Combinatorial RNA Design: Designability and Structure-Approximating Algorithm Towards a theory of RNA design

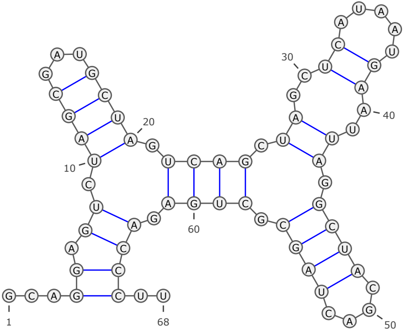
Jozef Haleš[1] Ján Maňuch[1,4] Yann Ponty[2,3] Ladislav Stacho[1]

[1]Simon Fraser University, Canada
[2]Pacific Institute for Mathematical Sciences, Canada
[3]CNRS – Ecole Polytechnique, France
[4]University of British Columbia, Canada

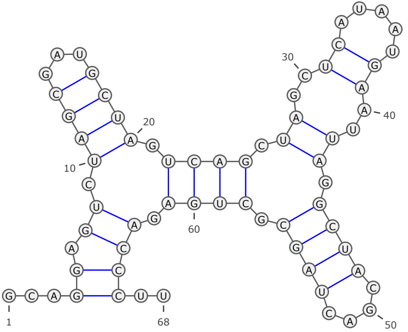
Benasque 2015

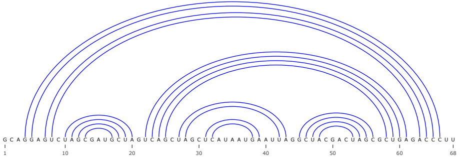
Results appear in **Combinatorial Pattern Matching'15** proceedings

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



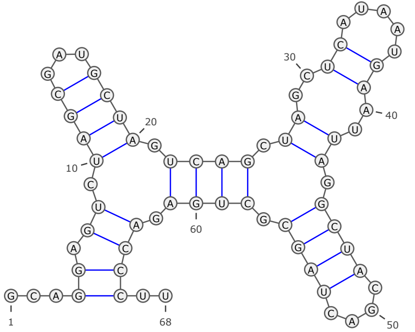
Structure = Bunch of **non-crossing** base-pairs.

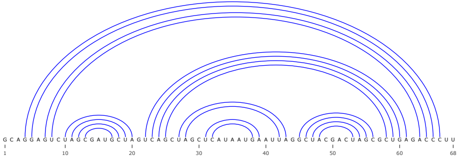




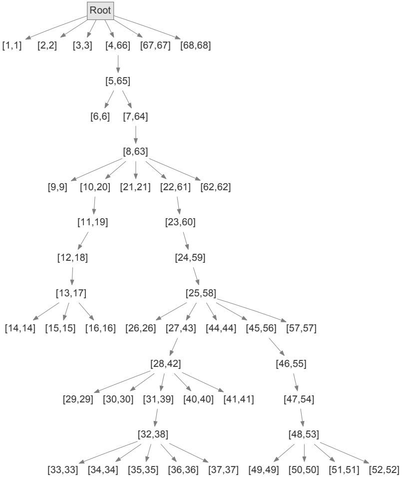
arc diagram

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





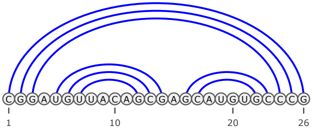
arc diagram

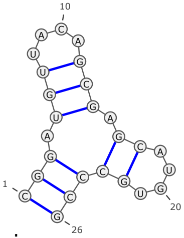


tree representation

RNA folding

Input: RNA sequence w, length $n := |w|$





- **RNA structure S:** (Partial) matching of positions in sequence w
- **Motifs:** Sequence/structure features (e.g. pairs, Stacking pairs, Nearest neighbor, ...)
- **Energy model.**

  Motif $\rightarrow$ Free-energy contribution $\Delta G(\cdot) \in \mathbb{R}^{-} \cup \{+\infty\}$

  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

Problem (RNA-FOLD$_{\mathcal{M}}$ problem)

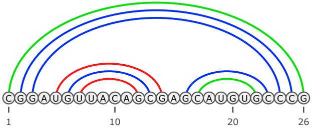
Input: *RNA sequence w*

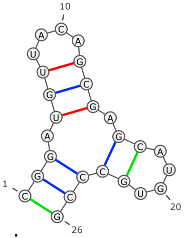
Output: *set of PKF structures* $\arg\min_{S \in \mathcal{S}_{|w|}} E_{\mathcal{M}}(w, S)$.

RNA-FOLD$_{\mathcal{M}}(w)$ solved in time $O(n^3/\log(n))$ [Frid et al. (2010),etc.].

RNA folding

Input: RNA sequence w, length $n := |w|$





- **RNA structure S:** (Partial) matching of positions in sequence w
- **Motifs:** Sequence/structure features (e.g. pairs, Stacking pairs, Nearest neighbor. . .)
- **Energy model.**

  Motif $\rightarrow$ Free-energy contribution $\Delta G(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$

  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

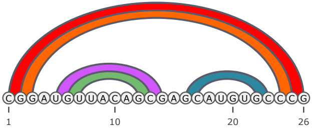
Problem (RNA-FOLD$_{\mathcal{M}}$ problem)

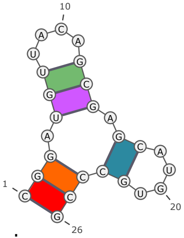
Input: *RNA sequence w*
Output: *set of PKF structures* $\arg\min_{S\in\mathcal{S}_{|w|}} E_{\mathcal{M}}(w,S)$.

