Statistical sampling and rational design of RNA

Yann Ponty

Analytical Genomics
A. Carbone's lab
INSERM U511/Paris 6
Paris, France

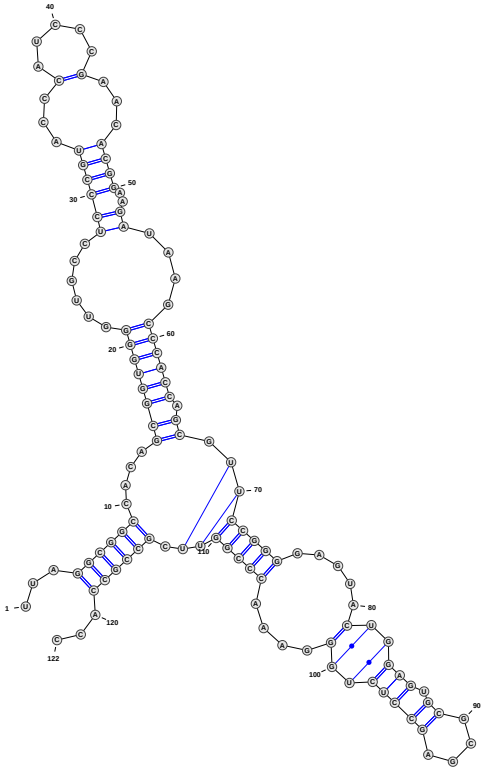
October 8, 2008

RNA structure

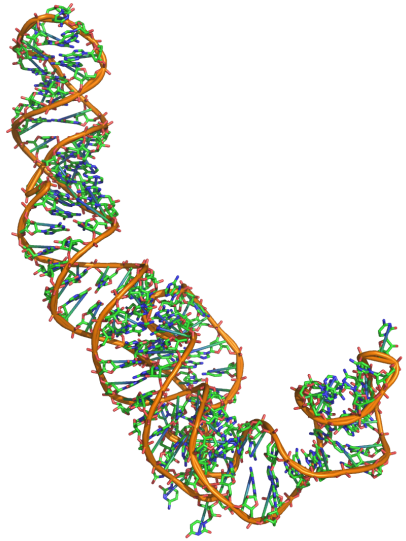
Three[1] levels of detail:

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Primary structure



Secondary structure



Tertiary structure

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

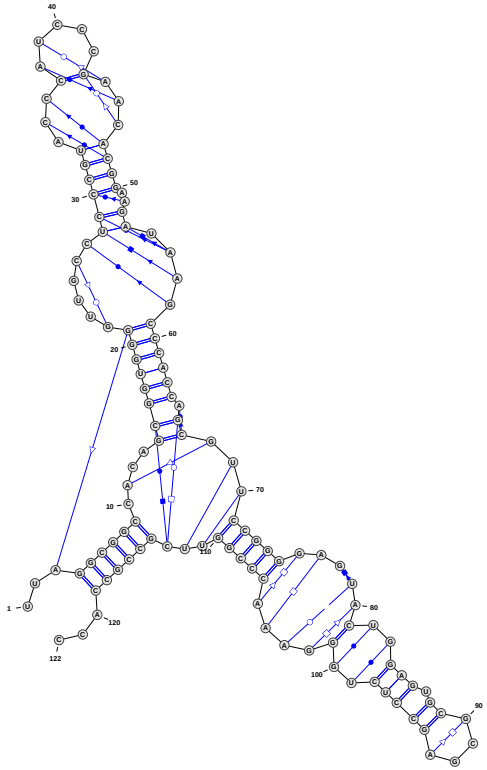
[1] Well, almost ...

RNA structure

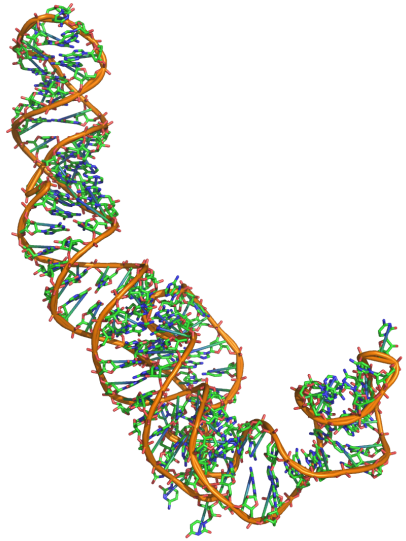
Three[1] levels of detail:

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Primary structure



Secondary^{+} structure

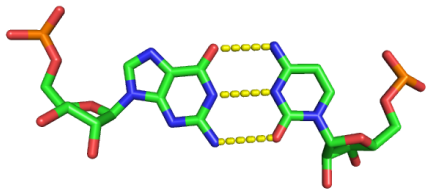


Tertiary structure

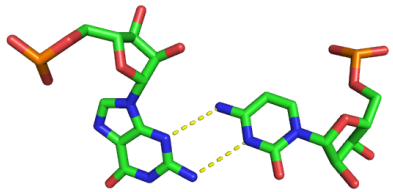
Source: 5s rRNA (PDBID: 1K73:B)

[1] Well, almost ...

- Non-canonical base-pairings:
  Base-pairs other than {(A-U), (C-G), (G-U)}
  Or interacting in a non WC/WC-Cis way [LW01].

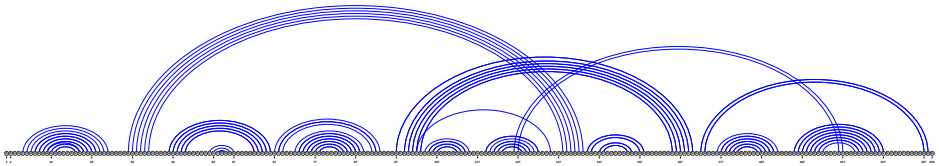


Canonical CG (WC/WC-Cis)



Non-canonical CG (Sugar/WC-Trans)

- Pseudoknots:



Pseudoknotted structure of a Group I Ribobozyme (PDBID: 1Y0Q:A)

Finding the best pseudoknotted structure:
$\Rightarrow$ NP-Complete [LP00] ...
... but polynomial for restricted classes [CDR+04].

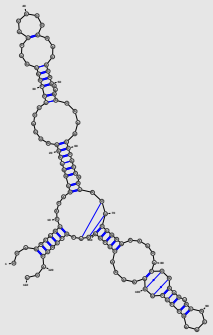
State of the art

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Sequence

Size Sens
<700 70-73%
[MSZT99, DCCG04]
16s,23s ∼50%
MCC∼ 0.5 [GG04]
Thermodynamics

Sens Spe MCC
∼75% ∼75% 0.8
Comparative [GG04]



2^{ary} structure

Reminder: $MCC = \frac{t^+ t^- - f^+ f^-}{\sqrt{(t^+ + f^+)(t^+ + f^-)(t^- + f^+)(t^- + f^-)}}$

State of the art

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Sequence

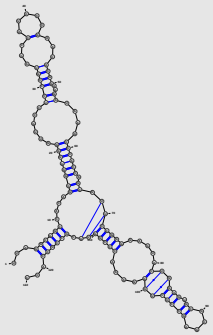
Size	Sens
<700	70-73%

[MSZT99, DCCG04]

16s,23s ∼50%

MCC∼ 0.5 [GG04]

