RNA
Secondary Structures

What is secondary structure?

- Set of canonical $\{AU, CG\}$ basepairs that form via hydrogen bonding when the molecule folds.

  They are called Watson-Crick basepairs.

- Also basepair GU is possible.

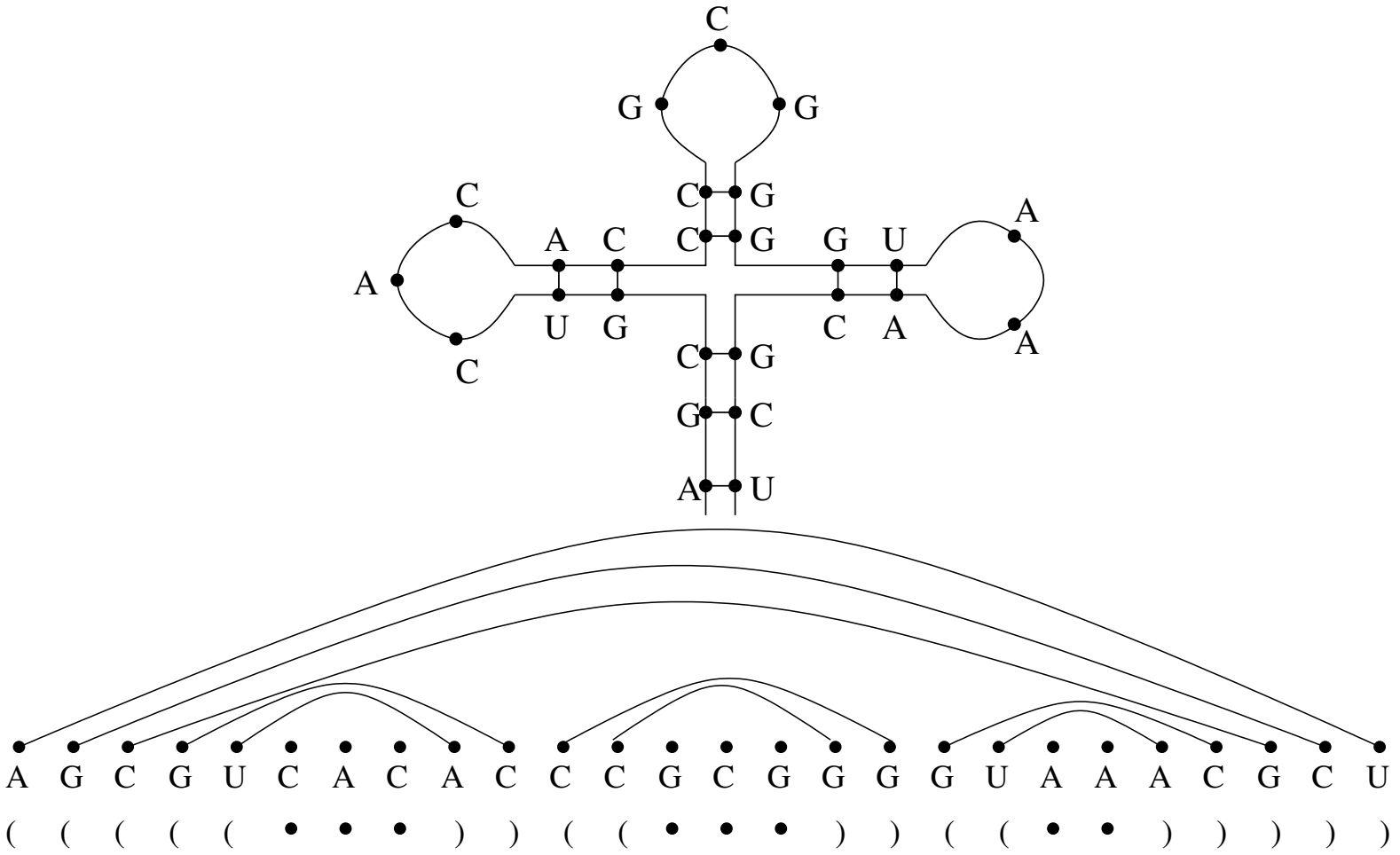
- Each base forms at most one pair

- Depends on temperature, ionic concentration, presence of metabolites, other environmental factors

What is secondary structure?

- There are three possible representations of secondary structure:
  - graphical,
  - dot-bracket,
  - dot-plot

