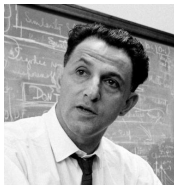


Approches combinatoires et programmation dynamique pour l'étude du repliement de l'ARN

Yann Ponty

Polytechnique/CNRS/INRIA AMIB – France

22 Mars 2010



R. Bellman



C. E. Wilson

Programmation dynamique = Technique algorithmique de résolution des problèmes d'optimisation.

Principe : Partant d'une équation liant le score optimal d'un (sous-)problème à celui de certains de ses sous-problèmes,

- Calculer efficacement le score optimal du problème initial
- Reconstruire la (ou les) solution(s) associée(s) au score optimal

Technique robuste face à des altérations *locales* de la fonction objectif

⇒ **Omniprésente en bioinformatique** : Alignement, repliement...

Difficulté : Trouver l'équation de programmation dynamique

Propriétés requises d'une équation de programmation dynamique =

- 1 Complétude* dans le parcours de l'espace des solutions
- 2 Correction du score optimal hérité des sous-problèmes
- 3 Non-ambiguïté de la décomposition

Modèles combinatoires peuvent aider :

Grammaire/spécification $\rightsquigarrow$ Récurrences de comptage

$\Rightarrow$ Complétude et non-ambiguïté

+ Correction

$\Rightarrow$ Équation de programmation dynamique

Difficulté : Trouver l'équation de programmation dynamique

Propriétés souhaitables d'une équation de programmation dynamique =

- ❶ Complétude* dans le parcours de l'espace des solutions
- ❷ Correction du score optimal hérité des sous-problèmes
- ❸ Non-ambiguïté de la décomposition

Modèles combinatoires peuvent aider :

Grammaire/spécification $\rightsquigarrow$ Récurrences de comptage

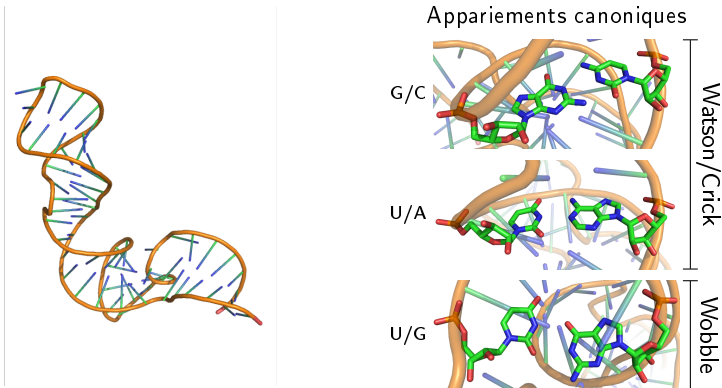
$\Rightarrow$ Complétude et non-ambiguïté

+ Correction

$\Rightarrow$ Équation de programmation dynamique

Repliement de l'ARN

ARN = Polymère linéaire composé de nucléotides (A,C,G,U)



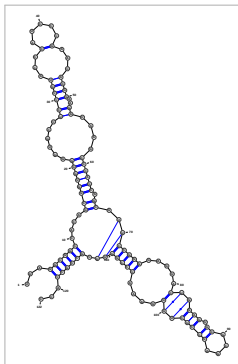
Repliement de l'ARN = Processus stochastique continu dirigé par (résultant en) un appariement des nucléotides.

Structure(s) de l'ARN

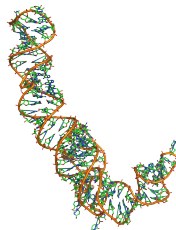
Trois niveaux de représentation :