Thermodynamics



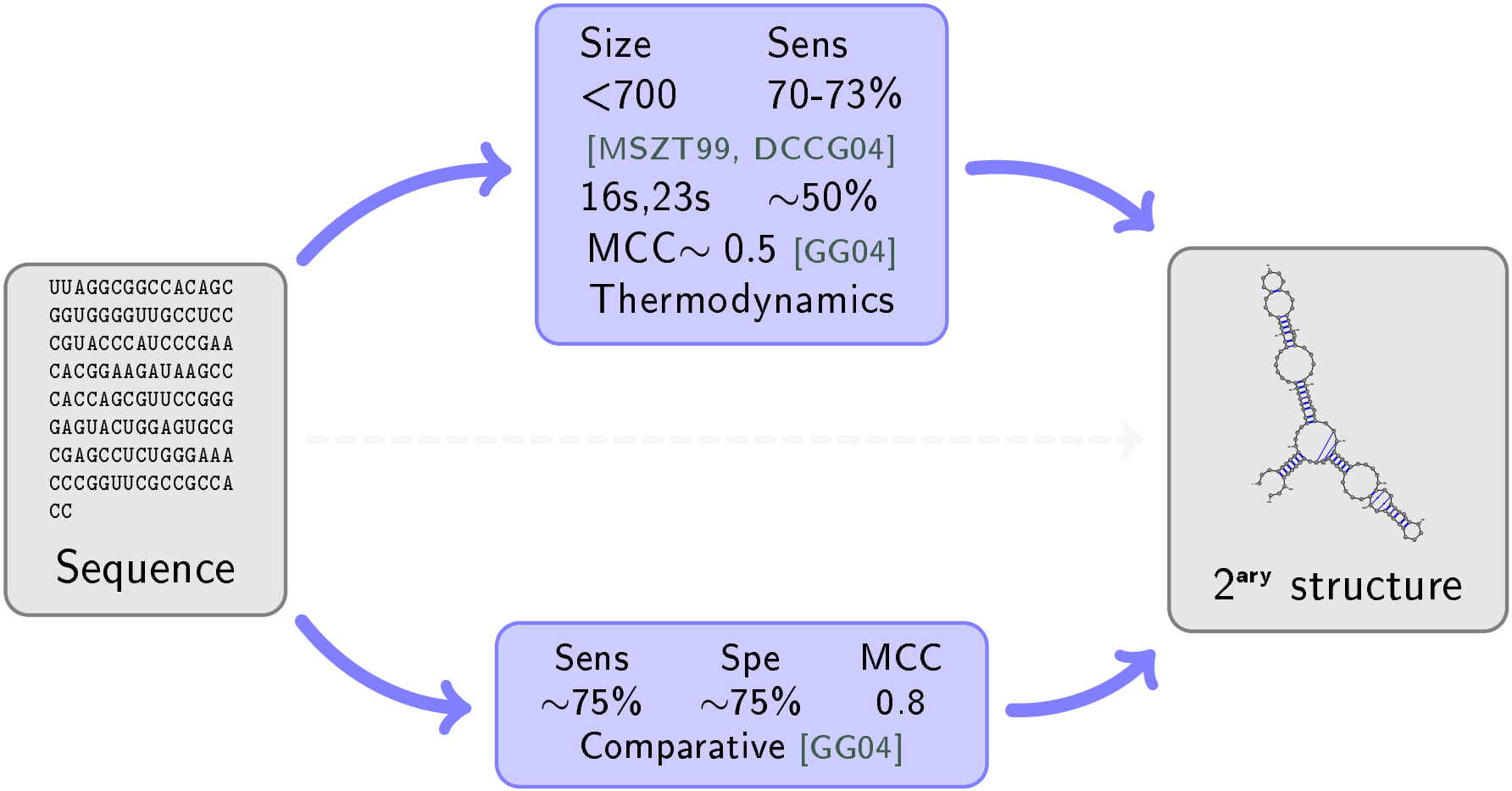
2^{ary} structure

Sens	Spe	MCC
∼75%	∼75%	0.8

Comparative [GG04]

Reminder: $MCC = \frac{t^+ t^- - f^+ f^-}{\sqrt{(t^+ + f^+)(t^+ + f^-)(t^- + f^+)(t^- + f^-)}}$

State of the art



Reminder: $MCC = \frac{t^+ t^- - f^+ f^-}{\sqrt{(t^+ + f^+)(t^+ + f^-)(t^- + f^+)(t^- + f^-)}}$

The future: Toward 3D automated prediction

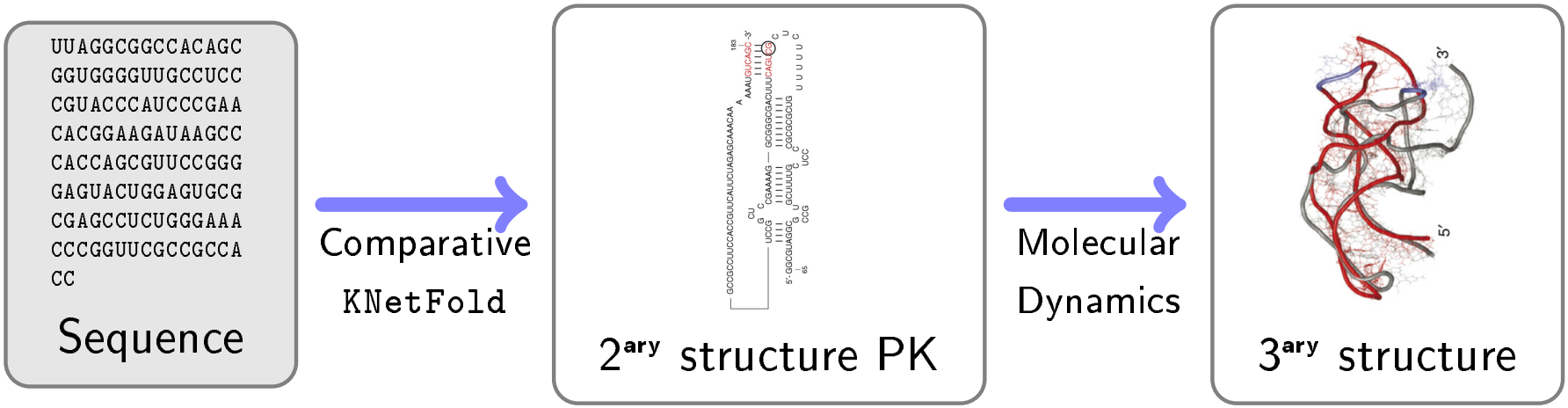
Goal: Go straight from sequence to 3D models!!!

- Comparative modeling + Molecular Dynamics: `RNA2D3D` [SYKB07]
- `MC-Fold/MC-sym` pipeline [PM08]

The future: Toward 3D automated prediction

Goal: Go straight from sequence to 3D models!!!

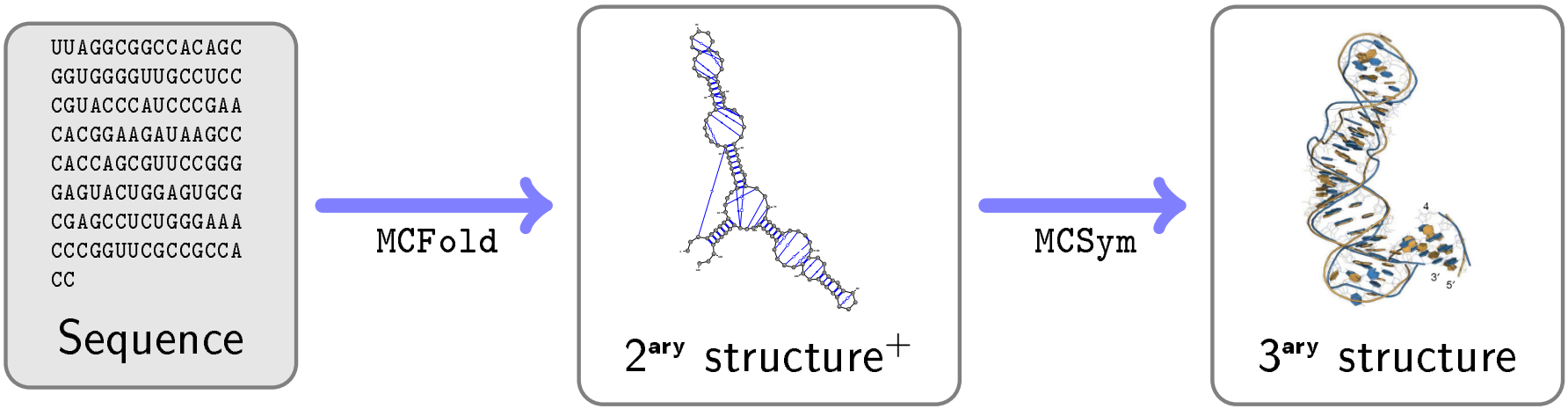
- Comparative modeling + Molecular Dynamics: `RNA2D3D` [SYKB07]
- `MC-Fold/MC-sym` pipeline [PM08]



The future: Toward 3D automated prediction

Goal: Go straight from sequence to 3D models!!!

