# Design d'ARN et génération aléatoire

Bonnie Berger[⋆] Alain Denise[♣] Srinivas Devadas[⋆]
Alex Levin[⋆] Mieszko Lis[⋆] **Yann Ponty**[•] Charles O'Donnell[⋆]
Vladimir Reinharz[‡] Stéphane Vialette[†] Jérôme Waldispühl[‡]
Yi Zhang[♢] Yu Zhou[♣,♢,⊙]

[⋆]CSail, MIT, USA

[♣]LRI, Univ. Paris-Sud, Orsay F-91405, France

[♢]State Key Laboratory of Virology, Wuhan University, China

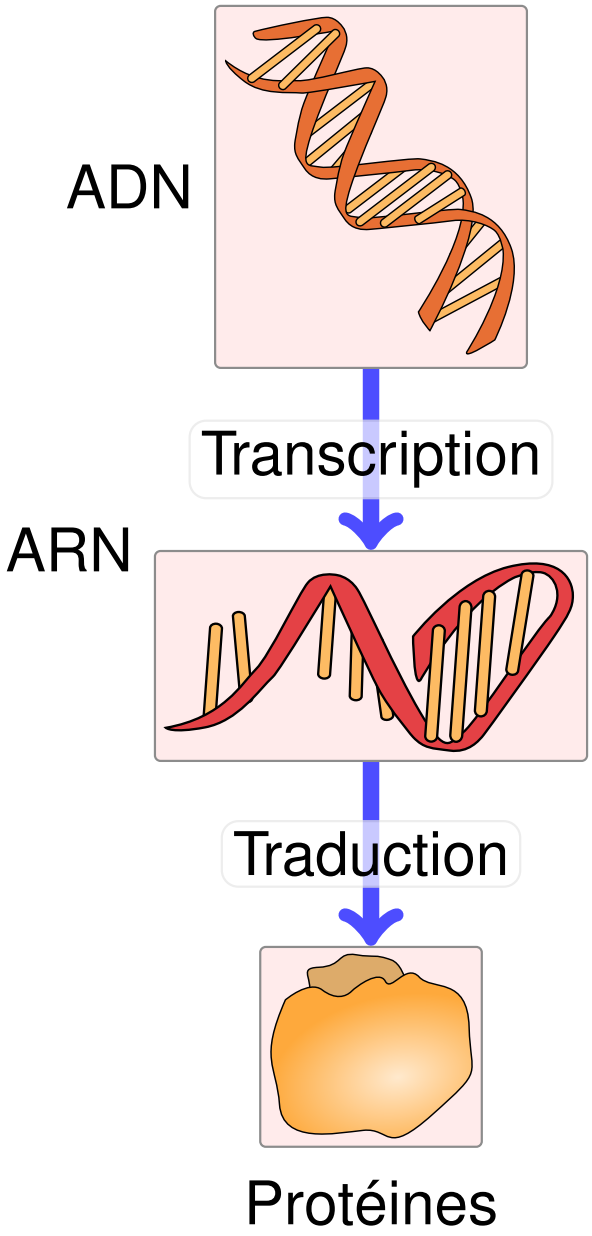
[⊙] Cellular and Molecular Medicine, Univ. of California San Diego, USA

[•] LIX, CNRS/Ecole Polytechnique, France + PIMS, Vancouver, Canada

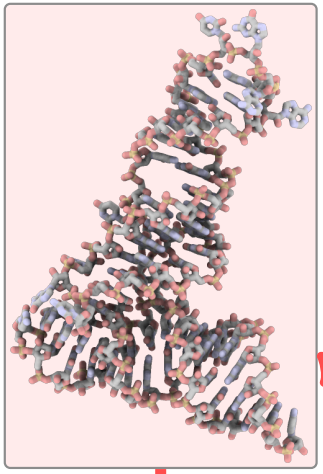
[†] LIGM, Université Paris-Est, France

[‡] Computer Science, Mc Gill University, Canada

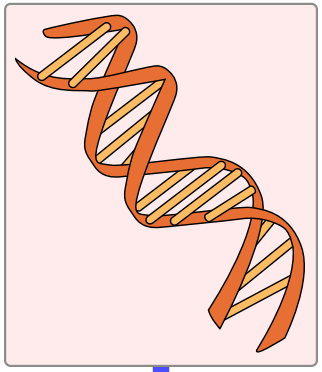
# Dogme fondamental de la biologie moléculaire

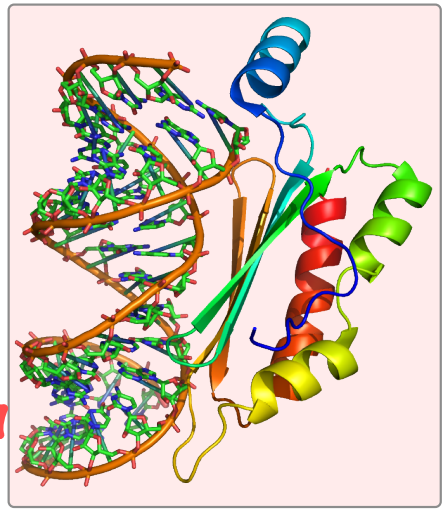


# Dogme fondamental de la biologie moléculaire



ADN



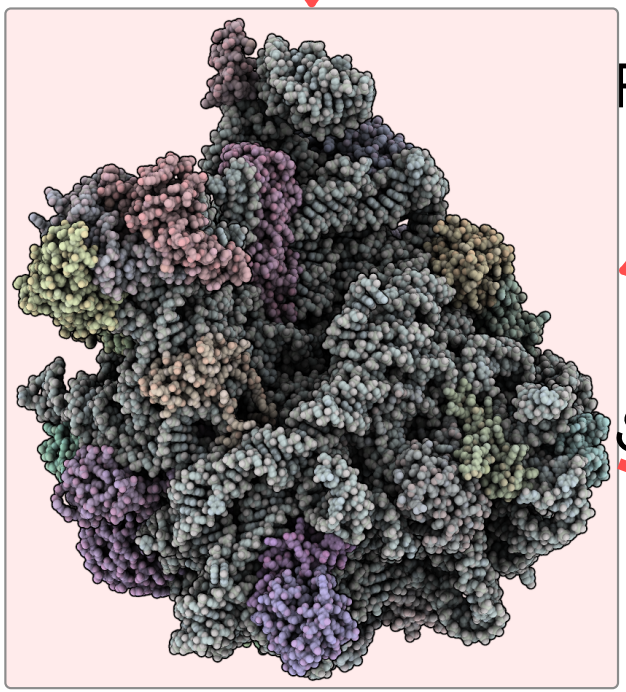


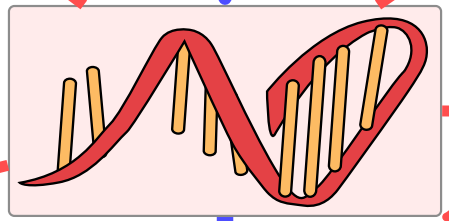
Transfert

Transcription

Maturation

RN

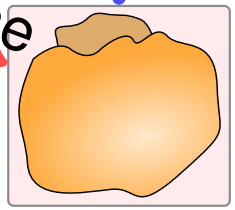




Régulation

Traduction

Synthèse



Protéines

Rôles de l'ARN :

- Messager
- Acteur traduction
- Régulateur
- Catalyseur
- Epissage
- ...

# Motivation d'un design contraint d'ARN

**Motivation :** Impact de la structure sur les stimulateurs de l'épissage (ESE).

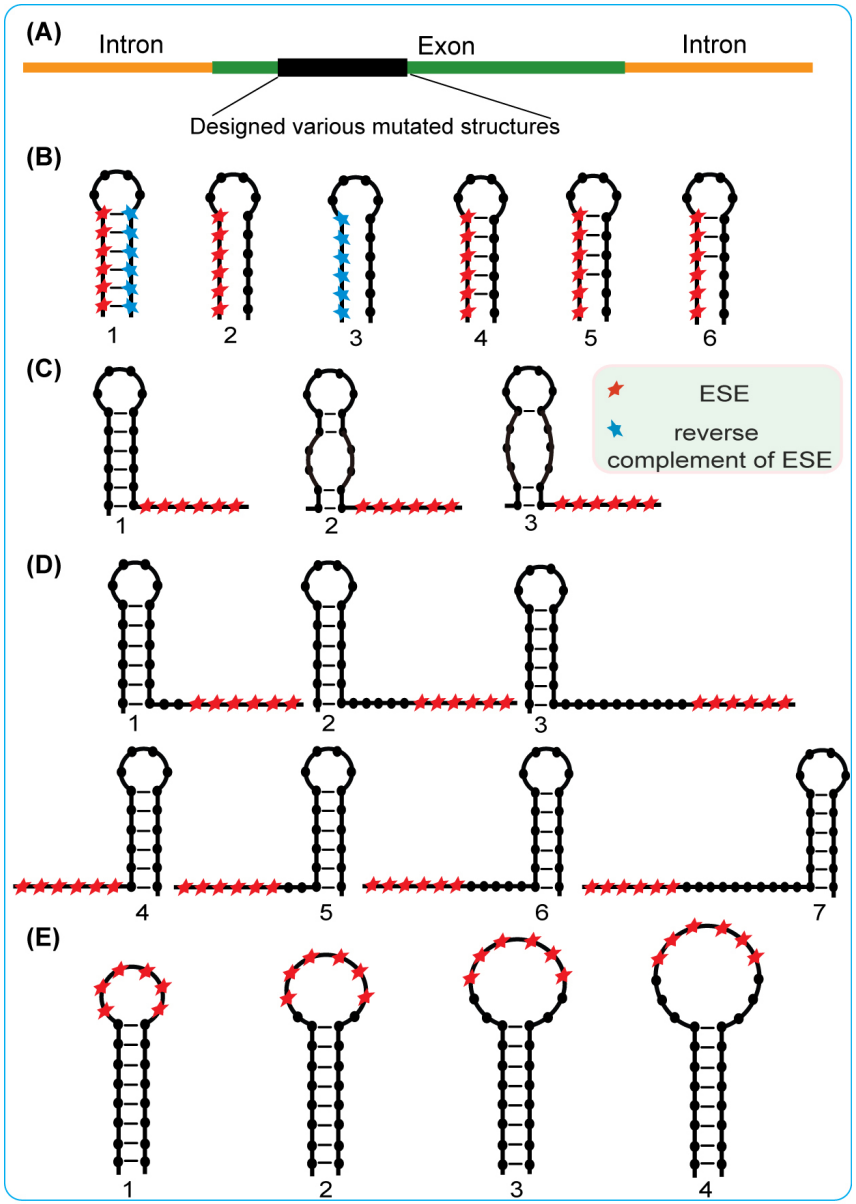
- Structures différentes ;
- Présence localisée d'ESE ;
- Eviter autres ESE ($\sim$1500 !)

Validation d'un effet du contexte structurel sur l'efficacité des ESE.

[Liu *et al*, FEBS Lett. 2010]

**Objectif :** Concevoir séq. d'ARN

1. Se repliant correctement
2. Présentant/évitant des motifs



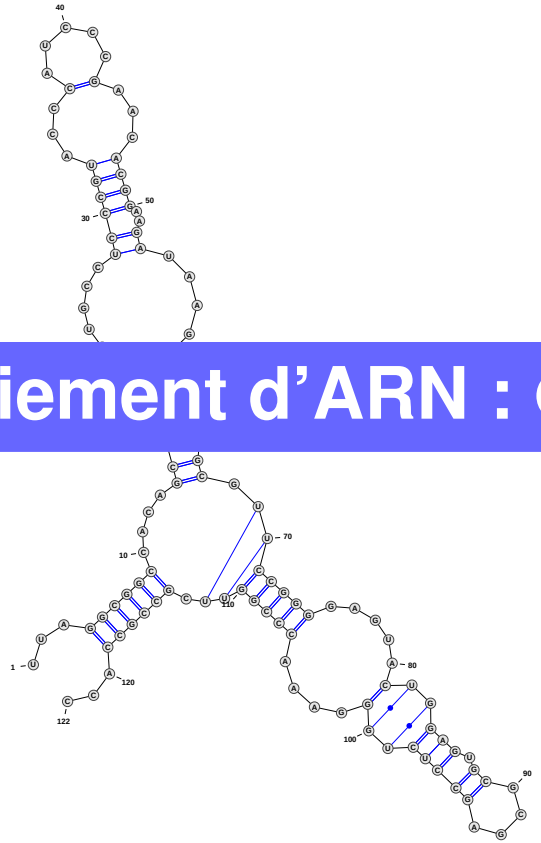
# I. Repliement inverse

## Garantir la structure

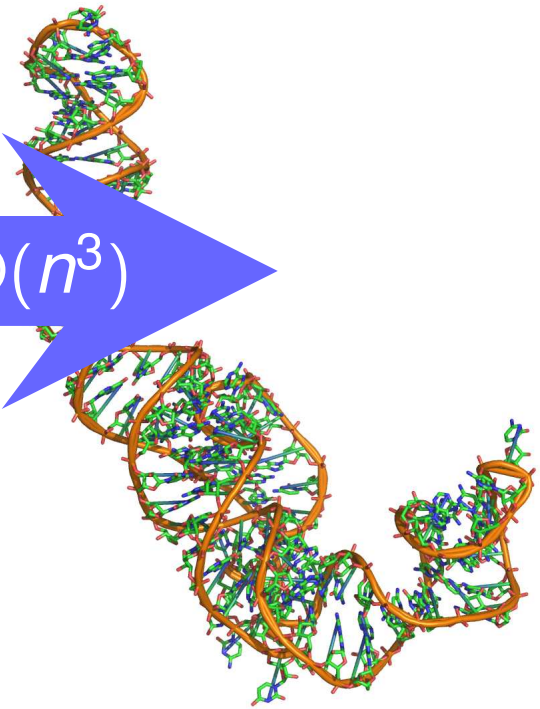
# Séquence et structure(s) de l'ARN

**ARN** = Polymère linéaire = Séquence sur $\{A, C, G, U\}^\star$

UUAGGCGGCCACAGC
GGUGGGGU
CGUACCCA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

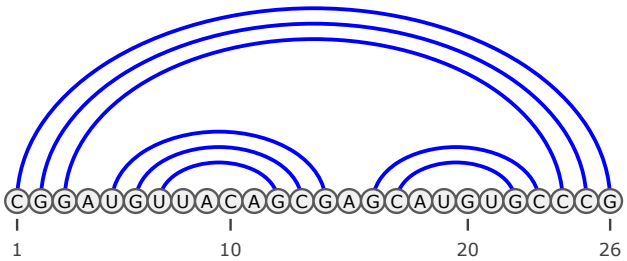


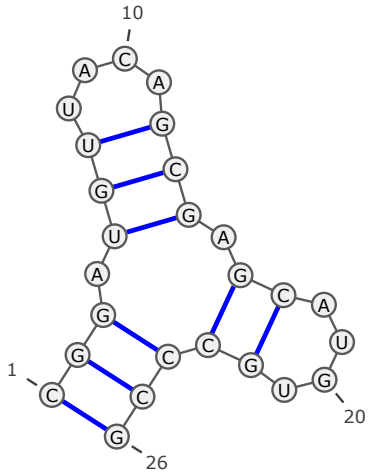
**Repliement d'ARN : $\Theta(n^3)$**



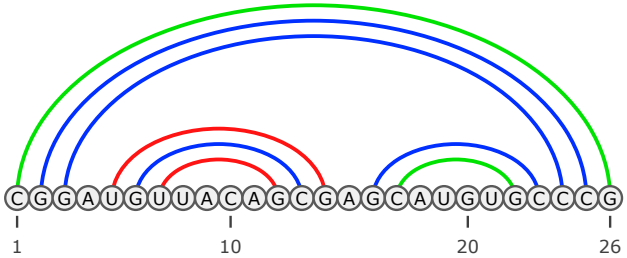
Structure Primaire | Structure Secondaire | Structure Tertiaire

