

Cours M2 BIBS

Repliement *in silico* de l'ARN

Yann Ponty

Bioinformatics Team
École Polytechnique/CNRS/INRIA AMIB

PIMS Vancouver
Simon Fraser University
Canada

<http://www.lix.polytechnique.fr/~ponty/index.php?page=bibscasm2014>

18 Novembre 2013

. . . ou comment gagner 1 million de dollars en rendant la monnaie !!

Problème : Vous disposez de pièces de **1**, **20** et **50** centimes. Le client souhaite minimiser la monnaie reçue (en nombre de pièces).

Comment rendre **N** en monnaie sans perdre un client ?

Stratégie 1 : Commencer par les *grosses* pièces puis compléter avec les *petites*.

$$21 = ??$$

...ou comment gagner 1 million de dollars en rendant la monnaie!!

Problème : Vous disposez de pièces de **1**, **20** et **50** centimes. Le client souhaite minimiser la monnaie reçue (en nombre de pièces).

Comment rendre **N** en monnaie sans perdre un client ?

Stratégie 1 : Commencer par les *grosses* pièces puis compléter avec les *petites*.

$$21 = \text{20} + \text{1}$$

55??

...ou comment gagner 1 million de dollars en rendant la monnaie!!

Problème : Vous disposez de pièces de **1**, **20** et **50** centimes. Le client souhaite minimiser la monnaie reçue (en nombre de pièces).

Comment rendre **N** en monnaie sans perdre un client ?

Stratégie 1 : Commencer par les *grosses* pièces puis compléter avec les *petites*.

$$21 = \text{€20} + \text{€1}$$

$$55 = \text{€50} + \text{€5} + \text{€5} + \text{€5} + \text{€5} + \text{€5}$$

60??

...ou comment gagner 1 million de dollars en rendant la monnaie!!

Problème : Vous disposez de pièces de **1**, **20** et **50** centimes. Le client souhaite minimiser la monnaie reçue (en nombre de pièces).

Comment rendre **N** en monnaie sans perdre un client ?

Stratégie 1 : Commencer par les *grosses* pièces puis compléter avec les *petites*.

$$21 = \text{€}20 + \text{€}1$$

$$55 = \text{€}50 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1$$

$$60 = \text{€}50 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 + \text{€}1 ??$$

...ou comment gagner 1 million de dollars en rendant la monnaie!!

Problème : Vous disposez de pièces de **1**, **20** et **50** centimes. Le client souhaite minimiser la monnaie reçue (en nombre de pièces).

Comment rendre **N** en monnaie sans perdre un client ?

Stratégie 1 : Commencer par les *grosses* pièces puis compléter avec les *petites*.

$$21 = \text{€20} + \text{€1}$$

$$55 = \text{€50} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1}$$

$$60 = \text{€50} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1} + \text{€1} ??$$

$$= \text{€20} + \text{€20} + \text{€20} !$$

Problème *a priori* (?!) non-résoluble en général par une approche *gloutonne* car problème plus simple NP-complet (Existe t il même une façon efficace de rendre la monnaie ? \Rightarrow 1M\$).

Stratégie 2 : Il existe une récurrence donnant le nombre minimal de pièce :

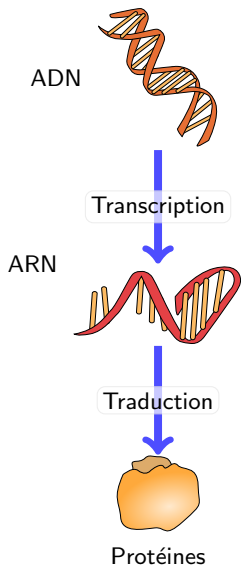
$$NbPieces(N) = \text{Min} \begin{cases} \text{1€} \rightarrow 1 + NbPieces(N - 1) \\ \text{2€} \rightarrow 1 + NbPieces(N - 20) \\ \text{5€} \rightarrow 1 + NbPieces(N - 50) \end{cases}$$

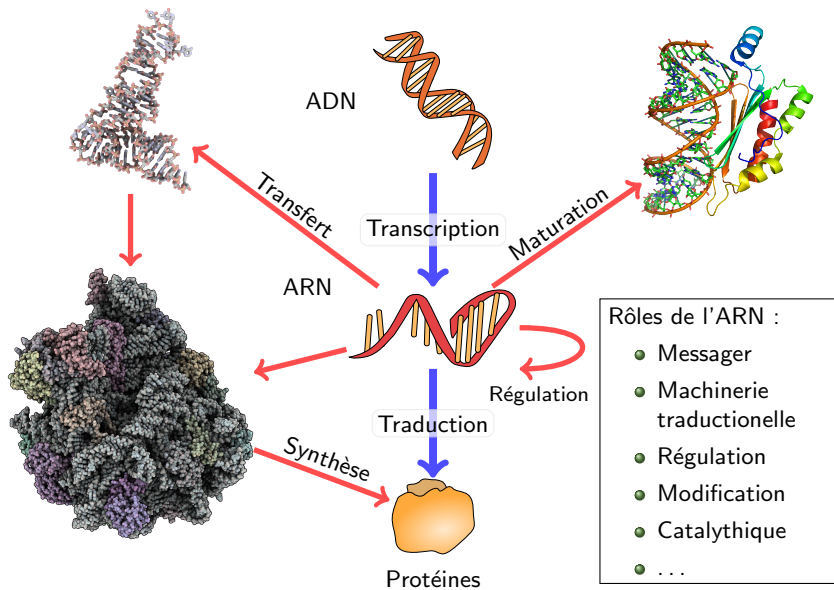
Avec un peu de mémoire (N résultats intermédiaires/cas à retenir), on peut alors répondre après $N \times \#Pièces$ calculs.

Remarque : On n'a pas gagné le million, car N a une valeur exponentielle sur son codage. Cet algorithme est donc en temps exponentiel au regard de la théorie de la complexité.

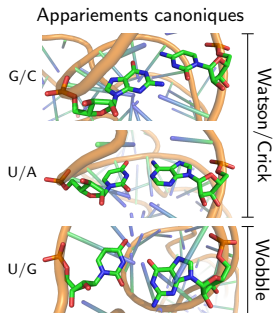
Mais on a optimisé, en évitant un parcours exhaustif de l'arbre des possibles :
 \Rightarrow Programmation dynamique.

- 1 Introduction
 - Fonction(s) de l'ARN
 - Repliement et structure
 - Représentations de la structure secondaire
- 2 Formalisation du repliement et outils disponibles
 - Aparté thermodynamique
 - Programmation dynamique : Rappels
- 3 Minimisation de l'énergie libre
 - Modèle de Nussinov
 - Modèle de Turner
 - MFold/Unafold
 - Performances et approches comparatives
 - Vers une prédiction ab-initio 3D
- 4 Ensemble de Boltzmann
 - Ensemble de Boltzmann
 - Nussinov : Minimisation \Rightarrow Comptage
 - Calcul de la fonction de partition
 - Échantillonnage statistique





ARN = Biopolymère composé de nucléotides A,C,G et U
A : Adénosine, C : Cytosine, G : Guanine et U : Uracile



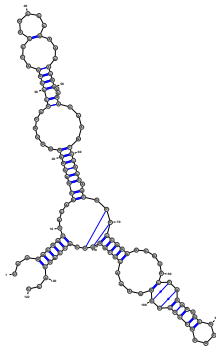
Repliement de l'ARN = Processus stochastique continu dirigé par (résultant en) un appariement des nucléotides.

Comprendre le repliement des ARN aide à comprendre et prédire leur fonction.

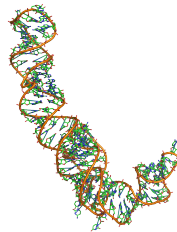
Trois¹ niveaux de représentation :

```
UUAGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCGAA
CACGGAAGAUAGCC
CACCAGCGUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGAAA
CCCGGUUCGCCCA
CC
```

Structure primaire



Structure secondaire



Structure tertiaire

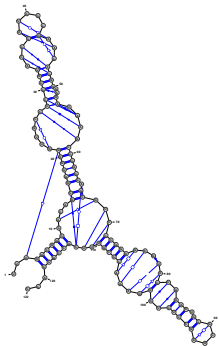
Source : 5s rRNA (PDB 1K73 :B)

1. Enfin, presque ...

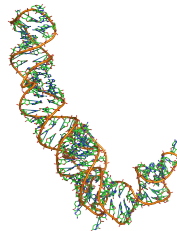
Trois¹ niveaux de représentation :

```
UUAGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCGAA
CACGGAAGAUAGCC
CACCAGCGUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGAAA
CCCGGUUCGCCCA
CC
```

Structure primaire



Structure secondaire⁺



Structure tertiaire

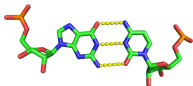
Source : 5s rRNA (PDB 1K73 :B)

1. Enfin, presque ...

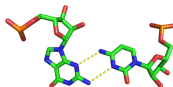
- Appariements non-canoniques

Toute paire de base **autre que** $\{(A-U), (C-G), (G-U)\}$

Ou interagissant sur un bord non-standard (WC/WC-Cis) [LW01].



Paire CG canonique (WC/WC-Cis)



Paire CG non canonique (Sucre/WC-Trans)

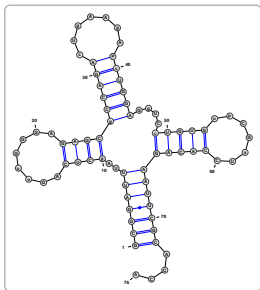
- Pseudonoeuds



Structure pseudonoeud d'un Ribozyme du Groupe I (PDBID : 1Y0Q :A)

Plus expressif, mais repliement général *in silico* avec pseudonoeud :

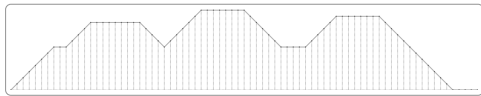
⇒ NP-Complet [LP00] ... polynomial pour certaines classes [CDR⁺04].



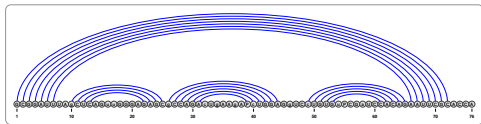
Graphe planaire (outer planar)

(((((...(((.....))))))(((.....))))...(((.....))))))....

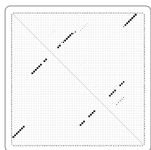
Expression bien parenthésée



Mountain view



Linéaire



Dot plot

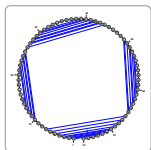
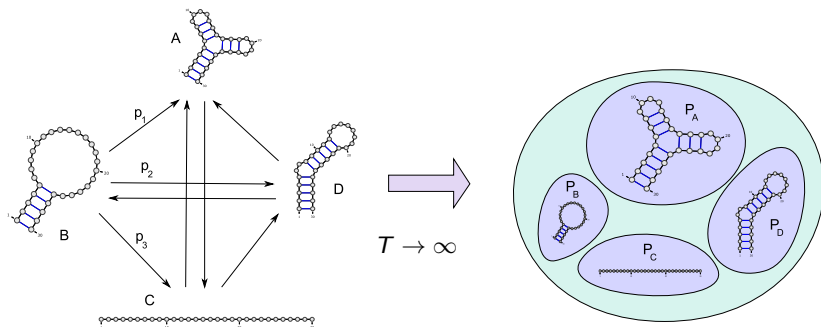


Diagramme de Feynman

Représentation différentes et équivalentes
⇒ Aide l'intuition algorithmique
+ Propriétés algébriques sympathiques
⇒ Algorithmique efficace !