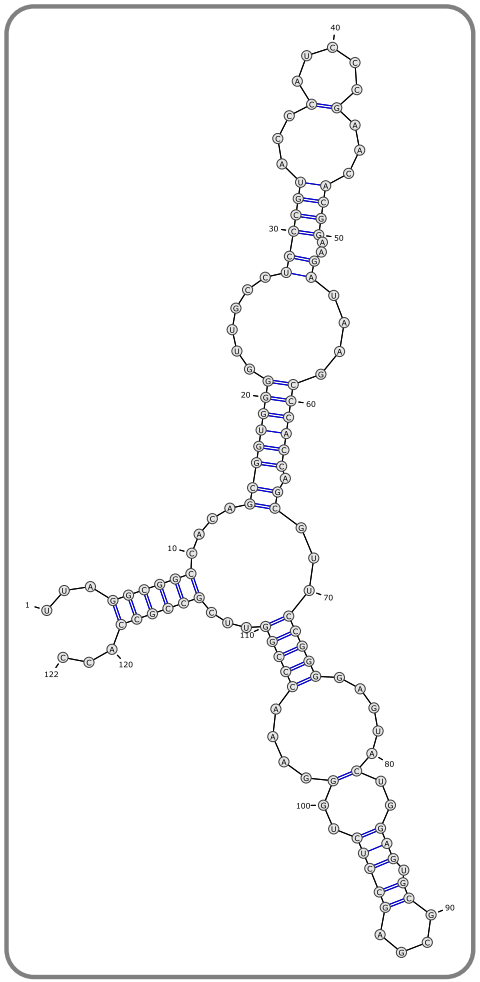
- Comparative modeling + Molecular Dynamics: RNA2D3D [SYKB07]
- MC-Fold/MC-sym pipeline [PM08]



Turner model

Relies on an unambiguous decomposition of 2^{ary} into a set of loops:

- Interior Loops
- Bulges
- Terminal Loops, aka Hairpin Loops
- Multiple Loops
- Stacking pairs

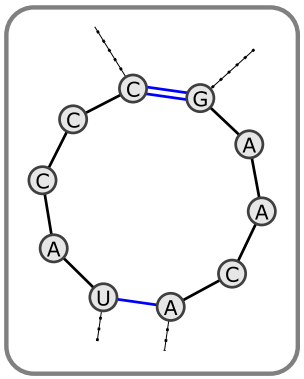


Experimentally-determined values and interpolations for individual contributions of loops.

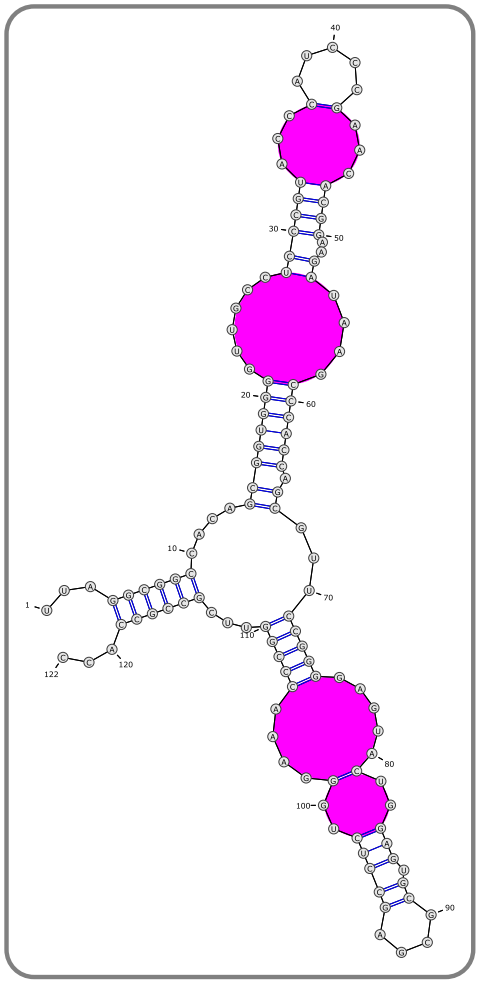
Turner model

Relies on an unambiguous decomposition of 2^{ary} into a set of loops:

- Interior Loops
- Bulges
- Terminal Loops, aka Hairpin Loops
- Multiple Loops
- Stacking pairs



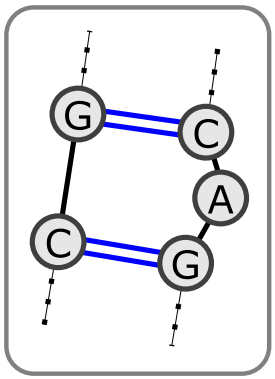
Experimentally-determined values and interpolations for individual contributions of loops.

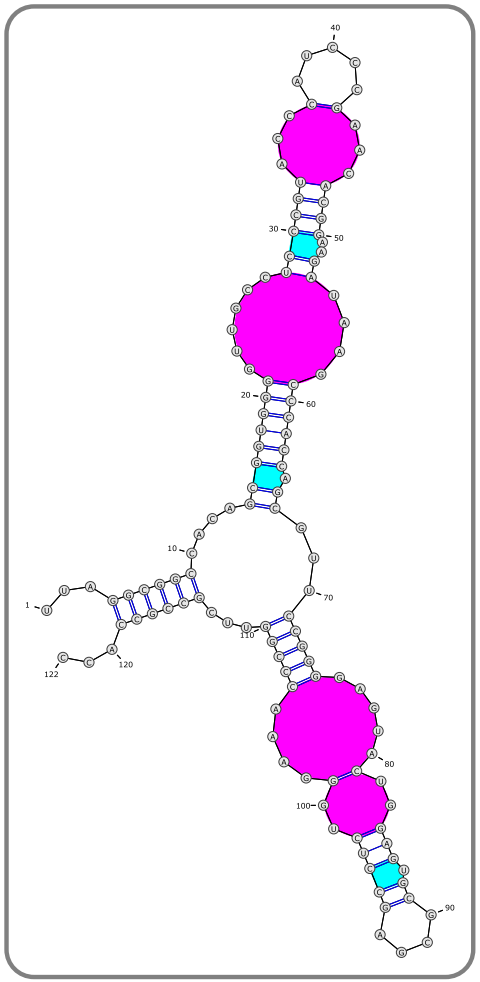


Turner model

Relies on an unambiguous decomposition of 2^{ary} into a set of loops:

- Interior Loops
- Bulges
- Terminal Loops, aka Hairpin Loops
- Multiple Loops
- Stacking pairs



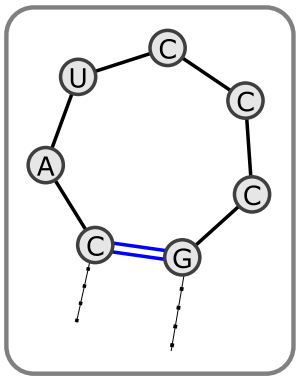


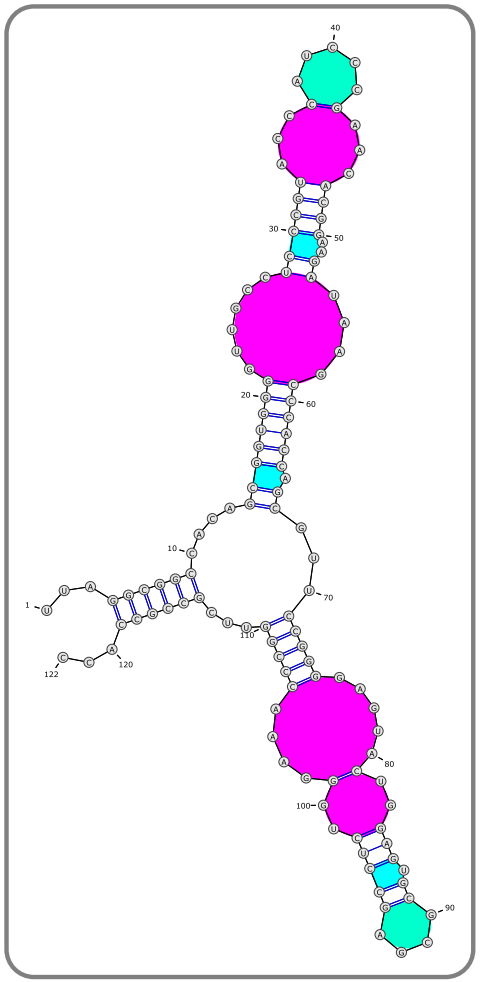
Experimentally-determined values and interpolations for individual contributions of loops.