RNA-FOLD$_{\mathcal{M}}(w)$ solved in time $O(n^3/\log(n))$ [Frid et al. (2010),etc.].

RNA folding

Input: RNA sequence w, length $n := |w|$





- **RNA structure S:** (Partial) matching of positions in sequence w
- **Motifs:** Sequence/structure features (e.g. pairs, Stacking pairs, Nearest neighbor. . .)
- **Energy model.**

  Motif $\rightarrow$ Free-energy contribution $\Delta G(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$

  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

Problem (RNA-FOLD$_\mathcal{M}$ **problem)**

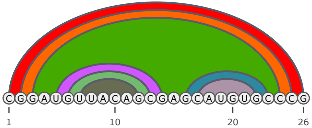
Input: *RNA sequence w*

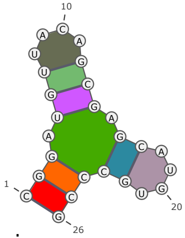
Output: *set of PKF structures* $\arg\min_{S \in \mathcal{S}_{|w|}} E_\mathcal{M}(w, S)$.

RNA-FOLD$_\mathcal{M}(w)$ solved in time $O(n^3 / \log(n))$ [Frid et al. (2010),etc.].

RNA folding

Input: RNA sequence w, length $n := |w|$





- **RNA structure S:** (Partial) matching of positions in sequence w
- **Motifs:** Sequence/structure features (e.g. pairs, Stacking pairs, Nearest neighbor. . .)
- **Energy model.**

  Motif $\rightarrow$ Free-energy contribution $\Delta G(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$

  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

Problem (RNA-FOLD$_{\mathcal{M}}$ problem)

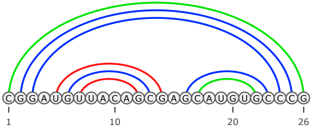
Input: *RNA sequence w*

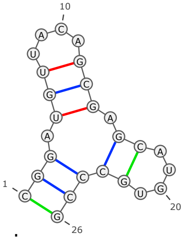
Output: *set of PKF structures* $\arg\min_{S \in \mathcal{S}_{|w|}} E_{\mathcal{M}}(w, S)$.

RNA-FOLD$_{\mathcal{M}}(w)$ solved in time $O(n^3/\log(n))$ [Frid et al. (2010),etc.].

RNA folding

Input: RNA sequence w, length $n := |w|$





- **RNA structure S:** (Partial) matching of positions in sequence w
- **Motifs:** Sequence/structure features (e.g. pairs, Stacking pairs, Nearest neighbor. . .)
- **Energy model.**

  Motif $\rightarrow$ Free-energy contribution $\Delta G(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$

  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

Problem (RNA-FOLD$_{\mathcal{M}}$ problem)

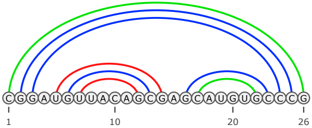
Input: *RNA sequence w*

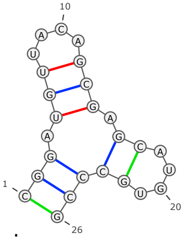
Output: *set of PKF structures* $\arg\min_{S \in \mathcal{S}_{|w|}} E_{\mathcal{M}}(w, S)$.

RNA-FOLD$_{\mathcal{M}}(w)$ solved in time $O(n^3/\log(n))$ [Frid et al. (2010),etc.].

RNA folding

Input: RNA sequence w, length $n := |w|$





- **RNA structure S:** (Partial) matching of positions in sequence w
- **Motifs:** Sequence/structure features (e.g. pairs, Stacking pairs, Nearest neighbor. . .)
- **Energy model.**

  Motif $\rightarrow$ Free-energy contribution $\Delta G(\cdot) \in \mathbb{R}^- \cup \{+\infty\}$

  Free-Energy $E_w(S)$: Sum over (independently contributing) motifs in S

Problem (RNA-FOLD$_\mathcal{M}$ problem)

Input: *RNA sequence w*
Output: *set of PKF structures* $\arg\min_{S \in \mathcal{S}_{|w|}} E_\mathcal{M}(w, S)$.

RNA-FOLD$_\mathcal{M}(w)$ solved in time $O(n^3/\log(n))$ [Frid et al. (2010),etc.].

RNA Design Problem

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

Problem (RNA-DESIGN$_{\mathcal{M},\Sigma.\Delta}$ **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. . .

$\Rightarrow$ **Stick to a simplified model!**

RNA Design Problem (simplified)

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

Problem (RNA-DESIGN$_\Sigma$ **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 (RNA-DESIGN$_\Sigma$ **problem)**

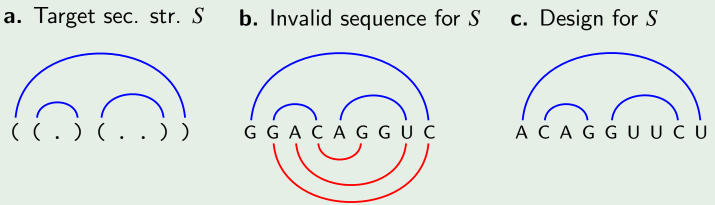
Input: *Secondary structure S*
Output: *RNA sequence $w \in \Sigma^\star$ — called a design for S — such that:*

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

or $\varnothing$ if no such sequence exists.

