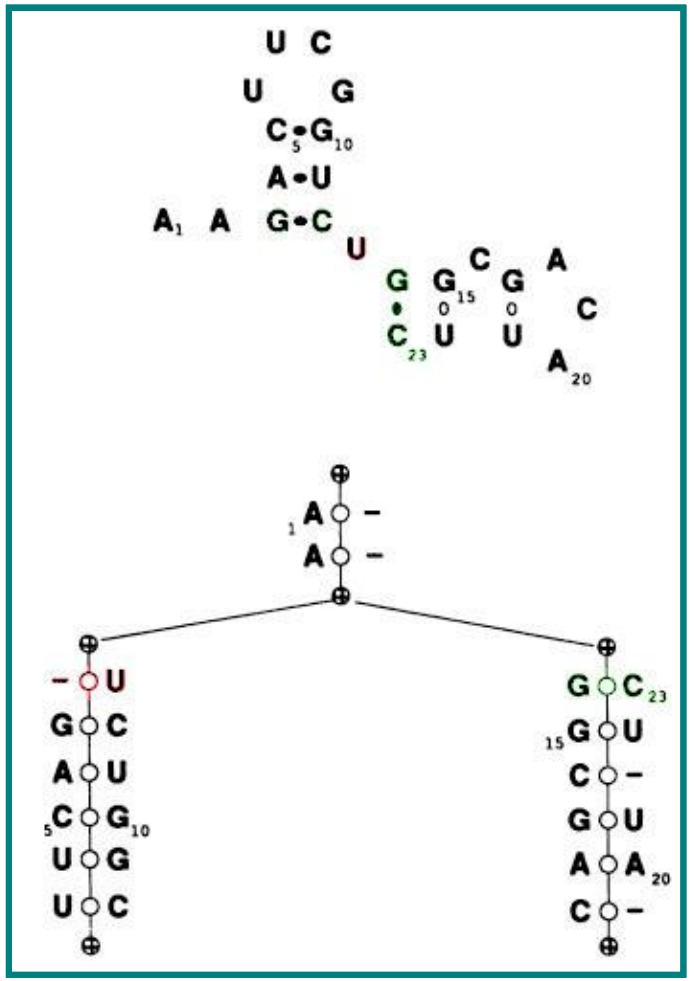
- Representation of RNA structure using Binary tree
- Nodes represent
  - Base pair if two bases are shown
  - Loop if base and “gap” (dash) are shown
- Pseudoknots still not represented
- Tree does not permit varying sequences
  - Mismatches
  - Insertions & Deletions



Images – Eddy et al.

# 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.

# Covariance Model

- **Covariance Model**: HMM which permits flexible alignment to an RNA structure – emission and transition probabilities
- Model trees based on finite number of states
  - Match states – sequence conforms to the model:
    - MATP: State in which bases are paired in the model and sequence.
    - MATL & MATR: State in which either right or left bulges in the sequence and the model.
  - Deletion – State in which there is deletion in the sequence when compared to the model.
  - Insertion – State in which there is an insertion relative to model.

# Covariance Model

- **Covariance Model**: HMM which permits flexible alignment to an RNA structure – emission and transition probabilities
- Transitions have probabilities.
  - Varying probability: Enter insertion, remain in current state, etc.
  - Bifurcation: No probability, describes path.

# Covariance Model (CM) Training Algorithm

- $S(i, j)$ = Score at indices $i$ and $j$ in RNA when aligned to the Covariance Model.

$$M_{i,j} = \sum_{x_i, x_j} f_{x_i x_j} \log_2 \frac{f_{x_i x_j}}{f_{x_i} f_{x_j}}$$

$$S(i, j) = \max \begin{cases} S(i+1, j-1) + M(i, j) \\ S(i+1, j) \\ S(i, j-1) \\ \max_{i<k<j} S(i, k) + S(k+1, j) \end{cases}$$

- Frequencies obtained by aligning model to “training data”—consists of sample sequences.
  - Reflect values which optimize alignment of sequences to model.

# Covariance Model (CM) Training Algorithm

- $S(i, j)$ = Score at indices $i$ and $j$ in RNA when aligned to the Covariance Model.

$$S(i, j) = \max \begin{cases} S(i+1, j-1) + M(i, j) \\ S(i+1, j) \\ S(i, j-1) \\ \max_{i<k<j} S(i, k) + S(k+1, j) \end{cases}$$

$$M_{i,j} = \sum_{x_i, x_j} f_{x_i x_j} \log_2 \frac{f_{x_i x_j}}{f_{x_i} f_{x_j}}$$

Frequency of seeing the symbols (A, C, G, U) together in locations $i$ and $j$ depending on symbol.