- 1 Introduction
 - Fonction(s) de l'ARN
 - Repliement et structure
 - Représentations de la structure secondaire
- 2 Formalisation du repliement et outils disponibles
 - Aparté thermodynamique
 - Programmation dynamique : Rappels
- 3 Minimisation de l'énergie libre
 - Modèle de Nussinov
 - Modèle de Turner
 - MFold/Unafold
 - Performances et approches comparatives
 - Vers une prédiction ab-initio 3D
- 4 Ensemble de Boltzmann
 - Ensemble de Boltzmann
 - Nussinov : Minimisation \Rightarrow Comptage
 - Calcul de la fonction de partition
 - Échantillonnage statistique

A l'échelle nanoscopique, la structure de l'ARN *fluctue*.



Convergence vers une **distribution stationnaire** de probabilité, l'**équilibre de Boltzmann**, où la probabilité est exponentiellement faible sur l'**énergie libre**.

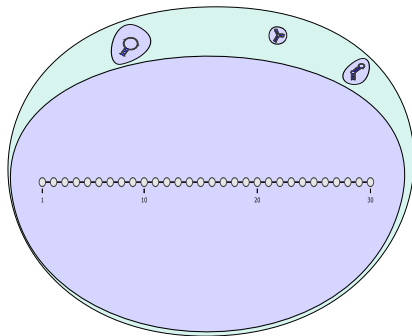
Corollaire : La conformation initiale est sans d'importance.

Problèmes soulevés :

Étant donnés des modèles pour l'**ensemble des conformations** et l'**énergie libre**.

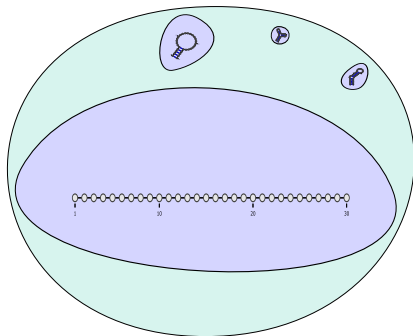
- Déterminer la structure la plus probable (= Énergie libre minimale) à l'équilibre
- Déterminer des propriétés moyennes de l'ensemble de Boltzmann

Transcription : ARN synthétisé sans appariement (Sauf exception)



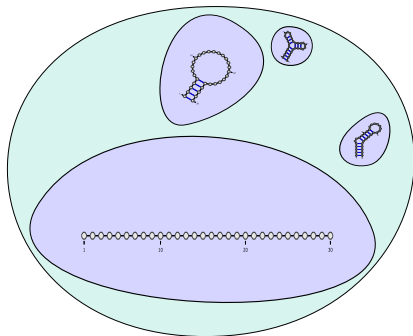
$T = 0$

Transcription : ARN synthétisé sans appariement (Sauf exception)



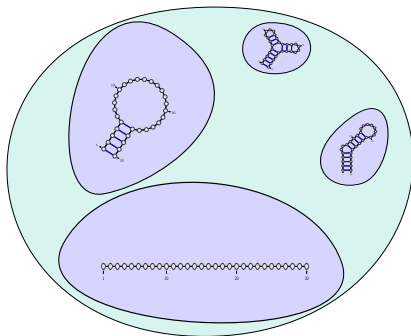
$T = 1h$

Transcription : ARN synthétisé sans appariement (Sauf exception)



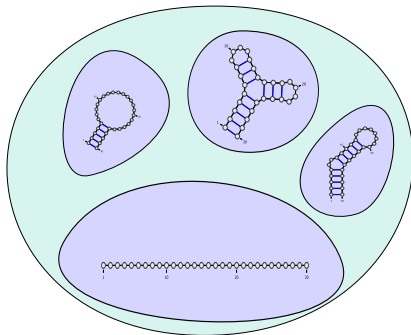
$$T = 2h$$

Transcription : ARN synthétisé sans appariement (Sauf exception)



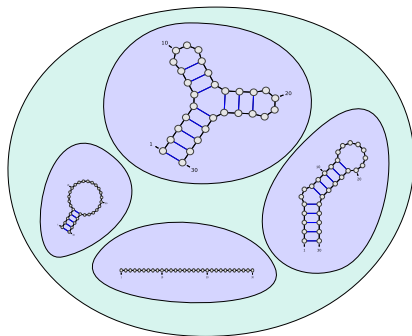
$T = 5h$

Transcription : ARN synthétisé sans appariement (Sauf exception)



$$T = 10h$$

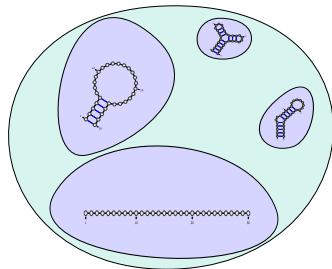
Transcription : ARN synthétisé sans appariement (Sauf exception)



$$T \rightarrow \infty$$

Mais majorité des ARNm dégradés avant 7h (Org. : Souris [SSN⁺09]).

Transcription : ARN synthétisé sans appariement (Sauf exception)



$T = 10h$

Mais majorité des ARNm dégradés avant 7h (Org. : Souris [SSN⁺09]).

- A. Déterminer la structure la plus probable (= Energie libre min.) à l'équilibre
- B. Déterminer des propriétés moyennes de l'ensemble de Boltzmann
- C. Déterminer la structure la plus probable à temps T .
(c.f. H. Isambert par simulation, NP-complet en déterministe [MTSC09])

- 1 Introduction
 - Fonction(s) de l'ARN
 - Repliement et structure
 - Représentations de la structure secondaire
- 2 Formalisation du repliement et outils disponibles
 - Aparté thermodynamique
 - Programmation dynamique : Rappels
- 3 Minimisation de l'énergie libre
 - Modèle de Nussinov
 - Modèle de Turner
 - MFold/Unafold
 - Performances et approches comparatives
 - Vers une prédiction ab-initio 3D
- 4 Ensemble de Boltzmann
 - Ensemble de Boltzmann
 - Nussinov : Minimisation \Rightarrow Comptage
 - Calcul de la fonction de partition
 - Échantillonnage statistique

Programmation dynamique = Technique générale pour l'optimisation.

Condition : Solution optimale pour P peut être reconstruite à partir de solutions pour des sous-problèmes strictes de P .

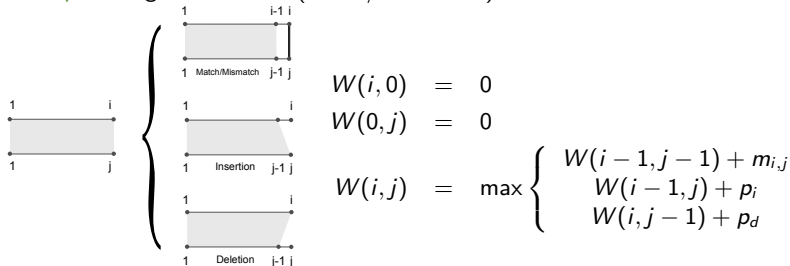
Bioinformatique :

Espace de solutions *discret* (alignements, repliements)

+ Fonction objectif *additive* (score, énergie)

⇒ Schéma de programmation dynamique efficace.

Exemple : Alignement local (Smith/Waterman)



Un schéma fait intervenir des *classes* de sous-problèmes dont on sait calculer le score du *champion*.

Étant donné un schéma, deux étapes :

- **Calcul matrices** : Sauvegarde des meilleurs scores sur classes de sous-problèmes (Ordre inverse de celui induit par les dépendances).
- **Remontée** : Reconstitue le parcours ayant mené au meilleur score. (Parcours = Instance)

Complexité du calcul dépend alors :

- **Taille** de l'espace des sous-problèmes
- **Nombres** de sous-problèmes considérés (#Termes décomposition)

Exemple S/W :

$$i : 1 \rightarrow n + 1 \Rightarrow \Theta(n)$$

$$j : 1 \rightarrow m + 1 \Rightarrow \Theta(m)$$

Trois opération pour chaque sous-calcul

$\Rightarrow \Theta(m.n)$ temps/mémoire



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

	A	C	A	C	A	C	T	A
0	0	0	0	0	0	0	0	0
A	0							
G	0							
C	0							
A	0							
C	0							
A	0							
C	0							
A	0							

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

	A	C	A	C	A	C	T	A
0	0	0	0	0	0	0	0	0
A	0	↓	→ 2					
G	0							
C	0							
A	0							
C	0							
A	0							
C	0							
A	0							

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1						
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Diagram illustrating the dynamic programming table for local sequence alignment. The table shows the alignment of AGCACACA (row) and ACACACTA (column). The value 2 is highlighted in the cell (A, A) at row 2, column 3, indicating a match. Red arrows point from the cell (A, A) to its neighbors (A, C) and (G, A), indicating transitions. A black arrow points from (A, C) to (A, A), and a black arrow points from (A, A) to (A, C).

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2					
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Diagram illustrating the dynamic programming table for local sequence alignment. The table shows the alignment of AGCACACA (row) and ACACACTA (column). The values in the table represent the maximum score for the alignment up to that point. Red arrows indicate the path of the optimal alignment: from (0,0) to (1,2) (A-C), then to (2,3) (G-A), and finally to (3,4) (C-C).

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1				
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Diagram illustrating the dynamic programming table for local sequence alignment. The table shows the alignment of AGCACACA (row) and ACACACTA (column). The values in the table represent the maximum score for the alignment up to that point. Red arrows indicate the path of the optimal alignment: (0,0) to (1,2), (1,2) to (2,3), (2,3) to (3,4), and (3,4) to (4,5). A grey arrow points from (3,5) to (4,5), and a black arrow points from (4,5) to (4,6).

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Red arrows indicate the path of the local alignment: (0,0) to (1,1) to (1,2) to (2,2) to (2,3) to (3,3) to (3,4) to (4,4) to (4,5) to (4,6) to (4,7) to (4,8) to (4,9).

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Diagram illustrating the dynamic programming table for local sequence alignment. The table shows the alignment of AGCACACA (row 1) and ACACACTA (row 2). Red arrows indicate the path of the optimal alignment: (0,0) to (1,1), (1,1) to (2,2), (2,2) to (3,3), (3,3) to (4,4), (4,4) to (5,5), (5,5) to (6,6), (6,6) to (7,7), (7,7) to (8,8), and (8,8) to (9,9). The values in the table represent the maximum score for each subproblem.

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0								
C	0								
A	0								
C	0								
A	0								

Diagram illustrating the dynamic programming table for local sequence alignment. The table shows the maximum score $W(i, j)$ for aligning the prefix of sequence 1 (AGCACACA) of length i with the prefix of sequence 2 (ACACACTA) of length j . Red arrows indicate the optimal path from the top-left cell (0,0) to the bottom-right cell (6,6), showing matches (A-C, C-A, A-C, C-A, A-C) and a deletion (T) at the end.