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCAUCCCGAA
CACGGAAGUAAGCC
CACCAGCGUUCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCA
CC
```

Structure primaire



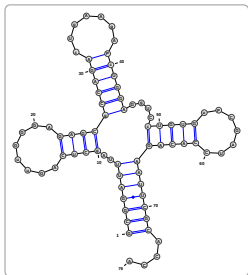
Structure secondaire



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

Structure tertiaire

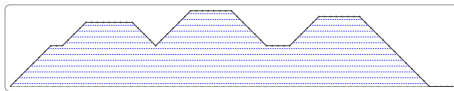
Structure secondaire



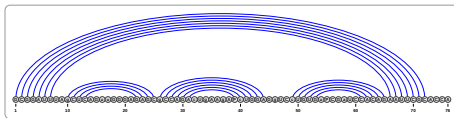
Graphe planaire (outer planar)

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

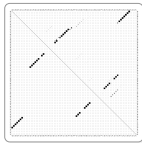
Expression bien parenthésée



Mountain view



Linéaire



Dot plot

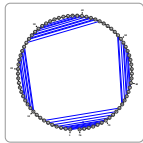


Diagramme de Feynman

Représentation différentes **mais**
Structure combinatoire commune

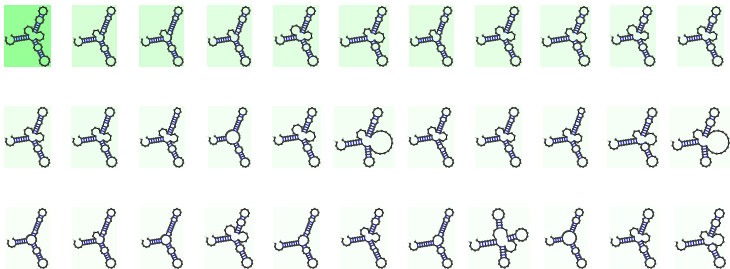
Objectif du repliement

Constat : Données génomiques (séquences) beaucoup plus faciles à obtenir expérimentalement que données structurales (Cristallographie, ...).

⇒ Intérêt à prédire la structure *in silico* (repliement).

Mais une séquence → Nombreuses ($\approx 1.8^n$ [ZS84]) str. sec. compatibles.

Exemple : ARN de transfert



Problème : Quelle est la (ou les) structure(s) fonctionnelle(s), ou native(s) ?

Structures secondaires = Mots bien-parenthésés + *Contrainte stérique* θ

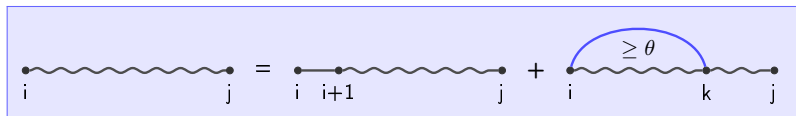


$$W \rightarrow \bullet W + (W_{>\theta}) W + \varepsilon$$

⇒ Série génératrice [Wat78] ($\theta = 1$) :

$$\mathcal{W}(z) = \frac{1 - z + z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4}}{2z^2}$$

Structure secondaire = Mots bien-parenthésés + *Contrainte stérique* θ



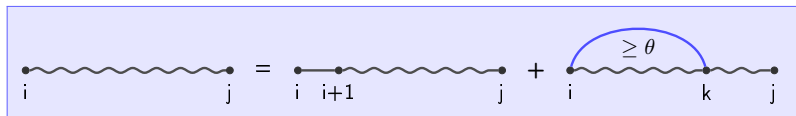
$\Rightarrow$ Comptage des str. sec. compatibles avec un ARN ω :

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

$$N_{i,j} = \sum \left\{ \begin{array}{ll} N_{i+1,j} & i \text{ non apparié} \\ \sum_{k=i+\theta+1}^j \psi_{i,k} \cdot N_{i+1,k-1} \cdot N_{k+1,j} & i \text{ apparié à } k \end{array} \right.$$

où $\psi_{i,k} = 1$ ssi appariement possible des bases ω_i/ω_k , et 0 sinon.

Structure secondaire = Mots bien-parenthésés + *Contrainte stérique* θ



Hypothèse [NJ80] : Parmi toutes les struc. sec., le repliement *fonctionnel* d'un ARN maximise le #paires de bases.

Décomposition engendre toutes les str. sec. compatibles avec ω
 + *Sous-structure optimale* $\Rightarrow$ Changement d'algèbre $(+, \times) \rightarrow (\max, +)$:

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

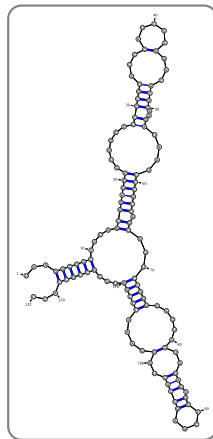
$$N_{i,j} = \max \begin{cases} N_{i+1,j} & i \text{ non apparié} \\ \max_{k=i+\theta+1}^j \psi_{i,k} + N_{i+1,k-1} + N_{k+1,j} & i \text{ apparié à } k \end{cases}$$

Algorithme en $\Theta(n^3)$, 60% des paires prédites sont valides.

Énergie libre additive sur celle des éléments d'une décomposition non-ambiguë en **boucles** de la structure 2^{aie} :

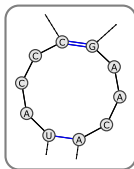
- Boucles internes
- Renflements
- Boucles terminales
- Boucles multiples
- Empilements

Énergies libres déterminées expérimentalement
+ Interpolation pour les grandes boucles
Énergie libre faible → repliement stable
⇒ **Minimiser l'énergie libre.**

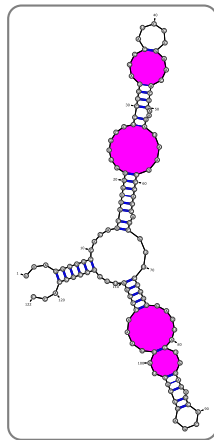


Énergie libre additive sur celle des éléments d'une décomposition non-ambiguë en **boucles** de la structure 2^{aire} :

