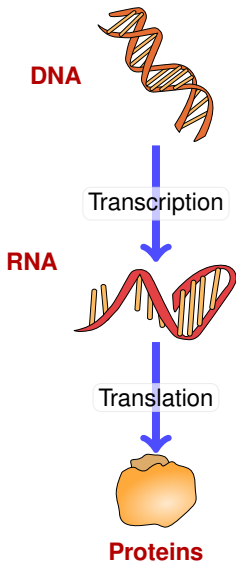


# Algorithmic aspects of negative and positive RNA design

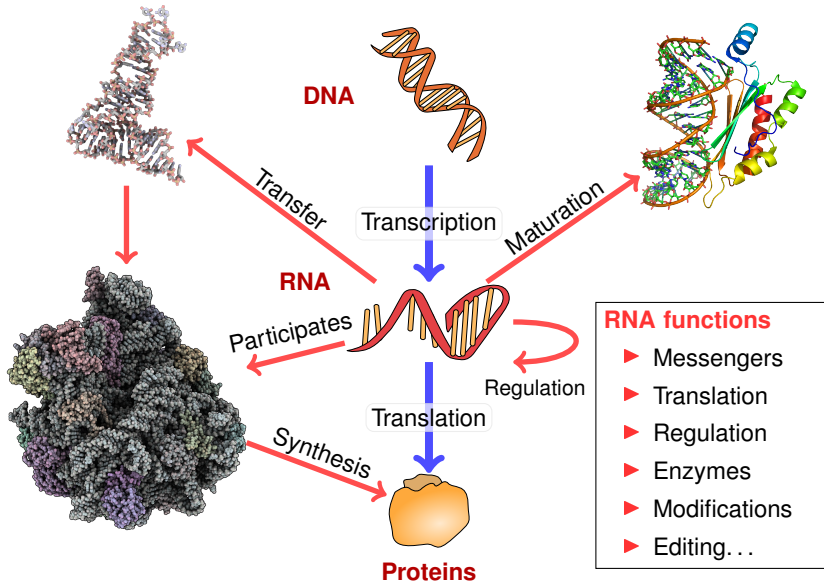
Yann Ponty

LIX, CNRS/Ecole Polytechnique

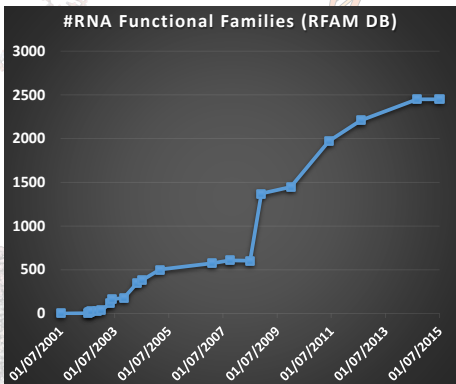
# Fundamental dogma of molecular biology



# Fundamental dogma of molecular biology (v2.0)



# Fundamental dogma of molecular biology (v2.0)



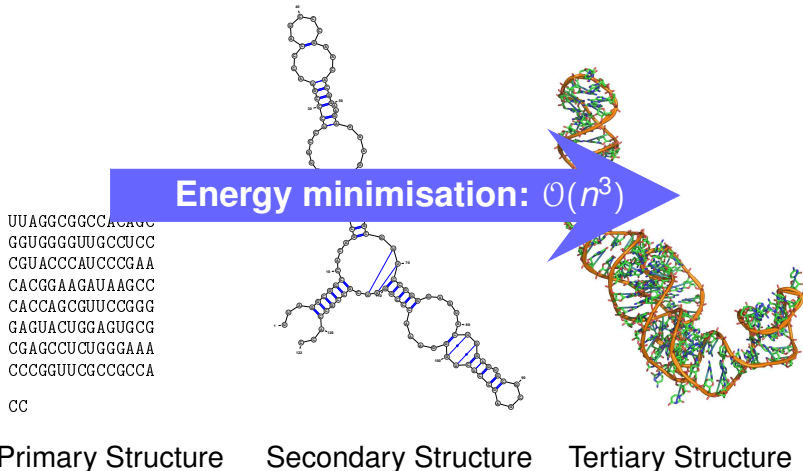
## RNA functions

- ▶ Messengers
- ▶ Translation
- ▶ Regulation
- ▶ Enzymes
- ▶ Modifications
- ▶ Editing...

Proteins

# RNA sequence and structure(s)

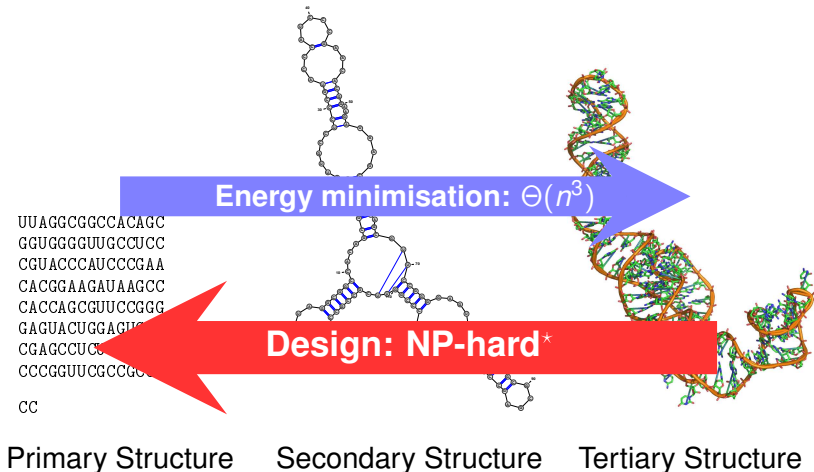
**RNA** = Linear Polymer = Sequence over  $\{A, C, G, U\}^*$



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

# RNA sequence and structure(s)

**RNA** = Linear Polymer = Sequence over  $\{A, C, G, U\}^*$



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

\*Finally! [Bonnet/Rzążewski/Sikora, RECOMB'18]

## Why we design RNAs

- ▶ **To create building blocks for synthetic systems**  
Rationally-designed RNAs increase orthogonality
- ▶ To assess the significance of observed phenomenon  
Random models should include every established characters...  
...including adoption of a single structure
- ▶ To test/push our understanding of how RNA folds  
Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- ▶ To help search for homologous sequences  
Incomplete covariance models hindered by limited training sets  
Design can be used to generalize existing alignments
- ▶ To fuel RNA-based therapeutics  
Sequence-based (siRNA, synthetic genes), but structure matters
- ▶ To perform controlled experiments

## Why we design RNAs

- ▶ To create building blocks for synthetic systems  
Rationally-designed RNAs increase orthogonality
- ▶ To assess the significance of observed phenomenon  
Random models should include every established characters. . .  
. . . including adoption of a single structure
- ▶ To test/push our understanding of how RNA folds  
Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- ▶ To help search for homologous sequences  
Incomplete covariance models hindered by limited training sets  
Design can be used to generalize existing alignments
- ▶ To fuel RNA-based therapeutics  
Sequence-based (siRNA, synthetic genes), but structure matters
- ▶ To perform controlled experiments



## Why we design RNAs

- ▶ To create building blocks for synthetic systems  
Rationally-designed RNAs increase orthogonality
- ▶ To assess the significance of observed phenomenon  
Random models should include every established characters. . .  
. . . including adoption of a single structure
- ▶ To test/push our understanding of how RNA folds  
Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- ▶ To help search for homologous sequences  
Incomplete covariance models hindered by limited training sets  
Design can be used to generalize existing alignments
- ▶ To fuel RNA-based therapeutics  
Sequence-based (siRNA, synthetic genes), but structure matters
- ▶ To perform controlled experiments

## Why we design RNAs

- ▶ To create building blocks for synthetic systems  
Rationally-designed RNAs increase orthogonality
- ▶ To assess the significance of observed phenomenon  
Random models should include every established characters. . .  
. . . including adoption of a single structure
- ▶ To test/push our understanding of how RNA folds  
Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- ▶ To help search for homologous sequences  
Incomplete covariance models hindered by limited training sets  
Design can be used to generalize existing alignments
- ▶ To fuel RNA-based therapeutics  
Sequence-based (siRNA, synthetic genes), but structure matters
- ▶ To perform controlled experiments

## Why we design RNAs

- ▶ To create building blocks for synthetic systems  
Rationally-designed RNAs increase orthogonality
- ▶ To assess the significance of observed phenomenon  
Random models should include every established characters. . .  
. . . including adoption of a single structure
- ▶ To test/push our understanding of how RNA folds  
Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- ▶ To help search for homologous sequences  
Incomplete covariance models hindered by limited training sets  
Design can be used to generalize existing alignments
- ▶ To fuel RNA-based therapeutics  
Sequence-based (siRNA, synthetic genes), but structure matters
- ▶ To perform controlled experiments

## Why we design RNAs

- ▶ To create building blocks for synthetic systems  
Rationally-designed RNAs increase orthogonality
- ▶ To assess the significance of observed phenomenon  
Random models should include every established characters. . .  
. . . including adoption of a single structure
- ▶ To test/push our understanding of how RNA folds  
Misfolding RNAs reveal gaps in our energy models and descriptors for the conformational spaces
- ▶ To help search for homologous sequences  
Incomplete covariance models hindered by limited training sets  
Design can be used to generalize existing alignments
- ▶ To fuel RNA-based therapeutics  
Sequence-based (siRNA, synthetic genes), but structure matters
- ▶ To perform controlled experiments

## The Nobel Prize in Physiology or Medicine 2006

---



Photo: L. Cicero  
**Andrew Z. Fire**  
Prize share: 1/2



Photo: J. Mollern  
**Craig C. Mello**  
Prize share: 1/2

## The Nobel Prize in Physiology or Medicine 2006



Photo: L. Cicero  
**Andrew Z. Fire**  
Prize share: 1/2



Photo: J. Molfert  
**Craig C. Mello**  
Prize share: 1/2



FDA approval August 2018

# Design stories

## The Nobel Prize in Physiology or Medicine 2006



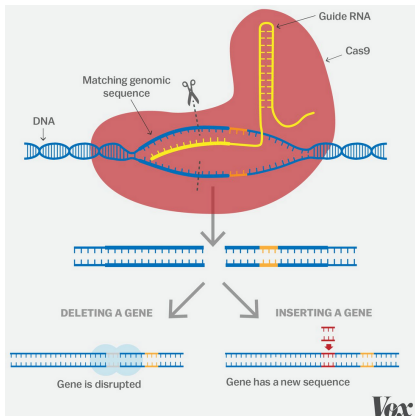
Photo: L. Cicero  
**Andrew Z. Fire**  
Prize share: 1/2



Photo: J. Moftern  
**Craig C. Mello**  
Prize share: 1/2



FDA approval August 2018



CRISPR: For better or worse...

## Abstract goals and means of molecular design

**But :** To achieve a predefined biological function, as abstracted by a model.

### Definition (Positive design)

To satisfy constraints induced by a model of function

**In practice:** To optimize affinity of interaction, to favor thermodynamic stability of a molecule, to respect sequence composition biases. . .

### Definition (Negative design)

To avoid unwanted functions

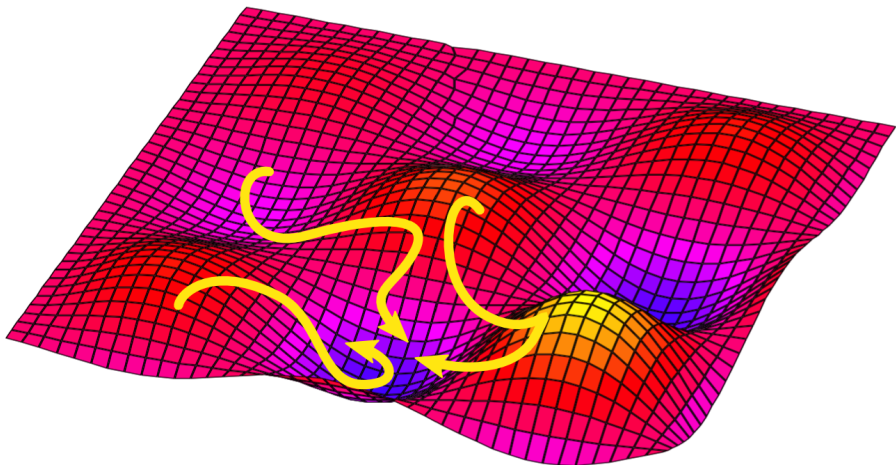
**In practice:** To avoid off-target interactions, non-functional alternative foldings, kinetic traps. . . (inverse combinatorial problems)

### In the context of RNA:

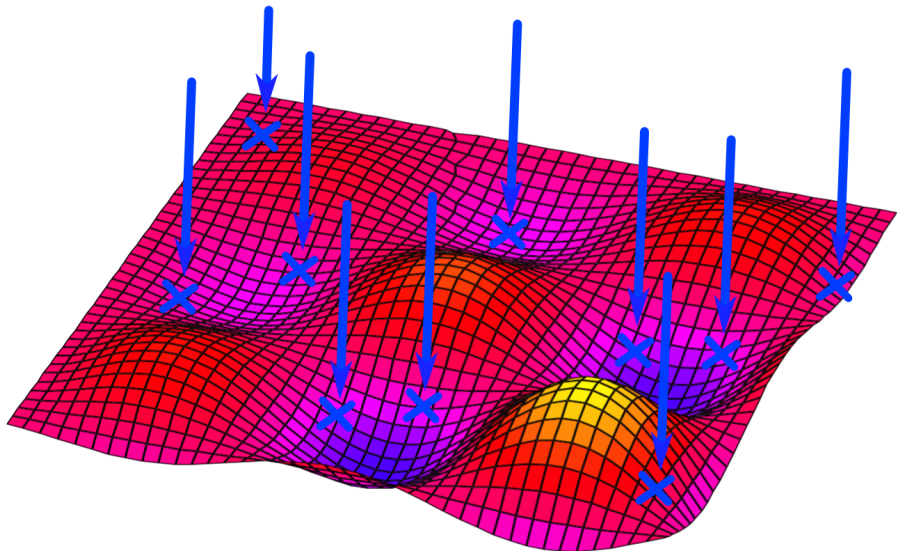
- ▶ **Positive design:** Seq/struct comparison, composition, +/- motifs, energie(s)  
→ Random generation, CSP
- ▶ **Negative design:** Target structure → Minimum Free-Energy + Boltzmann prob ↗  
→ Local search, exp algorithms, black magic (heuristics, \*NN, crowdsourcing. . .)



