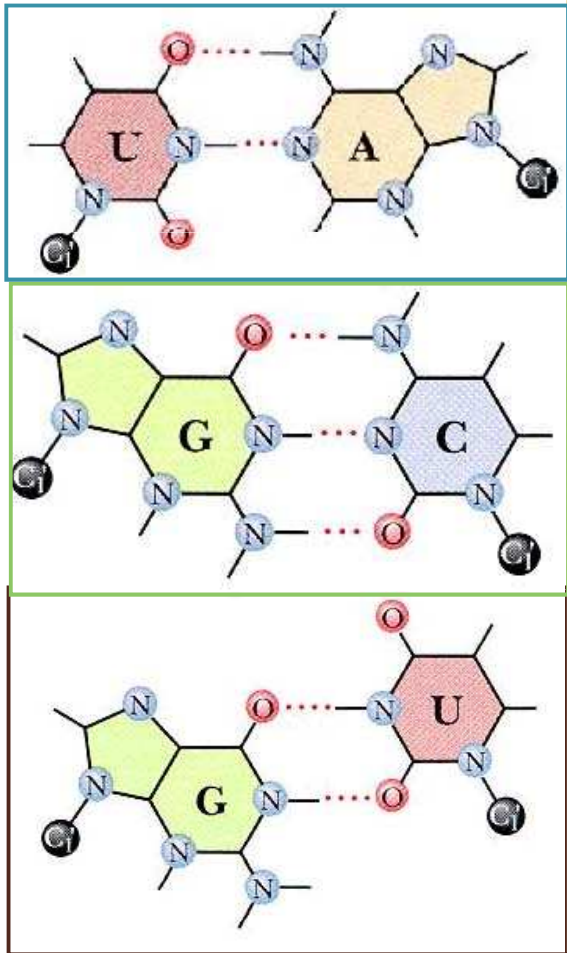
Estructura de RNA

Darío Canales
Ignacio Salas
Oscar Yañez

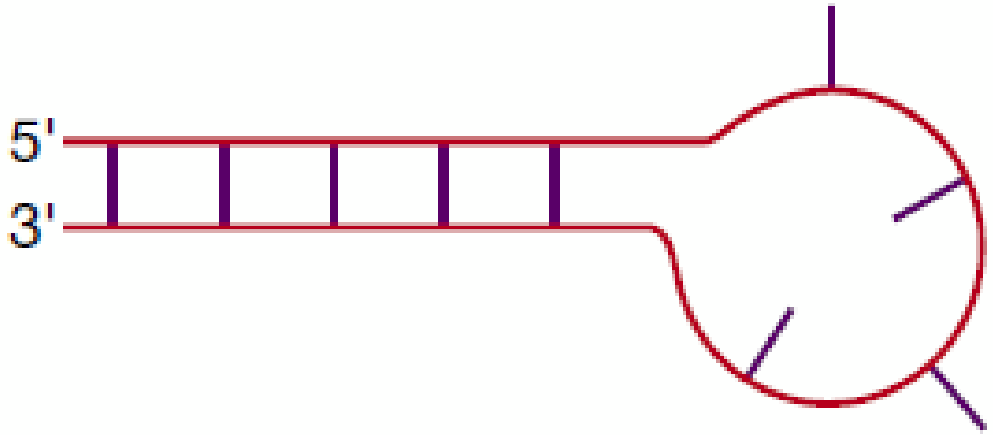
RNA

- Está presente tanto en las células procariotas como en las eucariotas.
- Pares de bases canónicas
  - A/U
  - G/C
  - G/U (par inestable)
- Algunas funciones:
  - Dirige la parte intermedia de la síntesis de proteínas
  - Catálisis
  - Regulan la expresión genética