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0								
A	0								
C	0								
A	0								

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
0	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Exemple : Alignement local de séquences AGCACACA et ACACACTA

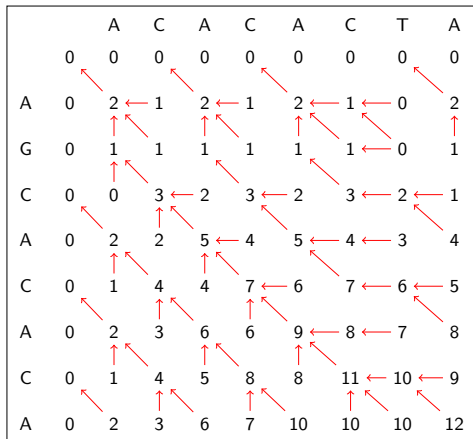
Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

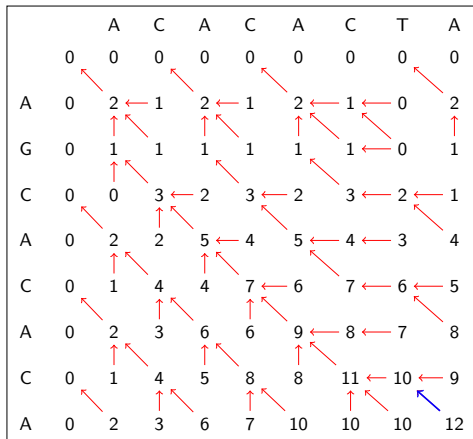
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

A
A



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

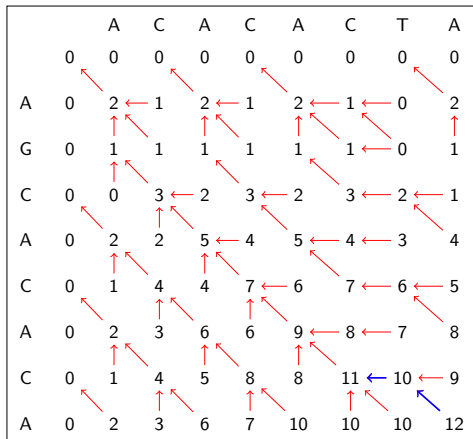
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

- A
T A



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

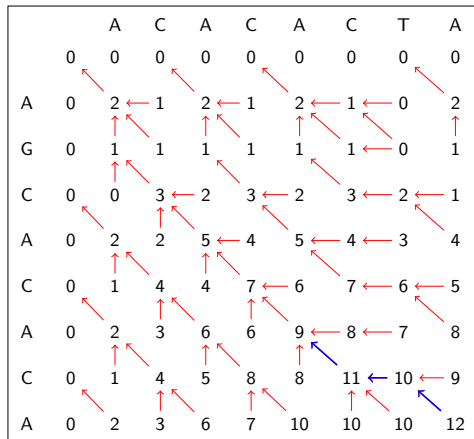
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

C - A
C T A



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

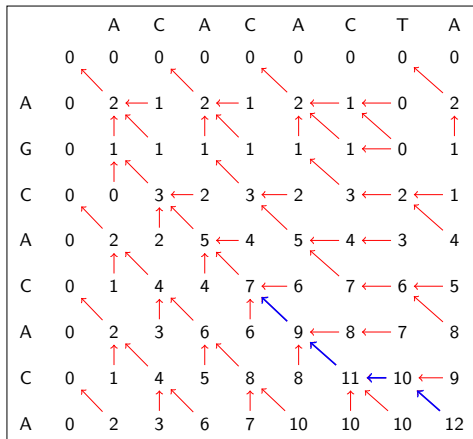
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

A C - A
A C T A



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

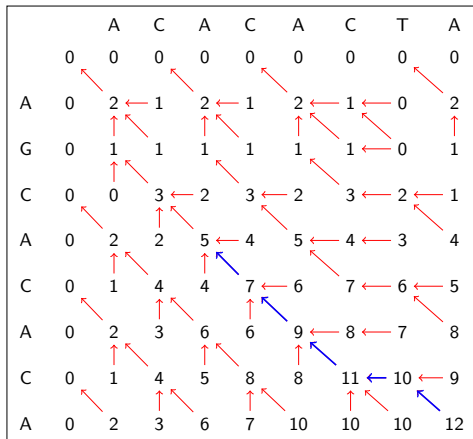
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

C A C - A
C A C T A



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

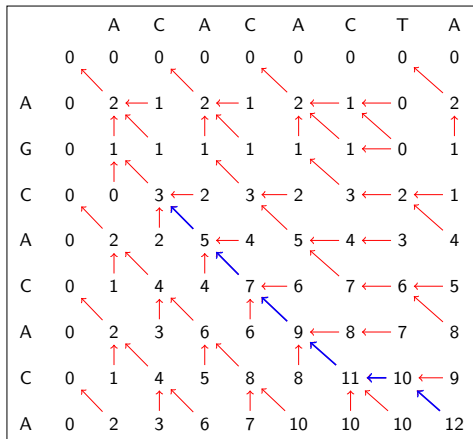
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

A	C	A	C	-	A
A	C	A	C	T	A



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

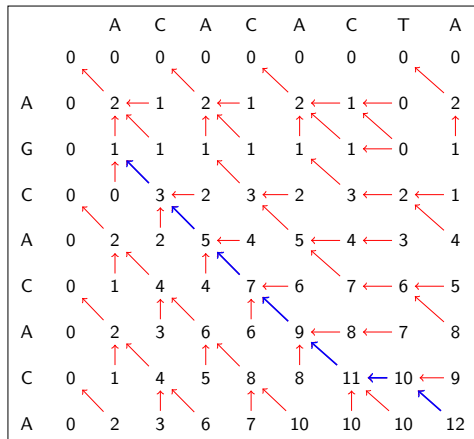
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

C A C A C - A
 C A C A C T A



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

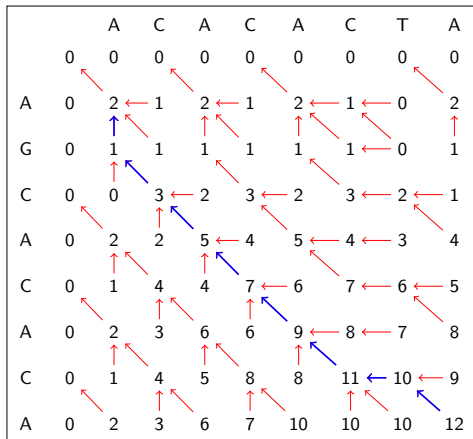
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

G C A C A C - A
 - C A C A C T A



Exemple : Alignement local de séquences AGCACACA et ACACACTA

Coûts : Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

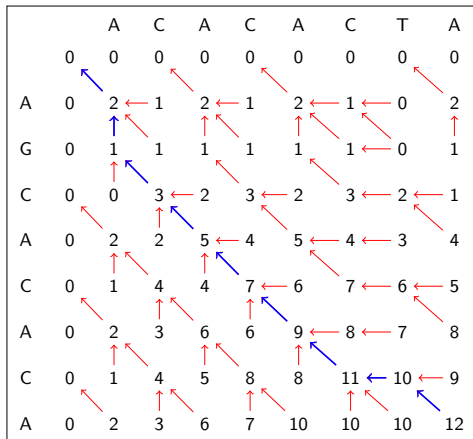
$$W(i, 0) = 0$$

$$W(0, j) = 0$$

$$W(i, j) = \max \begin{cases} W(i-1, j-1) + m_{i,j} \\ W(i-1, j) + p_i \\ W(i, j-1) + p_d \end{cases}$$

Meilleur alignement

A	G	C	A	C	A	C	-	A
A	-	C	A	C	A	C	T	A



Propriétés requise d'un schéma :

- **Validité** : \forall sous-problème, la valeur obtenue doit être celle de la fonction objectif.

Preuve souvent assez technique.

Propriétés souhaitables d'un schéma :

- **Complétude** : Espace des solutions engendré par la décomposition.
Des astuces algorithmiques peuvent *couper des branches* . . .
- **Non-ambiguïté** : Chaque solution est *engendrée* au plus une fois.

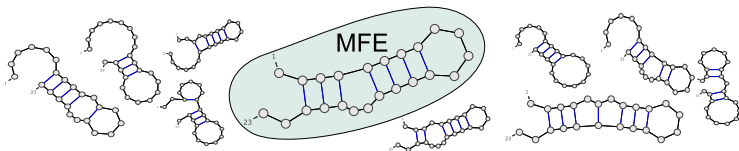
\Rightarrow Possibilité d'**énumérer** l'espace des solutions.

- 1 Introduction
 - Fonction(s) de l'ARN
 - Repliement et structure
 - Représentations de la structure secondaire
- 2 Formalisation du repliement et outils disponibles
 - Aparté thermodynamique
 - Programmation dynamique : Rappels
- 3 Minimisation de l'énergie libre
 - Modèle de Nussinov
 - Modèle de Turner
 - MFold/Unafold
 - Performances et approches comparatives
 - Vers une prédiction ab-initio 3D
- 4 Ensemble de Boltzmann
 - Ensemble de Boltzmann
 - Nussinov : Minimisation \Rightarrow Comptage
 - Calcul de la fonction de partition
 - Échantillonnage statistique

Problème A : Déterminer la structure d'énergie minimale.

Repliement *ab initio* =

Trouver structure d'un ARN ω uniquement à partir de sa séquence.



- **Conformations** : Ensemble S_ω des structures secondaires compatibles avec la structure primaire ω (contrainte d'appariements).
- **Fonction d'énergie** Énergie libre associant une valeur numérique $E_{\omega,S}$ (KCal.mol^{-1}) à tout couple séquence/conformation (ω, S) .
- **Structure native** : Conformation *fonctionnelle* de la molécule.

Remarques :

- Pas nécessairement unique (Cinétique ou structures bi-stables)
- Présence de pseudo-noeuds : Ambiguïté, quelle est la structure native ?

Modèle de Nussinov/Jacobson (NJ)

Plus proche voisins simple :

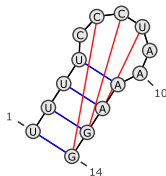
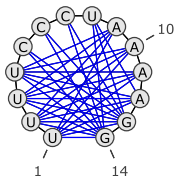
- Seuls les appariements contribuent à l'énergie
- Uniquement liaisons Watson/Crick (A/U,C/G) et Wobble (G/U)

$$\Rightarrow E_{\omega,S} = -\#Paires(S)$$

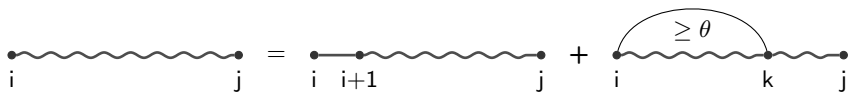
Repliement dans NJ \Leftrightarrow Maximisation du nombre de paires de bases.

Exemple :

UUUJCCCUAAAAGG

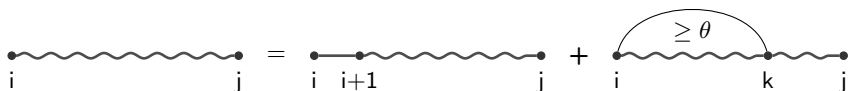


Variante : Pondérer les paires selon leur nombre de liaisons hydrogène
 $\Delta G(G \equiv C) = -3$ $\Delta G(A = U) = -2$ $\Delta G(G - U) = -1$



$$N_{i,t} = 0, \quad \forall t \in [i, i + \theta]$$

$$N_{i,j} = \min \begin{cases} N_{i+1,j} & i \text{ non apparié} \\ \min_{k=i+\theta+1}^j E_{i,k} + N_{i+1,k-1} + N_{k+1,j} & i \text{ apparié à } k \end{cases}$$



$$N_{i,t} = 0, \quad \forall t \in [i, i + \theta]$$

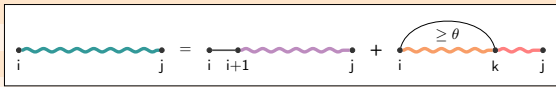
$$N_{i,j} = \min \begin{cases} N_{i+1,j} & i \text{ non apparié} \\ \min_{k=i+\theta+1}^j E_{i,k} + N_{i+1,k-1} + N_{k+1,j} & i \text{ apparié à } k \end{cases}$$

Correction : On cherche à montrer que l'énergie de la structure d'énergie la plus faible ($MFE_{1,n}$) est bien calculée dans $N_{1,n}$. Dans toute structure secondaire restreinte à $[i, j]$ la première position i est :

- **Soit non-appariée :** $MFE_{i,j}$ est constituée des appariements de $MFE_{i+1,j}$.
- **Soit appariée à k :** $MFE_{i,j}$ contient l'appariement (i, k) et l'union des appariements de $MFE_{i+1,k-1}$ et de $MFE_{k+1,j}$. En effet, tout appariement entre les régions $[i + 1, k - 1]$ et $[k + 1, j]$ **croiserait** (i, k) (Pseudonoed).