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

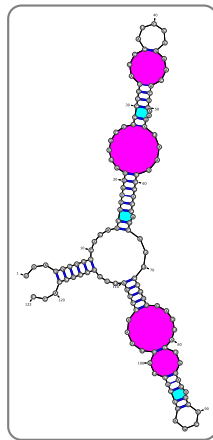
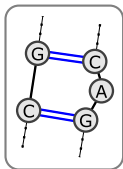


Énergies libres déterminées expérimentalement
+ Interpolation pour les grandes boucles
Énergie libre faible $\rightarrow$ repliement stable
 $\Rightarrow$ **Minimiser l'énergie libre.**



Énergie libre additive sur celle des éléments d'une décomposition non-ambiguë en **boucles** de la structure 2^{aire} :

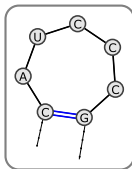
- Boucles internes
- Renflements
- Boucles terminales
- Boucles multiples
- Empilements



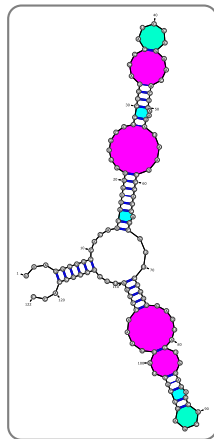
Énergies libres déterminées expérimentalement
+ Interpolation pour les grandes boucles
Énergie libre faible → repliement stable
⇒ **Minimiser l'énergie libre.**

Énergie libre additive sur celle des éléments d'une décomposition non-ambiguë en **boucles** de la structure 2^{aire} :

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

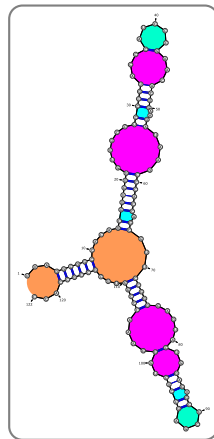
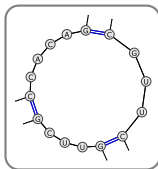


Énergies libres déterminées expérimentalement
+ Interpolation pour les grandes boucles
Énergie libre faible → repliement stable
⇒ **Minimiser l'énergie libre.**



Énergie libre additive sur celle des éléments d'une décomposition non-ambiguë en **boucles** de la structure 2^{aire} :

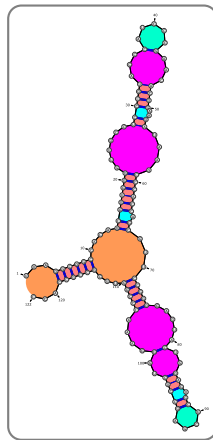
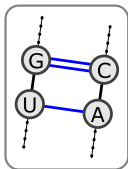
- Boucles internes
- Renflements
- Boucles terminales
- Boucles multiples
- Empilements



Énergies libres déterminées expérimentalement
+ Interpolation pour les grandes boucles
Énergie libre faible → repliement stable
⇒ **Minimiser l'énergie libre.**

Énergie libre additive sur celle des éléments d'une décomposition non-ambiguë en **boucles** de la structure 2^{aire} :

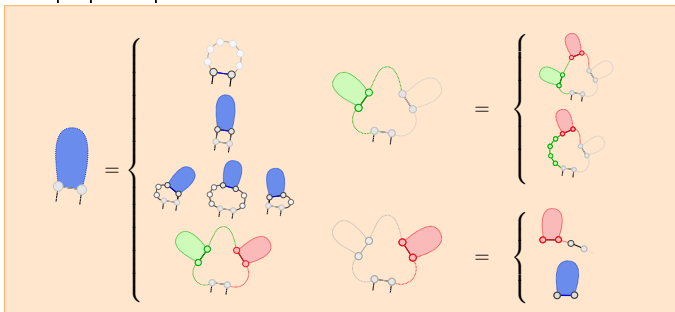
- Boucles internes
- Renflements
- Boucles terminales
- Boucles multiples
- Empilements



Énergies libres déterminées expérimentalement
+ Interpolation pour les grandes boucles
Énergie libre faible → repliement stable
⇒ **Minimiser l'énergie libre.**

Hypothèse [ZS81] : Le repliement *fonctionnel* d'un ARN minimise l'énergie libre.

Grammaire proposée par Zuker *et al* :

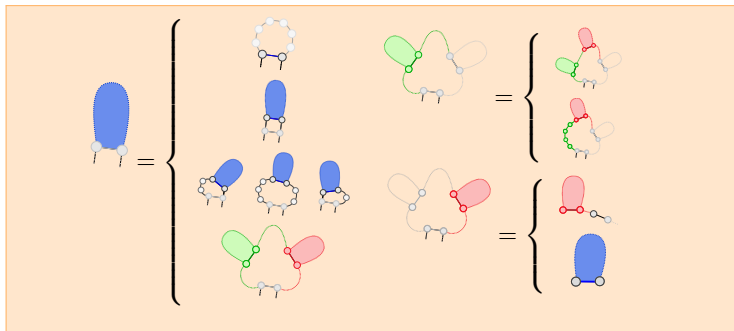


Grammaire non-ambiguë : OK !
 Complète ? Comparons les séries ordinaires ...

Hypothèse [ZS81] : Le repliement *fonctionnel* d'un ARN minimise l'énergie libre.

Rappel : Série génératrice des structures secondaires [Wat78]

$$\mathcal{W}(z) = \frac{1 - z + z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4}}{2z^2}$$



Hypothèse [ZS81] : Le repliement *fonctionnel* d'un ARN minimise l'énergie libre.

Rappel : Série génératrice des structures secondaires [Wat78]

$$\mathcal{W}(z) = \frac{1 - z + z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4}}{2z^2}$$

$$\mathcal{A}(z) = \begin{cases} S(z) \\ z^2 \mathcal{A}(z) \\ zS(z)z^2 \mathcal{A}(z) + z^2 \mathcal{A}(z)S(z)z \\ + zS(z)z^2 \mathcal{A}(z)S(z)z \\ B(z)C(z) \end{cases} \quad \begin{cases} B(z) = \begin{cases} B(z)C(z) \\ S(z)B(z) \end{cases} \\ C(z) = \begin{cases} C(z)z \\ z^2 \mathcal{A}(z) \end{cases} \end{cases}$$

$$S(z) = 1 + zS(z)$$

$$\begin{aligned} \mathcal{A}(z) &= \frac{1 - z - z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4}}{2z^2} \\ &= \mathcal{W}(z) - 1 \quad (\text{Oubli de la str. sec. de longueur 0}) \end{aligned}$$

