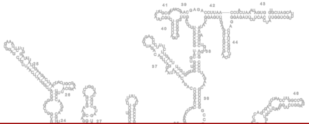
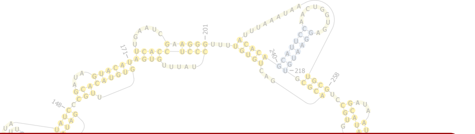
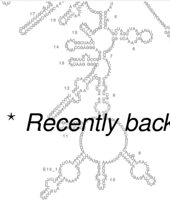
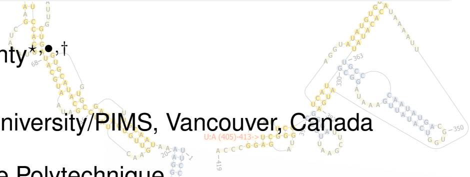




Towards firm foundations for the rational design of RNA molecules



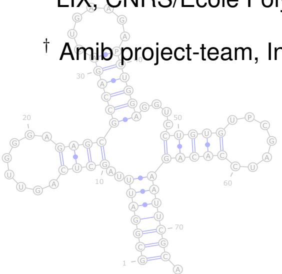


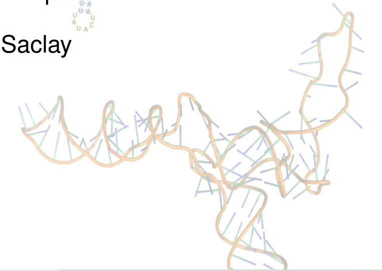
Yann Ponty[⋆,•,†]

[⋆] *Recently back from* Simon Fraser University/PIMS, Vancouver, Canada

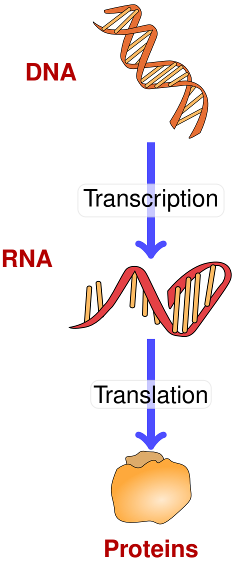
[•] LIX, CNRS/Ecole Polytechnique

[†] Amib project-team, Inria Saclay

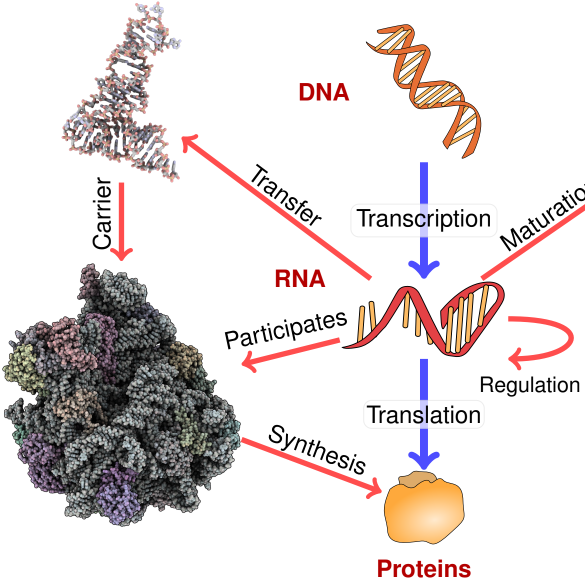




Fundamental *dogma* of molecular biology



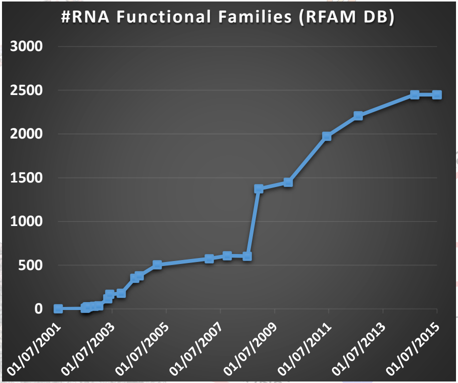
Fundamental *dogma* of molecular biology (v2.0)

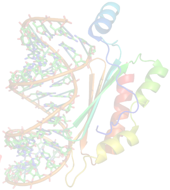


RNA functions

- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- . . .

Fundamental *dogma* of molecular biology

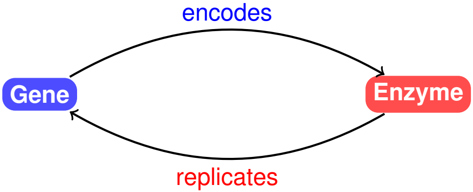




RNA functions

- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- . . .

RNA world: Resolving the *chicken vs egg* paradox at the origin of life...



The RNA World
THIRD EDITION

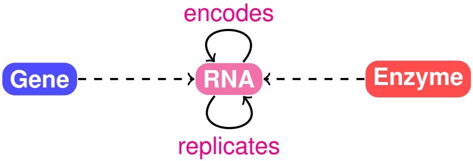
Edited by Raymond F. Gesteland
Thomas R. Cech
John F. Atkins

A gene big enough to specify an enzyme would be too big to replicate accurately without the aid of an enzyme of the very kind that it is trying to specify. So the system *apparently cannot get started*.

[...] This is the RNA World. To see how plausible it is, we need to look at why proteins are good at being enzymes but bad at being replicators; at why DNA is good at replicating but bad at being an enzyme; and finally why *RNA might just be good enough at both roles to break out of the Catch-22.*

R. Dawkins. *The Ancestor's Tale: A Pilgrimage to the Dawn of Evolution*

RNA world: Resolving the *chicken vs egg* paradox at the origin of life. . .



encodes

Gene RNA Enzyme

replicates

A gene big enough to specify an enzyme would be too big to replicate accurately without the aid of an enzyme of the very kind that it is trying to specify. So the system *apparently cannot get started*.

[. . .] This is the RNA World. To see how plausible it is, we need to look at why proteins are good at being enzymes but bad at being replicators; at why DNA is good at replicating but bad at being an enzyme; and finally why *RNA might just be good enough at both roles to break out of the Catch-22.*

R. Dawkins. *The Ancestor's Tale: A Pilgrimage to the Dawn of Evolution*

Controlled experiments through RNA design

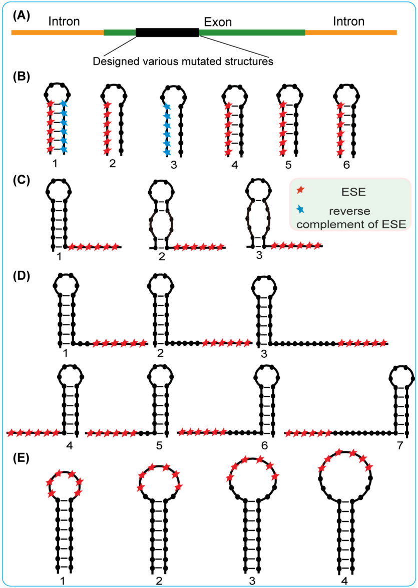
Motivation: Quantifying the impact of structure *S* on efficacy of a single Exon Splicing *E* Enhancers (ESE):

- Presence of given ESE motif *E*;
- Different structures $S_1, S_2 \ldots$;
- Avoid library of (∼1500!) documented ESEs motifs.

Objectives. Design RNA which:

1. Folds into a prescribed structure;
2. Features/avoids motifs.
3. Control GC%, Boltz. prob.. . . .

Structural context of ESE motif in transcript was shown to affect its functionality. [Liu *et al*, FEBS Lett. 2010]


(A)
Intron
Exon
Intron
Designed various mutated structures
(B)
1
2
3
4
5
6
(C)
1
2
3
ESE
reverse complement of ESE
(D)
1
2
3
4
5
6
7
(E)
1
2
3
4

Design objectives

Positive structural design

Optimize **affinity** of designed sequences towards target structure

Examples: Most stable sequence for given fold. . .

Negative structural design

Limit affinity of designed sequences towards **alternative structures**

Examples: Lowest free-energy, High Boltzmann probability/Low entropy. . .

Additional constraints:

- **Forbid** motif list to appear **anywhere** in design
- **Force** motif list to appear **each at least once**
- **Limit** available alternatives at certain positions
- **Control** overall composition (GC-content)

I. Inverse Folding

Designing a given structure

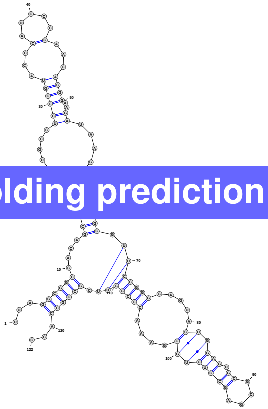
RNA sequence and structure(s)

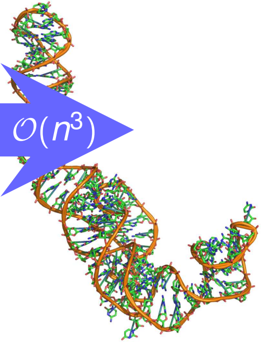
RNA = Linear Polymer = Sequence in $\{A, C, G, U\}^\star$