RNA Graphical and Dot-Bracket Representations

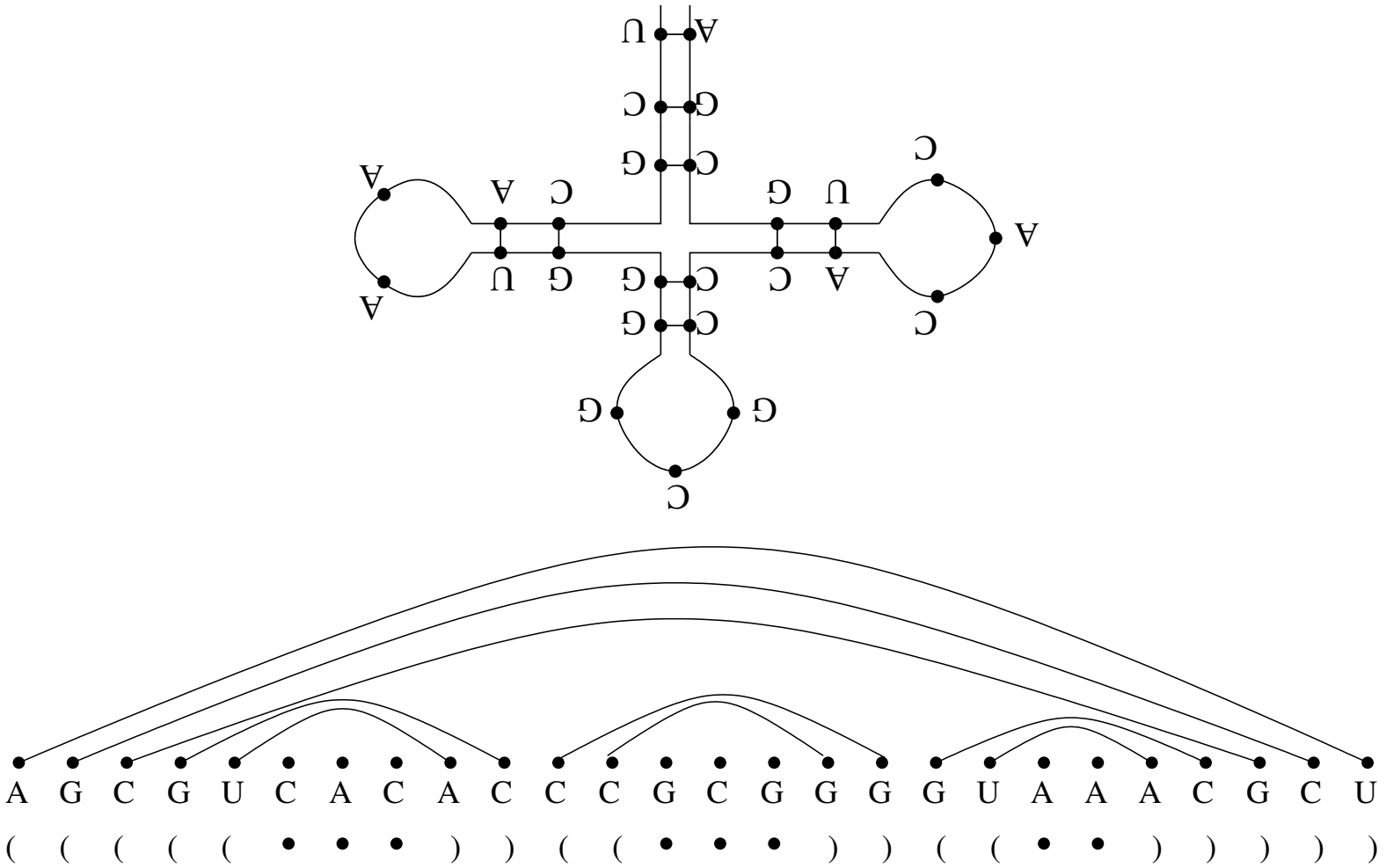


From Graphical to Dot-Bracket

There is a simple way to convert a Graphical representation to a Dot-bracket representation and vice versa.

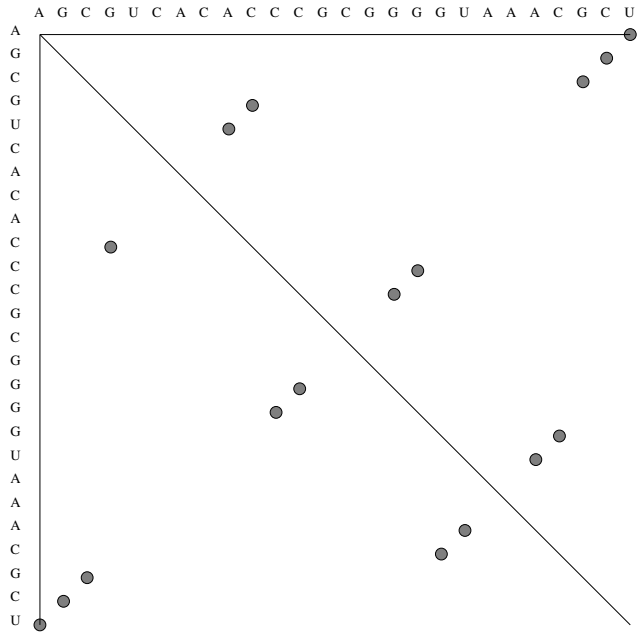
- Think of the links in the graphical representation as being formed from elastic band.
- Stretch the *outer opening*, in this case AU, until the whole RNA strand lies flat on a line.
- Stretch the remaining basepairs accordingly.

