Versatile fixed-parameter tractable sampling for multi-target RNA design

Sebastian Will

Bled 2018

TBI · University of Vienna

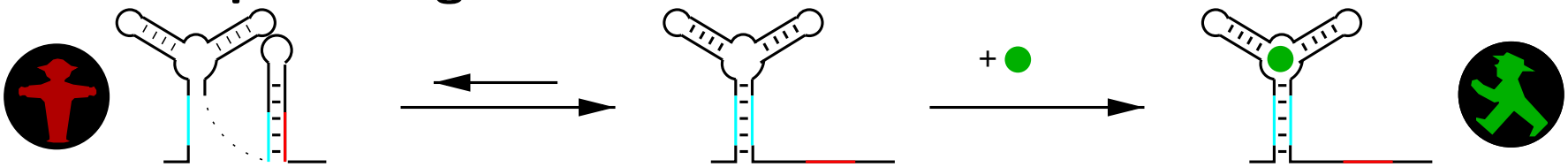
Fixed-Parameter Tractable Sampling for RNA Design with Multiple Target Structures

Stefan Hammer[1,2,3], Yann Ponty[4,5,⋆], Wei Wang[4,5], and Sebastian Will[2]

[1] University Leipzig, Department of Computer Science and Interdisciplinary Center

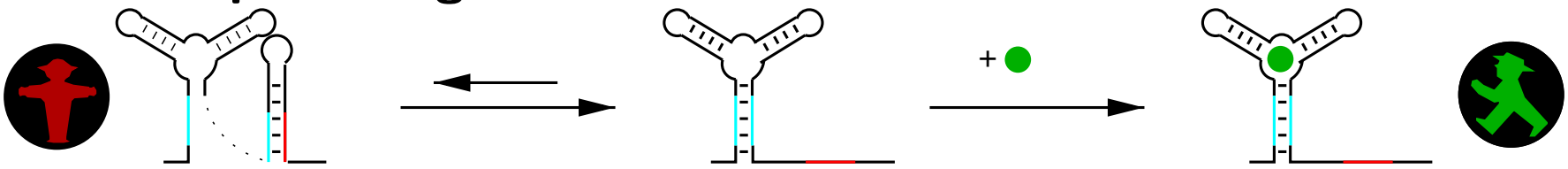
Multi-target design of RNA sequences

For example: design riboswitches for translational control

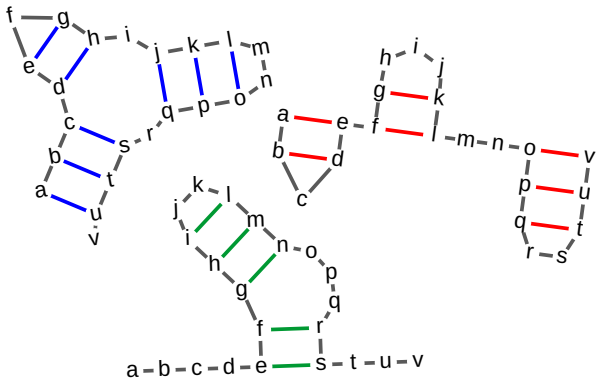


Multi-target design of RNA sequences

For example: design riboswitches for translational control



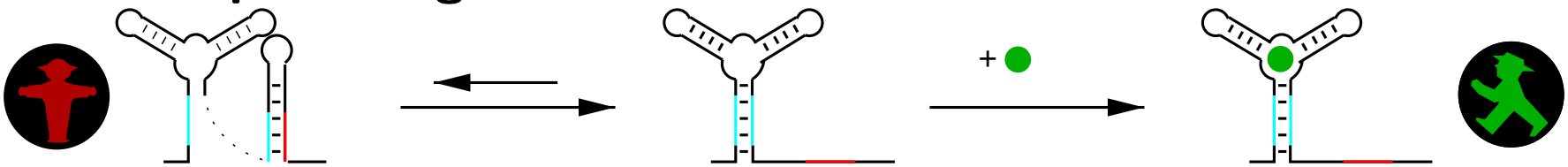
Multiple structures (=*multiple design targets*)