Turner model

Relies on an unambiguous decomposition of 2^{ary} into a set of loops:

- Interior Loops
- Bulges
- Terminal Loops, aka Hairpin Loops
- Multiple Loops
- Stacking pairs



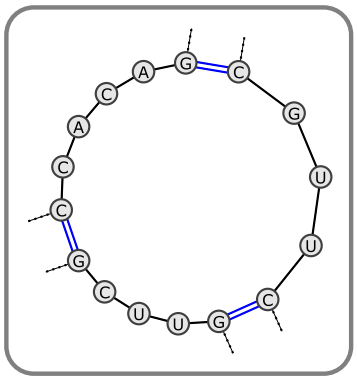


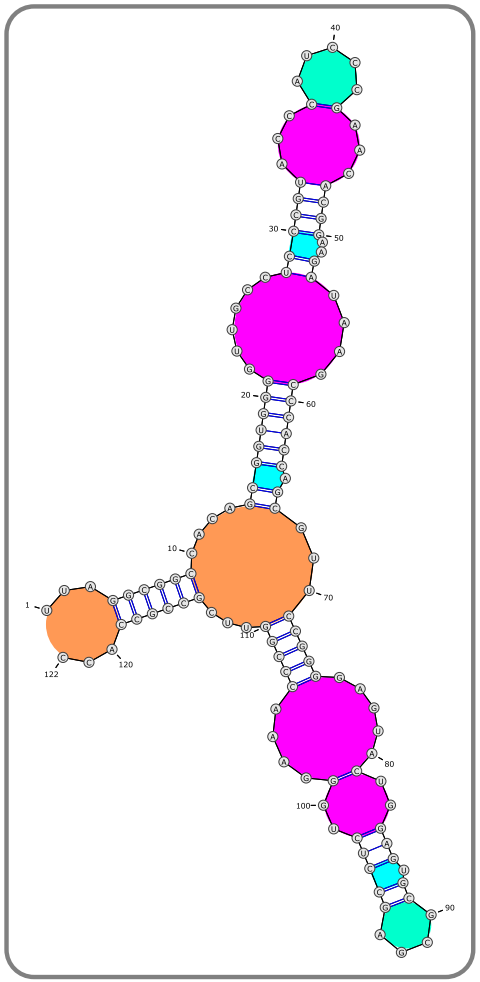
Experimentally-determined values and interpolations for individual contributions of loops.

Turner model

Relies on an unambiguous decomposition of 2^{ary} into a set of loops:

- Interior Loops
- Bulges
- Terminal Loops, aka Hairpin Loops
- Multiple Loops
- Stacking pairs



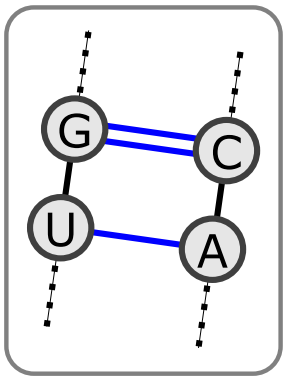


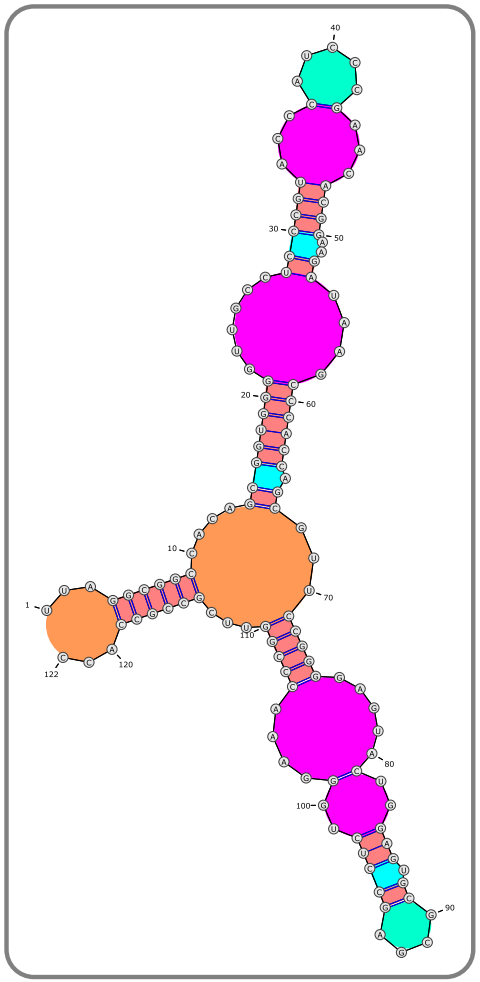
Experimentally-determined values and interpolations for individual contributions of loops.

Turner model

Relies on an unambiguous decomposition of 2^{ary} into a set of loops:

- Interior Loops
- Bulges
- Terminal Loops, aka Hairpin Loops
- Multiple Loops
- Stacking pairs





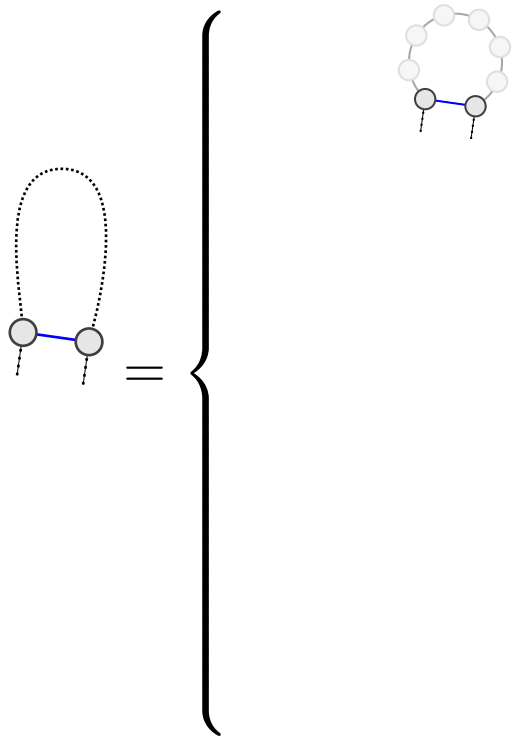
Experimentally-determined values and interpolations for individual contributions of loops.

MFE folding

Theorem (MFE hypothesis)

RNA folds into its minimum free-energy conformation

But $\sim 1.8^n$ secondary structures compatible with S of size n [ZS84].
$\Rightarrow$ Dynamic programming, i.e. enumeration

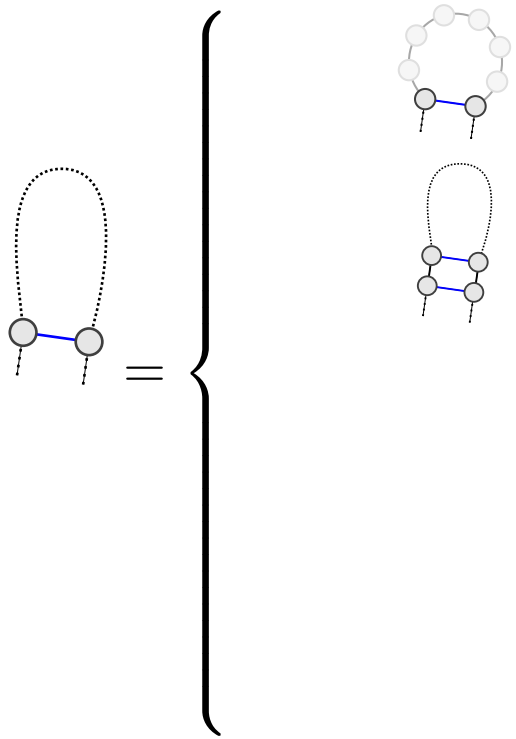


MFE folding

Theorem (MFE hypothesis)