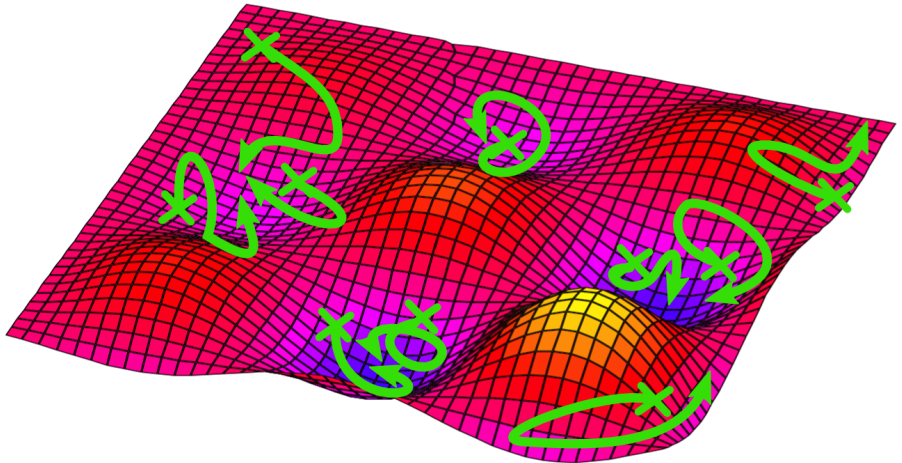
## Negative (Local) vs Positive (Global)



## Negative (Local) vs Positive (Global)



## Negative (Local) vs Positive (Global)



# **Part 1. Negative design**

## Existing approaches for negative design

Based on local search...

- ▶ RNAInverse - TBI Vienna
- ▶ Info-RNA - Backofen@Freiburg
- ▶ RNA-SSD - Condon@UBC
- ▶ (Inca)RNAFBinv - Barash@BGU
- ▶ NUPack - Pierce@Caltech

... bio-inspired algorithms...

- ▶ FRNAKenstein - Hein@Oxford
- ▶ AntaRNA - Backofen@Freiburg
- ▶ ERD - Ganjtabesh@Tehran

... exact approaches...

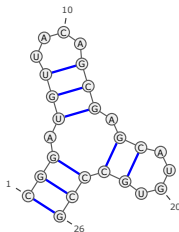
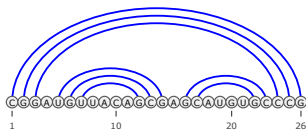
- ▶ RNAIFold - Clote@Boston College
- ▶ CO4 - Will@Leipzig

## Typical issues:

- ▶ Naive initialization strategies
- ▶ Synthesized sequences do not necessarily fold properly (kinetics)
- ▶ Overly GC-rich sequences
- ▶ No negative results

⇒ **Combinatorial foundations!**

# Energy model



**This talk:** Restriction to **valid** base-pairs =  $\{(A, U), (G, C), (G, U)\}$

▶ **RNA structure  $R$ :** Set of base pairs (BPs)

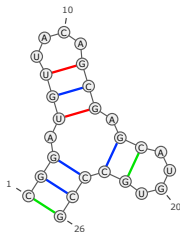
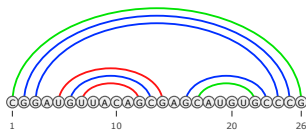
▶ **Motifs:** Connected positions + content (e.g. Base Pairs, Stems, Loops, ...)

▶ **Energy model:**

**Motif**  $\rightarrow$  Free-energy contribution  $\Delta(m, a) \in \mathbb{R} \cup \{+\infty\}$ ,  $m \subset [1, n]$ ,  $a \in \Sigma^{|m|}$

**Free-energy  $E(S, R)$ :** Sum of energies for motifs in  $R$ , given sequence  $S$

# Energy model



**This talk:** Restriction to **valid** base-pairs =  $\{(A, U), (G, C), (G, U)\}$

▶ **RNA structure  $R$ :** Set of base pairs (BPs)

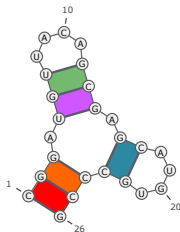
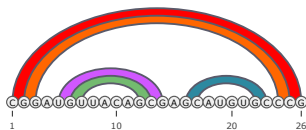
▶ **Motifs:** Connected positions + content (e.g. Base Pairs, Stacking, Loops...)

▶ **Energy model:**

**Motif**  $\rightarrow$  Free-energy contribution  $\Delta(m, a) \in \mathbb{R} \cup \{+\infty\}$ ,  $m \subset [1, n]$ ,  $a \in \Sigma^{|m|}$

**Free-energy  $E(S, R)$ :** Sum of energies for motifs in  $R$ , given sequence  $S$

# Energy model



**This talk:** Restriction to **valid** base-pairs =  $\{(A, U), (G, C), (G, U)\}$

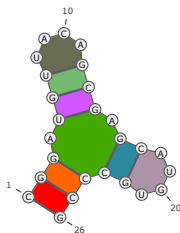
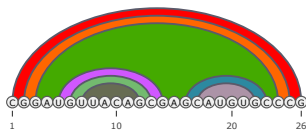
- ▶ **RNA structure  $R$ :** Set of base pairs (BPs)
- ▶ **Motifs:** Connected positions + content (e.g. Base Pairs, Stacking, Loops...)
- ▶ **Energy model:**

**Motif**  $\rightarrow$  Free-energy contribution  $\Delta(m, a) \in \mathbb{R} \cup \{+\infty\}$ ,  $m \subset [1, n]$ ,  $a \in \Sigma^{|m|}$

**Free-energy  $E(S, R)$ :** Sum of energies for motifs in  $R$ , given sequence  $S$



# Energy model



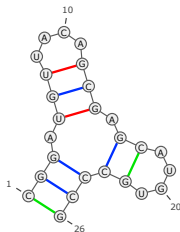
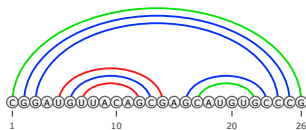
**This talk:** Restriction to **valid** base-pairs =  $\{(A, U), (G, C), (G, U)\}$

- ▶ **RNA structure  $R$ :** Set of base pairs (BPs)
- ▶ **Motifs:** Connected positions + content (e.g. Base Pairs, Stacking, Loops. . .)
- ▶ **Energy model:**

**Motif**  $\rightarrow$  Free-energy contribution  $\Delta(m, a) \in \mathbb{R} \cup \{+\infty\}$ ,  $m \subset [1, n]$ ,  $a \in \Sigma^{|m|}$

**Free-energy  $E(S, R)$ :** Sum of energies for motifs in  $R$ , given sequence  $S$

# Energy model



**This talk:** Restriction to **valid** base-pairs =  $\{(A, U), (G, C), (G, U)\}$

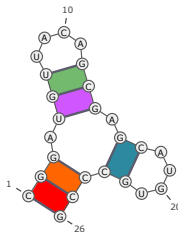
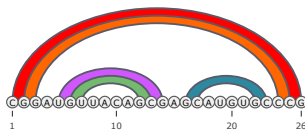
- ▶ **RNA structure  $R$ :** Set of base pairs (BPs)
- ▶ **Motifs:** Connected positions + content (e.g. Base Pairs, Stacking, Loops. . .)
- ▶ **Energy model:**

**Motif**  $\rightarrow$  Free-energy contribution  $\Delta(m, a) \in \mathbb{R} \cup \{+\infty\}$ ,  $m \subset [1, n]$ ,  $a \in \Sigma^{|m|}$

**Free-energy  $E(S, R)$ :** Sum of energies for motifs in  $R$ , given sequence  $S$

$$E_R = 2 \cdot \Delta \left( \begin{array}{c} \textcircled{U} \\ | \\ \textcircled{G} \end{array} \right) + 4 \cdot \Delta \left( \begin{array}{c} \textcircled{G} \\ | \\ \textcircled{C} \end{array} \right) + 2 \cdot \Delta \left( \begin{array}{c} \textcircled{C} \\ | \\ \textcircled{G} \end{array} \right)$$

# Energy model



**This talk:** Restriction to **valid** base-pairs =  $\{(A, U), (G, C), (G, U)\}$

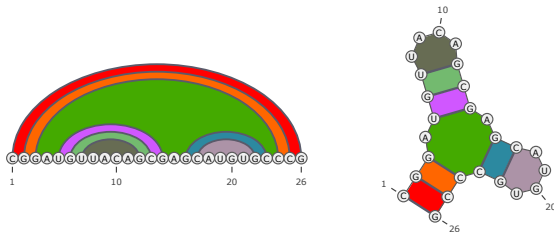
- ▶ **RNA structure  $R$ :** Set of base pairs (BPs)
- ▶ **Motifs:** Connected positions + content (e.g. Base Pairs, Stacking, Loops. . .)
- ▶ **Energy model:**

**Motif**  $\rightarrow$  Free-energy contribution  $\Delta(m, a) \in \mathbb{R} \cup \{+\infty\}$ ,  $m \subset [1, n]$ ,  $a \in \Sigma^{|m|}$

**Free-energy  $E(S, R)$ :** Sum of energies for motifs in  $R$ , given sequence  $S$

$$E_R = \Delta \left( \begin{array}{|c|c|} \hline C & G \\ \hline G & C \\ \hline \end{array} \right) + \Delta \left( \begin{array}{|c|c|} \hline G & G \\ \hline C & C \\ \hline \end{array} \right) + \Delta \left( \begin{array}{|c|c|} \hline U & G \\ \hline G & C \\ \hline \end{array} \right) + \Delta \left( \begin{array}{|c|c|} \hline U & G \\ \hline G & C \\ \hline \end{array} \right) + \Delta \left( \begin{array}{|c|c|} \hline U & G \\ \hline G & C \\ \hline \end{array} \right)$$

# Energy model



**This talk:** Restriction to **valid** base-pairs =  $\{(A, U), (G, C), (G, U)\}$

- ▶ **RNA structure  $R$ :** Set of base pairs (BPs)
- ▶ **Motifs:** Connected positions + content (e.g. Base Pairs, Stacking, Loops. . .)
- ▶ **Energy model:**

**Motif**  $\rightarrow$  Free-energy contribution  $\Delta(m, a) \in \mathbb{R} \cup \{+\infty\}$ ,  $m \subset [1, n]$ ,  $a \in \Sigma^{|m|}$

**Free-energy  $E(S, R)$ :** Sum of energies for motifs in  $R$ , given sequence  $S$

$$\begin{aligned}
 E_R = & \Delta \left( \begin{array}{c} \text{C} \quad \text{G} \\ | \quad | \\ \text{G} \quad \text{C} \end{array} \right) + \Delta \left( \begin{array}{c} \text{G} \quad \text{G} \\ | \quad | \\ \text{C} \quad \text{C} \end{array} \right) + \Delta \left( \begin{array}{c} \text{U} \quad \text{G} \\ | \quad | \\ \text{G} \quad \text{C} \end{array} \right) + \Delta \left( \begin{array}{c} \text{U} \quad \text{G} \\ | \quad | \\ \text{G} \quad \text{C} \end{array} \right) + \Delta \left( \begin{array}{c} \text{U} \quad \text{G} \\ | \quad | \\ \text{G} \quad \text{C} \end{array} \right) \\
 & + \Delta \left( \begin{array}{c} \text{A} \quad \text{C} \quad \text{A} \\ / \quad | \quad \backslash \\ \text{U} \quad \quad \text{G} \end{array} \right) + \Delta \left( \begin{array}{c} \text{U} \quad \text{G} \quad \text{A} \\ / \quad | \quad \backslash \\ \text{G} \quad \quad \text{G} \end{array} \right) + \Delta \left( \begin{array}{c} \text{C} \quad \text{A} \\ / \quad \backslash \\ \text{U} \quad \text{G} \end{array} \right)
 \end{aligned}$$

# RNA Inverse Folding

## Definition (INVERSE-FOLDING( $E$ ) problem)

**Input:** Secondary structure  $R$  + Energy distance  $\Delta > 0$ .

**Output:** RNA sequence  $S \in \Sigma^*$  such that:

$$\forall R' \in \mathcal{S}_{|S|} \setminus \{R\} : E(S, R') \geq E(S, R) + \Delta$$

or  $\emptyset$  if no such sequence exists.

**Difficult problem:** Probably no **obvious** DP decomposition

- ▶ NP-hard problem [Bonnet *et al*, RECOMB'18]. . . after almost 30 years!
- ▶ Existing algorithms: Heuristics or Exponential-time
- ▶ **Reason(s):** Non locality, no theoretical framework, too many parameters. . .