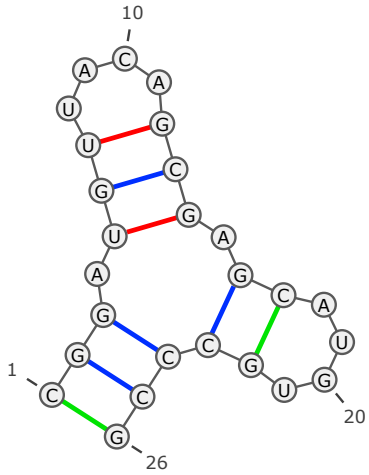
5s rRNA (PDBID : 1K73 :B)



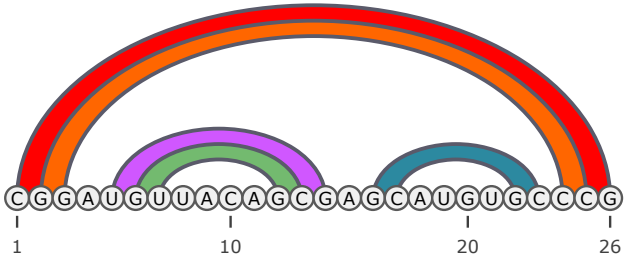



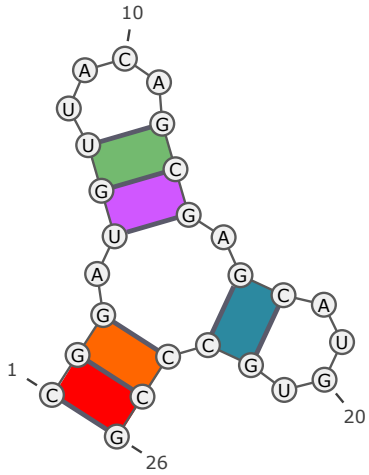

- **Structure d'ARN $S$ :** Couplage**sans croisement** d'un ARN $w$
- **Motifs** séquence/structure (ex. Paires, Empilements, Boucles . . . )
- **Modèle d'énergie :**
  - **Motif** $\rightarrow$ Contribution à l'énergie libre $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  - **Energie libre** $E_w(S)$ **:** Somme $\Delta$ sur les motifs de $S$



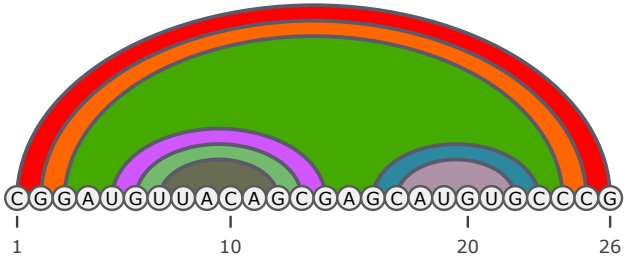



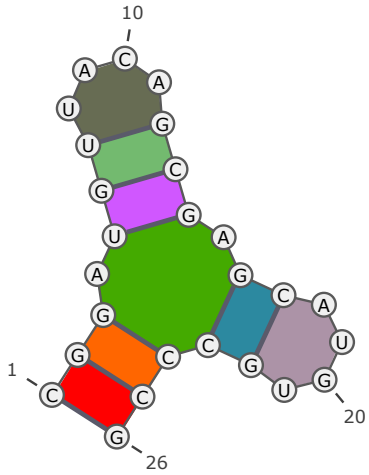

- **Structure d'ARN $S$ :** Couplage**sans croisement** d'un ARN $w$
- **Motifs** séquence/structure (ex. Paires, Empilements, Boucles . . . )
- **Modèle d'énergie :**
  - **Motif** $\rightarrow$ Contribution à l'énergie libre $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  - **Energie libre** $E_w(S)$ **:** Somme $\Delta$ sur les motifs de $S$



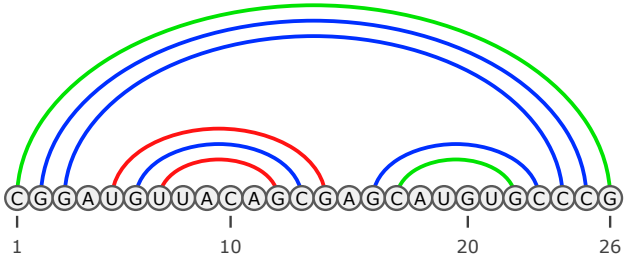



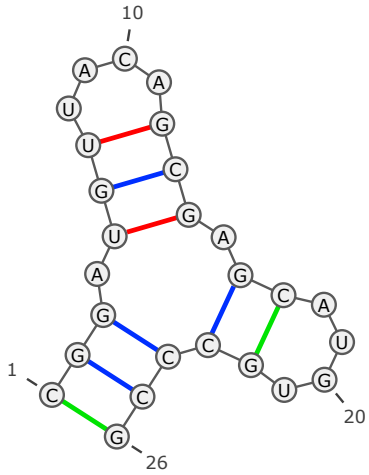

- **Structure d'ARN $S$ :** Couplage**sans croisement** d'un ARN $w$
- **Motifs** séquence/structure (ex. Paires, Empilements, Boucles . . . )
- **Modèle d'énergie :**
  - **Motif** $\rightarrow$ Contribution à l'énergie libre $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  - **Energie libre** $E_w(S)$ **:** Somme $\Delta$ sur les motifs de $S$



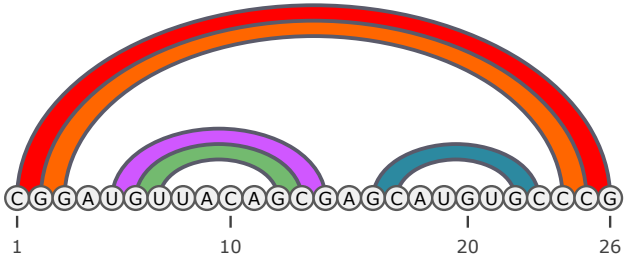

- **Structure d'ARN $S$ :** Couplage**sans croisement** d'un ARN $w$
- **Motifs** séquence/structure (ex. Paires, Empilements, Boucles . . . )
- **Modèle d'énergie :**
  - **Motif** $\rightarrow$ Contribution à l'énergie libre $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  - **Energie libre** $E_w(S)$ **:** Somme $\Delta$ sur les motifs de $S$

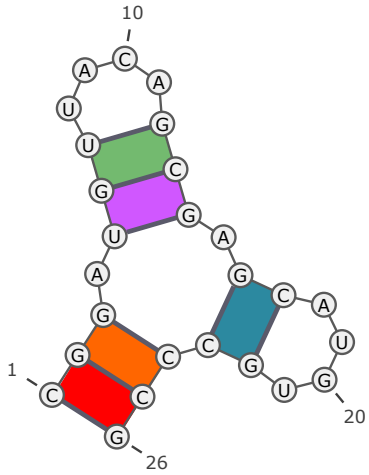



- **Structure d'ARN $S$ :** Couplage**sans croisement** d'un ARN $w$
- **Motifs** séquence/structure (ex. Paires, Empilements, Boucles . . . )
- **Modèle d'énergie :**
  **Motif** $\rightarrow$ Contribution à l'énergie libre $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  **Energie libre** $E_w(S)$ **:** Somme $\Delta$ sur les motifs de $S$

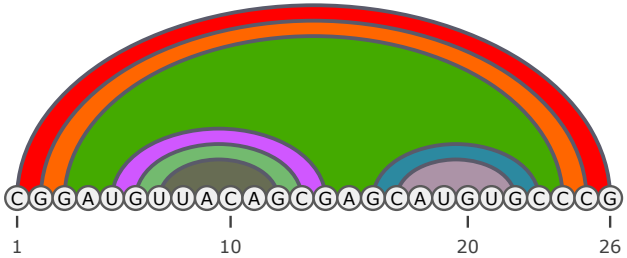
$$E_S = 2 \cdot \Delta \left( \begin{matrix} U \\ G \end{matrix} \right) + 4 \cdot \Delta \left( \begin{matrix} G \\ C \end{matrix} \right) + 2 \cdot \Delta \left( \begin{matrix} C \\ G \end{matrix} \right)$$

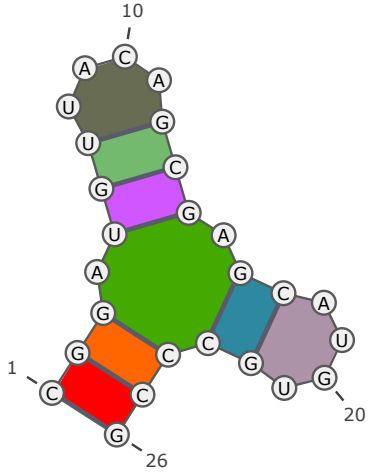



- **Structure d'ARN $S$ :** Couplage**sans croisement** d'un ARN $w$
- **Motifs** séquence/structure (ex. Paires, Empilements, Boucles . . . )
- **Modèle d'énergie :**
  **Motif** $\rightarrow$ Contribution à l'énergie libre $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  **Energie libre** $E_w(S)$ **:** Somme $\Delta$ sur les motifs de $S$

$$E_S = \Delta\left(\begin{matrix}C & G\\ G & C\end{matrix}\right) + \Delta\left(\begin{matrix}G & G\\ C & C\end{matrix}\right) + \Delta\left(\begin{matrix}U & G\\ G & C\end{matrix}\right) + \Delta\left(\begin{matrix}U & G\\ G & C\end{matrix}\right) + \Delta\left(\begin{matrix}U & G\\ G & C\end{matrix}\right)$$



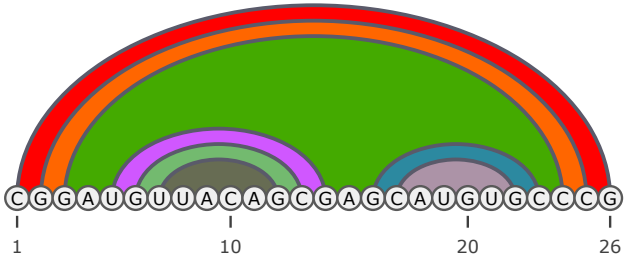



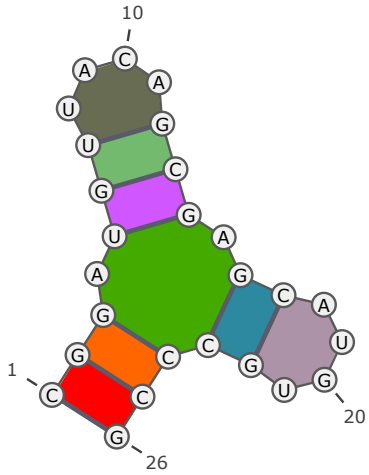

- **Structure d'ARN $S$ :** Couplage**sans croisement** d'un ARN $w$
- **Motifs** séquence/structure (ex. Paires, Empilements, Boucles . . . )
- **Modèle d'énergie :**

  **Motif** $\rightarrow$ Contribution à l'énergie libre $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$

  **Energie libre** $E_w(S)$ **:** Somme $\Delta$ sur les motifs de $S$

$$E_S = \Delta\left(\text{CG/GC}\right) + \Delta\left(\text{GG/CC}\right) + \Delta\left(\text{UG/GC}\right) + \Delta\left(\text{UG/GC}\right) + \Delta\left(\text{UG/GC}\right)$$
$$+ \Delta\left(\text{CACAUG}\right) + \Delta\left(\text{GAGAUCC}\right) + \Delta\left(\text{CAUGUG}\right)$$




- **Structure d'ARN $S$ :** Couplage**sans croisement** d'un ARN $w$
- **Motifs** séquence/structure (ex. Paires, Empilements, Boucles . . . )
- **Modèle d'énergie :**
  **Motif** $\rightarrow$ Contribution à l'énergie libre $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  **Energie libre** $E_w(S)$ **:** Somme $\Delta$ sur les motifs de $S$

**Definition (Problème** REPLIEMENT$(E)$**)**

**Entrée :** Séquence d'ARN $w \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}^*$.
**Sortie :** Couplage ***sans croisement*** $S^\star = \mathrm{argmin}_S E_w(S)$

$\Rightarrow$ Algorithme(s) en $\Theta(n^3)$ par **programmation dynamique**


Récurrence sur l'énergie minimale d'un repliement :

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

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

Schémas similaires pour des modèles d'énergie plus complexes.

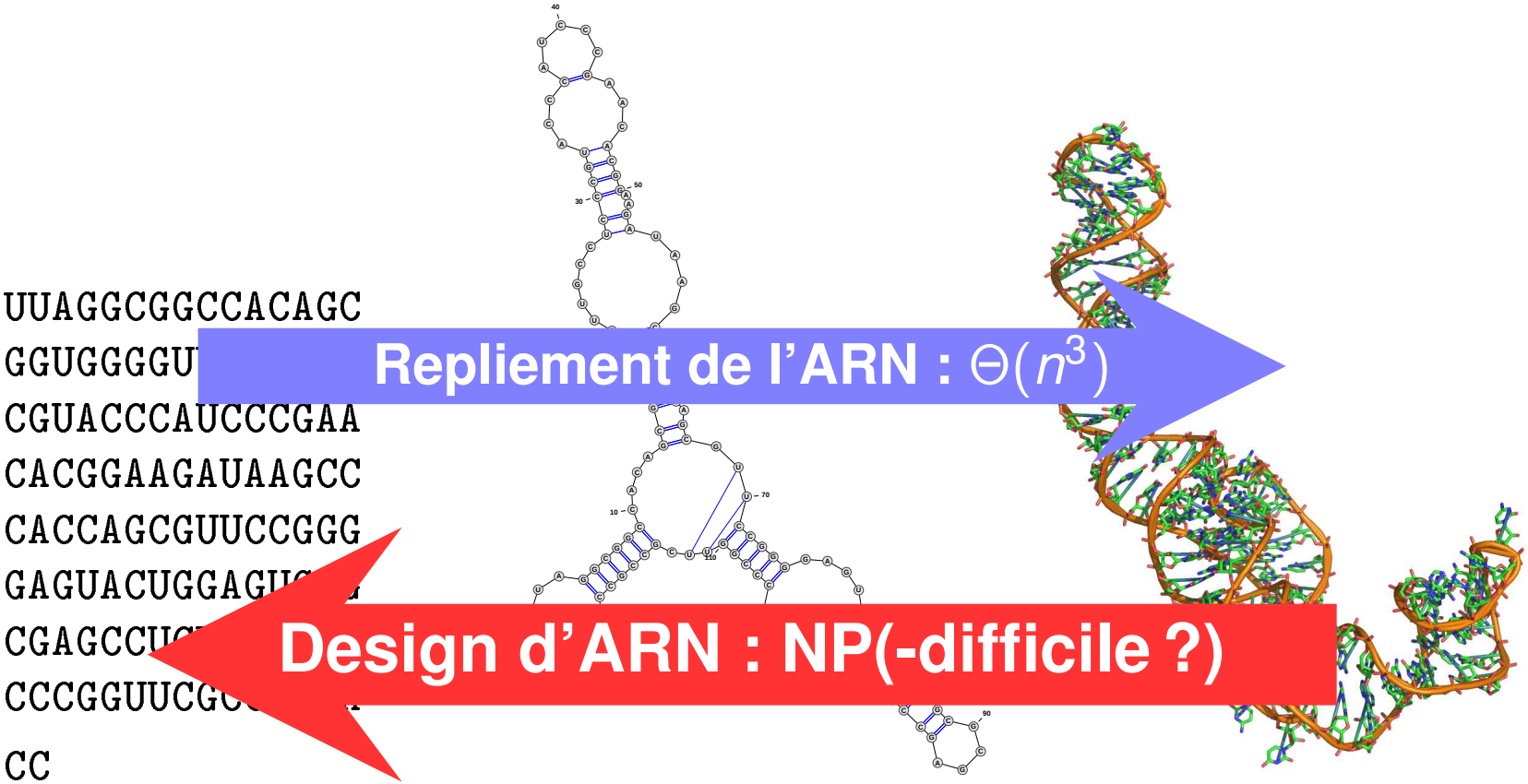
**Definition (Problème REPLIEMENT($E$))**