Stretching: From Graphical to Dot-Bracket



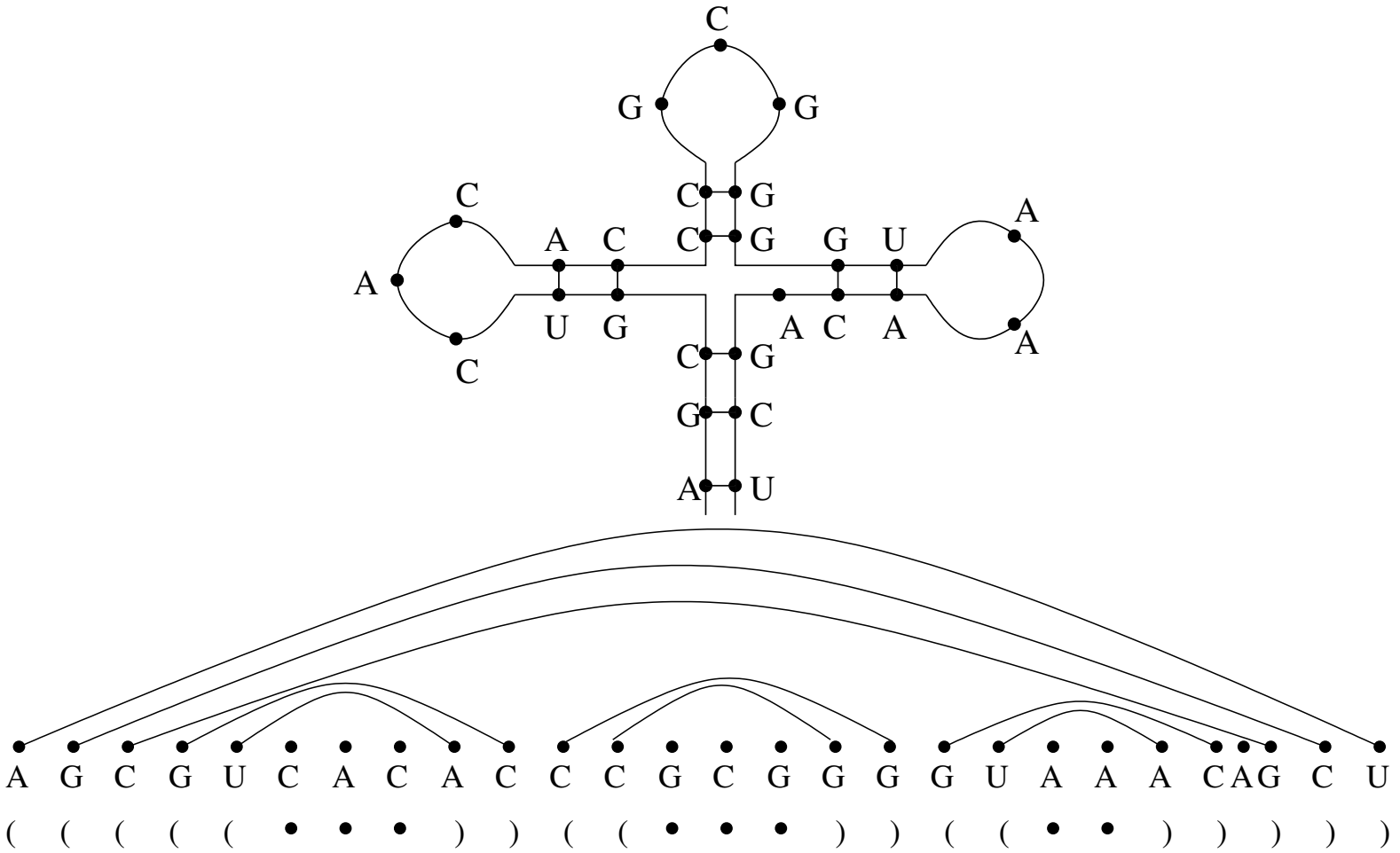
Dot-Plot Representation

List the bases in a column and a row in the order they occur in the RNA string.

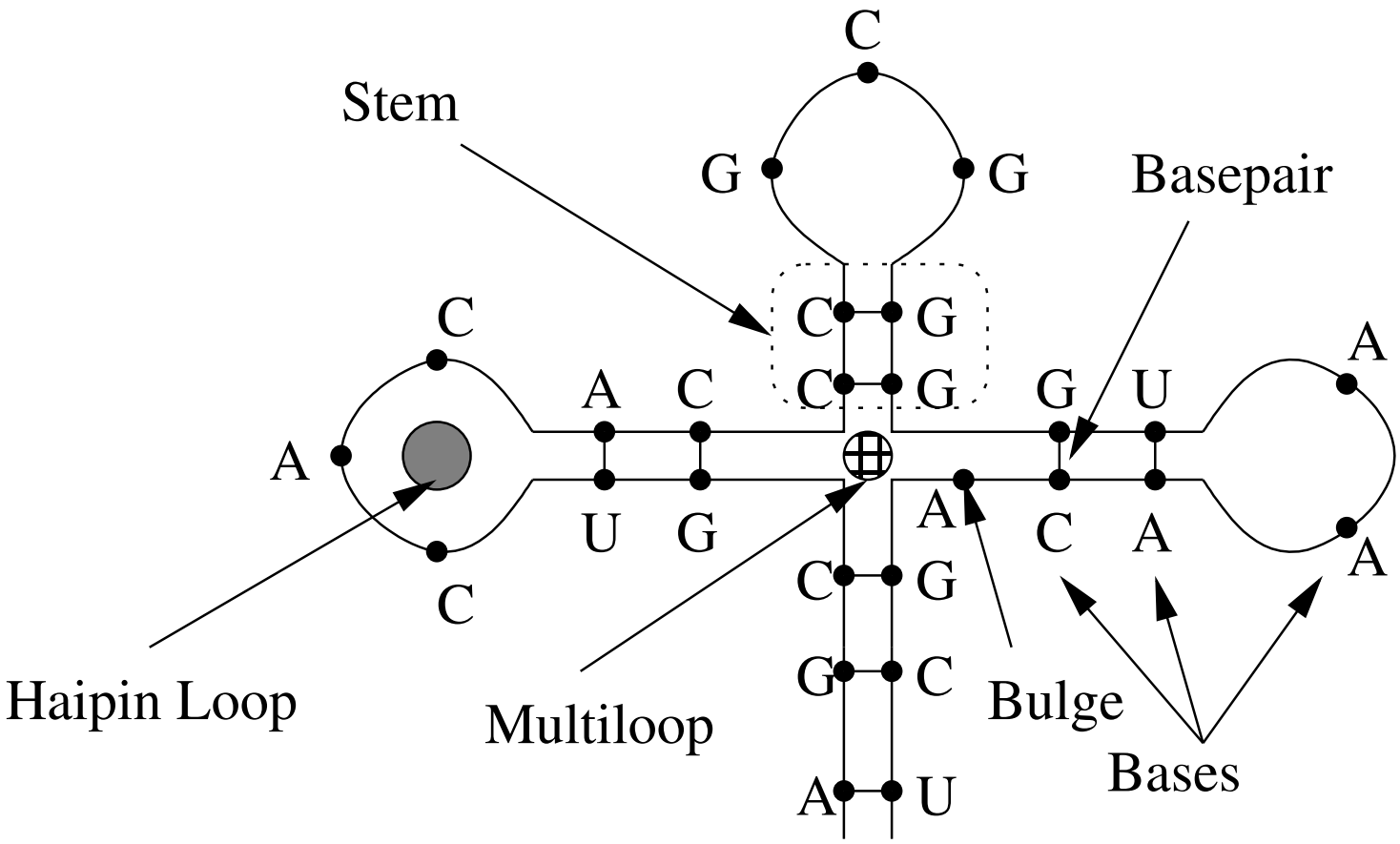


Put a "dot" in positions $(i, j), (j, i)$ if there is a basepair linking bases in positions i and j.