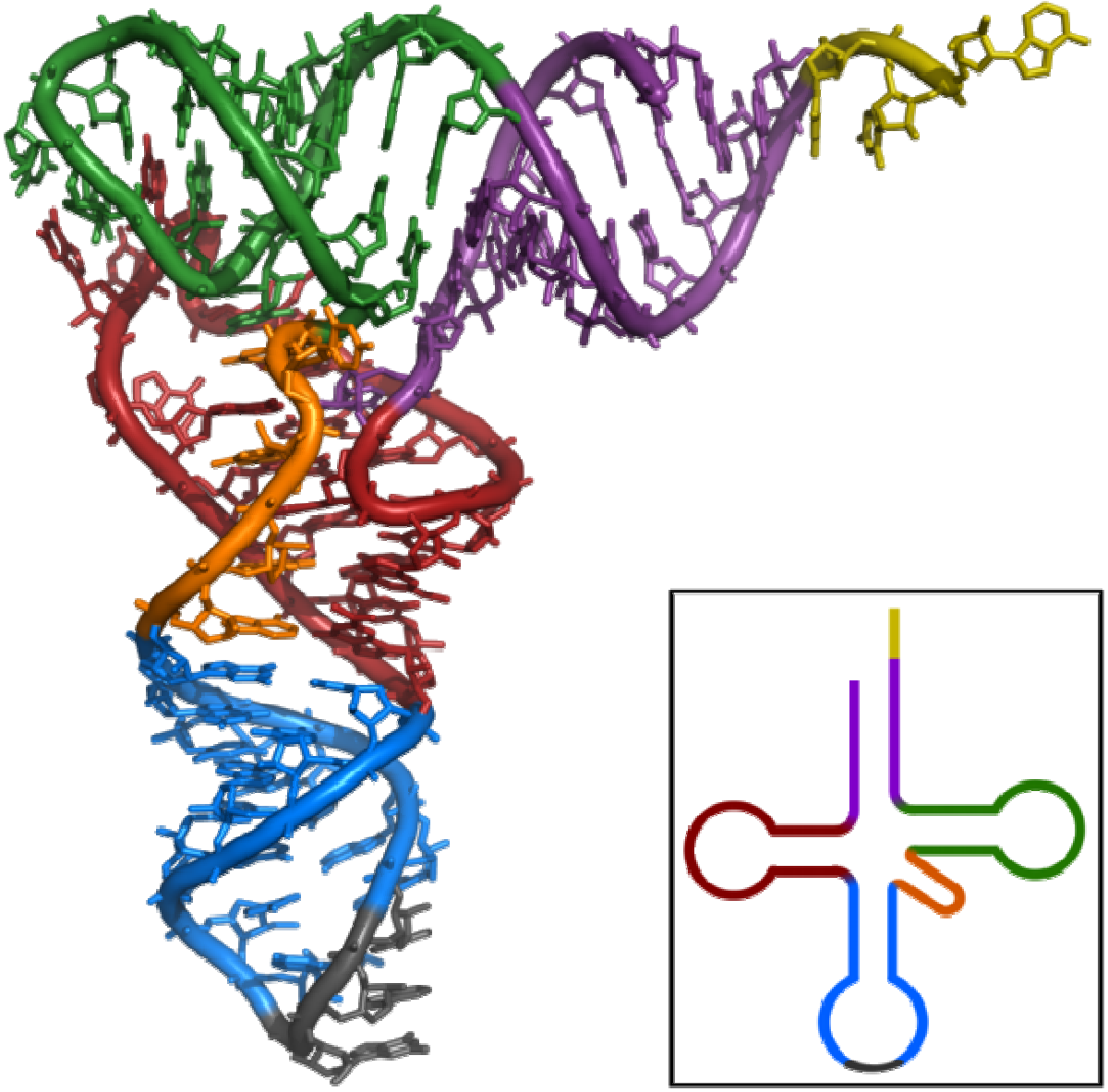
Estructura secundaria del RNA

- Se puede ver como un paso intermedio hacia una estructura en 3D
- Doble cadena de RNA formada por el plegamiento de una molécula simple



Estructura terciaria del RNA

- Hace referencia a la organización que tienen los elementos de la estructura secundaria en el espacio.
- Es el resultado del apilamiento de bases y de los enlaces por puente de hidrógeno entre diferentes partes de la molécula.