**Entrée :** Séquence d'ARN $w \in \{A, C, G, U\}^*$.
**Sortie :** Couplage ***sans croisement*** $S^\star = \text{argmin}_S E_w(S)$

$\Rightarrow$ Algorithme(s) en $\Theta(n^3)$ par **programmation dynamique**

# Séquence et structure(s) de l'ARN

**ARN** = Polymère linéaire = Séquence sur $\{A, C, G, U\}^\star$




Structure Primaire  Structure Secondaire  Structure Tertiaire

5s rRNA (PDBID : 1K73 :B)

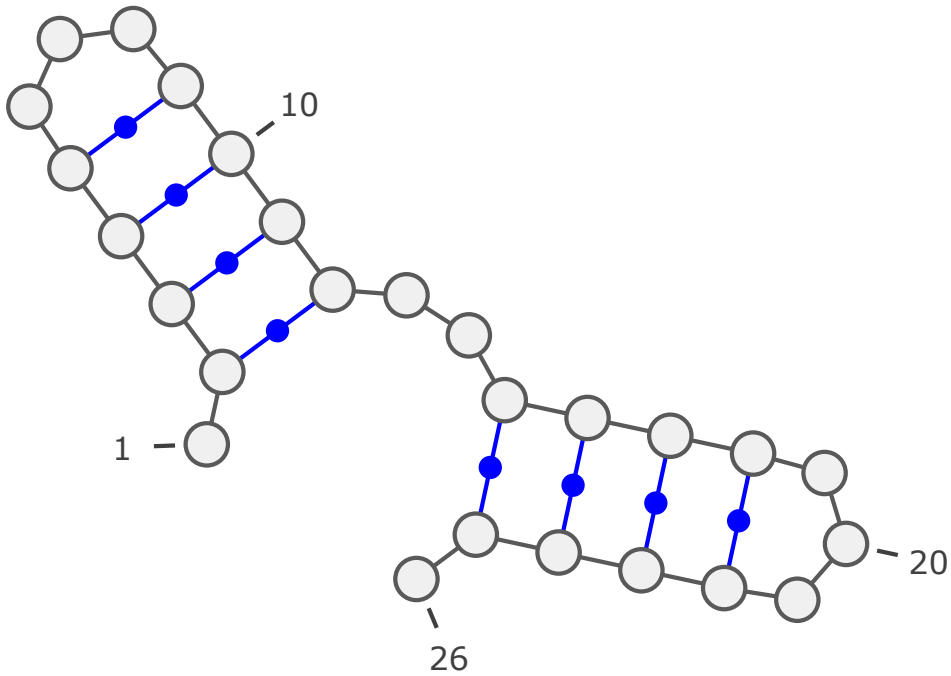
# Repliement inverse d'ARN

**Definition (Problème DESIGN($E$))**

**Entrée :** Couplage **sans croisement** $S \subseteq \mathcal{P}([1, n]^2)$.
**Sortie :** Séquence $w^\star$ telle que $S = \text{argmax}_{S'} E_{w^\star}(S')$

Difficile de **diviser pour régner** ...

**Exemple :**



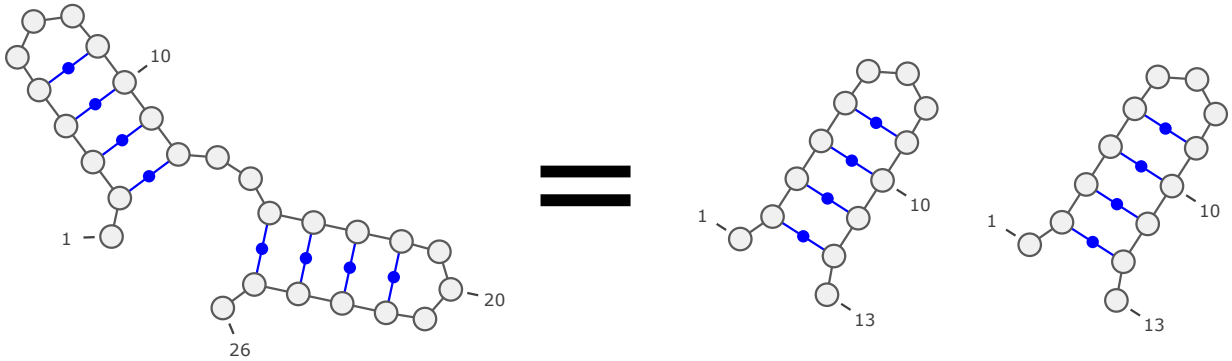
# Repliement inverse d'ARN

**Definition (Problème DESIGN($E$))**

**Entrée :** Couplage **sans croisement** $S \subseteq \mathcal{P}([1, n]^2)$.
**Sortie :** Séquence $w^\star$ telle que $S = \text{argmax}_{S'} E_{w^\star}(S')$

Difficile de **diviser pour régner** ...

**Exemple :**



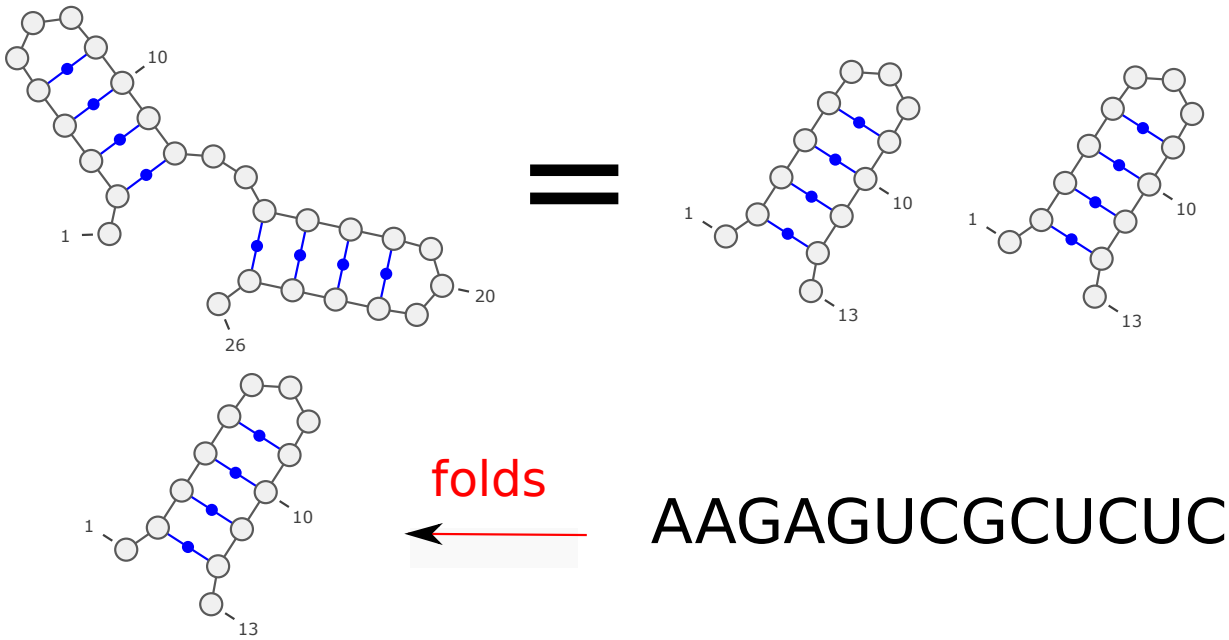
# Repliement inverse d'ARN

**Definition (Problème DESIGN($E$))**

**Entrée :** Couplage **sans croisement** $S \subseteq \mathcal{P}([1, n]^2)$.
**Sortie :** Séquence $w^\star$ telle que $S = \text{argmax}_{S'} E_{w^\star}(S')$

Difficile de **diviser pour régner** ...

**Exemple :**



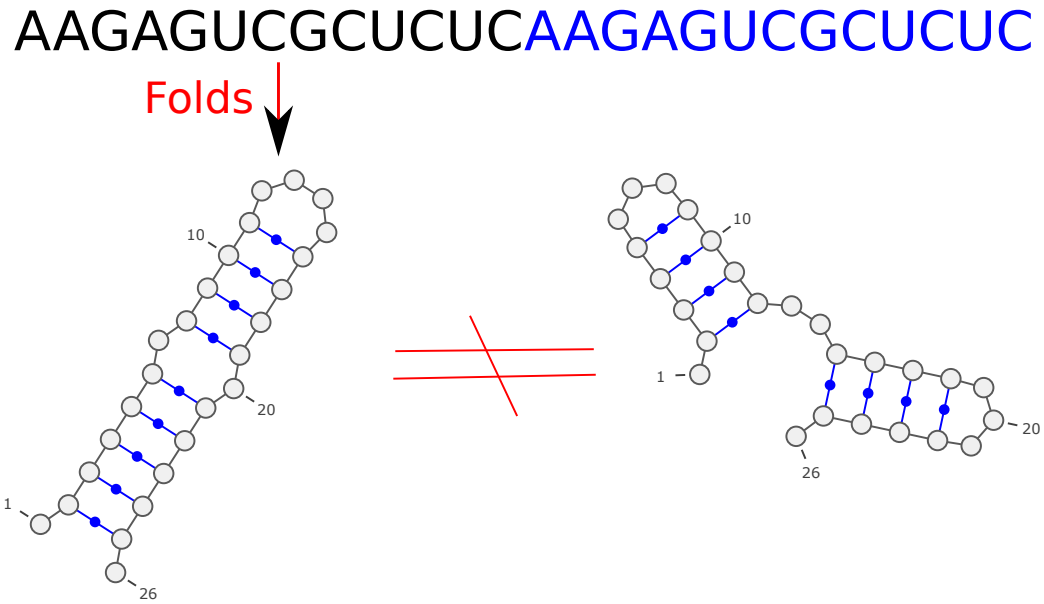
# Repliement inverse d'ARN

**Definition (Problème DESIGN($E$))**

**Entrée :** Couplage **sans croisement** $S \subseteq \mathcal{P}([1, n]^2)$.
**Sortie :** Séquence $w^\star$ telle que $S = \text{argmax}_{S'} E_{w^\star}(S')$

Difficile de **diviser pour régner** ...

**Exemple :**



# Repliement inverse d'ARN

**Definition (Problème DESIGN($E$))**

**Entrée :** Couplage **sans croisement** $S \subseteq \mathcal{P}([1, n]^2)$.
**Sortie :** Séquence $w^\star$ telle que $S = \text{argmax}_{S'} E_{w^\star}(S')$

Difficile de **diviser pour régner** . . . mais **complexité inconnue**

$\Rightarrow$ De nombreuses algorithmes basés sur :

. . . une recherche locale. . .

- RNAInverse
- Info-RNA
- RNA-SSD
- NUPack

. . . des algorithmes génétiques. . .

- RNAFBinv
- FRNAKenstein

. . . ou exponentiels exacts

- RNAIFold
- CO4

**Notre approche :** Génération aléatoire **pondérée**

# Échantillonnage global [Levin *et al*, NAR 12]

- **Distribution de Boltzmann (DB)** basée sur **l'affinité** avec $S$
- **Engendrer aléatoirement** selon **DB**
- **Replier** les séquences engendrées et **comparer** à la cible

Facteur de Boltzmann :

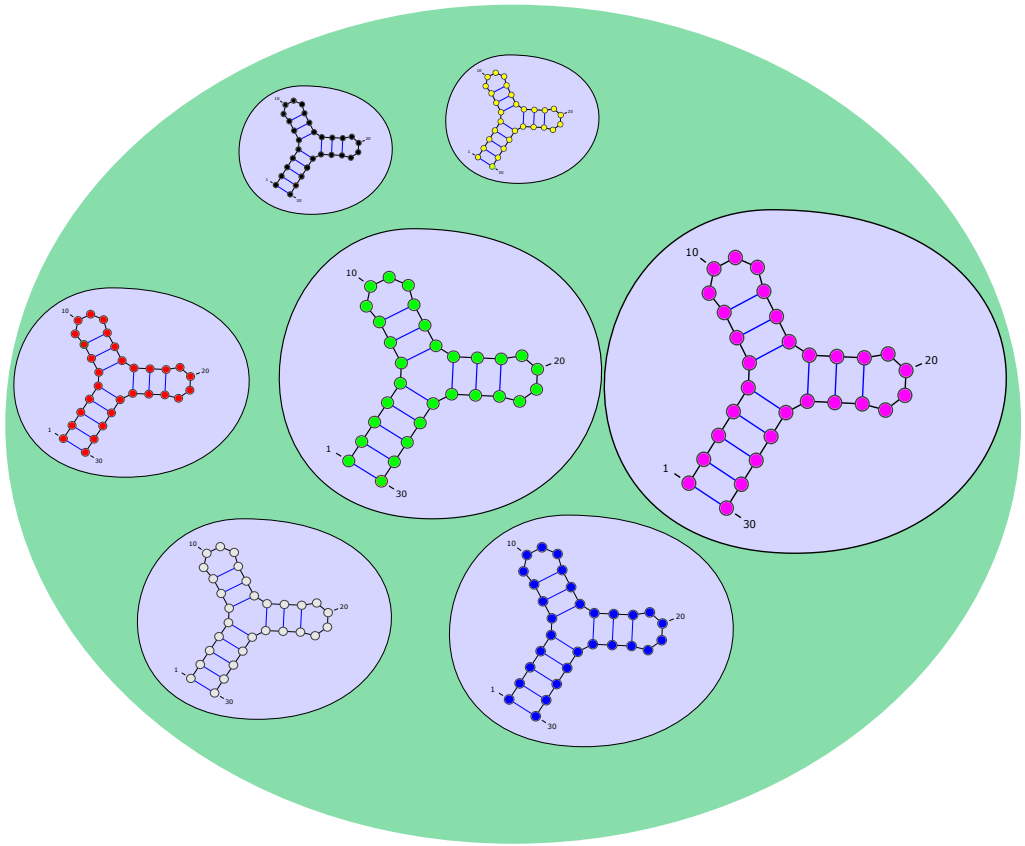
$$\mathcal{B}_w(S) := e^{\frac{-E_w(S)}{RT}}$$

Pseudo-fonction de partition :