MFE folding prediction: $\mathcal{O}(n^3)$

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA

CC
```





Primary Structure

Secondary Structure

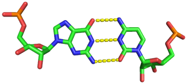
Tertiary Structure

5s rRNA (PDBID: 1K73:B)

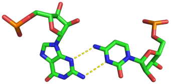
Crossing interactions

Excluded from the secondary structure:

- **Non-canonical base-pairs:**
  Any base-pair **other than** {(A-U), (C-G), (G-U)}
  OR interacting in a non-standard way (WC/WC-Cis) **[Leontis Westhof, RNA 2001]**.

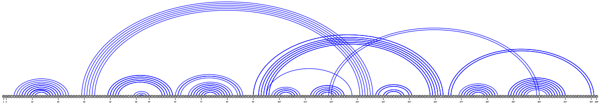


Canonical CG base-pair (WC/WC-Cis)



Non-canonical base-pair (Sugar/WC-Trans)

- **(Pseudo?)knots:** Crossing sets of nested stable base-pairs

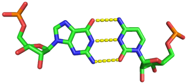


Group I Ribozyme (PDBID: 1Y0Q:A)

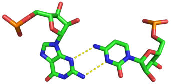
Crossing interactions

Excluded from the secondary structure:

- **Non-canonical base-pairs:**
  Any base-pair **other than** {(A-U), (C-G), (G-U)}
  OR interacting in a non-standard way (WC/WC-Cis) **[Leontis Westhof, RNA 2001]**.

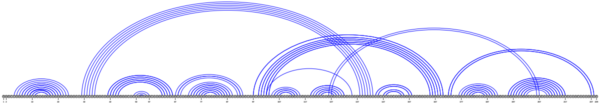


Canonical CG base-pair (WC/WC-Cis)



Non-canonical base-pair (Sugar/WC-Trans)

- **(Pseudo?)knots:** Crossing sets of nested stable base-pairs

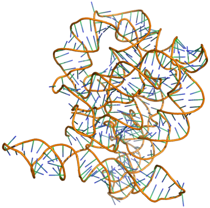


Group I Ribozyme (PDBID: 1Y0Q:A)

Crossing interactions

Excluded from the secondary structure:

- **Non-c**
  Any ba
  OR int **01]**.



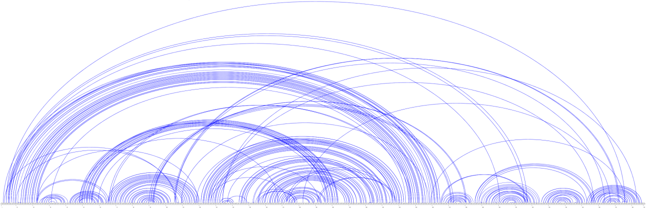
Crossing interactions **do exist**!

Example: Group II Intron (PDB ID: 3IGI)

But are **hard** to predict

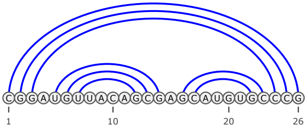
[Lyngsoe-ICALP'04]

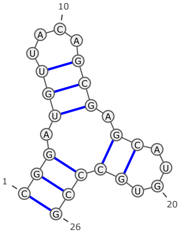
[Sheikh Backofen Ponty, CPM'12]



- **(Pseu**

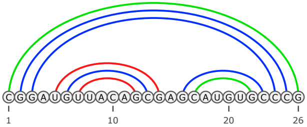
Problem statement

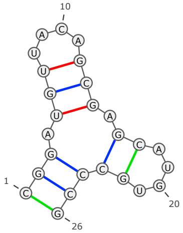




- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacking pairs, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

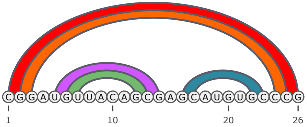
Problem statement

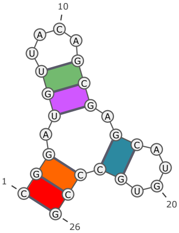




- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacking pairs, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

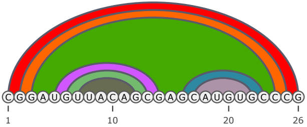
Problem statement

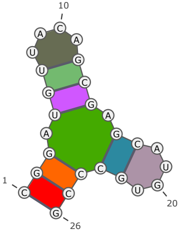




- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacking pairs, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

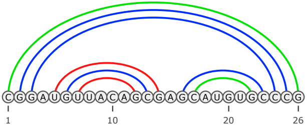
Problem statement

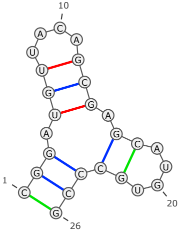




- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacking pairs, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

Problem statement

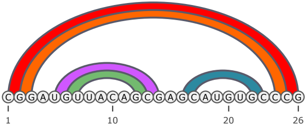


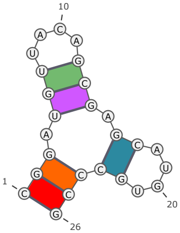


- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacking pairs, Loops. . .)
- **Energy model:**
  Motif $\rightarrow$ Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in *S*

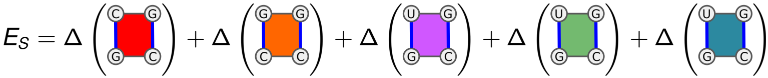
$$E_S = 2 \cdot \Delta \begin{pmatrix} \text{U} \\ | \\ \text{G} \end{pmatrix} + 4 \cdot \Delta \begin{pmatrix} \text{G} \\ | \\ \text{C} \end{pmatrix} + 2 \cdot \Delta \begin{pmatrix} \text{C} \\ | \\ \text{G} \end{pmatrix}$$

Problem statement

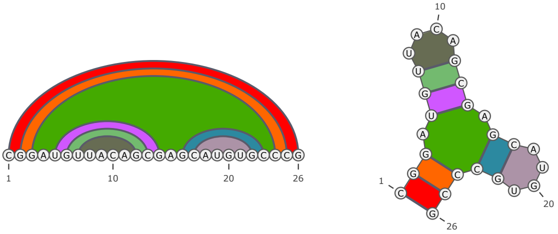




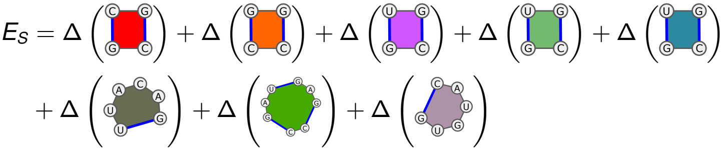
- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacking pairs, Loops. . .)
- **Energy model:**
  Motif → Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S



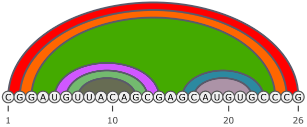
Problem statement

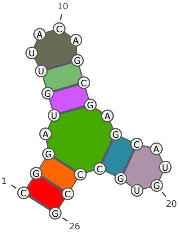


- **RNA structure *S*:** Non-crossing base-pairs for positions in sequence *w*
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacking pairs, Loops. . .)
- **Energy model:**
  Motif $\rightarrow$ Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$
  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in *S*


$E_S = \Delta(\;) + \Delta(\;) + \Delta(\;) + \Delta(\;) + \Delta(\;)$
$+ \Delta(\;) + \Delta(\;) + \Delta(\;)$
C G G C
G G C C
U G G C
U G G C
U G G C
A C A U G U
U G A G C C G A
C A U G U G

Problem statement





- **RNA structure** S**:** Non-crossing base-pairs for positions in sequence w
- **Motifs:** Sequence/structure features (e.g. Base-pairs, Stacking pairs, Loops. . .)
- **Energy model:**
  Motif $\rightarrow$ Free-energy contribution $\Delta(\cdot) \in \mathbb{R}^{-} \cup \{+\infty\}$
  Free-Energy $E_w(S)$**:** Sum over (independently contributing) motifs in S

Definition (MFE-PREDICT(E) **problem)**

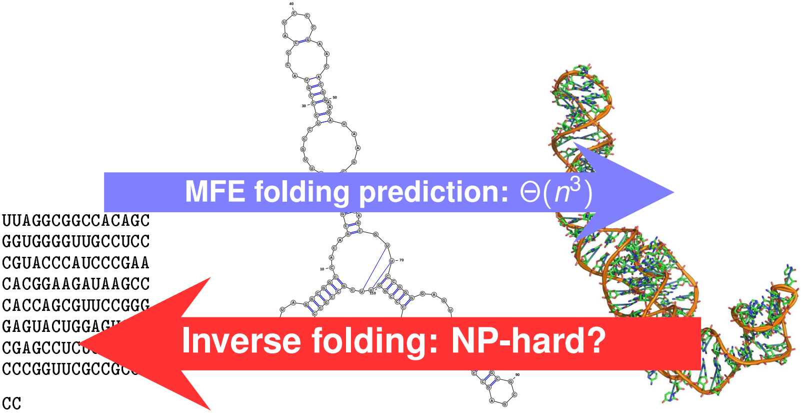
Input: RNA sequence $w \in \{A, C, G, U\}^*$
Output: Secondary struct. S^* with Minimal Free-Energy (MFE) $E_w(S^*)$

Problem solved **exactly** in $O(n^3)$ time.

[Nussinov Jacobson, PNAS 1980] [Zuker Stiegler, NAR 1981]. . . .

RNA inverse folding

RNA = Linear Polymer = Sequence in $\{A, C, G, U\}^\star$



UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAG
CGAGCCUC
CCCGGUUCGCCGC

CC

MFE folding prediction: $\Theta(n^3)$

Inverse folding: NP-hard?

Primary Structure Secondary Structure Structure Tertiaire

5s rRNA (PDBID: 1K73:B)

RNA Inverse Folding

Definition (INVERSE-FOLDING(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} : \; E_{w,S'} \geq E w, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Difficult problem: No **obvious** DP decomposition

- Existing algorithms: Heuristics or Exponential-time
- Complexity of problem unknown (despite **[Schnall Levin *et al*, ICML'08]**)
  Reason: Non locality, no theoretical frameworks, too many parameters. . .

RNA Inverse Folding

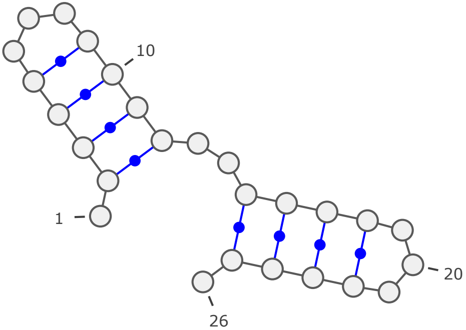
Definition (INVERSE-FOLDING(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} :\ E_{w,S'} \geq E w, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Example:



RNA Inverse Folding

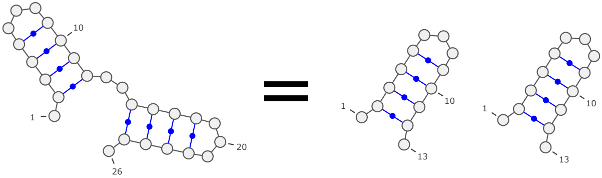
Definition (INVERSE-FOLDING(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} : \; E_{w,S'} \geq E w, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Example:



RNA Inverse Folding

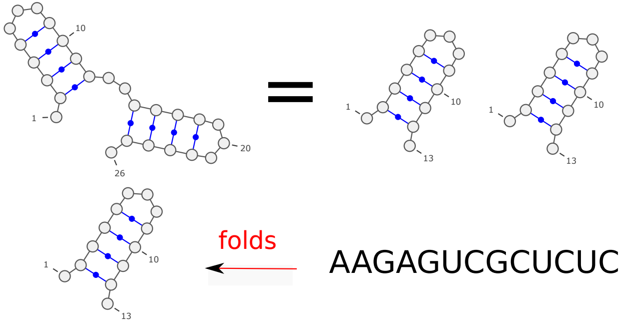
Definition (INVERSE-FOLDING(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

$$\forall S' \in \mathcal{S}|w| \setminus \{S\} :\ E_{w,S'} \geq Ew, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Example:



folds

AAGAGUCGCUCUC

RNA Inverse Folding

Definition (INVERSE-FOLDING(E) problem)

Input: Secondary structure S + Energy distance $\Delta > 0$.
Output: RNA sequence $w \in \Sigma^\star$ such that:

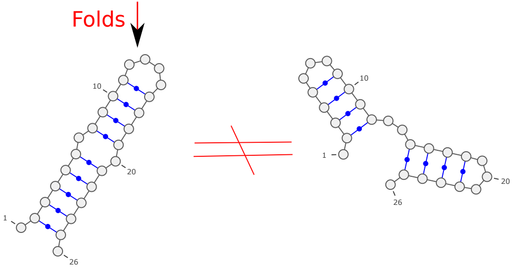
$$\forall S' \in \mathcal{S}|w| \setminus \{S\} : \; E_{w,S'} \geq Ew, S + \Delta$$

or $\varnothing$ if no such sequence exists.

Example:

AAGAGUCGCUCUCAAGAGUCGCUCUC

Folds



Existing approaches for negative design

Based on local search...

- RNAInverse - TBI Vienna
- Info-RNA - Backofen@Freiburg
- RNA-SSD - Condon@UBC
- NUPack - Pierce@Caltech

...bio-inspired algorithms...

- RNAFBinv - Barash@Ben Gurion
- FRNAKenstein - Hein@Oxford
- AntaRNA - Backofen@Freiburg

...exact approaches...

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

Typical issues:

- Naive initialization strategies
- Poor coverage of sequence space:
  Local search remain *confined* near initial sequence
- Drift towards GC-rich regions of sequence space

⇒ **Global sampling [Levin *et al*, NAR 12]**

Global sampling [Levin *et al*, NAR 12]

Target structure S

- **Boltzmann distribution** based on **affinity** towards S
- **Random generation** from **Boltzmann Distribution**
- **Fold** sampled sequences and **compare** to target

Boltzmann factor:

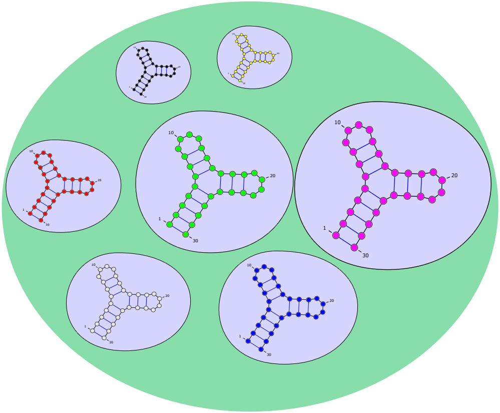
$$\mathcal{B}_w(S) := e^{\frac{-E_w(S)}{RT}}$$

Pseudo-Partition Function:

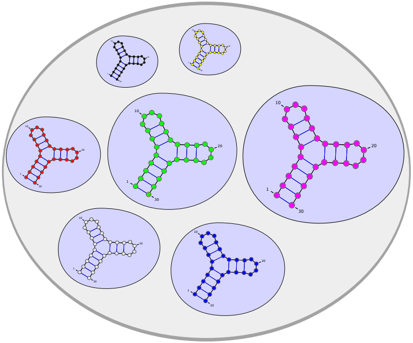
$$\mathcal{Z}(S) = \sum_{w\in\Sigma^*} \mathcal{B}_w(S)$$

Boltzmann probability:

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



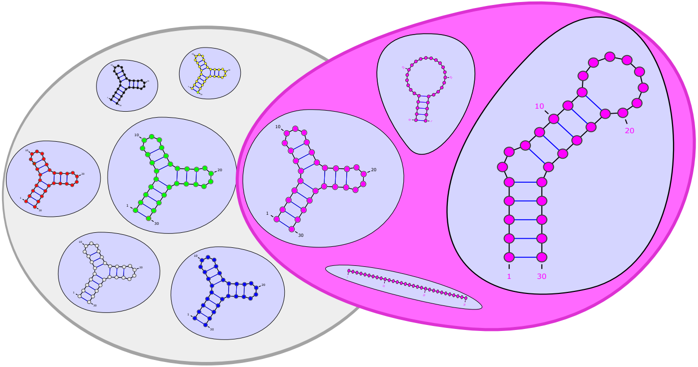
Limits of the approach



Heuristic: Strong affinity is neither sufficient, nor necessary for design, **but** . . .

- Strong **empirical** correlation affinity/success of design **[Levin et al, NAR 2012]**
- **Linear** time-complexity **[Reinharz Ponty Waldispühl, ISMB/ECCB'13]**
- **Composition** control **[Bodini Ponty, AofA'10] [Reinharz et al, ISMB/ECCB'13]**
- **Complementary** with local search approaches **[Reinharz et al, ISMB/ECCB'13]**

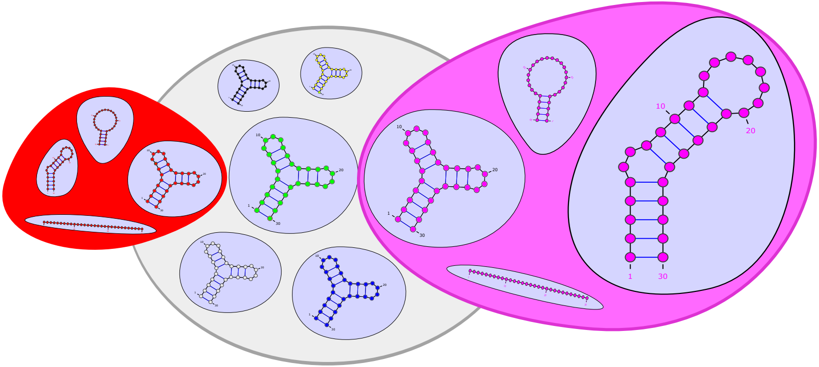
Limits of the approach



Heuristic: Strong affinity is **neither sufficient,** nor necessary for design, **but** . . .

- Strong **empirical** correlation affinity/success of design **[Levin et al, NAR 2012]**
- **Linear** time-complexity **[Reinharz Ponty Waldispühl, ISMB/ECCB'13]**
- **Composition** control **[Bodini Ponty, AofA'10] [Reinharz et al, ISMB/ECCB'13]**
- **Complementary** with local search approaches **[Reinharz et al, ISMB/ECCB'13]**

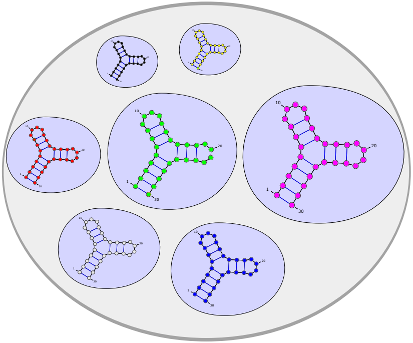
Limits of the approach



Heuristic: Strong affinity is **neither sufficient, nor necessary** for design, **but** . . .

- Strong **empirical** correlation affinity/success of design **[Levin et al, NAR 2012]**
- **Linear** time-complexity **[Reinharz Ponty Waldispühl, ISMB/ECCB'13]**
- **Composition** control **[Bodini Ponty, AofA'10] [Reinharz et al, ISMB/ECCB'13]**
- **Complementary** with local search approaches **[Reinharz et al, ISMB/ECCB'13]**

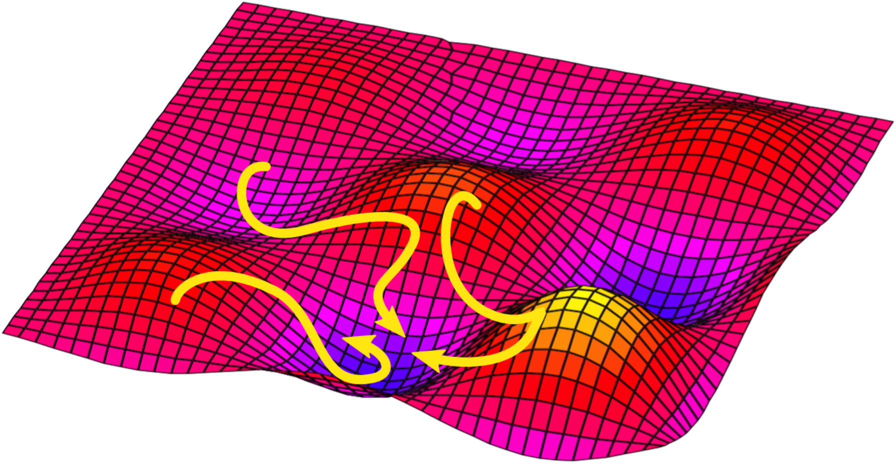
Limits of the approach



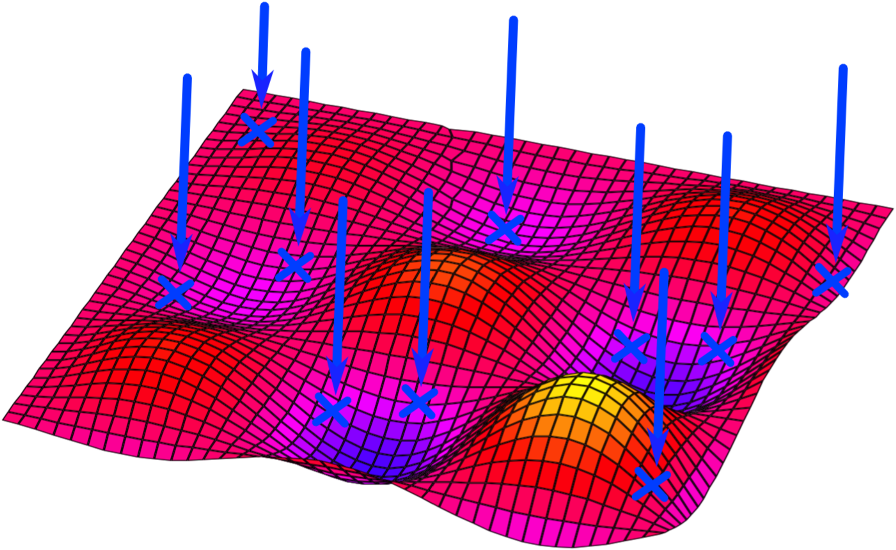
Heuristic: Strong affinity is **neither sufficient, nor necessary** for design, **but** . . .

- Strong **empirical** correlation affinity/success of design **[Levin et al, NAR 2012]**
- **Linear** time-complexity **[Reinharz Ponty Waldispühl, ISMB/ECCB'13]**
- **Composition** control **[Bodini Ponty, AofA'10] [Reinharz et al, ISMB/ECCB'13]**
- **Complementary** with local search approaches **[Reinharz et al, ISMB/ECCB'13]**

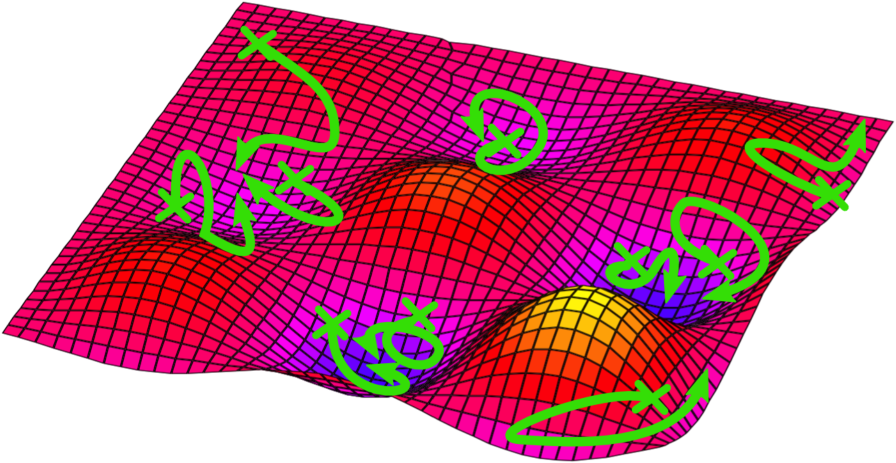
Local vs Global vs "Glocal"



Local vs Global vs "Glocal"



Local vs Global vs "Glocal"



II. Constrained design

Avoiding/forcing motifs

Existing approaches for negative design

Based on local search…

- RNAInverse - TBI Vienna
- Info-RNA - Backofen@Freiburg
- RNA-SSD - Condon@UBC
- NUPack - Pierce@Caltech

…bio-inspired algorithms…

- RNAFBinv - Barash@Ben Gurion
- FRNAKenstein - Hein@Oxford
- AntaRNA - Backofen@Freiburg

…exact approaches…

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

Few algorithms support avoided/mandatory motifs…

…none guarantees *reasonable* runtime.

Typical reasons:

- Deep local minima (Rugged landscape)
- Mandatory motifs ⇒ Late deadends (Branch and Bound)
- Forbidden motifs ⇒ Search space disconnection (Local Search)

Existing approaches for negative design

Based on local search…

- RNAInverse - TBI Vienna
- Info-RNA - Backofen@Freiburg
- RNA-SSD - Condon@UBC
- NUPack - Pierce@Caltech

…bio-inspired algorithms…

- RNAFBinv - Barash@Ben Gurion
- FRNAKenstein - Hein@Oxford
- AntaRNA - Backofen@Freiburg

…exact approaches…

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

Few algorithms support avoided/mandatory motifs…

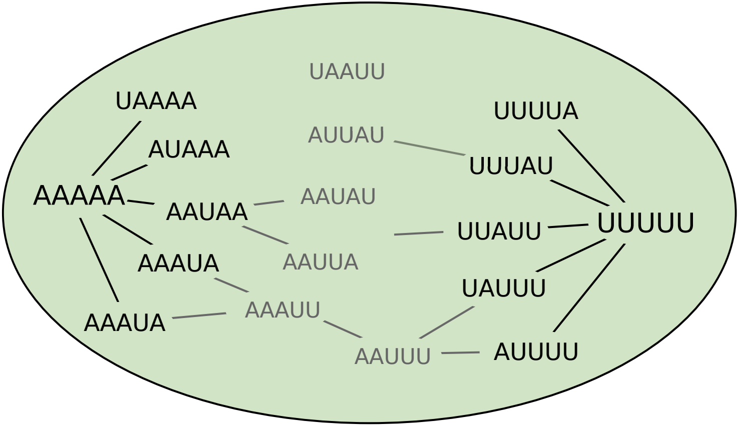
…none guarantees *reasonable* runtime.

Typical reasons:

- Deep local minima (Rugged landscape)
- Mandatory motifs ⇒ Late deadends (Branch and Bound)
- Forbidden motifs ⇒ Search space disconnection (Local Search)

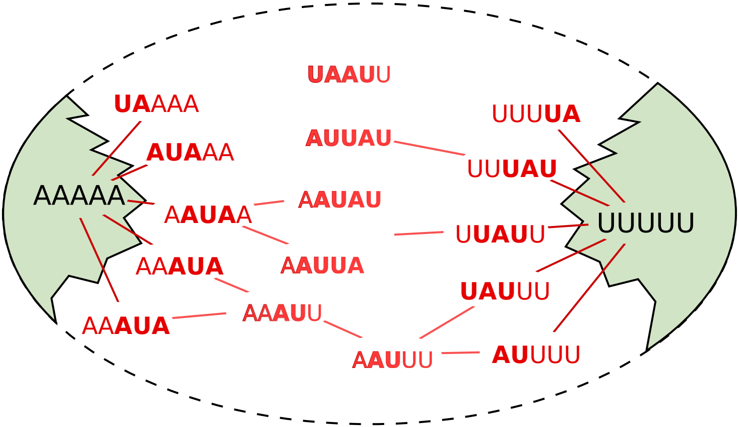