## Example

## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales *et al*, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  unpaired base(s) ( $m_{3,0}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable

[Yao *et al*, ACM-BCB'19]

# Designability in simple BP-based energy models

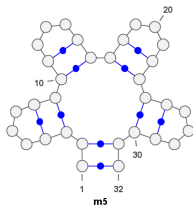
Partial characterization of **designable** structures [Hales *et al*, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  *unpaired base(s)* ( $m_{3\circ}$  motif).

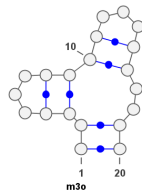
**Theorem:** Similar motifs exist for any **energy model** and **design criterion**

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable

[Yao *et al*, ACM-BCB'19]



$m_5$



$m_{3\circ}$

## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales et al, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  unpaired base(s) ( $m_{3\circ}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable

[Yao et al, ACM-BCB'19]

- ▶  $\exists$  **Separated** coloring for structure  $\Rightarrow$  Designable (+  $\Theta(n)$  algo.)

Base pairs  $\rightarrow$  3 colors:  $\bullet \rightarrow G \cdot C$ ;  $\circ \rightarrow C \cdot G$ ;  $\bullet \rightarrow A \cdot U$  or  $U \cdot A$ .

**Coloring rules:** Within each loop,  $\#\bullet \leq 1$ ,  $\#\circ \leq 1$ ,  $\#\bullet \leq 2$  and  $\#\bullet + \#\circ < 2$

**Level** of a base pair =  $\#\bullet - \#\circ$  on path to root.

**Separated** coloring =  $\bullet$  and unpaired positions occur at **different** levels



## Separated Coloring (example)

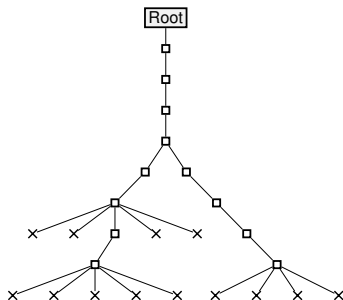
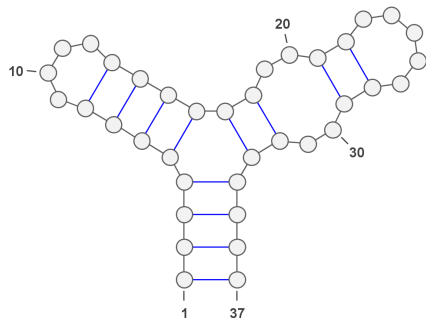
Base pairs  $\rightarrow$  3 colors:

●  $\rightarrow$  G · C;

○  $\rightarrow$  C · G;

●  $\rightarrow$  A · U or U · A.

**Coloring rules:** Within each loop,  $\# \bullet \leq 1$ ,  $\# \circ \leq 1$ ,  $\# \bullet \leq 2$  and  $\# \bullet + \# \circ < 2$



## Separated Coloring (example)

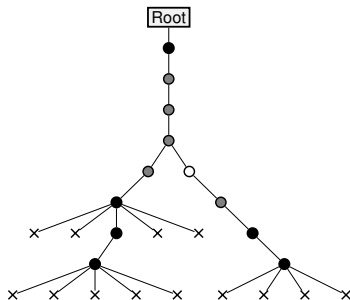
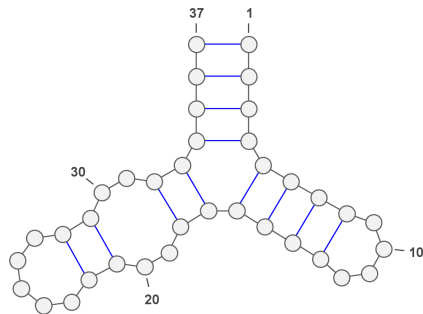
Base pairs  $\rightarrow$  3 colors:

●  $\rightarrow$  G · C;

○  $\rightarrow$  C · G;

●  $\rightarrow$  A · U or U · A.

**Coloring rules:** Within each loop,  $\# \bullet \leq 1$ ,  $\# \circ \leq 1$ ,  $\# \bullet \leq 2$  and  $\# \bullet + \# \circ < 2$



## Separated Coloring (example)

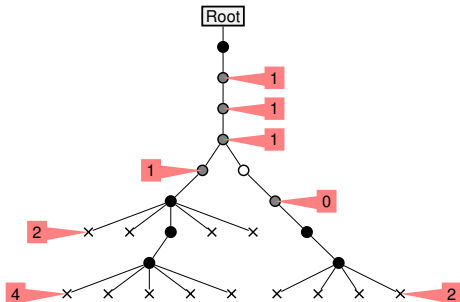
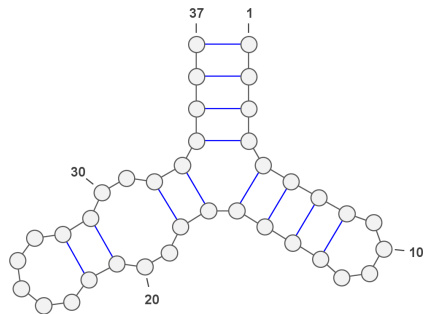
Base pairs  $\rightarrow$  3 colors:

●  $\rightarrow$  G · C;

○  $\rightarrow$  C · G;

●  $\rightarrow$  A · U or U · A.

**Coloring rules:** Within each loop,  $\# \bullet \leq 1$ ,  $\# \circ \leq 1$ ,  $\# \bullet \leq 2$  and  $\# \bullet + \# \circ < 2$



## Separated Coloring (example)

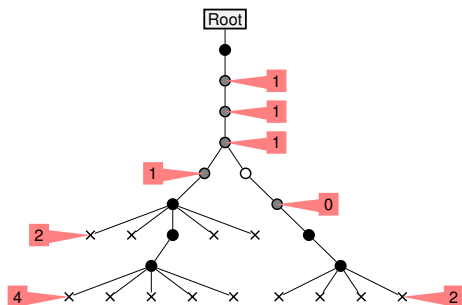
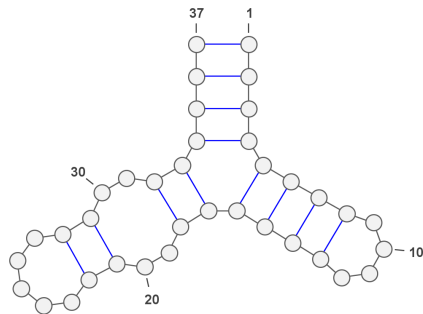
Base pairs  $\rightarrow$  3 colors:

●  $\rightarrow$  G · C;

○  $\rightarrow$  C · G;

●  $\rightarrow$  A · U or U · A.

**Coloring rules:** Within each loop,  $\#\bullet \leq 1$ ,  $\#\circ \leq 1$ ,  $\#\bullet \leq 2$  and  $\#\bullet + \#\circ < 2$



Levels of ●:  $\{0, 1\}$  + Levels of unpaired/leaves:  $\{2, 4\}$   $\Rightarrow$  Coloring is **separated**

## Separated Coloring (example)

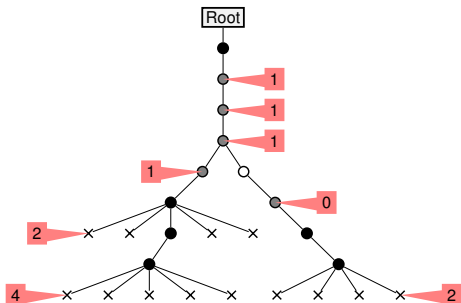
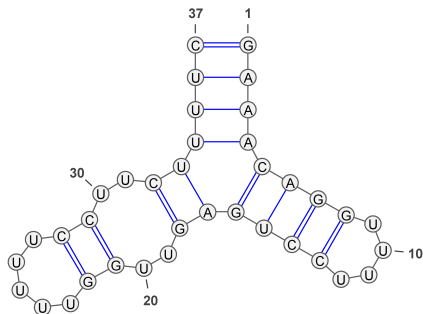
Base pairs  $\rightarrow$  3 colors:

●  $\rightarrow$  G · C;

○  $\rightarrow$  C · G;

●  $\rightarrow$  A · U or U · A.

**Coloring rules:** Within each loop,  $\#\bullet \leq 1$ ,  $\#\circ \leq 1$ ,  $\#\bullet \leq 2$  and  $\#\bullet + \#\circ < 2$



Levels of ●:  $\{0, 1\}$  + Levels of unpaired/leaves:  $\{2, 4\}$   $\Rightarrow$  Coloring is **separated**

**Design:** GAAAAGUUGGUUUUCCUUCUCAGGUUUUCCUGUUUC

## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales et al, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  unpaired base(s) ( $m_{3\circ}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable [Yao et al, ACM-BCB'19]

- ▶  $\exists$  **Separated** coloring for structure  $\Rightarrow$  Designable (+  $\Theta(n)$  algo.)

**Corollary:** Approximate design for any structure avoiding  $m_5$  and  $m_{3\circ}$  in  $\Theta(n)$  time

**Idea:** Insert new BPs on helices to **offset** unpaired/leaves and 

Open problems

## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales *et al*, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  unpaired base(s) ( $m_{3\circ}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable [Yao *et al*, ACM-BCB'19]

- ▶  $\exists$  **Separated** coloring for structure  $\Rightarrow$  Designable (+  $\Theta(n)$  algo.)

**Corollary:** Approximate design for any structure avoiding  $m_5$  and  $m_{3\circ}$  in  $\Theta(n)$  time

**Idea:** Insert new BPs on helices to **offset** unpaired/leaves and ●

Open problems

## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales et al, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  unpaired base(s) ( $m_{3\circ}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable [Yao et al, ACM-BCB'19]

- ▶  $\exists$  **Separated** coloring for structure  $\Rightarrow$  Designable (+  $\Theta(n)$  algo.)

**Corollary:** Approximate design for any structure avoiding  $m_5$  and  $m_{3\circ}$  in  $\Theta(n)$  time

**Idea:** Insert new BPs on helices to **offset** unpaired/leaves and ●

### Open problems

- ▶ Algorithm/characterization of separated-colorable tree?
- ▶ Inserting min #Base pairs: Complexity? Algorithm?
- ▶ Complex color sets for more realistic energy models?
- ▶ FPT design for some (yet unknown) parameters?
- ▶ In practice? Design (approximate) backbone + local search?



## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales et al, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  *unpaired base(s)* ( $m_{3\circ}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable [Yao et al, ACM-BCB'19]

- ▶  $\exists$  **Separated** coloring for structure  $\Rightarrow$  Designable (+  $\Theta(n)$  algo.)

**Corollary:** Approximate design for any structure avoiding  $m_5$  and  $m_{3\circ}$  in  $\Theta(n)$  time

**Idea:** Insert new BPs on helices to **offset** unpaired/leaves and ●

### Open problems

- ▶ Algorithm/characterization of separated-colorable tree?
- ▶ Inserting min #Base pairs: Complexity? Algorithm?
- ▶ Complex color sets for more realistic energy models?
- ▶ FPT design for some (yet unknown) parameters?
- ▶ In practice? Design (approximate) backbone + local search?

## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales et al, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  unpaired base(s) ( $m_{3\circ}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable [Yao et al, ACM-BCB'19]

- ▶  $\exists$  **Separated** coloring for structure  $\Rightarrow$  Designable (+  $\Theta(n)$  algo.)

**Corollary:** Approximate design for any structure avoiding  $m_5$  and  $m_{3\circ}$  in  $\Theta(n)$  time

**Idea:** Insert new BPs on helices to **offset** unpaired/leaves and ●

### Open problems

- ▶ Algorithm/characterization of separated-colorable tree?
- ▶ Inserting min #Base pairs: Complexity? Algorithm?
- ▶ Complex color sets for more realistic energy models?
- ▶ FPT design for some (yet unknown) parameters?
- ▶ In practice? Design (approximate) backbone + local search?

## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales et al, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  *unpaired base(s)* ( $m_{3\circ}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable [Yao et al, ACM-BCB'19]

- ▶  $\exists$  **Separated** coloring for structure  $\Rightarrow$  Designable (+  $\Theta(n)$  algo.)

**Corollary:** Approximate design for any structure avoiding  $m_5$  and  $m_{3\circ}$  in  $\Theta(n)$  time

**Idea:** Insert new BPs on helices to **offset** unpaired/leaves and ●

### Open problems

- ▶ Algorithm/characterization of separated-colorable tree?
- ▶ Inserting min #Base pairs: Complexity? Algorithm?
- ▶ Complex color sets for more realistic energy models?
- ▶ FPT design for some (yet unknown) parameters?
- ▶ In practice? Design (approximate) backbone + local search?

## Designability in simple BP-based energy models

Partial characterization of **designable** structures [Hales et al, CPM'15+Algorithmica'17]

- ▶ **Saturated structures:** Designable  $\Leftrightarrow$  Degree of multiloops  $\leq 4$  (+  $\Theta(n)$  algo.)
- ▶ Designable  $\Rightarrow$  No multiloop of *degree*  $\geq 5$  ( $m_5$  motif), or *degree*  $\geq 3$  with  $\geq 1$  unpaired base(s) ( $m_{3\circ}$  motif).

**Corollary:** Only an **exponentially small** (on  $n$ ) fraction of structs is designable [Yao et al, ACM-BCB'19]

- ▶  $\exists$  **Separated** coloring for structure  $\Rightarrow$  Designable (+  $\Theta(n)$  algo.)