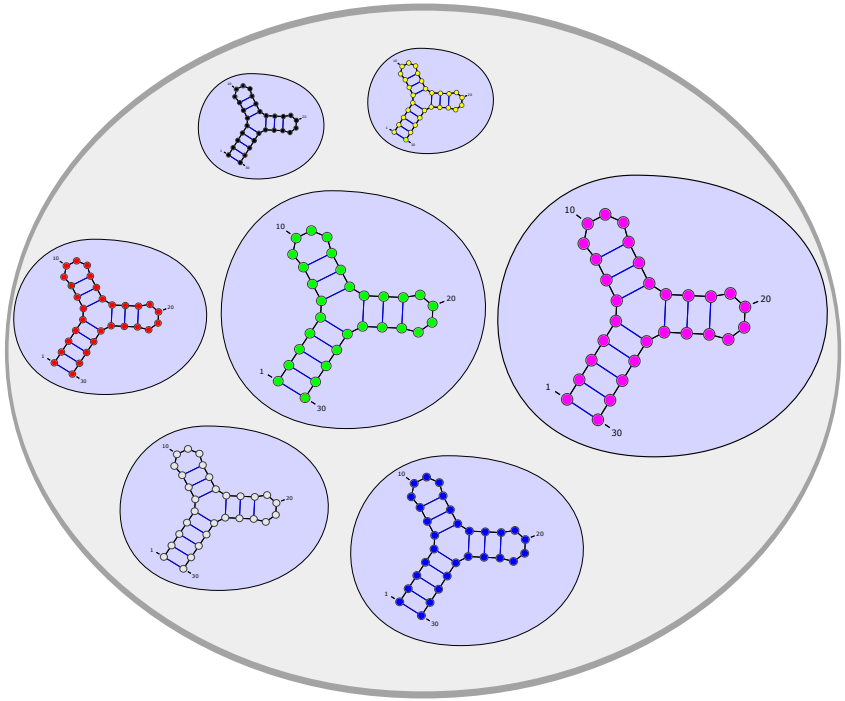
$$\mathcal{Z}(S) = \sum_{w \in \Sigma^*} \mathcal{B}_w(S)$$

Probabilité de Boltzmann :

$$p(s) := \frac{\mathcal{B}_w(S)}{\mathcal{Z}}$$



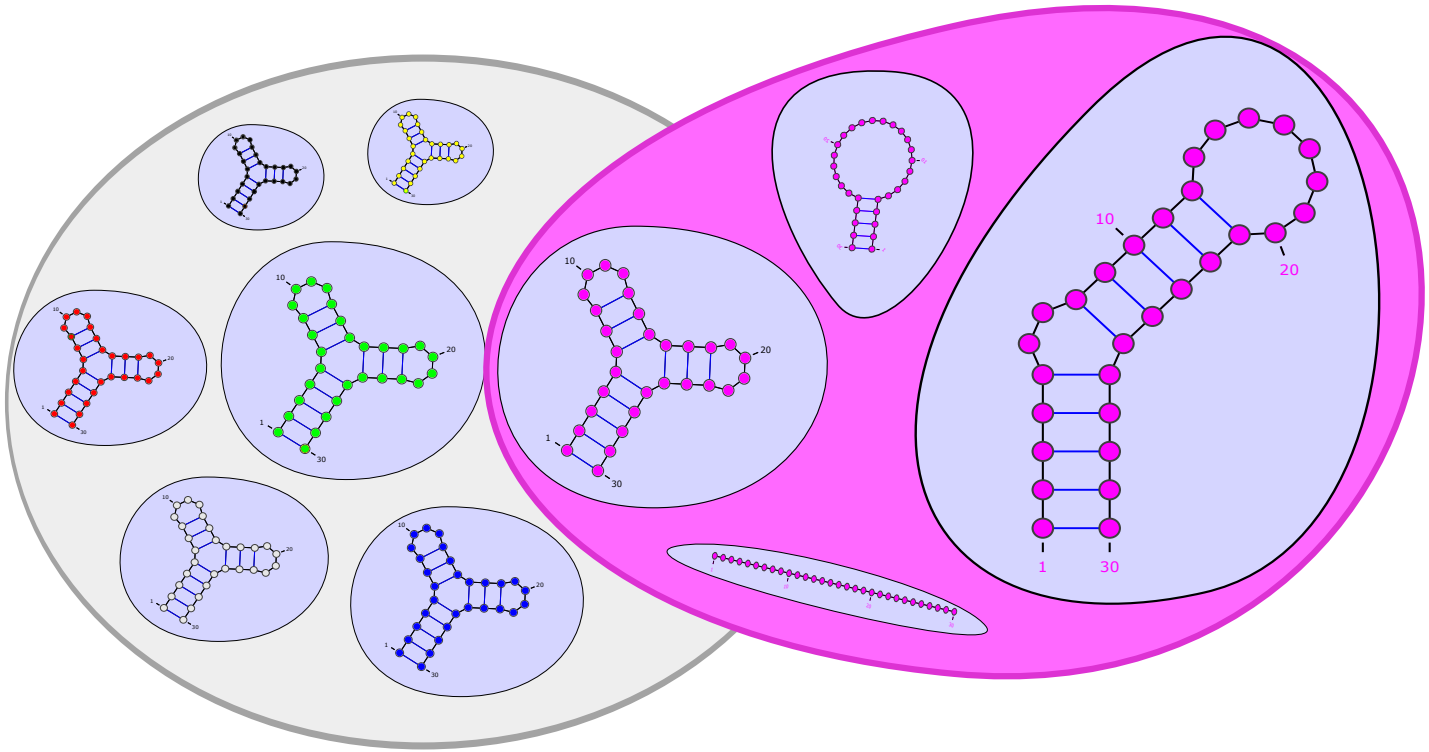
# Limites de l'approche



**Malheureusement**, une forte affinité n'est formellement ni suffisante, ni nécessaire pour être une solution au problème, **mais** . . .

- Forte corrélation **empirique** affinité/succès du design [Levin et al 12]
- **Contrôle** de la composition [Bodini P 10] [Reinharz P Waldispühl 10]

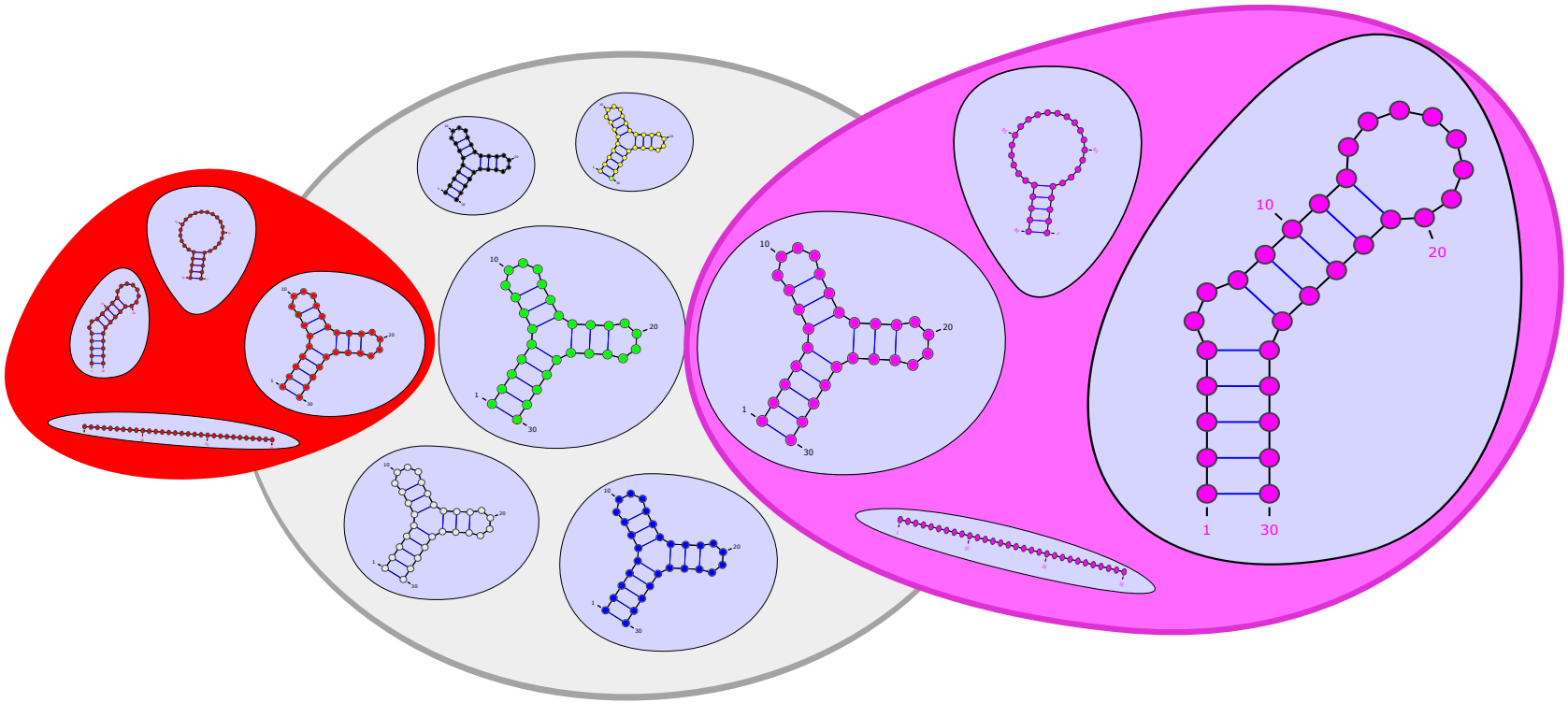
# Limites de l'approche




**Malheureusement**, une forte affinité n'est formellement **ni suffisante,** ni nécessaire pour être une solution au problème, **mais** . . .

- Forte corrélation **empirique** affinité/succès du design [Levin et al 12]
- **Contrôle** de la composition [Bodini P 10] [Reinharz P Waldispühl 10]

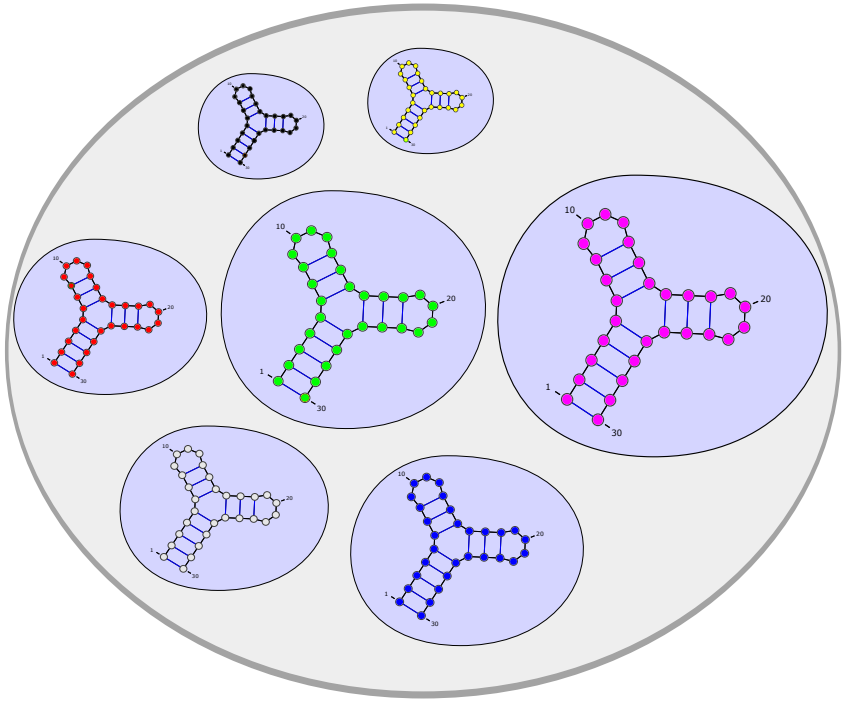
# Limites de l'approche




**Malheureusement**, une forte affinité n'est formellement **ni suffisante, ni nécessaire** pour être une solution au problème, **mais** . . .

- Forte corrélation **empirique** affinité/succès du design [Levin et al 12]
- **Contrôle** de la composition [Bodini P 10] [Reinharz P Waldispühl 10]

