Nucleic Acids design targeting integer-valued features: FPT counting and uniform sampling

Yann Ponty · Sebastian Will · Stefan Hammer

École Polytechnique · University of Vienna · University of Leipzig

WEPA 2018, Pisa
The central dogma

DNA → Transcription → RNA → Translation → Proteins
The central dogma V2

DNA

RNA

Carrier

Transfer

Transcription

Maturation

Regulation

Participates

Synthesis

Translation

Proteins

RNA functions
- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- ...
Sampling for multi-target RNA design

RNA design

GAUCUCACG GUUCAA

Structure prediction

A U
G C
Complementarity

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

(((.....)))..
Sampling for multi-target RNA design

RNA Design.

GAUCUCACGGGUCAAA

Complementarity

((((....))))..
Positive and negative RNA design

- **Positive structural design**
  Design sequences $S$ with high affinity to the given structure(s) $\mathcal{R}$.

  \[
  \text{Optimize energy } \sum_{R \in \mathcal{R}} E(R|S) \quad \text{(or target specific energies)}
  \]

  \[= \text{IN-design}\]

- **Negative structural design**
  Moreover, avoid high(er) affinity for all other structures.

  \[
  \text{Optimize probability } \prod_{R \in \mathcal{R}} \Pr(R|S)
  \]

  \[= \text{OUT-design}\]
Multi-target design of RNA sequences

Bio-example: design riboswitches for translational control
Multi-target design of RNA sequences

Bio-example: design riboswitches for translational control

Multiple structures (=multiple design targets)

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

(((.))).(((.(.))).)...)
((.))((.))...)...(((.)))
......(((.(.))))...))...
Multi-target design of RNA sequences

Bio-example: design riboswitches for translational control

Multiple structures (=multiple design targets)

Task: generate seq’s with specific properties

- Low/specific energy for multiple structures
- Forbid motifs to appear anywhere in design; Force, each at least once
- Control overall composition (GC-content)

Approach: controlled sampling
RNA sequence/structure compatibility

Complementarity of bases:

Given multiple secondary structures $\mathcal{R} = \{R_1, \ldots, R_k\}$ of length $n$, a sequence $S \in \{A, C, G, U\}$ is compatible with $(\mathcal{R}, n)$ iff

$$\forall (i, j) \in R \in \mathcal{R} : (S_i, S_j) \text{ is complementary}$$

Problems given $(\mathcal{R}, n)$:

- Decision: is there any compatible $S$
- Find/Construct a compatible $S$
- Count the compatible $S$
- Generate $S$ uniformly (among all compatible ones)
Multi-target compatible designs

Given \( \{R_1, \ldots, R_k\}, n \)

- **Decision**
  Theorem (Flamm et al., 2001)
  \( \mathcal{O}(n) \) algorithm: return bipartite(G)

- **Construct one**
  Theorem (Flamm et al., 2001)
  \( \Theta(n) \) algorithm: alternate G and C along cycles and paths

- **Counting**
  Theorem (Hammer/Wei/Ponty/Will, 2018)
  \( \Rightarrow \) Corollary: Counting designs is \( \#P \)-hard

- **Controlled (uniform, Boltzmann) sampling**
  \( FPT \) algorithm on treewidth (Hammer/Wei/Ponty/Will, 2018)
Uniform sampling for multiple structures

1 2 3 4 5
R1 ( . . ) .
R2 . ( ( ) )
R3 ( ( . ) )

• Complementarity

• Uneven distribution: e.g.
  • first position $A : C : G : U = 4 : 4 : 10 : 10$
  • second position, after selecting $G$,
    $A : G = 4 : 6, \ldots$

$\rightarrow$ counting enables uniform sampling
Counting is \#P-hard

Theorem [HWPW, 2018]: Counting of sequences for multiple targets is \#P-hard.

Proof (sketch):

- \#BIS (Counting bipartite independent sets) is \#P-hard [Ge, Štefankovič, 2012].
- Sequence counting is equivalent to counting independent sets.
Counting is \#P-hard

**Theorem [HWPW, 2018]:** Counting of sequences for multiple targets is \#P-hard.

**Proof (sketch):**
- \#BIS (Counting bipartite independent sets) is \#P-hard [Ge, Štefankovič, 2012].
- Sequence counting is *equivalent* to counting independent sets.
Counting is \#P-hard

Theorem [HWPW, 2018]: Counting of sequences for multiple targets is \#P-hard.