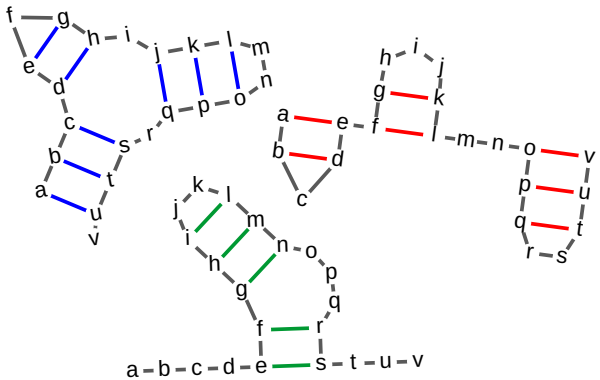
```
abcdefghijklmnopqrstuv
((((((.)).(((..))).))).
((.))((...))..(((..)))
....((((((..)))...)))...
```

Multi-target design of RNA sequences

For example: design riboswitches for translational control



Multiple structures (=*multiple design targets*)



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

Task: generate sequences with *specific properties*

- low/specific energy for multiple structures
- specific CG content
- specific energy differences
- specific sequence/structure motifs (enforce/forbid)

Approach:
(defined) sampling

Uniform sampling: multiple structures

1	2	3	4	5
(	.	.	)	.
.	(	(	)	)
(	(	.	)	)
A	**A**	**A**	**U**	**U**

Uniform sampling: multiple structures

1	2	3	4	5
(	.	.	)	.
.	(	(	)	)
(	(	.	)	)
A	**A**	**A**	**U**	**U**
A	**A**	**G**	**U**	**U**
A	**G**	**A**	**U**	**U**
A	**G**	**G**	**U**	**U**
G	**A**	**A**	**U**	**C**
G	**A**	**A**	**U**	**U**
G	**A**	**G**	**U**	**C**
G	**A**	**G**	**U**	**U**
G	**G**	**A**	**U**	**C**
G	**G**	**A**	**U**	**U**
G	**G**	**G**	**C**	**C**
G	**G**	**G**	**C**	**U**
G	**G**	**G**	**U**	**C**
G	**G**	**G**	**U**	**U**
⋮				

- For uniform: choose first position
  A : **C** : **G** : **U** $= 4 : 4 : 10 : 10$
  Then, e.g. after **G**, choose second
  A : **G** $= 4 : 6, \ldots$
- $\rightarrow$ **counting**
- (Why) is this hard?

Uniform sampling: multiple structures

1	2	3	4	5
(	.	.	)	.
.	(	(	)	)
(	(	.	)	)
A	A	A	U	U
A	A	G	U	U
A	G	A	U	U
A	G	G	U	U
G	A	A	U	C
G	A	A	U	U
G	A	G	U	C
G	A	G	U	U
G	G	A	U	C
G	G	A	U	U
G	G	G	C	C
G	G	G	C	U
G	G	G	U	C
G	G	G	U	U
⋮				

- For uniform: choose first position
  A : **C** : **G** : **U** $= 4 : 4 : 10 : 10$
  Then, e.g. after **G**, choose second
  A : **G** $= 4 : 6, \ldots$
- $\rightarrow$ **counting**
- (Why) is this hard?

Theorem: Counting of sequences for multiple targets is #P-hard.

Proof: equiv. to counting independent sets

1. dependency graph is bipartite

$$\{A, G\} \quad \text{vs.} \quad \{C, U\}$$

2. A and C cannot pair: independent
3. Selecting all As and Cs, i.e. independent sets, determines a design (and vice versa)

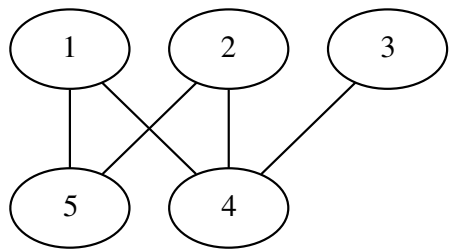
Systematic efficient counting (and sampling)