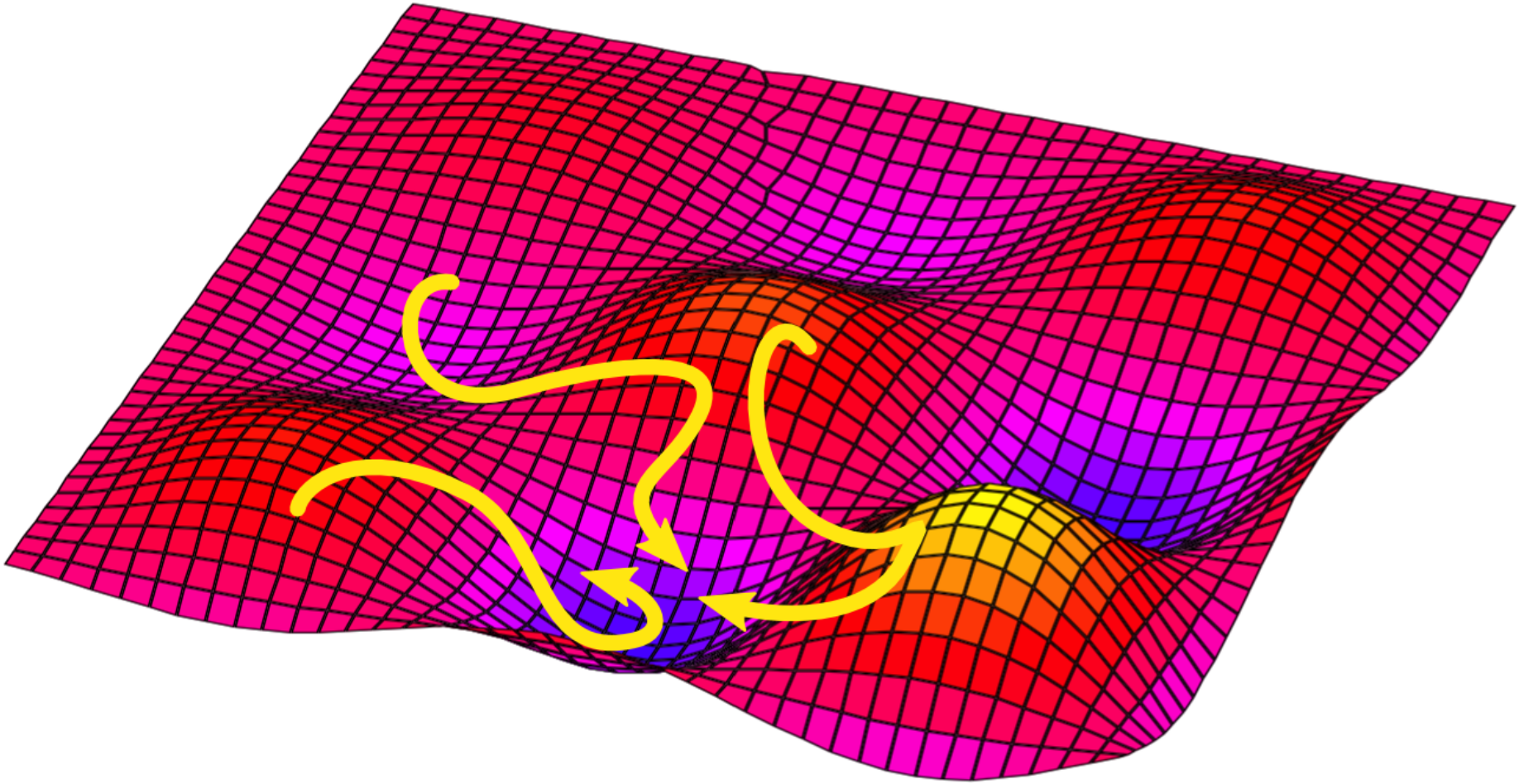
# Limites de l'approche



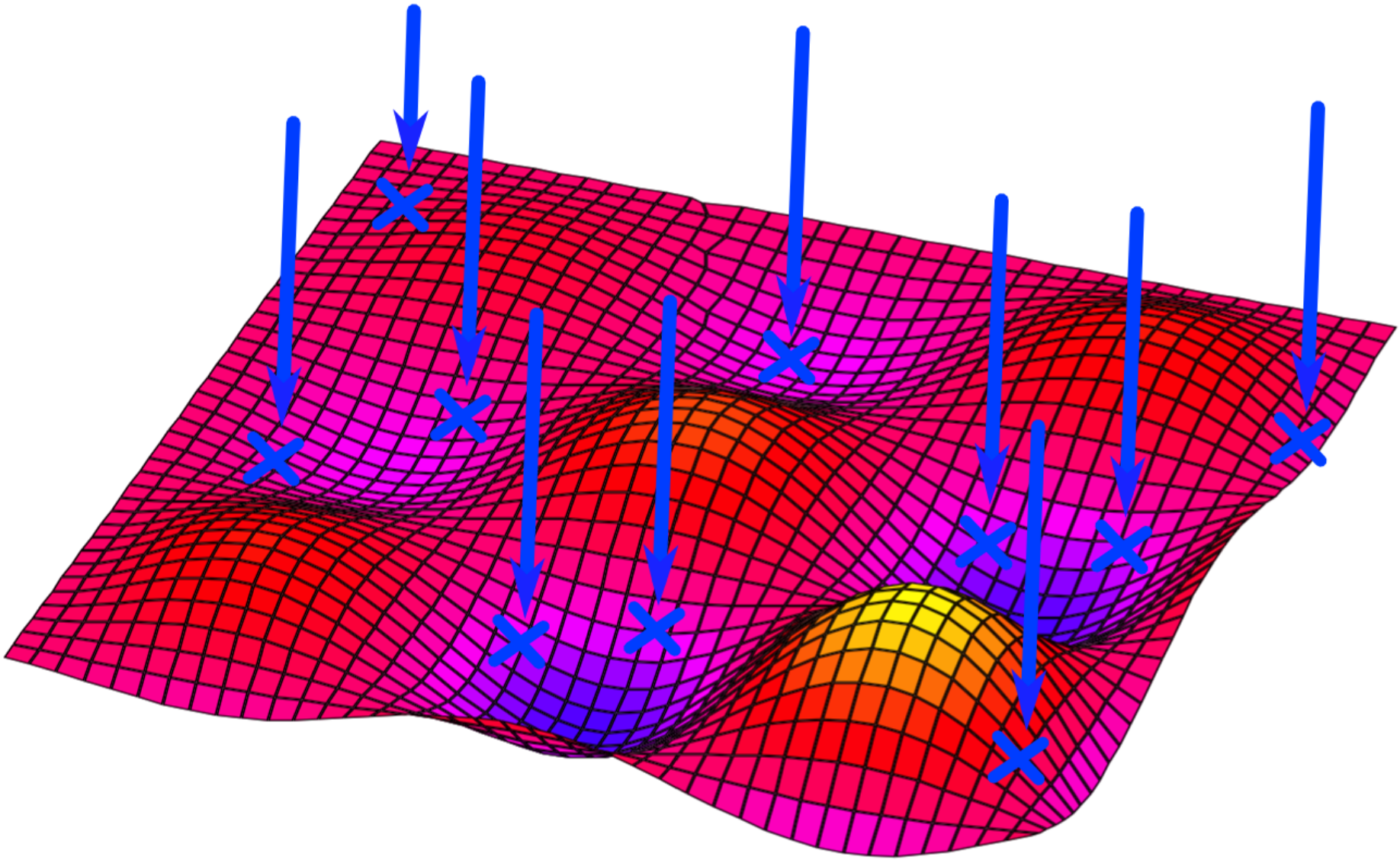
**Malheureusement**, une forte affinité n'est formellement **ni suffisante, ni nécessaire** pour être une solution au problème, **mais** . . .

- Forte corrélation **empirique** affinité/succès du design [Levin et al 12]
- **Contrôle** de la composition [Bodini P 10] [Reinharz P Waldispühl 10]

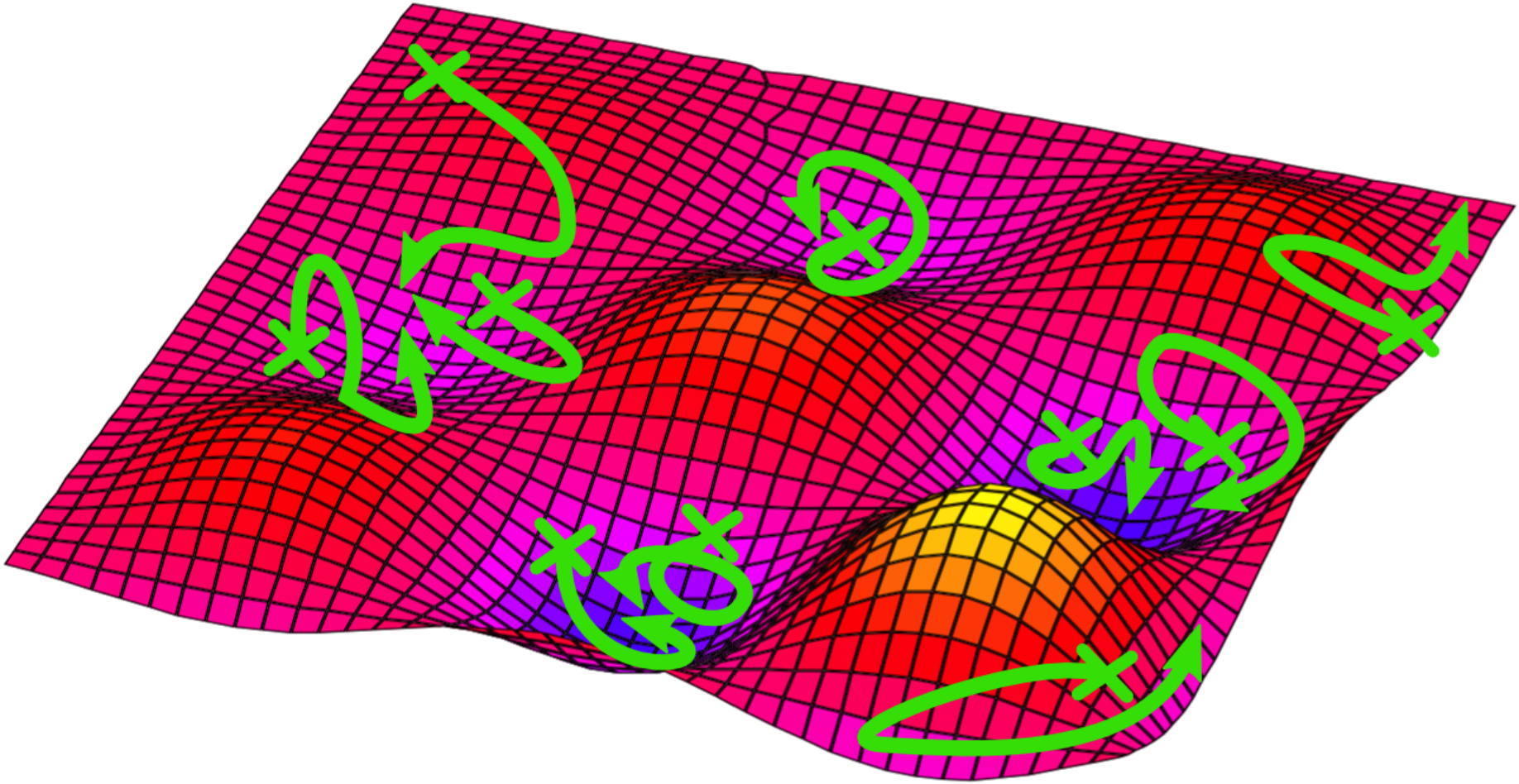
# Local vs Global vs "Glocal"



# Local vs Global vs "Glocal"



# Local vs Global vs "Glocal"



# Construire la grammaire (Modèle d'énergie simple)

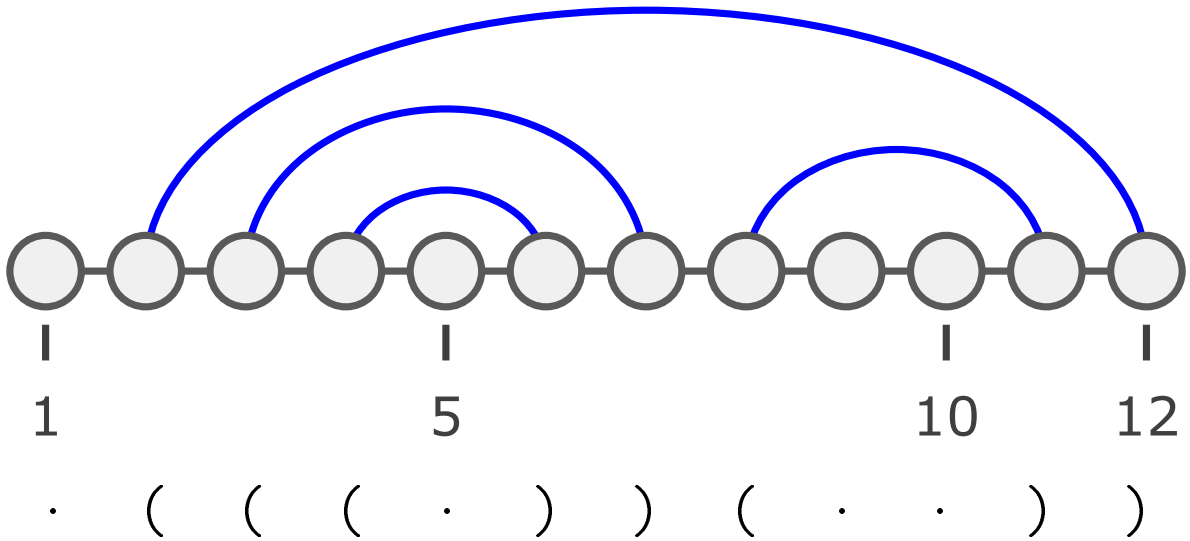
**Entrée :** Structure secondaire $S$

**A Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B Traduire l'ASA** en une grammaire ;

**C Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]



# Construire la grammaire (Modèle d'énergie simple)

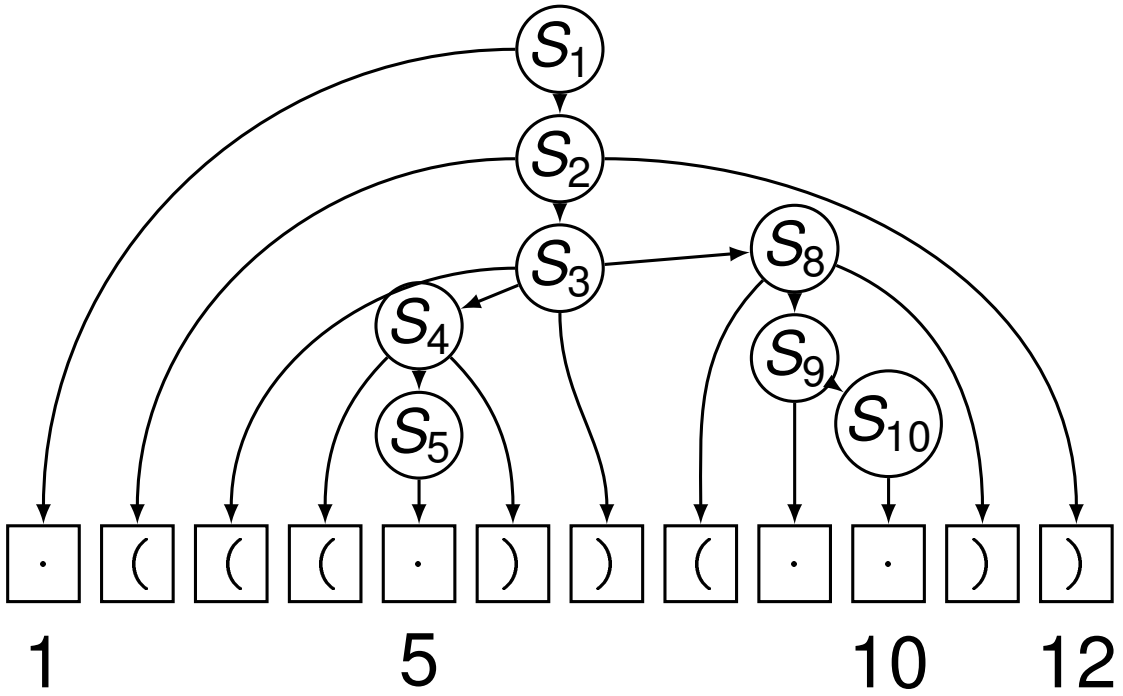
**Entrée :** Structure secondaire $S$

**A Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B Traduire l'ASA** en une grammaire ;

**C Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]



# Construire la grammaire (Modèle d'énergie simple)

**Entrée :** Structure secondaire $S$

**A** **Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B** **Traduire l'ASA** en une grammaire ;

**C** **Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]

$$S_1 \to . S_2 \qquad S_2 \to ( S_3 ) \qquad S_3 \to ( S_4 ) S_8 \qquad S_4 \to ( S_5 )$$
$$S_5 \to . \qquad S_8 \to ( S_9 ) \qquad S_9 \to . S_{10} \qquad S_{10} \to .$$

# Construire la grammaire (Modèle d'énergie simple)

**Entrée :** Structure secondaire $S$

**A Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B Traduire l'ASA** en une grammaire ;

**C Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]

$$
\begin{aligned}
V_1 &\to \mathrm{A}\, V_2 \mid \mathrm{C}\, V_2 \mid \mathrm{G}\, V_2 \mid \mathrm{U}\, V_2 \\
V_2 &\to \mathrm{A}\, V_3\, \mathrm{U} \mid \mathrm{C}\, V_3\, \mathrm{G} \mid \mathrm{G}\, V_3\, \mathrm{C} \mid \mathrm{G}\, V_3\, \mathrm{U} \mid \mathrm{U}\, V_3\, \mathrm{A} \mid \mathrm{U}\, V_3\, \mathrm{G} \\
V_3 &\to \mathrm{A}\, V_4\, \mathrm{U}\, V_8 \mid \mathrm{C}\, V_4\, \mathrm{G}\, V_8 \mid \mathrm{G}\, V_4\, \mathrm{C}\, V_8 \mid \mathrm{G}\, V_4\, \mathrm{U}\, V_8 \mid \mathrm{U}\, V_4\, \mathrm{A}\, V_8 \mid \mathrm{U}\, V_4\, \mathrm{G}\, V_8 \\
V_4 &\to \mathrm{A}\, V_5\, \mathrm{U} \mid \mathrm{C}\, V_5\, \mathrm{G} \mid \mathrm{G}\, V_5\, \mathrm{C} \mid \mathrm{G}\, V_5\, \mathrm{U} \mid \mathrm{U}\, V_5\, \mathrm{A} \mid \mathrm{U}\, V_5\, \mathrm{G} \\
V_5 &\to \mathrm{A} \mid \mathrm{C} \mid \mathrm{G} \mid \mathrm{U} \\
V_8 &\to \mathrm{A}\, V_9\, \mathrm{U} \mid \mathrm{C}\, V_9\, \mathrm{G} \mid \mathrm{G}\, V_9\, \mathrm{C} \mid \mathrm{G}\, V_9\, \mathrm{U} \mid \mathrm{U}\, V_9\, \mathrm{A} \mid \mathrm{U}\, V_9\, \mathrm{G} \\
V_9 &\to \mathrm{A}\, V_{10} \mid \mathrm{C}\, V_{10} \mid \mathrm{G}\, V_{10} \mid \mathrm{U}\, V_{10} \\
V_{10} &\to \mathrm{A} \mid \mathrm{C} \mid \mathrm{G} \mid \mathrm{U}
\end{aligned}
$$