Proof (sketch):
- \#BIS (Counting bipartite independent sets) is \#P-hard [Ge, Štefankovič, 2012].
- Sequence counting is equivalent to counting independent sets.
Theorem [HWPW, 2018]: Counting of sequences for multiple targets is \#P-hard.

Proof (sketch):

- \#BIS (Counting bipartite independent sets) is \#P-hard [Ge, Štefankovič, 2012].
- Sequence counting is equivalent to counting independent sets.
Theorem [HWPW, 2018]: Counting of sequences for multiple targets is \#P-hard.

Proof (sketch):

- \#BIS (Counting bipartite independent sets) is \#P-hard [Ge, Štefankovič, 2012].
- Sequence counting is equivalent to counting independent sets.
Counting is \#P-hard

**Theorem [HWPW, 2018]:** Counting of sequences for multiple targets is \#P-hard.

**Proof (sketch):**
- \#BIS (Counting bipartite independent sets) is \#P-hard [Ge, Štefankovič, 2012].
- Sequence counting is *equivalent* to counting independent sets.
Systematic counting and sampling

*Given:* $k$ structures, length $n$

**Recipe:**

1. Decompose dependency graph
2. Apply **dynamic programming** (CTE$^1$)
3. **Sample** (stochastic backtracking)

$$1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$$

( ( . . ) ) .

. ( ( ( ) ) )

. ( ( . ) ) .

**target structures**

**dependency graph**

**tree decomposition**

---

$^1$CTE = Cluster Tree Elimination (Rina Dechter)
**Systematic counting and sampling**

Given: $k$ structures, length $n$

Recipe:

1. Decompose dependency graph
2. Apply **dynamic programming (CTE)$^1$**
3. **Sample (stochastic backtracking)**

1 2 3 4 5 6 7  
( ( . . ) ) .  
. ( ( ( ) ) )  
. ( ( . ) ) .

target structures

---

$^1$CTE = Cluster Tree Elimination (Rina Dechter)
Systematic counting and sampling

Given: \( k \) structures, length \( n \)

Recipe:

1. Decompose dependency graph
2. Apply dynamic programming (CTE\(^1\))
3. Sample (stochastic backtracking)

---

\(^1\)CTE = Cluster Tree Elimination (Rina Dechter)
Systematic counting and sampling

Given: $k$ structures, length $n$

Recipe:

1. Decompose dependency graph
2. Apply dynamic programming $(CTE^1)$
3. Sample (stochastic backtracking)

1 2 3 4 5 6 7
( ( . . ) ) .
. ( ( ( ) ) )
. ( ( . ) ) .

target structures

dependency graph

tree decomposition

$^1$CTE = Cluster Tree Elimination (Rina Dechter)
Systematic counting and sampling

*Given:* $k$ structures, length $n$

**Recipe:**

1. Decompose dependency graph
2. Apply **dynamic programming (CTE)**
3. **Sample (stochastic backtracking)**

1 2 3 4 5 6 7
( ( . . ) ) .
. ( ( ( ) ) )
. ( ( . ) ) .

**target structures**

dependency graph

```
1
b16
6

2
b26
b27
b36

3
b35
b25

4
b35
b25

5

6

7
```

tree decomposition

```
3

5 3

b35
b35

2

b27
b26
b36
b45

1 6

b16
```

\[\text{Theorem: Counting and sampling is efficient for fixed tree width} \quad O(n^4 k^{4W} + tn^k) \rightarrow \text{can be improved}\]

\[\text{CTE} = \text{Cluster Tree Elimination (Rina Dechter)}\]
Systematic counting and sampling

Given: k structures, length n

Recipe:
1. Decompose dependency graph
2. Apply dynamic programming (CTE\textsuperscript{1})
3. Sample (stochastic backtracking)

1 2 3 4 5 6 7
( ( . . ) ) .
. ( ( ( ) ) )
. ( ( . . ) ) .