Another Example



Stems, Loops, Multiloops



Internal loops containing unpaired bases are called bulges.

RNA structures are essential for

- transcriptional and post-transcriptional regulation (iron response elements in UTRs of eukaryote transcripts, micro-RNAs from genomic sequences)
- expression of HIV genes (rev-response element, TAR hairpin)
- mediation of insertion of selenocysteine (RNA structural element prevents translation termination at a UGA codon and instead inserts selenocysteine)
- splicing
- perhaps helps explain 5% of highly-conserved non-genic sequence observed in vertebrates?

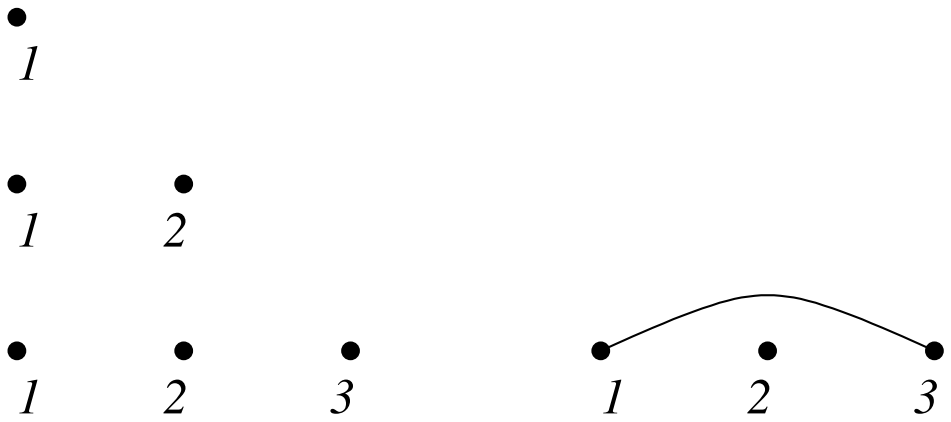
Why Study RNA Structure?

Tools to prediction of RNA structure help us

- gain insight on the genome
- shed insight on RNA 3D structure and ultimately function
- better align RNA sequences
- establish phylogenetic relationships among organisms
- design good microarray probes, or RNA molecules for disease therapy

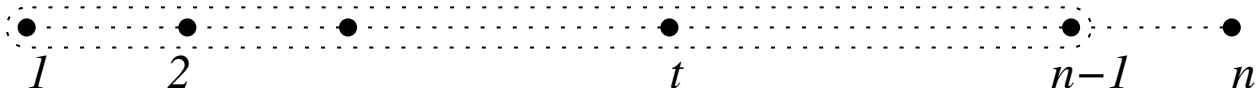
How Many Secondary Structures?

- Forget about A, U, C, G: and number the bases $1 \ldots n$.
- Let $S(n)$ be the number of secondary structures for the sequence $1 \ldots n$.
- What is $S(n)$? Well, $S(0) = 0$.
- From the picture below: $S(1) = S(2) = 1$ and $S(3) = 2$.

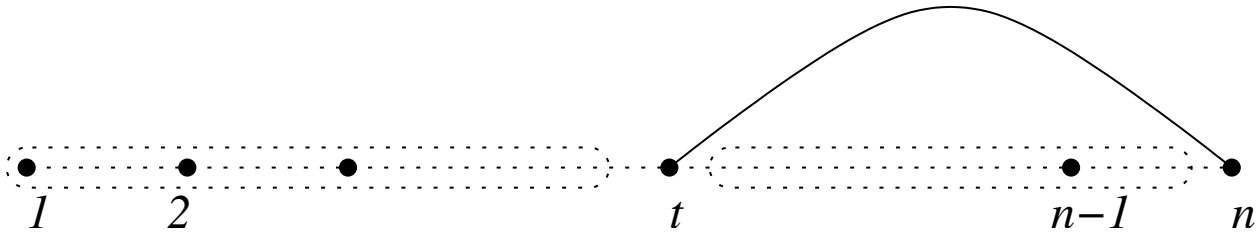


A Recursive Assumption

- Assume that we know $S(k)$, for all $k < n$. Can we compute $S(n)$? There are two cases.
- Either n is not paired with any other element, in which case we count $S(n-1)$ secondary structures.



- Or else n is paired with some other element $t < n$,



and therefore secondary structures are formed in $[1, t-1]$ and $[t+1, n-1]$.

A Recursive Equation

It follows that $S(n)$ satisfies the recursive equation

$$\begin{aligned} S(n) &= S(n-1) + S(n-2) + S(n-3)S(1) + \cdots + S(n-3)S(1) \\ &= S(n-1) + S(n-2) + \sum_{t=2}^{n-2} S(t-1)S(n-1-t) \end{aligned}$$

- How do you solve this equation and determine $S(n)$?
- Method uses generating functions! Think of $S(n)$ as the coefficients of a continuous function $f(x)$

$$y := f(x) = \sum_{n=1}^{\infty} S(n)x^n.$$

- Can you find a *functional* equation satisfied by $f(x)$?

A Functional Equation

Abbreviate $a_n := S(n)$. Recall that $a_0 = 0, a_1 = a_2 = 1$. We have shown that

$$a_n = a_{n-1} + a_{n-2} + \sum_{t=2}^{n-2} a_{t-1} a_{n-1-t}$$

Multiply both sides of equation by x^n to obtain

$$a_n x^n = x a_{n-1} x^{n-1} + x^2 a_{n-2} x^{n-2} + x^2 \sum_{t=2}^{n-2} x^{t-1} a_{t-1} x^{n-t-1} a_{n-1-t}$$

and take sums of both sides from $n = 2$ to ∞ and you derive the following equation $y - x = xy + x^2 y + x^2 y^2$, which implies that

$$x^2 y^2 + (x^2 + x - 1)y + x = 0$$

Asymptotic Formula

If we define $F(x,y) := x^2y^2 + (x^2+x-1)y + x$ then it follows from Bender's theorem that if (r,s) is the unique solution of the system

$$\begin{aligned} F(r,s) &= r^2s^2 + (r^2+r-1)s + r = 0 \quad (1) \\ \frac{\partial F}{\partial y}(r,s) &= 2r^2s + r^2 + r - 1 = 0 \quad (2) \end{aligned}$$

then

$$S(n) \sim \sqrt{\frac{rF_x(r,s)}{2\pi F_{yy}(r,s)}} n^{-3/2} r^{-n}. \quad (3)$$

It follows that

$$S(n) \sim \sqrt{\frac{15+7\sqrt{5}}{8\pi}} n^{-3/2} \left(\frac{3+\sqrt{5}}{2}\right)^n.$$

Secondary Structures with Exactly k Basepairs

Define $S_{n,k}$ as the set of secondary strucrures on $[1, n]$ with exactly k basepairs.

