Combinatorial insight into RNA structure

Analysis and Design of *ab-initio* RNA algorithms

Yann Ponty

Bioinformatics Team
École Polytechnique/CNRS/INRIA AMIB – France

April 24, 2011

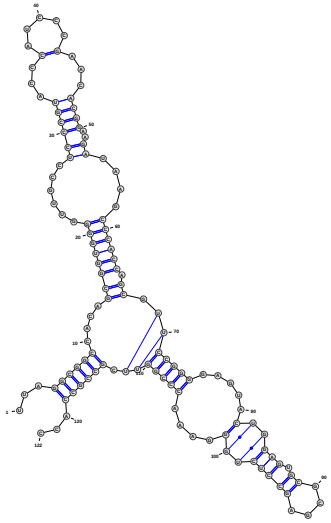
Outline

Part I

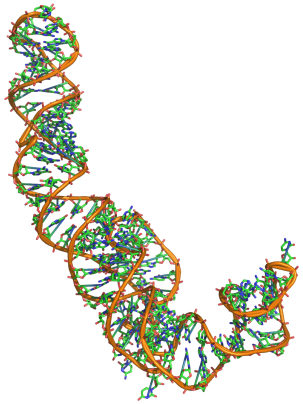
Foreword/Generating functions 101

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC
```

Primary structure



Secondary structure



Tertiary structure

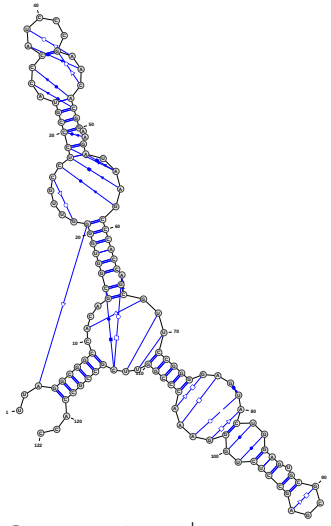
Source: 5s rRNA (PDBID: 1K73:B)

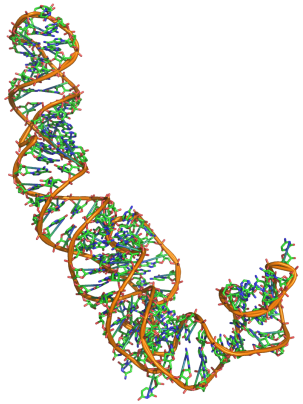
Definition

Secondary structures of RNA =
Maximal non-crossing subset of canonical base-pairs.

RNA structure

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC
```





Primary structure | Secondary$^+$ structure | Tertiary structure

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

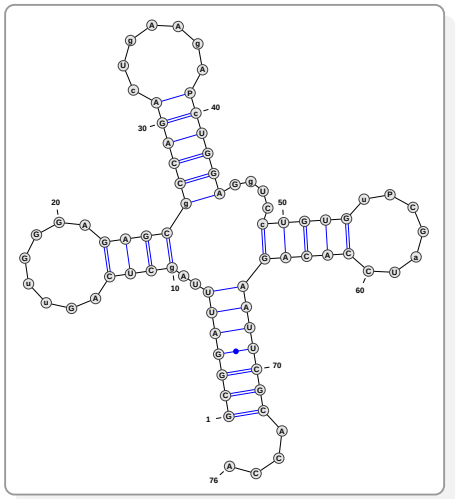
Definition

Secondary structures of RNA =
Maximal non-crossing subset of canonical base-pairs.

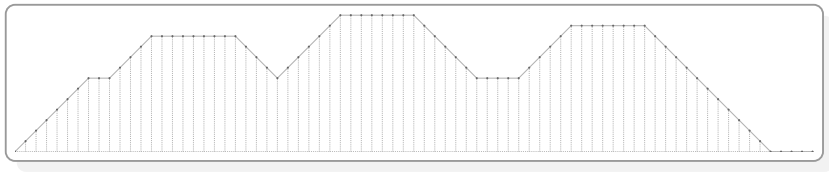
Various representations for a versatile molecule

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

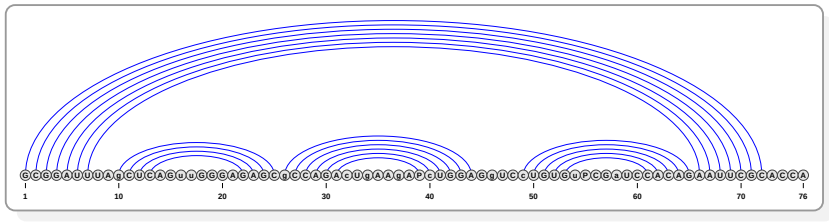
Well-parenthesized expression



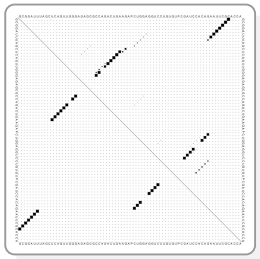
Outer planar graph



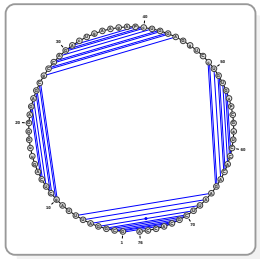
Mountain view



Non intersecting arcs



Dot plot



Feynman's diagram

Different objects
yet
Common combinatorial structure

Minimum Free-Energy (MFE)

Functional folding of an RNA = Minimum free-energy structure.

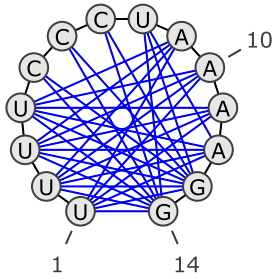
Nussinov/Jacobson (NJ) energy model: Simplest *nearest-neighbor* model.

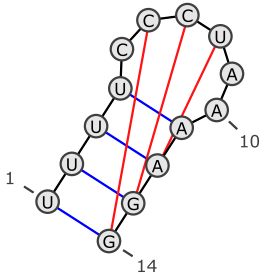
$$\text{Free-energy} = -\#\text{Base-pairs}$$

Folding in NJ $\Leftrightarrow$ Maximizing number of base-pairs.
($\Leftrightarrow$ Max independent set in circle graphs $\Leftrightarrow$ weighted CYK . . .)

Example:

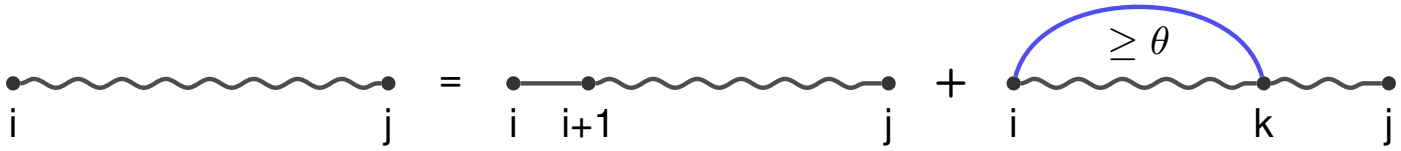
UUUUCCCUAAAAGG





Alt. : Assign weight to each base-pairs based on number of hydrogen bonds
$\Delta G(G{\equiv}C) = -3$ $\Delta G(A{=}U) = -2$ $\Delta G(G{-}U) = -1$

Minimization in Nussinov/Jacobson



Recurrence on the Minimal Free-Energy $N_{i,j}$ of any subinterval of an RNA :

$$
\begin{array}{rcl}
N_{i,t} & = & 0, \quad \forall t \in [i, i+\theta] \\
N_{i,j} & = & \min \begin{cases} N_{i+1,j} & (i \text{ unpaired}) \\ \min_{k=i+\theta+1}^{j} E_{i,j} + N_{i+1,k-1} + N_{k+1,j} & (i \text{ comp. with } k) \end{cases}
\end{array}
$$

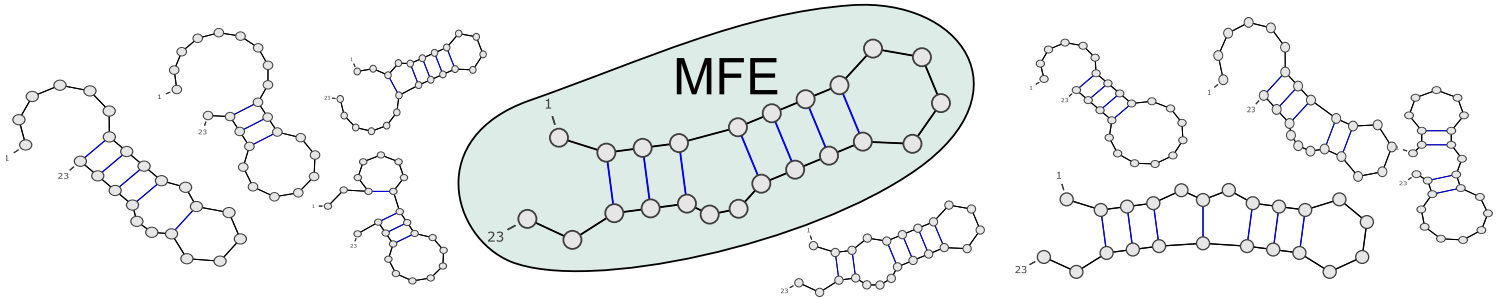
After computing N, the choices elected by min allow to rebuild (backtrack over) the MFE structure.

$\Rightarrow$ Dynamic programming for folding [NJ80].

Remark: Overly simplistic, but more complex energy models will typically already predict 73% of experimentally-determined base-pairs...

2004– : Boltzmann-ensemble paradigm

RNA *breathes* ⇒ There is not necessarily one functional conformation!



Boltzmann-ensemble paradigm

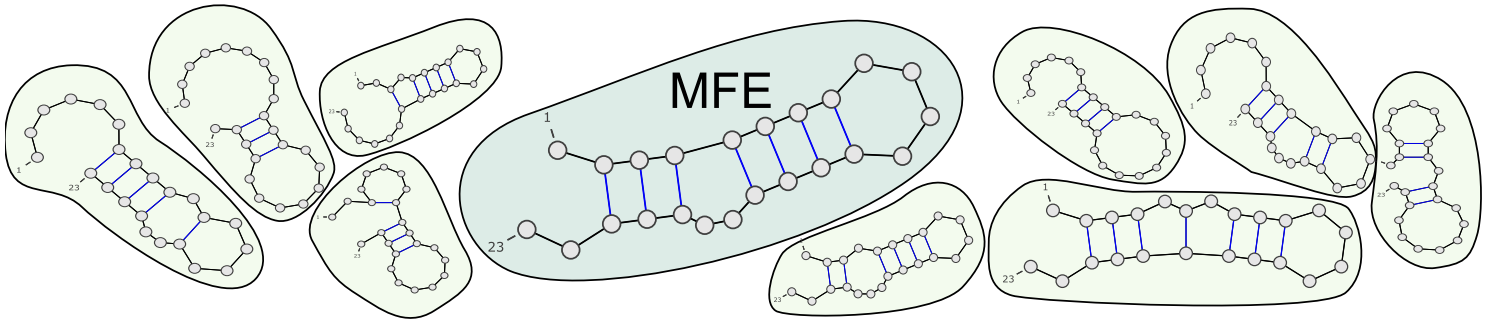
All possible secondary structures for an RNA coexist in a probability distribution, the Boltzmann distribution.

Bonus: MFE structure probability can be significantly remote, pointing toward unstructured RNAs.
Conversely, base-pairs having probability $> .99$ are valid 95% of the time.

- ⇒ Similar structures can bundle up.
- ⇒ Functional structures must be sought in sub-optimal structures.

2004– : Boltzmann-ensemble paradigm

RNA *breathes* ⇒ There is not necessarily one functional conformation!



Boltzmann-ensemble paradigm

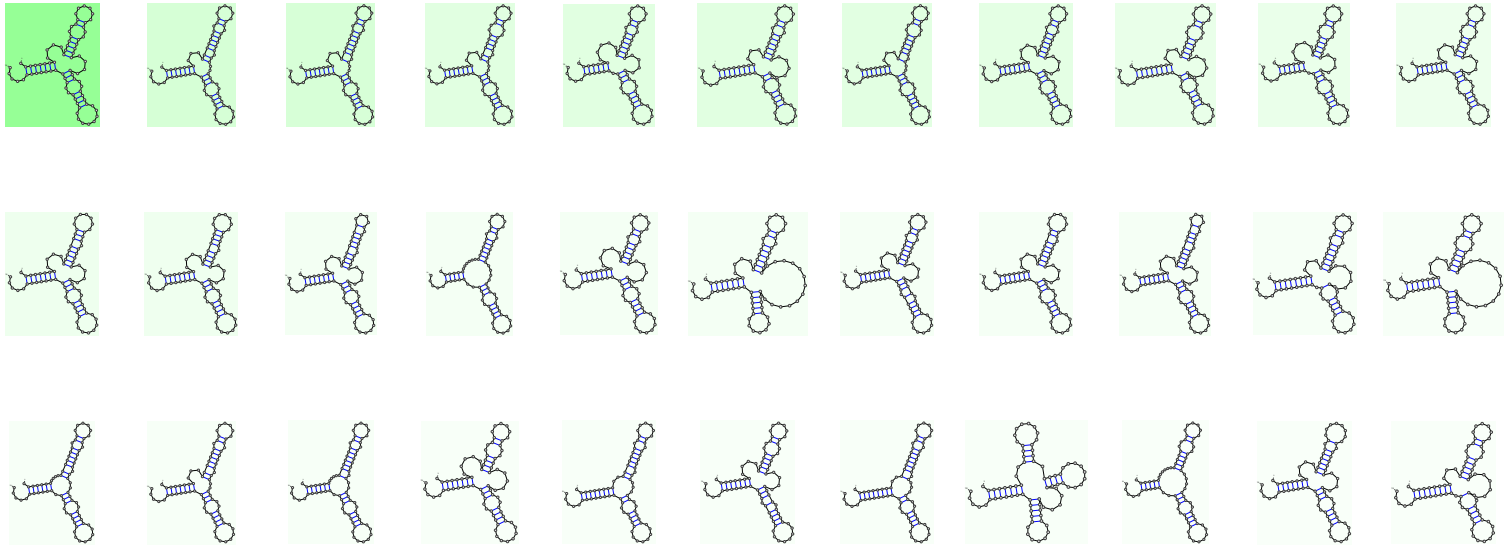
All possible secondary structures for an RNA coexist in a probability distribution, the Boltzmann distribution.

Bonus: MFE structure probability can be significantly remote, pointing toward unstructured RNAs.
Conversely, base-pairs having probability $> .99$ are valid 95% of the time.

- ⇒ Similar structures can bundle up.
- ⇒ Functional structures must be sought in sub-optimal structures.

Why use combinatorics?

Boltzmann ensemble is a (weighted) combinatorial class.



Studying it as such *cleans out the details* and helps:

- Access asymptotic properties of secondary structures
- Investigate worst and average-case complexities
- Design better algorithms and methods for RNA

Generating functions

Let $|\,.\,|$ be a size function over objects (Sequences, trees, ...).
Combinatorial classes are (infinite) sets $\mathcal{C}$ of objects whose restrictions $\mathcal{C}_n$ to objects of size n are of finite cardinality.

Definition (Generating functions)

Let $\mathcal{C}$ be a combinatorial class and $c_n = |\mathcal{C}_n|$ the number of objects of size n in $\mathcal{C}$, then the generating function of $\mathcal{C}$ is $C(z)$ s. t.

$$C(z) = \sum_{s \in C} z^{|s|} = \sum_{n \geq 0} c_n z^n$$

Closed forms for $C(z)$ are often easy to find ...

DNA example: $\mathcal{D} := \{a, c, g, t\}^* \Rightarrow d_n = 4^n$
and $C(z) = 1 + 4z + 16z^2 + 64z^3 + \ldots = \sum_{n \geq 0} 4^n z^n = \frac{1}{1-4z}$

...and very often much simpler than for c_n!!!

DSV/symbolic method

From a class specification, one can directly establish the gen. fun.

Historically on languages, from Schützenberger's observation that

Gen. fun. are commutative images of languages

Grammar	Generating function	Coefficients
$C \to \varepsilon$	$C(z) = z^0 = 1$	$c_n = \mathbb{1}_{\{0\}}(n)$
$C \to t$	$C(z) = z^1 = z$	$c_n = \mathbb{1}_{\{1\}}(n)$
$C \to A \mid B$	$C(z) = A(z) + B(z)$	$c_n = a_n + b_n$
$C \to A.B$	$C(z) = A(z) \cdot B(z)$	$c_n = \sum_{k=0}^{n} a_k b_{n-k}$

Remark: One needs to ensure that unions are disjoint and concatenations unambiguous.

DNA example : $\{a, c, g, t\}^* \Leftrightarrow D \to a.D \mid c.D \mid g.D \mid t.D \mid \varepsilon$

$$\Rightarrow D(z) = z \cdot D(z) + z \cdot D(z) + z \cdot D(z) + z \cdot D(z) + 1$$

DSV/symbolic method

From a class specification, one can directly establish the gen. fun.

Historically on languages, from Schützenberger's observation that

Gen. fun. are commutative images of languages

Grammar	Generating function	Coefficients
$C \to \varepsilon$	$C(z) = z^0 = 1$	$c_n = \mathbb{1}_{\{0\}}(n)$
$C \to t$	$C(z) = z^1 = z$	$c_n = \mathbb{1}_{\{1\}}(n)$
$C \to A \mid B$	$C(z) = A(z) + B(z)$	$c_n = a_n + b_n$
$C \to A.B$	$C(z) = A(z) \cdot B(z)$	$c_n = \sum_{k=0}^{n} a_k b_{n-k}$

Remark: One needs to ensure that unions are disjoint and concatenations unambiguous.

DNA example : $\{a, c, g, t\}^* \Leftrightarrow D \to a.D \mid c.D \mid g.D \mid t.D \mid \varepsilon$

$$\Rightarrow D(z) = 4z \cdot D(z) + 1$$

DSV/symbolic method

From a class specification, one can directly establish the gen. fun.

Historically on languages, from Schützenberger's observation that

Gen. fun. are commutative images of languages

Grammar	Generating function	Coefficients
$C \to \varepsilon$	$C(z) = z^0 = 1$	$c_n = \mathbb{1}_{\{0\}}(n)$
$C \to t$	$C(z) = z^1 = z$	$c_n = \mathbb{1}_{\{1\}}(n)$
$C \to A \mid B$	$C(z) = A(z) + B(z)$	$c_n = a_n + b_n$
$C \to A.B$	$C(z) = A(z) \cdot B(z)$	$c_n = \sum_{k=0}^{n} a_k b_{n-k}$

Remark: One needs to ensure that unions are disjoint and concatenations unambiguous.

DNA example : $\{a, c, g, t\}^* \Leftrightarrow D \to a.D \mid c.D \mid g.D \mid t.D \mid \varepsilon$

$$\Rightarrow D(z) = \frac{1}{1 - 4z}$$

Disclaimer

What follows, although true in this context, is embarassingly simplistic.
A rigorous presentation can (and must) be found in Flaj./Sedg. 08.

A singularity is a point $z = \rho$ where $C(z)$ is no longer analytic.
Asymptotics of coeff c_n are driven by the singularities of $C(z)$.

1st principle

Location of the dominant (smallest) singularity ρ dictates the exponential growth $\Rightarrow \frac{c_n}{\rho^{-n}} = o(\alpha^n),\ \forall \alpha > 1$.

DNA example: $D(z) = 1/(1-4z) \Rightarrow \rho = 1/4 \Rightarrow d_n \sim 4^n P(n)$.

Basic scale

2nd principle

Nature of ρ dictates subexponential part $P(n)$ s.t. $c_n \sim \rho^{-n} P(n)$.