Designable(Σ): All designable structures

Example

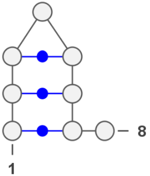


Our Results: Definitions and notations

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



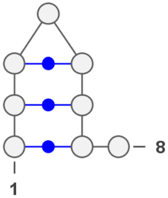
Unpaired$_S = \{4, 8\}$

Our Results: Definitions and notations

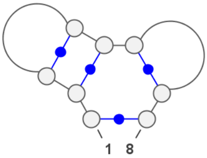
Given a secondary structure S:

- 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



not saturated



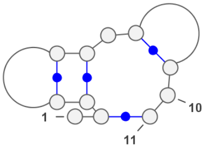
saturated

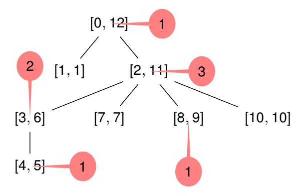
Our Results: Definitions and notations

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





$D(S) = 3$

Our Results: Designability over Restricted Alphabets

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

R1 For every $u \in \mathbb{N}^+$, Designable$(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R2 Designable$(\Sigma_{1,0})$ = (Saturated $\cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 Designable$(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Our Results: Designability over Restricted Alphabets

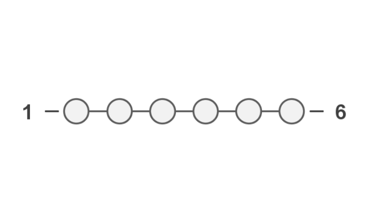
$\Sigma_{c,u}$ = Alphabet with c pairs of complementary bases and u unpairable bases.

R1 For every $u \in \mathbb{N}^+$, Designable$(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R2 Designable$(\Sigma_{1,0})$ = (Saturated $\cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 Designable$(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Example



Our Results: Designability over Restricted Alphabets

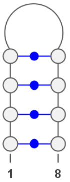
$\Sigma_{c,u}$ = Alphabet with *c* pairs of complementary bases and *u* unpairable bases.

R1 For every $u \in \mathbb{N}^+$, Designable$(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R2 Designable$(\Sigma_{1,0}) =$ (Saturated $\cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 Designable$(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Example



Our Results: Designability over Restricted Alphabets

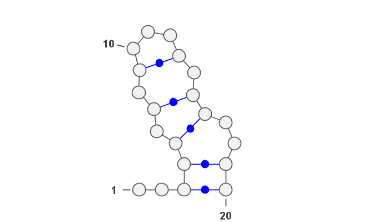
$\Sigma_{c,u}$ = Alphabet with c pairs of complementary bases and u unpairable bases.

R1 For every $u \in \mathbb{N}^+$, $\text{Designable}(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R2 $\text{Designable}(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 $\text{Designable}(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Example



Our Results: Designability over Restricted Alphabets

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

R1 For every $u \in \mathbb{N}^+$, Designable$(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R2 Designable$(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 Designable$(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

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 For every $u \in \mathbb{N}^+$, Designable$(\Sigma_{0,u}) = \{(n,\emptyset) \mid \forall n \in \mathbb{N}\}$;

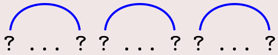
R2 Designable$(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n,\emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 Designable$(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Question: Why not degree 3?

Proof.

In the root:



— we can only use C · G or G · C □

Our Results: Designability over Restricted Alphabets

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

R1 For every $u \in \mathbb{N}^+$, Designable$(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R2 Designable$(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 Designable$(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Question: Why not degree 3?

Proof.

In the root:

C . . . G G . . . C C . . . G — one of them has to repeat □

Our Results: Designability over Restricted Alphabets

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

R1 For every $u \in \mathbb{N}^+$, $\text{Designable}(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

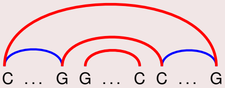
R2 $\text{Designable}(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 $\text{Designable}(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Question: Why not degree 3?

Proof.

In the root:



— there is an alternative fold □

Our Results: Designability over Restricted Alphabets

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

R1 For every $u \in \mathbb{N}^+$, $\text{Designable}(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

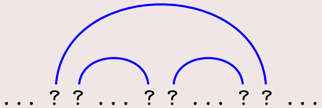
R2 $\text{Designable}(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 $\text{Designable}(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Question: Why not degree 3?

Proof.

In an internal node:



□

Our Results: Designability over Restricted Alphabets

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

R1 For every $u \in \mathbb{N}^+$, $\text{Designable}(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

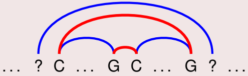
R2 $\text{Designable}(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 $\text{Designable}(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Question: Why not degree 3?

Proof.