Let $S(n, k)$ number of secondary strucrures on $[1, n]$ with exactly k basepairs. So $S(n, k) = |S_{n,k}|$.

Clearly,

$$S(n) = \sum_{k=0}^{\lfloor n/2 \rfloor} S(n, k)$$

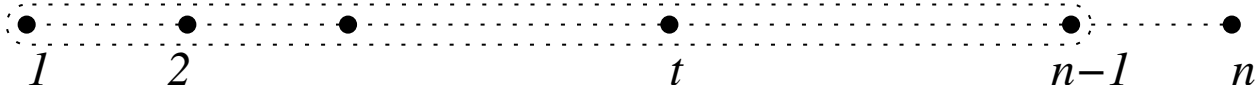
and it is easy to show as before that

$$S(n, k) = S(n-1, k) + \sum_{j=1}^{n-2} \sum_{i=0}^{k-1} S(j-1, i) S(n-1-j, k-1-i)$$

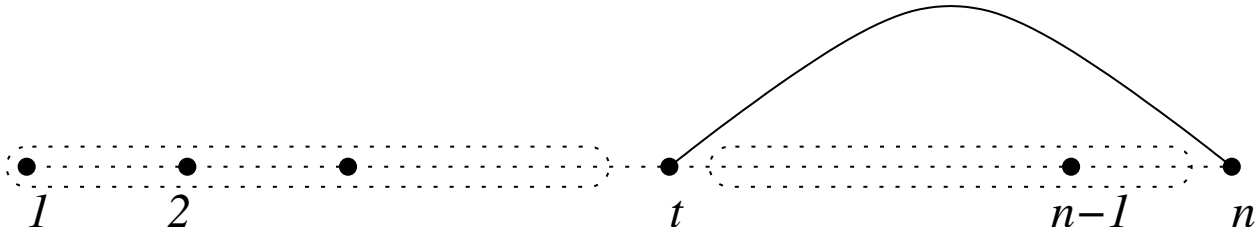
Can we compute $S(n, k)$?

Recursion

- Either n is not paired with any other element, in which case we count $S(n-1,k)$ secondary structures.



- Or else n is paired with some other element $t < n$. Remove this basepair and you have $k-1$ basepairs left.



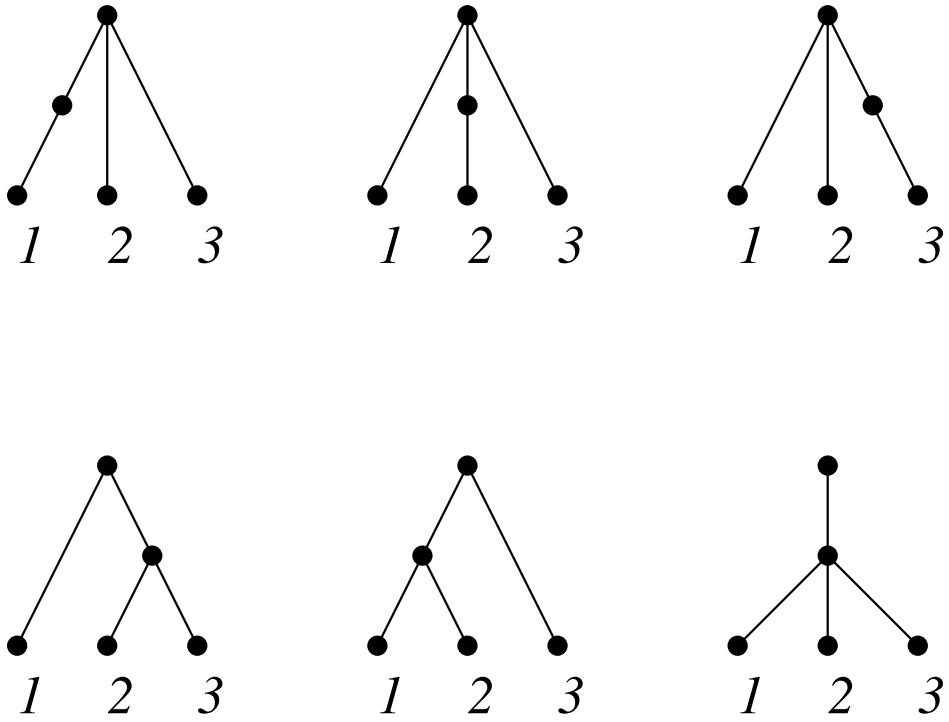
Then for some $i \leq k-1$, i basepairs are formed in $[1, t-1]$ and the remaining $k-1-i$ basepairs in $[t+1, n-1]$.

Equivalence of Trees and Secondary Structures

- A *linear tree* is a rooted tree along with a linear order on the set of children of each vertex.
- Let $T_{n,k}$ the set of unlabeled linear trees with n vertices and $n-k$ leaves.
- Let $T(n,k) := |T_{n,k}|$.