target structures

dependency graph
tree decomposition

---
\textsuperscript{1} CTE = Cluster Tree Elimination (Rina Dechter)
Systematic counting and sampling

*Given*: $k$ structures, length $n$

**Recipe:**

1. Decompose dependency graph
2. Apply **dynamic programming (CTE)$^1$**
3. **Sample (stochastic backtracking)**

```
1 2 3 4 5 6 7
( ( . . ) ) .
. ( ( ( ) ) )
. ( ( . ) ) .
target structures
```

dependency graph

```
5 3
c23 2 c2
4 5
b25 b45
c35
c3
c23
c23
c35
c35
c3
c3
sel 5
```

tree decomposition

\[ O(nk^4w + t nk) \rightarrow \text{can be improved} \]

---

$^1$CTE = Cluster Tree Elimination (Rina Dechter)
Systematic counting and sampling

**Given:** \( k \) structures, length \( n \)

**Recipe:**
1. Decompose dependency graph
2. Apply **dynamic programming** (CTE\(^1\))
3. **Sample** (stochastic backtracking)

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
( & ( & . & . & ) & ) & . \\
. & ( & ( & ( & ) & ) & ) \\
. & ( & ( & . & ) & ) & . \\
\end{array}
\]

**target structures**

**dependency graph**

**tree decomposition**

---
\(^1\)CTE = Cluster Tree Elimination (Rina Dechter)
Systematic counting and sampling

Given: $k$ structures, length $n$

Recipe:

1. Decompose dependency graph
2. Apply dynamic programming (CTE\textsuperscript{1})
3. Sample (stochastic backtracking)

1 2 3 4 5 6 7
(( . . ) ) .
. ( ( ( ) ) )
. ( ( . ) ) .

target structures

dependency graph

tree decomposition

\textsuperscript{1}CTE = Cluster Tree Elimination (Rina Dechter)
Systematic counting and sampling

Given: \( k \) structures, length \( n \)

Recipe:
1. Decompose dependency graph
2. Apply dynamic programming (CTE\(^1\))
3. Sample (stochastic backtracking)

\[
1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \\
( \ . \ . \ . \ ) \ . \\
. \ ( \ ( \ ( \ ) \ ) \ ) \\
. \ ( \ . \ . \ ) \ .
\]

target structures

\( \text{CTE} = \text{Cluster Tree Elimination (Rina Dechter)} \)
Systematic counting and sampling

Given: $k$ structures, length $n$

Recipe:
1. Decompose dependency graph
2. Apply dynamic programming (CTE$^1$)
3. Sample (stochastic backtracking)

\begin{equation}
\mathcal{O}(nk4^w + t nk)
\end{equation}

$^1$CTE = Cluster Tree Elimination (Rina Dechter)
Systematic counting and sampling

Given: \( k \) structures, length \( n \)

Recipe:
1. Decompose dependency graph
2. Apply dynamic programming (CTE\(^1\))
3. Sample (stochastic backtracking)

\[
\begin{align*}
1 & \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \\
( & \quad ( \quad . \quad . \quad ) \quad ) \\
. & \quad ( \quad ( \quad ( \quad ) \quad ) \quad ) \\
. & \quad ( \quad ( \quad . \quad ) \quad ) \\
\end{align*}
\]

target structures

Theorem: Counting and sampling is efficient for fixed tree width

\[
\mathcal{O}(n^k 4^w + t n k) \rightarrow \text{can be improved}
\]

\(^1\text{CTE} = \text{Cluster Tree Elimination (Rina Dechter)}\)
From uniform to Boltzmann sampling

**uniform** sampling \(\leftarrow\) counts

**Boltzmann** sampling \(\leftarrow\) partition functions

**Boltzmann sampling:** \( P(S) \propto \prod_\ell \pi^{-F_\ell(S)} \)

Features \( F_\ell \) can express energies as sums over feature contributions

\( \Rightarrow \) complex constraints \( F_\ell(S) = f_\ell^* \)
From uniform to Boltzmann sampling

uniform sampling $\leftarrow$ counts
Boltzmann sampling $\leftarrow$ partition functions

\begin{equation}
\text{Boltzmann sampling: } \quad P(S) \propto \prod_\ell \pi_- F_\ell(S)
\end{equation}