In an internal node:



— either we get a repeat, or... □

Our Results: Designability over Restricted Alphabets

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

R1 For every $u \in \mathbb{N}^+$, Designable$(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

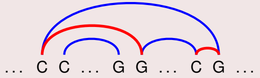
R2 Designable$(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 Designable$(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

Question: Why not degree 3?

Proof.

In an internal node:



— ...or, the parent has the reversed base pair of a child □

Our Results: Designability over Restricted Alphabets

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

R1 For every $u \in \mathbb{N}^+$, $\text{Designable}(\Sigma_{0,u}) = \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R2 $\text{Designable}(\Sigma_{1,0}) = (\text{Saturated} \cap \{S \mid D(S) \leq 2\}) \cup \{(n, \emptyset) \mid \forall n \in \mathbb{N}\}$;

R3 $\text{Designable}(\Sigma_{1,1}) = \{S \mid D(S) \leq 2\}$.

This can be easily generalized to:

Lemma

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

Our Results: Designability over the Complete Alphabet

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

Without unpaired position $\rightarrow$ complete characterization:

R4 Designable($\Sigma_{2,0}$) $\cap$ Saturated $= \{S \mid D(S) \leq 4\} \cap$ Saturated.

R5 (Necessary) $S \in$ Designable($\Sigma_{2,0}$) cannot contain "*a **pure** multiloop of degree ≥ 5*" (motif m_5) or "*a multiloop with unpaired position of degree ≥ 3*" (motif $m_{3\circ}$).

R6 (Sufficient) Separated = Set of structures that admit a separated (proper) coloring. Then Separated $\subset$ Designable($\Sigma_{2,0}$).

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 Designable($\Sigma_{2,0}$) $\cap$ Saturated $= \{S \mid D(S) \leq 4\} \cap$ Saturated.

R5 (Necessary) $S \in$ Designable($\Sigma_{2,0}$) cannot contain "*a **pure** multiloop of degree* ≥ 5" (motif m_5) or "*a multiloop with unpaired position of degree* ≥ 3" (motif $m_{3\circ}$).

R6 (Sufficient) Separated = Set of structures that admit a separated (proper) coloring. Then Separated $\subset$ Designable($\Sigma_{2,0}$).

Our Results: Designability over the Complete Alphabet

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

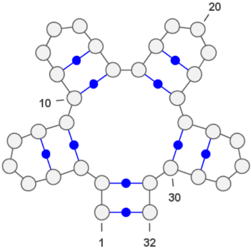
Without unpaired position → complete characterization:

R4 Designable($\Sigma_{2,0}$) $\cap$ Saturated $= \{S \mid D(S) \leq 4\} \cap$ Saturated.

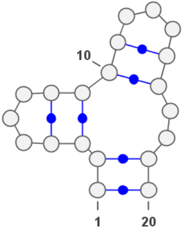
With unpaired positions → partial characterization:

R5 (Necessary) $S \in$ Designable($\Sigma_{2,0}$) cannot contain "*a **pure** multiloop of degree* ≥ 5" (motif m_5) or "*a multiloop with unpaired position of degree* ≥ 3" (motif $m_{3\circ}$).

R6 (Sufficient) Separated = Set of structures that admit a separated (proper) coloring. Then Separated $\subset$ Designable($\Sigma_{2,0}$).



m5



m3o

Our Results: Designability over the Complete Alphabet

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

Without unpaired position $\rightarrow$ complete characterization:

R4 Designable($\Sigma_{2,0}$) $\cap$ Saturated $= \{S \mid D(S) \leq 4\} \cap$ Saturated.

With unpaired positions $\rightarrow$ partial characterization:

R5 (Necessary) $S \in$ Designable($\Sigma_{2,0}$) cannot contain "*a **pure** multiloop of degree ≥ 5*" (motif m_5) or "*a multiloop with unpaired position of degree ≥ 3*" (motif $m_{3\circ}$).

R6 (Sufficient) Separated $=$ Set of structures that admit a separated (proper) coloring. Then Separated $\subset$ Designable($\Sigma_{2,0}$).

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

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

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:

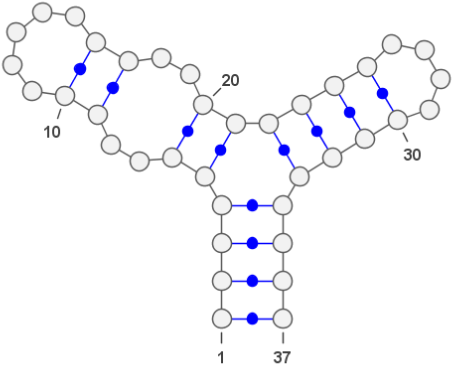
1. each internal node has at most one black, one white and two grey children;
2. a grey node has at most one grey child;
3. a black node does not have a white child; and
4. a white node does not have a black child.

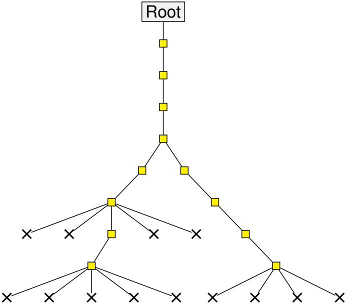
Level of a node = #black nodes $-$ #white nodes on the path to the root.