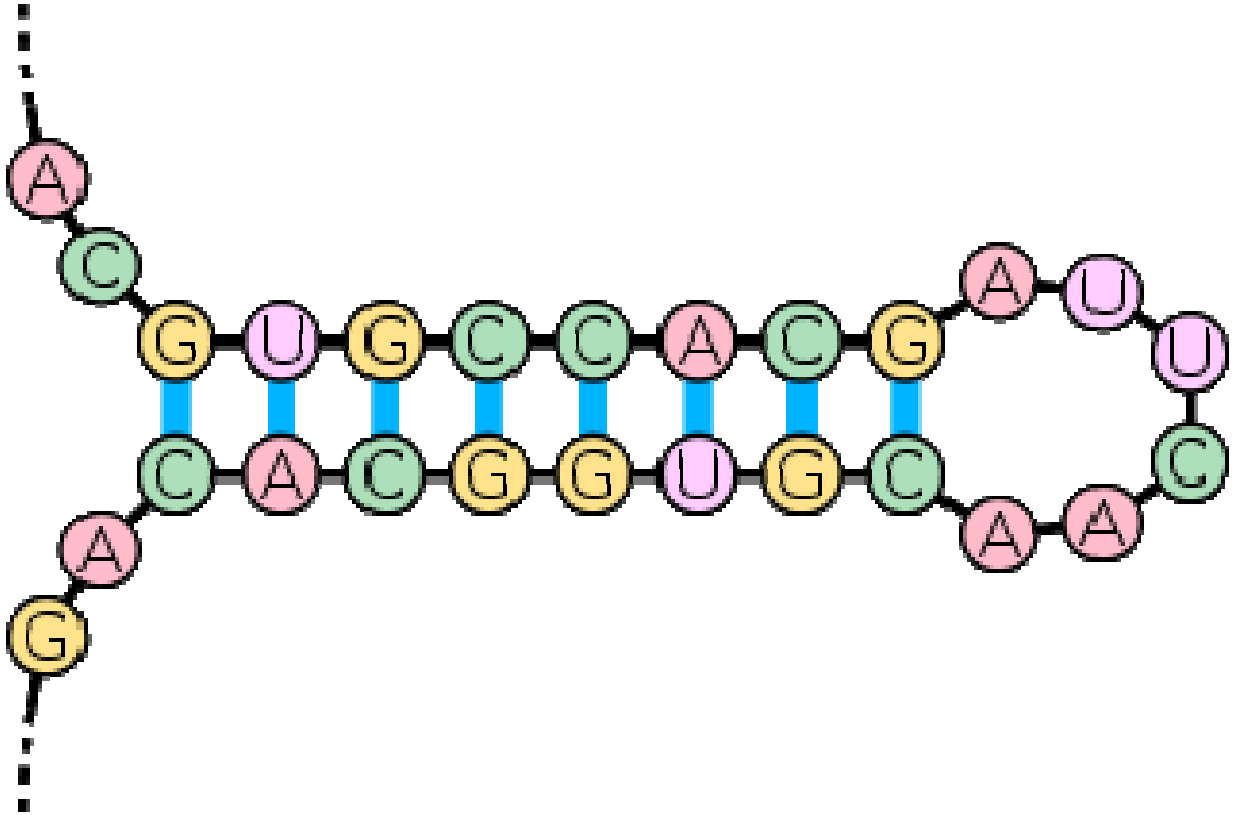


Predicción de la estructura del RNA

- El modelado de la estructura de RNA en 2D sirve para la construcción de bases de datos de moléculas de RNA
  - Formas similares, roles similares
- Muchos métodos usan programación dinámica
- La molécula se pliega sobre si misma

Predicción de la estructura de RNA

- Se buscan palíndromos
- Estructuras energéticamente estables

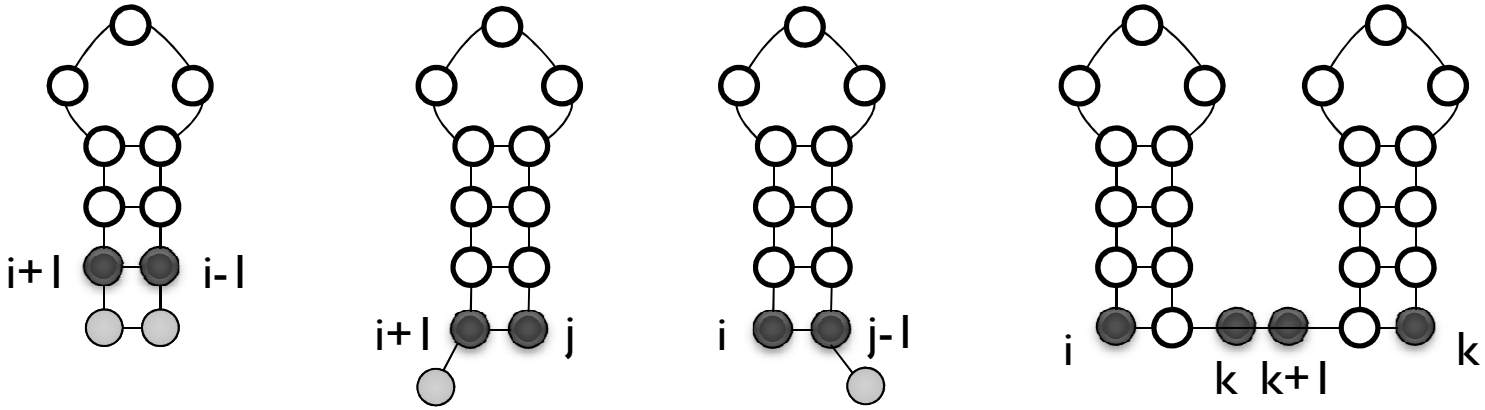


Algoritmo de Nussinov

Objetivo Hallar las estructura secundaria que maximice el numero de bases pareadas.

Algoritmo Recursivo Encuentra la mejor estructura para los inputs i,j intentando una de las siguientes 4 posibilidades:

- Agregar el par i, j sobre la mejor estructura i+1,j-1
- Agregar i sin aparear a la mejor estructura i+1,j
- Agregar j sin aparear a la mejor estructura i,j-1
- Combinar las dos estructuras optimas i,k y k+1,j



par i,j | i no pareada | j no pareada | bifurcación

Algoritmo de Nussinov

Sea S una secuencia de RNA de tamaño L.
Se define una matriz $\gamma_{L \times L}$ de folding correspondiente a la secuencia S.
Se establece una función $\delta(x_i, x_j) = 1$, si x_i y x_j se pueden parear y $\delta(x_i, x_j) = 0$, en caso contrario.