Features $F_\ell$ can express energies as sums over feature contributions
$\Rightarrow$ complex constraints $F_\ell(S) = f_\ell^*$

Energy models

- **Base pair model**
  "like counting" energy = sum of contributions per base pair

- **Stacking model**
  scores stacks (of two consec. bps)
  multi-ary feature contributions

- **and beyond**: full model, p-knots...
Dependency graphs

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

...base pair

A B
D
F
E
C
G
H
I
J

...stacking
Treewidths are typically low

Base pair model

Stacking model

bio-relevant
stress-test
Multi-target design to three RNA structures

Energy [kcal/mol]

Frequency

Boltzmann sample: 1000 low energy sequences; generated in seconds
Targeted samples: 1000 highly specific sequences; in minutes
Multi-target design to three RNA structures

Boltzmann sample: 1000 low energy sequences; generated in seconds
The positive RNA design problem

Problem

**IN:** structures $\mathcal{R}$, length $n$, $d$ features $F_1, \cdots, F_d$ and objective values $f^*_1, \cdots, f^*_d$

**OUT:** $t$ uniform random sequences $S$, compatible w/ $\mathcal{R}$, s.t.

$$\forall 1 \leq \ell \leq d : F_{\ell}(S) = f^*_\ell.$$
The positive RNA design problem

Problem

**IN:** structures $\mathcal{R}$, length $n$, $d$ features $F_1, \cdots, F_d$ and objective values $f_1^*, \cdots, f_d^*$

**OUT:** $t$ uniform random sequences $S$, compatible w/ $\mathcal{R}$, s.t.

$$\forall 1 \leq \ell \leq d : F_\ell(S) = f_\ell^*.$$ 

Method (Multi-dim. Boltzmann sampling)

- Choose initial weights $\pi_1, \ldots, \pi_d$
- Sample from Boltzmann-distribution, s.t. $Pr(S) \propto \prod_{\ell} \pi_{\ell}^{-F_\ell(S)}$
- Output samples that meet objective values
- Estimate feature means and adapt weights; iterate
Why multi-dim. Boltzmann sampling?

Problem

**IN:** structures \( \mathcal{R} \), length \( n \), \( d \) features \( F_1, \cdots, F_d \);

objective values \( f_1^*, \cdots, f_d^* \); and tolerance \( \varepsilon > 0 \)

**OUT:** \( t \) random sequences \( S \), compatible w/ \( \mathcal{R} \), s.t.

\[ \forall 1 \leq \ell \leq d : F_\ell(S) \in [f_\ell^* \cdot (1 - \varepsilon), f_\ell^* \cdot (1 + \varepsilon)] \]

Possible approaches:

- **Multi-dim. Boltzmann sampling (+ rejection step)**

- **Classified Dynamic Programming**
Why multi-dim. Boltzmann sampling?

Problem

IN: structures $\mathcal{R}$, length $n$, $d$ features $F_1, \cdots, F_d$; 
objective values $f_1^*, \cdots, f_d^*$; and tolerance $\varepsilon > 0$

OUT: $t$ random sequences $S$, compatible w/ $\mathcal{R}$, s.t.

$$\forall 1 \leq \ell \leq d : F_\ell(S) \in [f_\ell^* \cdot (1 - \varepsilon), f_\ell^* \cdot (1 + \varepsilon)]$$

Possible approaches:

- **Multi-dim. Boltzmann sampling (+ rejection step)**
  works well b/c distributions are typically concentrated
  - expect $O(1)$ rejections for $\varepsilon > 1/\sqrt{n}$,
  - $\Theta(n^{d/2})$ for $\varepsilon = 0$ [Bender et al., 1983; Drmota, 1997].

- **Classified Dynamic Programming**
Why multi-dim. Boltzmann sampling?