Recipe:

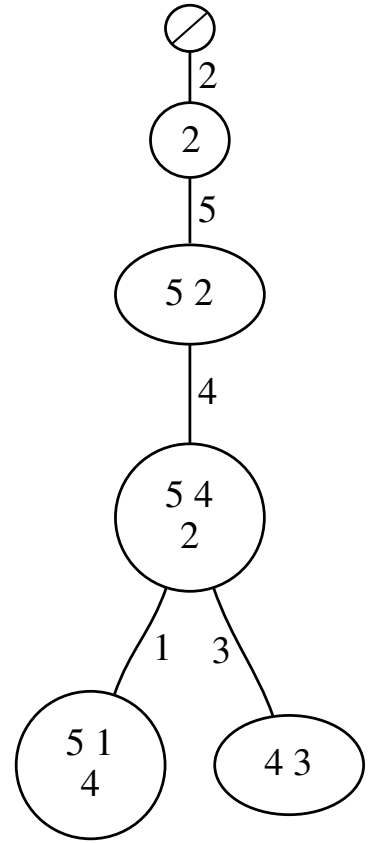
1. Decompose dependency graph
2. Apply **dynamic programming**
3. Sample

```
1  2  3  4  5
(  .  .  )  .
.  (  (  )  )
(  (  .  )  )
```

target structures



dependency graph



tree decomposition

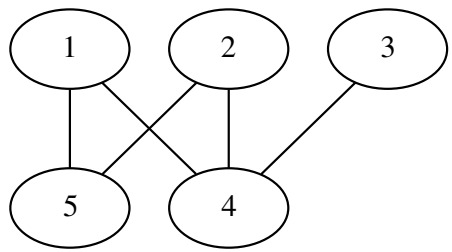
Systematic efficient counting (and sampling)

Recipe:

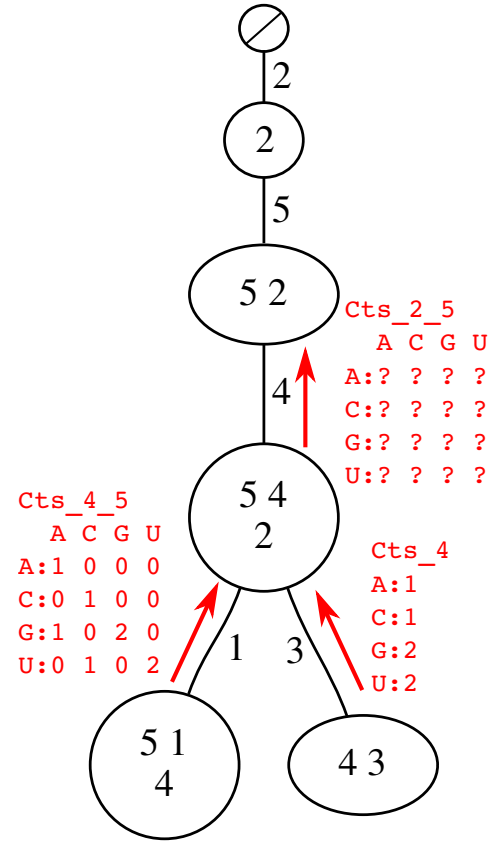
1. Decompose dependency graph
2. Apply **dynamic programming**
3. Sample

```
1  2  3  4  5
(  .  .  )  .
.  (  (  )  )
(  (  .  )  )
```

target structures



dependency graph



tree decomposition

Theorem: Counting and sampling efficient for fixed tree width.

(So far) Blueprint / RNAdesign *similar*, but *ear decomposition*

[Blueprint] Hammer, Tschiatschek, Flamm, Hofacker, Findeiß. *Bioinformatics*, 2017.
[RNAdesign] Hoener, Hammer, Abfalter, Hofacker, Flamm, Stadler, *Biopolymers*, 2013.