Basic scale: If one can rewrite $C(z)$ as

$$C(z) = f(z) + g(z)(1 - z/\rho)^{\alpha}$$

where f and g are analytic $\forall |z| < |\rho|$ and non-null at ρ, then

$$c_n \equiv [z^n]C(z) \sim \frac{g(\rho)\rho^{-n}}{\Gamma(-\alpha)n^{\alpha+1}}$$

Example: $D(z) = \frac{1}{1-4z} \Rightarrow c_n \sim 4^n$
$(\rho = 1/4, \alpha = -1, f(z) = 0,$ and $g(z) = 1)$

General methodology

Asymptotic estimates are obtained using a 4 steps meta-algorithm:

1. Find the right model
2. Translate into grammar
3. Translate into system and solve g. f.
4. Singularity analysis yields asymptotics

General methodology

Asymptotic estimates are obtained using a 4 steps meta-algorithm:

1. Find the right model
2. Translate into grammar
3. Translate into system and solve g. f.
4. Singularity analysis yields asymptotics

General methodology

Asymptotic estimates are obtained using a 4 steps meta-algorithm:

1. Find the right model
2. Translate into grammar
3. Translate into system and solve g. f.
4. Singularity analysis yields asymptotics

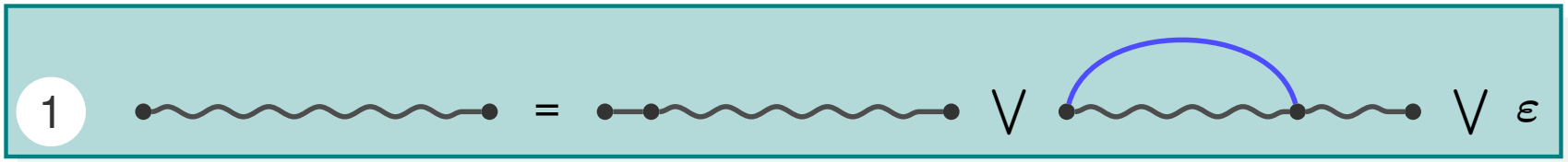
General methodology

Asymptotic estimates are obtained using a 4 steps meta-algorithm:

1. Find the right model
2. Translate into grammar
3. Translate into system and solve g. f.
4. Singularity analysis yields asymptotics

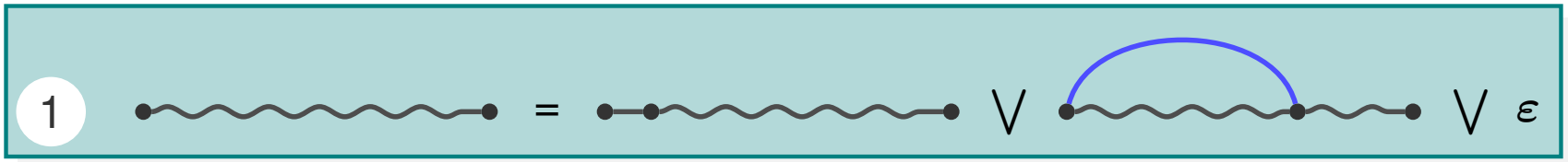
Appetizer: Motzkin words

Let us count dot-bracket notations (Motzkin words)



Appetizer: Motzkin words

Let us count dot-bracket notations (Motzkin words)



2 $M \rightarrow \bullet M \mid (M)M \mid \varepsilon$

Let us count dot-bracket notations (Motzkin words)

1 

2 $M \rightarrow \bullet\, M \mid (\, M \,)\, M \mid \varepsilon$

3 $M(z) = z \cdot M(z) + z \cdot M(z) \cdot z \cdot M(z) + 1$

$= \frac{1-z\pm\sqrt{1-2z-3z^2}}{2z^2}$

Let us count dot-bracket notations (Motzkin words)

1 

2 $M \rightarrow \bullet\, M \mid (\, M \,)\, M \mid \varepsilon$

3 $$M(z) = z \cdot M(z) + z \cdot M(z) \cdot z \cdot M(z) + 1$$

$$= \begin{cases} \frac{1-z+\sqrt{1-2z-3z^2}}{2z^2} = \frac{1}{z^2} - \frac{1}{z} - 1 - z - 2z^2 + \mathcal{O}(z^3) \\ \frac{1-z-\sqrt{1-2z-3z^2}}{2z^2} = 1 + z + 2z^2 + 4z^3 + 9z^4 + \mathcal{O}(z^5) \end{cases}$$

Let us count dot-bracket notations (Motzkin words)

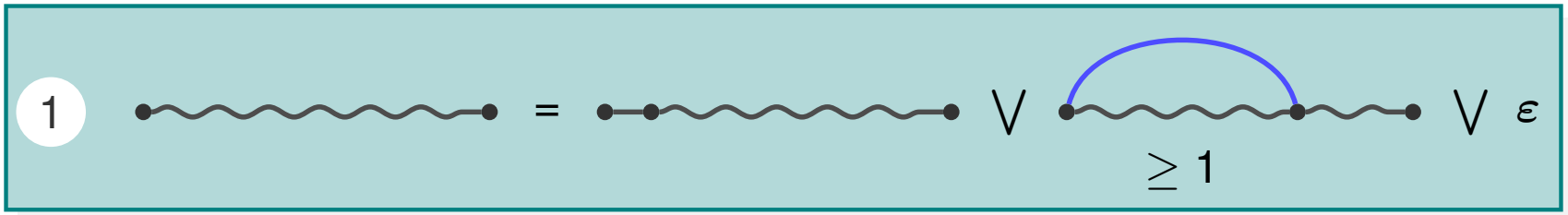
1 

2 $M \quad \rightarrow \quad \bullet\, M \quad | \quad (\, M \,)\, M \quad | \quad \varepsilon$

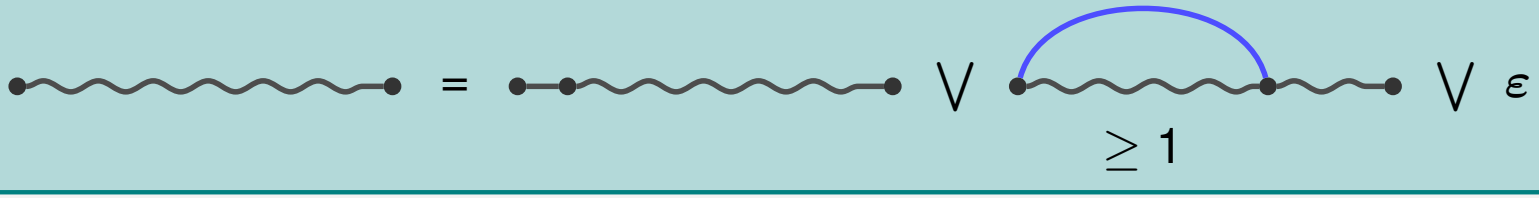
3 $$M(z) = z \cdot M(z) + z \cdot M(z) \cdot z \cdot M(z) + 1$$
$$= \frac{1-z-\sqrt{1-2z-3z^2}}{2z^2}$$

4 $\rho = 1/3$, $M(z) = \frac{1-z}{2z^2} - g(z) \cdot \sqrt{1-z/\rho}$, and $g(z) := \frac{\sqrt{1+z}}{2z^2}$
$\Rightarrow s_n \equiv [z^n]M(z) \sim \frac{g(\rho)\rho^{-n}}{\Gamma(-\alpha)n^{\alpha+1}} = \frac{3\sqrt{3}}{2\sqrt{\pi}} \cdot \frac{3^n}{n\sqrt{n}}(1 + \mathcal{O}(1/n))$

Consider RNA secondary structures (Waterman 78)



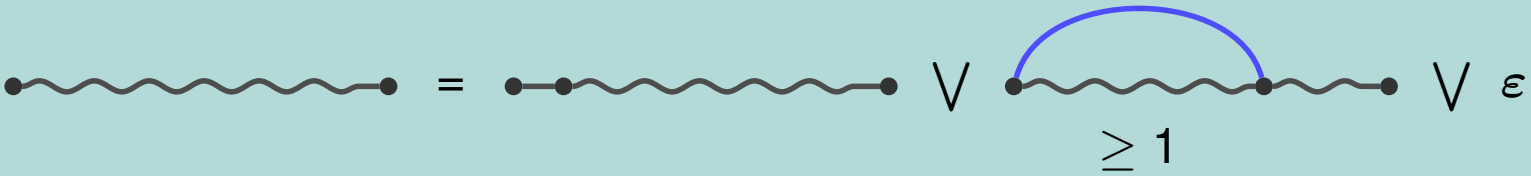
Consider RNA secondary structures (Waterman 78)

1 

2 $S \rightarrow \bullet\, S \mid (\, S_{>0}\,)\, S \mid \varepsilon$

Consider RNA secondary structures (Waterman 78)

1

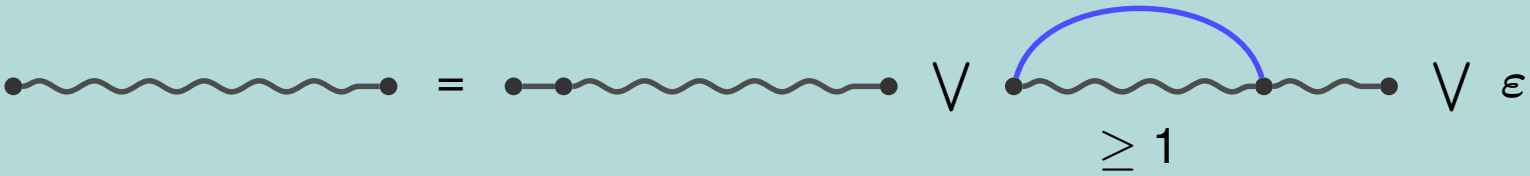


2

$$S \rightarrow \bullet\, S \mid (\,\mathbf{T}\,)\, S \mid \varepsilon$$

$$\mathbf{T} \rightarrow \bullet\, S \mid (\,\mathbf{T}\,)\, S$$

Consider RNA secondary structures (Waterman 78)

1 

2

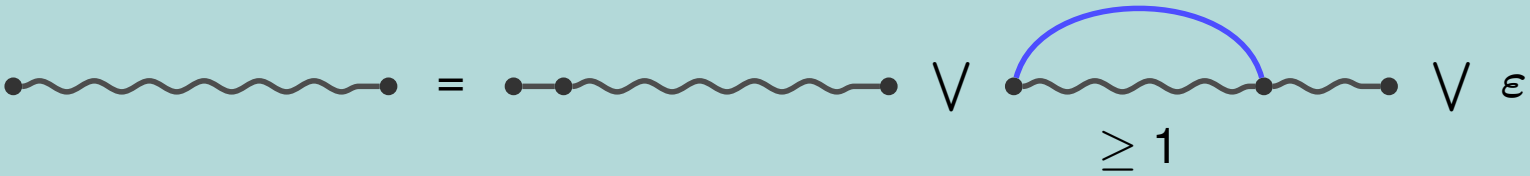
$$S \rightarrow \bullet\, S \mid (\,\mathbf{T}\,)\, S \mid \varepsilon$$

$$\mathbf{T} \rightarrow \bullet\, S \mid (\,\mathbf{T}\,)\, S$$

3

$$S(z) = \frac{1-z+z^2-\sqrt{1-2z-z^2-2z^3+z^4}}{2z^2}$$

Consider RNA secondary structures (Waterman 78)

1 

2

$$S \rightarrow \bullet\, S \mid (\,\mathbf{T}\,)\, S \mid \varepsilon$$

$$\mathbf{T} \rightarrow \bullet\, S \mid (\,\mathbf{T}\,)\, S$$

3

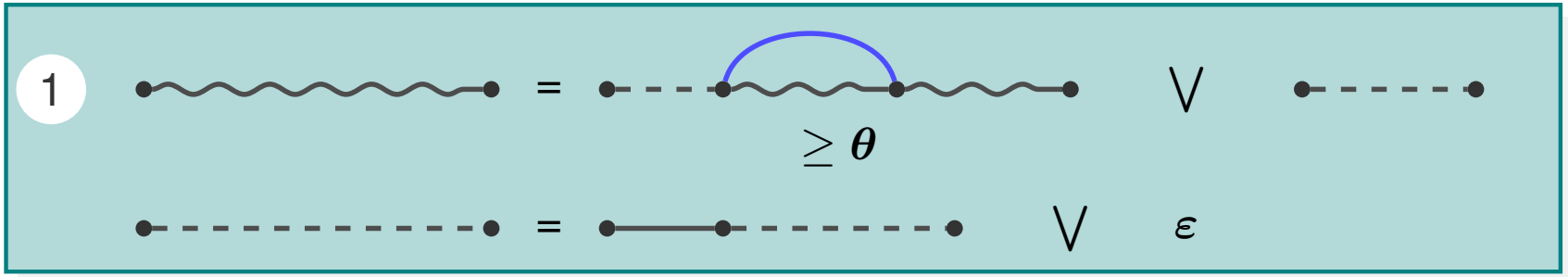
$$S(z) = \frac{1-z+z^2-\sqrt{1-2z-z^2-2z^3+z^4}}{2z^2}$$

4

$$\rho = \frac{3-\sqrt{5}}{2} = 1-\phi$$

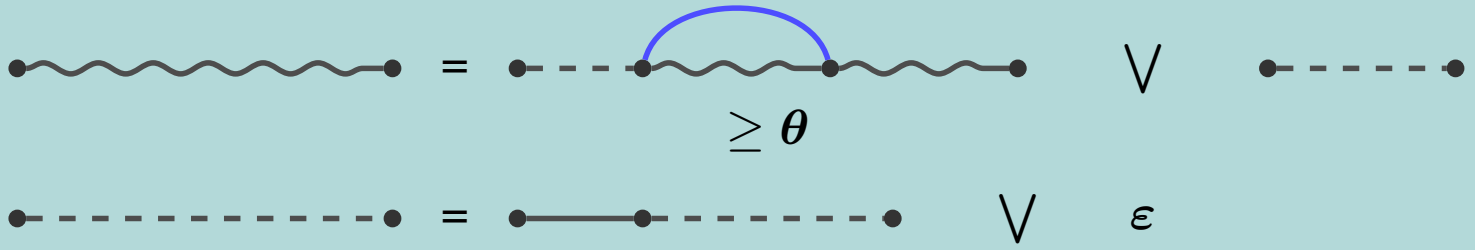
$$[z^n]S(z) = \sqrt{\frac{15+7\sqrt{5}}{8\pi}} \cdot \frac{\left(\frac{3+\sqrt{5}}{2}\right)^n}{n\sqrt{n}}(1+\mathcal{O}(1/n)) \sim 1.1 \cdot \frac{2.6^n}{n\sqrt{n}}$$

Let us generalize the θ constraint



Let us generalize the θ constraint

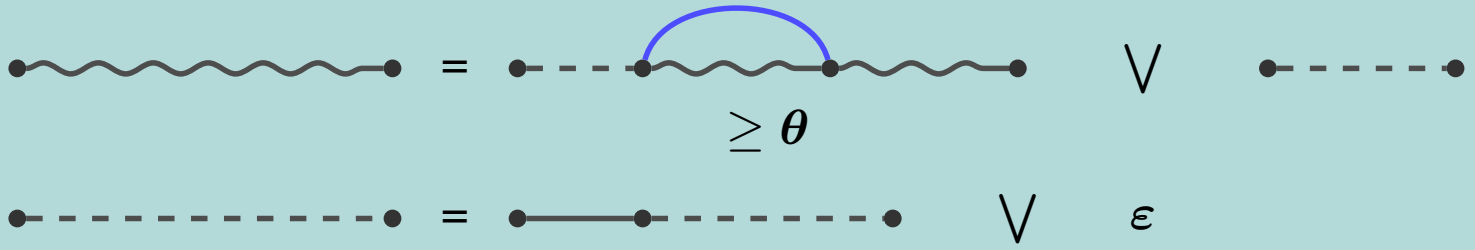
1



2 $S \rightarrow U\,(\,S_{\geq\theta}\,)\,S \mid U \qquad U \rightarrow \bullet\,U \mid \varepsilon$

Let us generalize the θ constraint

1



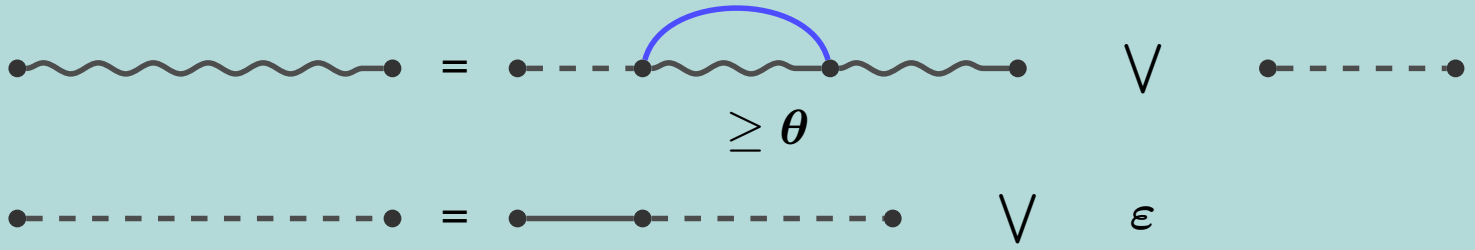
2

$$S \rightarrow U\,(\,\mathbf{T}\,)\,S \mid U \qquad U \rightarrow \bullet\, U \mid \varepsilon$$

$$\mathbf{T} \rightarrow U\,(\,\mathbf{T}\,)\,S \mid \bullet^{\theta}\, U$$

Let us generalize the θ constraint

1



2

$$S \rightarrow U\,(\,\mathbf{T}\,)\,S \mid U \qquad U \rightarrow \bullet\, U \mid \varepsilon$$

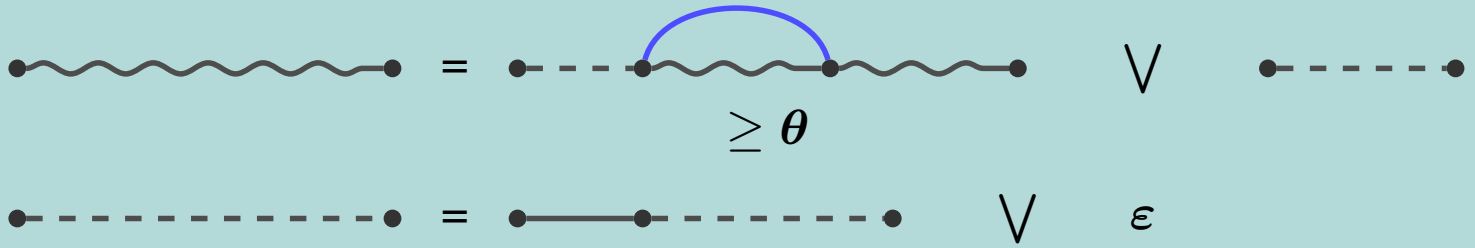
$$\mathbf{T} \rightarrow U\,(\,\mathbf{T}\,)\,S \mid \bullet^{\theta}\, U$$

3

$$S(z) = \frac{1-2z+2z^2-z^{\theta+2}-\sqrt{1-4z+4z^2-2z^{\theta+2}+4z^{\theta+3}-4z^{\theta+4}+z^{2\theta+4}}}{(1-z)2z^2}$$

Let us generalize the θ constraint

1



2

$$S \rightarrow U\,(\,\mathbf{T}\,)\,S \mid U \qquad U \rightarrow \bullet\, U \mid \varepsilon$$

$$\mathbf{T} \rightarrow U\,(\,\mathbf{T}\,)\,S \mid \bullet^{\theta}\, U$$

3

$$S(z) = \frac{1-2z+2z^2-z^{\theta+2}-\sqrt{1-4z+4z^2-2z^{\theta+2}+4z^{\theta+3}-4z^{\theta+4}+z^{2\theta+4}}}{(1-z)2z^2}$$

4

$$s_n \sim K.\frac{\beta^n}{n\sqrt{n}}(1 + \mathcal{O}(1/n))$$

θ	0	1	3	10
β	3.	2.62	2.29	2.02

Technical summary

Message #1

Finding the right decomposition (DP) is a combinatorial task.

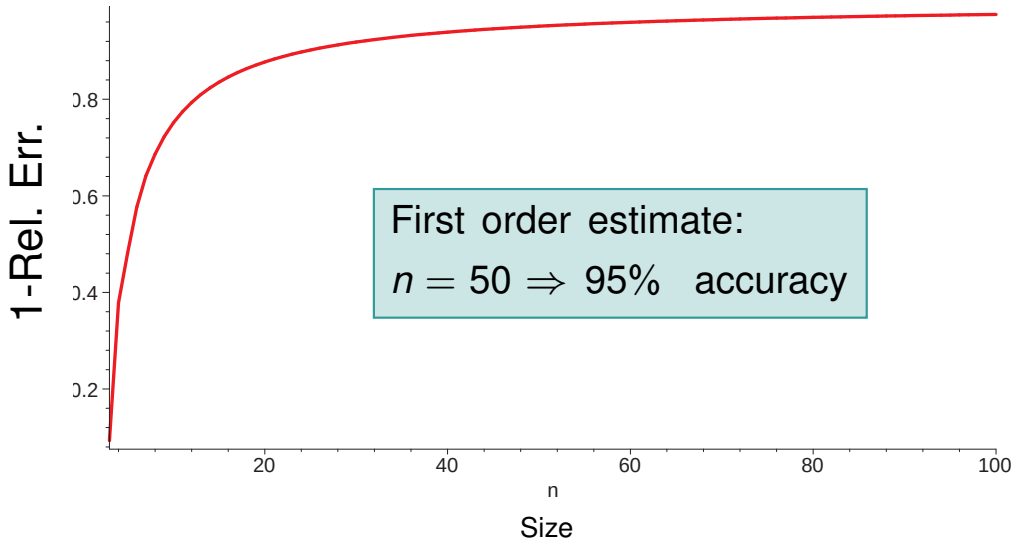
Technical summary

Message #1

Finding the right decomposition (DP) is a combinatorial task.