RNA folds into its minimum free-energy conformation

But $\sim 1.8^n$ secondary structures compatible with S of size n [ZS84].
$\Rightarrow$ Dynamic programming, i.e. enumeration

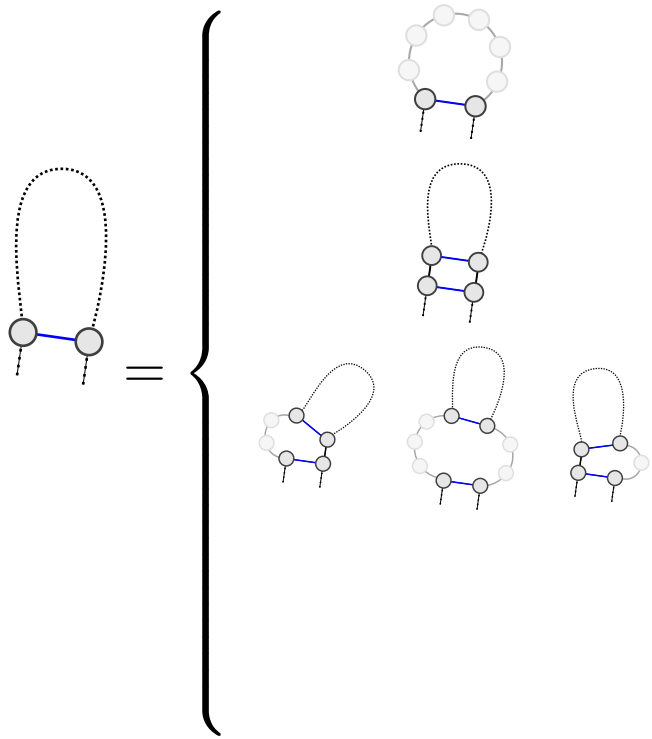


MFE folding

Theorem (MFE hypothesis)

RNA folds into its minimum free-energy conformation

But $\sim 1.8^n$ secondary structures compatible with S of size n [ZS84].
$\Rightarrow$ Dynamic programming, i.e. enumeration

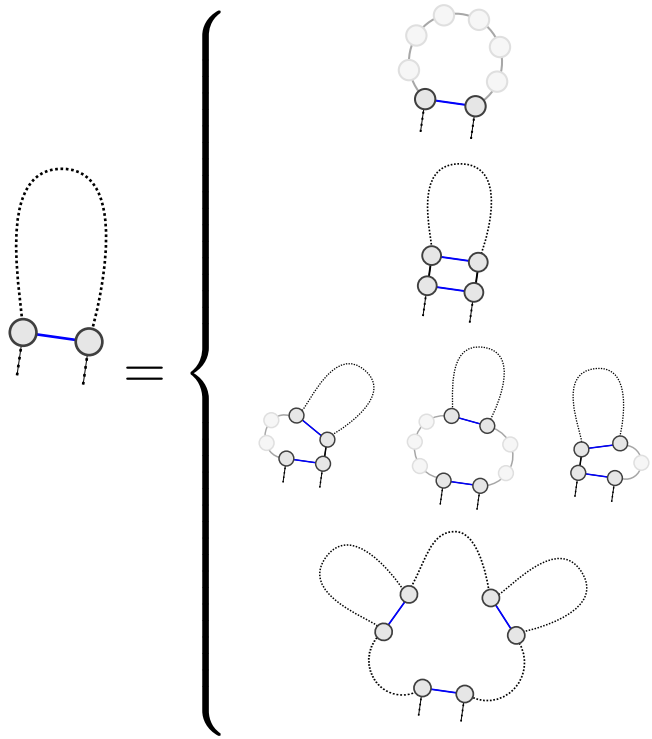


MFE folding

Theorem (MFE hypothesis)

RNA folds into its minimum free-energy conformation

But $\sim 1.8^n$ secondary structures compatible with S of size n [ZS84].
$\Rightarrow$ Dynamic programming, i.e. enumeration

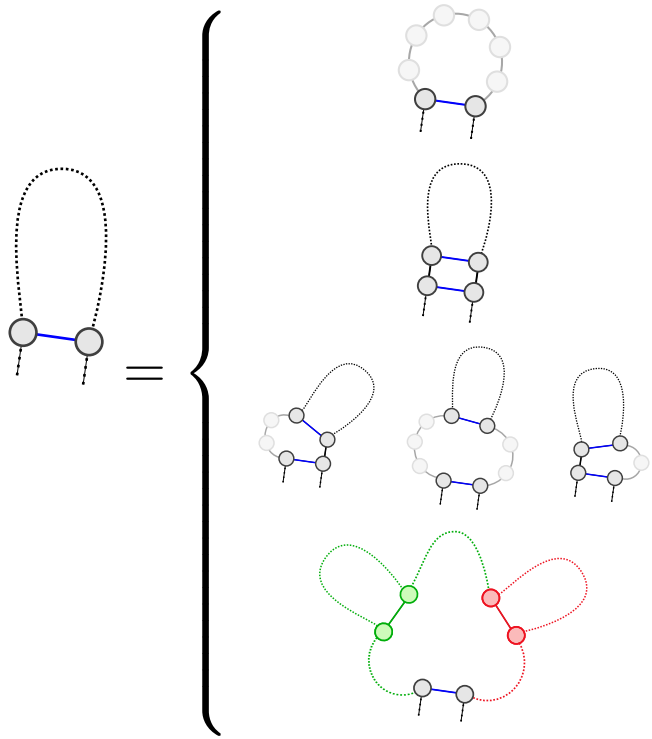


MFE folding

Theorem (MFE hypothesis)

RNA folds into its minimum free-energy conformation

But $\sim 1.8^n$ secondary structures compatible with S of size n [ZS84].
$\Rightarrow$ Dynamic programming, i.e. enumeration

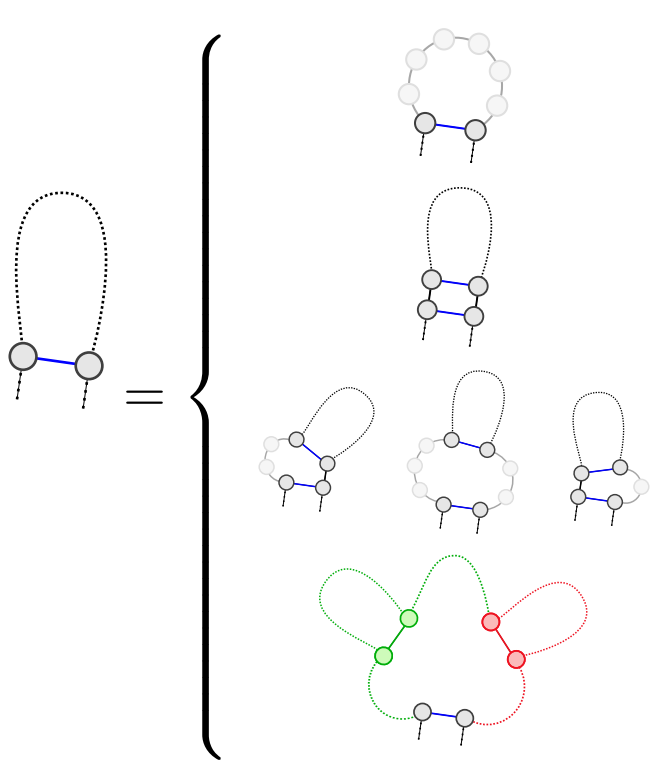


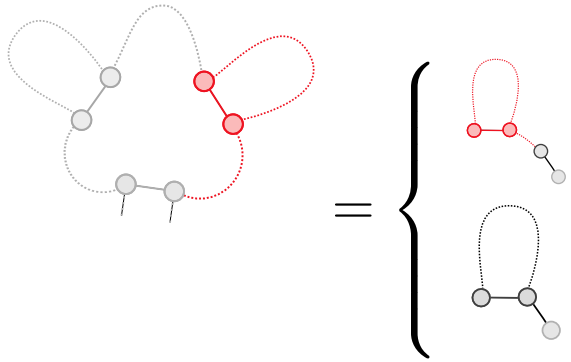
MFE folding

Theorem (MFE hypothesis)