# Construire la grammaire (Modèle d'énergie simple)

**Entrée :** Structure secondaire $S$

**A** **Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B** **Traduire l'ASA** en une grammaire ;

**C** **Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]

$$V_1 \to \mathrm{A}\, V_2 \mid \mathrm{C}\, V_2 \mid \mathrm{G}\, V_2 \mid \mathrm{U}\, V_2$$
$$V_2 \to \mathrm{A}\, V_3\, \mathrm{U} \mid \mathrm{C}\, V_3\, \mathrm{G} \mid \mathrm{G}\, V_3\, \mathrm{C} \mid \mathrm{G}\, V_3\, \mathrm{U} \mid \mathrm{U}\, V_3\, \mathrm{A} \mid \mathrm{U}\, V_3\, \mathrm{G}$$

**Récurrences + Génération**

$$v_2(n) := v_3(n-2) + v_3(n-2) + v_3(n-2) + v_3(n-2) + v_3(n-2) + v_3(n-2)$$
$$v_3(n) := \ldots$$

# Construire la grammaire (Modèle d'énergie simple)

**Entrée :** Structure secondaire $S$

**A** **Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B** **Traduire l'ASA** en une grammaire ;

**C** **Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]

$$V_1 \to \mathrm{A}\, V_2 \mid \mathrm{C}\, V_2 \mid \mathrm{G}\, V_2 \mid \mathrm{U}\, V_2$$
$$V_2 \to \mathrm{A}\, V_3\, \mathrm{U} \mid \mathrm{C}\, V_3\, \mathrm{G} \mid \mathrm{G}\, V_3\, \mathrm{C} \mid \mathrm{G}\, V_3\, \mathrm{U} \mid \mathrm{U}\, V_3\, \mathrm{A} \mid \mathrm{U}\, V_3\, \mathrm{G}$$

**Récurrences + Génération**

$$v_2 := v_3 + v_3 + v_3 + v_3 + v_3 + v_3$$
$$v_3 := \dots$$

# Construire la grammaire (Modèle d'énergie simple)

**Entrée :** Structure secondaire $S$

**A Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B Traduire l'ASA** en une grammaire ;

**C Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]

$$V_1 \to \mathrm{A}\, V_2 \mid \mathrm{C}\, V_2 \mid \mathrm{G}\, V_2 \mid \mathrm{U}\, V_2$$
$$V_2 \to \mathrm{A}\, V_3\, \mathrm{U} \mid \mathrm{C}\, V_3\, \mathrm{G} \mid \mathrm{G}\, V_3\, \mathrm{C} \mid \mathrm{G}\, V_3\, \mathrm{U} \mid \mathrm{U}\, V_3\, \mathrm{A} \mid \mathrm{U}\, V_3\, \mathrm{G}$$

**Récurrences + Génération**

$$v_2 := v_3 \cdot e^{-\Delta_{\mathrm{AU}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{CG}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{GC}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{GU}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{UA}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{UG}}/RT}$$
$$v_3 := \ldots$$

**+ Pondérations**

**+ Composition contrôlée** (Boltzmann multidim. [Bodini P 10])

**Rem. :** Grammaires réalistes plus complexes (effets *Markoviens*) ...

# Construire la grammaire (Modèle d'énergie simple)

**Entrée :** Structure secondaire $S$

**A** **Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B** **Traduire l'ASA** en une grammaire ;

**C** **Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]

$$V_1 \to \mathrm{A}\, V_2 \mid \mathrm{C}\, V_2 \mid \mathrm{G}\, V_2 \mid \mathrm{U}\, V_2$$

$$V_2 \to \mathrm{A}\, V_3 \mathrm{U} \mid \mathrm{C}\, V_3 \mathrm{G} \mid \mathrm{G}\, V_3 \mathrm{C} \mid \mathrm{G}\, V_3 \mathrm{U} \mid \mathrm{U}\, V_3 \mathrm{A} \mid \mathrm{U}\, V_3 \mathrm{G}$$

$$p = \frac{v_3 \cdot e^{-\Delta_{CG}/RT}}{v_2}$$

**Récurrence de génération**

$$v_2 := v_3 \cdot e^{-\Delta_{AU}/RT} + v_3 \cdot e^{-\Delta_{CG}/RT} + v_3 \cdot e^{-\Delta_{GC}/RT} + v_3 \cdot e^{-\Delta_{GU}/RT}$$
$$+ v_3 \cdot e^{-\Delta_{UA}/RT} + v_3 \cdot e^{-\Delta_{UG}/RT}$$

$$v_3 := \ldots$$

**+ Pondérations**

**+ Composition contrôlée** (Boltzmann multidim. [Bodini P 10])

**Rem. :** Grammaires réalistes plus complexes (effets *Markoviens*) ...

# Construire la grammaire (Modèle d'énergie simple)

**Entrée :** Structure secondaire $S$

**A Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B Traduire l'ASA** en une grammaire ;

**C Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**+ Génération Pondérée :** $\Theta^\star(n)$ puis $\Theta(k \cdot n)$ [Denise P Termier 10]

$$V_1 \to \mathrm{A}\, V_2 \mid \mathrm{C}\, V_2 \mid \mathrm{G}\, V_2 \mid \mathrm{U}\, V_2$$

$$V_2 \to \mathrm{A}\, V_3 \mid \mathrm{C}\, V_3 \mid \mathrm{G}\, V_3 \mathrm{U} \mid \mathrm{U}\, V_3 \mathrm{A} \mid \mathrm{U}\, V_3 \mathrm{G}$$

$$p = \frac{v_3 \cdot e^{-\Delta_{\mathrm{GC}}/RT}}{v_2}$$

**Récurrences + Génération**

$$v_2 := v_3 \cdot e^{-\Delta_{\mathrm{AU}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{CG}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{GC}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{GU}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{UA}}/RT} + v_3 \cdot e^{-\Delta_{\mathrm{UG}}/RT}$$

$$v_3 := \ldots$$

**+ Pondérations**

**+ Composition contrôlée** (Boltzmann multidim. [Bodini P 10])

**Rem. :** Grammaires réalistes plus complexes (effets *Markoviens*) ...

# II. Design sous contraintes

## Éviter des motifs

# But du design

**Constraintes supplémentaires :**

- **Interdire** des motifs d'apparaître **où que ce soit** ;
- **Obliger** des motifs à apparaître **au moins une fois** ;
- **Limiter** les bases disponibles à certaines positions.

Peu d'algorithmes tolèrent des motifs interdits/obligés . . .

. . . aucun ne garantit des temps d'exécutions *raisonnables*.

**Raisons typiques :**

- Motifs obligés $\Rightarrow$ Echecs tardifs (Branch and Bound)
- Motifs interdits $\Rightarrow$ Espace d'états déconnecté (Recherche Locale)

# But du design

**Constraintes supplémentaires :**

- **Interdire** des motifs d'apparaître **où que ce soit** ;
- **Obliger** des motifs à apparaître **au moins une fois** ;
- **Limiter** les bases disponibles à certaines positions.

Peu d'algorithmes tolèrent des motifs interdits/obligés . . .

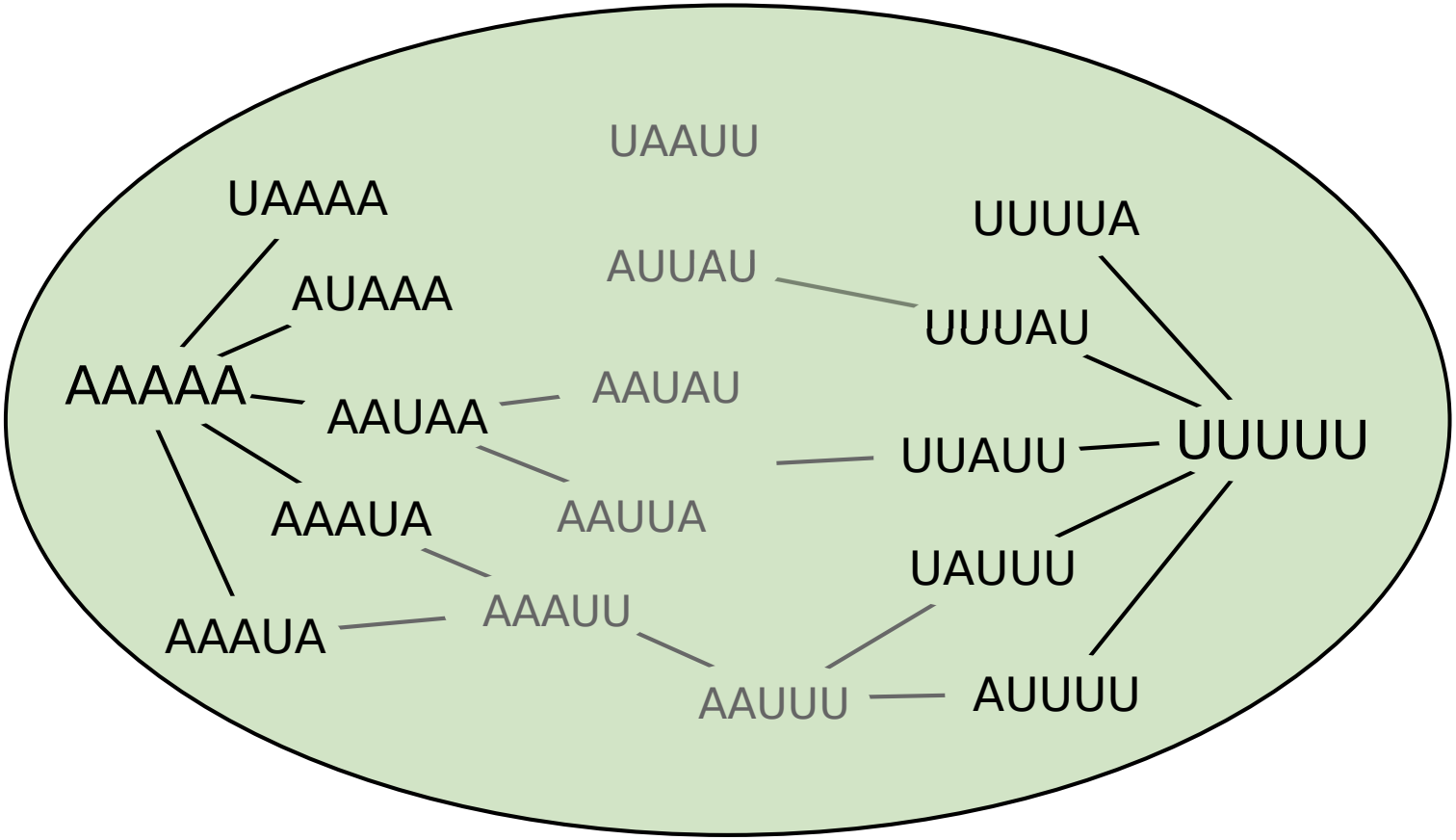
. . . aucun ne garantit des temps d'exécutions *raisonnables*.

**Raisons typiques :**

- Motifs obligés $\Rightarrow$ Echecs tardifs (Branch and Bound)
- Motifs interdits $\Rightarrow$ Espace d'états déconnecté (Recherche Locale)

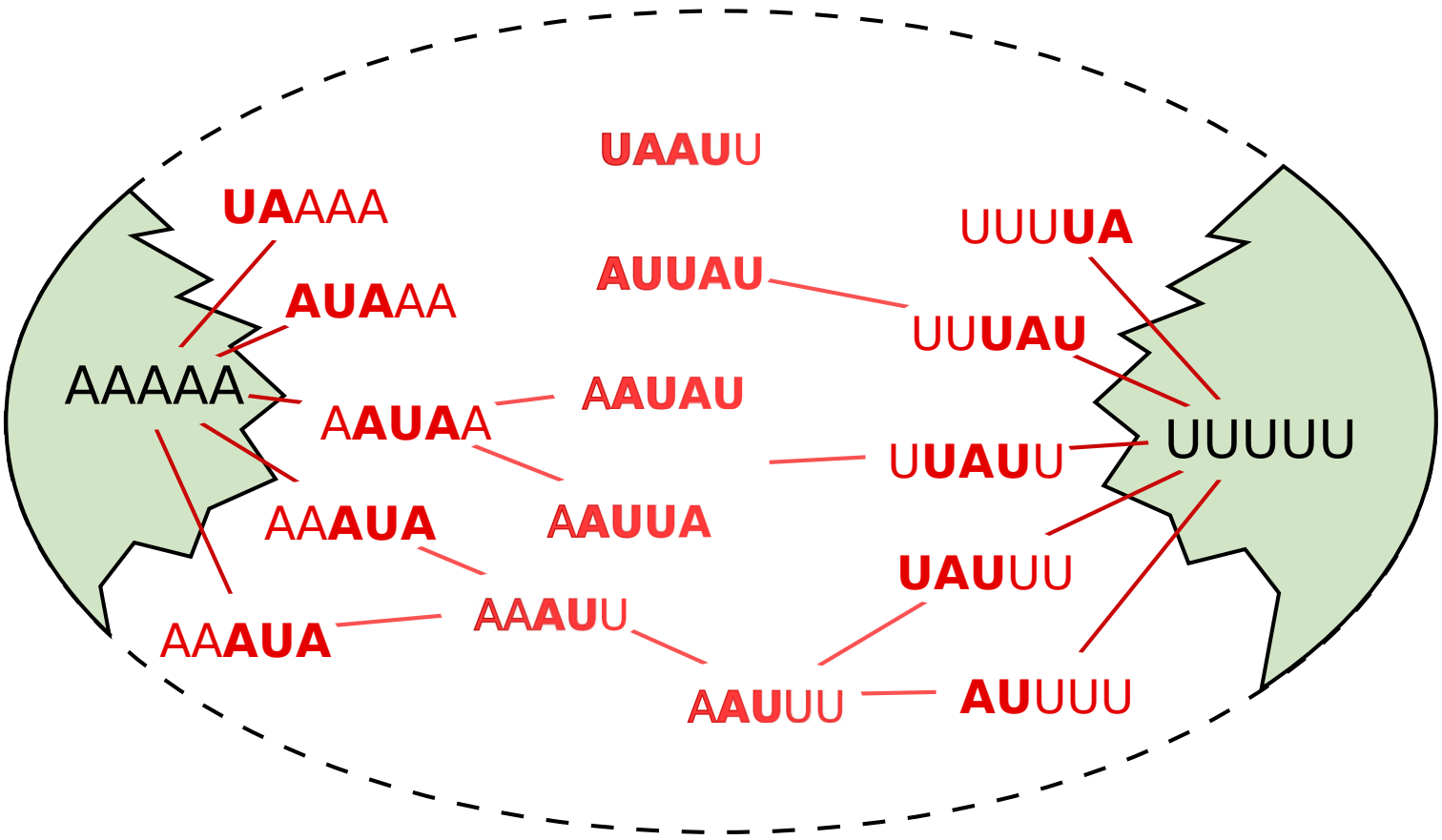
# Problèmes de recherche locale : Un exemple

Vocabulaire simplifié $\{A, U\}$



# Problèmes de recherche locale : Un exemple

Vocabulaire simplifié $\{A, U\}$ + **Motifs interdits** $\mathcal{F} = \{AU, UA\}$




$\Rightarrow \mathcal{F}$ peut **déconnecter** l'espace de recherche ($\forall$ **voisinage** )