Message #2

Applying automatic theorems gives precise asymptotic equivalents.
+ Asymptotic regime is reached for small lengths.



Secondary structures

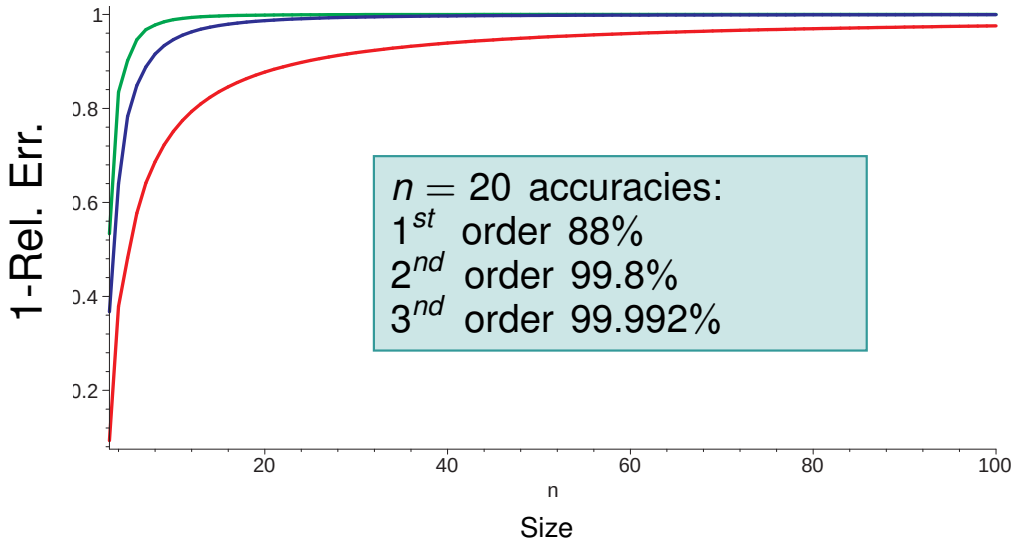
Technical summary

Message #1

Finding the right decomposition (DP) is a combinatorial task.

Message #2

Applying automatic theorems gives precise asymptotic equivalents.
+ Asymptotic regime is reached for small lengths.



Secondary structures

Technical summary

Message #1

Finding the right decomposition (DP) is a combinatorial task.

Message #2

Applying automatic theorems gives precise asymptotic equivalents.
+ Asymptotic regime is reached for small lengths.

Message #3

There is a large exponential number of structures of size n:
Homopolymer model: $\Omega(2^n)$ Stickiness model: $\mathcal{O}(1.8^n/n^{3/2})$

Part II

Enumeration of RNA shapes

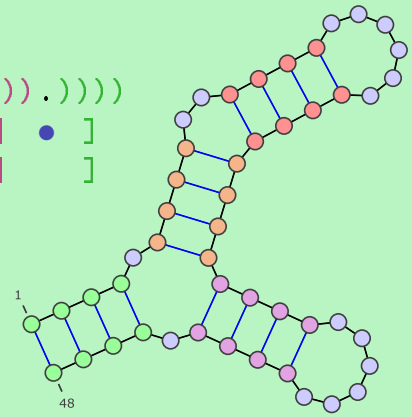
Definition (RNA shapes [Giegerich *et al*])

Coarse-grain representation hierarchy for RNA sec. struct.

Based on the *underlying backbone* structure.

Example

```
Sec. str. ((((.((((..((((......))))))))((((......)))).))))
π′-shape  [ • [ • [   •   ] ] [   •   ] • ]
π-shape   [   [     -       - ] [       ]   ]
```



Definition (RNA shapes [Giegerich *et al*])

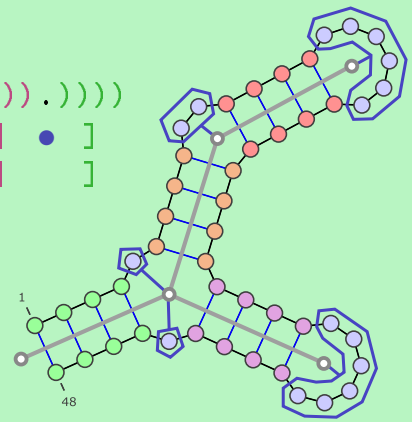
Coarse-grain representation hierarchy for RNA sec. struct.

Based on the *underlying backbone* structure.

Example

```
Sec. str. ((((.((((..((((......))))))))((((......))))).))))
π′-shape  [ • [ • [   •   ] ] [   •   ] • ]
π-shape   [   [     -     - ] [       ]   ]
```

Contract identical consecutive characters



Definition (RNA shapes [Giegerich *et al*])

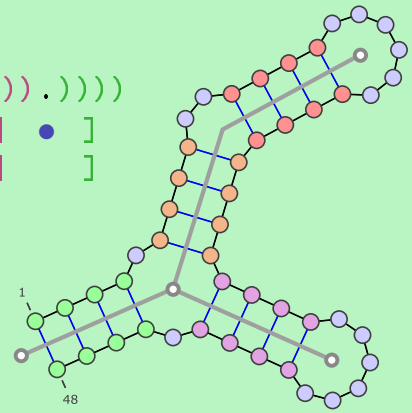
Coarse-grain representation hierarchy for RNA sec. struct.

Based on the *underlying backbone* structure.

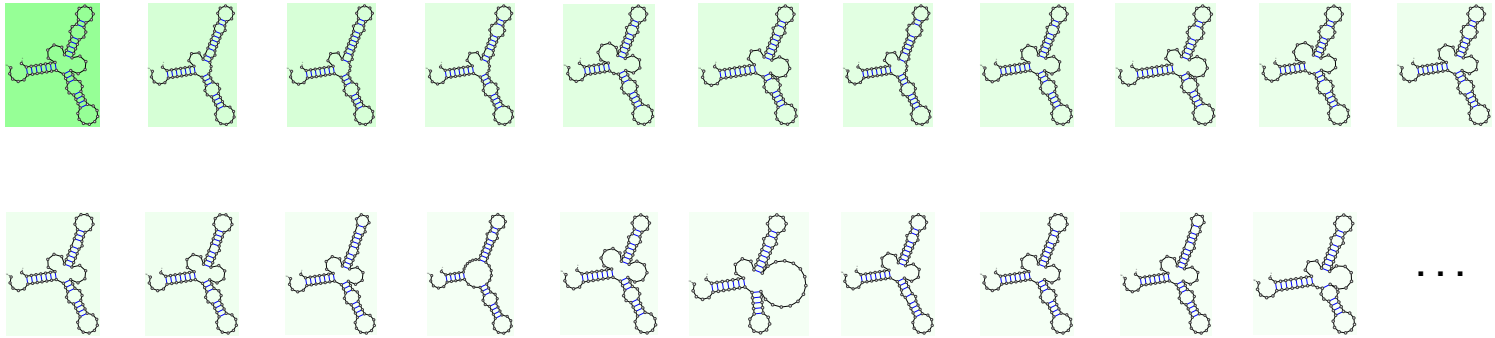
Example

```
Sec. str. ((((.((((..((((......))))))))((((......)))).))))
π′-shape  [ • [ • [   •   ] ] [   •   ] • ]
π-shape   [   [     -     - ] [       ]   ]
```

Remove unpaired regions
Contract nested helices

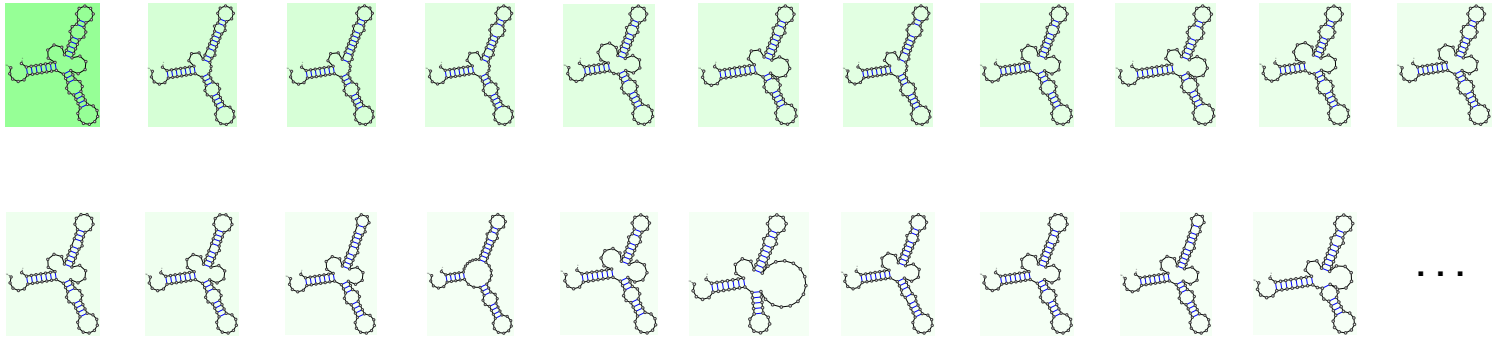


RNA shapes allow a hierarchical search in the Boltzmann ensemble

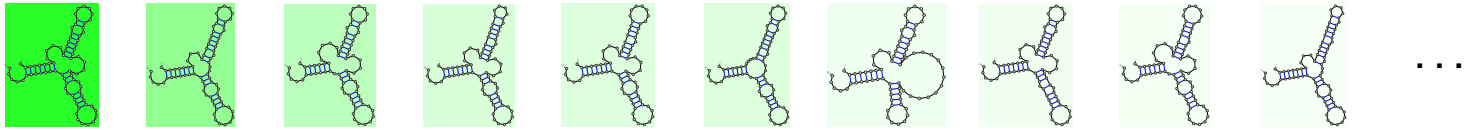


10000 samples ⇒ 1727 Secondary structures...

RNA shapes allow a hierarchical search in the Boltzmann ensemble

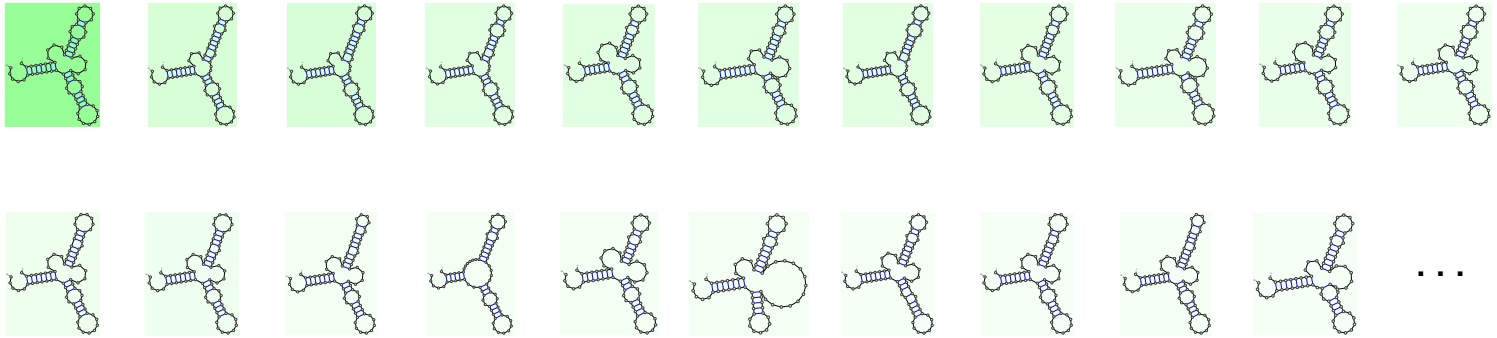


10000 samples $\Rightarrow$ 1727 Secondary structures...

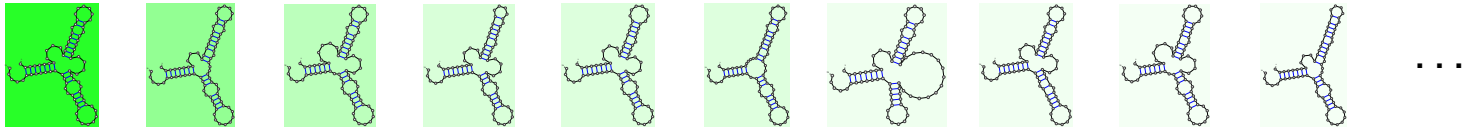


...406 π'-shapes...

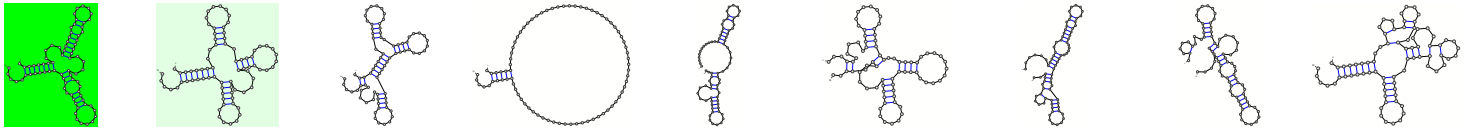
RNA shapes allow a hierarchical search in the Boltzmann ensemble



10000 samples $\Rightarrow$ 1727 Secondary structures...

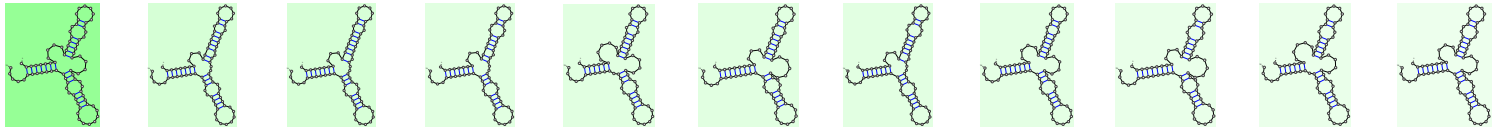


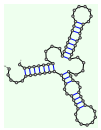
...406 π'-shapes...



...but only 9 π-shapes!

RNA shapes allow a hierarchical search in the Boltzmann ensemble



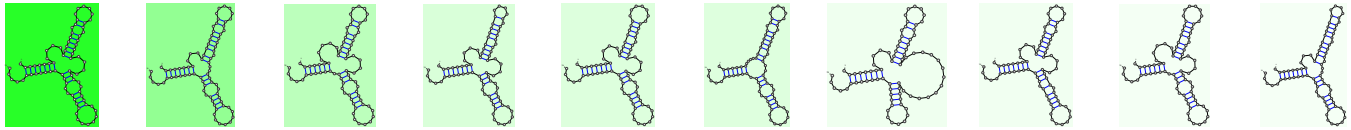


Can we reasonably perform an exhaustive search of all possible shapes compatible with input structure?

How many shapes must we investigate?

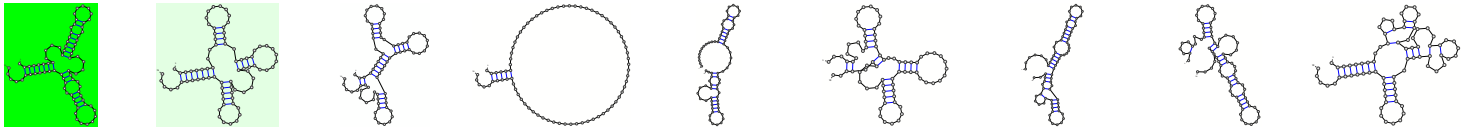
10000 samples ⇒ 1727 Secondary structures...

...



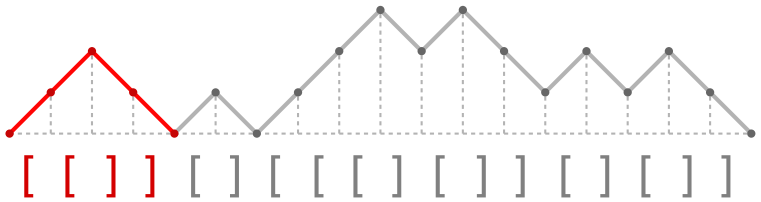
...

...406 π'-shapes...



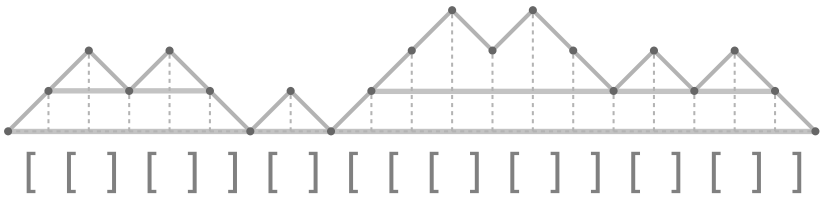
...but only 9 π-shapes!

Objective: Count π-shapes with $2n$ parentheses.



1 π-shapes are bracket words avoiding the `[[...]]` motif.

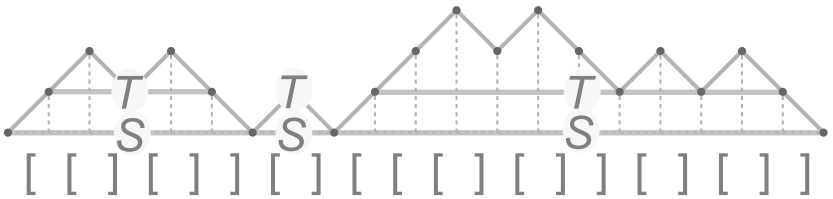
Objective: Count π-shapes with $2n$ parentheses.



1. π-shapes are bracket words avoiding the `[[...]]` motif.

2. $S \rightarrow [\, S_{/\{[\ldots]\}} \,]\, S \mid [\, S_{/\{[\ldots]\}} \,]$

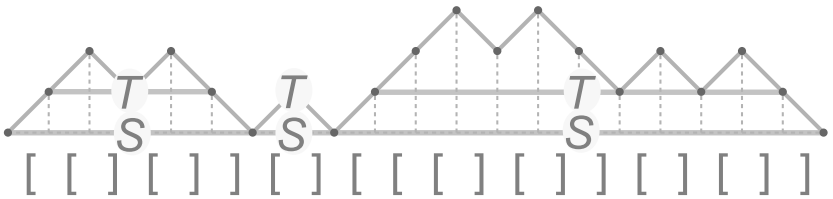
Objective: Count π-shapes with $2n$ parentheses.



1 π-shapes are bracket words avoiding the `[[...]]` motif.

2 $S \rightarrow [\,\mathbf{T}\,]\,S \mid [\,\mathbf{T}\,]$ $\qquad$ $\mathbf{T} \rightarrow [\,\mathbf{T}\,]\,S \mid \varepsilon$

Objective: Count π-shapes with $2n$ parentheses.

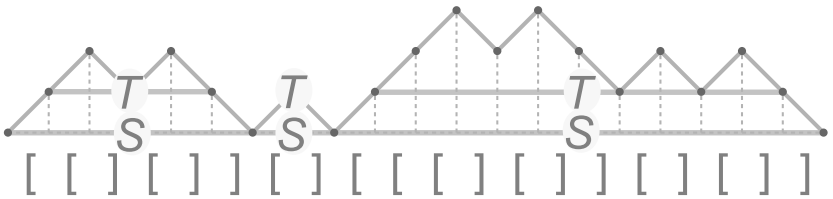


1 π-shapes are bracket words avoiding the `[[...]]` motif.

2 $S \rightarrow [\,\mathbf{T}\,]\,S \mid [\,\mathbf{T}\,]$ $\qquad$ $\mathbf{T} \rightarrow [\,\mathbf{T}\,]\,S \mid \varepsilon$

3 $$S(z) = \frac{1 - z^2 - \sqrt{1 - 2z^2 - 3z^4}}{2z^2}$$

Objective: Count π-shapes with $2n$ parentheses.



1 π-shapes are bracket words avoiding the `[[...]]` motif.