RNA folds into its minimum free-energy conformation

But $\sim 1.8^n$ secondary structures compatible with S of size n [ZS84].
$\Rightarrow$ Dynamic programming, i.e. enumeration



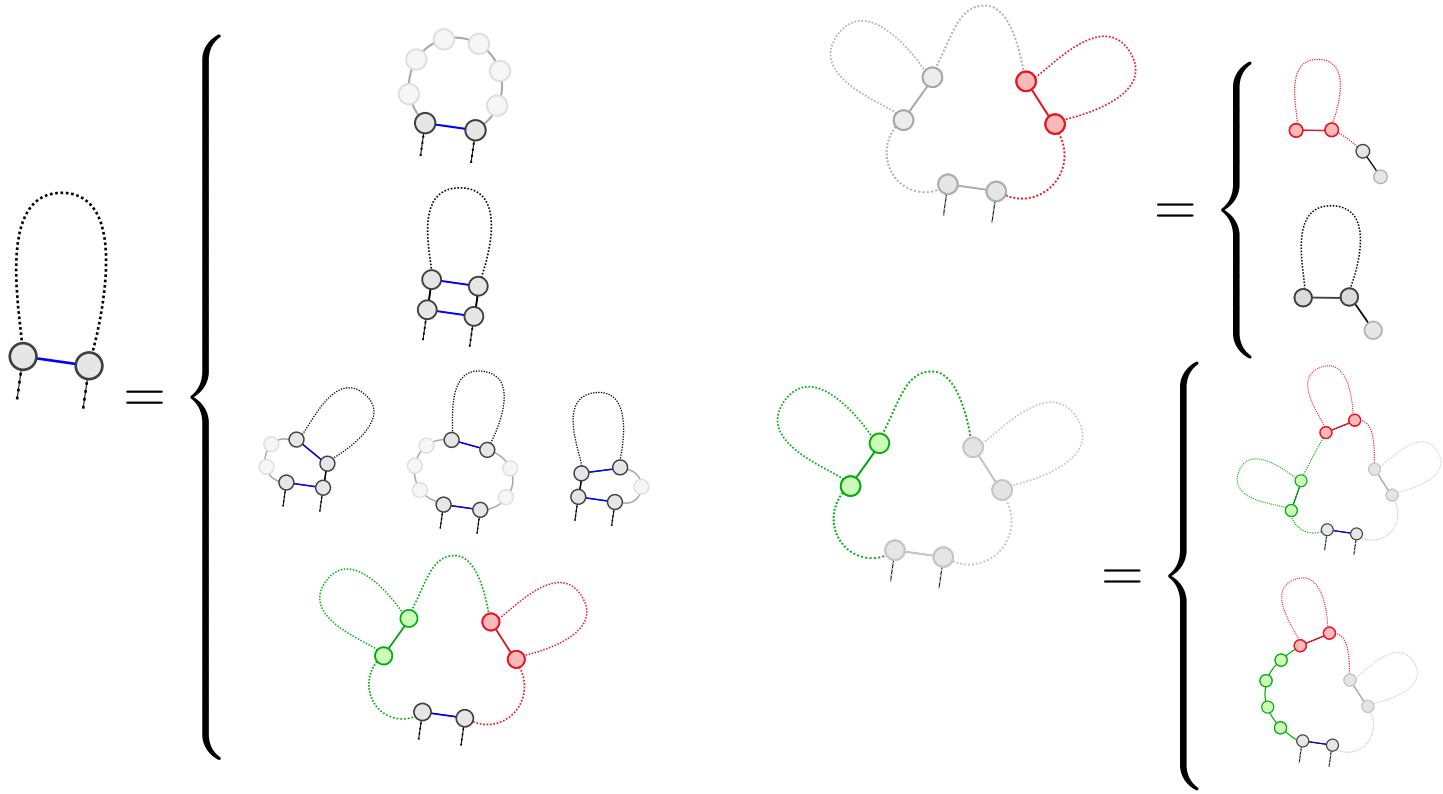


MFE folding

Theorem (MFE hypothesis)

RNA folds into its minimum free-energy conformation

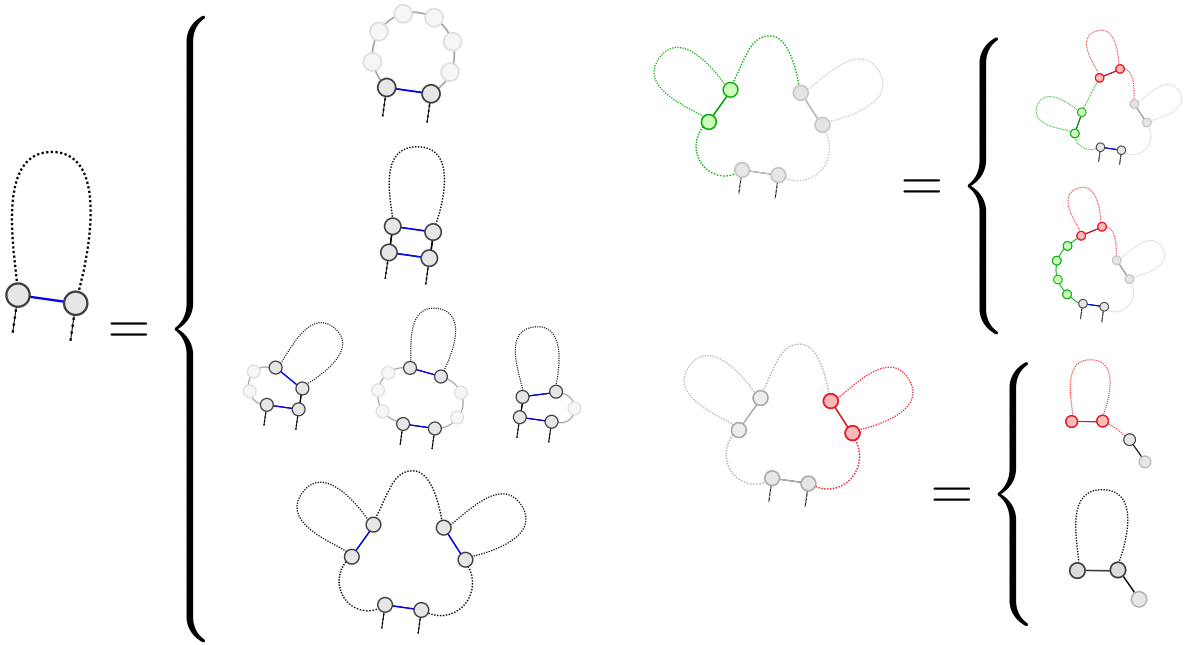
But $\sim 1.8^n$ secondary structures compatible with S of size n [ZS84].
$\Rightarrow$ Dynamic programming, i.e. enumeration



Validation

Proof of unambiguity $\Rightarrow$ Enumerative combinatorics
Waterman counted Sec. Str. [Wat78] and found the gen. fun. to be

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

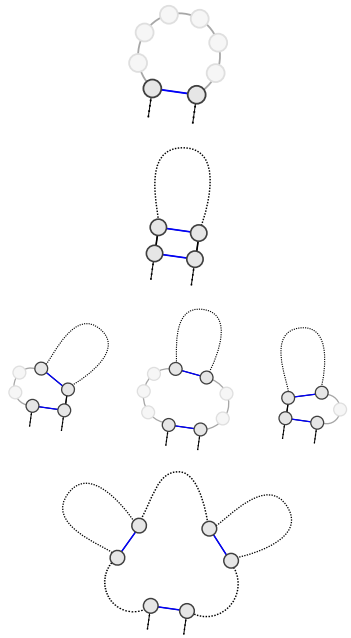


Validation

Proof of unambiguity $\Rightarrow$ Enumerative combinatorics
Waterman counted Sec. Str. [Wat78] and found the gen. fun. to be