**Alt. :** Chaîne de Markov sous-jacente **non ergodique**.

# Idée

**Utiliser la clôture des langages hors-contexte**

Motifs obligés
Motifs interdits

$\rightarrow$ Langage Rationnel $\mathcal{L}_C \in$ Reg

Compatibilité structure
+ Contraintes position
+ **Modèle d'énergie**

$\rightarrow$ Langage Hors-Contextes $\mathcal{L}_S \in$ LHC
**(Pondérés)**

**Théorème Folklore (constructif) :** Reg $\cap$ LHC(**P**) $\subseteq$ LHC(**P**)

**Construire une grammaire hors-contexte $\mathcal{G}$ pour $\mathcal{L}_C \cap \mathcal{L}_S$**
**+ Génération aléatoire**
**$\Rightarrow$ Échantillonnage global sous contrainte**

# Idée

**Utiliser la clôture des langages hors-contexte**

Motifs obligés
Motifs interdits

$\rightarrow$ Langage Rationnel $\mathcal{L}_C \in$ Reg

Compatibilité structure
+ Contraintes position
+ **Modèle d'énergie**

$\rightarrow$ Langage Hors-Contextes $\mathcal{L}_S \in$ LHC
**(Pondérés)**

**Théorème Folklore (constructif) :** Reg $\cap$ LHC(**P**) $\subseteq$ LHC(**P**)

**Construire une grammaire hors-contexte $\mathcal{G}$ pour $\mathcal{L}_C \cap \mathcal{L}_S$**
**+ Génération aléatoire**
**$\Rightarrow$ Échantillonnage global sous contrainte**

# Idée

**Utiliser la clôture des langages hors-contexte**

Motifs obligés
Motifs interdits
$\rightarrow$ Langage Rationnel $\mathcal{L}_C \in$ Reg

Compatibilité structure
+ Contraintes position
+ **Modèle d'énergie**
$\rightarrow$ Langage Hors-Contextes $\mathcal{L}_S \in$ LHC
**(Pondérés)**

**Théorème Folklore (constructif) :** Reg $\cap$ LHC(**P**) $\subseteq$ LHC(**P**)

**Construire une grammaire hors-contexte $\mathcal{G}$ pour $\mathcal{L}_C \cap \mathcal{L}_S$**
**+ Génération aléatoire**
**$\Rightarrow$ Échantillonnage global sous contrainte**

# Idée

**Utiliser la clôture des langages hors-contexte**

Motifs obligés
Motifs interdits
$\rightarrow$ Langage Rationnel $\mathcal{L}_C \in$ Reg

Compatibilité structure
+ Contraintes position
+ **Modèle d'énergie**
$\rightarrow$ Langage Hors-Contextes $\mathcal{L}_S \in$ LHC
**(Pondérés)**

**Théorème Folklore (constructif) :** Reg $\cap$ LHC(**P**) $\subseteq$ LHC(**P**)

**Construire une grammaire hors-contexte $\mathcal{G}$ pour $\mathcal{L}_C \cap \mathcal{L}_S$**
**+ Génération aléatoire**
$\Rightarrow$ **Échantillonnage global sous contrainte**

# Contraintes positionnelles

**Entrée :** Structure secondaire $S$

**A** **Construire l'arbre de syntaxe abstraite (ASA)** de $S$ ;

**B** **Traduire l'ASA** en une grammaire ;

**C** **Étendre** la grammaire pour engendrer les lettres **(+ Poids)**.

**D** **Limiter** les lettres autorisées à certaines positions

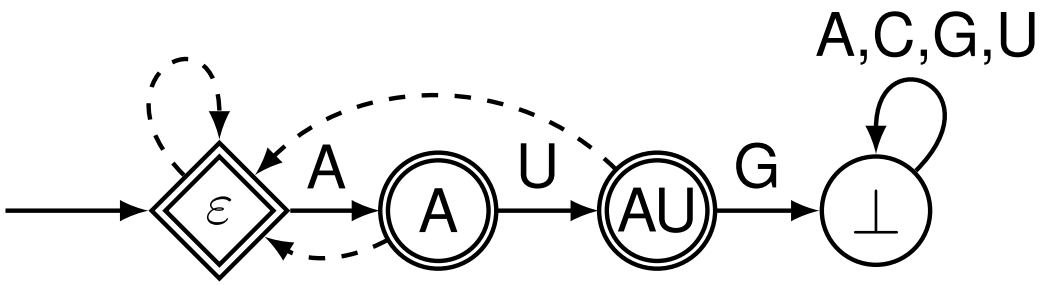
$$
\begin{aligned}
V_1 &\to \mathrm{A}\,V_2 \mid \mathrm{C}\,V_2 \mid \mathrm{G}\,V_2 \mid \mathrm{U}\,V_2\\
V_2 &\to \mathrm{A}\,V_3\,\mathrm{U} \mid \cancel{\mathrm{C}\,V_3\,\mathrm{G}} \mid \cancel{\mathrm{G}\,V_3\,\mathrm{C}} \mid \cancel{\mathrm{G}\,V_3\,\mathrm{U}} \mid \mathrm{U}\,V_3\,\mathrm{A} \mid \cancel{\mathrm{U}\,V_3\,\mathrm{G}}\\
V_3 &\to \mathrm{A}\,V_4\,\mathrm{U}\,V_8 \mid \cancel{\mathrm{C}\,V_4\,\mathrm{G}\,V_8} \mid \cancel{\mathrm{G}\,V_4\,\mathrm{C}\,V_8} \mid \cancel{\mathrm{G}\,V_4\,\mathrm{U}\,V_8} \mid \mathrm{U}\,V_4\,\mathrm{A}\,V_8 \mid \cancel{\mathrm{U}\,V_4\,\mathrm{G}\,V_8}\\
V_4 &\to \mathrm{A}\,V_5\,\mathrm{U} \mid \cancel{\mathrm{C}\,V_5\,\mathrm{G}} \mid \cancel{\mathrm{G}\,V_5\,\mathrm{C}} \mid \cancel{\mathrm{G}\,V_5\,\mathrm{U}} \mid \mathrm{U}\,V_5\,\mathrm{A} \mid \cancel{\mathrm{U}\,V_5\,\mathrm{G}}\\
V_5 &\to \mathrm{A} \mid \mathrm{C} \mid \mathrm{G} \mid \mathrm{U}\\
V_8 &\to \mathrm{A}\,V_9\,\mathrm{U} \mid \cancel{\mathrm{C}\,V_9\,\mathrm{G}} \mid \cancel{\mathrm{G}\,V_9\,\mathrm{C}} \mid \cancel{\mathrm{G}\,V_9\,\mathrm{U}} \mid \mathrm{U}\,V_9\,\mathrm{A} \mid \cancel{\mathrm{U}\,V_9\,\mathrm{G}}\\
V_9 &\to \mathrm{A}\,V_{10} \mid \mathrm{C}\,V_{10} \mid \mathrm{G}\,V_{10} \mid \mathrm{U}\,V_{10}\\
V_{10} &\to \mathrm{A} \mid \mathrm{C} \mid \mathrm{G} \mid \mathrm{U}
\end{aligned}
$$

# Construire un automate fini déterministe

Motifs interdits $\mathcal{F} = \{f_1, f_2 \ldots\}$ Motifs obligés $\mathcal{M} = \{m_1, m_2 \ldots\}$

- Éviter un **motif unique** $\Rightarrow$ Automate linéaire
- Eviter **plusieurs motifs** $\Rightarrow$ Construction **Aho-Corasick**
- Obliger un **motif** $\Rightarrow$ Automate linéaire (Complémenté)
- Obliger **plusieurs motifs en disjonction** $\Rightarrow$ Aho-Corasick

**Exemple :** Éviter AUG ($\mathcal{L}_{\overline{\text{AUG}}}$)

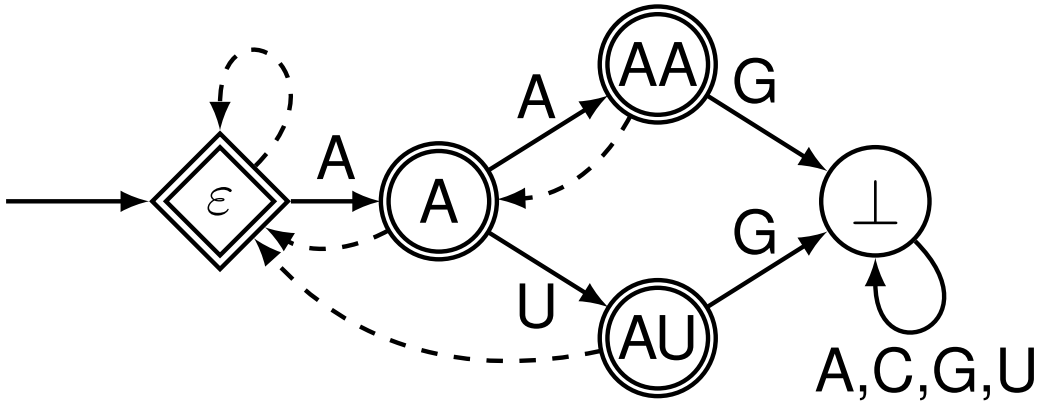


# Construire un automate fini déterministe

Motifs interdits $\mathcal{F} = \{f_1, f_2 \ldots\}$ Motifs obligés $\mathcal{M} = \{m_1, m_2 \ldots\}$

- Éviter un **motif unique** $\Rightarrow$ Automate linéaire
- Eviter **plusieurs motifs** $\Rightarrow$ Construction **Aho-Corasick**
- Obliger un **motif** $\Rightarrow$ Automate linéaire (Complémenté)
- Obliger **plusieurs motifs en disjonction** $\Rightarrow$ Aho-Corasick

**Exemple :** Éviter AUG **+** AAG ($\mathcal{L}_{\overline{\text{AUG}}} \cap \mathcal{L}_{\overline{\text{AAG}}}$)

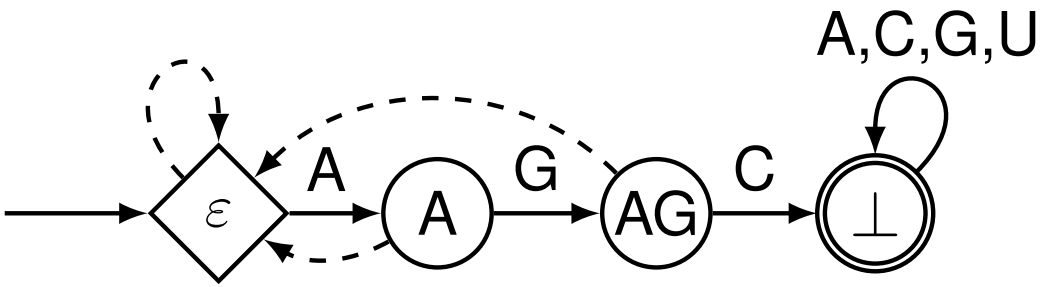


# Construire un automate fini déterministe

Motifs interdits $\mathcal{F} = \{f_1, f_2 \ldots\}$ Motifs obligés $\mathcal{M} = \{m_1, m_2 \ldots\}$

- Éviter un **motif unique** $\Rightarrow$ Automate linéaire
- Eviter **plusieurs motifs** $\Rightarrow$ Construction **Aho-Corasick**
- Obliger un **motif** $\Rightarrow$ Automate linéaire (Complémenté)
- Obliger **plusieurs motifs en disjonction** $\Rightarrow$ Aho-Corasick

**Exemple :** Obliger AGC ($\mathcal{L}_{\text{AGC}}$)

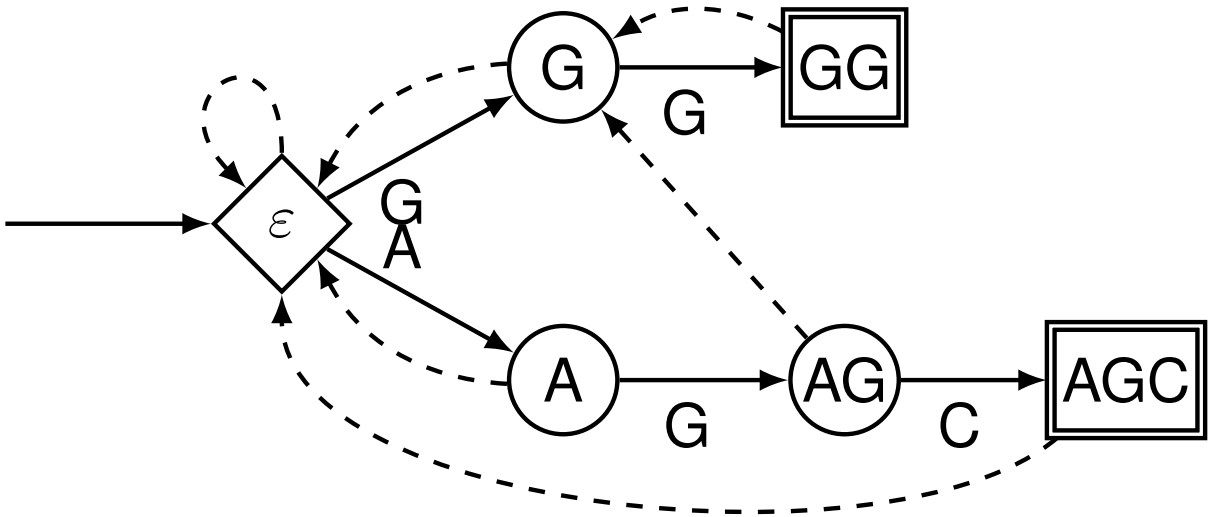


# Construire un automate fini déterministe

Motifs interdits $\mathcal{F} = \{f_1, f_2 \ldots\}$ Motifs obligés $\mathcal{M} = \{m_1, m_2 \ldots\}$

- Éviter un **motif unique** $\Rightarrow$ Automate linéaire
- Eviter **plusieurs motifs** $\Rightarrow$ Construction **Aho-Corasick**
- Obliger un **motif** $\Rightarrow$ Automate linéaire (Complémenté)
- Obliger **plusieurs motifs en disjonction** $\Rightarrow$ Aho-Corasick