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	

C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	



	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	

C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$\text{Sequence}(i, j) = \max \left(\text{Sequence}(i, i+1) + \text{Sequence}(i+1, j), \max_{k: k-i \geq \theta} \left(\text{Sequence}(i, k) + \text{Sequence}(k, j) \right) \right)$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	

C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$\text{Sequence } [i, \dots, j] = \text{Sequence } [i, i+1, \dots, j] + \text{Sequence } [i, \dots, k, \dots, j] \text{ (with loop } \geq \theta \text{)}$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	

C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$i \text{---} j = i \text{---} i+1 + i \text{---} k + \text{loop}(\geq \theta)$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

Diagram illustrating the Nussinov/Jacobson algorithm's recurrence relation. It shows a sequence of nucleotides from index i to j . The sequence is partitioned into two parts: a subsequence from i to $i+1$ and another from i to k . A curved arrow above the sequence from i to k is labeled with the inequality $\geq \theta$, representing a minimum energy threshold for a base pair. The diagram is an equation: a sequence from i to j equals a sequence from i to $i+1$ plus a sequence from i to k .

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

$$\text{Sequence}(i..j) = \text{Sequence}(i..i+1) + \text{Sequence}(i+1..j) + \text{Sequence}(i..k) \text{ with loop } \geq \theta$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	2	3	5	5	6	7	
U									0	0	0	2	3	5	5	5	7	
C										0	0	0	3	3	3	5	5	
U											0	0	0	2	2	2	3	
U												0	0	0	0	1	2	
A													0	0	0	0	0	
G														0	0	0	0	
A															0	0	0	
C																0	0	
G																	0	
A																		0

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

Diagram illustrating the recurrence relation for the Nussinov/Jacobson algorithm. It shows a sequence of nucleotides from index i to j . The sequence is decomposed into two parts: a sequence from i to $i+1$ and a sequence from $i+1$ to j . The second part is further decomposed into a sequence from i to k and a sequence from k to j , with a gap of at least θ between i and k .

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	2	2	4	5	7	7	8	10		
A						0	0	0	0	2	2	2	5	5	5	8	8		
C							0	0	0	0	0	2	5	5	5	8	8		
U								0	0	0	0	2	3	5	5	6	7		
U									0	0	0	2	3	5	5	5	7		
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	1	2		
A													0	0	0	0	0		
G														0	0	0	0		
A															0	0	0		
C																0	0		
G																	0		
A																		0	

$$\text{Sequence } [i, j] = \text{Sequence } [i, i+1] + \text{Sequence } [i+1, j] + \text{Sequence } [i, k] + \text{Sequence } [k, j]$$

The diagram shows a sequence of nucleotides from index i to j . The sequence is decomposed into two parts: a sequence from i to $i+1$ followed by a sequence from $i+1$ to j , plus a sequence from i to k followed by a sequence from k to j . A curved arrow above the sequence from i to k is labeled with the inequality $\geq \theta$, indicating a minimum gap constraint between the two sub-sequences.

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

$$\text{Sequence}(i, j) = \text{Sequence}(i, i+1) + \text{Sequence}(i+1, j) + \max_{k \geq i+1} \{ \text{Sequence}(i, k) + \text{Sequence}(k, j) \mid k - i \geq \theta \}$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$i \text{---} j = i \text{---} i+1 \text{---} j + i \text{---} k \text{---} j$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

Diagram illustrating the recurrence relation for the Nussinov algorithm. It shows a sequence of nucleotides from index i to j . The sequence is decomposed into two parts: a sequence from i to $i+1$ followed by a sequence from $i+1$ to j , plus a sequence from i to k followed by a sequence from k to j . A curved arrow above the sequence from i to k is labeled with the inequality $\geq \theta$, indicating a minimum energy threshold for a sub-structure.

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(.)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

Diagram illustrating the Nussinov/Jacobson algorithm's recurrence relation. It shows a sequence of nucleotides from index i to j . The sequence is partitioned into two parts: a subsequence from i to $i+1$ and a subsequence from $i+1$ to j . The second part is further partitioned into a subsequence from i to k and a subsequence from k to j . A curved arrow above the subsequence from i to k is labeled with the inequality $\geq \theta$, indicating a minimum energy threshold for a base pair.

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	((.))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

Diagram illustrating the Nussinov/Jacobson algorithm's recurrence relation. It shows a sequence of nucleotides from index i to j . The sequence is partitioned into two parts: a subsequence from i to $i+1$ and another from $i+1$ to j . Additionally, a pair of nucleotides (i, k) is shown forming a base pair, with a curved arrow above them labeled with the inequality $\geq \theta$. The diagram is presented as an equation: sequence($i..j$) = sequence($i..i+1$) + sequence($i+1..j$) + pair(i, k).

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	((.))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$\text{Sequence}(i, j) = \text{Sequence}(i, i+1) + \text{Sequence}(i+1, j) + \max_{k \geq i+1} \text{Sequence}(i, k) + \text{Sequence}(k, j) \quad \text{if } k - i \geq \theta$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	((.))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$\text{Sequence}(i, j) = \max \left(\text{Sequence}(i, i+1) + \text{Sequence}(i+1, j), \max_{k \geq i+1} \left(\text{Pair}(i, k) + \text{Sequence}(i, k) + \text{Sequence}(k, j) \right) \right)$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(((.	.	.)))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$\text{Sequence}(i, j) = \max \left(\text{Sequence}(i, i+1) + \text{Sequence}(i+1, j), \max_{k \geq i+1} \left(\text{Sequence}(i, k) + \text{Sequence}(k, j) \right) \right)$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(((.	.	.)))	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	2	3	5	6	7		
U									0	0	0	0	2	3	5	5	7	
C										0	0	0	0	3	3	5	5	
U											0	0	0	0	2	2	3	
U												0	0	0	0	1	2	
A													0	0	0	0	0	
G														0	0	0	0	
A															0	0	0	
C																0	0	
G																	0	
A																		0

$$i \dots j = i \dots i+1 \dots j + i \dots k \dots j$$

$$\text{with } \text{score}(i, k) \geq \theta$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(((.	.	.)))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$\text{Sequence } i \dots j = \text{Sequence } i \dots i+1 \dots j + \text{Sequence } i \dots k \dots j \text{ (with loop } \geq \theta \text{)}$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(((.	.	.)))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$\text{Sequence}(i, j) = \text{Sequence}(i, i+1) + \text{Sequence}(i+1, j) + \max_{k \geq i+1} \{ \text{Sequence}(i, k) + \text{Sequence}(k, j) \mid k - i \geq \theta \}$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(((.	.	.)))	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	2	3	5	5	6	7	
U									0	0	0	2	3	5	5	5	7	
C										0	0	0	3	3	3	5	5	
U											0	0	0	2	2	2	3	
U												0	0	0	0	1	2	
A													0	0	0	0	0	
G														0	0	0	0	
A															0	0	0	
C																0	0	
G																	0	
A																		0

$i \text{---} j = i \text{---} i+1 \text{---} j + i \text{---} k \text{---} j$

$\geq \theta$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(((.	.	.)))	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	7	
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(((.	.	.)))	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

Diagram illustrating the Nussinov algorithm recurrence relation:

$$i \text{---} j = i \text{---} i+1 \text{---} j + i \text{---} k \text{---} j \text{ (with } \geq \theta \text{ loop)}$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(((.	.	.)	.	(.)))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

Diagram illustrating the Nussinov algorithm recurrence relation. It shows a sequence of nucleotides from index i to j . The sequence is partitioned into three parts: a segment from i to $i+1$, a segment from $i+1$ to j , and a segment from i to k with a loop back to j . The loop is labeled with a greater-than-or-equal-to θ symbol. The equation is: sequence($i..j$) = sequence($i..i+1$) + sequence($i+1..j$) + sequence($i..k$) with a loop to j .

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(((.	.	.)	.	(.)))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$i \text{---} \dots \text{---} j = i \text{---} i+1 \text{---} \dots \text{---} j + i \text{---} k \text{---} \dots \text{---} j$

 where $k - i \geq \theta$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(((.	.	.)	.	(.)))	.	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8	8
U								0	0	0	0	0	2	3	5	5	6	7	7
U									0	0	0	0	2	3	5	5	5	7	7
C										0	0	0	0	3	3	3	5	5	5
U											0	0	0	0	2	2	2	3	3
U												0	0	0	0	0	1	2	2
A													0	0	0	0	0	0	0
G														0	0	0	0	0	0
A															0	0	0	0	0
C																0	0	0	0
G																	0	0	0
A																		0	0

$$i \text{---} j = i \text{---} i+1 \text{---} j + i \text{---} k \text{---} j$$

$$\text{with } i+1 \text{---} k \text{---} j \text{ and } i \text{---} k \text{---} j \text{ separated by } \geq \theta$$

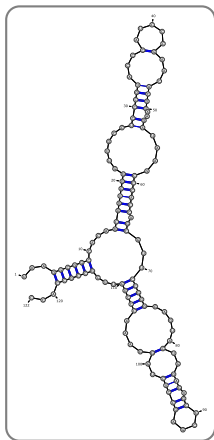
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(((.	.	.)	.	((.	.	.))))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	5	8	8	
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$i \text{---} j = i \text{---} i+1 \text{---} j + i \text{---} k \text{---} j$$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A	
	(((.	.	.)	.	((.	.	.))))	.	
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14	
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11	
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10	
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10	
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10	
A						0	0	0	0	0	2	2	2	5	5	5	8	8	
C							0	0	0	0	0	0	2	5	5	8	8		
U								0	0	0	0	0	2	3	5	5	6	7	
U									0	0	0	0	2	3	5	5	5	7	
C										0	0	0	0	3	3	3	5	5	
U											0	0	0	0	2	2	2	3	
U												0	0	0	0	0	1	2	
A													0	0	0	0	0	0	
G														0	0	0	0	0	
A															0	0	0	0	
C																0	0	0	
G																	0	0	
A																		0	

$$i \text{---} j = i \text{---} i+1 \text{---} j + i \text{---} k \text{---} j \text{ (loop } \geq \theta \text{)}$$

Basée sur décomposition **non-ambiguë** en **boucles** de la structure 2^aire :



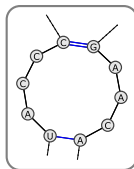
Énergies libres ΔG des boucles dépendent des bases, asymétrie, bases *libres* (dangle) ...

Déterminées expérimentalement
+ Interpolation pour grandes boucles.

Meilleure résultats grâce à la prise en compte de l'empilement.