Inicialización:

$$\gamma(i, i-1) = 0, \qquad i = 2 \ldots L$$
$$\gamma(i, i) = 0, \qquad i = 1 \ldots L$$

Llenado:

For i=1...L-1, j=i+1...L

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1, j) \\ \gamma(i, j-1) \\ \gamma(i+1, j-1) + \delta(x_i, x_j) \\ \max\limits_{i<k<j} \left[\gamma(i,k) + \gamma(k+1, j) \right] \end{cases}$$

Traceback

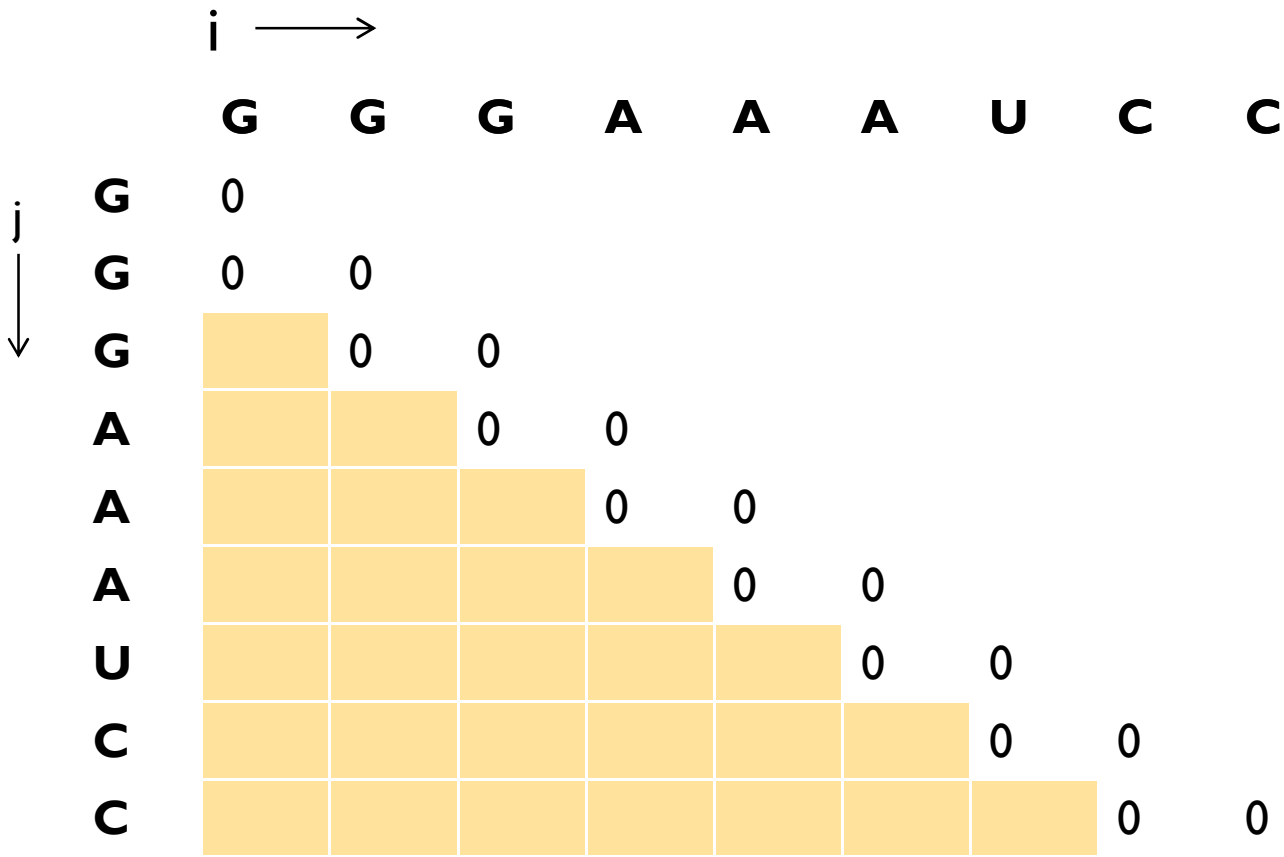
Inicialización: Push (1,L) en el stack
Recursión: Repetir hasta que el stack este vacío

pop(i,j)
if i > j continuar
else if $\gamma(i+1, j) = \gamma(i, j)$ *push (i+1, j)*
else if $\gamma(i, j-1) = \gamma(i, j)$ *push (i, j-1)*
else if $\gamma(i+1, j-1) + \delta ij = \gamma(i, j)$*:*
    registrar i, j como apareamiento
    push (i+1, j-1)
else for k= i+1 to j-1: if $\gamma(i,k) + \gamma(k+1,j) = \gamma(i,j)$*:*
    push (k+1,j)
    push (i,k)
    break

Ejemplo

Inicialización:

$$\gamma(i, i-1) = 0, \qquad i = 2 \ldots L$$
$$\gamma(i, i) = 0, \qquad i = 1 \ldots L$$