Example: The six trees of $T_{5,3}$

Example: The six trees in $T_{5,3}$



Poincare Duality: Trees and Secondary Structures

There is a bijection

$$S_{n+k-2,k-1} \rightarrow T_{n,k}.$$

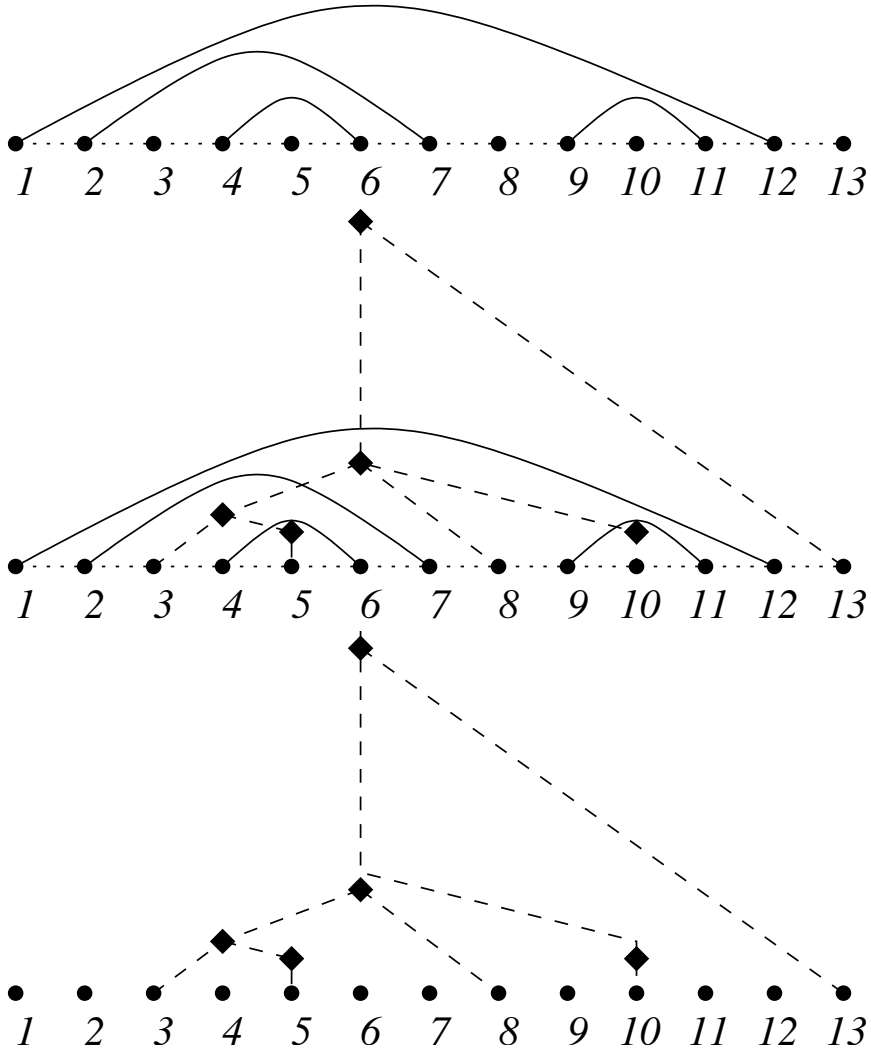
The algorithm is as follows:

1. Take a member of $S_{n+k-2,k-1}$ in loop form.
2. Put a node (the root) of the tree above the figure outside all loops.
3. Insert a node inside all loops visible from this node and connect them all to this node.
4. Iterate recursively.

Hence,

$$S(n+k-2,k-1) = T(n,k).$$

Example: Equivalence of Trees and Secondary Structures



Number of Trees

- It can be shown that

$$T(n,k) = \frac{1}{k-1}\binom{n-1}{k}\binom{n-2}{k-2}$$

- A well-known argument is being used here (See L. Lovasz, Comb. Prob. and Exercises, NH, 1979, 4.1 and 4.8): Consider n points $u_1, \ldots, u_n$ and n integers $d_1, \ldots, d_n$ such that $d_1 + \cdots d_n = 2n - 2$. The number of trees on points $u_1, \ldots, u_n$ in which u_i has degree d_i is given by the formula

$$\frac{(n-2)!}{(d_1-1)! \cdots (d_n-1)!}$$

- Observe that leaves have degree 1.
- Details of the rest of the proof of this are beyond our scope.

Number of Secondary Strauctures

- At least we can use this last formula to derive the number of secondary structures with a given number os basepairs.
- Using the previous bijection

$$\begin{aligned} S(n,k) &= T(n-k+1,k+1) \\ &= \frac{1}{k}\binom{n-k}{k+1}\binom{n-k+1}{k-1}. \end{aligned}$$

The reality is far more more complex: individual bases are "linked" with a certain energy!

MFE (minimum free energy) approach

Used to predict secondary structure.

- Hypothesis: an RNA molecule will fold into that secondary structure that minimizes its free energy
- Free energy of a structure (at fixed temperature, ionic concentration) is sum of loop energies
- Tables of loop energies are used to calculate energy of a structure

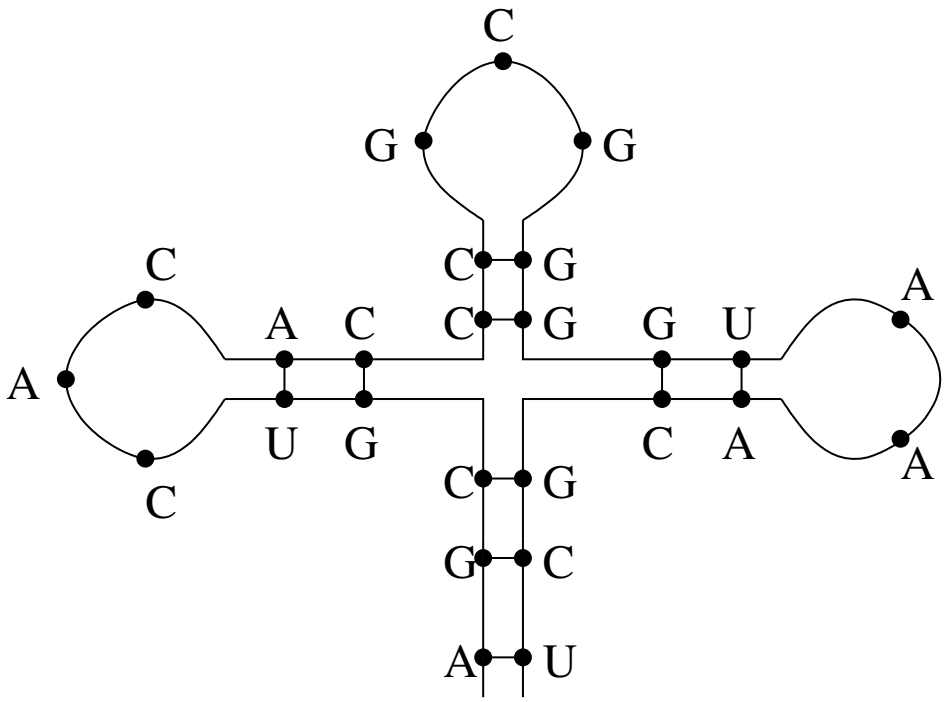
Given an energy table what is the secondary structure with minimum free energy?