- Frequencies obtained by aligning model to "training data"—consists of sample sequences.
  - Reflect values which optimize alignment of sequences to model.

# Covariance Model (CM) Training Algorithm

- $S(i, j)$ = Score at indices $i$ and $j$ in RNA when aligned to the Covariance Model.

$$S(i, j) = \max \begin{cases} S(i+1, j-1) + M(i, j) \\ S(i+1, j) \\ S(i, j-1) \\ \max_{i<k<j} S(i, k) + S(k+1, j) \end{cases}$$

$$M_{i,j} = \sum_{x_i, x_j} f_{x_i x_j} \log_2 \frac{f_{x_i x_j}}{f_{x_i} f_{x_j}}$$

Independent frequency of seeing the symbols (A, C, G, T) in locations i or j depending on symbol.

- Frequencies obtained by aligning model to “training data”—consists of sample sequences.
  - Reflect values which optimize alignment of sequences to model.

# Covariance Model (CM) Training Algorithm

- $S(i, j)$ = Score at indices $i$ and $j$ in RNA when aligned to the Covariance Model.

$$S(i, j) = \max \begin{cases} S(i+1, j-1) + M(i, j) \\ S(i+1, j) \\ S(i, j-1) \\ \max_{i<k<j} S(i, k) + S(k+1, j) \end{cases}$$

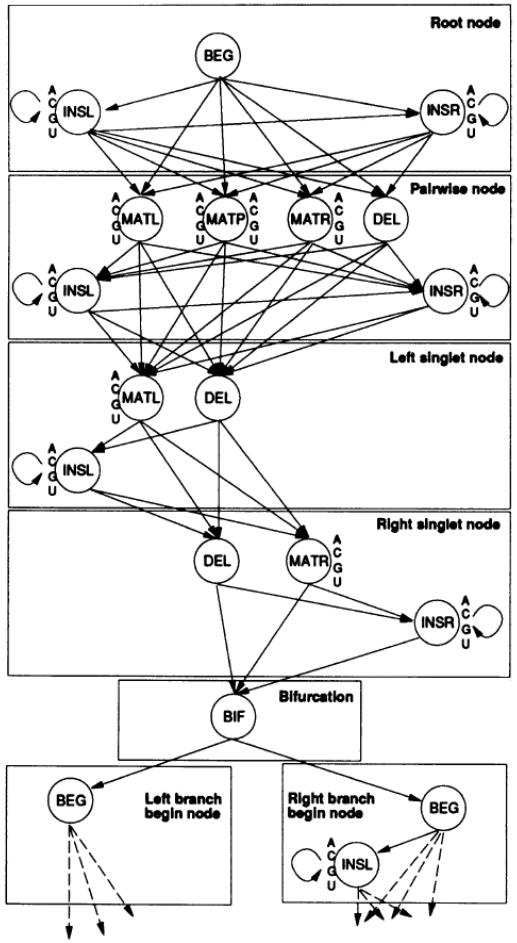
$$M_{i,j} = \sum_{x_i, x_j} f_{x_i x_j} \log_2 \frac{f_{x_i x_j}}{f_{x_i} f_{x_j}}$$

Independent frequency of seeing the symbols (A, C, G, U) in locations i or j depending on symbol.

- Frequencies obtained by aligning model to “training data”—consists of sample sequences.
  - Reflect values which optimize alignment of sequences to model.

# Alignment to CM Algorithm

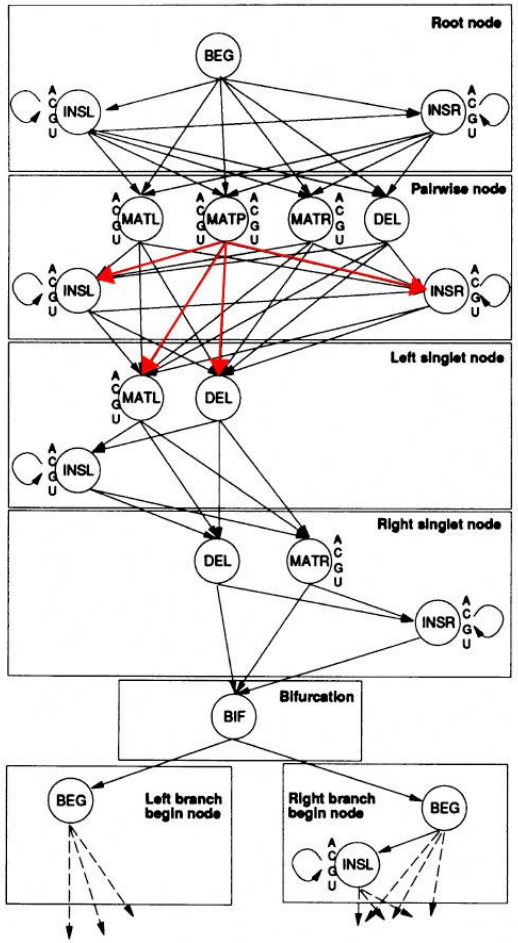
- Calculate the probability score of aligning RNA to CM.
- Three dimensional matrix—$O(n^3)$
  - Align sequence to given subtrees in CM.
  - For each subsequence, calculate all possible states.
- Subtrees evolve from bifurcations
  - For simplicity, left singlet is default.