Ejemplo

Llenado: For i=1...L-1, j=i+1...L

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

i ⟶

j ↓

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

Ejemplo

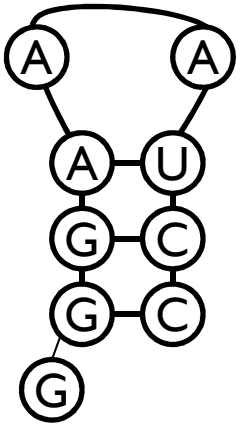
Traceback:

pop(i,j)
if i > j continuar
else if $\gamma(i+1, j) = \gamma(i, j)$ push (i+1, j)
else if $\gamma(i, j-1) = \gamma(i, j)$ push (i, j-1)
else if $\gamma(i+1, j-1) + \delta ij = \gamma(i, j)$:
registrar i, j como apareamiento
push (i+1, j-1)
else for k= i+1 to j-1: if $\gamma(i,k) + \gamma(k+1,j) = \gamma(i,j)$:
push (k+1,j)
push (i,k); break

i →

j ↓

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



Desventajas de Nussinov

- Este método asume que la mejor estructura es la que contiene una mayor cantidad de bases pareadas.
- No necesariamente entrega la estructura más estable.
- Hay muchas combinaciones posibles de apareamiento, pero Nussinov detecta generalmente sólo una variante.
- No se considera apilamiento de bases pareadas => Diferencias en estructura y estabilidad de hélices.
- No se considera tamaño de loops internos.

Solución: Minimizar energía libre

- Por Ley de la termodinámica, sólo una estructura secundaria es estable, normalmente, la que optimice la energía libre.

- Energía libre

$$G = H - TS$$

G: Energía Libre
H: Entalpía (energía que puede generar trabajo)
T: Temperatura Absoluta (Kelvin)
S: Entropía (medida de desorden)

- **Entalpía:** por apareamiento de bases
- **Entropía:** desorden en regiones no apareadas

Solución: Minimizar energía libre

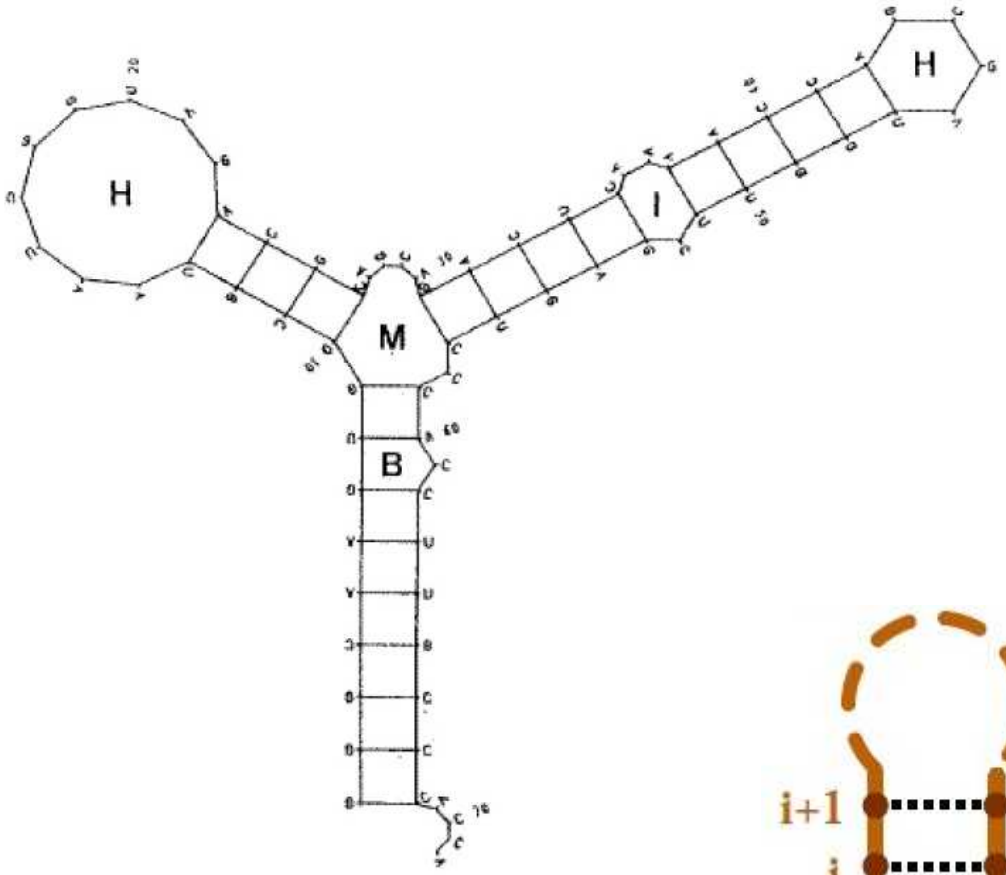
- Usualmente, se mide la diferencia

$$\Delta G = \Delta H - T\Delta S$$

- Lo que se mide → energía en loops, stacks y otros elementos de estructura secundaria más pequeños
- Energía libre total = sumatoria