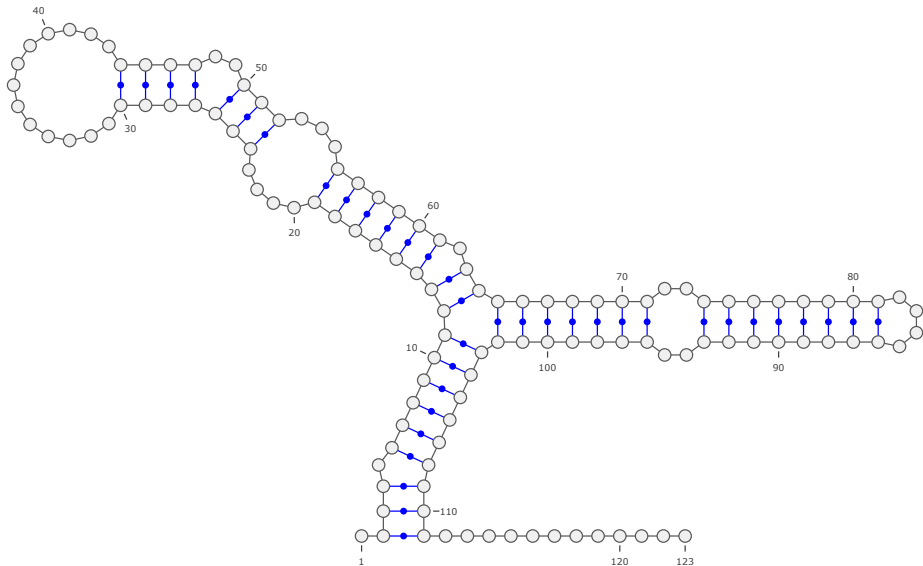
**Corollary:** Approximate design for any structure avoiding  $m_5$  and  $m_{3\circ}$  in  $\Theta(n)$  time

**Idea:** Insert new BPs on helices to **offset** unpaired/leaves and ●

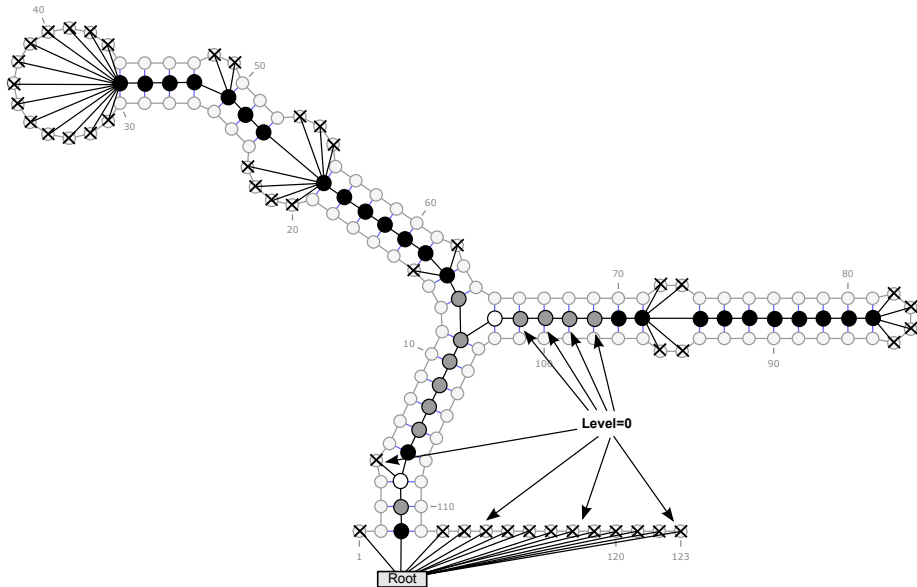
### Open problems

- ▶ Algorithm/characterization of separated-colorable tree?
- ▶ Inserting min #Base pairs: Complexity? Algorithm?
- ▶ Complex color sets for more realistic energy models?
- ▶ FPT design for some (yet unknown) parameters?
- ▶ In practice? Design (approximate) backbone + local search?

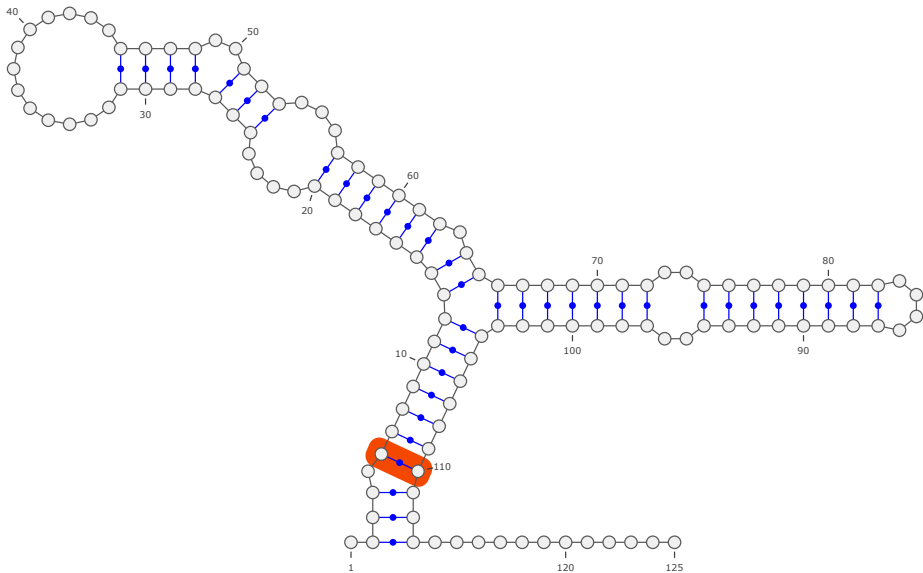
# In real life...



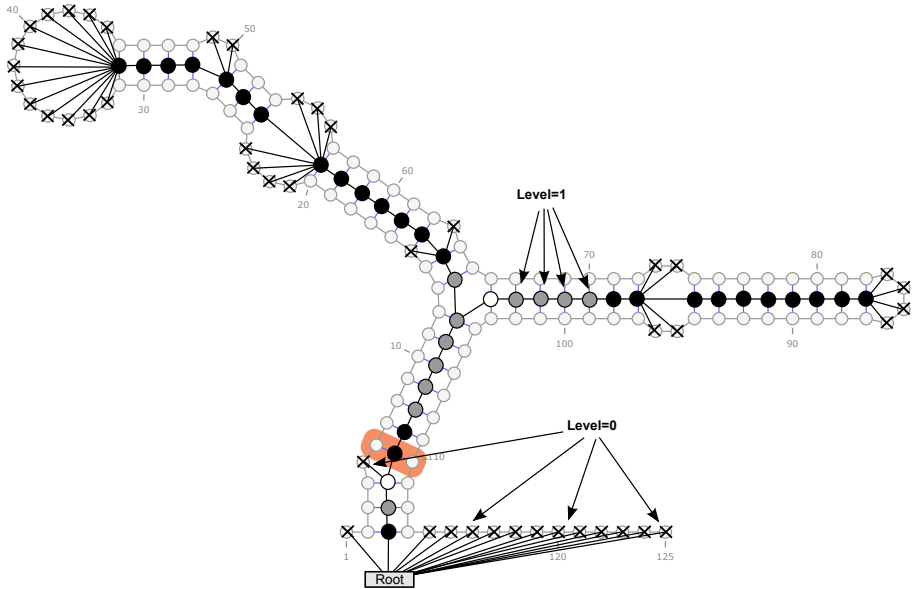
# In real life...



# In real life...



# In real life...

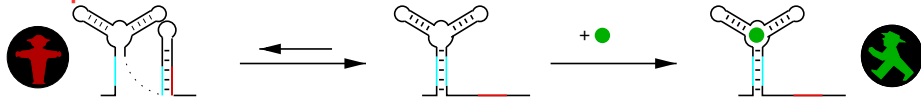




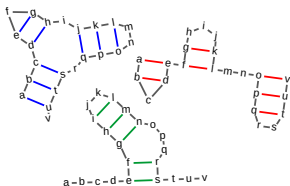
## **Part 2. Multiple positive design of RNA**

# Multiple RNA design: Motivation

**Example:** *Riboswitch* for translation control



Multiple target structures  $\rightarrow$  Multiple design of RNAs



abcdefghijklmnopqrstuv  
((((((.)).(((..))).)).).  
((.))((..))..(((.)))  
.....((((((.)))....))....

**Objective:** To **randomly** generate RNA sequences under constraints

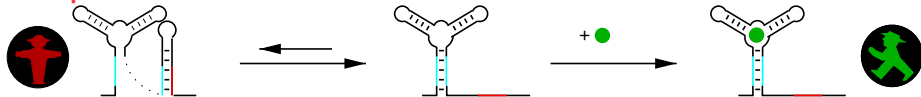
- 1 **Validity** for targeted structures wrt base pairing nucleotides
- 2 **Stability** (low free-energy, comparable across structures... ) of target structures
- 3 **Constrained composition:** (prescribed GC content), +/- motifs...

**Stochastic backtrack:** Pre-count and generate **valid** sequence (uniform distrib.)

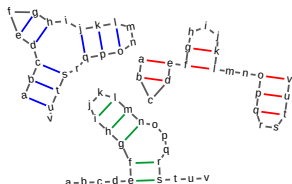
+ Further refinements using **local search**

# Multiple RNA design: Motivation

**Example:** *Riboswitch* for translation control



**Multiple target structures** → **Multiple design of RNAs**



abcdefghijklmnopqrstuv  
((((().)).(((.)).)).).  
((.))((...))..(((.)))  
.....((((((...)))....))....

**Objective:** To **randomly** generate RNA sequences under constraints

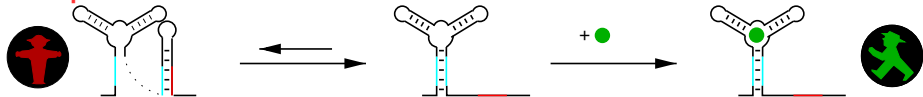
- 1 **Validity** for targeted structures wrt base pairing nucleotides
- 2 **Stability** (low free-energy, comparable across structures...) of target structures
- 3 **Constrained composition:** (prescribed GC content), +/- motifs...

**Stochastic backtrack:** Pre-count and generate **valid** sequence (uniform distrib.)

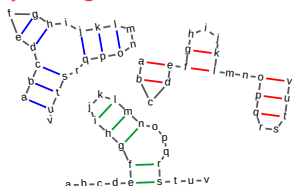
+ Further refinements using **local search**

# Multiple RNA design: Motivation

**Example:** *Riboswitch* for translation control



Multiple target structures  $\rightarrow$  Multiple design of RNAs



abcdefghijklmnopqrstuv  
((((().)).(((.)).)).).  
((.))((...))..(((.)))  
....((((((...)))....))....

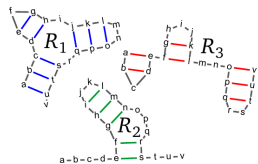
**Objective:** To **randomly** generate RNA sequences under constraints

- 1 **Validity** for targeted structures wrt base pairing nucleotides
- 2 **Stability** (low free-energy, comparable across structures. . .) of target structures
- 3 **Constrained composition:** (prescribed GC content), +/- motifs. . .

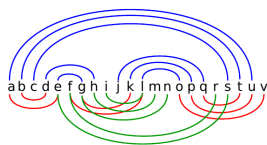
**Stochastic backtrack:** Pre-count and generate **valid** sequence (uniform distrib.)

+ Further refinements using **local search**

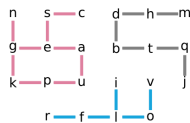
## Our problem (simplified)



i) Input Structures



ii) Merged Base-Pairs



iii) Compatibility Graph

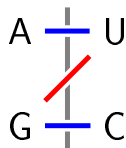
**Question:** How many valid sequences over  $\Sigma^n := \{A, C, G, U\}^n$  ?

### Problem (#ValidSequences)

**Input:** Secondary structures  $\mathcal{R} = \{R_1, \dots, R_k\}$  of length  $n$

**Output:** Num. of valid sequences

$$|\{S \in \Sigma^n \mid \forall (i, j) \in R_\ell, (S_i, S_j) \text{ forms a valid base pair}\}|$$

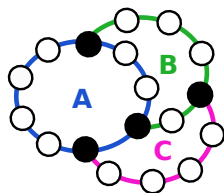


Valid base pairs

## State of the art

### Abfalter/Flamm/Stadler 2003:

- ▶ Ear decomposition [Whitney 1932]
- ▶ *Peel input graph* as paths  $A_1, \dots, A_k$  such that only the ends of  $A_i$  are in  $\cup_{j>i} A_j$
- ▶ **Dynamic programming:** Counting #valid paths for each component, conditioned by nucleotide chosen for its **anchors** (black nodes);
- ▶ Careful **combination** of values yields #valid sequences.