Basée sur décomposition **non-ambiguë** en **boucles** de la structure 2^aie :

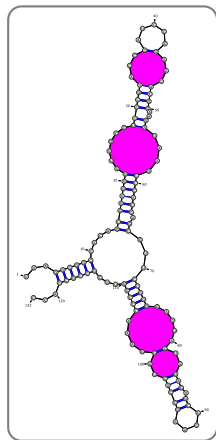
- Boucles internes



Énergies libres ΔG des boucles dépendent des bases, asymétrie, bases *libres* (dangle) ...

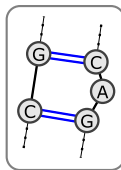
Déterminées expérimentalement
+ Interpolation pour grandes boucles.

Meilleure résultats grâce à la prise en compte de l'empilement.



Basée sur décomposition **non-ambiguë** en **boucles** de la structure 2^{aire} :

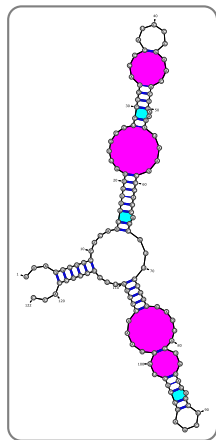
- Boucles internes
- Renflements



Énergies libres ΔG des boucles dépendent des bases, asymétrie, bases *libres* (dangle) ...

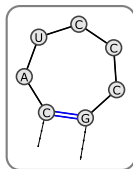
Déterminées expérimentalement
+ Interpolation pour grandes boucles.

Meilleure résultats grâce à la prise en compte de l'empilement.



Basée sur décomposition **non-ambiguë** en **boucles** de la structure 2^{aire} :

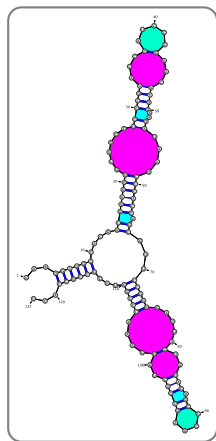
- Boucles internes
- Renflements
- Boucles terminales



Énergies libres ΔG des boucles dépendent des bases, asymétrie, bases *libres* (dangle) ...

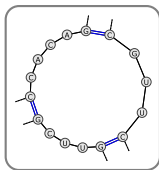
Déterminées expérimentalement
+ Interpolation pour grandes boucles.

Meilleure résultats grâce à la prise en compte de l'empilement.



Basée sur décomposition **non-ambiguë** en **boucles** de la structure 2^{aire} :

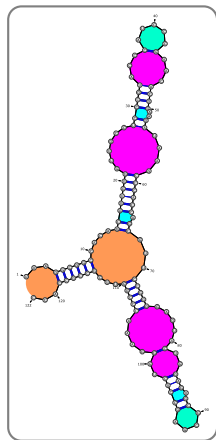
- Boucles internes
- Renflements
- Boucles terminales
- Boucles multiples



Énergies libres ΔG des boucles dépendent des bases, asymétrie, bases *libres* (dangle) ...

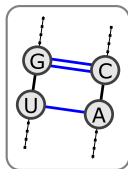
Déterminées expérimentalement
+ Interpolation pour grandes boucles.

Meilleure résultats grâce à la prise en compte de l'empilement.



Basée sur décomposition **non-ambiguë** en **boucles** de la structure 2^{aire} :

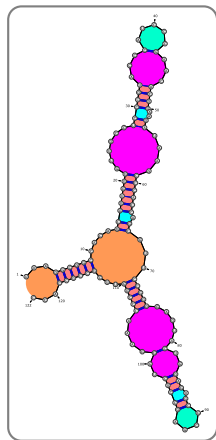
- Boucles internes
- Renflements
- Boucles terminales
- Boucles multiples
- Empilements

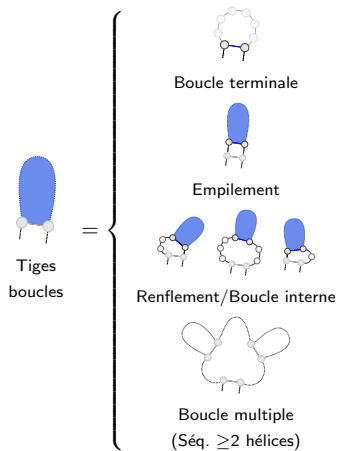


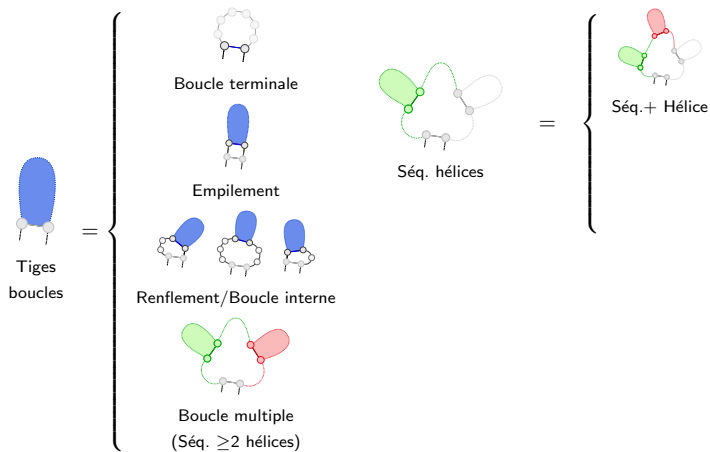
Énergies libres ΔG des boucles dépendent des bases, asymétrie, bases *libres* (dangle) ...

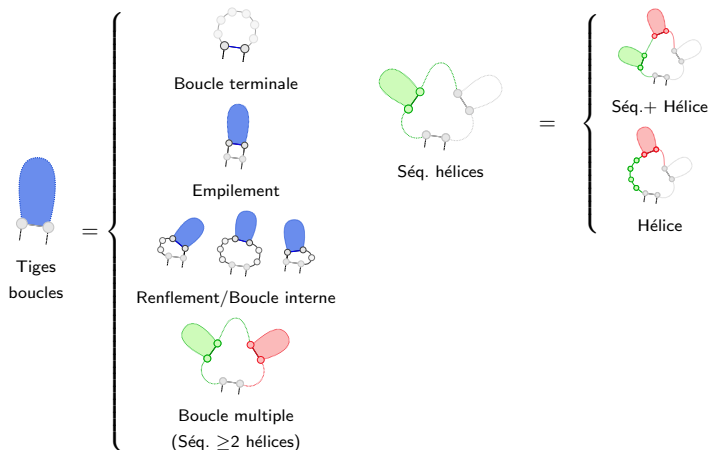
Déterminées expérimentalement
+ Interpolation pour grandes boucles.

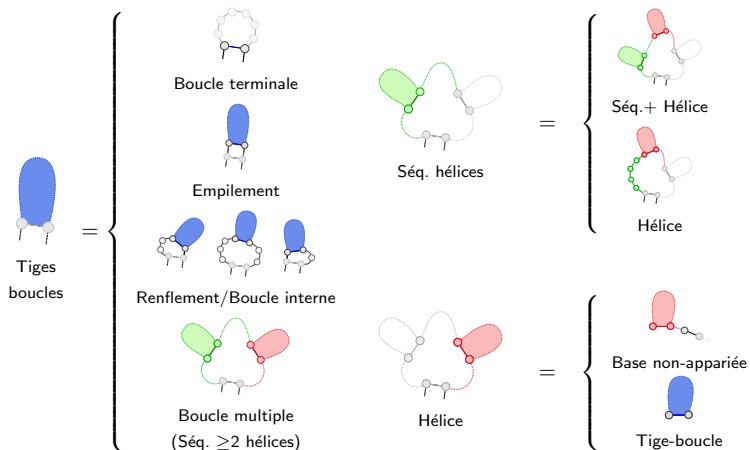
Meilleure résultats grâce à la prise en compte de l'empilement.



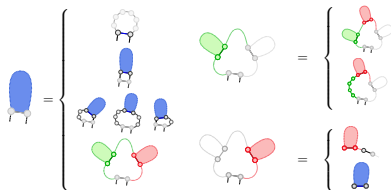








- $E_H(i, j)$: Energie de boucle terminale *fermée* par une paire (i, j)
- $E_{Bl}(i, j)$: Energie de renflement ou boucle interne *fermée* par une paire (i, j)
- $E_S(i, j)$: Energie d'empilement $(i, j)/(i + 1, j - 1)$
- a, c, b : Pénalité de boucle multiple, hélice et non-appariées dans multiboucle.



Calcul des matrices

$$\begin{aligned}
 \mathcal{M}'_{ij} &= \min \left\{ \begin{array}{l} E_H(i, j) \\ E_S(i, j) + \mathcal{M}'_{i+1, j-1} \\ \text{Min}_{i', j'} (E_{Bl}(i, i', j', j) + \mathcal{M}'_{i', j'}) \\ a + c + \text{Min}_k (\mathcal{M}_{i+1, k-1} + \mathcal{M}^1_{k, j-1}) \end{array} \right. \\
 \mathcal{M}_{ij} &= \text{Min}_k \left\{ \min (\mathcal{M}_{i, k-1}, b(k-1)) + \mathcal{M}^1_{k, j} \right\} \\
 \mathcal{M}^1_{ij} &= \text{Min}_k \left\{ b + \mathcal{M}^1_{i, j-1}, c + \mathcal{M}'_{ij} \right\}
 \end{aligned}$$

Reconstruction de la structure d'énergie minimale :

$$\begin{aligned}
 \mathcal{M}'_{i,j} &= \text{Min} \left\{ \begin{array}{l} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\ \text{Min}_{i',j'} (E_{BI}(i,i',j',j) + \mathcal{M}'_{i',j'}) \\ a + c + \text{Min}_k (\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1}) \end{array} \right\} \\
 \mathcal{M}_{i,j} &= \text{Min}_k \left\{ \min (\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\} \\
 \mathcal{M}^1_{i,j} &= \text{Min}_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}
 \end{aligned}$$

2. Avec une astuce pour les bulges/boucles internes ...

Reconstruction de la structure d'énergie minimale :

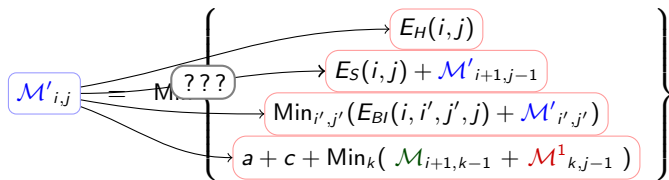
$$\mathcal{M}'_{i,j} = \text{Min} \left\{ \begin{array}{l} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\ \text{Min}_{i',j'} (E_{BI}(i,i',j',j) + \mathcal{M}'_{i',j'}) \\ a + c + \text{Min}_k (\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1}) \end{array} \right\}$$

$$\mathcal{M}_{i,j} = \text{Min}_k \left\{ \min(\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\}$$

$$\mathcal{M}^1_{i,j} = \text{Min}_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}$$

2. Avec une astuce pour les bulges/boucles internes ...

Reconstruction de la structure d'énergie minimale :



$$\mathcal{M}_{i,j} = \text{Min}_k \left\{ \min(\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\}$$

$$\mathcal{M}^1_{i,j} = \text{Min}_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}$$

$\mathcal{O}(n)$ contributeurs potentiels au Min :

⇒ Complexité au pire en $\mathcal{O}(n^2)$ pour un backtrack naïf.