From uniform sampling to Boltzmann sampling

- *counts* for all subtrees $\longrightarrow$ *uniform* sampling
- Analogously: *partition functions* $\longrightarrow$ *Boltzmann* sampling

$$\textit{Boltzmann sampling}\text{: } \mathrm{P}(S) \sim \exp(-\beta E(S)).$$

From uniform sampling to Boltzmann sampling

- *counts* for all subtrees $\longrightarrow$ *uniform* sampling
- Analogously: *partition functions* $\longrightarrow$ *Boltzmann* sampling

$$\textit{Boltzmann sampling}\text{: } \mathrm{P}(S) \sim \exp(-\beta E(S)).$$

- Energy = (weighted) sum of energies for single structures

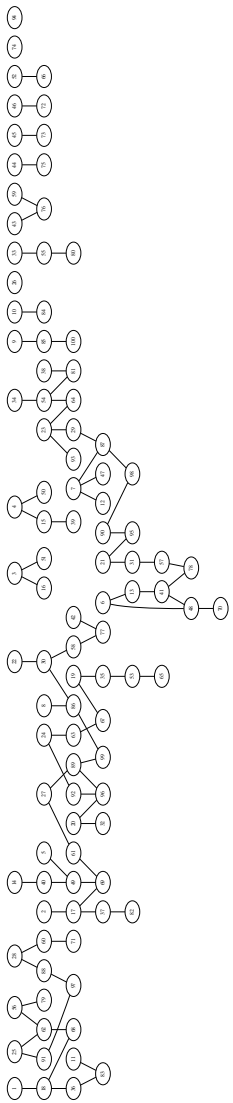
- **energy models**
  - **Base pair model**
    "like counting"
  - **Nearest neighbor model (Turner model)**
    requires multi-ary dependencies: **constraint framework***
  - **Stacking model**
    "in-between", score stacks (4-cliques of dependencies)

* Constraint networks / cluster tree elimination [Rina Dechter]