2 $S \rightarrow [\,\mathbf{T}\,]\,S \mid [\,\mathbf{T}\,]$ $\qquad$ $\mathbf{T} \rightarrow [\,\mathbf{T}\,]\,S \mid \varepsilon$

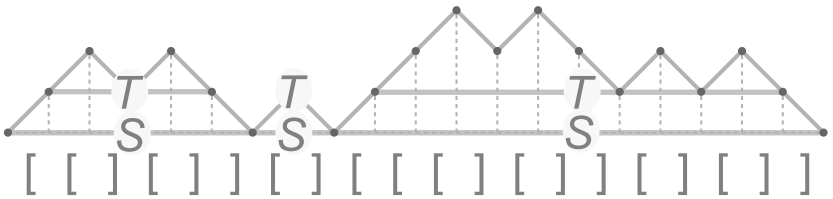
3 $$S(z) = \frac{1 - z^2 - \sqrt{1 - 2z^2 - 3z^4}}{2z^2}$$

4 $$s_{2n} \sim \frac{\sqrt{3}}{2\sqrt{\pi}} \cdot \frac{3^n}{n\sqrt{\pi}}(1 + \mathcal{O}(1/n)) \quad \text{and} \quad s_{2n+1} = 0$$

Remark: Doesn't this look familiar?

These are Motzkin words, i.e. our first example.

Objective: Count π-shapes with $2n$ parentheses.



1 π-shapes are bracket words avoiding the `[[...]]` motif.

2 $S \rightarrow [\,\mathbf{T}\,]\,S \mid [\,\mathbf{T}\,]$ $\qquad$ $\mathbf{T} \rightarrow [\,\mathbf{T}\,]\,S \mid \varepsilon$

3 $$S(z) = \frac{1 - z^2 - \sqrt{1 - 2z^2 - 3z^4}}{2z^2}$$

4 $$s_{2n} \sim \frac{\sqrt{3}}{2\sqrt{\pi}} \cdot \frac{3^n}{n\sqrt{\pi}}(1 + \mathcal{O}(1/n)) \quad \text{and} \quad s_{2n+1} = 0$$

Remark: Doesn't this look familiar?
These are Motzkin words, i.e. our first example.

Limitations

Number of π-shapes of size n

$\neq$

Number of π-shapes compatible with RNA of size n

Reasons:

1. Shapes of size $\leq n$ should be considered
2. Forming a hairpin loop [ ] *takes* at least $\theta + 2$ bases

2 $\quad S \rightarrow [\,T\,]\,S \mid [\,T\,] \qquad T \rightarrow [\,T\,]\,S \mid \varepsilon$

3 $$S(z) = \frac{1 - z^2 - \sqrt{1 - 2z^2 - 3z^4}}{2z^2}$$

4 For n even: $s_{2n} \sim \frac{3\sqrt{3}}{4\sqrt{\pi}} \cdot \frac{3^n}{n\sqrt{\pi}}(1 + \mathcal{O}(1/n)) \approx 0.48 \cdot \frac{3^n}{n\sqrt{n}}$

Limitations

Number of π-shapes of size n

$\neq$

Number of π-shapes compatible with RNA of size n

Reasons:

1. Shapes of size $\leq n$ should be considered
2. Forming a hairpin loop [ ] *takes* at least $\theta + 2$ bases

2

$$S \rightarrow [\,T\,]\,S \mid [\,T\,] \qquad T \rightarrow [\,T\,]\,S \mid \bullet^{\theta}$$

$$R \rightarrow \square\,\mathrm{S} \mid \varepsilon$$

3

$$R(z) = \frac{1 - z^2 - \sqrt{1 - 2z^2 - 3z^4}}{2z^2(\mathbf{1} - \mathbf{z})}$$

4

$$r_{2n} \sim \frac{3\sqrt{3}}{4\sqrt{\pi}} \cdot \frac{3^n}{n\sqrt{n}}(1 + \mathcal{O}(1/n)) \Rightarrow r_n \approx 2.07 \cdot \frac{1.73^n}{n\sqrt{n}}$$

Limitations

Number of π-shapes of size n

$\neq$

Number of π-shapes compatible with RNA of size n

Reasons:

1. Shapes of size $\leq n$ should be considered
2. Forming a hairpin loop [ ] *takes* at least $\theta + 2$ bases

2

$$S \rightarrow [\,T\,]\,S \mid [\,T\,] \qquad T \rightarrow [\,T\,]\,S \mid \bullet^{\theta}$$

$$R \rightarrow \square\,\mathrm{S} \mid \varepsilon$$

3

$$R(z) = \frac{1 - z^{\theta+2} - \sqrt{1 - 2z^{\theta+2} - 4z^{\theta+4} + z^{2\theta+4}}}{2z^2(1-z)}$$

4

$$\theta = 3 \Rightarrow r_n \approx 2.44\frac{1.32^n}{n\sqrt{n}}$$

Objective: Count π'-shapes compatible with RNA of length n.

1. π'-shapes = bracket words avoiding motifs `[[...]]` and $\bullet\bullet$

2. $R \to \square\, R \mid S \qquad S \to U\,[\,T\,]\,S \mid U \qquad U \to \diamond \mid \varepsilon$

   $T \to U\,[\,T\,]\,U\,[\,T\,]\,S \mid \diamond\,[\,T\,] \mid [\,T\,]\diamond \mid \diamond\,[\,T\,]\diamond \mid \bullet^{\theta}$

3. $\theta = 3,\ R(z) = \frac{1+2z^2+2z^3+z^4-z^5-z^6-\sqrt{1-4z^3-2z^4-2z^5+2z^6-7z^8-z^{10}+2z^{11}+z^{12}}}{2z^2(1-z^2)}$

4. $r_n \sim 1.27\,\frac{1.81^n}{n\sqrt{n}}$

Objective: Count π'-shapes compatible with RNA of length n.

1. π'-shapes = bracket words avoiding motifs `[[...]]` and $\bullet\bullet$

2. $R \to \square\, R \mid S \qquad S \to U\,[\,T\,]\,S \mid U \qquad U \to \diamond \mid \varepsilon$

   $T \to U\,[\,T\,]\,U\,[\,T\,]\,S \mid \diamond\,[\,T\,] \mid [\,T\,]\diamond \mid \diamond\,[\,T\,]\diamond \mid \bullet^{\theta}$

3. $\theta = 3,\ R(z) = \frac{1+2z^2+2z^3+z^4-z^5-z^6-\sqrt{1-4z^3-2z^4-2z^5+2z^6-7z^8-z^{10}+2z^{11}+z^{12}}}{2z^2(1-z^2)}$

4. $r_n \sim 1.27\,\frac{1.81^n}{n\sqrt{n}}$

Objective: Count π'-shapes compatible with RNA of length n.

1. π'-shapes = bracket words avoiding motifs `[[...]]` and $\bullet\bullet$

2. $R \to \square\, R \mid S \qquad S \to U\,[\,T\,]\,S \mid U \qquad U \to \diamond \mid \varepsilon$

   $T \to U\,[\,T\,]\,U\,[\,T\,]\,S \mid \diamond\,[\,T\,] \mid [\,T\,]\diamond \mid \diamond\,[\,T\,]\diamond \mid \bullet^{\theta}$

3. $\theta = 3,\ R(z) = \frac{1+2z^2+2z^3+z^4-z^5-z^6-\sqrt{1-4z^3-2z^4-2z^5+2z^6-7z^8-z^{10}+2z^{11}+z^{12}}}{2z^2(1-z^2)}$

4. $r_n \sim 1.27\,\frac{1.81^n}{n\sqrt{n}}$

Objective: Count π'-shapes compatible with RNA of length n.

1. π'-shapes = bracket words avoiding motifs `[[...]]` and $\bullet\bullet$

2. $R \to \square\, R \mid S \qquad S \to U\,[\,T\,]\,S \mid U \qquad U \to \diamond \mid \varepsilon$

   $T \to U\,[\,T\,]\,U\,[\,T\,]\,S \mid \diamond\,[\,T\,] \mid [\,T\,]\diamond \mid \diamond\,[\,T\,]\diamond \mid \bullet^{\theta}$

3. $\theta = 3,\ R(z) = \frac{1+2z^2+2z^3+z^4-z^5-z^6-\sqrt{1-4z^3-2z^4-2z^5+2z^6-7z^8-z^{10}+2z^{11}+z^{12}}}{2z^2(1-z^2)}$

4. $$r_n \sim 1.27\frac{1.81^n}{n\sqrt{n}}$$

Summary

Model	Asymptotic number
Sec. str. on n – Combinatorial	$1.1 \cdot \frac{\mathbf{2.6}^n}{n\sqrt{n}}$
Sec. str. on n – Empirical	$0.04 \cdot \frac{\mathbf{1.4}^n}{n\sqrt{n}}$
π-shapes of size n	$1.38 \cdot \frac{\mathbf{1.73}^n}{n\sqrt{n}}$
π-shapes compatible with sec. str. on n	$2.44 \cdot \frac{\mathbf{1.32}^n}{n\sqrt{n}}$
π-shapes – Empirical	$0.21 \cdot \frac{\mathbf{1.1}^n}{n\sqrt{n}}$
π'-shapes of size n	$0.99 \cdot \frac{\mathbf{2.41}^n}{n\sqrt{n}}$
π'-shapes compatible with sec. str. on n	$1.28 \cdot \frac{\mathbf{1.81}^n}{n\sqrt{n}}$

Message #4

The number of π-shapes (coarsest) is upper-bounded by 1.3^n.
$\Rightarrow$ Shape-based indexing is feasible through exhaustive search.

Part III

Realistic models for random RNA structures

Motivation

Random generation can be used to:

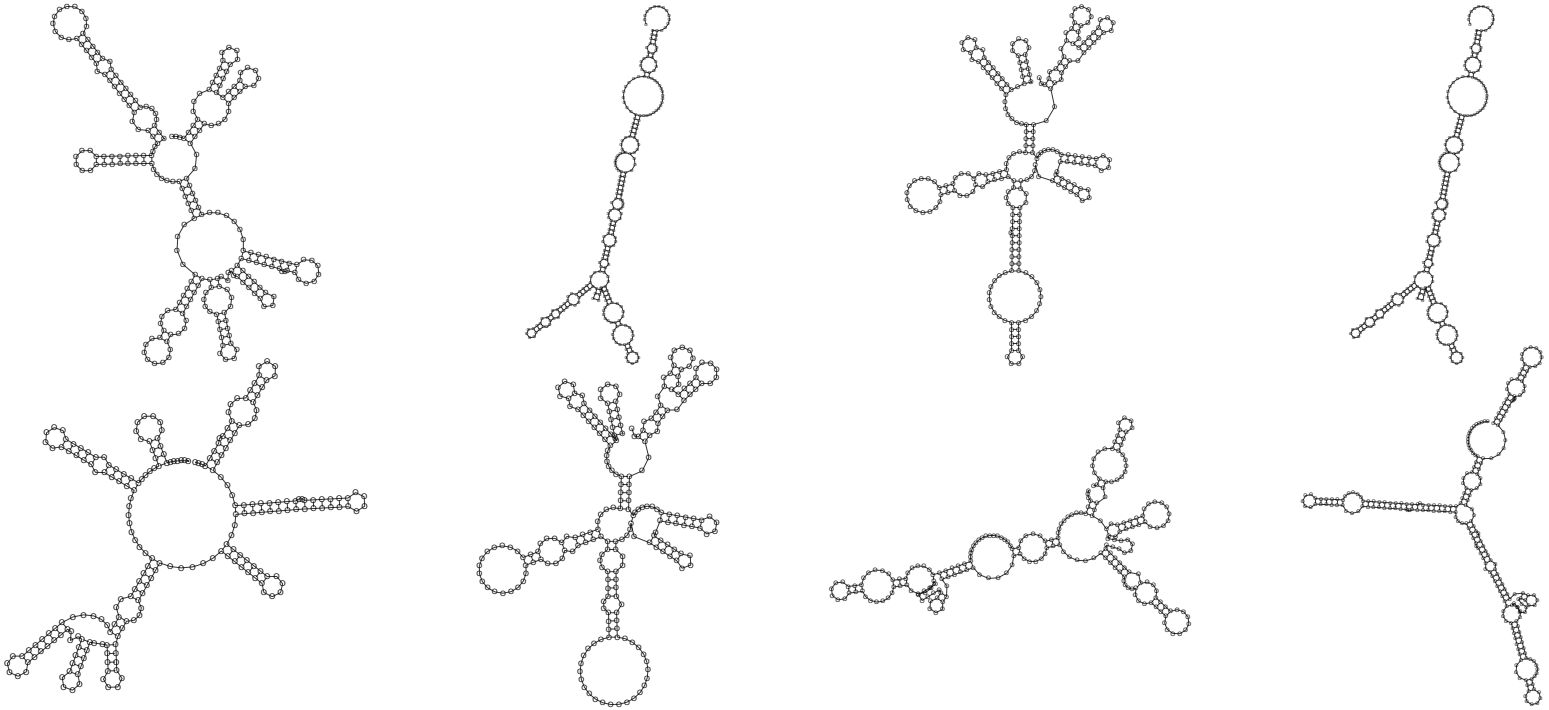
- Assess the significance of observed phenomena
- Estimate the practical complexity of algorithms

Worst-case complexity can be *misleading*:

- Statistical sampling: $\Theta(n^2) \to \mathcal{O}(n\sqrt{n})$
- Tree alignment: $\Theta(n^4) \to \mathcal{O}(n^2)$

Average case analysis: Hidden uniformity hypothesis
Yet all secondary structures are not equally likely observed as native structures!

RNA secondary structures (Mathews DB)



Grammar approach for the random generation

RNA secondary structures = ()-free Motzkin words [Wat78]

+ Uniform random generation (Boltzmann, recursive . . .)

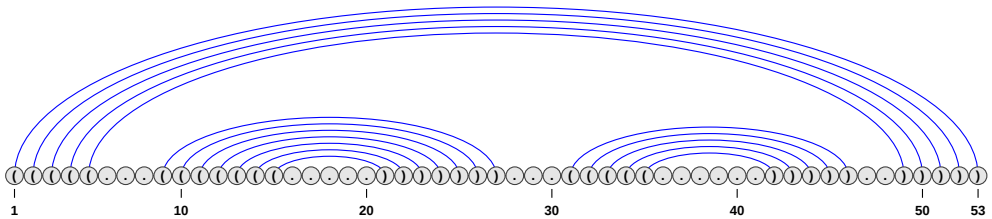
Grammar approach for the random generation

RNA secondary structures = ()-free Motzkin words [Wat78]

+ Uniform random generation (Boltzmann, recursive . . .)

Example:

(((((• • • (((((((• • • • • •))))))) • • • (((((• • • • • • •)))))) • •)))))



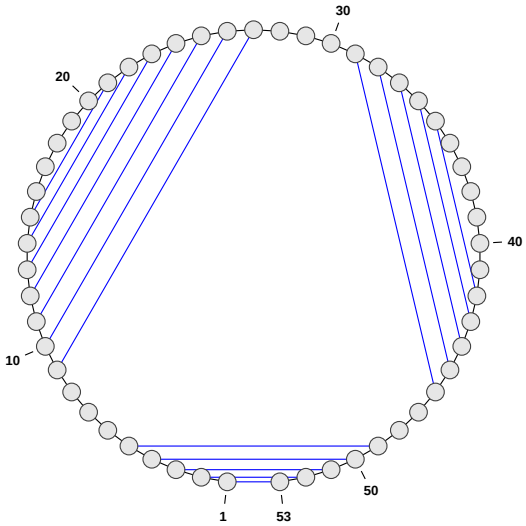
Grammar approach for the random generation

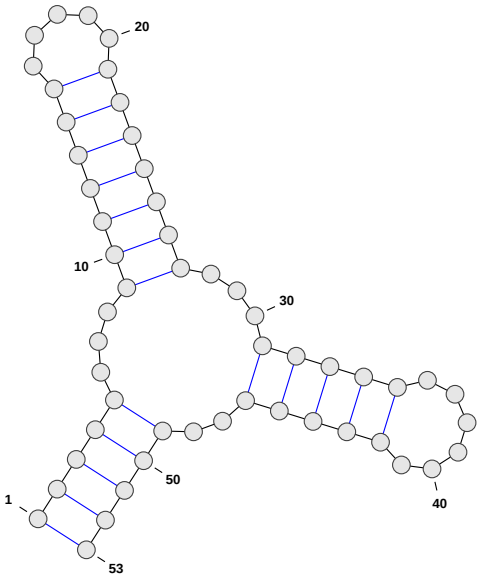
RNA secondary structures = ()-free Motzkin words [Wat78]

+ Uniform random generation (Boltzmann, recursive . . .)

Example:

(((((• • • (((((((• • • • • •))))))) • • • (((((• • • • • • •)))))) • •)))))

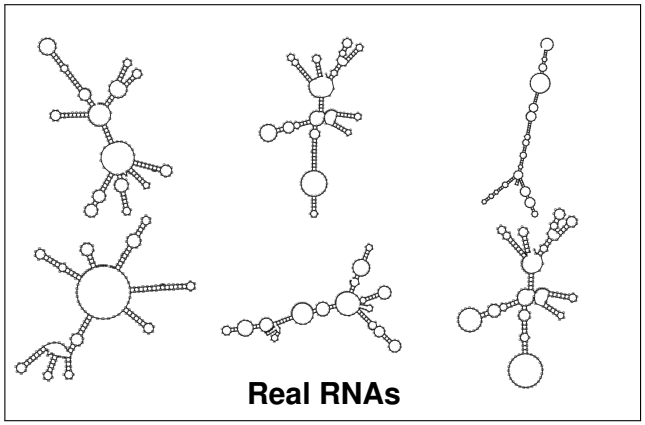


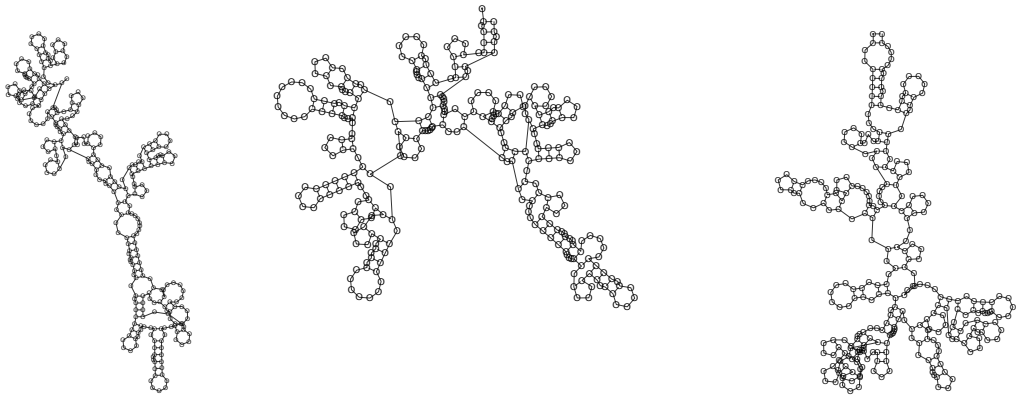


Grammar approach for the random generation

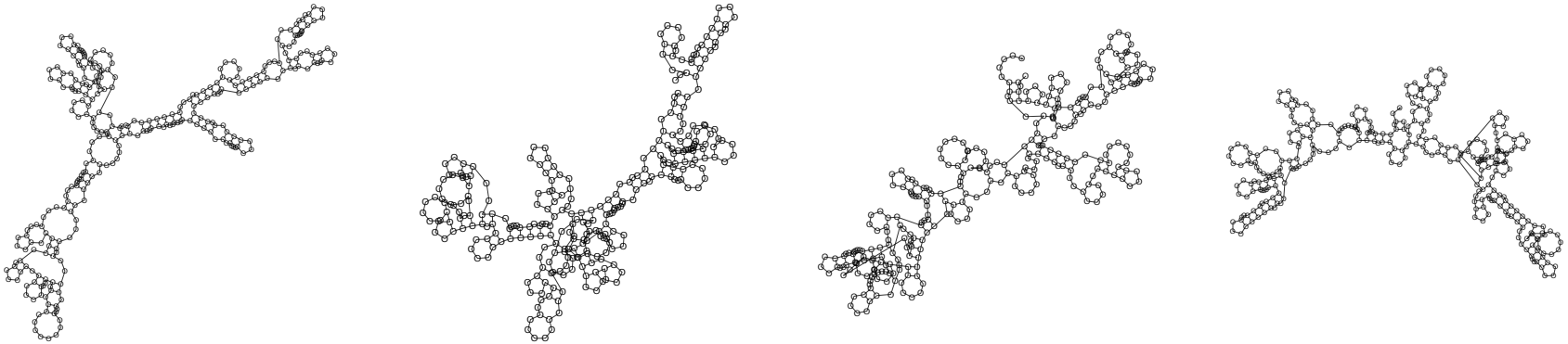
RNA secondary structures = ()-free Motzkin words [Wat78]

+ Uniform random generation (Boltzmann, recursive . . .)