Problem

**IN:** structures $\mathcal{R}$, length $n$, $d$ features $F_1, \cdots, F_d$;
objective values $f_1^*, \cdots, f_d^*$; and tolerance $\varepsilon > 0$

**OUT:** $t$ random sequences $S$, compatible w/ $\mathcal{R}$, s.t.

$$\forall 1 \leq \ell \leq d : F_\ell(S) \in [f_\ell^* \cdot (1 - \varepsilon), f_\ell^* \cdot (1 + \varepsilon)]$$

Possible approaches:

- **Multi-dim. Boltzmann sampling (+ rejection step)**
  works well b/c distributions are typically concentrated
  - expect $O(1)$ rejections for $\varepsilon > 1/\sqrt{n}$,
  - $\Theta(n^{d/2})$ for $\varepsilon = 0$ \[Bender et al., 1983; Drmota, 1997\].

- **Classified Dynamic Programming**
  - convolution: $\times \Theta(n^{2d})$ time / $\Theta(n^d)$ space \[Cupal et al., 1996\]
  - using DFT to avoid convolution allows more efficient uniform
  sampling over range (case $\varepsilon > 0$) \[cf. Senter et al., 2012\]
Multi-target design to three RNA structures

Boltzmann sample: 1000 low energy sequences; generated in seconds
Multi-target design to three RNA structures

**Boltzmann sample**: 1000 low energy sequences; generated in seconds

**Targeted samples**: 1000 highly specific sequences; in minutes
Multi-target design to three RNA structures

**Boltzmann sample**: 1000 low energy sequences; generated in seconds

**Targeted samples**: 1000 highly specific sequences; in minutes
Multi-target design to three RNA structures

Boltzmann sample: 1000 low energy sequences; generated in seconds
Targeted samples: 1000 highly specific sequences; in minutes
Boltzmann outperforms uniform sampling for negative multi-target RNA design

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RedPrint</th>
<th>Uniform</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seeds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2str</td>
<td>21.67 (±4.38)</td>
<td>37.74 (±6.45)</td>
<td>73%</td>
</tr>
<tr>
<td>3str</td>
<td>18.09 (±3.98)</td>
<td>30.49 (±5.41)</td>
<td>71%</td>
</tr>
<tr>
<td>4str</td>
<td>19.94 (±3.84)</td>
<td>32.29 (±5.24)</td>
<td>63%</td>
</tr>
<tr>
<td>Optimized</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2str</td>
<td>5.84 (±1.31)</td>
<td>7.95 (±1.76)</td>
<td>28%</td>
</tr>
<tr>
<td>3str</td>
<td>5.08 (±1.10)</td>
<td>7.04 (±1.52)</td>
<td>31%</td>
</tr>
<tr>
<td>4str</td>
<td>8.77 (±1.48)</td>
<td>13.13 (±2.13)</td>
<td>37%</td>
</tr>
</tbody>
</table>

Multi-target design objective [Blueprint] on the Modena benchmark

https://github.com/yannponty/RNARedPrint

Complex sequence constraints

**Task:** forbid a set $\mathcal{W}$ of subwords of length $\leq k$

**Naïve:** add $k$-ary constraints for each $k$ successive sequence positions

**Proposed:**
- construct Aho-Corasick automaton (states $Q$)
- extend alphabet from $\Sigma$ to $Q \times \Sigma$
- restrict consecutive positions to transitions of the automaton (adds Hamiltonian path of binary constraints)
- new complexity $O(n \cdot |R| \cdot (|\Sigma| \cdot |Q|)^{w'+1})$; new tree width $w'$ (!)

**Similarly:** enforce subwords

$\textit{transfers ideas of}$ [Zhou et al, 2013]
• Satisfies multiple constraints and targets multiple complex properties; **Improves quality and feasibility of RNA design**
  complex constraints by multi-dimensional Boltzmann sampling
• Based on Constraint Networks and Tree Decomposition/CTE: 
  **Generic system to extend RNA design . . .**
  **. . . and develop novel sampling-based tools**
• **Theorems:** counting is \#P-hard; Boltzmann-sampling is FPT
• **Perspectives and Open Questions:**
  • effect on tree-width of complex constraints like forbidding motifs?
    (e.g. this adds hamiltonian path of dependencies)
  • how to (better) ensure uniformity within range of feature values?
  • complexity of generation, stronger complexity bounds?
  • how to extend towards FPT negative design?

Co-authors

Stefan Hammer  Yann Ponty  Wei Wang

Team

TBI (Ivo Hofacker) at
universität wien

Funding

ANR  FWF  SEVENTH FRAMEWORK PROGRAMME