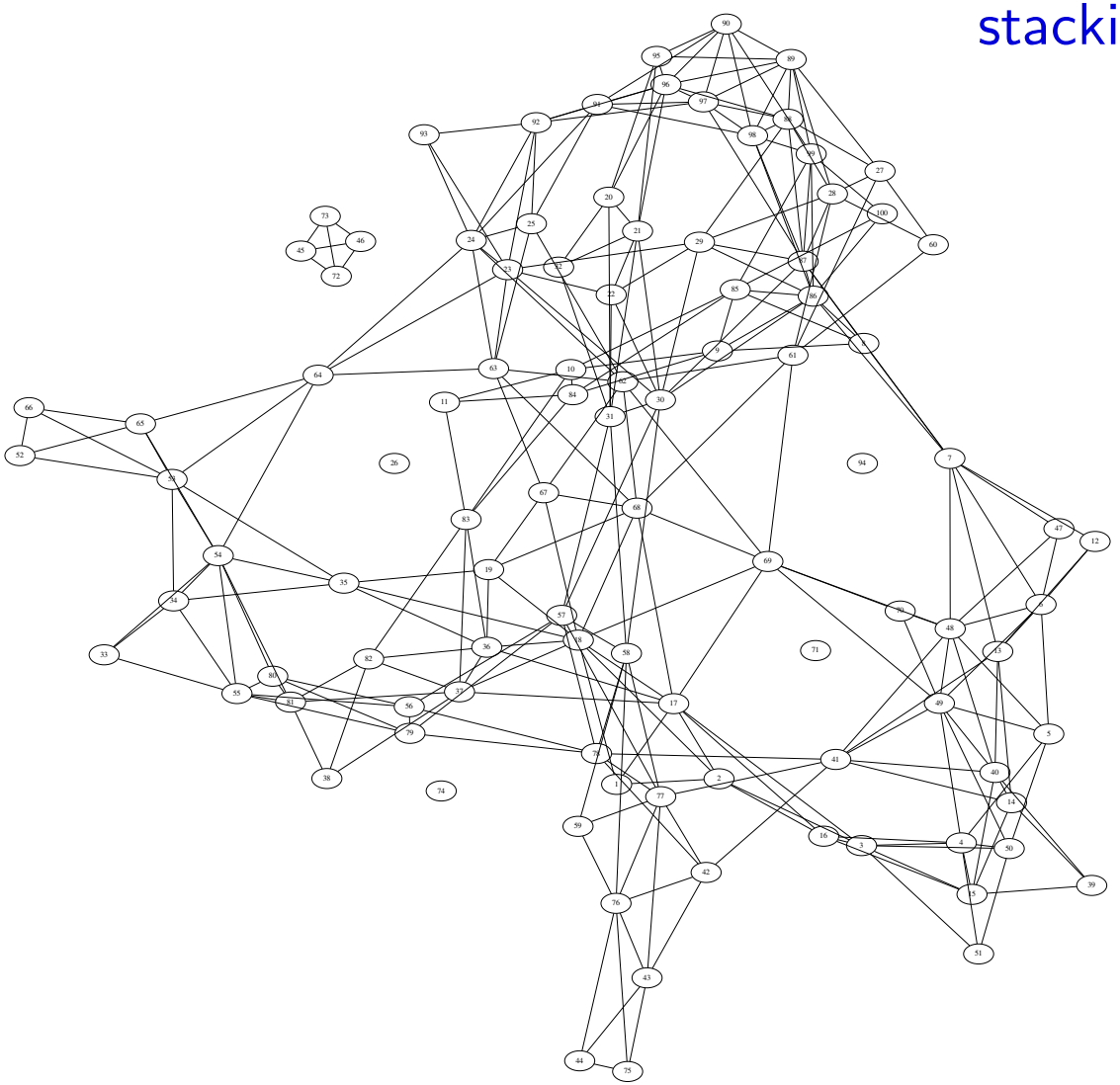
Example dependency graphs

```
((((.((....)).)))).((.(((.((((.....(((..((((((.((..(((.(.....).)))..)).)).)))).))))..))))..))))
..(((((.....(((.((((((.....)))).))).))).....)))))..(((((((((...))).).....))))))...((((((....))))))
......((((.....(((...(((.((.((.(((....((......))...))).)).)))))..)))............)))))..((((...)))).
```