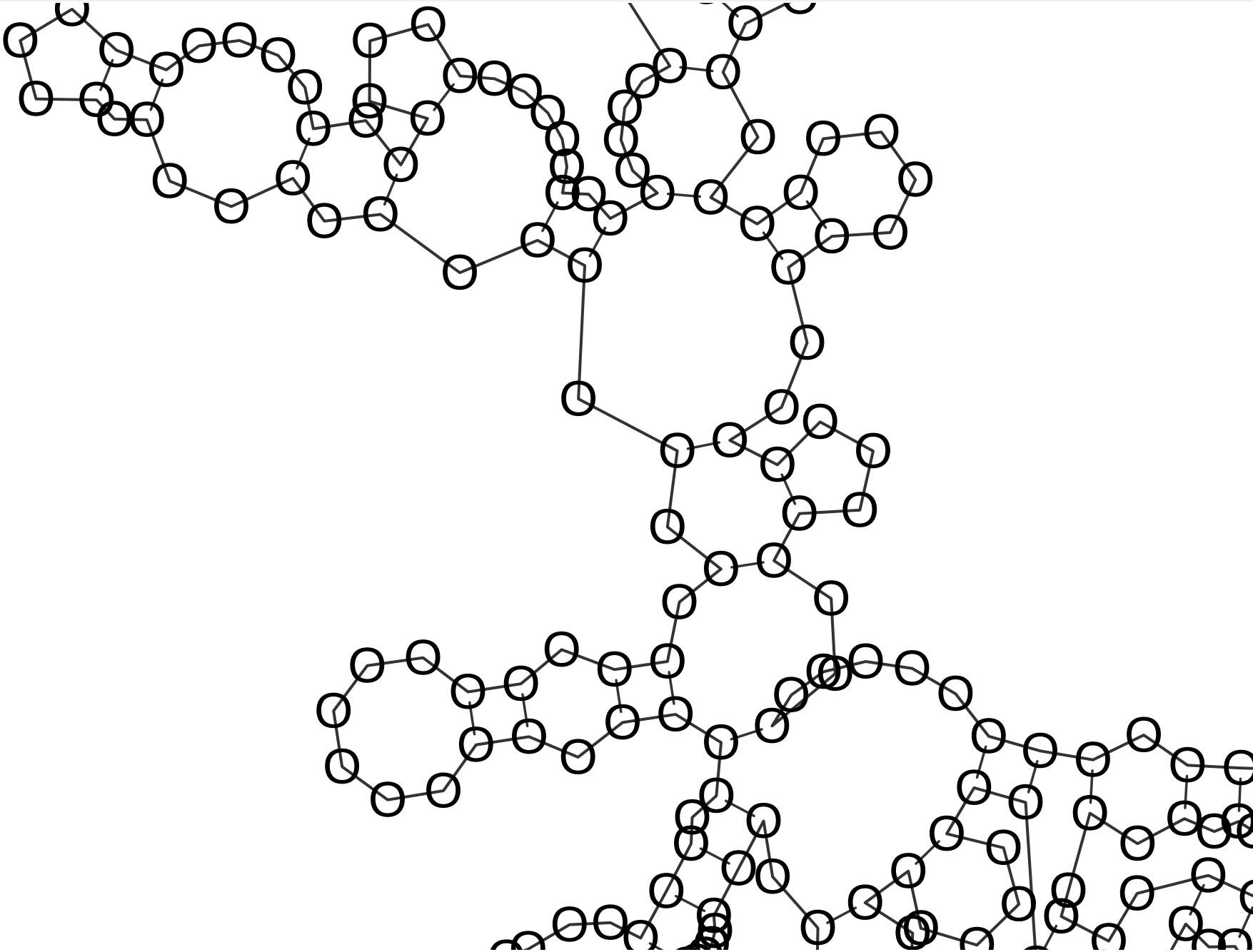




Random uniform RNAs



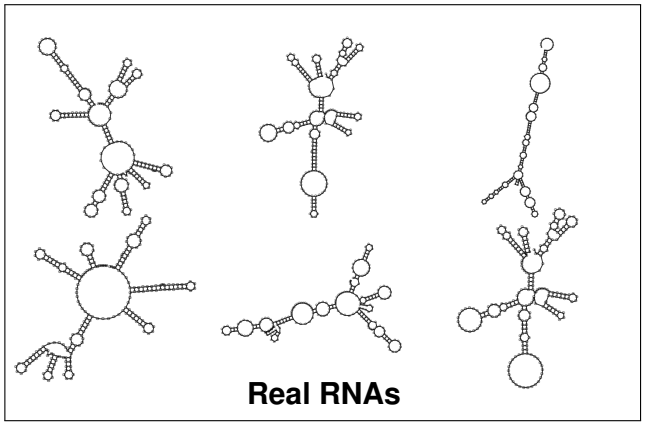
Grammar approach for the random generation

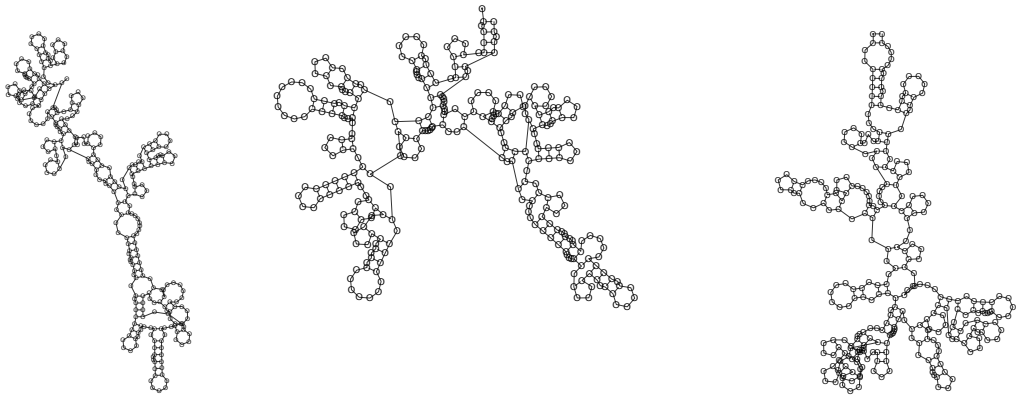


Grammar approach for the random generation

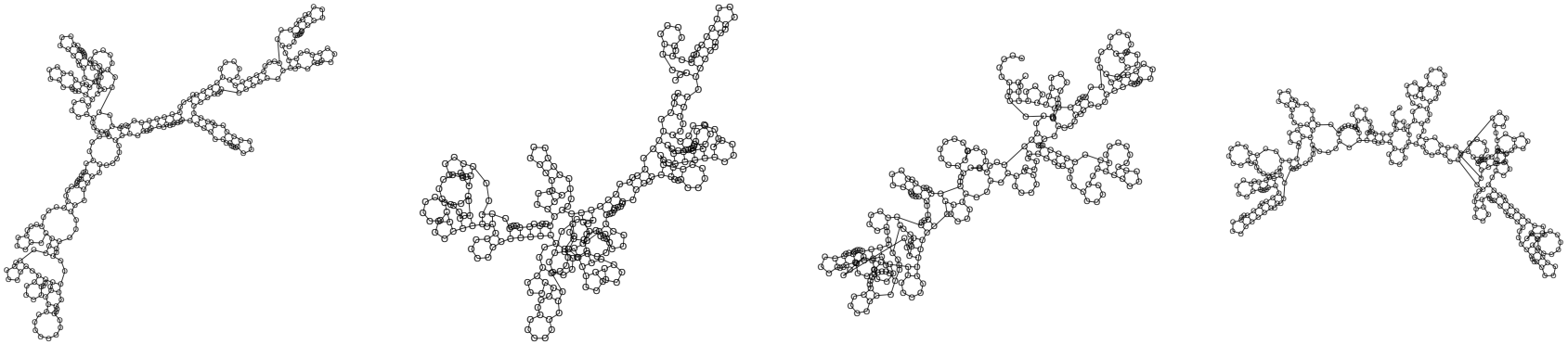
RNA secondary structures = ()-free Motzkin words [Wat78]

+ Uniform random generation (Boltzmann, recursive . . .)





Random uniform RNAs



Grammar approach for the random generation

RNA secondary structures = ()-free Motzkin words [Wat78]

+ Uniform random generation (Boltzmann, recursive . . .)

+ Sequence/structure annotation (h: Base-pair, H: Helix, . . .)

Example:

```
( ( ( ( ( • • • ( ( ( ( ( ( ( • • • • • ) ) ) ) ) ) ) • • • ( ( ( ( ( • • • • • • ) ) ) ) ) • • ) ) ) ) )
H h h h h M m m H h h h h h h T t t t t h h h h h h h m m m H h h h h T t t t t t h h h h h m m h h h h h
```

Annotate native structures from Mathews database
$\Rightarrow$ Proportions of occurrences of each terminal

Grammar can be changed to generate already annotated structures
$\Rightarrow$ Evaluate expected proportions in the uniform model

Available techniques:

- Empirical
- Recursive/Dynamic programming
- Flajolet–Symbolic, . . .

Grammar approach for the random generation

RNA secondary structures = ()-free Motzkin words [Wat78]

+ Uniform random generation (Boltzmann, recursive . . .)

+ Sequence/structure annotation (h: Base-pair, H: Helix, . . .)

Feature	B	b	I	i	M	m	T	t	H	h
$\mathcal{M}_0$ (%)	7.2	5.6	2.8	7.3	3.7	7.6	5.2	14.5	**18.6**	**27.5**
Exper.	1.5	2.3	1.9	11.2	1.1	9.0	2.6	16.6	**4.8**	**48.9**

Proportions of symbols associated with structural features:
Uniform model $\mathcal{M}_0$ vs Observed ($n = 300$)

As suspected, avg helix length $(\mathrm{H} + \mathrm{h})/(2 * \mathrm{H})$ is way off!
($\mathcal{M}_0$: 1.23, Exp: 5.59)

Limits of the uniform model

Message #1

RNA secondary structures $\Rightarrow$ Uniform model = Bad!

Null-hypothesis too easily refuted/Overrated statistical significance
Bonzai structures might yield biased estimates.

Message #2

Need (at least) to account for constraints on min/avg sizes of features.

- Helices
- Sequences of unpaired bases
- Degree of branching loops

Constraints captured by grammars in two ways:

- Min: Tweak grammar
- Avg: Modify distribution $\Rightarrow$ Weighted probability distribution

Definition (Context-free grammar)

Context-free grammar = 4-tuple $(\Sigma, \mathcal{N}, \mathcal{P}, \mathcal{S})$:

- Σ: Alphabet.
- $\mathcal{N}$: Non-terminal symbols.
- $\mathcal{P}$: Set of production rules $N \to X \in \mathcal{N} \times \{\Sigma \cup \mathcal{N}\}^*$.
- $\mathcal{S}$: Axiom, or initial non-terminal.

Definition (Weighted context-free grammar [DRT00])

A **weighted** context-free grammar is a **5**-tuple $\mathcal{G}_\pi = (\Sigma, \mathcal{N}, \mathcal{P}, \mathcal{S}, \boldsymbol{\pi})$:

- Σ, $\mathcal{N}$, $\mathcal{P}$, $\mathcal{S}$: Same as previously.
- π: **Weight function** $\pi : \Sigma \to \mathbb{R}$.

Then $\mathcal{G}_\pi$ implicitly defines a **weighted probability distribution** $\mathcal{W}$:

$$\forall \omega \in \mathcal{L}(\mathcal{G}_\pi),\ \mathbb{P}(\omega) = \frac{\pi(\omega)}{\pi\left(\mathcal{L}(\mathcal{G}_\pi)\right)} = \frac{\prod_{i=1}^{|\omega|} \pi(\omega_i)}{\pi\left(\mathcal{L}(\mathcal{G}_\pi)\right)}.$$

Generation: Generating k words of size n takes $\mathcal{O}(n^2 + n\log(n).k)^*$ or $\mathcal{O}(n\log(n).k)^*$ using symbolic preprocessing (linear recurrences).

*Arithmetic operations $\Rightarrow$ $\Theta(n)$ overhead OR tricky interval arithmetics [DZ99]

Definition (Weighted context-free grammar [DRT00])

A **weighted** context-free grammar is a **5**-tuple $\mathcal{G}_\pi = (\Sigma, \mathcal{N}, \mathcal{P}, \mathcal{S}, \boldsymbol{\pi})$:

- Σ, $\mathcal{N}$, $\mathcal{P}$, $\mathcal{S}$: Same as previously.
- π: **Weight function** $\pi : \Sigma \to \mathbb{R}$.

Then $\mathcal{G}_\pi$ implicitly defines a **weighted probability distribution** $\mathcal{W}$:

$$\forall \omega \in \mathcal{L}(\mathcal{G}_\pi),\ \mathbb{P}(\omega) = \frac{\pi(\omega)}{\pi\left(\mathcal{L}(\mathcal{G}_\pi)\right)} = \frac{\prod_{i=1}^{|\omega|} \pi(\omega_i)}{\pi\left(\mathcal{L}(\mathcal{G}_\pi)\right)}.$$

Generation: Generating k words of size n takes $\mathcal{O}(n^2 + n\log(n).k)^*$ or $\mathcal{O}(n\log(n).k)^*$ using symbolic preprocessing (linear recurrences).

*Arithmetic operations $\Rightarrow \Theta(n)$ overhead OR tricky interval arithmetics [DZ99]

Summary

1 – Uniform models are bad!

2 – Let's use weighted distribution!

Which one?

Weights should be computed such that:

- Sampled structure follow Boltzmann distribution
  Not very good for modeling native structure
- Expectation is maximized
  Not very easy (WCFG$\neq$SCFG)
  Overfitting vs Inconsistance
- Aiming at observed average properties
  Average random objects should mimic features of the training set

How to compute weights targeting a given observation?

Summary

1 – Uniform models are bad!

2 – Let's use weighted distribution!

Which one?

Weights should be computed such that:

- Sampled structure follow Boltzmann distribution
  Not very good for modeling native structure
- Expectation is maximized
  Not very easy (WCFG$\neq$SCFG)
  Overfitting vs Inconsistance
- Aiming at observed average properties
  Average random objects should mimic features of the training set

How to compute weights targeting a given observation?

Summary

1 – Uniform models are bad!

2 – Let's use weighted distribution!

Which one?

Weights should be computed such that:

- Sampled structure follow Boltzmann distribution
  Not very good for modeling native structure
- Expectation is maximized
  Not very easy (WCFG$\neq$SCFG)
  Overfitting vs Inconsistance
- Aiming at observed average properties
  Average random objects should mimic features of the training set

How to compute weights targeting a given observation?

Computing suitable weights

$f^{\pi}_{n,i}$: Proportion of Z_i in words of size n drawn from a π-distribution
$f^{\pi}_{*,i}$: Asymptotic proportion of Z_i in π ($f^{\pi}_{*,i} = \lim_{n\to\infty} f^{\pi}_{n,i}$)

Problem

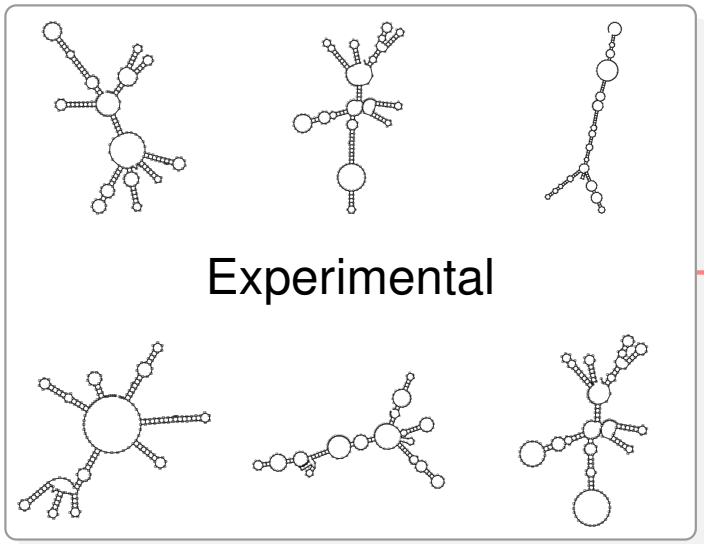
Input: Grammar $\mathcal{G}$, target frequencies (μ_i), length n
Goal: Find π_0 such that $\forall i \in [1, |\Sigma|],\ f^{\pi_0}_{n,i} = \mu_i$

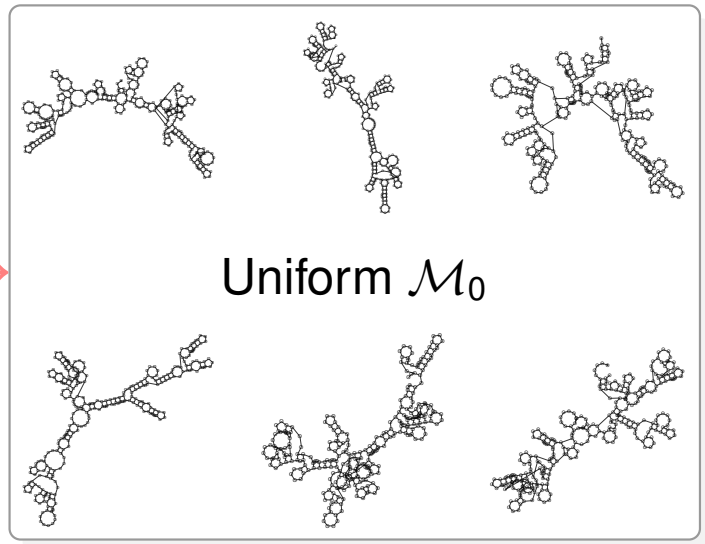
- Analytic (asymptotic) Rational
- Analytic (asymptotic) Algebraic
- Automatic heuristic (Fixed size)
  Computation of $f^{\pi}_{n,i}$ easier than that of weight vector π_0
  + *Partial monotonicity*: $(\pi(Z_i) \nearrow) \to (f^{\pi}_{n,i} \nearrow)$
  $\Rightarrow$ Rephrase as an continuous optimization problem

$$\text{Objective function: } g(\pi) = \sqrt{\sum_i \left(f^{\pi}_{n,i} - \mu_i\right)^2}$$

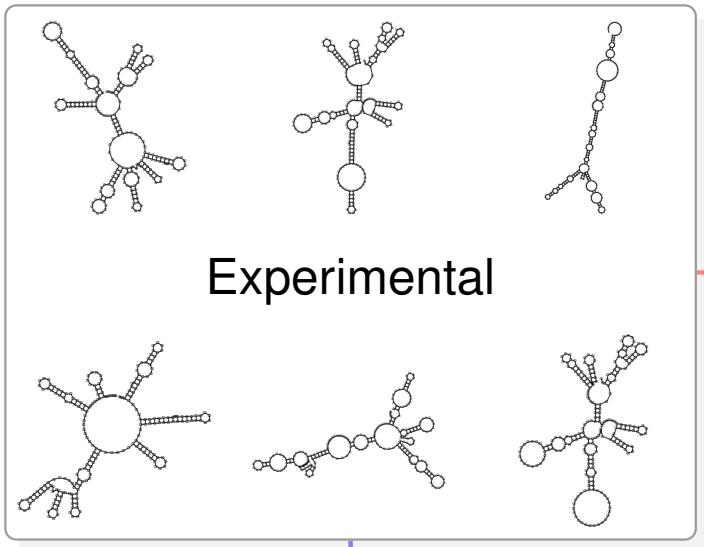
We use the CONDOR optimizer software package [Ber05].