⇒ Schéma complet, correct par construction ⇒ Minimisation accessible.

Hypothèse [ZS81] : Le repliement *fonctionnel* d'un ARN minimise l'énergie libre.

Méthode : Grammaire $\rightarrow$ Récurrences de comptage
 + Changement d'algèbre $(+, \times) \rightarrow (\min, +)$

Algorithme MFold [ZS81] :

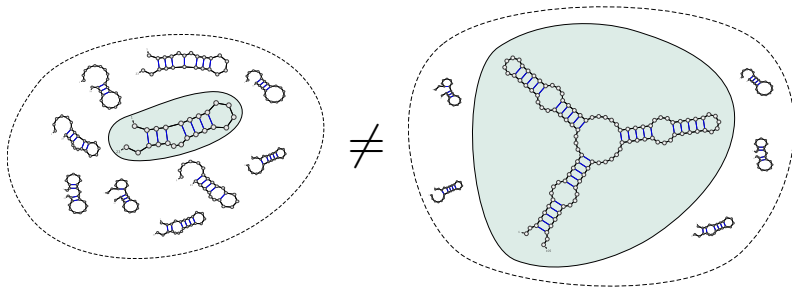
$$\begin{aligned} \mathcal{M}'_{i,j} &= \min \begin{cases} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\ \text{Min}_{i',j'}(E_{BI}(i,i',j',j) + \mathcal{M}'_{i',j'}) \\ a + c + \text{Min}_k(\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1}) \end{cases} \\ \mathcal{M}_{i,j} &= \text{Min}_k \{ \min(\mathcal{M}_{i,k-1},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

- $E_H(i,j)$: Energie de boucle terminale *fermée* par une paire (i,j)
- $E_{BI}(i,j)$: 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.

Prédit 73% des paires de bases correctement.

Méthode publiée puis étendue dans 5 articles cités $\sim 10\,000$ fois !

Constat : Repliement d'énergie libre minimale **n'est pas toujours prédominant**.
Pire, le repliement optimal est parfois faiblement représentatif de l'ensemble.



Idée [McC90] : Supposer une distribution de Boltzmann sur l'ensemble $\mathcal{S}_\omega$ des structures compatibles avec une séquence d'ARN ω .

Str. sec. S , énergie libre $E_{S,\omega}$ $\rightarrow$ Facteur de Boltzmann $\mathcal{B}_{S,\omega} = e^{\frac{-E_{S,\omega}}{RT}}$

Probabilité de Boltzmann $P_{S,\omega} = \frac{\mathcal{B}_{S,\omega}}{\mathcal{Z}_\omega}$

où $\mathcal{Z}_\omega := \sum_{S \in \mathcal{S}_\omega} e^{\frac{-E_{S,\omega}}{RT}}$ est la **fonction de partition** d'un ARN ω .

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

Principe : Pondérer chaque boucle d'énergie libre Δ par son facteur de Boltzmann *local* $e^{\frac{-\Delta}{RT}}$ (Grammaires pondérées [DRT00]).

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

Bonus : Extraction des probabilités d'appariement pour tout couple de bases.

Quelle structure(s) représentante(s) pour l'ensemble de Boltzmann ?

Algorithme **SFold** [DL03]

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

Remontée stochastique : Choisir un des cas proportionnellement à sa contribution, et **réitérer récursivement**.

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

$\Leftrightarrow$ Méthode récursive [Wil77] pour la génération aléatoire pondérée [DRT00].

Ding et al [DCL05] engendrent puis calculent une structure consensus.

$\Rightarrow$ Même sensibilité que MFold mais **gain de 30% sur la spécificité**.

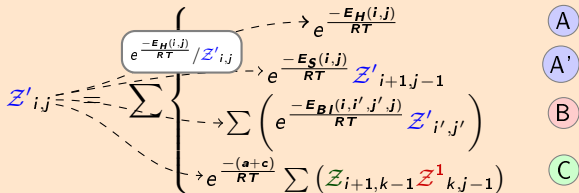
Précalcul en $\mathcal{O}(n^3)$, **génération** en $\mathcal{O}(n^2)$ au pire :

En **moyenne** ? Dans quelle **distribution** ? Comment optimiser ?

Algorithme **SFold** [DL03]

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

Remontée stochastique : Choisir un des cas proportionnellement à sa contribution, et **réitérer récursivement**.



$\Leftrightarrow$ Méthode récursive [Wil77] pour la génération aléatoire pondérée [DRT00].

Ding et al [DCL05] engendrent puis calculent une structure consensus.

$\Rightarrow$ Même sensibilité que MFold mais **gain de 30% sur la spécificité**.

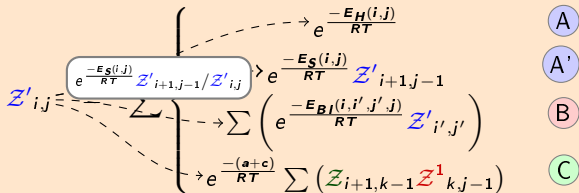
Précalcul en $\mathcal{O}(n^3)$, **génération** en $\mathcal{O}(n^2)$ au pire :