**Complexity:**  $\Theta(n \cdot 4^\Omega)$  where  $\Omega = \text{Max \#anchors}$ . Worst-case:  $\Omega \in \Theta(n)$

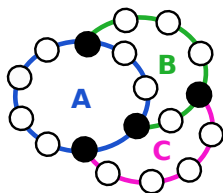
### Some comments:

- ▶ Is this optimal? Other algorithms/parameters?
- ▶ Which extensions possible? (Multidim.) Boltzmann-Gibbs distrib.
- ▶ Is this exp. really necessary? **Probably** since counting #P-hard

## State of the art

### Abfalter/Flamm/Stadler 2003:

- ▶ Ear decomposition [Whitney 1932]
- ▶ *Peel input graph* as paths  $A_1, \dots, A_k$  such that only the ends of  $A_i$  are in  $\cup_{j>i} A_j$
- ▶ **Dynamic programming:** Counting #valid paths for each component, conditioned by nucleotide chosen for its **anchors** (black nodes);
- ▶ Careful **combination** of values yields #valid sequences.



**Complexity:**  $\Theta(n \cdot 4^\Omega)$  where  $\Omega = \text{Max \#anchors}$ . Worst-case:  $\Omega \in \Theta(n)$

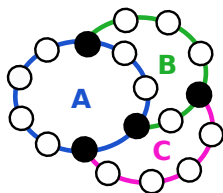
### Some comments:

- ▶ Is this optimal? Other algorithms/parameters?
- ▶ Which extensions possible? (Multidim.) Boltzmann-Gibbs distrib.
- ▶ Is this exp. really necessary? **Probably** since counting #P-hard

## State of the art

### Abfalter/Flamm/Stadler 2003:

- ▶ Ear decomposition [Whitney 1932]
- ▶ *Peel input graph* as paths  $A_1, \dots, A_k$  such that only the ends of  $A_i$  are in  $\cup_{j>i} A_j$
- ▶ **Dynamic programming:** Counting #valid paths for each component, conditioned by nucleotide chosen for its **anchors** (black nodes);
- ▶ Careful **combination** of values yields #valid sequences.



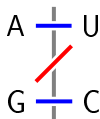
**Complexity:**  $\Theta(n \cdot 4^\Omega)$  where  $\Omega = \text{Max \#anchors}$ . Worst-case:  $\Omega \in \Theta(n)$

### Some comments:

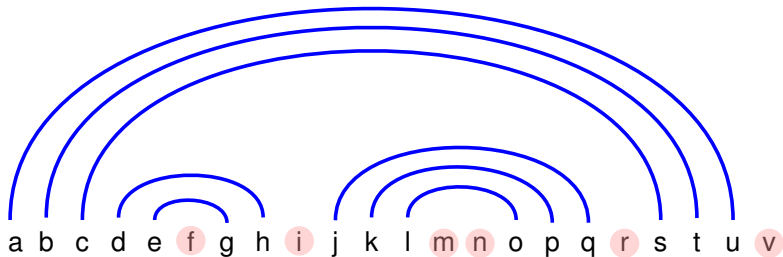
- ▶ Is this optimal? Other algorithms/parameters?
- ▶ Which extensions possible? (Multidim.) Boltzmann-Gibbs distrib.
- ▶ Is this exp. really necessary? **Probably** since counting #P-hard



## Counting valid sequences: WC/Wobble + single structure



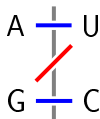
Valid base pairs (BPs) = Including **Wobble** base pairs



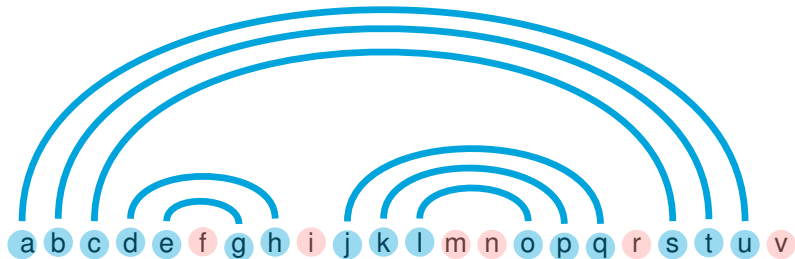
**Question:** How many **valid** sequences?

**Answer:**  $4^{\#Unpaired} \times 6^{\#BPs} \rightarrow 6879707136$

## Counting valid sequences: WC/Wobble + single structure



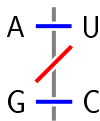
**Valid base pairs (BPs)** = Including **Wobble** base pairs



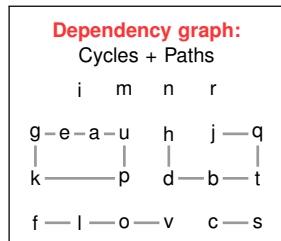
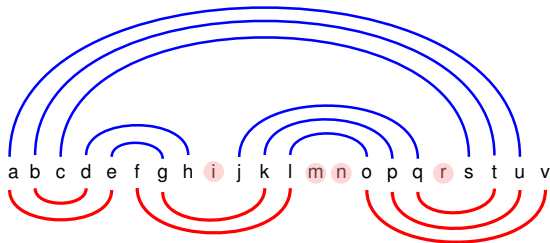
**Question:** How many **valid** sequences?

**Answer:**  $4^{\#Unpaired} \times 6^{\#BPs} \rightarrow 6\ 879\ 707\ 136$

## Counting valid sequences: WC/Wobble + Two structures



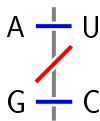
Valid base pairs (BPs) = Including **Wobble** base pairs



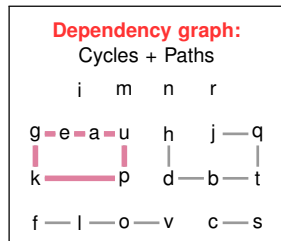
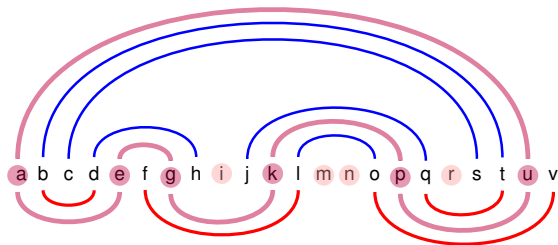
**Question:** How many **valid** sequences?

**Answer:**  $\neq \emptyset!$  (dep. graph and valid BPs both **bipartite** [Flamm *et al*, RNA 2001])

## Counting valid sequences: WC/Wobble + Two structures



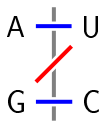
Valid base pairs (BPs) = Including **Wobble** base pairs



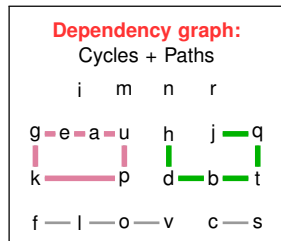
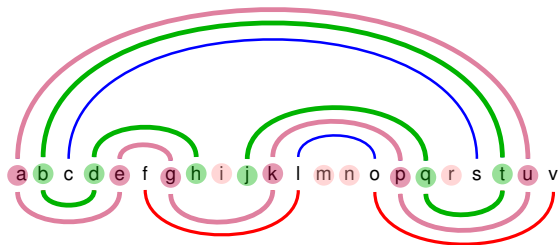
**Question:** How many **valid** sequences?

**Answer:**  $\neq \emptyset!$  (dep. graph and valid BPs both **bipartite** [Flamm *et al*, RNA 2001])

## Counting valid sequences: WC/Wobble + Two structures



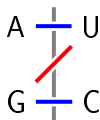
Valid base pairs (BPs) = Including **Wobble** base pairs



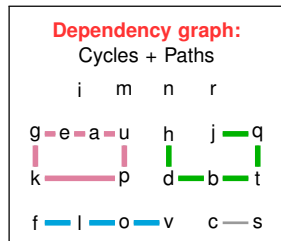
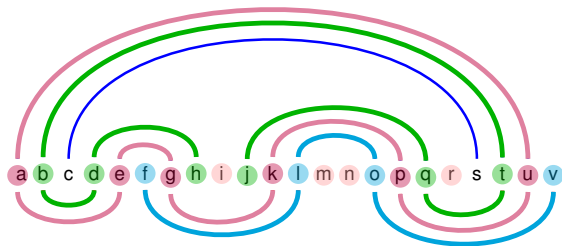
**Question:** How many **valid** sequences?

**Answer:**  $\neq \emptyset!$  (dep. graph and valid BPs both **bipartite** [Flamm *et al*, RNA 2001])

## Counting valid sequences: WC/Wobble + Two structures



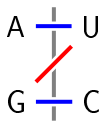
Valid base pairs (BPs) = Including **Wobble** base pairs



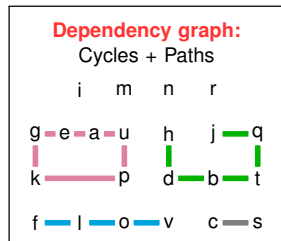
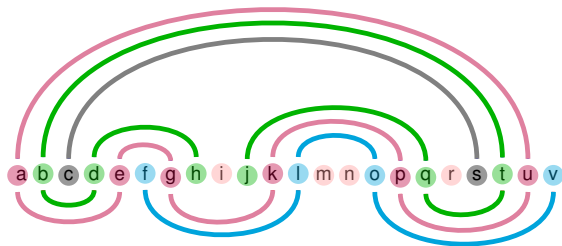
**Question:** How many **valid** sequences?

**Answer:**  $\neq \emptyset!$  (dep. graph and valid BPs both **bipartite** [Flamm *et al*, RNA 2001])

## Counting valid sequences: WC/Wobble + Two structures



Valid base pairs (BPs) = Including **Wobble** base pairs



**Question:** How many **valid** sequences?

**Answer:**  $\neq \emptyset!$  (dep. graph and valid BPs both **bipartite** [Flamm *et al*, RNA 2001])

$$\# \text{Designs}(G) = \prod_{c \in \text{CC}(G)} \# \text{Designs}(cc)$$

## Counting valid sequences for paths and cycles

$p(n)$  : #Valid sequences for **path** of length  $n$ .

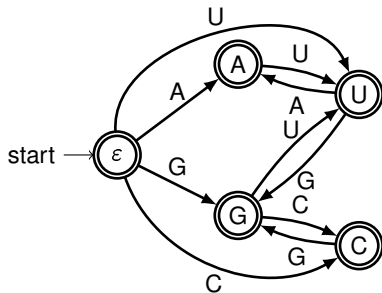
$c(n)$  : #Valid sequences for **cycle** of length  $n$ .

**Theorem (#Valid sequences for paths and cycles)**

$$p(n) = 2 \mathcal{F}_{n+2} \quad \text{et} \quad c(n) = 2 \mathcal{F}_n + 4 \mathcal{F}_{n-1}$$

where  $\mathcal{F}_n$  is the  $n$ -th Fibonacci number.

**For paths:** A simple automaton...



**Remark:**  $A \leftrightarrow C/G \leftrightarrow U$  symmetry



## Counting valid sequences for paths and cycles

$p(n)$  : #Valid sequences for **path** of length  $n$ .

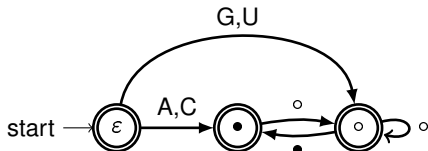
$c(n)$  : #Valid sequences for **cycle** of length  $n$ .

### Theorem (#Valid sequences for paths and cycles)

$$p(n) = 2 \mathcal{F}_{n+2} \quad \text{et} \quad c(n) = 2 \mathcal{F}_n + 4 \mathcal{F}_{n-1}$$

where  $\mathcal{F}_n$  is the  $n$ -th Fibonacci number.

**For paths:** A simple automaton...



**Remark:**  $A \leftrightarrow C/G \leftrightarrow U$  symmetry

## Counting valid sequences for paths and cycles

$p(n)$  : #Valid sequences for **path** of length  $n$ .

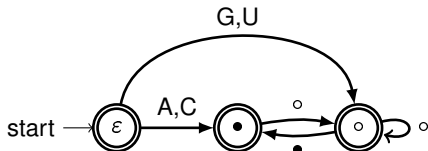
$c(n)$  : #Valid sequences for **cycle** of length  $n$ .

### Theorem (#Valid sequences for paths and cycles)

$$p(n) = 2 \mathcal{F}_{n+2} \quad \text{et} \quad c(n) = 2 \mathcal{F}_n + 4 \mathcal{F}_{n-1}$$

where  $\mathcal{F}_n$  is the  $n$ -th Fibonacci number.

**For paths:** A simple automaton...



**Remark:**  $A \leftrightarrow C/G \leftrightarrow U$  symmetry

$$m_{\bullet}(n) = m_{\circ}(n - 1)$$

## Counting valid sequences for paths and cycles

$p(n)$  : #Valid sequences for **path** of length  $n$ .

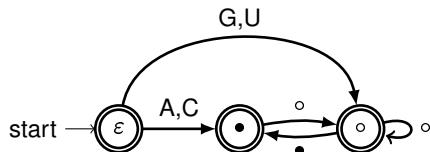
$c(n)$  : #Valid sequences for **cycle** of length  $n$ .

### Theorem (#Valid sequences for paths and cycles)

$$p(n) = 2 \mathcal{F}_{n+2} \quad \text{et} \quad c(n) = 2 \mathcal{F}_n + 4 \mathcal{F}_{n-1}$$

where  $\mathcal{F}_n$  is the  $n$ -th Fibonacci number.

**For paths:** A simple automaton...



**Remark:**  $A \leftrightarrow C/G \leftrightarrow U$  symmetry

$$m_{\bullet}(n) = m_{\circ}(n-1)$$

$$m_{\circ}(n) = m_{\circ}(n-1) + m_{\bullet}(n-1)$$

$$= m_{\circ}(n-1) + m_{\circ}(n-2)$$

$$= \mathcal{F}(n+2)$$

## Counting valid sequences for paths and cycles

$p(n)$  : #Valid sequences for **path** of length  $n$ .