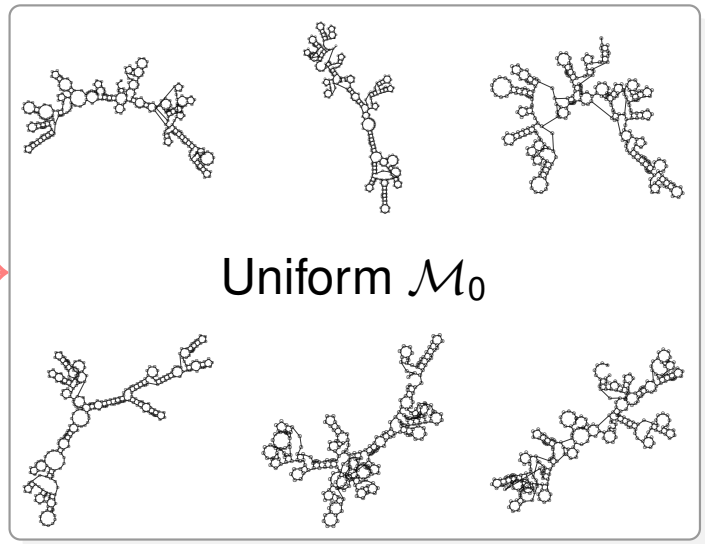
Application

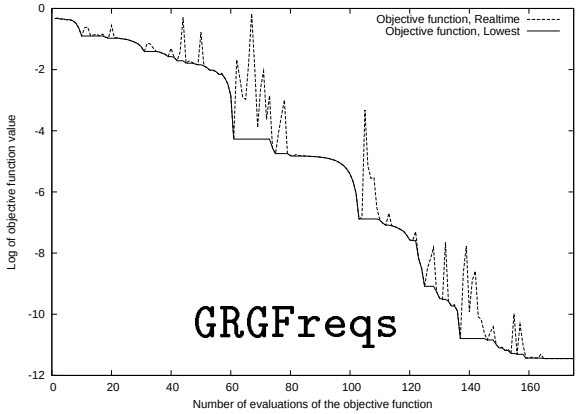




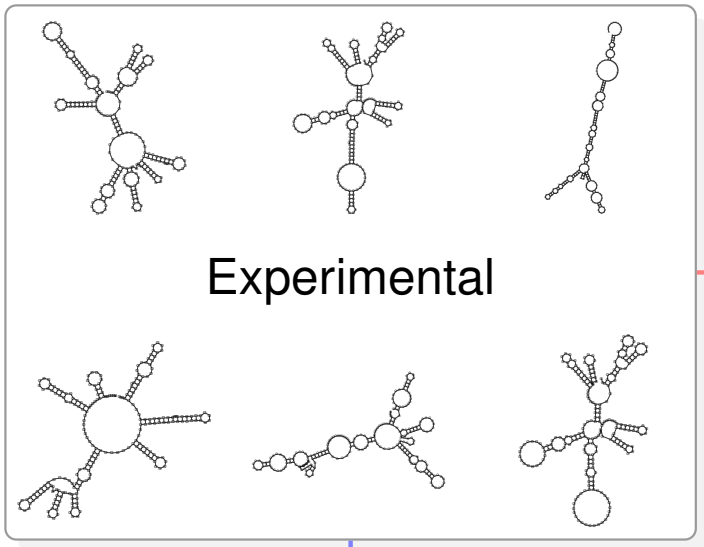
Application

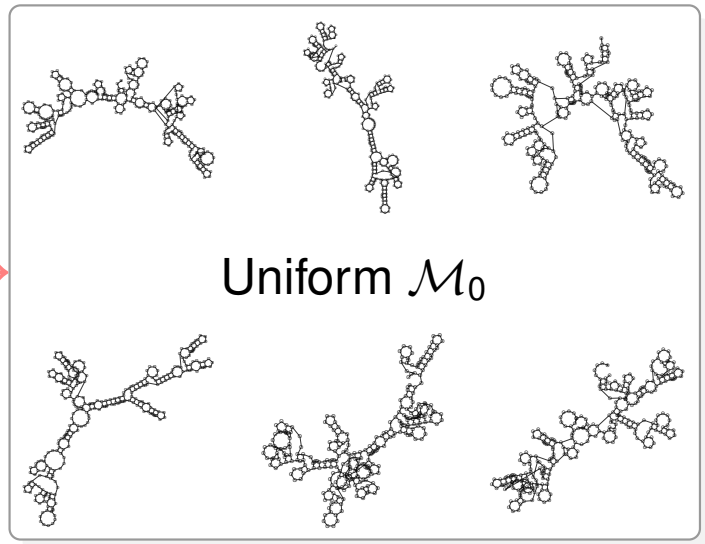


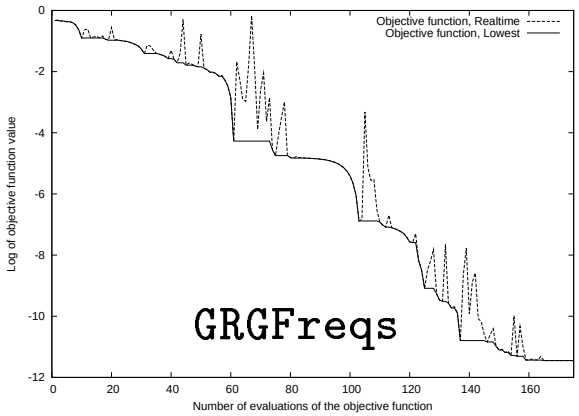


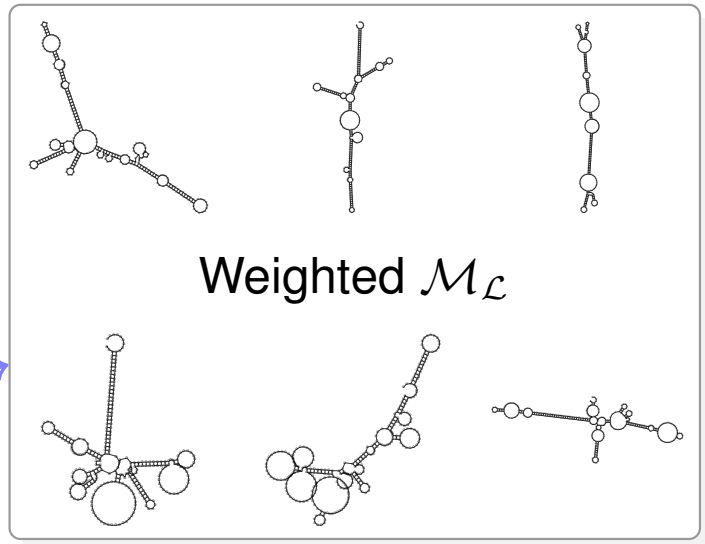


Application

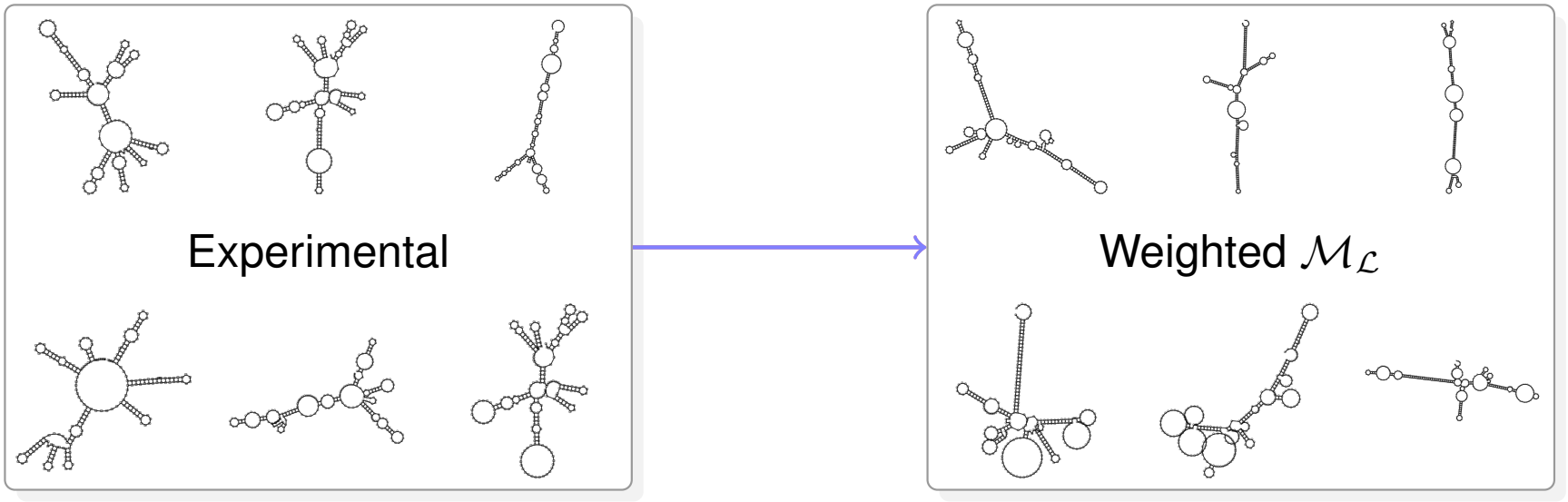






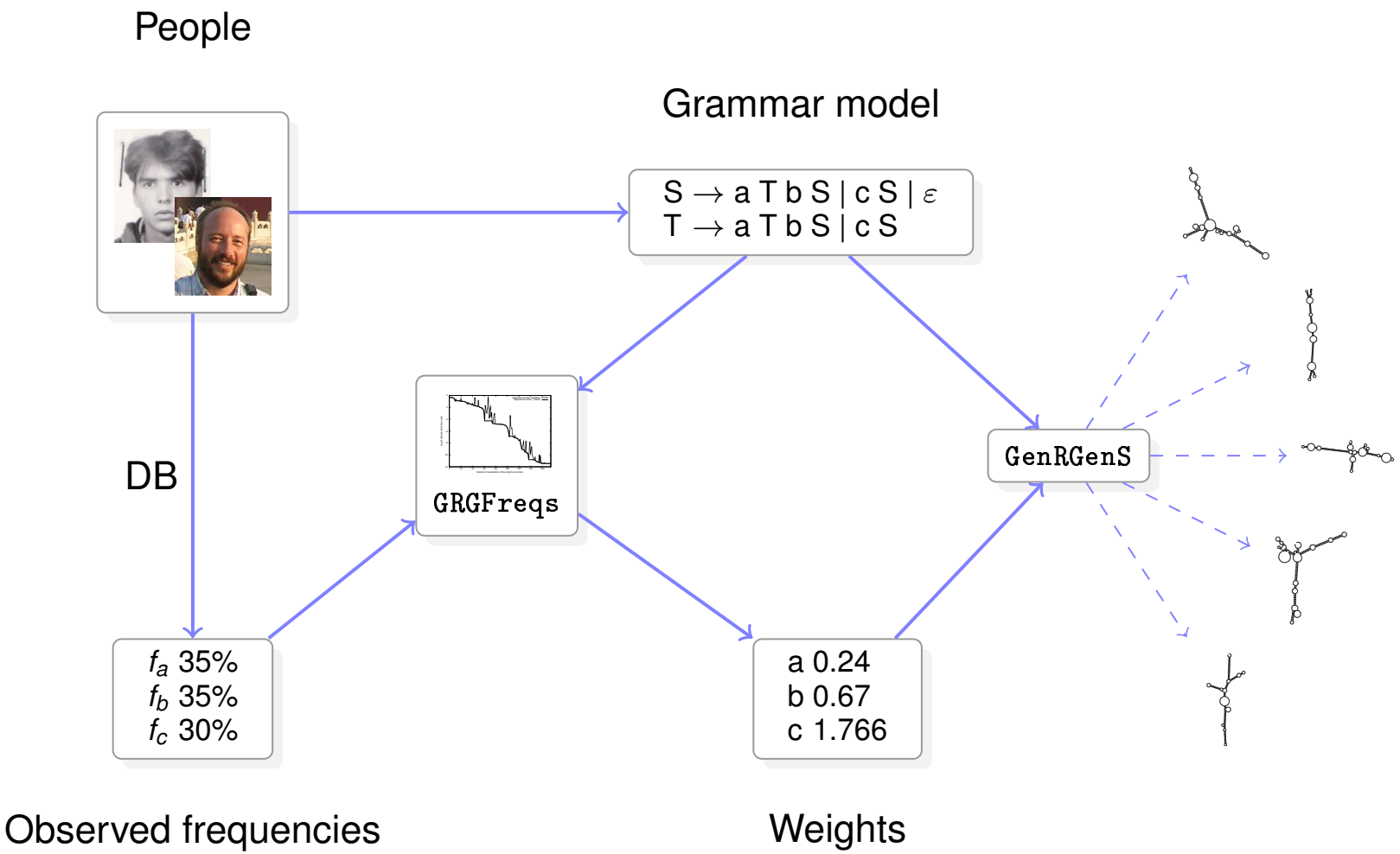


Application



Feature	B	b	I	i	M	m	T	t	H	h
Weight	1.15	1.15	1.15	1.15	2.75	1	1.15	1.15	.00308	1.25
$\mathcal{M}_0$ (%)	7.2	5.6	2.8	7.3	**3.7**	**7.6**	5.2	14.5	**18.6**	**27.5**
$\mathcal{M}_{\mathcal{L}}$ (%)	0.6	2.9	1.5	16.1	**1.1**	**9.0**	1.8	13.3	**4.8**	**48.9**
Exp.	1.5	2.3	1.9	11.2	**1.1**	**9.0**	2.6	16.6	**4.8**	**48.9**

Conclusion/Perspectives



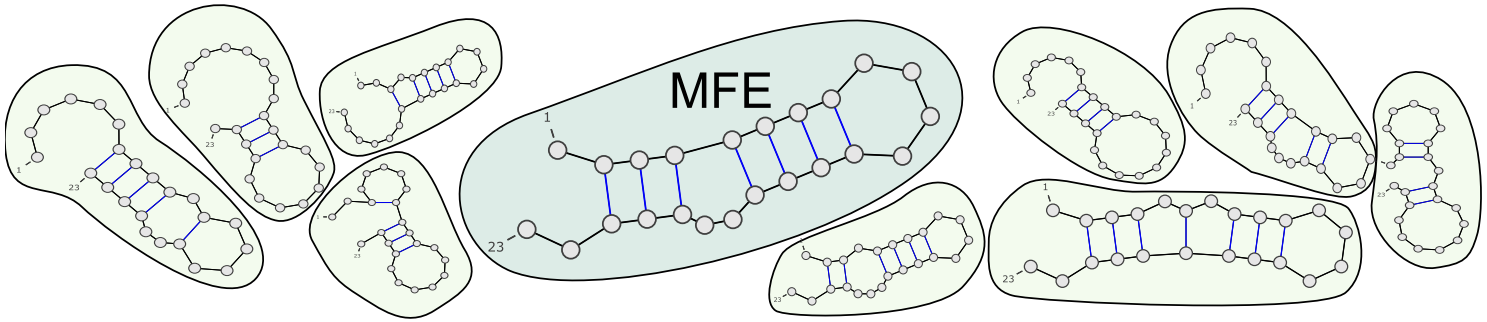
Now we can generate random structures fulfilling any desired properties.

Part IV

Improved statistical sampling

2004– : Boltzmann-ensemble paradigm

RNA *breathes* ⇒ There is not necessarily one functional conformation!



Boltzmann-ensemble paradigm

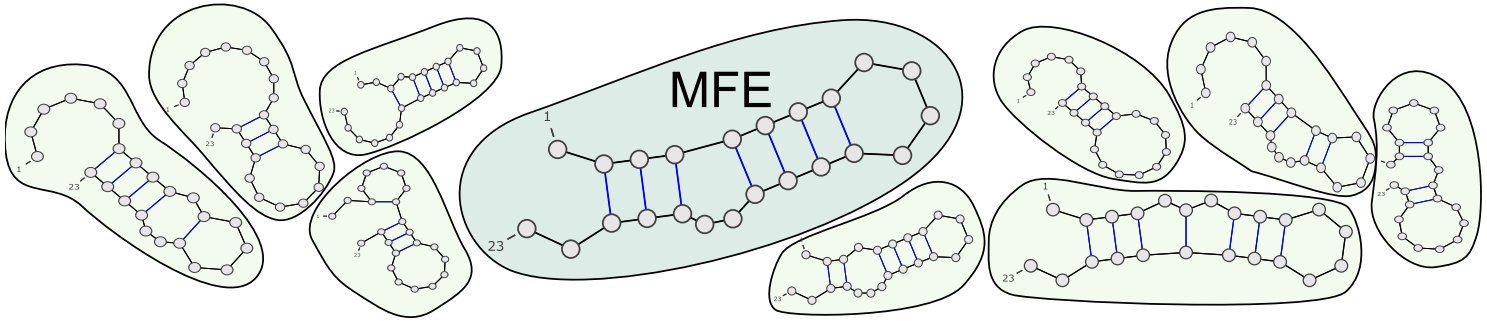
All possible secondary structures for an RNA coexist in a probability distribution, the Boltzmann distribution.

Consequently: MFE structure probability can be significantly remote (*unstructured RNAs?*).
Conversely, base-pairs having probability $> .99$ are valid 95% of the time.

- ⇒ Similar structures can bundle up.
- ⇒ Functional structures must be sought in sub-optimal structures.

2004– : Boltzmann-ensemble paradigm

RNA *breathes* ⇒ There is not necessarily one functional conformation!



Boltzmann-ensemble paradigm

All possible secondary structures for an RNA coexist in a probability distribution, the Boltzmann distribution.

Experiment: [DCL05]

- Sample structures from Boltzmann probability
- Cluster based on structural distance
- Build consensus structure based on most-probable cluster

⇒ Relative improvement for specificity (+17.6%) and sensitivity (+21.74%, except Group II Introns)

Boltzmann distribution: Definitions

A Boltzmann distribution assigns a weight to each structure S for an RNA ω, called the Boltzmann factor $e^{\frac{-E_{S,\omega}}{RT}}$ where:

- $E_{S,\omega}$ is the free-energy of S (kCal.mol^{-1})
- T is the temperature (K)
- R is the Boltzmann *constant* (1.986.10^{-3} kCal.K^{-1}.mol^{-1})

Distribution is renormalized on $\mathcal{S}_\omega$ by the partition function

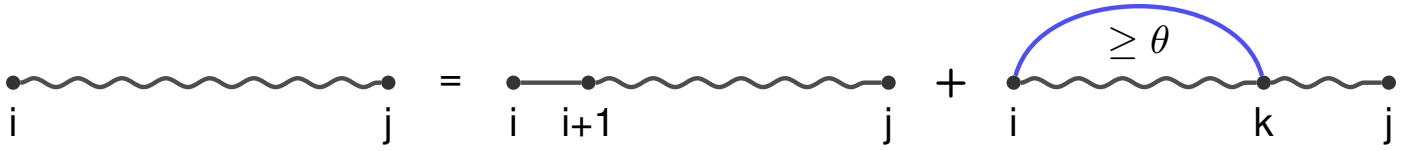
$$\mathcal{Z}_\omega = \sum_{S \in \mathcal{S}_\omega} e^{\frac{-E_{S,\omega}}{RT}}.$$

where $\mathcal{S}_\omega$ is the set of conformations compatible with ω.

The Boltzmann probability of a structure S is therefore

$$P_{S,\omega} = \frac{e^{\frac{-E_{S,\omega}}{RT}}}{\mathcal{Z}_\omega}.$$

Partition function



Recurrence on the minimal free-energy of a folding:

$$
\begin{aligned}
N_{i,t} &= 0, \quad \forall t \in [i, i+\theta] \\
N_{i,j} &= \min \begin{cases} N_{i+1,j} & (i \text{ unpaired}) \\ \min_{k=i+\theta+1}^{j} E_{i,j} + N_{i+1,k-1} + N_{k+1,j} & (i \text{ comp. with } k) \end{cases}
\end{aligned}
$$

+ Unambiguity

$\Rightarrow$ Recurrence on partition function :

$$
\begin{aligned}
\mathcal{Z}_{i,t} &= 1, \quad \forall t \in [i, i+\theta] \\
\mathcal{Z}_{i,j} &= \sum \begin{cases} \mathcal{Z}_{i+1,j} & (i \text{ unpaired}) \\ \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} & (i \text{ comp. with } k) \end{cases}
\end{aligned}
$$

+ Some slightly non-trivial computations. . .

= Base-pair probabilities in the Boltzmann ensemble [McC90].

Summands of $\mathcal{Z}$ $\Leftrightarrow$ Cumulated Boltzmann weight of all accessible structures.

$\Rightarrow$ Stochastic Backtrack recursively generates compatible structures with respect to the Boltzmann distribution [DL03].

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{Z}_{i,j} = \sum \begin{cases} \mathcal{Z}_{i+1,j} & \text{A} \\ \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} & \text{B}_k \end{cases}$$

$$A \,|\, B_{i+\theta+1} \,|\, B_{i+\theta+2} \,|\, B_{i+\theta+3} \,|\, \ldots \,|\, B_{j-2} \,|\, B_{j-1} \,|\, B_j$$

$\Rightarrow$ Worst-case complexity in $\mathcal{O}(n^2 k)$ for k samples.
Could we do better than that to sample more at the same computational cost?

Summands of $\mathcal{Z}$ $\Leftrightarrow$ Cumulated Boltzmann weight of all accessible structures.

$\Rightarrow$ Stochastic Backtrack recursively generates compatible structures with respect to the Boltzmann distribution [DL03].

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{Z}_{i,j} = \sum \begin{cases} ??? \longrightarrow \mathcal{Z}_{i+1,j} \\ ??? \longrightarrow \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{cases}$$

A

B_k

$$A \,|\, B_{i+\theta+1} \,|\, B_{i+\theta+2} \,|\, B_{i+\theta+3} \,|\, \ldots \,|\, B_{j-2} \,|\, B_{j-1} \,|\, B_j$$

$\Rightarrow$ Worst-case complexity in $\mathcal{O}(n^2 k)$ for k samples.