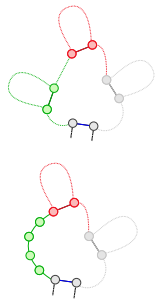
$$\mathcal{W}(z) = \frac{1 - z + z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4}}{2z^2}$$

$$\mathcal{A}(z) = \left\{ \right.$$



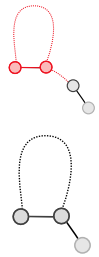
$$\left. \right\}$$

$$\mathcal{B}(z) = \left\{ \right.$$



$$\left. \right\}$$

$$\mathcal{C}(z) = \left\{ \right.$$



$$\left. \right\}$$

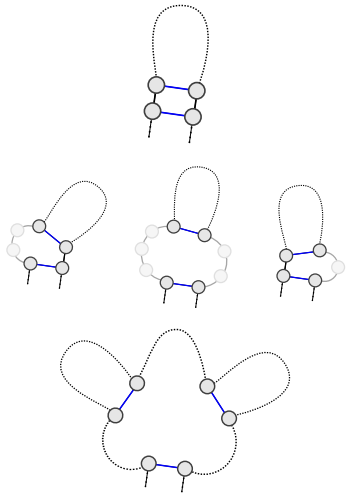
$$\mathcal{S}(z) = 1 + z\mathcal{S}(z)$$

Validation

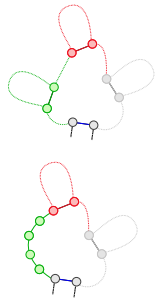
Proof of unambiguity $\Rightarrow$ Enumerative combinatorics
Waterman counted Sec. Str. [Wat78] and found the gen. fun. to be

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

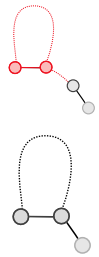
$$\mathcal{A}(z) = \left\{ \mathcal{S}(z) \right.$$



$$\mathcal{B}(z) = \left\{ \right.$$



$$\mathcal{C}(z) = \left\{ \right.$$



$$\mathcal{S}(z) = 1 + z\mathcal{S}(z)$$

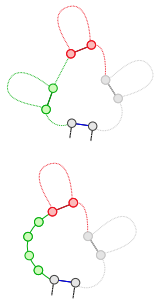
Validation

Proof of unambiguity $\Rightarrow$ Enumerative combinatorics
Waterman counted Sec. Str. [Wat78] and found the gen. fun. to be

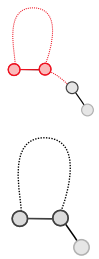
$$\mathcal{W}(z) = \frac{1 - z + z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4}}{2z^2}$$

$$\mathcal{A}(z) = \begin{cases} \mathcal{S}(z) \\ z^2\mathcal{A}(z) \\ z\mathcal{S}(z)z^2\mathcal{A}(z) + z^2\mathcal{A}(z)\mathcal{S}(z)z \\ +z\mathcal{S}(z)z^2\mathcal{A}(z)\mathcal{S}(z)z \\ \mathcal{B}(z)\mathcal{C}(z) \end{cases}$$

$$\mathcal{B}(z) = \begin{cases} \\ \\ \end{cases}$$



$$\mathcal{C}(z) = \begin{cases} \\ \\ \end{cases}$$



$$\mathcal{S}(z) = 1 + z\mathcal{S}(z)$$

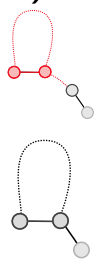
Validation

Proof of unambiguity $\Rightarrow$ Enumerative combinatorics
Waterman counted Sec. Str. [Wat78] and found the gen. fun. to be

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

$$\mathcal{A}(z) = \begin{cases} \mathcal{S}(z) \\ z^2\mathcal{A}(z) \\ z\mathcal{S}(z)z^2\mathcal{A}(z) + z^2\mathcal{A}(z)\mathcal{S}(z)z \\ +z\mathcal{S}(z)z^2\mathcal{A}(z)\mathcal{S}(z)z \\ \mathcal{B}(z)\mathcal{C}(z) \end{cases} \qquad \mathcal{B}(z) = \begin{cases} \mathcal{B}(z)\mathcal{C}(z) \\ \mathcal{S}(z)\mathcal{B}(z) \end{cases}$$

$$\mathcal{C}(z) = \begin{cases} \\ \\ \end{cases}$$



$$\mathcal{S}(z) = 1 + z\mathcal{S}(z)$$

Validation

Proof of unambiguity $\Rightarrow$ Enumerative combinatorics
Waterman counted Sec. Str. [Wat78] and found the gen. fun. to be

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

$$\mathcal{A}(z) = \begin{cases} \mathcal{S}(z) \\ z^2\mathcal{A}(z) \\ z\mathcal{S}(z)z^2\mathcal{A}(z) + z^2\mathcal{A}(z)\mathcal{S}(z)z \\ +z\mathcal{S}(z)z^2\mathcal{A}(z)\mathcal{S}(z)z \\ \mathcal{B}(z)\mathcal{C}(z) \end{cases} \qquad \mathcal{B}(z) = \begin{cases} \mathcal{B}(z)\mathcal{C}(z) \\ \mathcal{S}(z)\mathcal{B}(z) \end{cases} \qquad \mathcal{C}(z) = \begin{cases} \mathcal{C}(z)z \\ z^2\mathcal{A}(z) \end{cases}$$

$$\mathcal{S}(z) = 1 + z\mathcal{S}(z)$$

Validation

Proof of unambiguity $\Rightarrow$ Enumerative combinatorics
Waterman counted Sec. Str. [Wat78] and found the gen. fun. to be

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

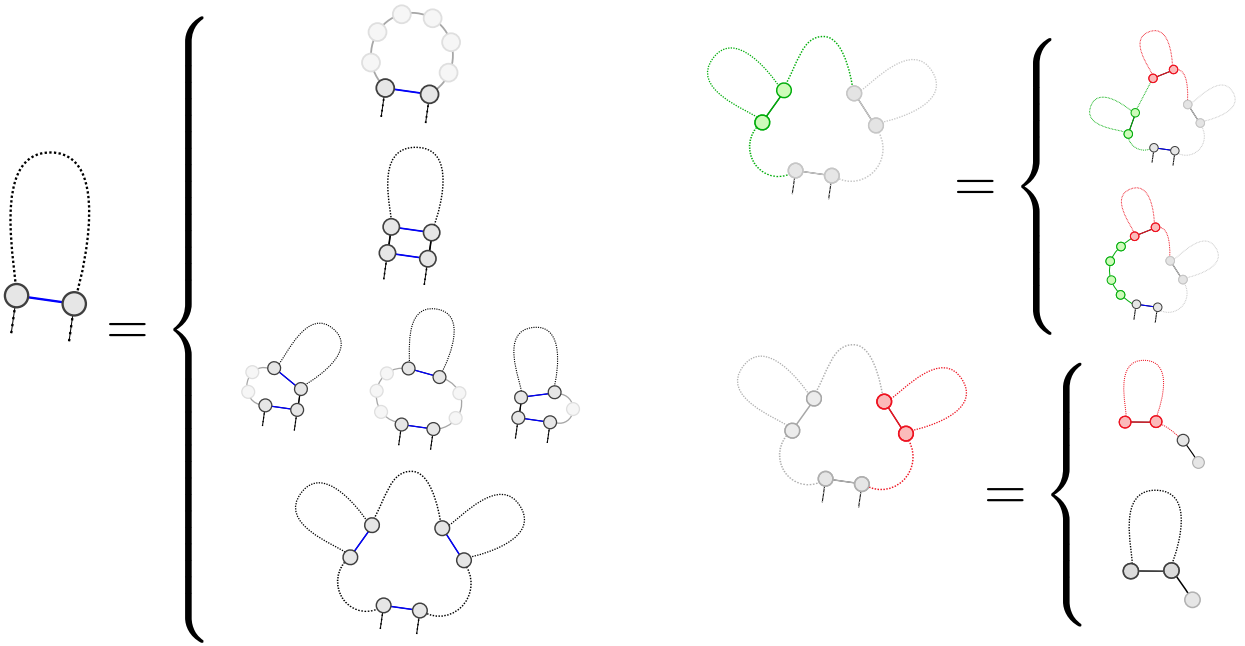
$$\mathcal{A}(z) = \begin{cases} \mathcal{S}(z) \\ z^2\mathcal{A}(z) \\ z\mathcal{S}(z)z^2\mathcal{A}(z) + z^2\mathcal{A}(z)\mathcal{S}(z)z \\ \quad + z\mathcal{S}(z)z^2\mathcal{A}(z)\mathcal{S}(z)z \\ \mathcal{B}(z)\mathcal{C}(z) \end{cases} \qquad \mathcal{B}(z) = \begin{cases} \mathcal{B}(z)\mathcal{C}(z) \\ \mathcal{S}(z)\mathcal{B}(z) \end{cases} \qquad \mathcal{C}(z) = \begin{cases} \mathcal{C}(z)z \\ z^2\mathcal{A}(z) \end{cases}$$

$$\begin{aligned} \Rightarrow \mathcal{A}(z) &= \frac{1 - z - z^2 - \sqrt{1 - 2z - z^2 - 2z^3 + z^4}}{2z^2} \\ &= \mathcal{W}(z) - 1 \end{aligned}$$

Woops, we forgot the empty structure for size 0 RNAs!