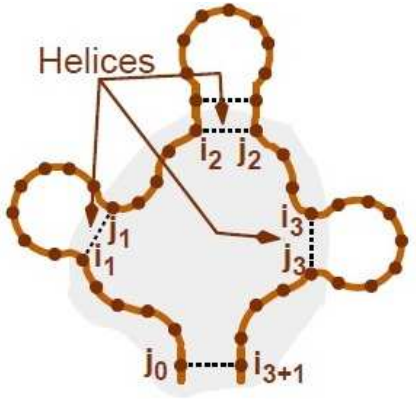
Algoritmo de Zuker: Definiciones

Tipos de ciclos



- H - Hairpin Loop
- I – Interior Loop
- B – Bulge
- M – Multiple Loop
  o Multiloop

Stacking



Multiloop

Algoritmo de Zuker: Definiciones

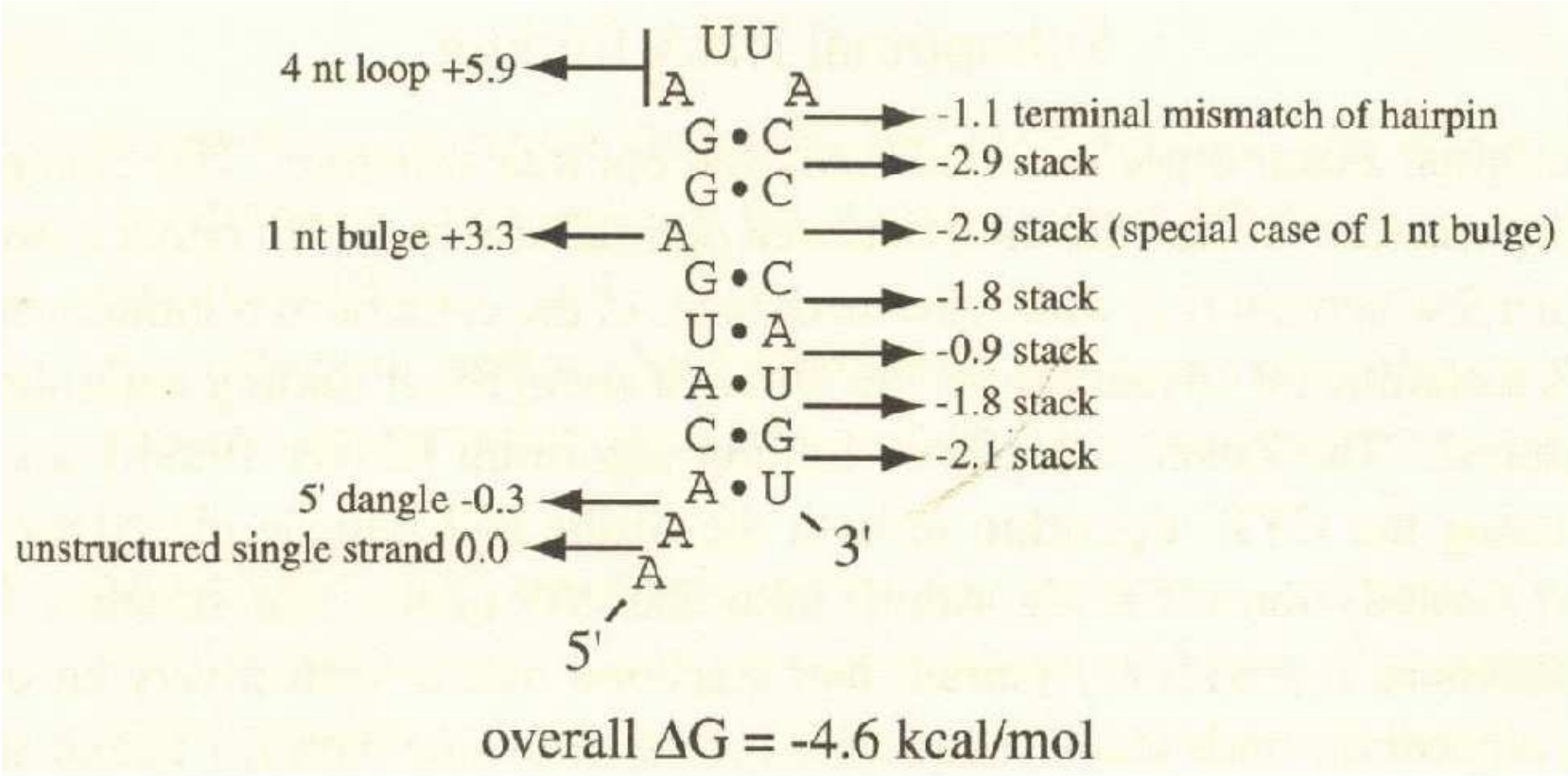
Contribuciones de energía

- Cada tipo de estructura (ciclo) tiene una contribución de energía diferente.

hairpin loop (i,j):	$eH(i,j)$
stacking (i,j):	$eS(i,j,i+1,j-1)$
internal loop (i,j,i,j'):	$eL(i,j,i',j')$
multiloop:	$eM(j_0,i_1,j_1,\ldots,i_k,j_k,i_{k+1})$

  - El cálculo de la contribución del multiloop es muy costoso.
  - Se usa una versión simplificada de la contribución, que considera *contribución del cierre del loop*, *número de hélices* y *número de bases no pareadas dentro del loop*.