Images—Eddy et al.

# Alignment to CM Algorithm

- For each calculation, take into account:
  - Transition (T) to next state.
  - Emission probability (P) in the state as determined by training data.

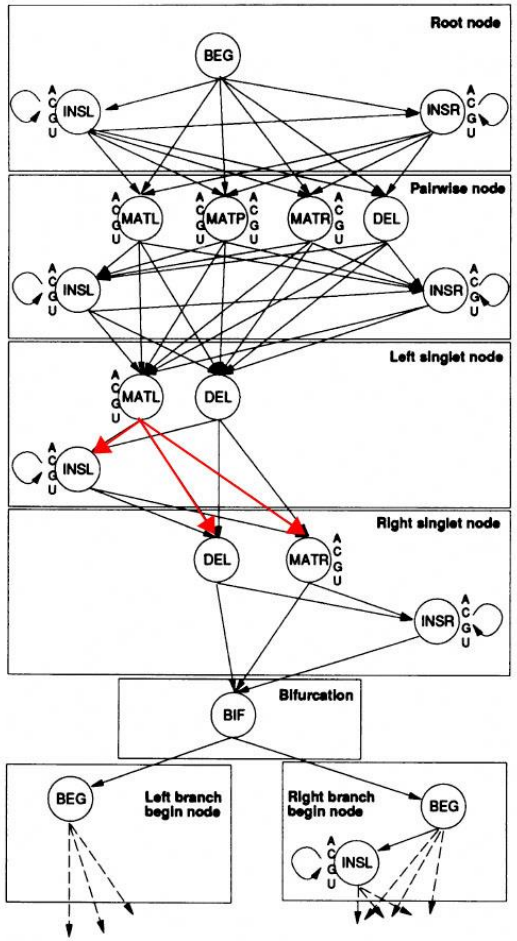



$$S_{i,j,y}(y = MATP) = \max_{y_{next}}[S_{i+1,j-1,y_{next}} + \log \mathcal{T}(y_{next} \mid y) + \log \mathcal{P}(x_i, x_j \mid y)]$$

Images—Eddy et al.

# Alignment to CM Algorithm

- For each calculation, take into account:
  - Transition (T) to next state.
  - Emission probability (P) in the state as determined by training data.

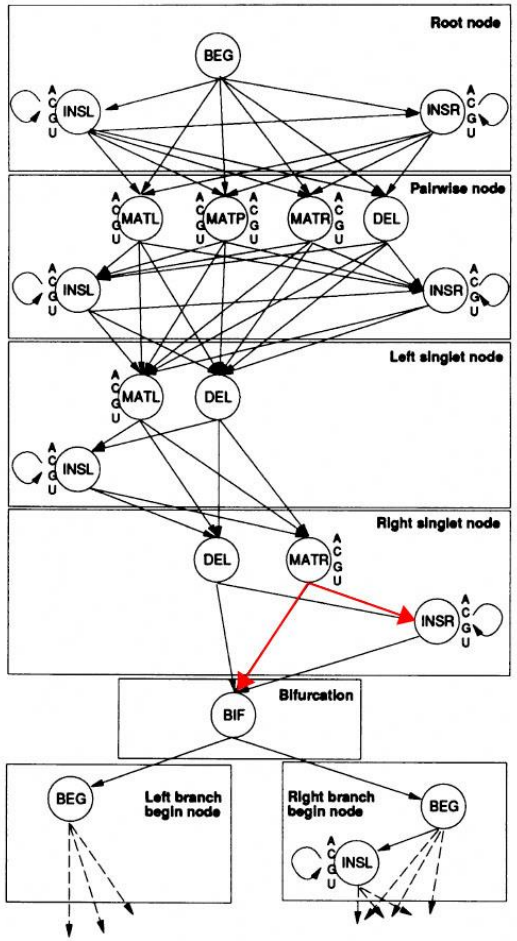



$$S_{i,j,y}(y = MATL, INSL) = \max_{y_{next}}[S_{i+1,j,y_{next}} + \log \mathcal{T}(y_{next} \mid y) + \log \mathcal{P}(x_i \mid y)]$$

Images—Eddy et al.