En **moyenne** ? Dans quelle **distribution** ? Comment optimiser ?

Algorithme **SFold** [DL03]

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

Remontée stochastique : Choisir un des cas proportionnellement à sa contribution, et **réitérer récursivement**.



$\Leftrightarrow$ Méthode récursive [Wil77] pour la génération aléatoire pondérée [DRT00].

Ding et al [DCL05] engendrent puis calculent une structure consensus.

$\Rightarrow$ Même sensibilité que MFold mais **gain de 30% sur la spécificité**.

Précalcul en $\mathcal{O}(n^3)$, **génération** en $\mathcal{O}(n^2)$ au pire :

En **moyenne** ? Dans quelle **distribution** ? Comment optimiser ?

Modèle Homopolymère : Toutes paires de bases autorisées.

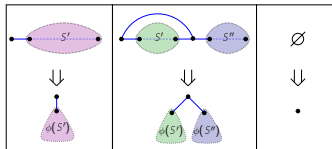
Distribution de Boltzmann : Basée sur $bp(\cdot)$ #paires de bases (Nussinov/Jacobson).

$$\text{Structure } S \rightarrow \text{Probabilité } \frac{e^{\frac{bp(S)}{RT}}}{Z}$$

Theorem

Soit n la longueur de l'ARN et k le nombre d'échantillons engendrés.

La **complexité en moyenne** de l'échantillonnage est en $\Theta(n^3 + kn\sqrt{n})$.



Remarque 1 : Le coût de génération $c(S)$ d'une structure S est dominé par la somme des tailles des fils gauches $t \in \mathcal{T}_g(S)$ de l'arbre unaire/binaire associé :

$$c(S) = n - \#bp(S)/2 + \sum_{t \in \mathcal{T}_g(S)} |t| - \theta$$

Modèle Homopolymère : Toutes paires de bases autorisées.

Distribution de Boltzmann : Basée sur $bp(\cdot)$ #paires de bases (Nussinov/Jacobson).

$$\text{Structure } S \rightarrow \text{Probabilité } \frac{e^{\frac{bp(S)}{RT}}}{Z}$$

Theorem

Soit n la longueur de l'ARN et k le nombre d'échantillons engendrés.

La **complexité en moyenne** de l'échantillonnage est en $\Theta(n^3 + kn\sqrt{n})$.

On s'intéresse à l'espérance de la v.a. X **coût d'une génération**. On a alors

$$\mathbb{E}(X \mid n) = \sum_{|S|=n} c(S) \cdot \frac{e^{\frac{bp(S)}{RT}}}{Z_n} = \frac{\sum_{|S|=n} c(S) \cdot e^{\frac{bp(S)}{RT}}}{\sum_{|S|=n} e^{\frac{bp(S)}{RT}}}.$$

On sépare notre étude en introduisant deux s.g. $C(z)$ et $P_f(z)$ telles que

$$C(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} c(S) z^{|S|} \quad \text{et} \quad P_f(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} z^{|S|}.$$

On note alors que $\mathbb{E}(X \mid n) = [z^n] C(z) / [z^n] P_f(z)$.

Modèle Homopolymère : Toutes paires de bases autorisées.

Distribution de Boltzmann : Basée sur $bp(\cdot)$ #paires de bases (Nussinov/Jacobson).

$$\text{Structure } S \rightarrow \text{Probabilité } \frac{e^{\frac{bp(S)}{RT}}}{Z}$$

Theorem

Soit n la longueur de l'ARN et k le nombre d'échantillons engendrés.

La **complexité en moyenne** de l'échantillonnage est en $\Theta(n^3 + kn\sqrt{n})$.

$C(z)$ et $P_f(z)$ sont alors les solutions positives du système

$$\begin{aligned} C(z) &= z(P_f(z) + C(z)) + z^2 e^{\frac{1}{RT}} (1 - \theta) P_f^{\geq \theta}(z) P_f(z) \\ &\quad + z^3 e^{\frac{1}{RT}} \frac{\partial P_f^{\geq \theta}(z)}{\partial z} P_f(z) + z^2 e^{\frac{1}{RT}} C^{\geq \theta}(z) P_f(z) \\ &\quad + z^2 e^{\frac{1}{RT}} P_f^{\geq \theta}(z) C(z) \\ P_f(z) &= z^2 e^{\frac{1}{RT}} P_f^{\geq \theta}(z) P_f(z) + z P_f(z) + 1 \end{aligned}$$

où $S^{\geq \theta}(z)$ est la s.g. $S(z)$ privée de ses termes de degré $< \theta$.