Ejemplo


4 nt loop +5.9
U U
A A
-1.1 terminal mismatch of hairpin
G • C
-2.9 stack
G • C
1 nt bulge +3.3
A
-2.9 stack (special case of 1 nt bulge)
G • C
-1.8 stack
U • A
-0.9 stack
A • U
-1.8 stack
C • G
-2.1 stack
5' dangle -0.3
A • U
unstructured single strand 0.0
A
A
3'
5'
overall ΔG = -4.6 kcal/mol

Problema de minimización de energía libre de Zuker

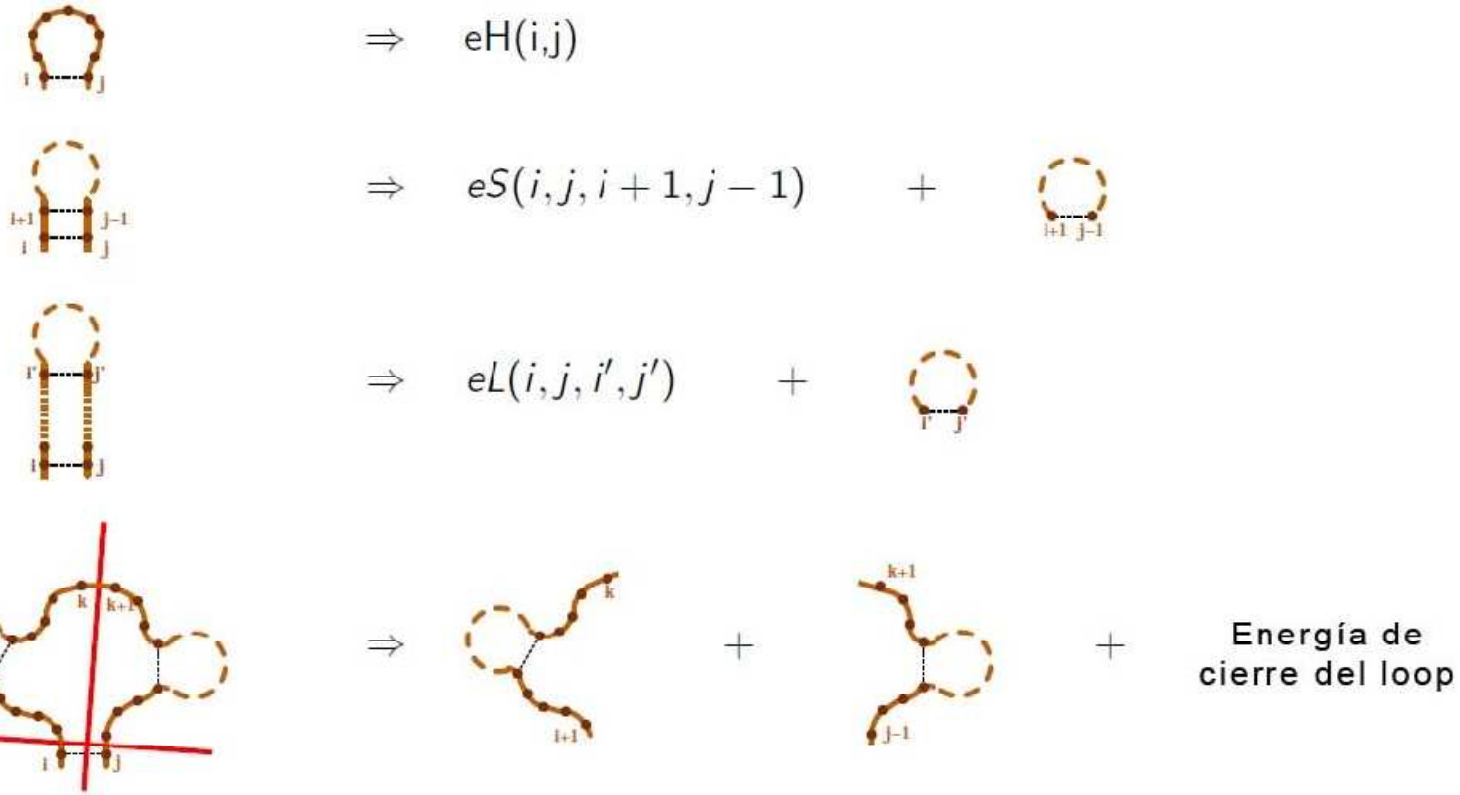
- INPUT: secuencia S de RNA
- OUTPUT: estructura secundaria RNA de S de mínima energía libre

- Hace una comparación entre las posibles estructuras secundarias obtenidas.

- Solución eficiente → Programación Dinámica mediante ecuaciones recursivas

Problema de minimización de energía libre de Zuker

Se determina la energía asociada dividiendo la estructura resultante.