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

Our Results: Separated Coloring (example)

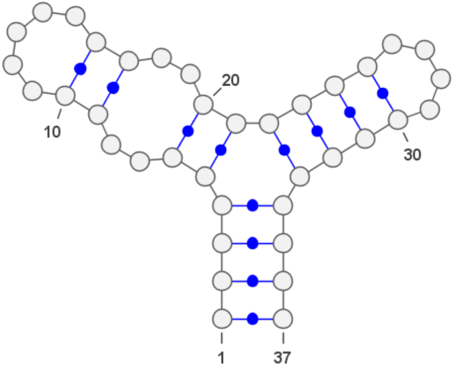
Descendant restrictions: Any node $\rightarrow \leq 1$ black & ≤ 1 White & ≤ 2 Grey;
Grey $\rightarrow 0/1$ Grey; Black $\rightarrow 0$ White; White $\rightarrow 0$ Black.
(● → GC ○ → CG ◉ → AU|UA × → U)

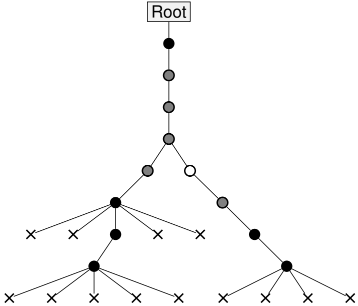




Our Results: Separated Coloring (example)

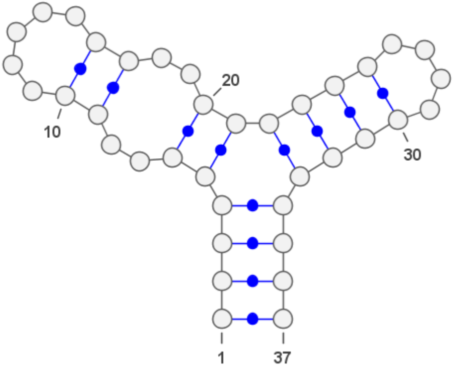
Descendant restrictions: Any node $\rightarrow \leq 1$ black & ≤ 1 White & ≤ 2 Grey;
Grey $\rightarrow 0/1$ Grey; Black $\rightarrow$ 0 White; White $\rightarrow$ 0 Black.
(● → GC   ○ → CG   ◉ → AU|UA   × → U)

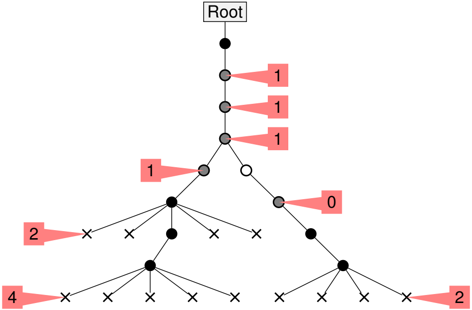




Our Results: Separated Coloring (example)

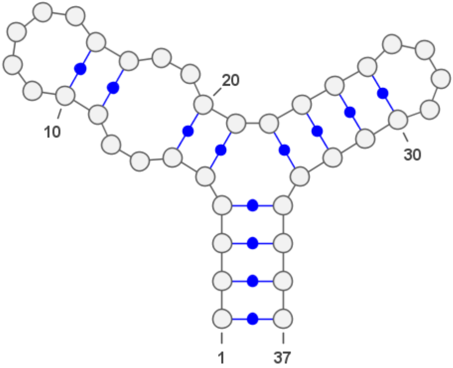
Descendant restrictions: Any node $\rightarrow \leq 1$ black & ≤ 1 White & ≤ 2 Grey;
Grey $\rightarrow 0/1$ Grey; Black $\rightarrow$ 0 White; White $\rightarrow$ 0 Black.
(● → GC ○ → CG ◉ → AU|UA × → U)

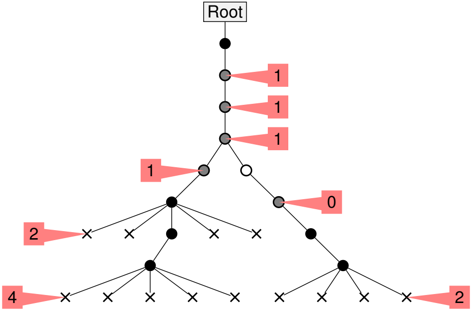




Our Results: Separated Coloring (example)

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.
(● $\rightarrow$ GC ○ $\rightarrow$ CG ◉ $\rightarrow$ AU|UA × $\rightarrow$ U)

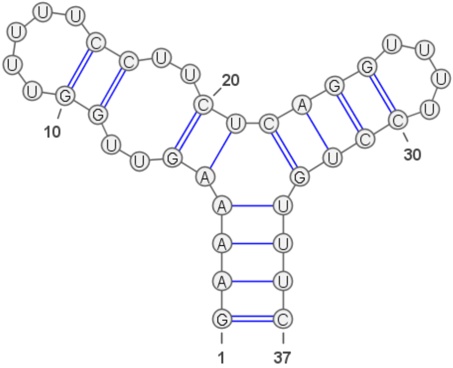


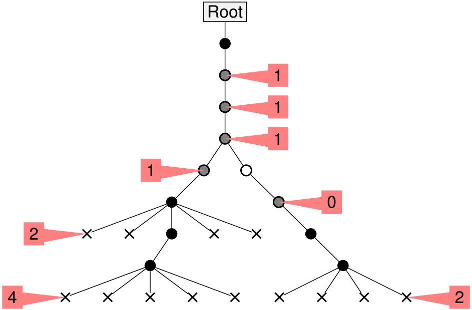


Levels of grey nodes: 0,1
Levels of leaves: 2,4
Separated coloring

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.
(● $\rightarrow$ GC ○ $\rightarrow$ CG ● $\rightarrow$ AU|UA × $\rightarrow$ U)