Modèle Homopolymère : Toutes paires de bases autorisées.

Distribution de Boltzmann : Basée sur $bp(\cdot)$ #paires de bases (Nussinov/Jacobson).

$$\text{Structure } S \rightarrow \text{Probabilité } \frac{e^{\frac{bp(S)}{RT}}}{Z}$$

Theorem

Soit n la longueur de l'ARN et k le nombre d'échantillons engendrés.

La **complexité en moyenne** de l'échantillonnage est en $\Theta(n^3 + kn\sqrt{n})$.

⇒ Grosses séries génératrices un peu complexes (Merci Maple !). . .

+ **Analyse de singularités** :

$$[z^n]P_f(z) \sim \frac{\kappa}{\rho^n n\sqrt{n}}(1 + \mathcal{O}(1/n)) \quad [z^n]C(z) \sim \frac{\kappa'}{\rho^n}(1 + \mathcal{O}(1/\sqrt{n}))$$

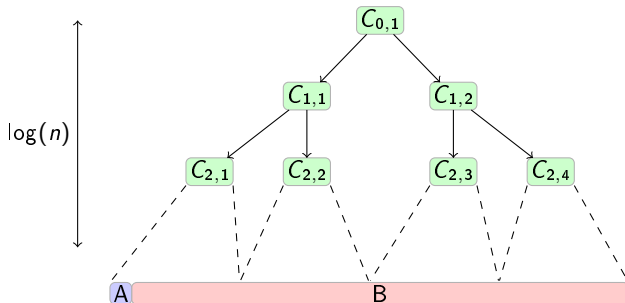
On obtient ainsi un équivalent asymptotique pour $\mathbb{E}(X \mid n)$:

$$\mathbb{E}(X \mid n) = \frac{[z^n]C(z)}{[z^n]P_f(z)} \sim \frac{\kappa'}{\kappa} n\sqrt{n}(1 + \mathcal{O}(1/\sqrt{n})).$$



En moyenne, on perd du temps à tester des décompositions infructueuses !

- Organisation hiérarchique des contributions



⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$, mémoire en $\Theta(n^3)$.

- Parcours Boustrophédon



Décomposition inégale

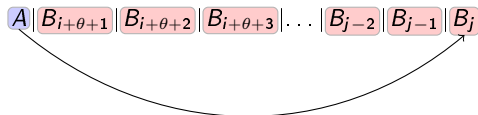
⇒ Coût immédiat faible, récursion sur intervalle de taille $\mathcal{O}(j-i-c)$.

En moyenne, on perd du temps à tester des décompositions infructueuses !

- Organisation hiérarchique des contributions

⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$, mémoire en $\Theta(n^3)$.

- Parcours Boustrophédon



Décomposition inégale

⇒ Coût *immédiat* faible, récursion sur intervalle de taille $\mathcal{O}(j-i-c)$.

Décomposition égale ⇒ Coût *immédiat* élevé, diviser pour régner.

Complexité au pire solution de

$$f(n) = \max_{k \in [1, n-1]} (f(k) + f(n-k) + 2 \min(k, n-k)) \in \mathcal{O}(n \log(n)) \text{ [FZV94]}.$$

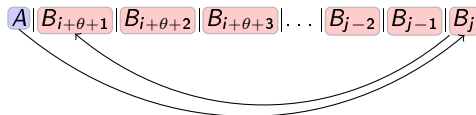
⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$ sans précalcul supplémentaire.

En moyenne, on perd du temps à tester des décompositions infructueuses !

- Organisation hiérarchique des contributions

⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$, mémoire en $\Theta(n^3)$.

- Parcours Boustrophédon



Décomposition inégale

⇒ Coût *immédiat* faible, récursion sur intervalle de taille $\mathcal{O}(j-i-c)$.

Décomposition égale ⇒ Coût *immédiat* élevé, diviser pour régner.

Complexité au pire solution de

$$f(n) = \max_{k \in [1, n-1]} (f(k) + f(n-k) + 2 \min(k, n-k)) \in \mathcal{O}(n \log(n)) \text{ [FZV94].}$$

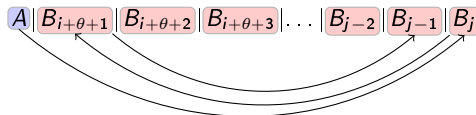
⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$ sans précalcul supplémentaire.

En moyenne, on perd du temps à tester des décompositions infructueuses !

- Organisation hiérarchique des contributions

⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$, mémoire en $\Theta(n^3)$.

- Parcours Boustrophédon



Décomposition inégale

⇒ Coût *immédiat* faible, récursion sur intervalle de taille $\mathcal{O}(j-i-c)$.

Décomposition égale ⇒ Coût *immédiat* élevé, diviser pour régner.

Complexité au pire solution de

$$f(n) = \max_{k \in [1, n-1]} (f(k) + f(n-k) + 2 \min(k, n-k)) \in \mathcal{O}(n \log(n)) \text{ [FZV94]}.$$

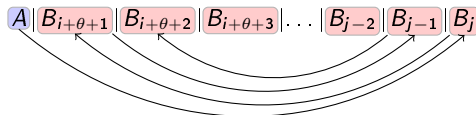
⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$ sans précalcul supplémentaire.