base pair



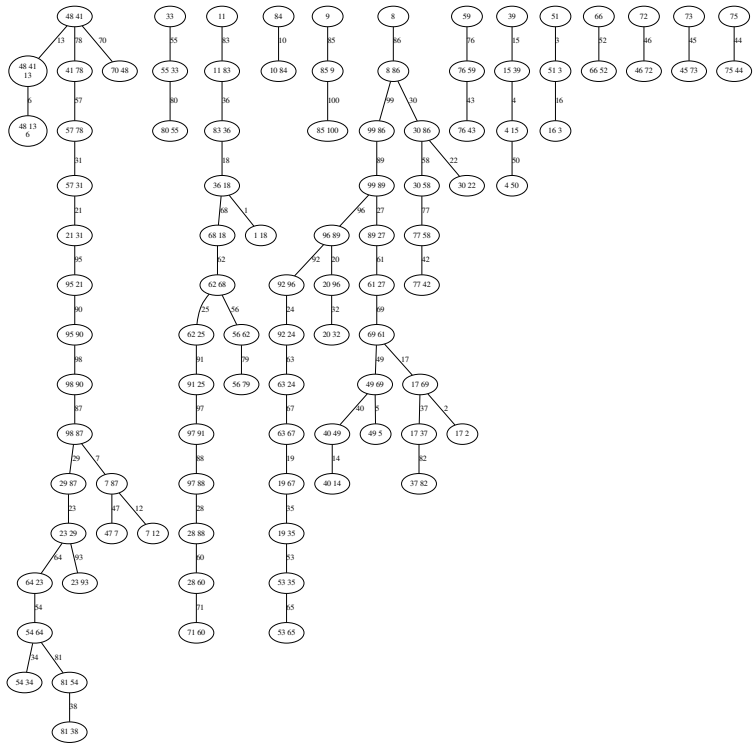
stacking



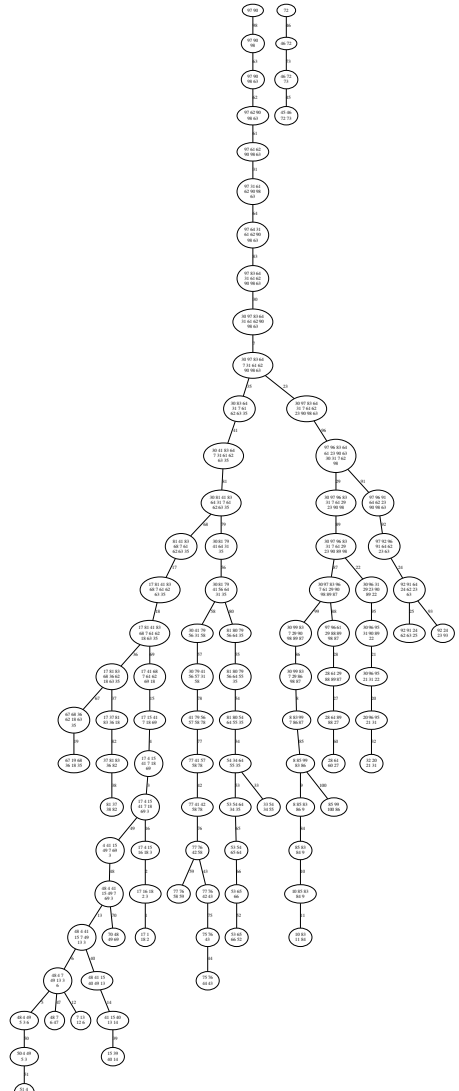
Example tree decompositions

```
((((.((....)).)))).((.(((.((((.....(((..((((((.((..(((.(.....).)))..)).)).)))).))).))).))).)).....
..(((((.....(((.(((((((.....))))..))).))).....)))))..(((((((((...))).).....))))))...((((((....))))))
......((((.....(((...(((.((.((.(((....((......))...))).)).)))))..)))............)))))..((((...)))).
```

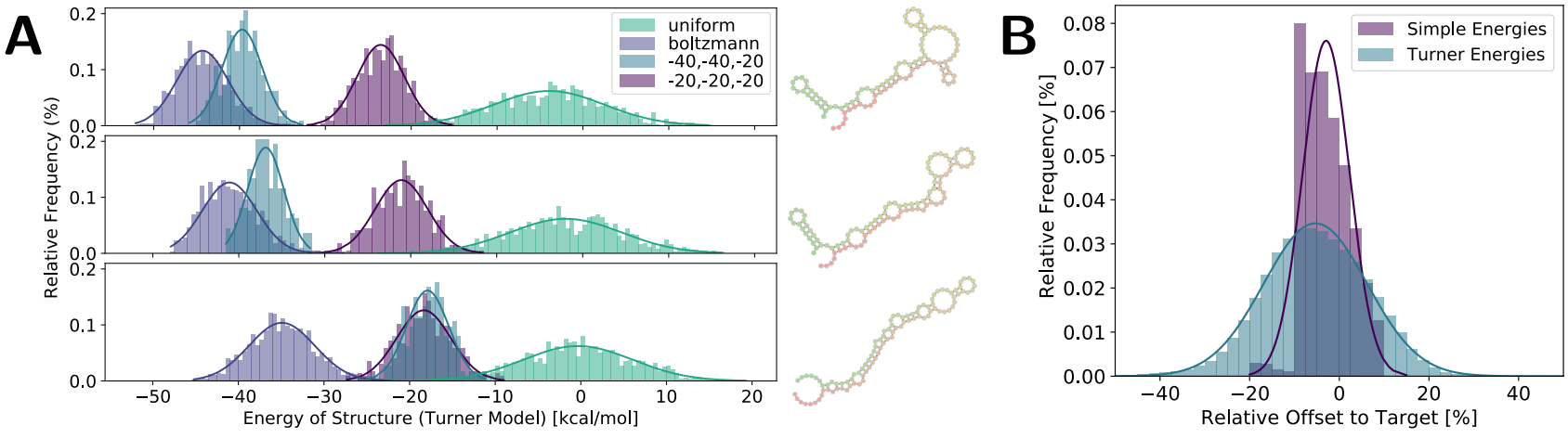
base pair



stacking



Targeting specific properties: multi-dimensional Boltzmann sampling



Weight and combine single structure energies and features ("GC content")