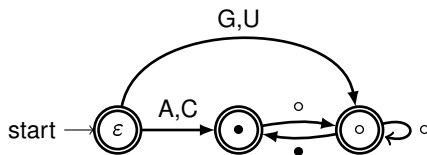
$c(n)$  : #Valid sequences for **cycle** of length  $n$ .

### Theorem (#Valid sequences for paths and cycles)

$$p(n) = 2\mathcal{F}_{n+2} \quad \text{et} \quad c(n) = 2\mathcal{F}_n + 4\mathcal{F}_{n-1}$$

where  $\mathcal{F}_n$  is the  $n$ -th Fibonacci number.

**For paths:** A simple automaton...



**Remark:**  $A \leftrightarrow C/G \leftrightarrow U$  symmetry

$$m_{\bullet}(n) = m_{\circ}(n-1)$$

$$m_{\circ}(n) = m_{\circ}(n-1) + m_{\bullet}(n-1)$$

$$= m_{\circ}(n-1) + m_{\circ}(n-2)$$

$$= \mathcal{F}(n+2)$$

(Since  $m_{\circ}(0) = 1$  and  $m_{\circ}(1) = 2$ )

## Counting valid sequences for paths and cycles

$p(n)$  : #Valid sequences for **path** of length  $n$ .

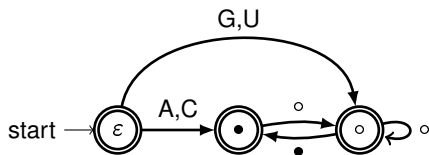
$c(n)$  : #Valid sequences for **cycle** of length  $n$ .

### Theorem (#Valid sequences for paths and cycles)

$$p(n) = 2\mathcal{F}_{n+2} \quad \text{et} \quad c(n) = 2\mathcal{F}_n + 4\mathcal{F}_{n-1}$$

where  $\mathcal{F}_n$  is the  $n$ -th Fibonacci number.

**For paths:** A simple automaton...



**Remark:**  $A \leftrightarrow C/G \leftrightarrow U$  symmetry

$$m_{\bullet}(n) = m_{\circ}(n-1)$$

$$\begin{aligned} m_{\circ}(n) &= m_{\circ}(n-1) + m_{\bullet}(n-1) \\ &= m_{\circ}(n-1) + m_{\circ}(n-2) \\ &= \mathcal{F}(n+2) \end{aligned}$$

(Since  $m_{\circ}(0) = 1$  and  $m_{\bullet}(1) = 2$ )

$$p(n) := m_{\epsilon}(n) = 2m_{\bullet}(n-1) + 2m_{\circ}(n-1) = 2(\mathcal{F}(n) + \mathcal{F}(n+1)) = 2\mathcal{F}(n+2)$$

## Counting valid sequences for paths and cycles

$p(n)$  : #Valid sequences for **path** of length  $n$ .

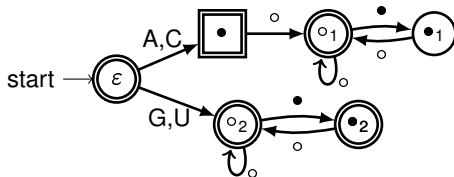
$c(n)$  : #Valid sequences for **cycle** of length  $n$ .

**Theorem (#Valid sequences for paths and cycles)**

$$p(n) = 2 \mathcal{F}_{n+2} \quad \text{et} \quad c(n) = 2 \mathcal{F}_n + 4 \mathcal{F}_{n-1}$$

where  $\mathcal{F}_n$  is the  $n$ -th Fibonacci number.

**For cycles:** A slightly more complex automaton...



## Counting valid sequences for paths and cycles

$p(n)$  and  $c(n)$ : #Valid sequences for **paths** and **cycles** of length  $n$ .

### Theorem (#Valid sequences for paths and cycles)

$$p(n) = 2 \mathcal{F}_{n+2} \quad \text{et} \quad c(n) = 2 \mathcal{F}_n + 4 \mathcal{F}_{n-1}$$

where  $\mathcal{F}_n$  is the  $n$ -th Fibonacci number.

$G$ : Dependency graph, merging the two structures (max degree  $\leq 2$ ).  
 $G$  uniquely decomposed in  $\mathcal{P}(G)$  **paths** and  $\mathcal{C}(G)$  **cycles**.

### Theorem (#Valid sequences for 2-structures)

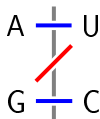
The number  $\#\text{Designs}(G)$  of valid sequences for  $G$  is

$$\#\text{Designs}(G) = \prod_{p \in \mathcal{P}(G)} 2 \mathcal{F}_{|p|+2} \times \prod_{c \in \mathcal{C}(G)} (2 \mathcal{F}_{|c|} + 4 \mathcal{F}_{|c|-1})$$

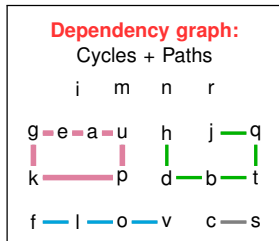
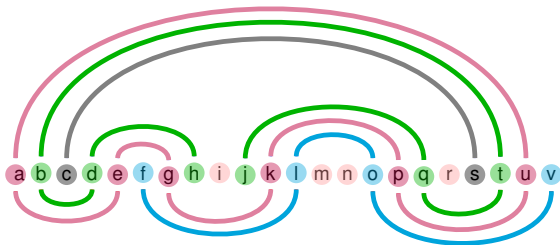
**Caterpillar tree:**  $\frac{(2+\sqrt{3}) \times (1+\sqrt{3})^n + (2-\sqrt{3}) \times (1-\sqrt{3})^n}{2}$  ( $n$  nodes)

**Complete binary:**  $2 a_k$  (height  $k$ )  $a_k = (a_{k-2} + 1)^4 + 2(a_{k-1} + 1)(a_{k-2} + 1)^2 + (a_{k-1} + 1)^2 - 1$

# Counting valid sequences: WC/Wobble + Two structures



Valid base pairs (BPs) = Including **Wobble** base pairs



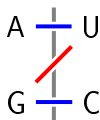
**Question:** How many **valid** sequences?

**Answer:**  $\neq \emptyset!$  (both BP and dependency graphs **bipartite**)

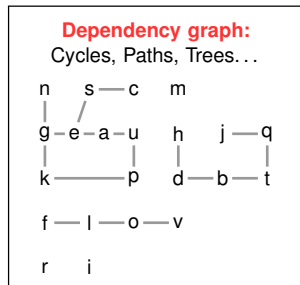
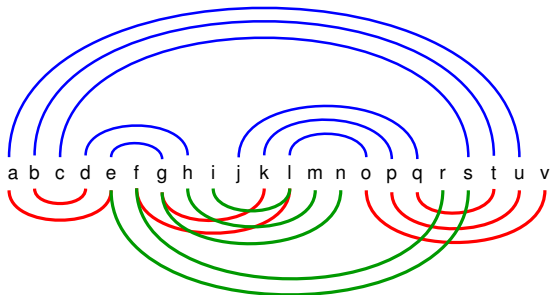
$$\# \text{Designs}(G) = \prod_{c \in CC(G)} \# \text{Designs}(cc) = 2\,322\,432$$



# Counting valid sequences: WC/Wobble + > 2 structures



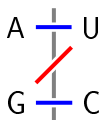
Valid base pairs (BPs) = Including **Wobble** base pairs



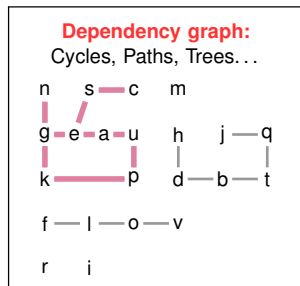
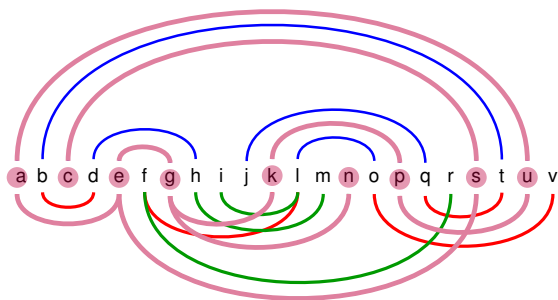
**Question:** How many valid **sequences**?

**Answer:** Non-bipartite  $\rightarrow \emptyset$ ; Bipartite  $\rightarrow \text{????}$

## Counting valid sequences: WC/Wobble + > 2 structures



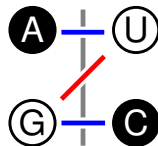
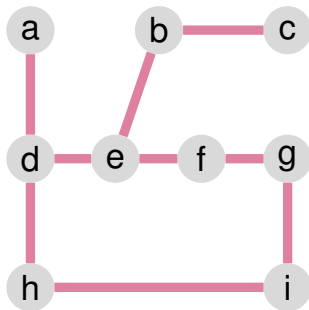
Valid base pairs (BPs) = Including **Wobble** base pairs



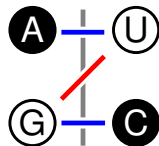
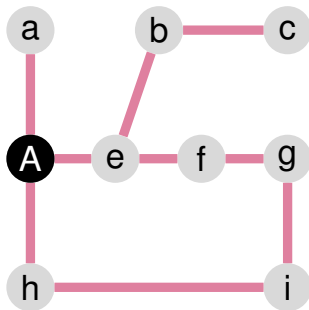
**Question:** How many valid **sequences**?

**Answer:** Non-bipartite  $\rightarrow \emptyset$ ; Bipartite  $\rightarrow \text{????}$

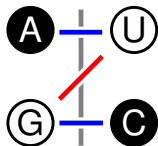
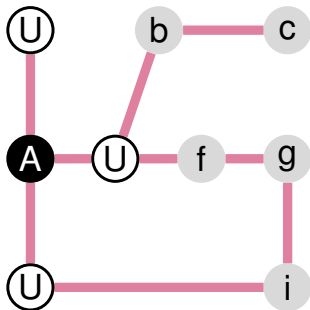
## Independent sets $\Leftrightarrow$ Valid sequences



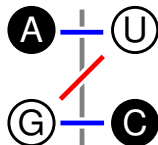
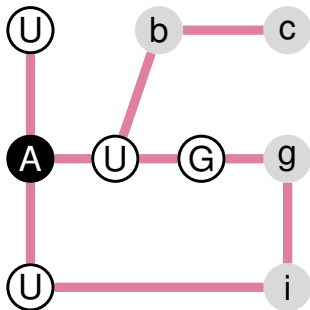
## Independent sets $\Leftrightarrow$ Valid sequences



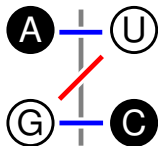
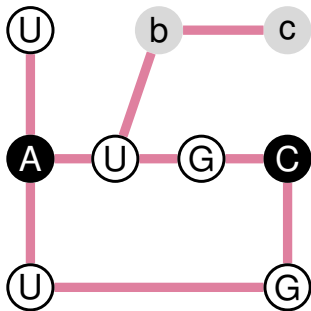
## Independent sets $\Leftrightarrow$ Valid sequences



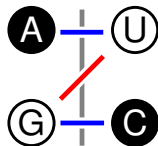
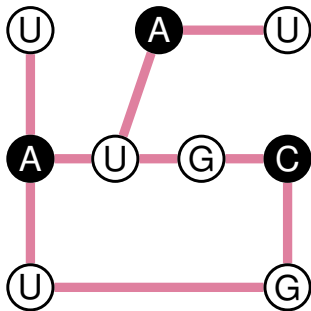
## Independent sets $\Leftrightarrow$ Valid sequences



## Independent sets $\Leftrightarrow$ Valid sequences

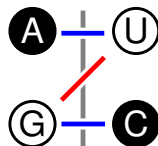
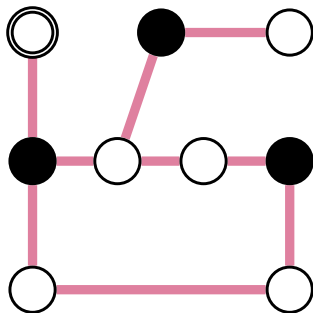


## Independent sets $\Leftrightarrow$ Valid sequences





## Independent sets $\Leftrightarrow$ Valid sequences

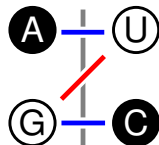
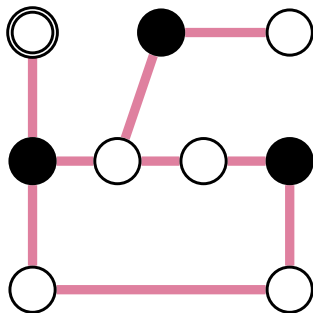


**Remark:** **Black circles** non-adjacent in valid sequences

Up to trivial symmetry\* (e.g. north-western position  $\in \{U, C\}$ ):

$$\text{Designs}^*(cc) \subseteq \text{IndSets}(cc)$$

## Independent sets $\Leftrightarrow$ Valid sequences



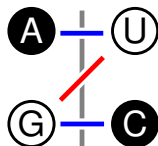
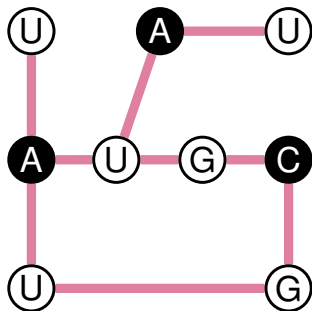
**Remark:** **Black circles** non-adjacent in valid sequences

Up to trivial symmetry\* (e.g. north-western position  $\in \{U, C\}$ ):

$$\text{Designs}^*(cc) \subseteq \text{IndSets}(cc)$$

Independent Sets (black) + NW  $\in \{U, C\} \Rightarrow$  Valid sequence

## Independent sets $\Leftrightarrow$ Valid sequences



**Remark:** **Black circles** non-adjacent in valid sequences

Up to trivial symmetry\* (e.g. north-western position  $\in \{U, C\}$ ):

$$\text{Designs}^*(cc) \subseteq \text{IndSets}(cc)$$

Independent Sets (black) + NW  $\in \{U, C\} \Rightarrow$  Valid sequence

$\Rightarrow$  Bijection between  $\text{Designs}^*(cc)$  and  $\text{IndSets}(cc)$ .