Could we do better than that to sample more at the same computational cost?

Summands of $\mathcal{Z}$ $\Leftrightarrow$ Cumulated Boltzmann weight of all accessible structures.

$\Rightarrow$ Stochastic Backtrack recursively generates compatible structures with respect to the Boltzmann distribution [DL03].

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{Z}_{i,j} = \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j}$$

$\mathcal{Z}_{i+1,j}/\mathcal{Z}_{i,j}$ → $\mathcal{Z}_{i+1,j}$

$\frac{e^{\frac{1}{RT}} \mathcal{Z}_{i+1,k-1} \mathcal{Z}_{k+1,j}}{\mathcal{Z}_{i,j}}$ → r

$A \,|\, B_{i+\theta+1} \,|\, B_{i+\theta+2} \,|\, B_{i+\theta+3} \,|\, \ldots \,|\, B_{j-2} \,|\, B_{j-1} \,|\, B_j$

A

B_k

$\Rightarrow$ Worst-case complexity in $\mathcal{O}(n^2 k)$ for k samples.
Could we do better than that to sample more at the same computational cost?

Summands of $\mathcal{Z}$ $\Leftrightarrow$ Cumulated Boltzmann weight of all accessible structures.

$\Rightarrow$ Stochastic Backtrack recursively generates compatible structures with respect to the Boltzmann distribution [DL03].

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{Z}_{i,j} = \sum \begin{cases} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{cases}$$

A

B_k

r

$$A \,|\, B_{i+\theta+1} \,|\, B_{i+\theta+2} \,|\, B_{i+\theta+3} \,|\, \ldots \,|\, B_{j-2} \,|\, B_{j-1} \,|\, B_j$$

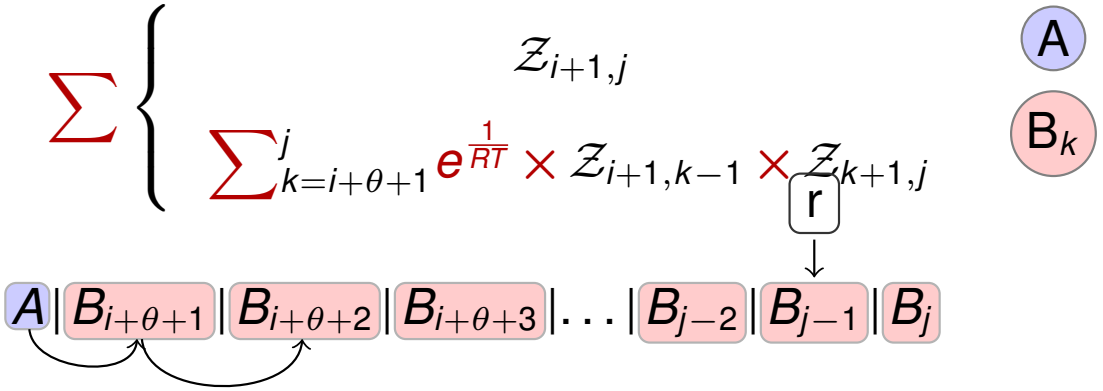
$\Rightarrow$ Worst-case complexity in $\mathcal{O}(n^2 k)$ for k samples.
Could we do better than that to sample more at the same computational cost?

Summands of $\mathcal{Z}$ $\Leftrightarrow$ Cumulated Boltzmann weight of all accessible structures.

$\Rightarrow$ Stochastic Backtrack recursively generates compatible structures with respect to the Boltzmann distribution [DL03].

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{Z}_{i,j} = \sum \begin{cases} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{cases}$$



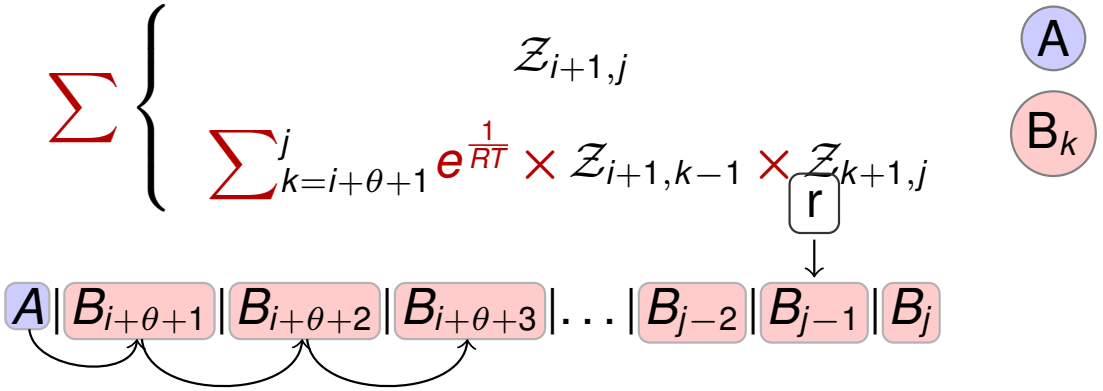
$\Rightarrow$ Worst-case complexity in $\mathcal{O}(n^2 k)$ for k samples.
Could we do better than that to sample more at the same computational cost?

Summands of $\mathcal{Z}$ $\Leftrightarrow$ Cumulated Boltzmann weight of all accessible structures.

$\Rightarrow$ Stochastic Backtrack recursively generates compatible structures with respect to the Boltzmann distribution [DL03].

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{Z}_{i,j} = \sum \begin{cases} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{cases}$$



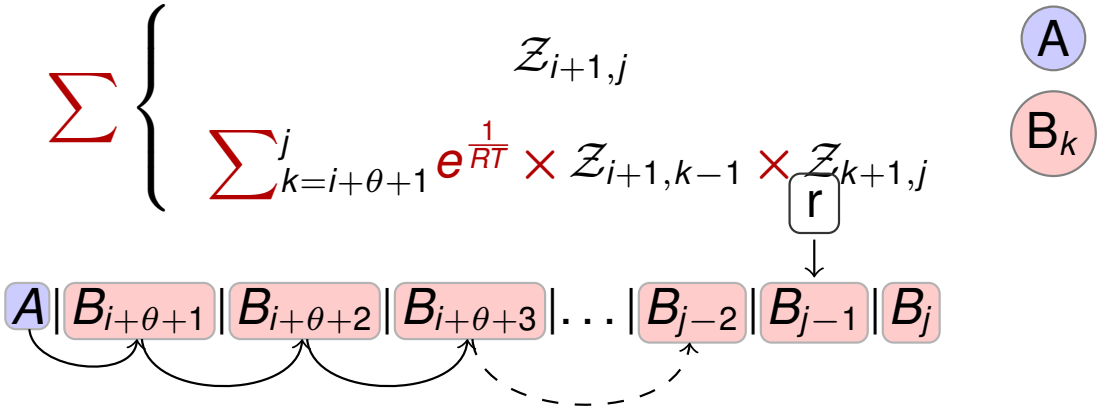
$\Rightarrow$ Worst-case complexity in $\mathcal{O}(n^2 k)$ for k samples.
Could we do better than that to sample more at the same computational cost?

Summands of $\mathcal{Z}$ $\Leftrightarrow$ Cumulated Boltzmann weight of all accessible structures.

$\Rightarrow$ Stochastic Backtrack recursively generates compatible structures with respect to the Boltzmann distribution [DL03].

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{Z}_{i,j} = \sum \begin{cases} \mathcal{Z}_{i+1,j} \\ \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \end{cases}$$



$\Rightarrow$ Worst-case complexity in $\mathcal{O}(n^2 k)$ for k samples.

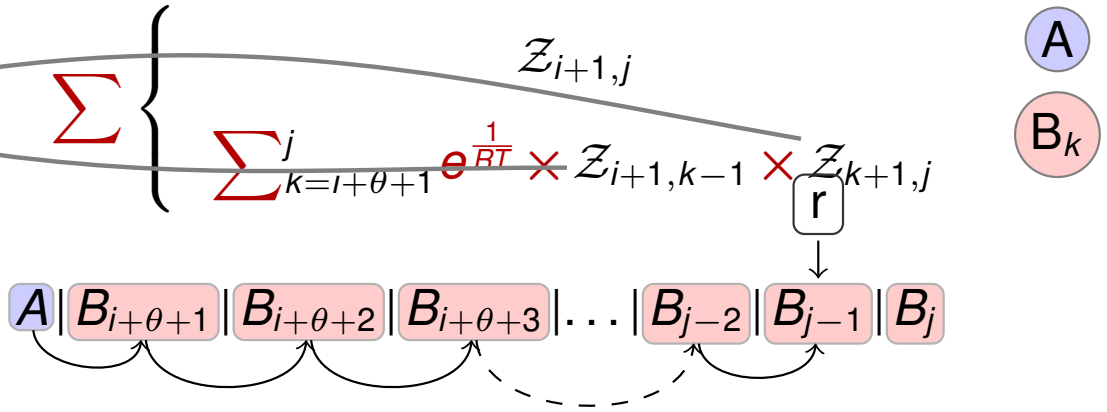
Could we do better than that to sample more at the same computational cost?

Summands of $\mathcal{Z}$ $\Leftrightarrow$ Cumulated Boltzmann weight of all accessible structures.

$\Rightarrow$ Stochastic Backtrack recursively generates compatible structures with respect to the Boltzmann distribution [DL03].

$$\mathcal{Z}_{i,t} = 1, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{Z}_{i,j} = \sum \left\{ \mathcal{Z}_{i+1,j} \quad \sum_{k=i+\theta+1}^{j} e^{\frac{1}{RT}} \times \mathcal{Z}_{i+1,k-1} \times \mathcal{Z}_{k+1,j} \right.$$



$\Rightarrow$ Worst-case complexity in $\mathcal{O}(n^2 k)$ for k samples.

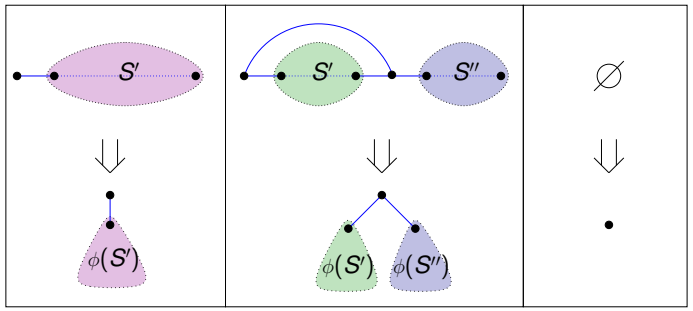
Could we do better than that to sample more at the same computational cost?

Average-case analysis

Homopolymer model: Any base-pair allowed.
Boltzmann distribution: Based on a Nussinov/Jacobson energy model.

Theorem

Let n be the RNA length et k the #samples.
The average-case complexity of statistical sampling is in $\Theta(n^3 + kn\sqrt{n})$.



Remark 1: Cost $c(S)$ of generating a structure S is dominated by the cumulated size of the leftward children $t \in \mathcal{T}_g(S)$ in the associated unary/binary tree:

$$c(S) = n - \#bp(S)/2 + \sum_{t \in \mathcal{T}_g(S)} |t| - \theta$$

Average-case analysis

Homopolymer model: Any base-pair allowed.
Boltzmann distribution: Based on a Nussinov/Jacobson energy model.

Theorem

Let n be the RNA length et k the #samples.
The average-case complexity of statistical sampling is in $\Theta(n^3 + kn\sqrt{n})$.

Let us focus on the expectation of the *generation cost* random variable X . We have

$$\mathbb{E}(X \mid n) = \sum_{|S|=n} c(S) \cdot \frac{e^{\frac{bp(S)}{RT}}}{\mathcal{Z}_n} = \frac{\sum_{|S|=n} c(S) \cdot e^{\frac{bp(S)}{RT}}}{\sum_{|S|=n} e^{\frac{bp(S)}{RT}}}.$$

Let us consider separately the two generating functions $C(z)$ and $P_f(z)$ such that

$$C(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} c(S) z^{|S|} \quad \text{et} \quad P_f(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} z^{|S|}.$$

Let us remark that $\mathbb{E}(X \mid n) = [z^n]C(z)/[z^n]P_f(z)$.

Average-case analysis

Homopolymer model: Any base-pair allowed.
Boltzmann distribution: Based on a Nussinov/Jacobson energy model.

Theorem

Let n be the RNA length et k the #samples.
The average-case complexity of statistical sampling is in $\Theta(n^3 + kn\sqrt{n})$.

$C(z)$ and $P_f(z)$ are the positive solutions of the system

$$\begin{aligned}
C(z) &= z\left(P_f(z) + C(z)\right) + z^2 e^{\frac{1}{RT}}(1-\theta)P_f^{\geq\theta}(z)P_f(z) \\
&+ z^3 e^{\frac{1}{RT}} \frac{\partial P_f^{\geq\theta}(z)}{\partial z} P_f(z) + z^2 e^{\frac{1}{RT}} C^{\geq\theta}(z)P_f(z) \\
&+ z^2 e^{\frac{1}{RT}} P_f^{\geq\theta}(z)C(z) \\
P_f(z) &= z^2 e^{\frac{1}{RT}} P_f^{\geq\theta}(z)P_f(z) + zP_f(z) + 1
\end{aligned}$$

where $S^{\geq\theta}(z)$ is the truncated gen. fun. $S(z)$, deprived of its terms of degrees $< \theta$.

Average-case analysis

Homopolymer model: Any base-pair allowed.
Boltzmann distribution: Based on a Nussinov/Jacobson energy model.

Theorem

Let n be the RNA length et k the #samples.
The average-case complexity of statistical sampling is in $\Theta(n^3 + kn\sqrt{n})$.

$\Rightarrow$ Big generating functions (Thanks to Maple !)...

+ Singularity analysis:

$$[z^n]P_f(z) \sim \frac{\kappa}{\rho^n n\sqrt{n}}(1 + \mathcal{O}(1/n)) \qquad [z^n]C(z) \sim \frac{\kappa'}{\rho^n}(1 + \mathcal{O}(1/\sqrt{n}))$$

We obtain an asymptotic equivalent for $\mathbb{E}(X \mid n)$:

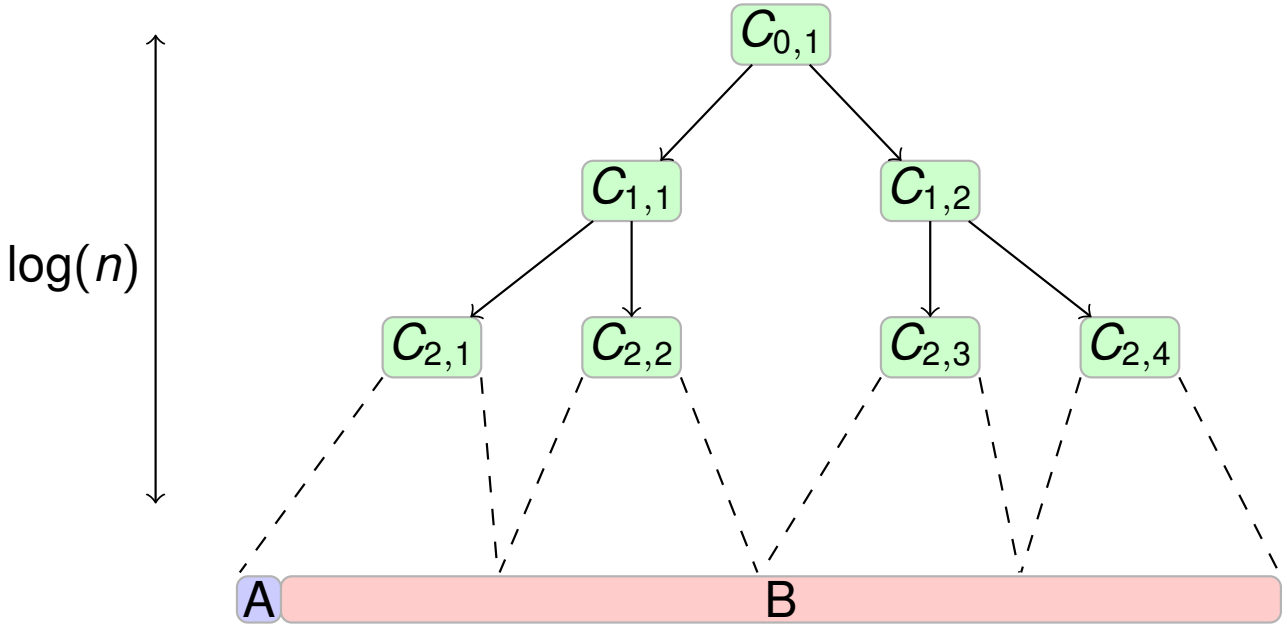
$$\mathbb{E}(X \mid n) = \frac{[z^n]C(z)}{[z^n]P_f(z)} \sim \frac{\kappa'}{\kappa} n\sqrt{n}(1 + \mathcal{O}(1/\sqrt{n})).$$

$\square$

Remark: Bivariate version of the 4 steps program...

Rationale: Most of the time is spent on testing unsuitable decompositions!

- Hierarchical organization of contributions



$\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$, memory in $\Theta(n^3)$.

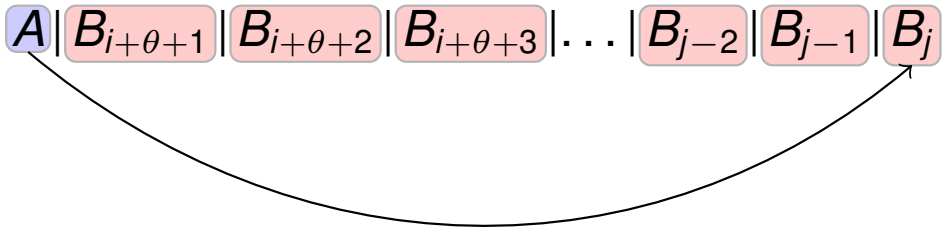
- Boustrophédon search

Unbalanced decompositions

$\Rightarrow$ Low *immediate* cost, recursion on intervals of lengths $\mathcal{O}(c)/\mathcal{O}(j\text{-}i\text{-}c)$.

Rationale: Most of the time is spent on testing unsuitable decompositions!

- Hierarchical organization of contributions
  $\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$, memory in $\Theta(n^3)$.
- Boustrophédon search



Unbalanced decompositions
$\Rightarrow$ Low *immediate* cost, recursion on intervals of lengths $\mathcal{O}(c)/\mathcal{O}(j\text{-}i\text{-}c)$.

Décomposition égale
$\Rightarrow$ *Immediate* cost is high ($\mathcal{O}(j\text{-}i\text{-}c)$), divide and conquer.

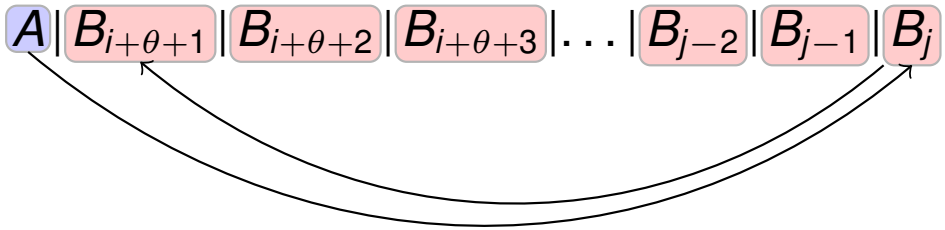
Worst-case complexity solution of

$$f(n) = \max_{k\in[1,n-1]}(f(k) + f(n-k) + 2\min(k, n-k)) \in \mathcal{O}(n\log(n)) \text{ [FZV94]}.$$

$\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$ without any precomputation.

Rationale: Most of the time is spent on testing unsuitable decompositions!

- Hierarchical organization of contributions
  $\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$, memory in $\Theta(n^3)$.
- Boustrophédon search



Unbalanced decompositions
$\Rightarrow$ Low *immediate* cost, recursion on intervals of lengths $\mathcal{O}(c)/\mathcal{O}(j\text{-}i\text{-}c)$.

Décomposition égale
$\Rightarrow$ *Immediate* cost is high ($\mathcal{O}(j\text{-}i\text{-}c)$), divide and conquer.