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

$\Rightarrow$ **Design:** GAAAAGUUGGUUUUUCCUUCUCAGGUUUUCCUGUUUC

Our Results: Designability over the complete alphabet

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

Without unpaired position $\rightarrow$ complete characterization:

R4 Designable($\Sigma_{2,0}$) $\cap$ Saturated $= \{S \mid D(S) \leq 4\} \cap$ Saturated.

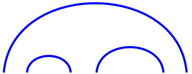
With unpaired positions $\rightarrow$ partial characterization:

R5 (Necessary) $S \in$ Designable($\Sigma_{2,0}$) cannot contain "*a **pure** multiloop of degree* ≥ 5" (motif m_5) or "*a multiloop with unpaired position of degree* ≥ 3" (motif $m_{3\circ}$).

R6 (Sufficient) Separated $=$ Set of structures that admit a separated (proper) coloring. Then Separated $\subset$ Designable($\Sigma_{2,0}$).

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

Our Results: *k*-Stutter (example)

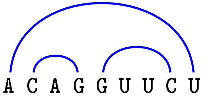


Designable structure: ((.) (. .))

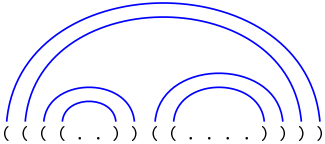
Then 2-stutter is designable as well:

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

Designable structure:



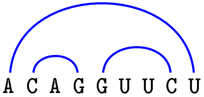
Then 2-stutter is designable as well:



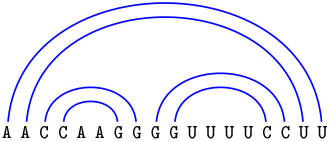
Proof idea: Use König's Theorem (size of max. matching = size of min. vertex cover) to show that an MFE structure of the stutter sequence can't connect a region to two different regions.

Our Results: *k*-Stutter (example)

Designable structure:



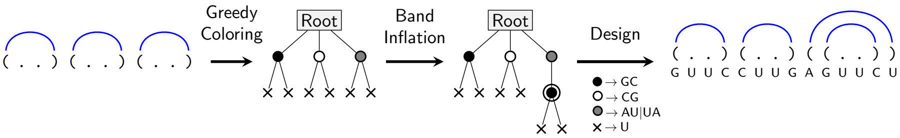
Then 2-stutter is designable as well:



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

Our Results: Structure-Approximating Algorithm

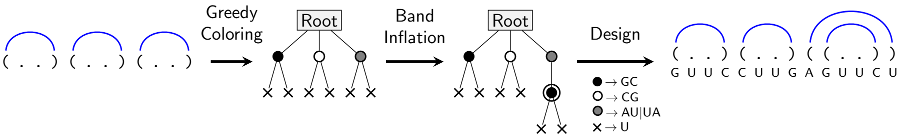
R8 Any structure S without m_5 and $m_{3\circ}$ can be transformed in $\Theta(n)$ time into a designable structure S', by adding at most a single base-pair to its helices.



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

Our Results: Structure-Approximating Algorithm

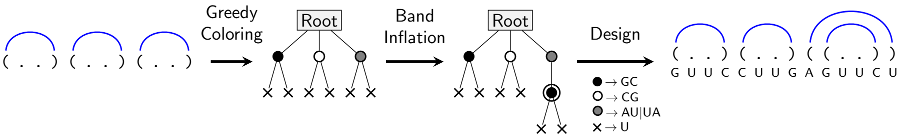
R8 Any structure S without m_5 and $m_{3\circ}$ can be transformed in $\Theta(n)$ time into a designable structure S', by adding at most a single base-pair to its helices.



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

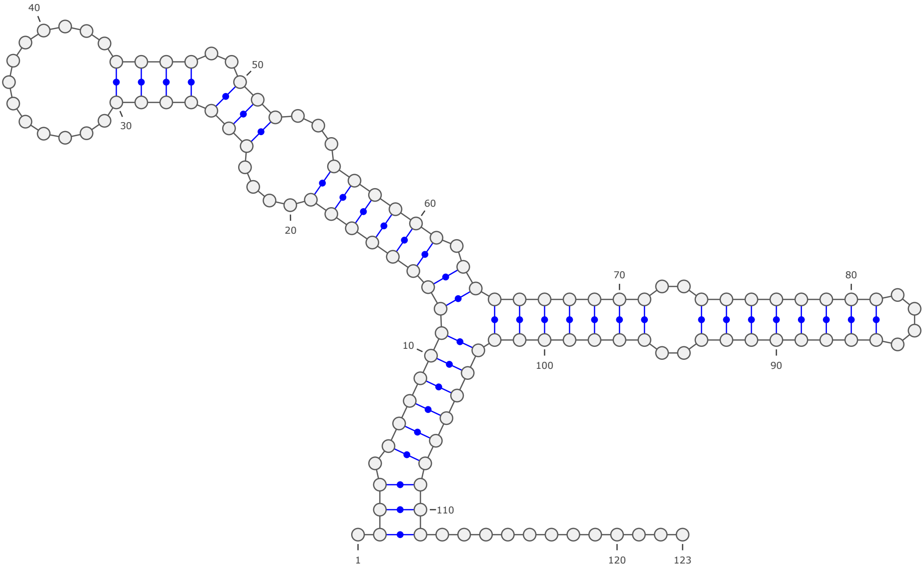
Our Results: Structure-Approximating Algorithm

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.