## Valid sequences and independent sets

### Theorem (#Designs and ind. sets in connected bipartite graphs)

Let  $G$  be a **bipartite and connected** dependency graph:

$$\#Designs(G) = 2 \times \#Designs^*(G) = 2 \times \#IndSets(G)$$

For **bipartite** dependency graph  $G$ , one has:

$$\#Designs(G) = \prod_{cc \in CC(G)} 2 \times \#IndSets(cc) = 2^{|CC(G)|} \times \#IndSets(G)$$

**But**  $\#IndSets(G)$  is **#P-hard** on bipartite graphs (**#BIS**) [Dyer & Greenhill'00]  
(+ Any graph  $G$  is the dependency graph of some structure family)

**So**  $\exists$  Poly-Time algorithm for  $\#Designs(G) \rightarrow$  Poly-Time algorithm for **#BIS**...

### Theorem

Counting  $\#Designs$  is **#P-hard**.

No Poly-Time algorithm for  $\#Designs(G)$  **unless**  $\#P = FP (\Rightarrow P = NP)$

## Valid sequences and independent sets

### Theorem (#Designs and ind. sets in connected bipartite graphs)

Let  $G$  be a **bipartite and connected** dependency graph:

$$\#Designs(G) = 2 \times \#Designs^*(G) = 2 \times \#IndSets(G)$$

For **bipartite** dependency graph  $G$ , one has:

$$\#Designs(G) = \prod_{cc \in CC(G)} 2 \times \#IndSets(cc) = 2^{|CC(G)|} \times \#IndSets(G)$$

But  $\#IndSets(G)$  is **#P-hard** on bipartite graphs (**#BIS**) [Dyer & Greenhill'00]  
(+ Any graph  $G$  is the dependency graph of some structure family)

So  $\exists$  Poly-Time algorithm for  $\#Designs(G) \rightarrow$  Poly-Time algorithm for **#BIS**...

### Theorem

Counting  $\#Designs$  is **#P-hard**.

No Poly-Time algorithm for  $\#Designs(G)$  **unless**  $\#P = FP (\Rightarrow P = NP)$

## Valid sequences and independent sets

### Theorem (#Designs and ind. sets in connected bipartite graphs)

Let  $G$  be a **bipartite and connected** dependency graph:

$$\#Designs(G) = 2 \times \#Designs^*(G) = 2 \times \#IndSets(G)$$

For **bipartite** dependency graph  $G$ , one has:

$$\#Designs(G) = \prod_{cc \in CC(G)} 2 \times \#IndSets(cc) = 2^{|CC(G)|} \times \#IndSets(G)$$

**But**  $\#IndSets(G)$  is **#P-hard** on bipartite graphs (**#BIS**) [Dyer & Greenhill'00]

(+ Any graph  $G$  is the dependency graph of some structure family)

**So**  $\exists$  Poly-Time algorithm for  $\#Designs(G) \rightarrow$  Poly-Time algorithm for **#BIS**...

### Theorem

Counting  $\#Designs$  is #P-hard.

No Poly-Time algorithm for  $\#Designs(G)$  **unless**  $\#P = FP (\Rightarrow P = NP)$

## Valid sequences and independent sets

### Theorem (#Designs and ind. sets in connected bipartite graphs)

Let  $G$  be a **bipartite and connected** dependency graph:

$$\#Designs(G) = 2 \times \#Designs^*(G) = 2 \times \#IndSets(G)$$

For **bipartite** dependency graph  $G$ , one has:

$$\#Designs(G) = \prod_{cc \in CC(G)} 2 \times \#IndSets(cc) = 2^{|CC(G)|} \times \#IndSets(G)$$

**But**  $\#IndSets(G)$  is **#P-hard** on bipartite graphs ( $\#BIS$ ) [Dyer & Greenhill'00]

(+ Any graph  $G$  is the dependency graph of some structure family)

**So**  $\exists$  Poly-Time algorithm for  $\#Designs(G) \rightarrow$  Poly-Time algorithm for  $\#BIS$ ...

### Theorem

Counting  $\#Designs$  is **#P-hard**.

No Poly-Time algorithm for  $\#Designs(G)$  **unless**  $\#P = FP (\Rightarrow P = NP)$

## Consequences

### Corollary (#Approximability for $\leq 5$ structures) [Weitz'06]

For  $\leq 5$  structures (crossings allowed), #Design( $G$ ) can be approximated within **any ratio** in **Poly-time** (PTAS)

### Corollary (#BIS-hardness for $> 5$ structures) [Cai, Galanis *et al* 16]

For more than 5 structures (crossings allowed), #Design is **equally as hard** to approximate as general #BIS.

**Why crossings/Pseudoknots?** Because any bipartite graph of max degree  $\Delta$  can be **decomposed** in  $\Delta$  matchings in **Poly-Time** (Vizing theorem).

Connection between **counting** and **sampling** [Jerrum/Valliant/Vazirani'86].

### Conjecture (#BIS-hardness of multiple positive design)

**Quasi-uniform generation** as hard as approximation of general #BIS

$\Rightarrow$  **Sampling** #P hard?



## Consequences

### Corollary (#Approximability for $\leq 5$ structures) [Weitz'06]

For  $\leq 5$  structures (crossings allowed), #Design( $G$ ) can be approximated within **any ratio** in **Poly-time** (PTAS)

### Corollary (#BIS-hardness for $> 5$ structures) [Cai, Galanis *et al* 16]

For more than 5 structures (crossings allowed), #Design is **equally as hard** to approximate as general #BIS.

**Why crossings/Pseudokots?** Because any bipartite graph of max degree  $\Delta$  can be **decomposed** in  $\Delta$  matchings in **Poly-Time** (Vizing theorem).

Connection between **counting** and **sampling** [Jerrum/Valliant/Vazirani'86].

Conjecture (#BIS-hardness of multiple positive design)

**Quasi-uniform generation** as hard as approximation of general #BIS

$\Rightarrow$  **Sampling** #P hard?

## Consequences

### Corollary (#Approximability for $\leq 5$ structures) [Weitz'06]

For  $\leq 5$  structures (crossings allowed), #Design( $G$ ) can be approximated within **any ratio** in **Poly-time** (PTAS)

### Corollary (#BIS-hardness for $> 5$ structures) [Cai, Galanis *et al* 16]

For more than 5 structures (crossings allowed), #Design is **equally as hard** to approximate as general #BIS.

**Why crossings/Pseudokots?** Because any bipartite graph of max degree  $\Delta$  can be **decomposed** in  $\Delta$  matchings in **Poly-Time** (Vizing theorem).

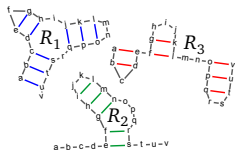
Connection between **counting** and **sampling** [Jerrum/Valiant/Vazirani'86].

### Conjecture (#BIS-hardness of multiple positive design)

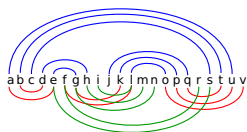
**Quasi-uniform generation** as hard as approximation of general #BIS

⇒ **Sampling** #P hard?

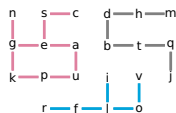
# Tree decomposition and Boltzmann sampling of sequences



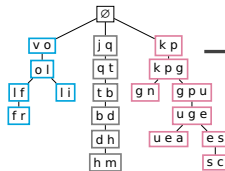
i) Input Structures



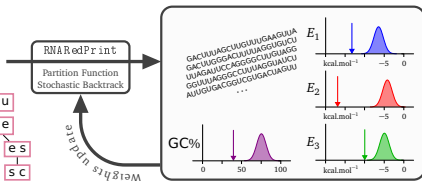
ii) Merged Base-Pairs



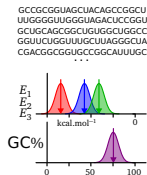
iii) Compatibility Graph



iv) Tree Decomposition



v) Weight Optimization (Adaptive Sampling)



vi) Final Designs

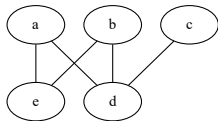
# Tree decomposition and width

**Tree decomposition**  $T$  for a graph  $G = (V, E)$ :

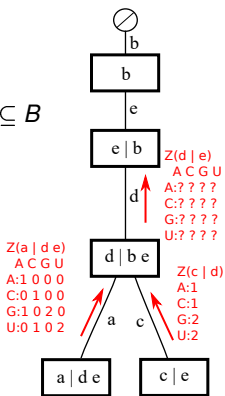
- Nodes of  $T =$  Some subsets of  $V$
- All vertices present:**  $\forall v \in V, \exists b \in B$  s.t.  $v \in b$
- All edges present:**  $\forall (v, v') \in E, \exists b \in B$  s.t.  $\{v, v'\} \subseteq b$
- Nodes having  $v \in V$  form a **connected** subtree

a	b	c	d	e
(	.	.	)	.
.	(	(	)	.
(	(	.	)	)

**Target structures**



**Dependency graph**



$b = \{b_1, b_2, \dots\}$  : node of  $D$

$T_b$  : subtree rooted at  $b$

$w$  : **Width** of tree decomposition  $D$  ( $= \max_{b \in B} |b| - 1$ )

$$\mathcal{Z}(T_b | b_2 \leftarrow v_2 \dots) = \sum_{\substack{b_1 \leftarrow v_1 \\ v_1 \in \{A, C, G, U\}}} \prod_{c \text{ fils de } b} \mathcal{Z}(T_c | b_1 \leftarrow v_1, b_2 \leftarrow v_2 \dots)$$

**Complexity:**  $\Theta(nmk + nk2^w)$  for **uniform generation** of  $m$  sequences ( $k$  structs)

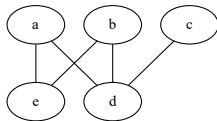
# Tree decomposition and width

**Tree decomposition**  $T$  for a graph  $G = (V, E)$ :

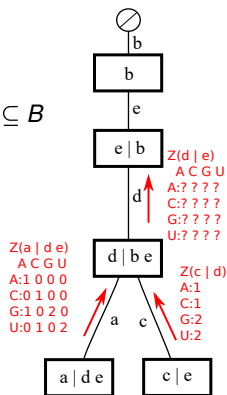
- Nodes of  $T =$  Some subsets of  $V$
- All vertices present:**  $\forall v \in V, \exists b \in B$  s.t.  $v \in b$
- All edges present:**  $\forall (v, v') \in E, \exists b \in B$  s.t.  $\{v, v'\} \subseteq b$
- Nodes having  $v \in V$  form a **connected** subtree

a	b	c	d	e
(	.	.	)	.
.	(	(	)	.
(	(	.	)	)

Target structures



Dependency graph



Tree decomposition

$b = \{b_1, b_2, \dots\}$  : node of  $D$

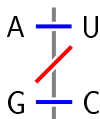
$T_b$  : subtree rooted at  $b$

$w$  : **Width** of tree decomposition  $D$  ( $= \max_{b \in B} |b| - 1$ )

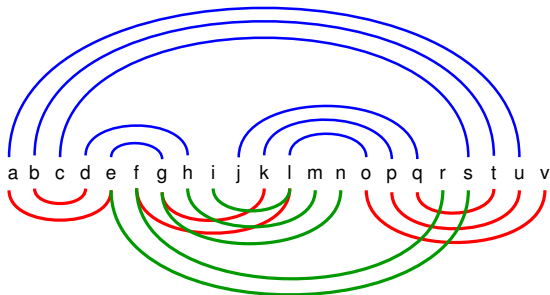
$$\mathcal{Z}(T_b | b_2 \leftarrow v_2 \dots) = \sum_{\substack{b_1 \leftarrow v_1 \\ v_1 \in \{A, C, G, U\}}} \prod_{c \text{ fils de } b} \mathcal{Z}(T_c | b_1 \leftarrow v_1, b_2 \leftarrow v_2 \dots)$$

**Complexity:**  $\Theta(nmk + nk2^w)$  for **uniform generation** of  $m$  sequences ( $k$  structs)

## Counting valid sequences: WC/Wobble + > 2 structures

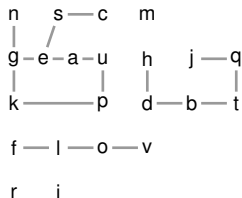


Valid base pairs (BPs) = Including **Wobble** base pairs



### Dependency graph:

Cycles, Paths, Trees...