Reconstruction de la structure d'énergie minimale :

$$\mathcal{M}'_{i,j} = \text{Min} \left\{ \begin{array}{l} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\ \text{Min}_{i',j'} (E_{Bl}(i,i',j',j) + \mathcal{M}'_{i',j'}) \\ a + c + \text{Min}_k (\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1}) \end{array} \right\}$$

$$\mathcal{M}_{i,j} = \text{Min}_k \left\{ \min (\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\}$$

$$\mathcal{M}^1_{i,j} = \text{Min}_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}$$

$\mathcal{O}(n)$ contributeurs potentiels au Min :

⇒ Complexité **au pire en $\mathcal{O}(n^2)$** pour un **backtrack naïf**.

Garder les meilleures contributions aux Min ⇒ **Backtrack en $\mathcal{O}(n)$**

Complexités temps/mémoire en $\mathcal{O}(n^3)/\mathcal{O}(n^2)$ pour le précalcul²

2. Avec une astuce pour les bulges/boucles internes ...

Reconstruction de la structure d'énergie minimale :

$$\mathcal{M}'_{i,j} = \text{Min} \left\{ \begin{array}{l} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\ \text{Min}_{i',j'} (E_{BI}(i,i',j',j) + \mathcal{M}'_{i',j'}) \\ a + c + \text{Min}_k (\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1}) \end{array} \right\}$$

$$\mathcal{M}_{i,j} \leftarrow \text{Min}_k \left\{ \min(\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\}$$

$$\mathcal{M}^1_{i,j} \leftarrow \text{Min}_k \left\{ b + \mathcal{M}^1_{i,j-1} + c + \mathcal{M}'_{i,j} \right\}$$

$\mathcal{O}(n)$ contributeurs potentiels au Min :

⇒ Complexité **au pire en $\mathcal{O}(n^2)$** pour un **backtrack naïf**.

Garder les meilleures contributions aux Min ⇒ **Backtrack en $\mathcal{O}(n)$**

Complexités temps/mémoire en $\mathcal{O}(n^3)/\mathcal{O}(n^2)$ pour le précalcul²

⇒ **UnaFold [MZ08]** calcule la structure secondaire d'énergie minimale.

2. Avec une astuce pour les bulges/boucles internes ...

Definition (Repliement ab initio)

Partant de la séquence, trouver la conformation minimisant une fonction d'énergie.

Avantages :

- Explication mécanique
- Complexité raisonnable
 $\mathcal{O}(n^3)/\mathcal{O}(n^2)$ temps/mémoire
- Exploration exhaustive

Limites :

- Pas de cinétique
- Pas d'info évolutive
- Performances limitées

Definition (Approche comparative)

Partant de plusieurs séquences homologues ou d'un alignement, trouver une conformation de score (énergie+alignement) élevé.

Avantages :

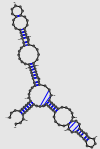
- Meilleures performances
- Affinement permanent

Limites :

- Complexité élevée
- Exploration non-exhaustive

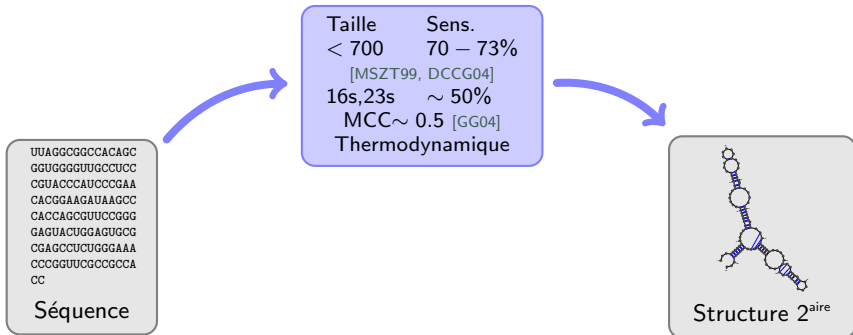
```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAAGCC
CACCAGCGUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCCCA
CC
```

Séquence

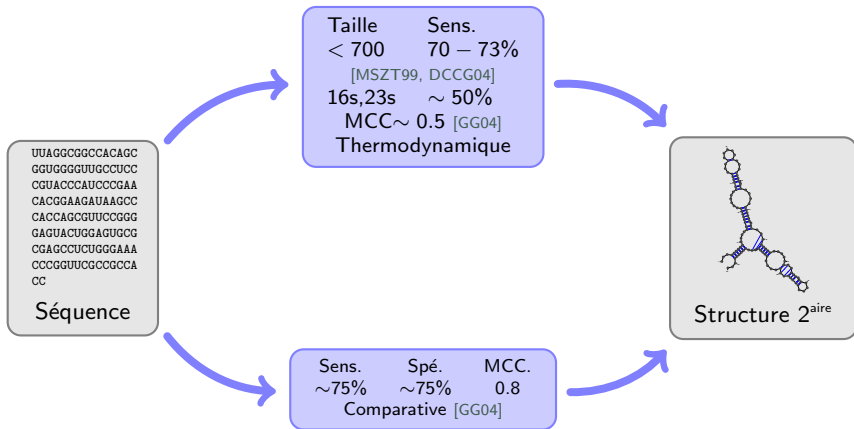


Structure 2^{aire}

$$\text{Rappel : } MCC = \frac{t^+ t^- - f^+ f^-}{\sqrt{(t^+ + f^+)(t^+ + f^-)(t^- + f^+)(t^- + f^-)}}$$



Rappel :
$$MCC = \frac{t^+t^- - f^+f^-}{\sqrt{(t^++f^+)(t^++f^-)(t^-+f^+)(t^-+f^-)}}$$



Rappel :
$$MCC = \frac{t^+ t^- - f^+ f^-}{\sqrt{(t^+ + f^+)(t^+ + f^-)(t^- + f^+)(t^- + f^-)}}$$

But : De la séquence à des modèles tri-dimensionnels !!!