Problem with local approaches: An example

Simplified vocabulary $\{A, U\}$



Problem with local approaches: An example

Simplified vocabulary $\{A, U\}$ **+** **Forbidden motifs** $\mathcal{F} = \{AU, UA\}$



⇒ $\mathcal{F}$ may **disconnect** search space (holds for **any** move set!)

Idea

Use formal language constructs to constrain global sampling

Forced motifs
Avoided motifs $\rightarrow$ Regular language $\mathcal{L}_C \in$ Reg

Structure compatibility
+ Positional constraints $\rightarrow$ **Weighted** Context-Free Lang $\mathcal{L}_S \in$ CFL
+ Energy Model

Folklore theorem (constructive): Reg $\cap$ (**W**)CFL $\subseteq$ (**W**)CFL

Build weighted context-free grammar $\mathcal{G}$ for $\mathcal{L}_C \cap \mathcal{L}_S$
+ Random generation

$\Rightarrow$ **Global sampling under constraints**

Idea

Use formal language constructs to constrain global sampling

Forced motifs
Avoided motifs → Regular language $\mathcal{L}_C \in$ Reg

Structure compatibility
+ Positional constraints → **Weighted** Context-Free Lang $\mathcal{L}_S \in$ CFL
+ Energy Model

Folklore theorem (constructive): Reg $\cap$ (**W**)CFL $\subseteq$ (**W**)CFL

Build weighted context-free grammar $\mathcal{G}$ for $\mathcal{L}_C \cap \mathcal{L}_S$
+ Random generation

$\Rightarrow$ **Global sampling under constraints**

Idea

Use formal language constructs to constrain global sampling

Forced motifs
Avoided motifs
$\rightarrow$ Regular language $\mathcal{L}_C \in$ Reg

Structure compatibility
+ Positional constraints
+ Energy Model
$\rightarrow$ **Weighted** Context-Free Lang $\mathcal{L}_S \in$ CFL

Folklore theorem (constructive): Reg $\cap$ (**W**)CFL $\subseteq$ (**W**)CFL

Build weighted context-free grammar $\mathcal{G}$ for $\mathcal{L}_C \cap \mathcal{L}_S$
+ Random generation

$\Rightarrow$ **Global sampling under constraints**

Idea

Use formal language constructs to constrain global sampling

Forced motifs
Avoided motifs → Regular language $\mathcal{L}_C \in$ Reg

Structure compatibility
+ Positional constraints → **Weighted** Context-Free Lang $\mathcal{L}_S \in$ CFL
+ Energy Model

Folklore theorem (constructive): Reg $\cap$ (**W**)CFL $\subseteq$ (**W**)CFL

Build weighted context-free grammar $\mathcal{G}$ for $\mathcal{L}_C \cap \mathcal{L}_S$
+ Random generation

⇒ Global sampling under constraints

Building the Finite State Automaton

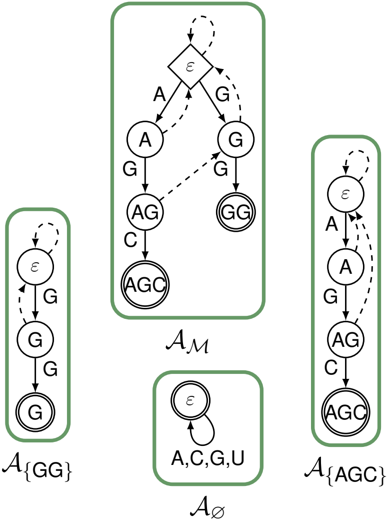
To force multiple words, **keep track** of generated words:

- Create disjunctive automaton for each $\mathcal{M}' \subseteq \mathcal{M}$
- **Reroute** accepting states
- Accepting state = no **forced word** remaining (ε in $\mathcal{A}_\varnothing$)
- Forbidden words can be added to sub-automata

#States:

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

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



ε
A
G
A
G
G
G
AG
GG
C
AGC
$\mathcal{A}_{\mathcal{M}}$
ε
G
G
G
G
$\mathcal{A}_{\{\mathsf{GG}\}}$
ε
A,C,G,U
$\mathcal{A}_\varnothing$
ε
A
A
G
AG
C
AGC
$\mathcal{A}_{\{\mathsf{AGC}\}}$

Building the Finite State Automaton

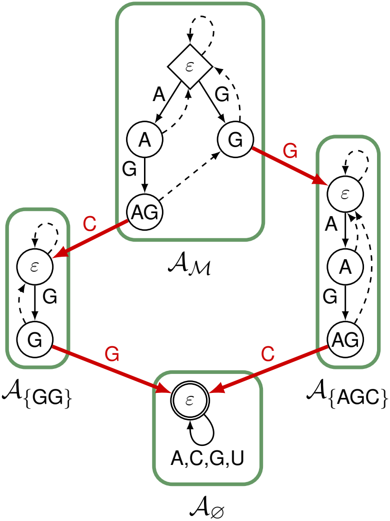
To force multiple words, **keep track** of generated words:

- Create disjunctive automaton for each $\mathcal{M}' \subseteq \mathcal{M}$
- **Reroute** accepting states
- Accepting state = no **forced word** remaining (ε in $\mathcal{A}_\varnothing$)
- Forbidden words can be added to sub-automata

#States:

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

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



Building the Finite State Automaton

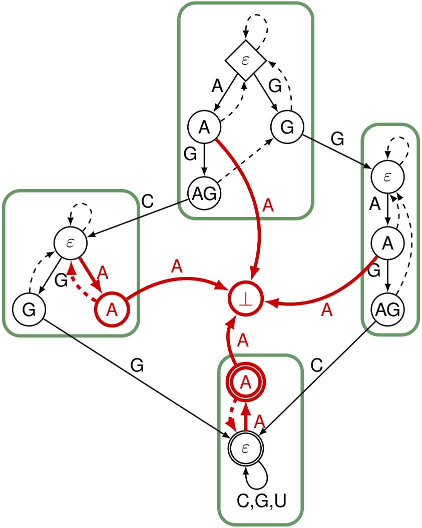
To force multiple words, **keep track** of generated words:

- Create disjunctive automaton for each $\mathcal{M}' \subseteq \mathcal{M}$
- **Reroute** accepting states
- Accepting state = no **forced word** remaining (ε in $\mathcal{A}_\varnothing$)
- Forbidden words can be added to sub-automata

#States:

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

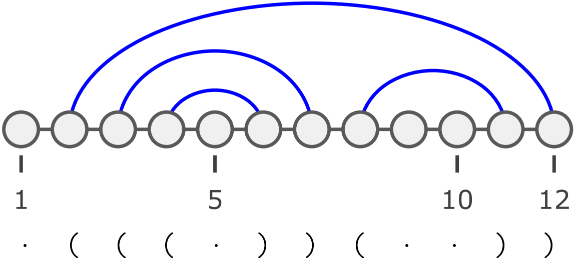
Example: $\mathcal{M} = \{\text{AGC}, \text{GG}\}; \mathcal{F} = \{\text{AA}\}$



Building the grammar

Input: Secondary Structure *S* + Positional constraints

- **A Create Parse Tree** for secondary structure
- **B Translate Parse Tree** into single-word grammar
- **C Expand** grammar to instantiate compatible base/base-pairs
- **D Restrict** to bases/base-pairs allowed at each position



Building the grammar

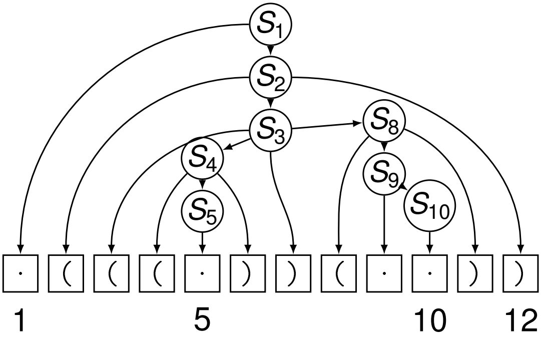
Input: Secondary Structure S + Positional constraints

A Create Parse Tree for secondary structure

B Translate Parse Tree into single-word grammar

C Expand grammar to instantiate compatible base/base-pairs

D Restrict to bases/base-pairs allowed at each position



Building the grammar

Input: Secondary Structure *S* + Positional constraints

- **A Create Parse Tree** for secondary structure
- **B Translate Parse Tree** into single-word grammar
- **C Expand** grammar to instantiate compatible base/base-pairs
- **D Restrict** to bases/base-pairs allowed at each position

$$\begin{array}{llll} S_1 \to .\,S_2 & S_2 \to (\,S_3\,) & S_3 \to (\,S_4\,)\,S_8 & S_4 \to (\,S_5\,) \\ S_5 \to . & S_8 \to (\,S_9\,) & S_9 \to .\,S_{10} & S_{10} \to . \end{array}$$

Building the grammar

Input: Secondary Structure S + Positional constraints

A **Create Parse Tree** for secondary structure

B **Translate Parse Tree** into single-word grammar

C **Expand** grammar to instantiate compatible base/base-pairs

D **Restrict** to bases/base-pairs allowed at each position

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

Building the grammar

Input: Secondary Structure *S* **+ Positional constraints**