En moyenne, on perd du temps à tester des décompositions infructueuses !

- Organisation hiérarchique des contributions

⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$, mémoire en $\Theta(n^3)$.

- Parcours Boustrophédon



Décomposition inégale

⇒ Coût *immédiat* faible, récursion sur intervalle de taille $\mathcal{O}(j-i-c)$.

Décomposition égale ⇒ Coût *immédiat* élevé, diviser pour régner.

Complexité au pire solution de

$$f(n) = \max_{k \in [1, n-1]} (f(k) + f(n-k) + 2 \min(k, n-k)) \in \mathcal{O}(n \log(n)) \text{ [FZV94]}.$$

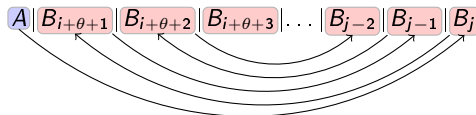
⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$ sans précalcul supplémentaire.

En moyenne, on perd du temps à tester des décompositions infructueuses !

- Organisation hiérarchique des contributions

⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$, mémoire en $\Theta(n^3)$.

- Parcours Boustrophédon



Décomposition inégale

⇒ Coût *immédiat* faible, récursion sur intervalle de taille $\mathcal{O}(j-i-c)$.

Décomposition égale ⇒ Coût *immédiat* élevé, diviser pour régner.

Complexité au pire solution de

$$f(n) = \max_{k \in [1, n-1]} (f(k) + f(n-k) + 2 \min(k, n-k)) \in \mathcal{O}(n \log(n)) \text{ [FZV94].}$$

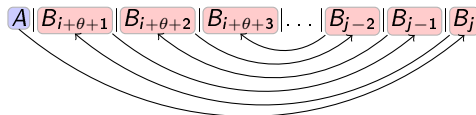
⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$ sans précalcul supplémentaire.

En moyenne, on perd du temps à tester des décompositions infructueuses !

- Organisation hiérarchique des contributions

⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$, mémoire en $\Theta(n^3)$.

- Parcours Boustrophédon



Décomposition inégale

⇒ Coût *immédiat* faible, récursion sur intervalle de taille $\mathcal{O}(j-i-c)$.

Décomposition égale ⇒ Coût *immédiat* élevé, diviser pour régner.

Complexité au pire solution de

$$f(n) = \max_{k \in [1, n-1]} (f(k) + f(n-k) + 2 \min(k, n-k)) \in \mathcal{O}(n \log(n)) \text{ [FZV94].}$$

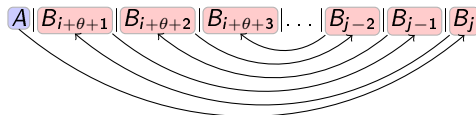
⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$ sans précalcul supplémentaire.

En moyenne, on perd du temps à tester des décompositions infructueuses !

- Organisation hiérarchique des contributions

⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$, mémoire en $\Theta(n^3)$.

- Parcours Boustrophédon



Décomposition inégale

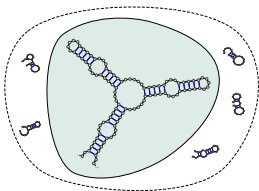
⇒ Coût *immédiat* faible, récursion sur intervalle de taille $\mathcal{O}(j-i-c)$.

Décomposition égale ⇒ Coût *immédiat* élevé, diviser pour régner.

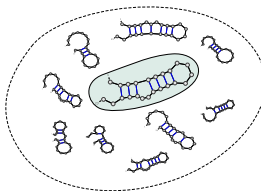
Complexité **au pire** solution de

$$f(n) = \max_{k \in [1, n-1]} (f(k) + f(n-k) + 2 \min(k, n-k)) \in \mathcal{O}(n \log(n)) \text{ [FZV94].}$$

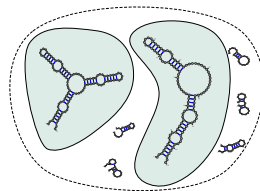
⇒ Génération en $\mathcal{O}(n^3 + kn \log(n))$ **sans précalcul supplémentaire**.



Repliement fonctionnel ?



Repliement mal défini :
ARNm ?



ARN bi-stable ou
phénomène cinétique ?

Constat : Plusieurs scénarios possibles pour l'ensemble de Boltzmann.

⇒ Extraire des indicateurs numériques discriminant les scénarios.

Idée : Marquer occurrences de boucles et extraire les espérances, variances ...

Problème : Contraintes d'appariements induisent un *manque de symétrie*.

Série génératrices et dérivées partielles difficilement accessibles ...

⇒ Utiliser un (ou plusieurs) pointage(s) partiel(s) [DFLS04] de la grammaire.