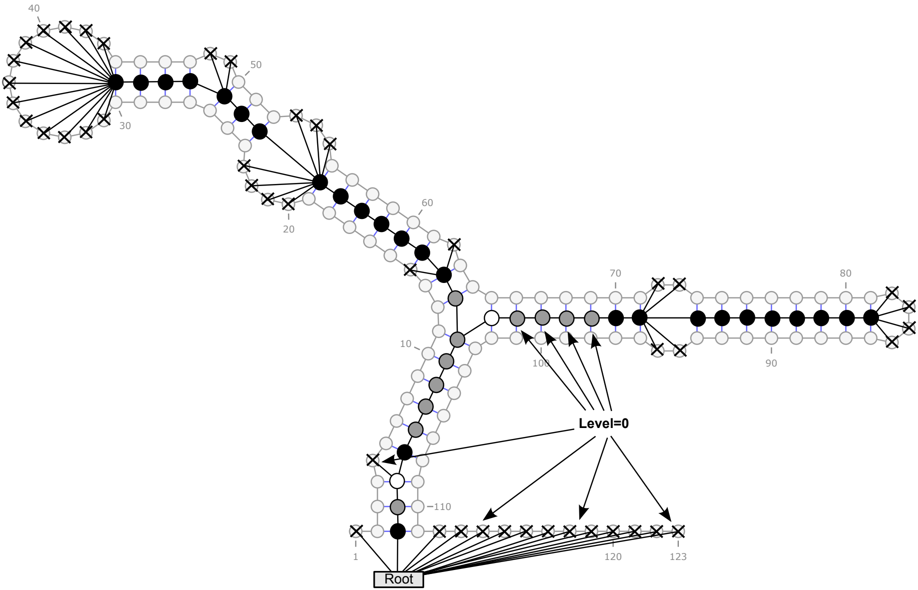


Main idea: Offset grey vertices and leaves to odd/even levels
→ Coloring is now **separated**