Matriz W

$$W(i,j) = min \left\{ \begin{array}{c} W(i+1,j) \\ W(i,j-1) \\ V(i,j) \\ \text{mín}_{i<k<j}[W(i,k) + W(k+1,j)] \end{array} \right\}$$

- W(i + 1,j) y W(i,j - 1) cuando i y j no están pareados
- V(i,j) cuando i y j están pareados
- El último, están preados pero no necesariamente entre si

Matriz V

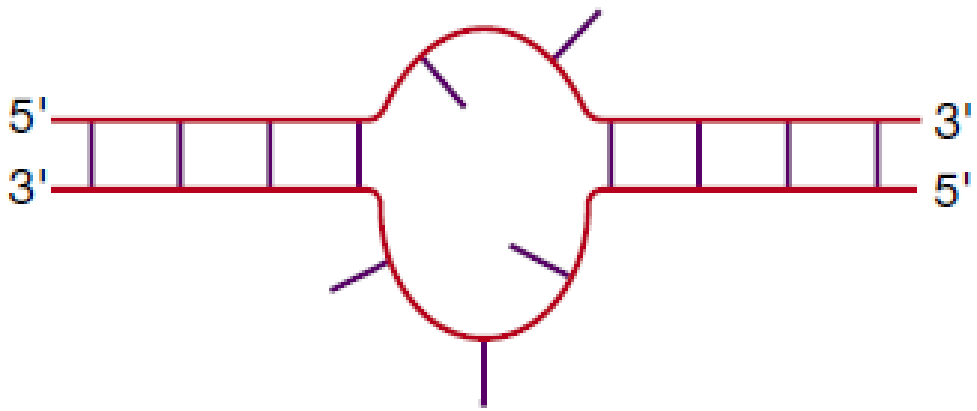
$$V(i,j) = min \left\{ \begin{array}{c} eh(i,j) \\ es(i,j) + V(i+1, j-1) \\ VBI(i,j) \\ VM(i,j) \end{array} \right\}$$

- eh: energía de la horquilla
- es: energía del stack
- VBI: energía del bucle interno
- VM: energía del multiloop

Matriz VBI

$$VBI(i,j) = \text{mín}_{i<i'<j'<j/i'-i+j-j'>2}[ebi(i,j,i',j') + V(i',j')]$$

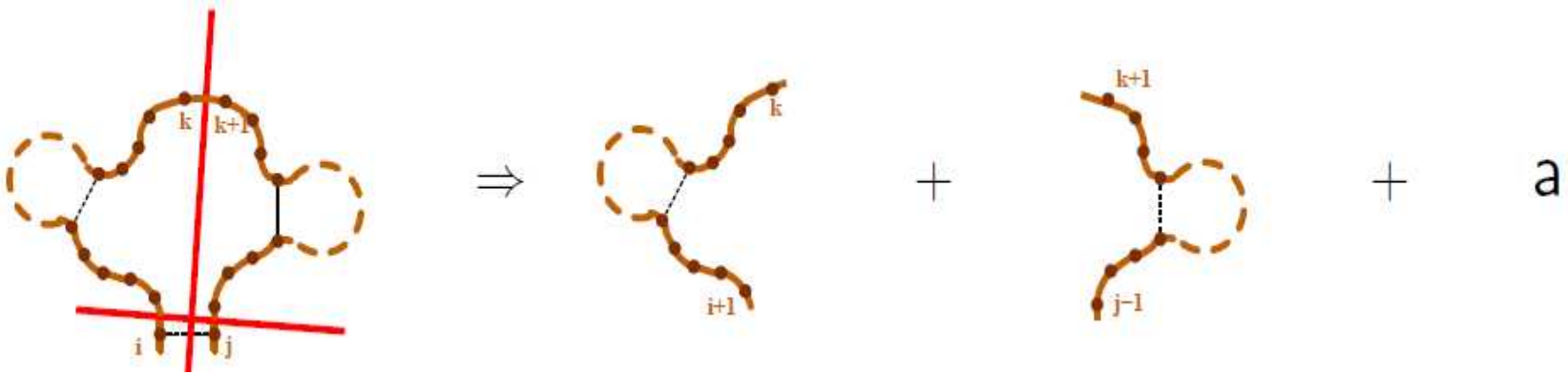
- Se ven todas las combinaciones que pueden producir un bucle interno para un i' y j' que este entre i y j



Matriz VM

$$VM(i,j) = \min_{i<k<j-1}[W(i+1,k) + W(k+1,j-1)] + a$$

- Se ven las diferentes formas de separar un multiloop en dos partes, agregando un valor "a" de cierre del loop



Algoritmo de Zuker

- El tiempo de Zuker esta en $O(n^4)$
- Si se quisiera hacer con estructuras terciarias, sería un problema de programación dinámica multidimensional y con 3 matrices
  - Impracticable
- El traceback es como Nussinov

Tarea

- ¿Que significa que los métodos de predicción de estructuras de RNA busquen palíndromos, como la minimización de energía libre afecta estos resultados?
- Enviar a personaje.ts@gmail.com