A **Learn weights** by adaptive scheme

$\rightarrow$ target specific energies and GC content

B **Sampling**: targets Turner energies by linear fitting of energies

Boltzmann vs. uniform sampling for multi-target RNA design

	Dataset	Redprint	Uniform	Improvement
Seeds	2str	21.67 (±4.38)	37.74 (±6.45)	73%
	3str	18.09 (±3.98)	30.49 (±5.41)	71%
	4str	19.94 (±3.84)	32.29 (±5.24)	63%
Optimized	2str	5.84 (±1.31)	7.95 (±1.76)	28%
	3str	5.08 (±1.10)	7.04 (±1.52)	31%
	4str	8.77(±1.48)	13.13 (±2.13)	37%

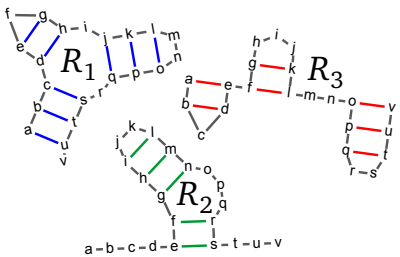
Multi-target design objective [Blueprint] on the **Modena benchmark**

Modena benchmark: Taneda. *BMC Bioinformatics*, 2015.

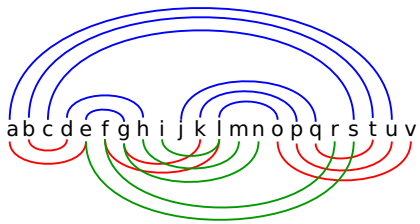
`2str` = "`rnatabupath`"

Thank you!

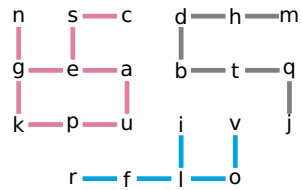
Questions? $\Rightarrow$ me,  , and/or 



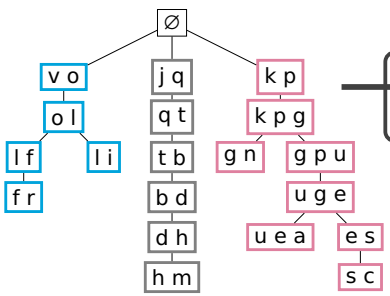
i) Input Structures



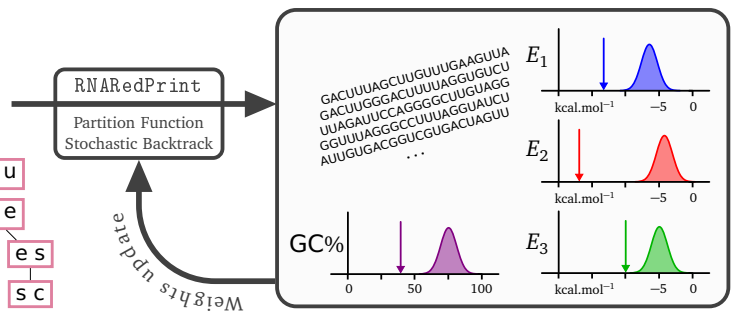
ii) Merged Base-Pairs



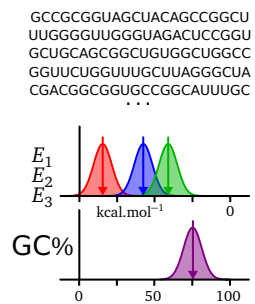
iii) Compatibility Graph



iv) Tree Decomposition



v) Weight Optimization (Adaptive Sampling)



vi) Final Designs

(workflow shown for base pair energy model; for more sophisticated models and applications, we introduced *n*-ary dependencies in a flexible constraint framework)