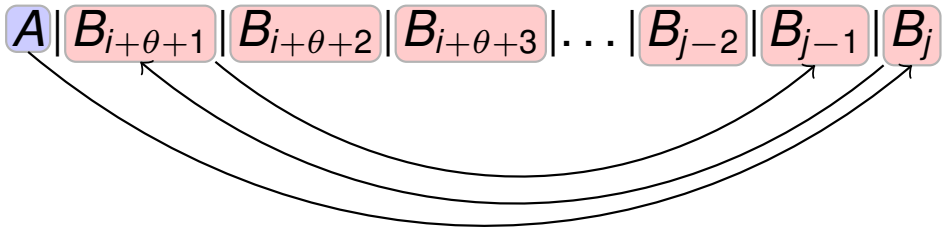
Worst-case complexity solution of

$$f(n) = \max_{k\in[1,n-1]}(f(k) + f(n-k) + 2\min(k, n-k)) \in \mathcal{O}(n\log(n))$$ [FZV94].

$\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$ without any precomputation.

Rationale: Most of the time is spent on testing unsuitable decompositions!

- Hierarchical organization of contributions
  $\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$, memory in $\Theta(n^3)$.
- Boustrophédon search



Unbalanced decompositions
$\Rightarrow$ Low *immediate* cost, recursion on intervals of lengths $\mathcal{O}(c)/\mathcal{O}(j\text{-}i\text{-}c)$.

Décomposition égale
$\Rightarrow$ *Immediate* cost is high ($\mathcal{O}(j\text{-}i\text{-}c)$), divide and conquer.

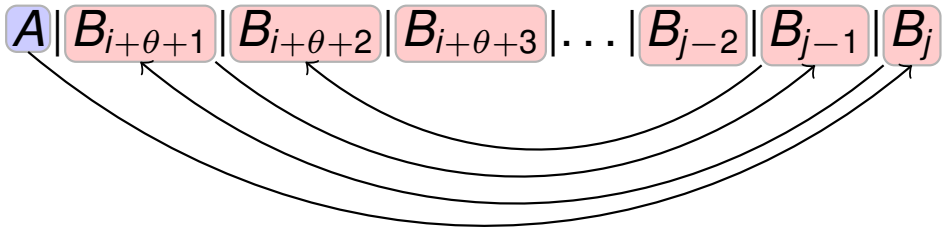
Worst-case complexity solution of

$$f(n) = \max_{k\in[1,n-1]}(f(k) + f(n-k) + 2\min(k, n-k)) \in \mathcal{O}(n\log(n)) \text{ [FZV94]}.$$

$\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$ without any precomputation.

Rationale: Most of the time is spent on testing unsuitable decompositions!

- Hierarchical organization of contributions
  $\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$, memory in $\Theta(n^3)$.
- Boustrophédon search



Unbalanced decompositions
$\Rightarrow$ Low *immediate* cost, recursion on intervals of lengths $\mathcal{O}(c)/\mathcal{O}(j\text{-}i\text{-}c)$.

Décomposition égale
$\Rightarrow$ *Immediate* cost is high ($\mathcal{O}(j\text{-}i\text{-}c)$), divide and conquer.

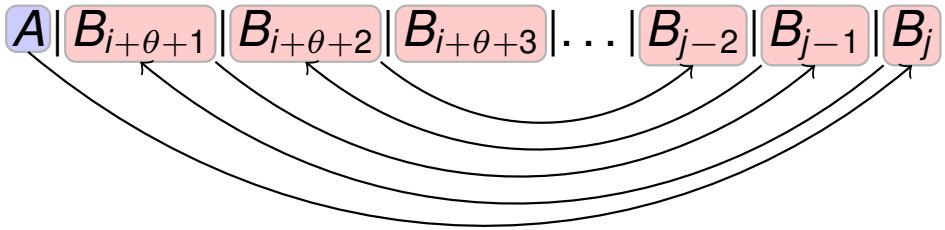
Worst-case complexity solution of

$$f(n) = \max_{k\in[1,n-1]}(f(k) + f(n-k) + 2\min(k, n-k)) \in \mathcal{O}(n\log(n))$$ [FZV94].

$\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$ without any precomputation.

Rationale: Most of the time is spent on testing unsuitable decompositions!

- Hierarchical organization of contributions
  $\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$, memory in $\Theta(n^3)$.
- Boustrophédon search



Unbalanced decompositions
$\Rightarrow$ Low *immediate* cost, recursion on intervals of lengths $\mathcal{O}(c)/\mathcal{O}(j\text{-}i\text{-}c)$.

Décomposition égale
$\Rightarrow$ *Immediate* cost is high ($\mathcal{O}(j\text{-}i\text{-}c)$), divide and conquer.

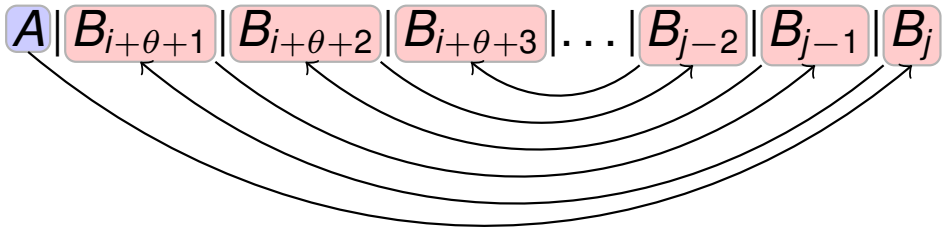
Worst-case complexity solution of

$$f(n) = \max_{k\in[1,n-1]}(f(k) + f(n-k) + 2\min(k, n-k)) \in \mathcal{O}(n\log(n))$$ [FZV94].

$\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$ without any precomputation.

Rationale: Most of the time is spent on testing unsuitable decompositions!

- Hierarchical organization of contributions
  $\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$, memory in $\Theta(n^3)$.
- Boustrophédon search



Unbalanced decompositions
$\Rightarrow$ Low *immediate* cost, recursion on intervals of lengths $\mathcal{O}(c)/\mathcal{O}(j\text{-}i\text{-}c)$.

Décomposition égale
$\Rightarrow$ *Immediate* cost is high ($\mathcal{O}(j\text{-}i\text{-}c)$), divide and conquer.

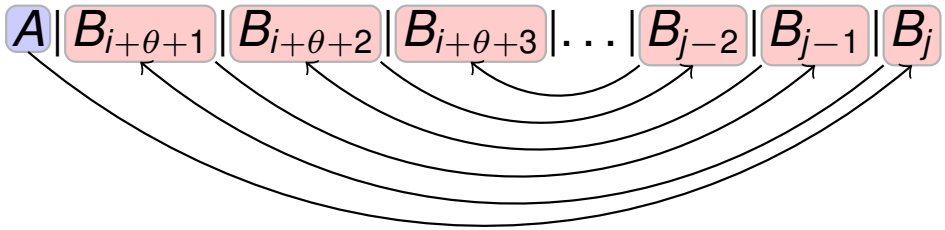
Worst-case complexity solution of

$$f(n) = \max_{k\in[1,n-1]}(f(k) + f(n-k) + 2\min(k, n-k)) \in \mathcal{O}(n\log(n)) \text{ [FZV94]}.$$

$\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$ without any precomputation.

Rationale: Most of the time is spent on testing unsuitable decompositions!

- Hierarchical organization of contributions
  $\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$, memory in $\Theta(n^3)$.
- Boustrophédon search



Unbalanced decompositions
$\Rightarrow$ Low *immediate* cost, recursion on intervals of lengths $\mathcal{O}(c)/\mathcal{O}(j\text{-}i\text{-}c)$.

Décomposition égale
$\Rightarrow$ *Immediate* cost is high ($\mathcal{O}(j\text{-}i\text{-}c)$), divide and conquer.

Worst-case complexity solution of

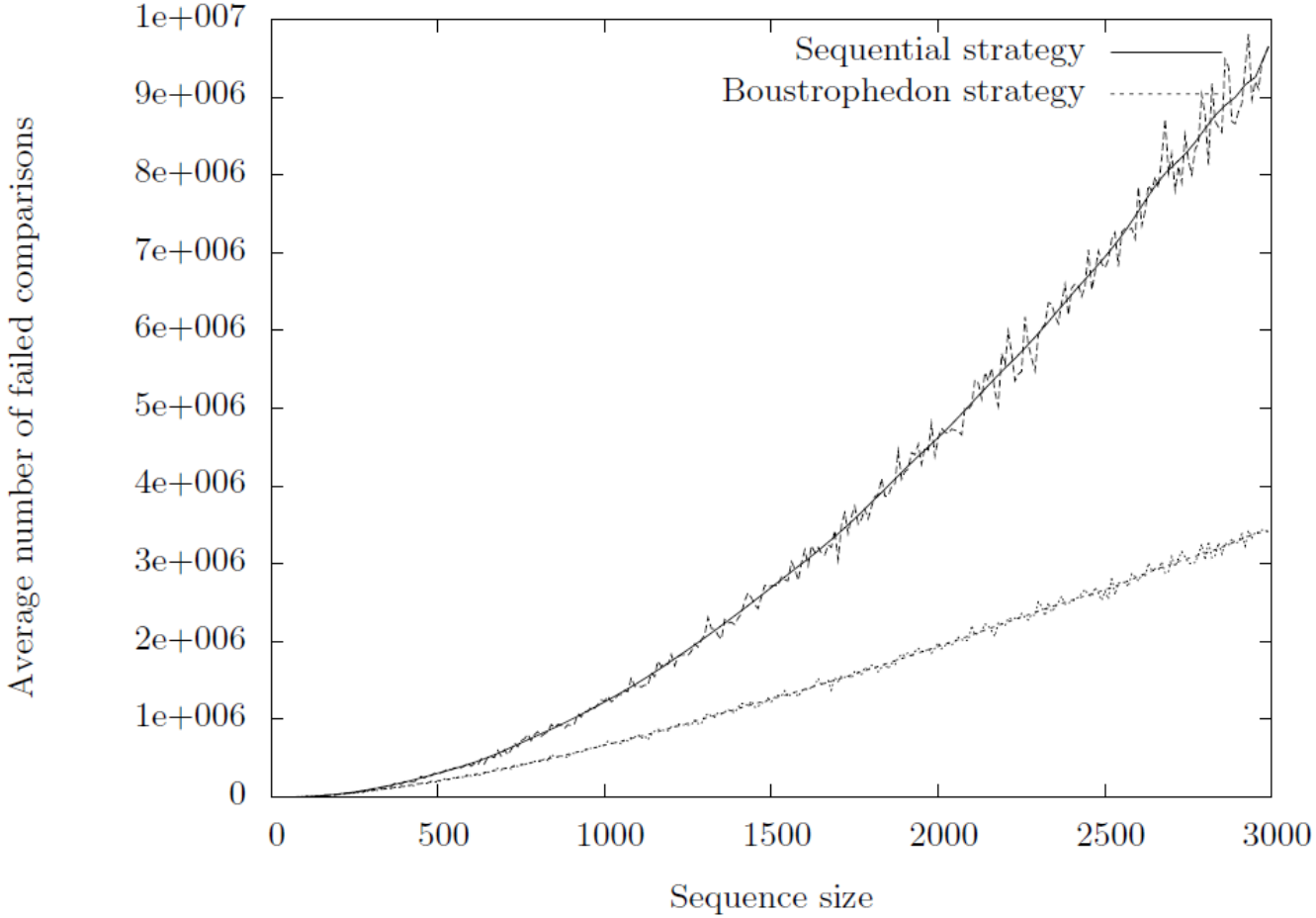
$$f(n) = \max_{k\in[1,n-1]}(f(k) + f(n-k) + 2\min(k, n-k)) \in \mathcal{O}(n\log(n))$$ [FZV94].

$\Rightarrow$ Generation in $\mathcal{O}(n^3 + kn\log(n))$ without any precomputation.

Results

- Implementation in `Unafold` (successor of `MFold`) using full Turner model.
- Random sequences: Markov order 1 model based on a 5s rRNA.

Implementation in `Unafold` (successor of `MFold`).



Part V

Conclusion

Summary

Using enumerative (models) and analytic (asymptotics) techniques, one can:

- Accurately enumerate RNA Shapes [JCB08].

- Improved random models for structures [Bioinf.06, TCS10].
- Analyze and optimize statistical sampling algorithms [JMB08].

- Analyze Boltzmann samples sets of structures [GASCOM10].

- Traverse the full mutational landscape [Recomb11].

- Explore enriched conformational spaces (PKs, MC-Fold) [...].

- Address constrained RNA design problems [...].

Summary

Using enumerative (models) and analytic (asymptotics) techniques, one can:

- Accurately enumerate RNA Shapes [JCB08].

- Improved random models for structures [Bioinf.06, TCS10].

- Analyze and optimize statistical sampling algorithms [JMB08].
- Analyze Boltzmann samples sets of structures [GASCOM10].

- Traverse the full mutational landscape [Recomb11].

- Explore enriched conformational spaces (PKs, MC-Fold) [...].

- Address constrained RNA design problems [...].

Summary

Using enumerative (models) and analytic (asymptotics) techniques, one can:

- Accurately enumerate RNA Shapes [JCB08].

- Improved random models for structures [Bioinf.06, TCS10].
- Analyze and optimize statistical sampling algorithms [JMB08].

- Analyze Boltzmann samples sets of structures [GASCOM10].

- Traverse the full mutational landscape [Recomb11].

- Explore enriched conformational spaces (PKs, MC-Fold) [...].

- Address constrained RNA design problems [...].

Summary

Using enumerative (models) and analytic (asymptotics) techniques, one can:

- Accurately enumerate RNA Shapes [JCB08].

- Improved random models for structures [Bioinf.06, TCS10].
- Analyze and optimize statistical sampling algorithms [JMB08].

- Analyze Boltzmann samples sets of structures [GASCOM10].

- Traverse the full mutational landscape [Recomb11].

- Explore enriched conformational spaces (PKs, MC-Fold) [...].

- Address constrained RNA design problems [...].

Summary

Using enumerative (models) and analytic (asymptotics) techniques, one can:

- Accurately enumerate RNA Shapes [JCB08].

- Improved random models for structures [Bioinf.06, TCS10].
- Analyze and optimize statistical sampling algorithms [JMB08].

- Analyze Boltzmann samples sets of structures [GASCOM10].

- Traverse the full mutational landscape [Recomb11].

- Explore enriched conformational spaces (PKs, MC-Fold) [...].

- Address constrained RNA design problems [...].

Summary

Using enumerative (models) and analytic (asymptotics) techniques, one can:

- Accurately enumerate RNA Shapes [JCB08].

- Improved random models for structures [Bioinf.06, TCS10].
- Analyze and optimize statistical sampling algorithms [JMB08].

- Analyze Boltzmann samples sets of structures [GASCOM10].

- Traverse the full mutational landscape [Recomb11].

- Explore enriched conformational spaces (PKs, MC-Fold) [...].

- Address constrained RNA design problems [...].

Summary

Using enumerative (models) and analytic (asymptotics) techniques, one can:

- Accurately enumerate RNA Shapes [JCB08].

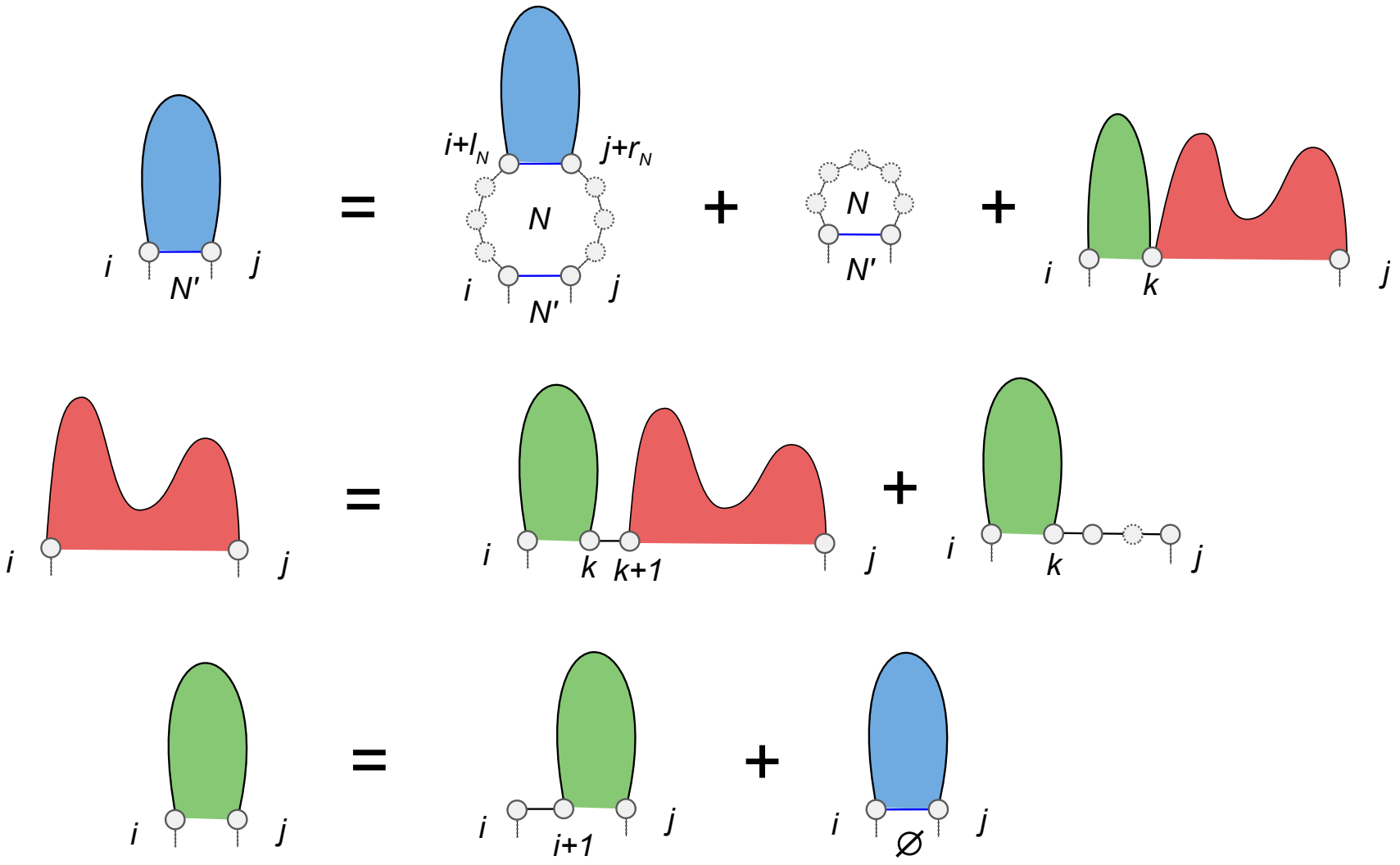
- Improved random models for structures [Bioinf.06, TCS10].

- Analyze and optimize statistical sampling algorithms [JMB08].
- Analyze Boltzmann samples sets of structures [GASCOM10].

- Traverse the full mutational landscape [Recomb11].

- Explore enriched conformational spaces (PKs, MC-Fold) [...].

- Address constrained RNA design problems [...].


i
j
N'
=
i+l_N
j+r_N
N
i
j
N'
+
N
N'
+
i
k
j
i
j
=
i
k
k+1
j
+
i
k
j
i
j
=
i
i+1
j
+
i
∅
j

References I

F. Vanden Berghen.
CONDOR: a constrained, non-linear,derivative-free parallel optimizer for continuous, high computing load, noisy objective functions.
PhD thesis, IRIDIA, Universite Libre de Belgique, 2005.

Y. Ding, C. Y. Chan, and C. E. Lawrence.
RNA secondary structure prediction by centroids in a boltzmann weighted ensemble.
RNA, 11:1157–1166, 2005.

Y. Ding and E. Lawrence.
A statistical sampling algorithm for RNA secondary structure prediction.
Nucleic Acids Research, 31(24):7280–7301, 2003.

A. Denise, O. Roques, and M. Termier.
Random generation of words of context-free languages according to the frequencies of letters.
In D. Gardy and A. Mokkadem, editors, *Mathematics and Computer Science: Algorithms, Trees, Combinatorics and probabilities*, Trends in Mathematics, pages 113–125. Birkhaüser, 2000.

A. Denise and P. Zimmermann.
Uniform random generation of decomposable structures using floating-point arithmetic.
Theor. Comput. Sci., 218(2):233–248, 1999.

P. Flajolet, P. Zimmermann, and B. Van Cutsem.
Calculus for the random generation of labelled combinatorial structures.
Theoretical Computer Science, 132:1–35, 1994.

J.S. McCaskill.
The equilibrium partition function and base pair binding probabilities for RNA secondary structure.
Biopolymers, 29:1105–1119, 1990.

R. Nussinov and A.B. Jacobson.
Fast algorithm for predicting the secondary structure of single-stranded RNA.
Proc Natl Acad Sci U S A, 77:6903–13, 1980.

References II

M. S. Waterman.
Secondary structure of single stranded nucleic acids.
Advances in Mathematics Supplementary Studies, 1(1):167–212, 1978.