A **Create Parse Tree** for secondary structure

B **Translate Parse Tree** into single-word grammar

C **Expand** grammar to instantiate compatible base/base-pairs

D **Restrict** to bases/base-pairs allowed at each position

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

Random generation

Combine CFG and automaton → CFG (Multiplying $\#$Rules by $|Q|^3$)

GenRGenS [Ponty Termier Denise, Bioinformatics 2006]:

- Precomputes #words for each non-terminal
- Random Generation w.r.t. **weighted distribution**

Energy models:

- **Uniform distribution**
- **Nussinov energy model**
- **Stacking-pairs model (Turner 2004)**
  Based on refined, yet similar, grammar

Overall complexity: $|S| \cdot 2^{3|\mathcal{M}|} \cdot \left(\sum_i |f_i| + \sum_j |m_j|\right)^3$

- **Linear** on $|S|$
- **Exponential** on $|\mathcal{M}|$, but **NP-Hard** problem

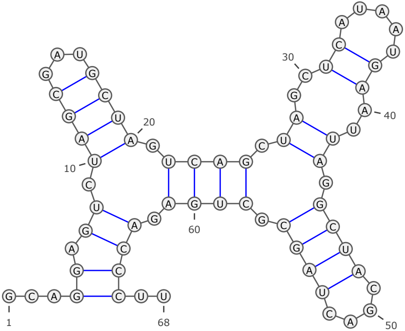
II. Combinatorial design

A minimal installment of negative design

[Haleš Maňuch Ponty Stacho, CPM'15]

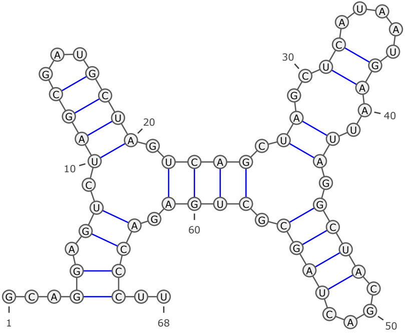
Representations of Secondary Structures

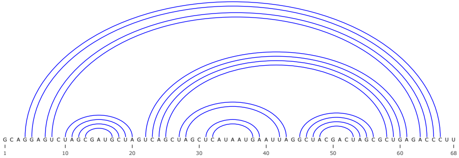
Structure = Bunch of **non-crossing** base-pairs.


G C A G G A G U C U A G C G A U G C U A G U C A G C U A G C U C A U A A U G A A U U A G G C U A C G A C U A G C G C U G A G A C C C U U
1
10
20
30
40
50
60
68

Representations of Secondary Structures

Structure = Bunch of **non-crossing** base-pairs.

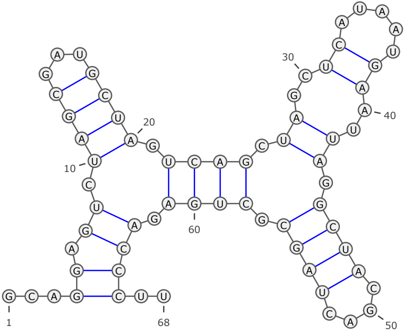


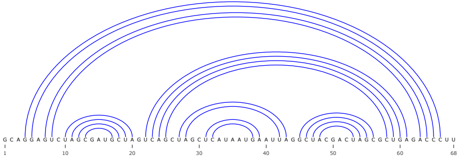

G C A G G A G U C U A G C G A U G C U A G U C A G C U A G C U C A U A A U G A A U U A G G C U A C G A C U A G C G C U G A G A C C C U U
1
10
20
30
40
50
60
68

arc diagram

Representations of Secondary Structures

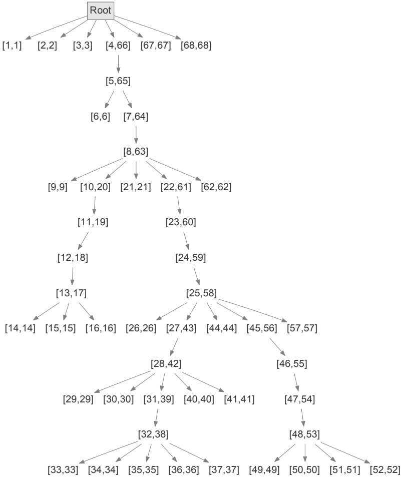
Structure = Bunch of **non-crossing** base-pairs.





G C A G G A G U C U A G C G A U G C U A G U C A G C U A G C U C A U A A U G A A U U A G G C U A C G A C U A G C G C U G A G A C C C U U
1
10
20
30
40
50
60
68

arc diagram



Root

[1,1] [2,2] [3,3] [4,66] [67,67] [68,68]

[5,65]

[6,6] [7,64]

[8,63]

[9,9] [10,20] [21,21] [22,61] [62,62]

[11,19] [23,60]

[12,18] [24,59]

[13,17] [25,58]

[14,14] [15,15] [16,16] [26,26] [27,43] [44,44] [45,56] [57,57]

[28,42] [46,55]

[29,29] [30,30] [31,39] [40,40] [41,41] [47,54]

[32,38] [48,53]

[33,33] [34,34] [35,35] [36,36] [37,37] [49,49] [50,50] [51,51] [52,52]

tree representation

RNA Design Problem

Let $\mathcal{M}$ be an energy model.

Problem (INVERSE-FOLDING($\mathcal{M}, \Sigma, \Delta$) problem)

Input: *Secondary structure S + Energy distance $\Delta > 0$*
Output: *RNA sequence $w \in \Sigma^\star$ — called a design for S — such that:*

$$\forall S' \in \mathcal{S}_{|w|} \setminus \{S\} : \; E_{\mathcal{M}}(w, S') \geq E_{\mathcal{M}}(w, S) + \Delta$$

or $\varnothing$ if no such sequence exists.

Difficult problem: No obvious DP decomposition

- Existing algorithms: Heuristics or Exponential-time
- Complexity of problem unknown (despite [Schnall Levin et al (2008)])
  Reason: Non locality, no theoretical frameworks, too many parameters. . .

⇒ **Stick to a simplified model!**

RNA Design Problem (simplified)

Simplified formulation for Watson-Crick model $\mathcal{W}$ and $\Delta = 1$:

Problem (INVERSE-FOLDING(Σ) problem)

Input: *Secondary structure S*
Output: *RNA sequence $w \in \Sigma^\star$ — called a design for S — such that:*

$$\text{RNA-FOLD}_{\mathcal{W}}(w) = \{S\}$$

or $\varnothing$ if no such sequence exists.

Designable(Σ)**:** All designable structures

RNA Design Problem (simplified)

Simplified formulation for Watson-Crick model $\mathcal{W}$ and $\Delta = 1$:

Problem (INVERSE-FOLDING(Σ) problem)

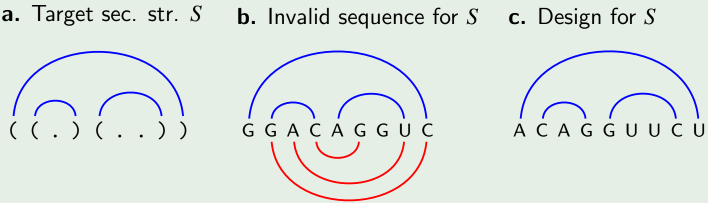
Input: *Secondary structure S*
Output: *RNA sequence $w \in \Sigma^\star$ — called a design for S — such that:*

$$\text{RNA-FOLD}_{\mathcal{W}}(w) = \{S\}$$

or $\varnothing$ if no such sequence exists.

Designable(Σ): All designable structures

Example

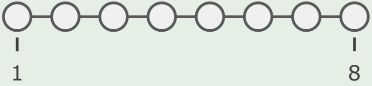


Our Results: Definitions and notations

Given a secondary structure S:

- Unpaired$_S$ = Set of all unpaired positions of S.
- S is **saturated** $\Leftrightarrow$ Unpaired$_S = \varnothing$.
  Saturated = Set of all saturated structures.
- **Paired degree of base-pair** = #Helices on the loop.
- $D(S)$ = Maximal *paired degree* of nodes in the tree representation of S.

Example



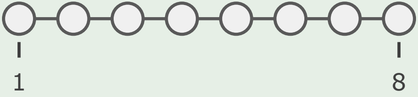
Unpaired$_S = \{4, 8\}$

Our Results: Definitions and notations

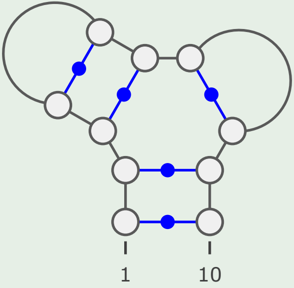
Given a secondary structure S:

- Unpaired$_S$ = Set of all unpaired positions of S.
- S is **saturated** $\Leftrightarrow$ Unpaired$_S = \varnothing$.
  Saturated = Set of all saturated structures.
- **Paired degree of base-pair** = #Helices on the loop.
- $D(S)$ = Maximal *paired degree* of nodes in the tree representation of S.

Example



Unsaturated



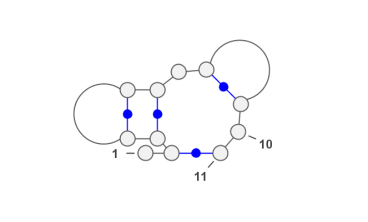
Saturated

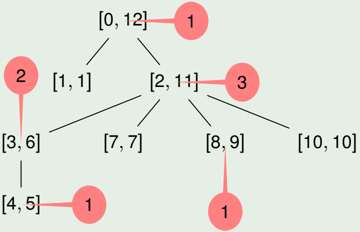
Our Results: Definitions and notations

Given a secondary structure S:

- $\text{Unpaired}_S =$ Set of all unpaired positions of S.
- S is **saturated** $\Leftrightarrow \text{Unpaired}_S = \varnothing$.
  Saturated $=$ Set of all saturated structures.
- **Paired degree of base-pair** $=$ #Helices on the loop.
- $D(S) =$ Maximal *paired degree* of nodes in the tree representation of S.

Example





$D(S) = 3$

Our Results: Designability over Restricted Alphabets

$\Sigma_{c,u} =$ Alphabet with c pairs of complementary bases and u unpairable bases.

R1 $\Sigma_{0,u} \Rightarrow$ Designable $=$ Empty (single-stranded) structures;

R2 $\Sigma_{1,0} \Rightarrow$ Designable $=$ Saturated with degree$\leq$ 2 + empty structures ;