# Alignment to CM Algorithm

- For each calculation, take into account:
  - Transition (T) to next state.
  - Emission probability (P) in the state as determined by training data.



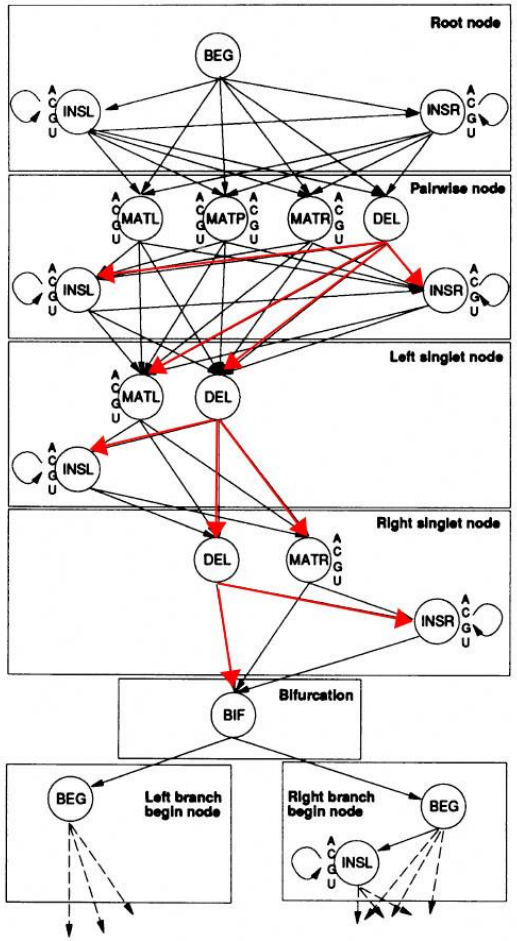

$$S_{i,j,y}(y = MATR, INSR) = \max_{y_{next}}[S_{i,j-1,y_{next}} + \log \mathcal{T}(y_{next} \mid y) + \log \mathcal{P}(x_j \mid y)]$$

Images—Eddy et al.

# Alignment to CM Algorithm

- For each calculation, take into account:
  - Transition (T) to next state.
  - Emission probability (P) in the state as determined by training data.

- Deletion—does not have emission probability associated with it.



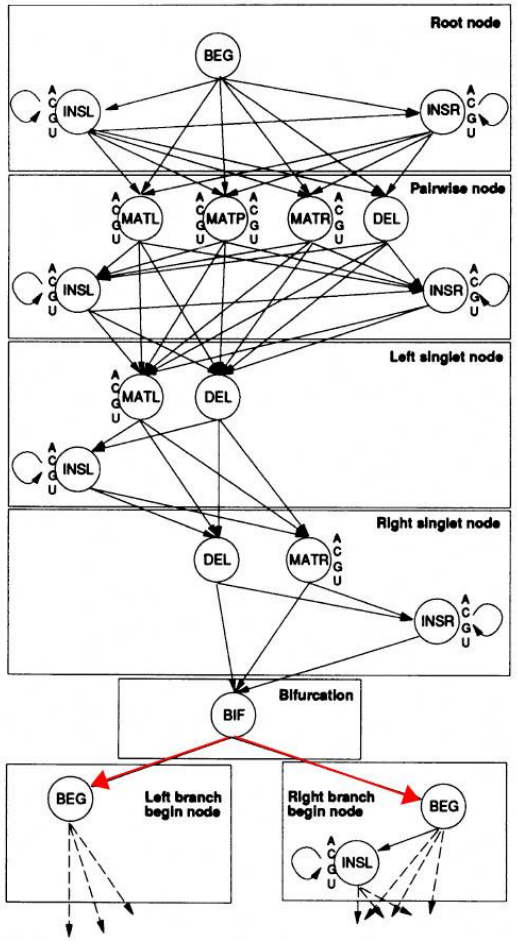

$$S_{i,j,y}(y = DEL) = \max_{y_{next}}[S_{i,j,y_{next}} + \log \mathcal{T}(y_{next} \mid y)]$$

Images—Eddy et al.

# Alignment to CM Algorithm

- For each calculation, take into account:
  - Transition (T) to next state.
  - Emission probability (P) in the state as determined by training data.

- Deletion—does not have emission probability associated with it.
- Bifurcation—does not have state probability associated with it.




$$S_{i,j,v}(y = BIFURC) = \max_{i-1<=mid<=j} [S_{i,mid,y_{left}} + S_{mid+1,j,y_{right}}]$$

Images—Eddy et al.

# Covariance Model Drawbacks

- Needs to be well trained.

- Not suitable for searches of large RNA.
  - Structural complexity of large RNA cannot be modeled
  - Runtime
  - Memory requirements