Minimum Free Energy

- Given an energy table, E.g.,

Basepair	-1
Internal Loop	$+1.1$

- Is this an MFE secondary structure?



Naive Algorithm

- Naive Algorithm:
  1. Enumerate all possible secondary structures;
  2. Calculate energy of each;
  3. Output that structure which has lowest energy
- Problem: many structures to enumerate! A 50mer could have more than 5000 billion structures
- DP (dynamic programming) algorithm: avoids this problem, but minimizes over restricted structure types

Ruth Nussinov and Ann Jacobson, 1980

- One of the first beautiful ideas in CMB!
- Based on the

  "more is less" principle: by calculating more than you need, less work is needed overall

- Construct mfe structure for whole strand from mfe structures for substrands

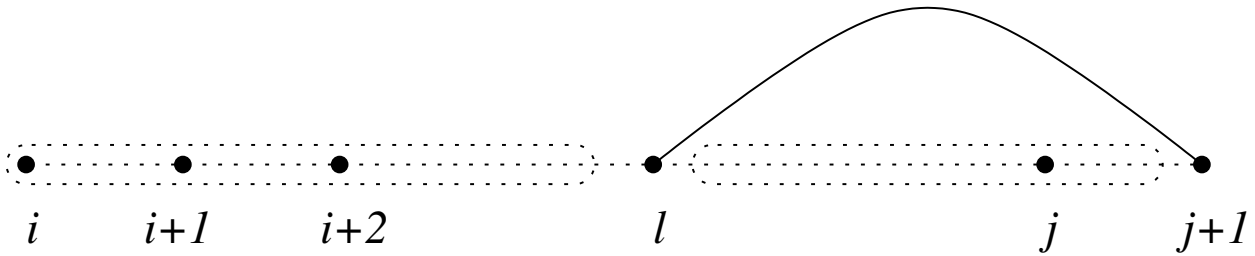
Minimum Free Energy

- Define

$$\rho(a,b) = \begin{cases} 1 & \text{if } a,b \text{ can basepair} \\ 0 & \text{otherwise} \end{cases}$$

- Given a sequence $a_1 a_2 \cdots a_n$ in $\{A,U,C,G\}^n$ let $X_{i,j}$ be the max number of basepairs in $a_i a_{i+1} \cdots a_j$.
- Observe that $X_{i,j+1}$ is the maximum of $X_{i,j}$ and

$$\max\{(X_{i,l-1} + 1 + X_{l+1,j})\rho(a_l, a_{j+1}) : 1 \leq l \leq j-1\}$$



- Time complexity is $\sum_{i<j\leq n}(j-i) \in O(n^3)$.

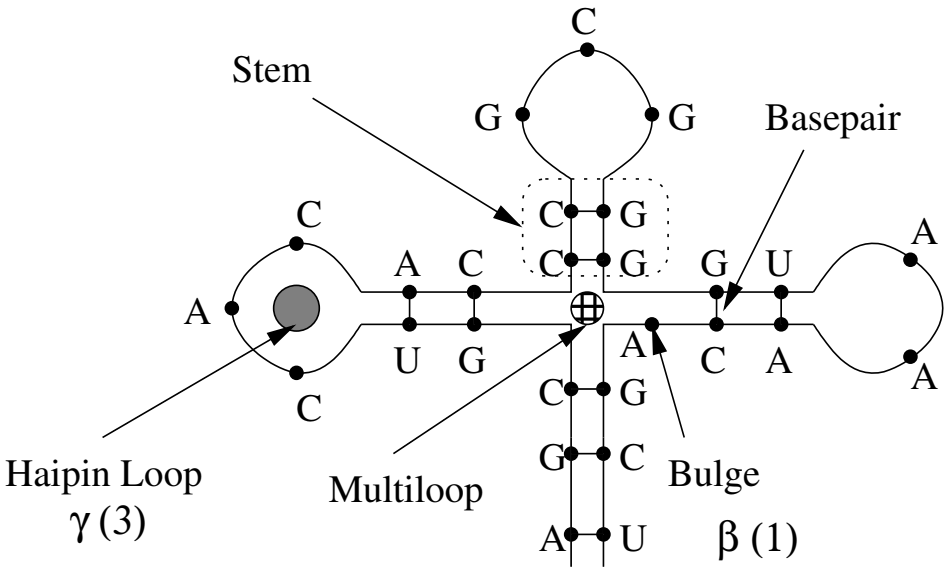
Other Energy Contributions

Problem is in fact much more complex.

Other energy functions contribute to the free energy of $a_1 a_2 \cdots a_n$.