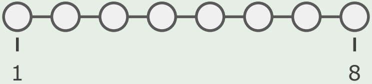
R3 $\Sigma_{1,1} \Rightarrow$ Designable $=$ Degree$\leq$ 2.

Our Results: Designability over Restricted Alphabets

$\Sigma_{c,u} =$ Alphabet with c pairs of complementary bases and u unpairable bases.

- **R1** $\Sigma_{0,u} \Rightarrow$ Designable $=$ Empty (single-stranded) structures;
- **R2** $\Sigma_{1,0} \Rightarrow$ Designable $=$ Saturated with degree$\leq$ 2 + empty structures ;
- **R3** $\Sigma_{1,1} \Rightarrow$ Designable $=$ Degree$\leq$ 2.

Example



Our Results: Designability over Restricted Alphabets

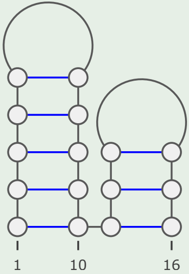
$\Sigma_{c,u} =$ Alphabet with c pairs of complementary bases and u unpairable bases.

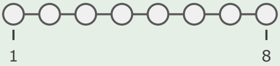
R1 $\Sigma_{0,u} \Rightarrow$ Designable $=$ Empty (single-stranded) structures;

R2 $\Sigma_{1,0} \Rightarrow$ Designable $=$ Saturated with degree$\leq$ 2 + empty structures ;

R3 $\Sigma_{1,1} \Rightarrow$ Designable $=$ Degree$\leq$ 2.

Example





Our Results: Designability over Restricted Alphabets

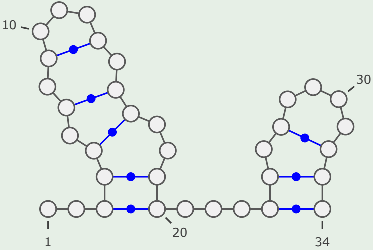
$\Sigma_{c,u} =$ Alphabet with c pairs of complementary bases and u unpairable bases.

R1 $\Sigma_{0,u} \Rightarrow$ Designable $=$ Empty (single-stranded) structures;

R2 $\Sigma_{1,0} \Rightarrow$ Designable $=$ Saturated with degree≤ 2 + empty structures ;

R3 $\Sigma_{1,1} \Rightarrow$ Designable $=$ Degree≤ 2.

Example



+ miRNAs, some lncRNAs. . .

Our Results: Designability over Restricted Alphabets

$\Sigma_{c,u} =$ Alphabet with c pairs of complementary bases and u unpairable bases.

R1 $\Sigma_{0,u} \Rightarrow$ Designable $=$ Empty (single-stranded) structures;

R2 $\Sigma_{1,0} \Rightarrow$ Designable $=$ Saturated with degree$\leq$ 2 + empty structures ;

R3 $\Sigma_{1,1} \Rightarrow$ Designable $=$ Degree$\leq$ 2.

Question: Why not degree 3?

Proof.

□

Our Results: Designability over Restricted Alphabets

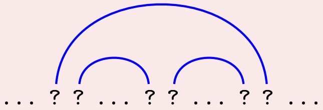
$\Sigma_{c,u}$ = Alphabet with *c* pairs of complementary bases and *u* unpairable bases.

- **R1** $\Sigma_{0,u} \Rightarrow$ Designable = Empty (single-stranded) structures;
- **R2** $\Sigma_{1,0} \Rightarrow$ Designable = Saturated with degree$\leq$ 2 + empty structures ;
- **R3** $\Sigma_{1,1} \Rightarrow$ Designable = Degree$\leq$ 2.

Question: Why not degree 3?

Proof.

Within an internal node:



□

Our Results: Designability over Restricted Alphabets

$\Sigma_{c,u}$ = Alphabet with c pairs of complementary bases and u unpairable bases.

R1 $\Sigma_{0,u} \Rightarrow$ Designable = Empty (single-stranded) structures;

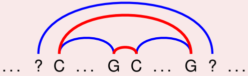
R2 $\Sigma_{1,0} \Rightarrow$ Designable = Saturated with degree≤ 2 + empty structures ;

R3 $\Sigma_{1,1} \Rightarrow$ Designable = Degree≤ 2.

Question: Why not degree 3?

Proof.

Within an internal node:



Either we get a repeat...

□

Our Results: Designability over Restricted Alphabets

$\Sigma_{c,u}$ = Alphabet with c pairs of complementary bases and u unpairable bases.

R1 $\Sigma_{0,u} \Rightarrow$ Designable = Empty (single-stranded) structures;

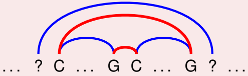
R2 $\Sigma_{1,0} \Rightarrow$ Designable = Saturated with degree≤ 2 + empty structures ;

R3 $\Sigma_{1,1} \Rightarrow$ Designable = Degree≤ 2.

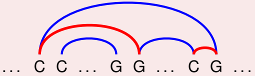
Question: Why not degree 3?

Proof.

Within an internal node:



Either we get a repeat. . .



. . . or some parent/child have complementary pairs.

+ Same principle at the root level. □

Our Results: Designability over Restricted Alphabets

$\Sigma_{c,u} =$ Alphabet with c pairs of complementary bases and u unpairable bases.

R1 $\Sigma_{0,u} \Rightarrow$ Designable $=$ Empty (single-stranded) structures;

R2 $\Sigma_{1,0} \Rightarrow$ Designable $=$ Saturated with degree$\leq$ 2 + empty structures ;

R3 $\Sigma_{1,1} \Rightarrow$ Designable $=$ Degree$\leq$ 2.

This can be easily generalized to:

Lemma

For any structure S in Designable($\Sigma_{c,u}$)*, $D(S) \leq 2c$.*

Our Results: Designability over the Complete Alphabet

$\Sigma_{2,0} = \{A, U, C, G\}$ + $\{G - C, A - U\}$ base pairs.

Without unpaired position → complete characterization:

R4 $\Sigma_{2,0} \Rightarrow$ Saturated Designable = Degree≤ 4.

With unpaired positions → partial characterization:

R5 (Necessary) Designable structure cannot contain "*a multiloop of degree* ≥ 5" (motif m_5) or "*a multiloop with unpaired position of degree* ≥ 3" (motif $m_{3\circ}$).

R6 (Sufficient) Separated = Structure that admit a separated (proper) coloring. Then any Separated **structure is Designable in** $\Sigma_{2,0}$.

Our Results: Designability over the Complete Alphabet

$\Sigma_{2,0} = \{\text{A}, \text{U}, \text{C}, \text{G}\}$ + $\{\text{G} - \text{C}, \text{A} - \text{U}\}$ base pairs.

Without unpaired position → complete characterization:

R4 $\Sigma_{2,0} \Rightarrow$ Saturated Designable $=$ Degree≤ 4.

With unpaired positions → partial characterization:

R5 (Necessary) Designable structure cannot contain "*a multiloop of degree* ≥ 5" (motif m_5) or "*a multiloop with unpaired position of degree* ≥ 3" (motif $m_{3\circ}$).

R6 (Sufficient) Separated $=$ Structure that admit a separated (proper) coloring. Then any Separated **structure is Designable in** $\Sigma_{2,0}$.

Our Results: Designability over the Complete Alphabet

$\Sigma_{2,0} = \{A, U, C, G\}$ + $\{G - C, A - U\}$ base pairs.

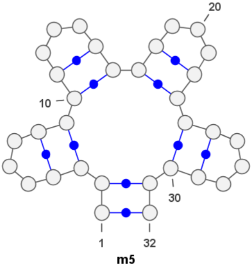
Without unpaired position → complete characterization:

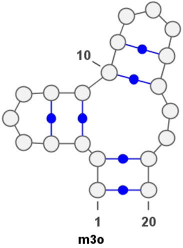
R4 $\Sigma_{2,0} \Rightarrow$ Saturated Designable = Degree$\leq$ 4.

With unpaired positions → partial characterization:

R5 (Necessary) Designable structure cannot contain "*a multiloop of degree* $\geq$ 5" (motif m_5) or "*a multiloop with unpaired position of degree* $\geq$ 3" (motif $m_{3\circ}$).

R6 (Sufficient) Separated = Structure that admit a separated (proper) coloring. Then any Separated **structure is Designable in** $\Sigma_{2,0}$.





Our Results: Designability over the Complete Alphabet

$\Sigma_{2,0} = \{\mathsf{A}, \mathsf{U}, \mathsf{C}, \mathsf{G}\}$ + $\{\mathsf{G} - \mathsf{C}, \mathsf{A} - \mathsf{U}\}$ base pairs.

Without unpaired position → complete characterization:

R4 $\Sigma_{2,0} \Rightarrow$ Saturated Designable $=$ Degree$\leq$ 4.

With unpaired positions → partial characterization:

R5 (Necessary) Designable structure cannot contain "*a multiloop of degree* $\geq$ 5" (motif m_5) or "*a multiloop with unpaired position of degree* $\geq$ 3" (motif $m_{3\,\circ}$).

R6 (Sufficient) Separated $=$ Structure that admit a separated (proper) coloring. Then any Separated **structure is Designable in** $\Sigma_{2,0}$.

Our Results: Separated Coloring

From the tree representation T_S of structure S, color every paired node of T_S:

- black → $G \cdot C$;
- white → $C \cdot G$;
- grey → $A \cdot U$ or $U \cdot A$.

Proper coloring:

1. each internal node has at most one black, one white and two grey children;
2. a grey node has at most one grey child;
3. a black node does not have a white child; and
4. a white node does not have a black child.

Level of a node = #black nodes − #white nodes on the path to the root.

Separated coloring: Levels of grey nodes $\cap$ Levels of unpaired nodes $= \varnothing$

Our Results: Separated Coloring

From the tree representation T_S of structure S, color every paired node of T_S:

- black → $G \cdot C$;
- white → $C \cdot G$;
- grey → $A \cdot U$ or $U \cdot A$.

Proper coloring:

1. each internal node has at most one black, one white and two grey children;
2. a grey node has at most one grey child;
3. a black node does not have a white child; and
4. a white node does not have a black child.

Level of a node = #black nodes − #white nodes on the path to the root.

Separated coloring: Levels of grey nodes $\cap$ Levels of unpaired nodes $= \varnothing$

Our Results: Separated Coloring

From the tree representation T_S of structure S, color every paired node of T_S:

- black $\rightarrow$ G $\cdot$ C;
- white $\rightarrow$ C $\cdot$ G;
- grey $\rightarrow$ A $\cdot$ U or U $\cdot$ A.

Proper coloring:

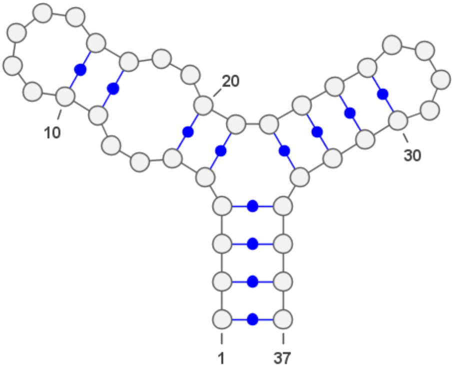
1. each internal node has at most one black, one white and two grey children;
2. a grey node has at most one grey child;
3. a black node does not have a white child; and
4. a white node does not have a black child.

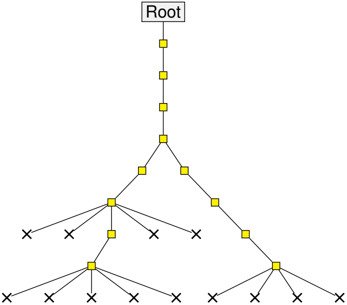
Level of a node = #black nodes $-$ #white nodes on the path to the root.

Separated coloring: Levels of grey nodes $\cap$ Levels of unpaired nodes $= \varnothing$

Our Results: Separated Coloring (example)

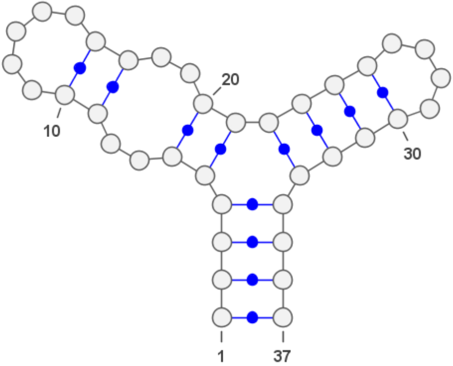
Descendant restrictions: Any node → $\leq$ 1 black & $\leq$ 1 White & $\leq$ 2 Grey;
Grey → 0/1 Grey; Black → 0 White; White → 0 Black.
(● → GC ○ → CG ◉ → AU|UA × → U)

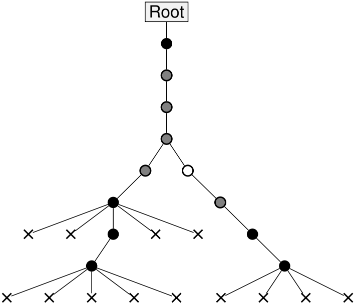




Our Results: Separated Coloring (example)

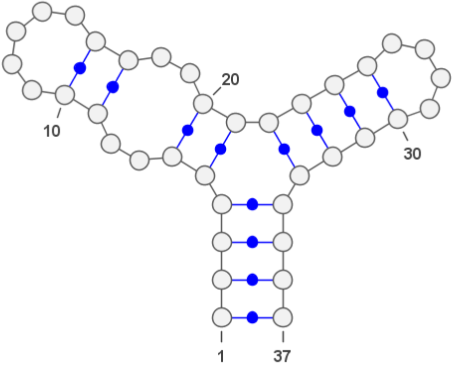
Descendant restrictions: Any node → ≤ 1 black & ≤ 1 White & ≤ 2 Grey;
Grey → 0/1 Grey; Black → 0 White; White → 0 Black.
(● → GC ○ → CG ◍ → AU|UA × → U)

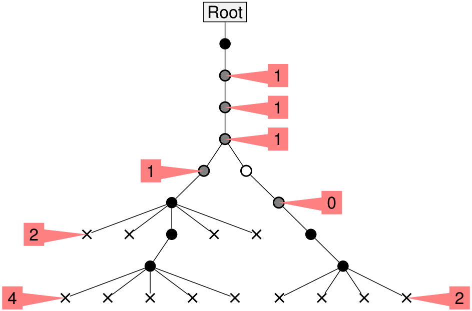




Our Results: Separated Coloring (example)

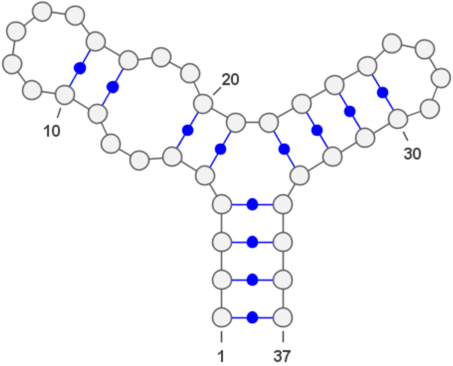
Descendant restrictions: Any node → ≤ 1 black & ≤ 1 White & ≤ 2 Grey;
Grey → 0/1 Grey; Black → 0 White; White → 0 Black.
(● → GC ○ → CG ● → AU|UA × → U)

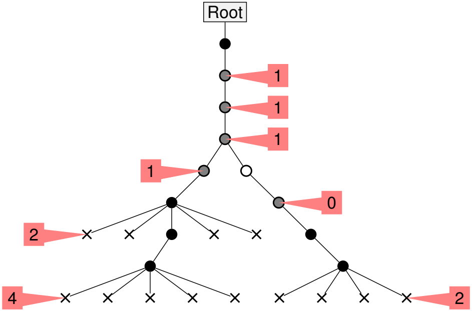




Our Results: Separated Coloring (example)

Descendant restrictions: Any node → ≤ 1 black & ≤ 1 White & ≤ 2 Grey; Grey → 0/1 Grey; Black → 0 White; White → 0 Black.
(● → GC ○ → CG ● → AU|UA × → U)

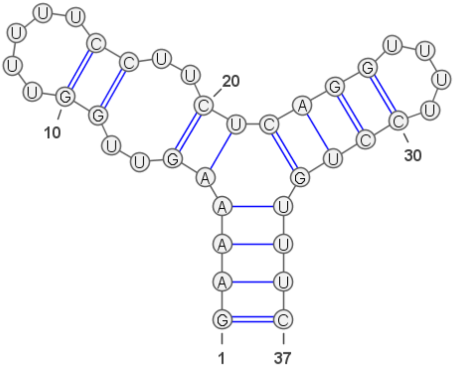


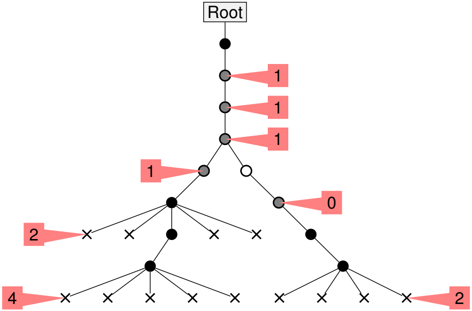


Levels of grey nodes: 0,1
Levels of leaves: 2,4
Separated coloring

Our Results: Separated Coloring (example)

Descendant restrictions: Any node $\rightarrow \leq 1$ black & ≤ 1 White & ≤ 2 Grey;
Grey $\rightarrow 0/1$ Grey; Black $\rightarrow$ 0 White; White $\rightarrow$ 0 Black.
(● → GC ○ → CG ◍ → AU|UA × → U)





Levels of grey nodes: 0,1

Levels of leaves: 2,4

Separated coloring

⇒ **Design:** GAAAAGUUGGUUUUCCUUCUCAGGUUUUCCUGUUUC

Our Results: Designability over the complete alphabet

$\Sigma_{2,0} = \{\mathsf{A}, \mathsf{U}, \mathsf{C}, \mathsf{G}\} + \{\mathsf{G} - \mathsf{C}, \mathsf{A} - \mathsf{U}\}$ base pairs.

Without unpaired position → complete characterization:

R4 $\Sigma_{2,0} \Rightarrow$ Saturated Designable $=$ Degree≤ 4.

With unpaired positions → partial characterization:

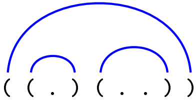
R5 (Necessary) Designable structure cannot contain “*a multiloop of degree* ≥ 5” (motif m_5) or “*a multiloop with unpaired position of degree* ≥ 3” (motif $m_{3\circ}$).

R6 (Sufficient) Separated $=$ Structure that admit a separated (proper) coloring. Then any Separated **structure is Designable in** $\Sigma_{2,0}$.

R7 If $S \in$ Designable(), then k-stutter $S^{[k]} \in$ Designable($\Sigma_{2,0}$).

Our Results: *k*-Stutter (example)

Designable structure:

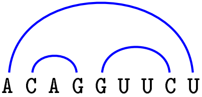


((.) (. .))

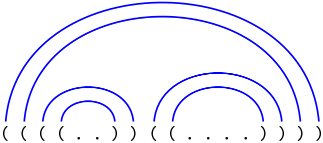
Then 2-stutter is designable as well:

Our Results: *k*-Stutter (example)

Designable structure:

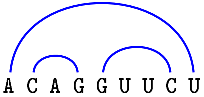


Then 2-stutter is designable as well:

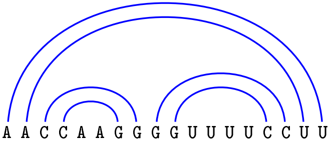


Our Results: *k*-Stutter (example)

Designable structure:

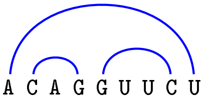


Then 2-stutter is designable as well:

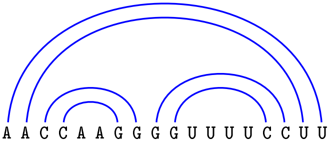


Our Results: *k*-Stutter (example)

Designable structure:



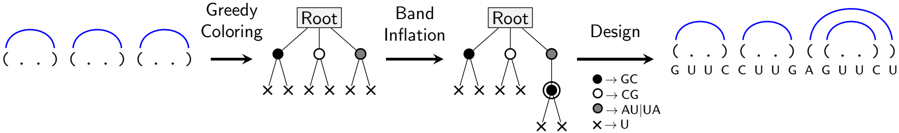
Then 2-stutter is designable as well:



Proof idea: Use König's Theorem (size of max. matching = size of min. vertex cover) to show that an MFE structure for the stutter sequence can't connect a region to two different regions.

Our Results: Structure-Approximating Algorithm

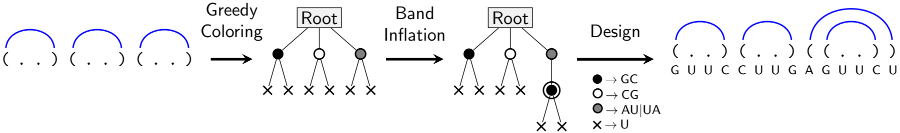
R8 Any structure S without m_5 and $m_{3\circ}$ can be transformed in $\Theta(n)$ time into a designable structure S', by adding at most a single base-pair to its helices.



Main idea: Offset grey vertices and leaves to odd/even levels
→ Coloring is now **separated**

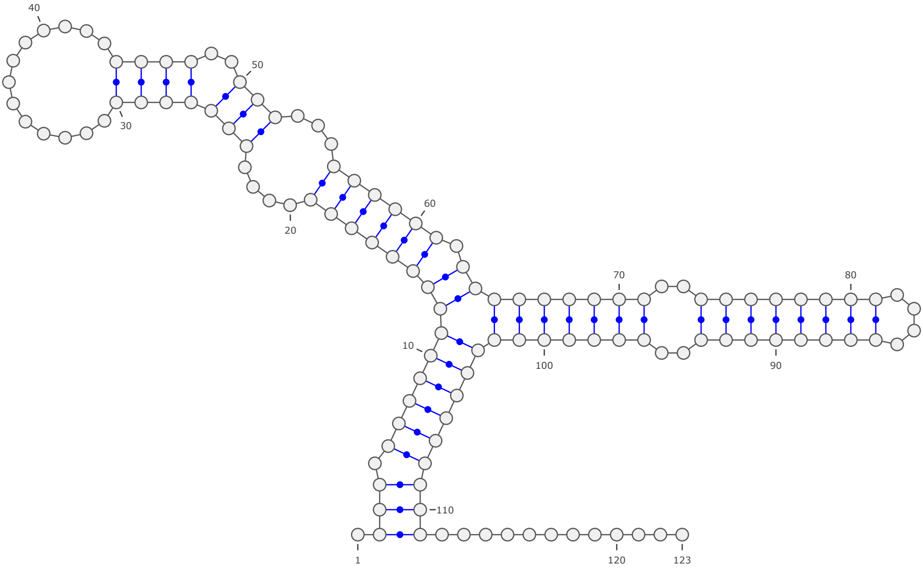
Our Results: Structure-Approximating Algorithm

R8 Any structure S without m_5 and $m_{3\circ}$ can be transformed in $\Theta(n)$ time into a designable structure S', by adding at most a single base-pair to its helices.

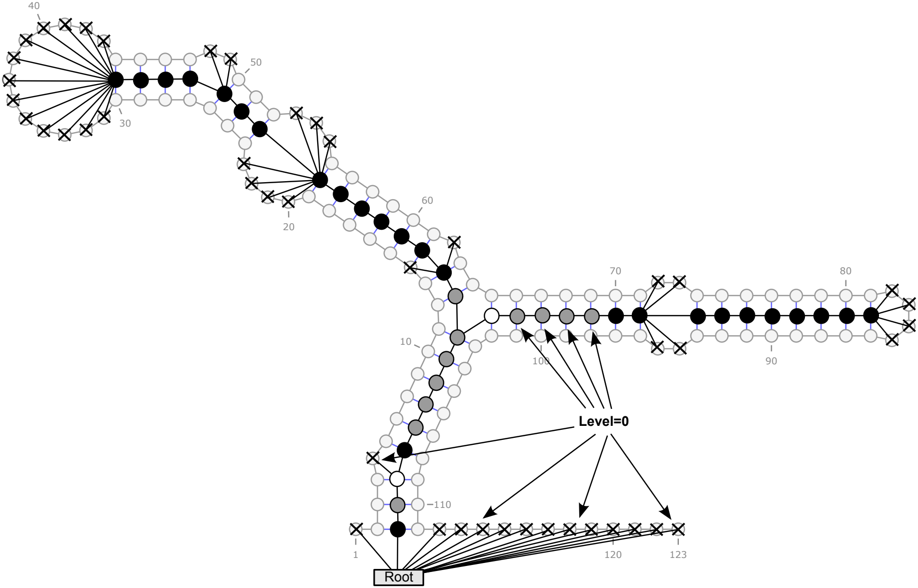


Main idea: Offset grey vertices and leaves to odd/even levels
→ Coloring is now **separated**

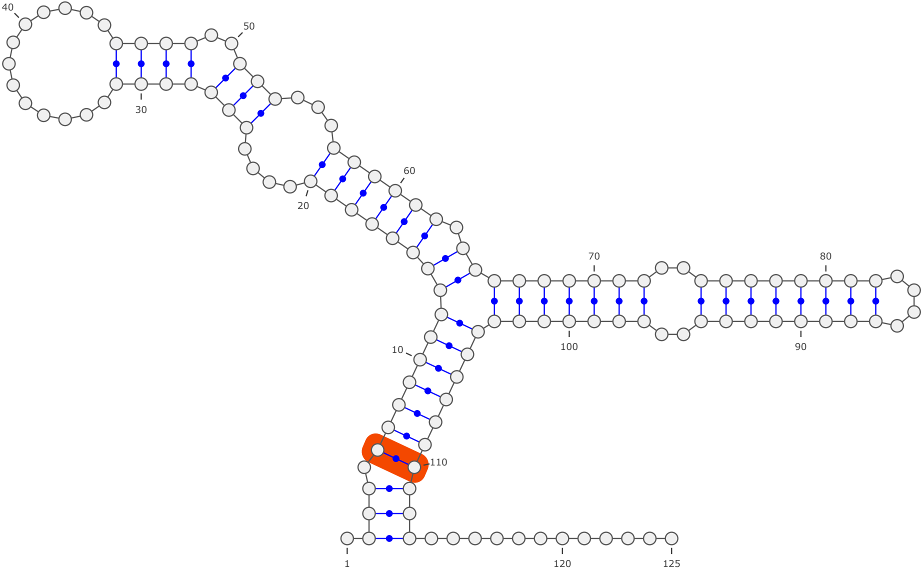
Example



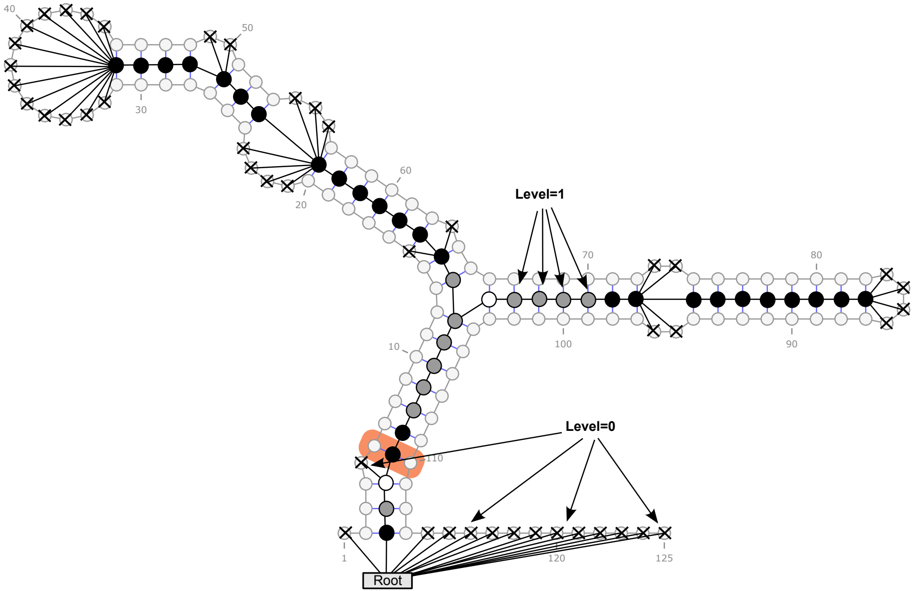
Example



Example



Example



Remarks

- Results also hold in **Nussinov** energy model ($A-U, G-C, G-U$ + weights) $\Rightarrow$**Stacking** energy model? **Turner**?

- Characterized classes are mostly **easy**:
  - **Designable** classes $\rightarrow$ Linear time **algorithms**
  - **Non-designable** classes $\rightarrow$ Linear time **membership tests**

- **Forbidden local motifs** (*e.g.* m_5 & $m_{3\circ}$) can be found in any energy model

  $\Rightarrow$ **Designable structures** $\subset$ **Tree-like** objects with **forbidden motifs**

  $+$ **Basic analytic combinatorics** (*à la* Philippe Flajolet):
  - #Secondary structures $\in \Theta\left(\frac{\alpha^n}{n\sqrt{n}}\right)$ $(\theta = 0 \rightarrow \alpha = 3)$
  - #Designable structures $\in \mathcal{O}\left(\frac{\beta^n}{n\sqrt{n}}\right)$, $\beta < \alpha$

  Proportion of designable structures: $\left(\frac{\beta}{\alpha}\right)^n$, **exponentially decreasing** with *n*.

  Possible consequences on **RNA neutral network** studies
  + motivation for identifying **new forbidden motifs**

Remarks

- Results also hold in **Nussinov** energy model ($A-U, G-C, G-U$ + weights)
  ⇒**Stacking** energy model? **Turner**?

- Characterized classes are mostly **easy**:
  - **Designable** classes → Linear time **algorithms**
  - **Non-designable** classes → Linear time **membership tests**

- **Forbidden local motifs** (*e.g.* m_5 & $m_{3\circ}$) can be found in any energy model
  ⇒ **Designable structures** $\subset$ **Tree-like** objects with **forbidden motifs**
  $+$ **Basic analytic combinatorics** (*à la* Philippe Flajolet):
  - #Secondary structures $\in \Theta\left(\frac{\alpha^n}{n\sqrt{n}}\right)$ $(\theta = 0 \to \alpha = 3)$
  - #Designable structures $\in \mathcal{O}\left(\frac{\beta^n}{n\sqrt{n}}\right), \beta < \alpha$

  Proportion of designable structures: $\left(\frac{\beta}{\alpha}\right)^n$, **exponentially decreasing** with n.

  Possible consequences on **RNA neutral network** studies
  + motivation for identifying **new forbidden motifs**

Remarks

- Results also hold in **Nussinov** energy model ($\mathrm{A}-\mathrm{U}, \mathrm{G}-\mathrm{C}, \mathrm{G}-\mathrm{U}$ + weights)
  ⇒**Stacking** energy model? **Turner**?

- Characterized classes are mostly **easy**:
  - **Designable** classes → Linear time **algorithms**
  - **Non-designable** classes → Linear time **membership tests**

- **Forbidden local motifs** (*e.g.* m_5 & $m_{3\,\circ}$) can be found in any energy model

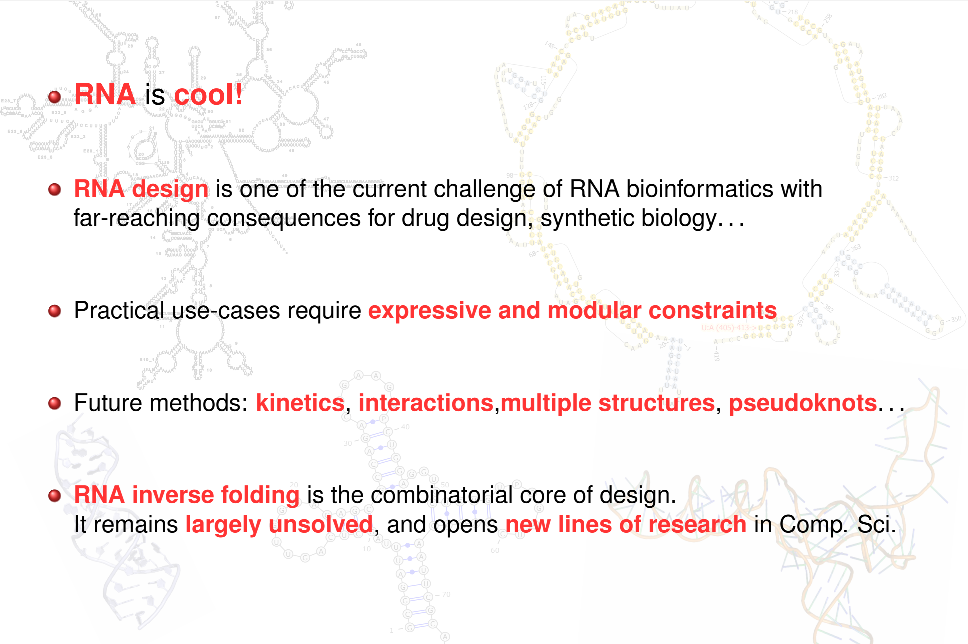
  ⇒ **Designable structures** $\subset$ **Tree-like** objects with **forbidden motifs**

  $+$ **Basic analytic combinatorics** (*à la* Philippe Flajolet):
  - #Secondary structures $\in \Theta\left(\frac{\alpha^n}{n\sqrt{n}}\right)$ $(\theta = 0 \to \alpha = 3)$
  - #Designable structures $\in \mathcal{O}\left(\frac{\beta^n}{n\sqrt{n}}\right)$ $, \beta < \alpha$

  Proportion of designable structures: $\left(\frac{\beta}{\alpha}\right)^n$, **exponentially decreasing** with *n*.

  Possible consequences on **RNA neutral network** studies
  + motivation for identifying **new forbidden motifs**

Conclusions



- **RNA** is **cool!**
- **RNA design** is one of the current challenge of RNA bioinformatics with far-reaching consequences for drug design, synthetic biology. . .
- Practical use-cases require **expressive and modular constraints**
- Future methods: **kinetics**, **interactions**,**multiple structures**, **pseudoknots**. . .
- **RNA inverse folding** is the combinatorial core of design.
  It remains **largely unsolved**, and opens **new lines of research** in Comp. Sci.

Thanks

University McGill 🇨🇦

Vladimir Reinharz
Jérôme Waldispühl

MIT 🇺🇸

Bonnie Berger
Srinivas Devadas
Alex Levin
Mieszko Lis
Charles O'Donnell

LRI – Univ. Paris Sud 🇫🇷

Alain Denise
Vincent Le Gallic

Wuhan University 🇨🇳

Yi Zhang
Yu Zhou

LIGM – Marne la Vallée 🇫🇷

Stéphane Vialette

LIX – Ecole Polytechnique 🇫🇷

Mireille Regnier

Simon Fraser University 🇨🇦

Jozef Hales
Jan Manuch (UBC)
Ladislav Stacho

Cédric Chauve
Julien Courtiel

TBI Vienna 🇦🇹

Ronnie Lorenz
Andrea Tanzer

Job offers: PhD & Postdoc on RNA kinetics@Inria Saclay+Lille

Agence Nationale de la Recherche ANR   Inria INVENTEURS DU MONDE NUMÉRIQUE   cnrs dépasser les frontières