**Exemple :** Obliger AGC **ou** GG ($\mathcal{L}_{\mathrm{AGC}} \bigcup \mathcal{L}_{\mathrm{GG}}$)



# Construire un automate fini déterministe

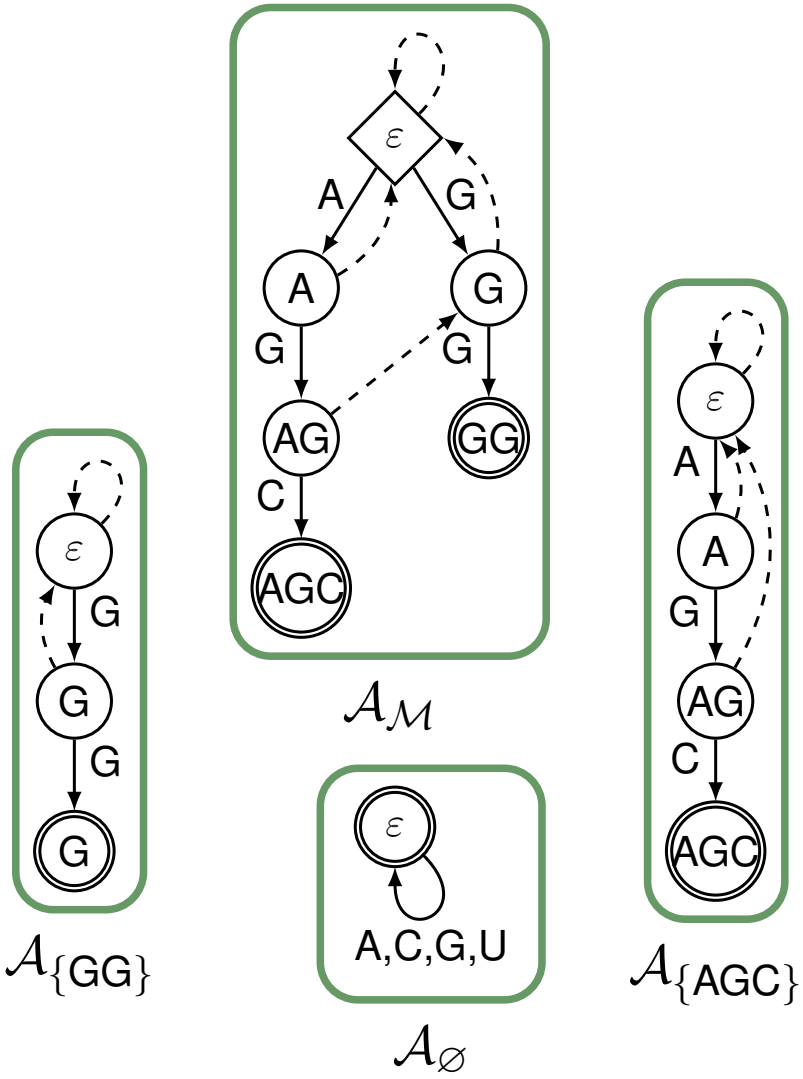
Obliger l'apparition des obligés
$\rightarrow$ **retenir ceux à engendrer** :

- Créer l'automate disjonctif $\mathcal{A}_{\mathcal{M}'}$ pour tout $\mathcal{M}' \subseteq \mathcal{M}$
- **Rediriger** états acceptants
- État acceptant = **Tout motif obligé** est généré ($\varepsilon$ in $\mathcal{A}_\varnothing$)
- Ajouter motifs interdits aux sous-automates

**#États :**

$$O\left(2^{|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)\right)$$

**Example :** $\mathcal{M} = \{\mathrm{AGC}, \mathrm{GG}\}$



# Construire un automate fini déterministe

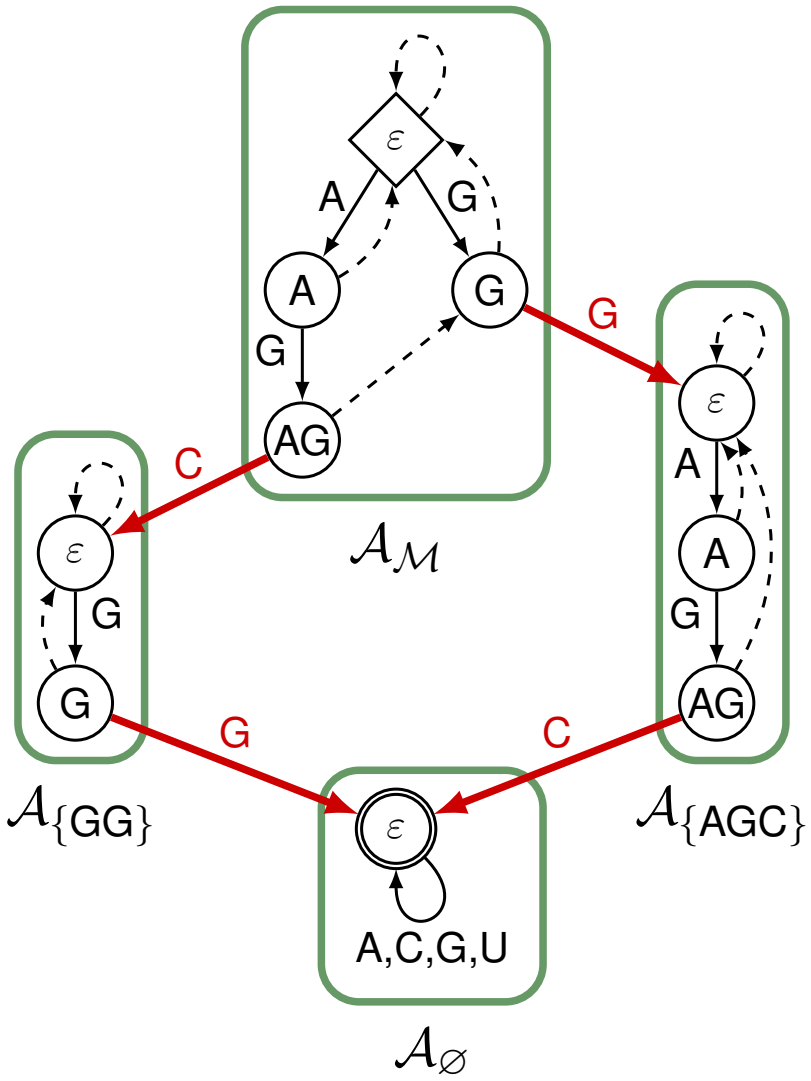
Obliger l'apparition des obligés
$\rightarrow$ **retenir ceux à engendrer** :

- Créer l'automate disjonctif $\mathcal{A}_{\mathcal{M}'}$ pour tout $\mathcal{M}' \subseteq \mathcal{M}$
- **Rediriger** états acceptants
- État acceptant = **Tout motif obligé** est généré ($\varepsilon$ in $\mathcal{A}_\varnothing$)
- Ajouter motifs interdits aux sous-automates

**#États :**

$$O\left(2^{|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)\right)$$

**Example :** $\mathcal{M} = \{\mathrm{AGC}, \mathrm{GG}\}$



# Construire un automate fini déterministe

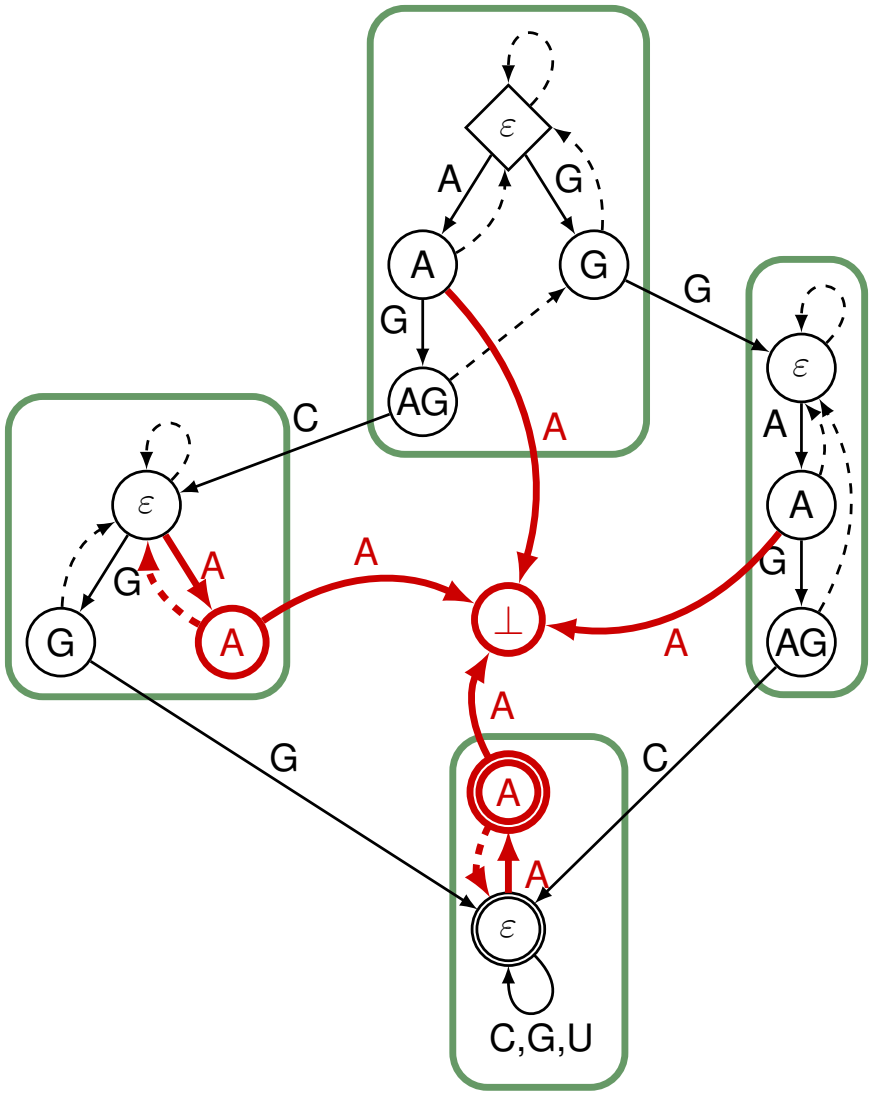
Obliger l'apparition des obligés
$\rightarrow$ **retenir ceux à engendrer** :

- Créer l'automate disjonctif $\mathcal{A}_{\mathcal{M}'}$ pour tout $\mathcal{M}' \subseteq \mathcal{M}$
- **Rediriger** états acceptants
- État acceptant = **Tout motif obligé** est généré ($\varepsilon$ in $\mathcal{A}_\varnothing$)
- Ajouter motifs interdits aux sous-automates

**#États :**

$$O\left(2^{|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)\right)$$

**Example :** $\mathcal{M} = \{\mathsf{AGC}, \mathsf{GG}\}; \mathcal{F} = \{\mathsf{AA}\}$



# Combiner une grammaire $\mathcal{G}$ et un automate $\mathcal{A}$

**Idée (Folklore) :**

- **Simuler l'exécution** de $\mathcal{A}$ lors des dérivations de $\mathcal{G}$ ;
- Tout non-terminal $V_{q\to q'}$ ***s'engage*** à ***relier*** deux états de $\mathcal{A}$ ;
- Les règles terminales **vérifient** le respect de l'engagement ;
- Certaines règles deviennent **improductives** $\to$ nettoyage.

$$V \to t\ V' \quad \Longrightarrow \quad V_{q\to q'} \to t\ V'_{\delta(q,t)\to q'}$$

$$V \to t\ V'\ t' \quad \Longrightarrow \quad V_{q\to q'} \to t\ V'_{\delta(q,t)\to q''}\ t' \text{ t.q. } \delta(q'',t') = q'$$

$$V \to t\ V'\ t'\ V'' \quad \Longrightarrow \quad V_{q\to q'} \to t\ V'_{\delta(q,t)\to q''}\ t'\ V''_{\delta(q'',t')\to q'}$$

$$V \to t \quad \Longrightarrow \quad \begin{cases} V_{q\to q'} \to t & \text{if } \delta(q,t) = q' \\ \varnothing & \text{otherwise} \end{cases}$$

$\delta(\cdot,\cdot)$ **:** Fonction de transition de $\mathcal{A}$

# Complexité

- **Automate :** Temps $\approx$ #États $|Q| \in \mathcal{O}\left(2^{|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)\right)$
- **Grammaire initiale :** #Règles $\approx$ #Non Terminaux $\in \Theta(n)$
- **Grammaire contrainte :**
  #Non Terminaux $|N| \in \mathcal{O}(n \cdot |Q|^2)$
  #Règles $|R| \in \mathcal{O}(n \cdot |Q|^3)$
- **Génération aléatoire**
  **Précalcul :** $\Theta^\star(|R|)$ temps/$\Theta(|N|)$ mémoire
  **Génération :** $\Theta(k \cdot n \cdot |Q|)$ pour $k$ séquences

  **Rem. :** Longueur des mots produits déterminée par NT.

L'algorithme entier a **complexité linéaire sur la taille**

mais la **taille totale des motifs** influence le précalcul.

# Conclusion/Perspectives

**Fait :**

- Algorithme en $\Theta(n)$ (+ctes raisonnables) pour le design **positif**
- Outils :
  - Méthode récursive [Flajolet Van Cutsem Zimmermann 94, Denise P Termier 10]
  - Boltzmann multidimensionnel [Bodini P 10]
- Impliqué dans une expérience biologique **contrôlée**
- Aléa pondéré **compétitif/complémentaire** avec recherche locale

# Conclusion/Perspectives

**Fait :**

- Algorithme en $\Theta(n)$ (+ctes raisonnables) pour le design **positif**
- Outils :
  - Méthode récursive [Flajolet Van Cutsem Zimmermann 94, Denise P Termier 10]
  - Boltzmann multidimensionnel [Bodini P 10]
- Impliqué dans une expérience biologique **contrôlée**
- Aléa pondéré **compétitif/complémentaire** avec recherche locale

**À faire/comprendre :**

- Meilleures constructions pour produits automates/grammaires.
  En particulier, non-terminal $\rightarrow$ taille $\Rightarrow$ **gain asymptotique ?**
- Plus petits automates **sans minimiser** ? [Ait Mous Bassino Nicaud 12]
- Génération **sans redondance** [Lorenz P 12]
- Etude de complexité (Obstacles : optim. inverse, dénaturation . . . )

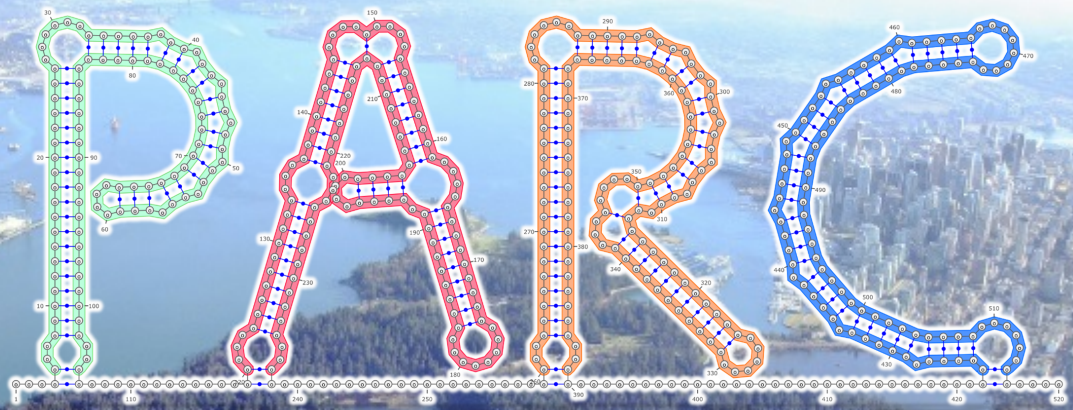
# Remerciements


- **Co-auteurs**
  - Yu Zhou (UCSD)
  - Stéphane Vialette (LIGM)
  - Jérôme Waldispühl
    + Vlad Reinharz (McGill)
  - Yi Zhang (Wuhan, China)
  - Alain Denise (LRI/IGM)

**Merci de votre attention !**


CRG Applied Combinatorics

SFU SIMON FRASER UNIVERSITY MATHEMATICS



# PIMS Analytic RNA Combinatorics

## April 15 - 16 2014 Simon Fraser University, Burnaby

Speakers

Ralf Bundschuh (Ohio State)
Anne Condon (UBC)
Christine Heitsch (Georgia Tech)
Jan Manuch (UBC/SFU)
Markus Nebel (TU Kaiserslautern)
Yann Ponty (CNRS/Polytechnique)

Current RNA bioinformatics strongly relies on discrete abstractions and decomposition schemes for the conformation of nucleic acids. This workshop will bring together scientists from bioinformatics, mathematics and computer science in an attempt to bridge the gap between the enumeration and (asymptotic) analysis of models for RNA structure, and the design of RNA algorithms.

http://tinyurl.com/parc-2014 contact: mmishna@sfu.ca yann.ponty@lix.polytechnique.fr