- $\alpha(a, b)$ = free energy of basepair $\{a, b\}$
- η = stacking energy of adjacent basepairs
- Destabilization energies
  - $\xi(k)$ = destabilization free-energy of an end loop of k bases
  - $\beta(k)$ = destabilization free-energy of bulge of k bases
  - $\gamma(k)$ = destabilization free-energy of an interior loop of k bases

Example: Other Energy Contributions



- $\xi(k)$ = destabilization free-energy of an end loop of k bases
- $\beta(k)$ = destabilization free-energy of bulge of k bases
- $\gamma(k)$ = destabilization free-energy of an interior loop of k bases

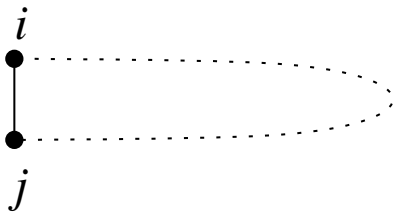
MFE for Hairpin Loops

$H_{i,j}$ is min free energy single hairpin structure on $a_i a_{i+1} \cdots a_j$, for $i < j$, where a_i and a_j basepair and there is a single end loop.

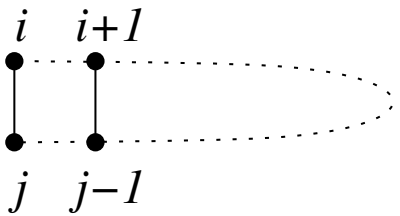
If a_i and a_j cannot basepair set $H_{i,j} = \infty$.

$H_{i,j}$ is minimum of five quantities.

(a) End Loop: $\alpha(a_i, a_j) + \xi(j - i + 1)$

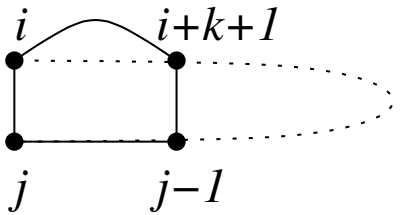


(b) Helix Extension (stacking bps): $\alpha(a_i, a_j) + \eta + H_{i+1,j-1}$

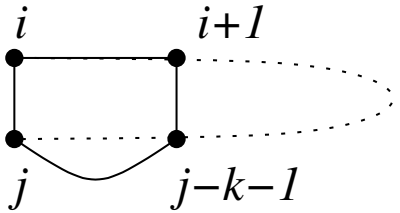


MFE for Hairpin Loops

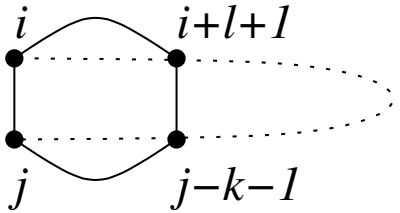
(c) Bulge: $\min_{k \geq 1}\{\alpha(a_i, a_j) + \beta(k) + H_{i+k+1,j-1}\}$



(d) Bulge: $\min_{k \geq 1}\{\alpha(a_i, a_j) + \beta(k) + H_{i+1,j-k-1}\}$



(e) Interior Loop: $\min_{l,k \geq 1}\{\alpha(a_i, a_j) + \gamma(l, k) + H_{i+l+1,j-k-1}\}$



Computation Time

Time Complexity is $O(n^4)$. Why?

Take each of the five steps previously described.

- Steps (a) & (b):

$$\sum_{1 \le i < j \le n} 1 \in O(n^2)$$

- Steps (c) & (d):

$$\sum_{1 \le i < j \le n} (j - i) \in O(n^3)$$

- Step (e):

$$\sum_{1 \le i < j \le n} \left(\sum_{i' \le i < j \le j'} 1 \right) \in O(n^4)$$

Dynamic Programming

- Construct a matrix $(H_{i,j})$:

	a_n	a_{n-1}	$\cdots$	a_2	a_1
a_1	$H_{1,n}$	$H_{1,n-1}$	$\cdots$	$H_{1,2}$	$H_{1,1}$
a_2	$H_{2,n}$	$H_{2,n-1}$	$\cdots$	$H_{2,2}$	
$\vdots$	$\vdots$	$\vdots$	$\cdots$		
a_{n-1}	$H_{n-1,n}$	$H_{n-1,n-1}$			
a_n	$H_{n,n}$				

- If a_i and a_j cannot basepair set $H_{i,j} = \infty$.
- If a_i and a_j can basepair and $j - i - 1 \geq m$ (m is the min endloop size) $H_{i,j}$ to the value computed before.

Dynamic Programming: Reducing Computation Time

Note that interior loops must be of size $\geq m$, for some value m.

Now $H_{i,j}$ results from the five situations:

- **End Loop:** $\alpha(a_i, a_j) + \xi(j - i + 1)$
- **Helix Extension:** $\alpha(a_i, a_j) + \eta + H_{i+1,j-1}$
- **Bulge:** $\min_{k \geq 1}\{\alpha(a_i, a_j) + \beta(k) + H_{i+k+1,j-1}\}$
- **Bulge:** $\min_{k \geq 1}\{\alpha(a_i, a_j) + \beta(k) + H_{i+1,j-k-1}\}$
- **Interior Loop:** $\min_{l,k \geq 1}\{\alpha(a_i, a_j) + \gamma(l, k) + H_{i+l+1,j-k-1}\}$

	j	$j-1$	$j-2$	$\cdots$
i	α			
$i+1$		η	β	
$i+2$		β	γ	
$\vdots$				

Reducing Computation Time to $O(n^3)$

- Given a pair (i, j), consider the set of candidate positions $Cd(i, j) = \{(k, l) : l - k - 1 \geq m, k \geq i + 2, j - 2 \geq l\}$
- The interior loop has size $s = (j - i - 1) - (l - k + 1) = (j - i) - (l - k) - 2$
- Along lines such that $l - k = constant$ the interior loop *destabilization* function $\gamma(s)$ is constant.
- For each pair (i, j) store the values $H^*_{i,j}(s) := \min\{H_{k,l} : (k, l) \in Cd(i, j) \& s = (j - i) - (l - k) - 2\}$
- When moving from $j - i = c$ to $j - i = c + 1$ each vector can be updated in time $O(n)$.
- Best interior loop: $\min\{\alpha(a_i, b_j) + \gamma((j - i) - (k - l)) + H^*_{i,j}(s)\}$