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAGCC
CACCAGCGUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCCA
CC
```

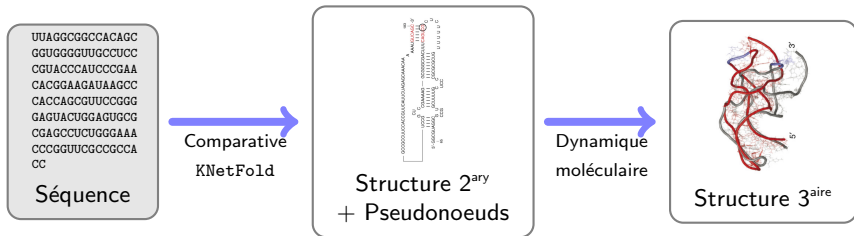
Séquence



Structure 3^{aire}

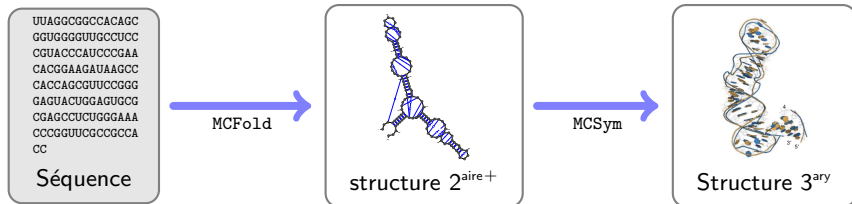
But : De la séquence à des modèles tri-dimensionnels !!!

- Modèles comparatifs + Dynamique moléculaires : RNA2D3D [SYKB07]



But : De la séquence à des modèles tri-dimensionnels !!!

- Pipeline MC-Fold/MC-sym [PM08]

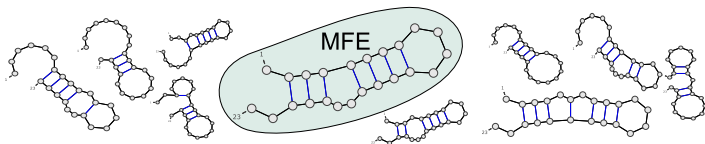


- 1 Introduction
 - Fonction(s) de l'ARN
 - Repliement et structure
 - Représentations de la structure secondaire
- 2 Formalisation du repliement et outils disponibles
 - Aparté thermodynamique
 - Programmation dynamique : Rappels
- 3 Minimisation de l'énergie libre
 - Modèle de Nussinov
 - Modèle de Turner
 - MFold/Unafold
 - Performances et approches comparatives
 - Vers une prédiction ab-initio 3D
- 4 Ensemble de Boltzmann
 - Ensemble de Boltzmann
 - Nussinov : Minimisation \Rightarrow Comptage
 - Calcul de la fonction de partition
 - Échantillonnage statistique

L'ARN *respire* \Rightarrow Il n'existe pas UNE unique conformation native.

Nouveau paradigme

Les conformations d'un ARN *coexistent* dans une *distribution de Boltzmann*.



Conséquence : La probabilité de la MFE peut être négligeable.

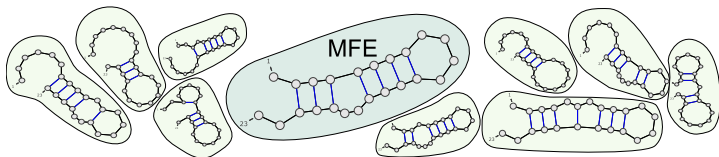
\Rightarrow Comprendre les modes d'actions de l'ARN exige de prendre en considération l'ensemble des structures.

En particulier, des structures proches peuvent se *grouper* et devenir l'hypothèse la plus réaliste dans la recherche d'une conformation fonctionnelle.

L'ARN *respire* \Rightarrow Il n'existe pas UNE unique conformation native.

Nouveau paradigme

Les conformations d'un ARN **coexistent** dans une **distribution de Boltzmann**.



Conséquence : La probabilité de la MFE peut être négligeable.

\Rightarrow Comprendre les modes d'actions de l'ARN exige de prendre en considération l'ensemble des structures.

En particulier, des structures proches peuvent se *grouper* et devenir l'hypothèse la plus réaliste dans la recherche d'une conformation fonctionnelle.

Une distribution de Boltzmann pondère chaque structure S pour un ARN ω par un **facteur de Boltzmann** $\mathcal{B}_{S,\omega} = e^{\frac{-E_{S,\omega}}{RT}}$ où :

- $E_{S,\omega}$ est l'énergie libre de S (kCal.mol⁻¹)
- T est la température (K)
- R est la constante des gaz parfaits (1.986.10⁻³ kCal.K⁻¹.mol⁻¹)

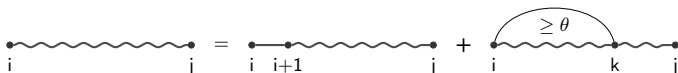
Distribution renormalisée sur \mathcal{S}_ω par la **fonction de partition**

$$\mathcal{Z}_\omega = \sum_{S \in \mathcal{S}_\omega} e^{\frac{-E_{S,\omega}}{RT}}.$$

où \mathcal{S}_ω est l'ensemble des conformations compatibles avec ω .

La **probabilité de Boltzmann** d'une structure S est alors donnée par

$$P_{S,\omega} = \frac{e^{\frac{-E_{S,\omega}}{RT}}}{\mathcal{Z}_\omega}.$$



Récurrance sur l'énergie minimale d'un repliement :

$$N_{i,t} = 0, \quad \forall t \in [i, i + \theta]$$

$$N_{i,j} = \min \begin{cases} N_{i+1,j} & (i \text{ non apparié}) \\ \min_{k=i+\theta+1}^j E_{i,k} + N_{i+1,k-1} + N_{k+1,j} & (i \text{ comp. avec } k) \end{cases}$$

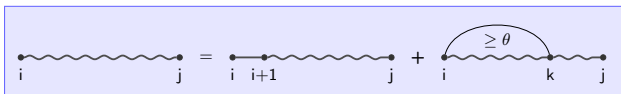
Récurrance de comptage des structures compatibles :

$$C_{i,t} = 1, \quad \forall t \in [j, i + \theta]$$

$$C_{i,j} = \sum \begin{cases} C_{i+1,j} & (i \text{ non apparié}) \\ \sum_{k=i+\theta+1}^j 1 \times C_{i+1,k-1} \times C_{k+1,j} & (i \text{ comp. avec } k) \end{cases}$$

La décomposition est importante, le reste (MFE, comptage...) suit !

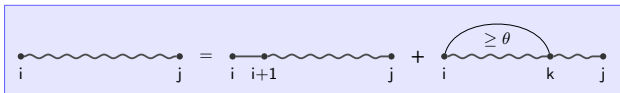
Fonction de partition = Comptage pondéré des structures compatibles



$$Z_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$Z_{i,j} = \sum \left\{ \begin{array}{l} Z_{i+1,j} \\ \sum_{k=i+\theta+1}^j 1 \times Z_{i+1,k-1} \times Z_{k+1,j} \end{array} \right.$$

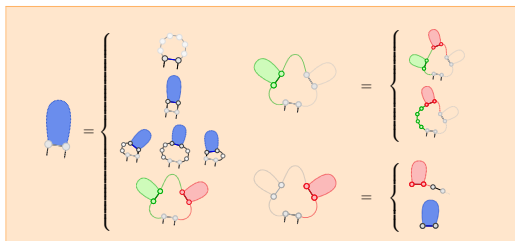
Fonction de partition = Comptage **pondéré** des structures compatibles



$$Z_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

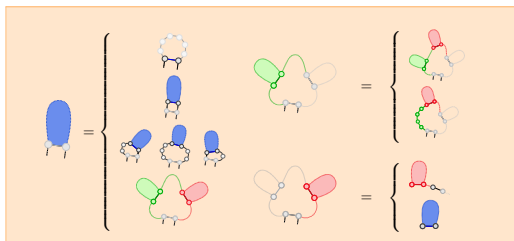
$$Z_{i,j} = \sum \left\{ \begin{array}{l} Z_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{-\frac{E_{bp}(i,k)}{RT}} \times Z_{i+1,k-1} \times Z_{k+1,j} \end{array} \right.$$

Fonction de partition = Comptage pondéré des structures compatibles



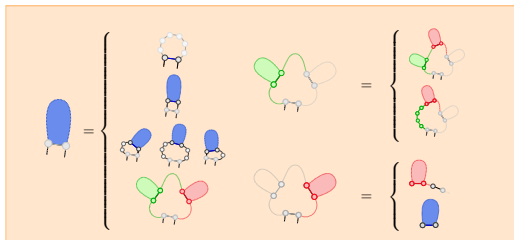
$$\begin{aligned}
 \mathcal{M}'_{i,j} &= \text{Min} \left\{ \begin{array}{l} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\ \text{Min}(E_{BI}(i,i',j',j) + \mathcal{M}'_{i',j'}) \\ a + c + \text{Min}(\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1}) \end{array} \right\} \\
 \mathcal{M}_{i,j} &= \text{Min} \left\{ \text{Min}(\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\} \\
 \mathcal{M}^1_{i,j} &= \text{Min} \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}
 \end{aligned}$$

Fonction de partition = Comptage pondéré des structures compatibles



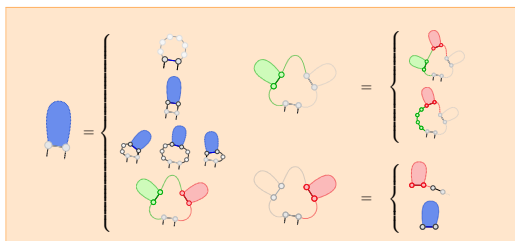
$$\begin{aligned}
 \mathcal{M}'_{ij} &= \text{Min} \left\{ \begin{array}{l} e^{\frac{-E_H(i,j)}{RT}} \\ e^{\frac{-E_S(i,j)}{RT}} + \mathcal{M}'_{i+1,j-1} \\ \text{Min} \left(e^{\frac{-E_{BJ}(i,i',j',j)}{RT}} + \mathcal{M}'_{i'j'} \right) \\ e^{\frac{-(a+c)}{RT}} + \text{Min} (\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{kj-1}) \end{array} \right. \\
 \mathcal{M}_{ij} &= \text{Min} \left\{ \text{Min} \left(\mathcal{M}_{i,k-1}, e^{\frac{-b(k-1)}{RT}} \right) + \mathcal{M}^1_{kj} \right\} \\
 \mathcal{M}^1_{ij} &= \text{Min} \left\{ e^{\frac{-b}{RT}} + \mathcal{M}^1_{i,j-1}, e^{\frac{-c}{RT}} + \mathcal{M}'_{ij} \right\}
 \end{aligned}$$

Fonction de partition = Comptage pondéré des structures compatibles



$$\begin{aligned}
 \mathcal{M}'_{i,j} &= \text{Min} \left\{ \begin{array}{l} e^{\frac{-E_H(i,j)}{RT}} \\ e^{\frac{-E_C(i,j)}{RT}} \mathcal{M}'_{i+1,j-1} \\ \text{Min} \left(e^{\frac{-E_B(i,i',j',j)}{RT}} \mathcal{M}'_{i',j'} \right) \\ e^{\frac{-(a+c)}{RT}} \text{Min} (\mathcal{M}_{i+1,k-1} \mathcal{M}^1_{k,j-1}) \end{array} \right. \\
 \mathcal{M}_{i,j} &= \text{Min} \left\{ \text{Min} \left(\mathcal{M}_{i,k-1}, e^{\frac{-b(k-1)}{RT}} \right) \mathcal{M}^1_{k,j} \right\} \\
 \mathcal{M}^1_{i,j} &= \text{Min} \left\{ e^{\frac{-b}{RT}} \mathcal{M}^1_{i,j-1}, e^{\frac{-c}{RT}} \mathcal{M}'_{i,j} \right\}
 \end{aligned}$$

Fonction de partition = Comptage pondéré des structures compatibles



$$\begin{aligned}
 \mathcal{Z}'(i, j) &= \sum \left\{ \begin{aligned} &e^{-\frac{E_H(i, j)}{RT}} \\ &e^{-\frac{E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \\ &+ \sum \left(e^{-\frac{E_{BI}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \\ &+ e^{-\frac{-(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \end{aligned} \right. \\
 \mathcal{Z}(i, j) &= \sum \left(\mathcal{Z}(i, k-1) + e^{-\frac{b(k-1)}{RT}} \right) \mathcal{Z}^1(k, j) \\
 \mathcal{Z}^1(i, j) &= e^{-\frac{b}{RT}} \mathcal{Z}^1(i, j-1) + e^{-\frac{c}{RT}} \mathcal{Z}'(i, j)
 \end{aligned}$$

Fonction de partition = Comptage pondéré des structures compatibles

$$\begin{aligned} \mathcal{Z}_{i,t} &= 1, \quad \forall t \in [i, i + \theta] \\ \mathcal{Z}_{i,j} &= \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right. \end{aligned}$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma

Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

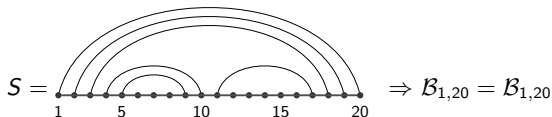
Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right)$$

$$= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT}$$

Exemple :



Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

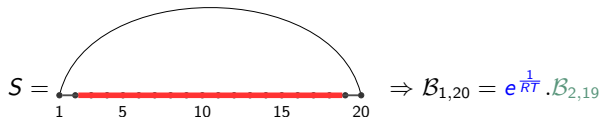
Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right)$$

$$= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT}$$

Exemple :



Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right)$$

$$= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT}$$

Exemple :

$S =$ $\Rightarrow \mathcal{B}_{1,20} = e^{\frac{-1}{RT}} \cdot e^{\frac{1}{RT}} \cdot \mathcal{B}_{3,18}$

Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

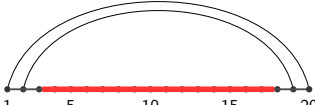
Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right)$$

$$= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT}$$

Exemple :



$$S = \quad \Rightarrow \mathcal{B}_{1,20} = e^{\frac{2}{RT}} \cdot \mathcal{B}_{3,18}$$

Fonction de partition = Comptage pondéré des structures compatibles

$$\begin{aligned}
 \mathcal{Z}_{i,t} &= 1, \quad \forall t \in [i, i + \theta] \\
 \mathcal{Z}_{i,j} &= \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.
 \end{aligned}$$

Validité de la fonction de partition :

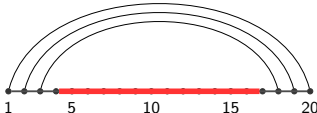
- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

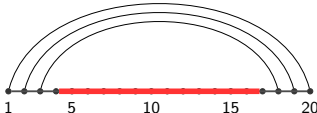
Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\begin{aligned}
 (e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' &= e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \\
 &= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT})
 \end{aligned}$$

Exemple :



$S =$

 $\Rightarrow \mathcal{B}_{1,20} = e^{\frac{3}{RT}} \cdot \mathcal{B}_{4,17}$

Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right)$$

$$= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT}$$

Exemple :

$S =$ $\Rightarrow \mathcal{B}_{1,20} = e^{\frac{4}{RT}} \cdot \mathcal{B}_{5,9} \cdot \mathcal{B}_{11,17}$

Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

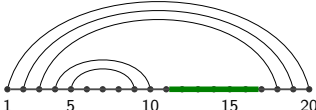
Facteur d'un backtrack = Produit des facteurs de ses parties

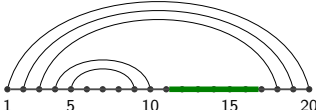
Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right.$$

$$\left. = \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT} \right)$$

Exemple :



$S =$  $\Rightarrow \mathcal{B}_{1,20} = e^{\frac{5}{RT}} \cdot \mathcal{B}_{11,17}$

Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right)$$

$$= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT}$$

Exemple :

$S =$ $\Rightarrow \mathcal{B}_{1,20} = e^{\frac{6}{RT}} \cdot \mathcal{B}_{12,16}$

Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right.$$

$$\left. = \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT} \right)$$

Exemple :

$S =$ $\Rightarrow \mathcal{B}_{1,20} = e^{\frac{6}{RT}} \cdot \mathcal{B}_{13,16}$

Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right)$$

$$= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT}$$

Exemple :

$S =$ $\Rightarrow \mathcal{B}_{1,20} = e^{\frac{6}{RT}} \cdot \mathcal{B}_{14,16}$

Fonction de partition = Comptage pondéré des structures compatibles

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i + \theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \begin{array}{l} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^j e^{\frac{-E_{bp}(i,k)}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{array} \right.$$