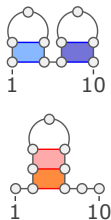


**Question:** How many valid **sequences**?

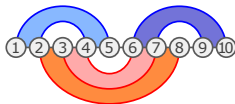
**Answer:** Non-bipartite  $\rightarrow \emptyset$ ; Bipartite  $\rightarrow 496\,672$

# Our problem for general free-energy models

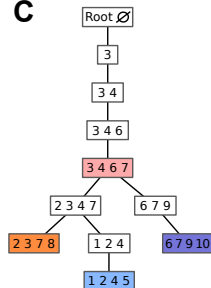
A



B



C



**Question:** Which partition function for **valid sequences**

## Problem (PFDesigns)

**Input:** Structures  $\mathcal{R} = \{R_1, \dots, R_k\}$  of length  $n$  + Weight  $(x_1, \dots, x_k)$

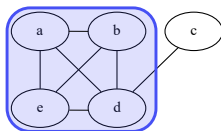
**Output:** Partition function

$$\mathcal{Z} = \sum_{\substack{S \in \Sigma^n \\ S \text{ valid for } \mathcal{R}}} \prod_{i=1}^k x_i^{E(S, R_i)}$$

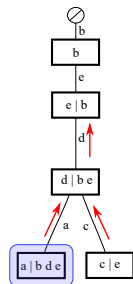
# Counting/sampling, the Boltzmann-Gibbs way

a	b	c	d	e
(	.	.	)	.
.	(	(	)	)
(	(	.	)	)

Target Structures



Dependency Hypergraph



Tree Decomposition

$b = \{b_1, b_2, \dots\}$  : node of  $D$

$T_b$  : subtree rooted at  $b$

$w$  : **Width** of treedecomposition  $D$

$$\mathcal{Z}(T_b \mid b_2 \leftarrow v_2 \dots) = \sum_{\substack{b_1 \leftarrow v_1 \\ v_1 \in \{A, C, G, U\}}} \prod_{i=1}^k \chi_i^{\sum_{E \in b} E(b, v_1, \dots)} \prod_{c \text{ fils de } b} \mathcal{Z}(T_c \mid b_1 \leftarrow v_1, \dots)$$

**Complexity:**  $\Theta(nmk + nk2^{w+\#CC})$  for sampling in Boltzmann-Gibbs distrib.

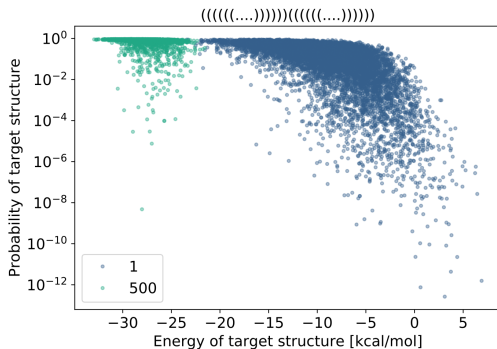


# Practical impact of Boltzmann-Gibbs sampling

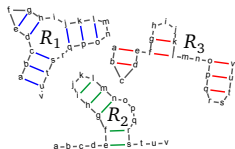
Boltzmann probability of **structure**  $R$ , pour une séquence  $S$ :

$$\mathbb{P}(R | S) = \frac{e^{-\frac{E(S,R)}{\beta T}}}{\mathcal{Z}_S} \quad \mathcal{Z}_S := \sum_R e^{-\frac{E(S,R)}{\beta T}}$$

Objectif classique du design négatif ( $\rightarrow$  spécificité)



# RNAredPrint: a flexible method for (positive) design



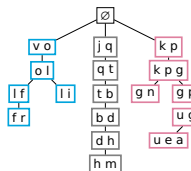
i) Input Structures



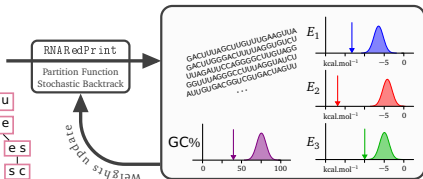
ii) Merged Base-Pairs



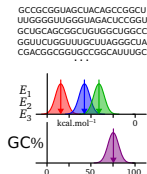
iii) Compatibility Graph



iv) Tree Decomposition



v) Weight Optimization (Adaptive Sampling)



vi) Final Designs

[Hammer/P/Wang/Will, RECOMB'18 + BMC Bioinfo 2019]

- ▶ **Fixed Parameter Tractable** algorithm based on **tree width**
- ▶ **Uniform or Boltzmann-Gibbs** sampling, to favor diversity and stability
- ▶ **Multidimensional Boltzmann sampling** for controlling free-energy, GC%...

<https://github.com/yannponty/RNAredPrint>

# Multidimensional Boltzmann sampling

## Multidimensional Boltzmann sampling [Bodini, P, DMTCS 2011]

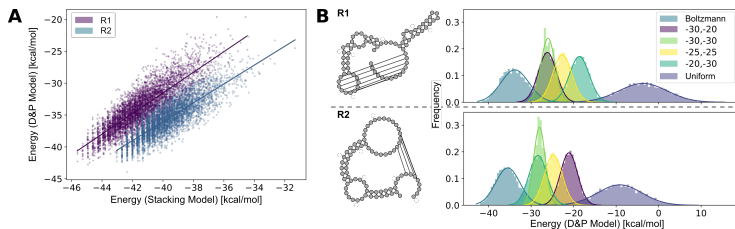
**Input:** Targeted free-energies  $(E_\ell^*)_{\ell=1}^k$ , weights  $(x_\ell)_{\ell=1}^k$  such that  $\mathbb{E}(E(w, S_\ell)) = E_\ell^*, \forall \ell$ :

$$\mathbb{P}(w \mid x_1 \cdots x_k) \sim \prod_{\ell=1}^k x_\ell^{E(w, S_\ell)} + \text{Efficient rejection} \rightarrow \mathcal{O}(n^{k/2}) \text{ exact}/\mathcal{O}(\alpha^k) \text{ approx.}$$

**Empirical** efficiency for additive *concentrated* constraints (GC%, dinucleotides ...)  
→ Partial functions → Hyper-edges, aka cliques<sup>1</sup>

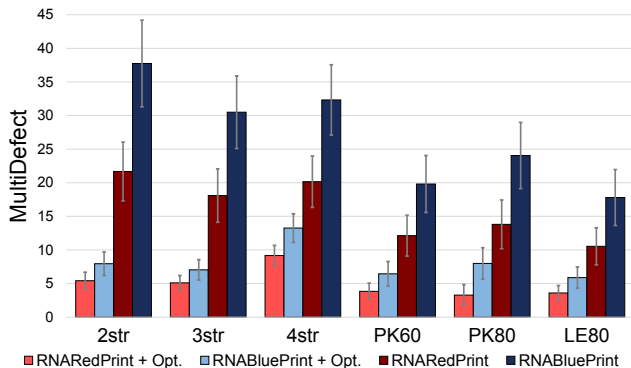


General framework for integer-valued constraints; Concentration tests.



<sup>1</sup>But tree width ↗

## Strangely enough, it actually works!



$$\text{MultiDefect}(S, R_1 \cdots R_k) := \frac{\sum_{\ell=1}^k E(S, R_\ell) - EFE(S)}{k} + \frac{\sum_{1 \leq \ell < j \leq k} |E(S, R_\ell) - E(S, R_j)|}{2 \binom{k}{2}}$$

where  $EFE$  = ensemble free-energy  $EFE(S) := -\beta T \log \mathcal{Z}_S$ .

# Conclusion

## Our contribution :

- ▶ General framework for generating constrained sequences  
Ideas similar to/generalized from CTE framework (R. Dechter);
- ▶ Application to multiple RNA design, proven #P hard;
- ▶ Uses efficient rejection scheme for practical control of complex constraints;
- ▶ Practical efficiency (reasonable tree width).

## Perspectives :

 Complexity of sequence generation for  $k < 5$  structures?

 How to deal with additional sequence constraints? (DFA "product")

 How to locally navigate the space of valid sequences? (Local search)

 How to simplify dense graphs? (DCA potentials)

# Conclusion

## Our contribution :

- ▶ General framework for generating constrained sequences  
Ideas similar to/generalized from CTE framework (R. Dechter);
- ▶ Application to multiple RNA design, proven #P hard;
- ▶ Uses efficient rejection scheme for practical control of complex constraints;
- ▶ Practical efficiency (reasonable tree width).

## Perspectives :

 Complexity of sequence generation for  $k < 5$  structures?

 How to deal with additional sequence constraints? (DFA "product")

 How to locally navigate the space of valid sequences? (Local search)

 How to simplify dense graphs? (DCA potentials)

# Conclusion

## Our contribution :

- ▶ General framework for generating constrained sequences  
Ideas similar to/generalized from CTE framework (R. Dechter);
- ▶ Application to multiple RNA design, proven #P hard;
- ▶ Uses efficient rejection scheme for practical control of complex constraints;
  - ▶ Practical efficiency (reasonable tree width).

## Perspectives :

 Complexity of sequence generation for  $k < 5$  structures?

 How to deal with additional sequence constraints? (DFA "product")

 How to locally navigate the space of valid sequences? (Local search)

 How to simplify dense graphs? (DCA potentials)

# Conclusion

## Our contribution :

- ▶ General framework for generating constrained sequences  
Ideas similar to/generalized from CTE framework (R. Dechter);
- ▶ Application to multiple RNA design, proven #P hard;
- ▶ Uses efficient rejection scheme for practical control of complex constraints;
- ▶ Practical efficiency (reasonable tree width).

## Perspectives :

 Complexity of sequence generation for  $k < 5$  structures?

 How to deal with additional sequence constraints? (DFA "product")

 How to locally navigate the space of valid sequences? (Local search)

 How to simplify dense graphs? (DCA potentials)



# Conclusion

## Our contribution :

- ▶ General framework for generating constrained sequences  
Ideas similar to/generalized from CTE framework (R. Dechter);
- ▶ Application to multiple RNA design, proven #P hard;
- ▶ Uses efficient rejection scheme for practical control of complex constraints;
- ▶ Practical efficiency (reasonable tree width).

## Perspectives :

 Complexity of sequence generation for  $k < 5$  structures?

 How to deal with additional sequence constraints? (DFA "product")

 How to locally navigate the space of valid sequences? (Local search)

 How to simplify dense graphs? (DCA potentials)


# Conclusion

## Our contribution :

- ▶ General framework for generating constrained sequences  
Ideas similar to/generalized from CTE framework (R. Dechter);
- ▶ Application to multiple RNA design, proven #P hard;
- ▶ Uses efficient rejection scheme for practical control of complex constraints;
- ▶ Practical efficiency (reasonable tree width).

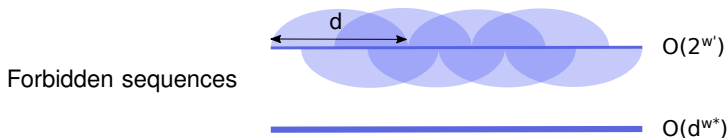
## Perspectives :

 Complexity of sequence generation for  $k < 5$  structures?

 How to deal with additional sequence constraints? (DFA "product")

 How to locally navigate the space of valid sequences? (Local search)

 How to simplify dense graphs? (DCA potentials)





# Conclusion


## Our contribution :

- ▶ General framework for generating constrained sequences  
Ideas similar to/generalized from CTE framework (R. Dechter);
- ▶ Application to multiple RNA design, proven #P hard;
- ▶ Uses efficient rejection scheme for practical control of complex constraints;
- ▶ Practical efficiency (reasonable tree width).

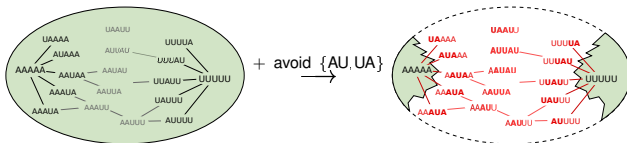
## Perspectives :

 Complexity of sequence generation for  $k < 5$  structures?

 How to deal with additional sequence constraints? (DFA "product")

 How to locally navigate the space of valid sequences? (Local search)

 How to simplify dense graphs? (DCA potentials)







# Conclusion

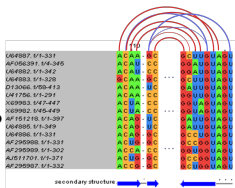
## Our contribution :

- ▶ General framework for generating constrained sequences  
Ideas similar to/generalized from CTE framework (R. Dechter);
- ▶ Application to multiple RNA design, proven #P hard;
- ▶ Uses efficient rejection scheme for practical control of complex constraints;
- ▶ Practical efficiency (reasonable tree width).

## Perspectives :

-  Complexity of sequence generation for  $k < 5$  structures?
-  How to deal with additional sequence constraints? (DFA "product")
-  How to locally navigate the space of valid sequences? (Local search)
-  How to simplify dense graphs? (DCA potentials)

Largest vertex set given tree-width *budget*?



# Merci – תודה – Thank you

## Collaborators:



Ecole Polytechnique

- ▶ M. Régnier, A. Héliou, H.T. Yao



Simon Fraser University

- ▶ J. Hales, J. Manuch, L. Stacho
- ▶ C. Chauve



McGill University

- ▶ J. Waldispühl



Université du Québec à Montréal

- ▶ V. Reinharz



University of Vienna

- ▶ S. Will, S. Hammer



Ben Gurion University

- ▶ D. Barash, M. Drory Retwitzer, A. Churkin

## Supported by:



Agence Nationale de la Recherche  
ANR

FWF

משרד המדע,  
הטכנולוגיה והחלל  
Ministry of Science, Technology & Space