APPENDIX

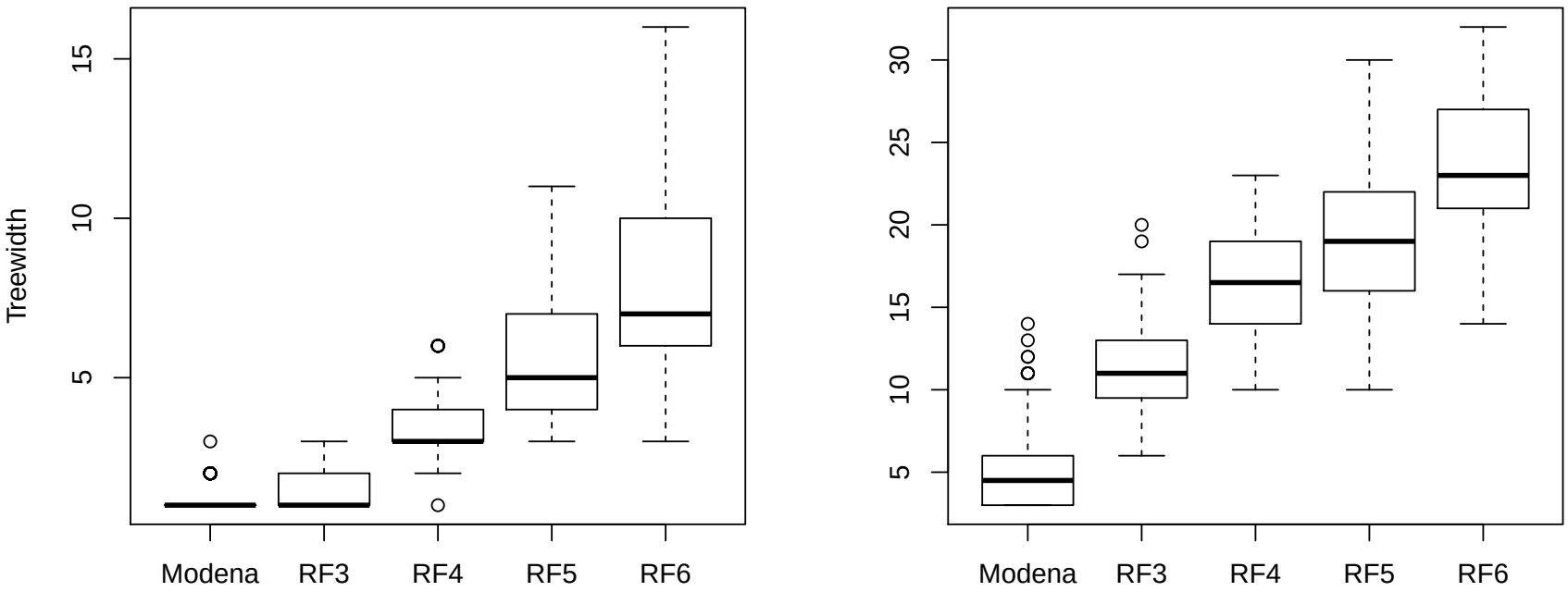
Uniform sampling: one single structure

1	2	3	4
.	(	.	)
A	**A**	**A**	**U**
A	**C**	**A**	**G**
A	**G**	**A**	**C**
A	**G**	**A**	**U**
A	**G**	**A**	**U**
	⋮		
U	**U**	**U**	**A**
U	**U**	**U**	**G**

$\rightarrow 4 \cdot 6 \cdot 4 = 96$

- draw unpaired bases uniformly
- choose 1st end of each base pair, s.t. **A** : **C** : **G** : **U** $= 1 : 1 : 2 : 2$
- select 2nd end accordingly (if first is **G** or **U**: choose)

Tree widths over benchmark sets



Base pair model vs. stacking model