Validité de la fonction de partition :

- Exhaustivité/non ambiguïté du schéma
- Correction du facteur de Boltzmann

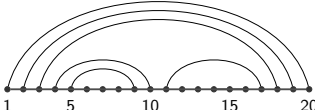
Facteur d'un backtrack = Produit des facteurs de ses parties

Contributions énergétiques passent à l'exposant

$$\left(e^{-a/RT} \cdot \mathcal{Z}^1 \cdot \mathcal{Z}' = e^{-a/RT} \cdot \sum_x e^{-E_x/RT} \cdot \sum_y e^{-E_y/RT} \right)$$

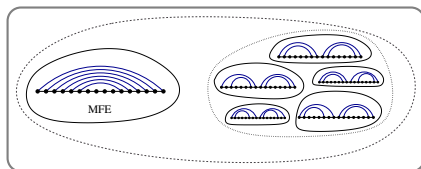
$$= \sum_{x,y} e^{-a/RT} \cdot e^{-E_x/RT} \cdot e^{-E_y/RT} = \sum_{x,y} e^{-(a+E_x+E_y)/RT}$$

Exemple :

$S =$  $\Rightarrow \mathcal{B}_{1,20} = e^{\frac{6}{RT}} = e^{\frac{-E_S}{RT}}$

La MFE (Probabilité maximale) peut être **largement dominée** par un ensemble \mathcal{B} de sous-optimaux **structurellement similaires**.

⇒ Conformation fonctionnelle trouvée plus probablement dans \mathcal{B} .



Expérience : [DCL05]

- Échantillonner des structures selon une probabilité de Boltzmann
- Effectuer un clustering
- Construire structure consensus dans le plus lourd cluster

⇒ Amélioration relative pour spécificité (+17.6%) et sensibilité (+21.74%, sauf Introns du groupe II)

Problème

Comment engendrer des structures dans la distribution de Boltzmann ?

Algorithme (Reformulation SFold [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

$$\mathcal{Z}'(i,j) \stackrel{=}{=} \boxed{???} \begin{cases} \rightarrow e^{-\frac{E_H(i,j)}{RT}} + e^{-\frac{E_S(i,j)}{RT}} \mathcal{Z}'(i+1, j-1) & \text{A} \\ \rightarrow \sum \left(e^{-\frac{E_{BJ}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) & \text{B} \\ \rightarrow e^{-\frac{-(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) & \text{C} \end{cases}$$

Algorithme (Reformulation SFo1d [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- ➊ Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{-\frac{E_H(i, j)}{RT}} + e^{-\frac{E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{-\frac{E_{BI}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{-\frac{-(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$

Algorithme (Reformulation SFold [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- ➊ Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{\frac{-E_H(i, j)}{RT}} + e^{\frac{-E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \\ \sum \left(e^{\frac{-E_{BI}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \\ e^{\frac{-(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \end{array} \right. \begin{array}{l} \textcircled{A} \\ \textcircled{B} \\ \textcircled{C} \end{array}$$

Algorithme (Reformulation SFo1d [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- 1 Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- 2 Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{\frac{-E_H(i, j)}{RT}} + e^{\frac{-E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$

The diagram shows a sequence of states: $A_1, A_2, B_i, B_{i+1}, \dots, B_{j-1}, B_j, C_i, C_{i+1}, \dots, C_{j-1}, C_j$. States A_1, A_2 are blue, $B_i, B_{i+1}, \dots, B_{j-1}, B_j$ are pink, and $C_i, C_{i+1}, \dots, C_{j-1}, C_j$ are green. A curved arrow connects A_1 and A_2 . A box labeled r is positioned above C_i , with a downward arrow pointing to it.

Algorithme (Reformulation SFo1d [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- 1 Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- 2 Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{\frac{-E_H(i, j)}{RT}} + e^{\frac{-E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$

Algorithme (Reformulation SFo1d [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- 1 Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- 2 Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{-\frac{E_H(i, j)}{RT}} + e^{-\frac{E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{-\frac{E_{BI}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{-\frac{-(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$

Algorithme (Reformulation SFo1d [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- 1 Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- 2 Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{-\frac{E_H(i,j)}{RT}} + e^{-\frac{E_S(i,j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{-\frac{E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{-\frac{-(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$

The diagram below the equation shows a sequence of states: $A_1, A_2, B_i, B_{i+1}, \dots, B_{j-1}, B_j, C_i, C_{i+1}, \dots, C_{j-1}, C_j$. A box labeled r is positioned above C_i with a downward arrow pointing to it. Curved arrows connect adjacent states: $A_1 \rightarrow A_2$, $A_2 \rightarrow B_i$, $B_i \rightarrow B_{i+1}$, $B_{i+1} \rightarrow \dots$, $\dots \rightarrow B_{j-1}$, $B_{j-1} \rightarrow B_j$, $B_j \rightarrow C_i$, $C_i \rightarrow C_{i+1}$, $C_{i+1} \rightarrow \dots$, $\dots \rightarrow C_{j-1}$, $C_{j-1} \rightarrow C_j$. A dashed curved arrow also connects B_i to B_j .

Algorithme (Reformulation SFo1d [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- ① Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- ② Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$
- ③ Répéter sur les sous-structures

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{-\frac{E_H(i, j)}{RT}} + e^{-\frac{E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{-\frac{E_{BJ}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{-\frac{(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$

Correction : Tout $S \in \mathcal{S}_w$ engendrée de façon unique (Unambiguïté de Turner)

La probabilité d'engendrer S est donc

$$p_S = \frac{\mathcal{B}(E_1)}{\mathcal{B}(S_w)} \cdot \frac{\mathcal{B}(E_2)}{\mathcal{B}(E_1)} \cdot \frac{\mathcal{B}(E_3)}{\mathcal{B}(E_2)} \cdots \frac{\mathcal{B}(\{S\})}{\mathcal{B}(E_m)}$$

Algorithme (Reformulation SFo1d [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- ① Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- ② Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$
- ③ Répéter sur les sous-structures

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{-\frac{E_H(i, j)}{RT}} + e^{-\frac{E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{-\frac{E_{BJ}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{-\frac{(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$

Correction : Tout $S \in \mathcal{S}_w$ engendrée de façon unique (Unambiguïté de Turner)

La probabilité d'engendrer S est donc

$$p_S = \frac{1}{\mathcal{B}(\mathcal{S}_w)} \cdot \frac{1}{1} \cdot \frac{1}{1} \cdots \frac{\mathcal{B}(\{S\})}{1}$$

Algorithme (Reformulation SFo1d [DL03])

Précalcul : Calculer les matrices (\mathcal{Z} , \mathcal{Z}' , \mathcal{Z}^1) des fonctions de partition.

Remontée stochastique :

- ① Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- ② Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$
- ③ Répéter sur les sous-structures

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{-\frac{E_H(i, j)}{RT}} + e^{-\frac{E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{-\frac{E_{Bj}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{-\frac{(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$

Correction : Tout $S \in \mathcal{S}_\omega$ engendrée de façon unique (Unambiguïté de Turner)

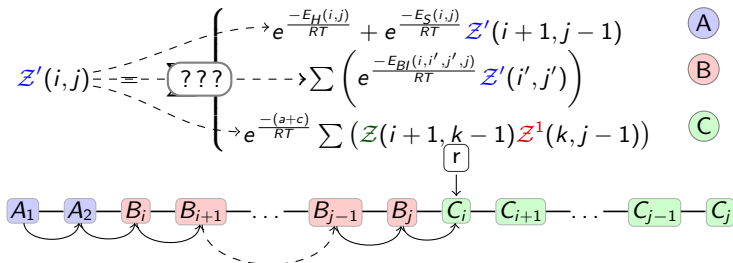
La probabilité d'engendrer S est donc

$$p_S = \frac{\mathcal{B}(\{S\})}{\mathcal{B}(\mathcal{S}_\omega)} = \frac{e^{-E_S/RT}}{\mathcal{Z}} = P_{S, \omega}$$

Complexité ???

Algorithme (Reformulation SFold [DL03])

- 1 Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- 2 Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$
- 3 Répéter sur les sous-structures

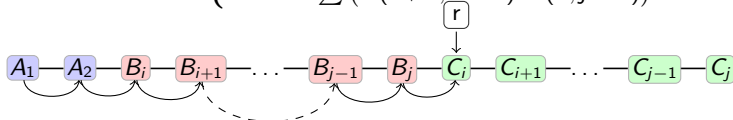


Complexité en moyenne en $\Theta(n\sqrt{n})$ dans l'hypothèse **tout appariement**.
 Adaptation d'un parcours **Boustrophedon** $\Rightarrow \mathcal{O}(n \log nk)$ au pire.

Algorithme (Reformulation SFold [DL03])

- 1 Générer un nombre aléatoire r dans $[0, \mathcal{Z}'(i, j))$
- 2 Retirer à r les contributions à $\mathcal{Z}'(i, j)$, jusqu'à ce que $r < 0$
- 3 Répéter sur les sous-structures

$$\mathcal{Z}'(i, j) = \sum \left\{ \begin{array}{l} e^{-\frac{E_H(i, j)}{RT}} + e^{-\frac{E_S(i, j)}{RT}} \mathcal{Z}'(i+1, j-1) \quad \text{A} \\ \sum \left(e^{-\frac{E_{BL}(i, i', j', j)}{RT}} \mathcal{Z}'(i', j') \right) \quad \text{B} \\ e^{-\frac{(a+c)}{RT}} \sum (\mathcal{Z}(i+1, k-1) \mathcal{Z}^1(k, j-1)) \quad \text{C} \end{array} \right.$$



Après $\Theta(n)$ opérations, on réitère sur un interval de taille $n - 1$
 \Rightarrow Complexité du cas au pire en $\mathcal{O}(n^2 k)$ pour k échantillons

Complexité en moyenne en $\Theta(n\sqrt{n})$ dans l'hypothèse tout appariement.
 Adaptation d'un parcours Boustrophedon $\Rightarrow \mathcal{O}(n \log nk)$ au pire.



A. Condon, B. Davy, B. Rastegari, S. Zhao, and F. Tarrant.

Classifying RNA pseudoknotted structures.

Theoretical Computer Science, 320(1) :35–50, 2004.



K. Doshi, J. J. Cannone, C. Cobaugh, and R. R. Gutell.

Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for rna secondary structure prediction.

BMC Bioinformatics, 5(1) :105, 2004.



Y. Ding, C. Y. Chan, and C. E. Lawrence.

RNA secondary structure prediction by centroids in a boltzmann weighted ensemble.

RNA, 11 :1157–1166, 2005.



Y. Ding and E. Lawrence.

A statistical sampling algorithm for RNA secondary structure prediction.

Nucleic Acids Research, 31(24) :7280–7301, 2003.



P. Gardner and R. Giegerich.

A comprehensive comparison of comparative rna structure prediction approaches.

BMC Bioinformatics, 5(1) :140, 2004.



R. B. Lyngsø and C. N. S. Pedersen.

RNA pseudoknot prediction in energy-based models.

Journal of Computational Biology, 7(3-4) :409–427, 2000.



N. Leontis and E. Westhof.

Geometric nomenclature and classification of RNA base pairs.

RNA, 7 :499–512, 2001.



D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner.

Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure.

J Mol Biol, 288 :911–940, 1999.



Ján Maňuch, Chris Thachuk, Ladislav Stacho, and Anne Condon.

Np-completeness of the direct energy barrier problem without pseudoknots.
pages 106–115, 2009.



N. R. Markham and M. Zuker.

Bioinformatics, chapter UNAFold, pages 3–31.
Springer, 2008.



M. Parisien and F. Major.

The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data.
Nature, 452(7183) :51–55, 2008.



Lioudmila V Sharova, Alexei A Sharov, Timur Nedorezov, Yulan Piao, Nabeebi Shaik, and Minoru S H Ko.

Database for mrna half-life of 19 977 genes obtained by dna microarray analysis of pluripotent and differentiating mouse embryonic stem cells.
DNA Res, 16(1) :45–58, Feb 2009.



B. A. Shapiro, Y. G. Yingling, W. Kasprzak, and E. Bindewald.

Bridging the gap in rna structure prediction.
Curr Opin Struct Biol, 17(2) :157–165, Apr 2007.