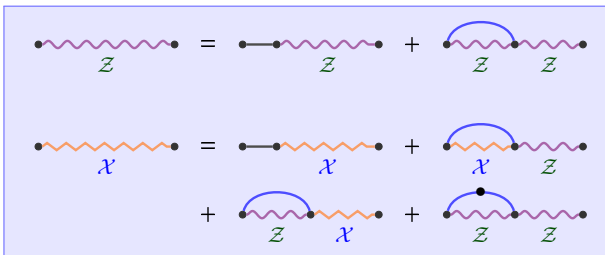
Exemple

Objectif : Espérance E_ω du nombre de paires de bases $bp(\cdot)$.

$$E_\omega = \sum_{s \in \mathcal{S}_\omega} bp(s) \cdot p_s = \frac{\sum_{s \in \mathcal{S}_\omega} bp(s) \cdot e^{\frac{-E_{\omega,s}}{RT}}}{\mathcal{Z}_\omega} = \frac{\chi_\omega}{\mathcal{Z}_\omega}.$$

où $\mathcal{Z}_\omega$ est la fonction de partition et χ_ω le coeff. d'une dérivée partielle.

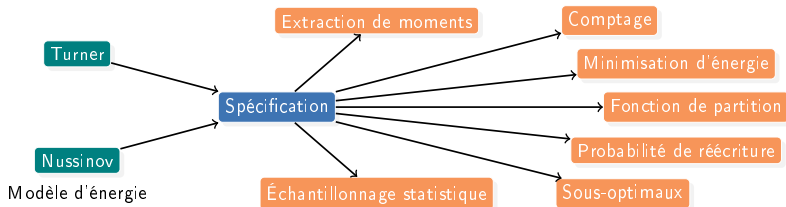
Dérivée connaît un analogue au niveau des grammaires hors-contexte [DFLS04].



$\Rightarrow$ Calcul en $\Theta(n^3)$ de χ_ω puis E_ω est à portée de main.

(Grammaire $\rightarrow$ Réc. comptage $\rightarrow$ Contraintes d'appariements)

Remarque : Généralise un algorithme similaire pour l'énergie libre [MMN05].



Limites liées à l'absence de symétrie (appariements) → Difficile de :

- Générer par Boltzmann [DFLS04] dans l'ensemble de Boltzmann.
- Récurrences linéaires à coeff. polynômiaux pour la fonction de partition.
- Supposer des distributions normales (Grammaire fortement connexe)

Perspectives :

- Quel sous-ensemble de structures domine l'ensemble de Boltzmann ?
~ Collisions lors de l'échantillonnage (En cours avec D. Gardy)
- Étendre les espaces de conformations : Pseudonoeuds (cf C. Saule)
- Grammaires non-suffisantes → Hypergraphes ?
- Design, i.e. génération d'une séquence ayant un repliement donné.



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.



P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer.

Boltzmann samplers for the random generation of combinatorial structures.
Combinatorics, Probability, and Computing, 13(4–5):577–625, 2004.
Special issue on Analysis of Algorithms.



Y. Ding and E. Lawrence.

A statistical sampling algorithm for RNA secondary structure prediction.
Nucleic Acids Research, 31(24):7280–7301, 2003.



A. Denise, O. Roques, and M. Termier.

Random generation of words of context-free languages according to the frequencies of letters.
In D. Gardy and A. Mokkadem, editors, *Mathematics and Computer Science: Algorithms, Trees, Combinatorics and probabilities*, Trends in Mathematics, pages 113–125. Birkhäuser, 2000.



P. Flajolet, P. Zimmermann, and B. Van Cutsem.

Calculus for the random generation of labelled combinatorial structures.
Theoretical Computer Science, 132:1–35, 1994.



J.S. McCaskill.

The equilibrium partition function and base pair binding probabilities for RNA secondary structure.
Biopolymers, 29:1105–1119, 1990.



István Miklós, Imtraud M Meyer, and Borbála Nagy.

Moments of the boltzmann distribution for rna secondary structures.
Bull Math Biol, 67(5):1031–1047, Sep 2005.



R. Nussinov and A.B. Jacobson.

Fast algorithm for predicting the secondary structure of single-stranded RNA.
Proc Natl Acad Sci U S A, 77:6903–13, 1980.



M. S. Waterman.

Secondary structure of single stranded nucleic acids.
Advances in Mathematics Supplementary Studies, 1(1):167–212, 1978.



H. S. Wilf.

A unified setting for sequencing, ranking, and selection algorithms for combinatorial objects.
Advances in Mathematics, 24:281–291, 1977.



M. Zuker and P. Stiegler.

Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information.
Nucleic Acids Res., 9:133–148, 1981.



M. Zuker and D. Sankoff.

Rna secondary structures and their prediction.
Bull Math Bio, 46:591–621, 1984.