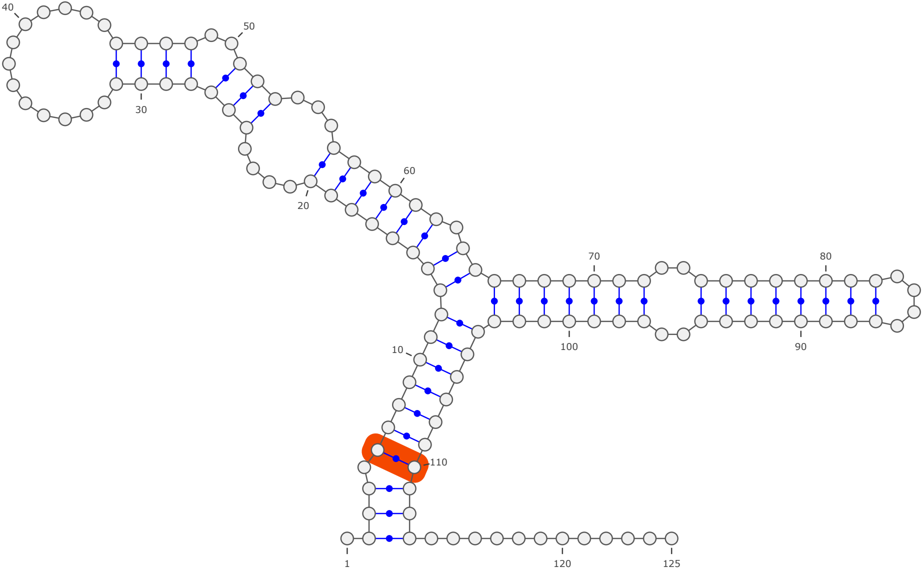
Example



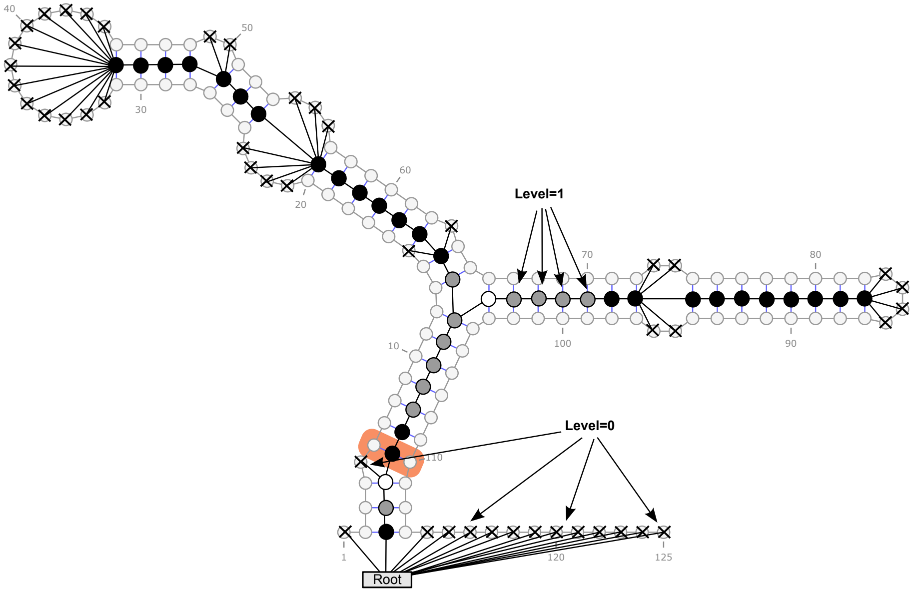
Example



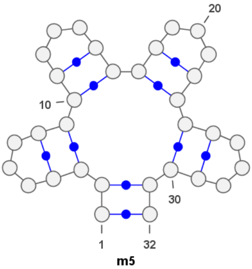
Example



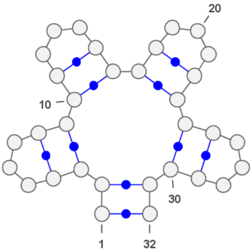
Example

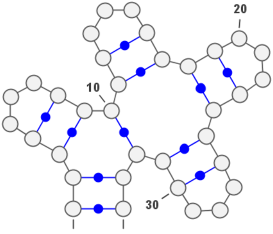


Remark: Breaking motifs



Remark: Breaking motifs



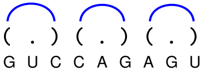


Open Problems and Future Work

1. What's the complexity of RNA-DESIGN problem? Could it be polynomial?
2. What's the complexity of RNA-DESIGN problem restricted to designs that use only one base for all unpaired position?
3. What's the complexity of determining if a structure has a separated coloring?
4. Extend the results to more complex energy models.
   Our results hold for the **Base-pair sum** model, as long as $-\delta_B(G, U)$ is smaller than $-\delta_B(C, G)$ and $-\delta_B(A, U)$.
5. Better bound on #arcs added by our structure approximation algorithm.
6. Combinatorics of designable structures.

Open Problems and Future Work

1. What's the complexity of RNA-DESIGN problem? Could it be polynomial?
2. What's the complexity of RNA-DESIGN problem restricted to designs that use only one base for all unpaired position?



3. What's the complexity of determining if a structure has a separated coloring?
4. Extend the results to more complex energy models.
   Our results hold for the **Base-pair sum** model, as long as $-\delta_B(G, U)$ is smaller than $-\delta_B(C, G)$ and $-\delta_B(A, U)$.
5. Better bound on #arcs added by our structure approximation algorithm.
6. Combinatorics of designable structures.

1. What's the complexity of RNA-DESIGN problem? Could it be polynomial?
2. What's the complexity of RNA-DESIGN problem restricted to designs that use only one base for all unpaired position?
3. What's the complexity of determining if a structure has a separated coloring?
4. Extend the results to more complex energy models.
   Our results hold for the **Base-pair sum** model, as long as $-\delta_B(G, U)$ is smaller than $-\delta_B(C, G)$ and $-\delta_B(A, U)$.
5. Better bound on #arcs added by our structure approximation algorithm.
6. Combinatorics of designable structures.

Open Problems and Future Work

1. What's the complexity of RNA-DESIGN problem? Could it be polynomial?
2. What's the complexity of RNA-DESIGN problem restricted to designs that use only one base for all unpaired position?
3. What's the complexity of determining if a structure has a separated coloring?
4. Extend the results to more complex energy models.
   Our results hold for the **Base-pair sum** model, as long as $-\delta_B(G, U)$ is smaller than $-\delta_B(C, G)$ and $-\delta_B(A, U)$.
5. Better bound on #arcs added by our structure approximation algorithm.
6. Combinatorics of designable structures.

Open Problems and Future Work

1. What's the complexity of RNA-DESIGN problem? Could it be polynomial?
2. What's the complexity of RNA-DESIGN problem restricted to designs that use only one base for all unpaired position?
3. What's the complexity of determining if a structure has a separated coloring?
4. Extend the results to more complex energy models.
   Our results hold for the **Base-pair sum** model, as long as $-\delta_B(G, U)$ is smaller than $-\delta_B(C, G)$ and $-\delta_B(A, U)$.
5. Better bound on #arcs added by our structure approximation algorithm.
6. Combinatorics of designable structures.

Open Problems and Future Work

1. What's the complexity of RNA-DESIGN problem? Could it be polynomial?
2. What's the complexity of RNA-DESIGN problem restricted to designs that use only one base for all unpaired position?
3. What's the complexity of determining if a structure has a separated coloring?
4. Extend the results to more complex energy models.
   Our results hold for the **Base-pair sum** model, as long as $-\delta_B(G, U)$ is smaller than $-\delta_B(C, G)$ and $-\delta_B(A, U)$.
5. Better bound on #arcs added by our structure approximation algorithm.
6. Combinatorics of designable structures.