MFE folding

- $E_H(i,j)$: Energy of hairpin loop with closing pair (i,j)
- $E_{BI}(i,j)$: Energy of bulge or internal loop with closing pair (i,j)
- $E_S(i,j)$: Energy of stacking pairs $(i,j)/(i+1,j-1)$
- a,c,b: Penalties for multiloop, hairpins and unpaired bases in multiloop.



MFE folding

- $E_H(i,j)$: Energy of hairpin loop with closing pair (i,j)
- $E_{BI}(i,j)$: Energy of bulge or internal loop with closing pair (i,j)
- $E_S(i,j)$: Energy of stacking pairs $(i,j)/(i+1,j-1)$
- a,c,b: Penalties for multiloop, hairpins and unpaired bases in multiloop.

$$
\begin{aligned}
\mathcal{M}'(i,j) &= \text{Min}\left\{\begin{array}{c} E_H(i,j) \\ E_S(i,j)+\mathcal{M}'(i+1,j-1) \\ \text{Min}(E_{BI}(i,i',j',j)+\mathcal{M}'(i',j')) \\ a+c+\text{Min}(\mathcal{M}'(i+1,k-1)+\mathcal{M}^1(k,j-1)) \end{array}\right\} \\
\mathcal{M}(i,j) &= \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1),b(k-1)\right)+\mathcal{M}^1(k,j)\right\} \\
\mathcal{M}^1(i,j) &= \text{Min}\left\{b+\mathcal{M}^1(i,j-1),c+\mathcal{M}'(i,j)\right\}
\end{aligned}
$$

Backtracking

While reconstructing the m.f.e. conformation:

$$\mathcal{M}'(i,j) = \text{Min}\left\{\begin{array}{c} E_H(i,j) \\ E_S(i,j)+\mathcal{M}'(i+1,j-1) \\ \text{Min}(E_{BI}(i,i',j',j)+\mathcal{M}'(i',j')) \\ a+c+\text{Min}(\ \mathcal{M}'(i+1,k-1)+\mathcal{M}^1(k,j-1)\ ) \end{array}\right\}$$

$$\mathcal{M}(i,j) = \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1), b(k-1)\right)+\mathcal{M}^1(k,j)\right\}$$

$$\mathcal{M}^1(i,j) = \text{Min}\left\{b+\mathcal{M}^1(i,j-1), c+\mathcal{M}'(i,j)\right\}$$

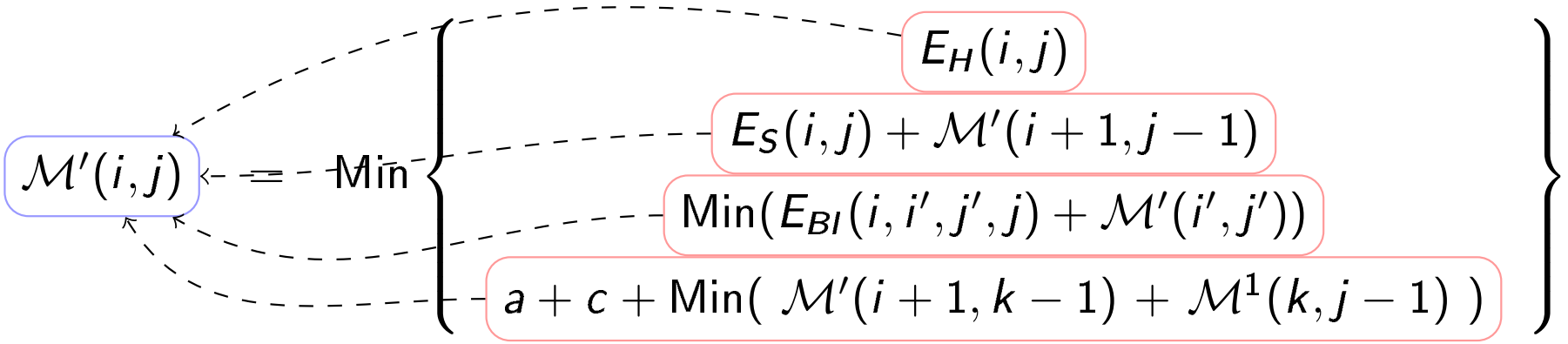
$\mathcal{O}(n)$ potential contributors to the Min:
$\Rightarrow$ $\mathcal{O}(n^2)$ worst-case complexity for naive traceback
Keep track of best contribution to Min $\Rightarrow$ $\mathcal{O}(n)$ worst-case traceback

$\mathcal{O}(n^3)$ time complexity for filling matrices[2]
$\Rightarrow$ `UnaFold` [MZ08] finds the minimal free-energy (MFE) structure.

[2] Slightly altered contribution for internal/bulges...

Backtracking

While reconstructing the m.f.e. conformation:



$$\mathcal{M}(i,j) = \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1), b(k-1)\right) + \mathcal{M}^1(k,j)\right\}$$

$$\mathcal{M}^1(i,j) = \text{Min}\left\{b + \mathcal{M}^1(i,j-1), c + \mathcal{M}'(i,j)\right\}$$

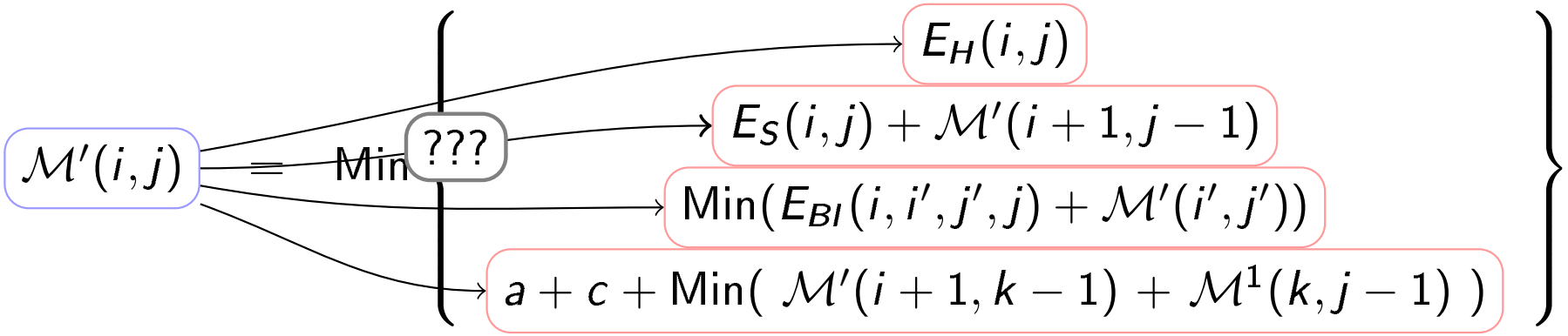
$\mathcal{O}(n)$ potential contributors to the Min:
$\Rightarrow$ $\mathcal{O}(n^2)$ worst-case complexity for naive traceback
Keep track of best contribution to Min $\Rightarrow$ $\mathcal{O}(n)$ worst-case traceback

$\mathcal{O}(n^3)$ time complexity for filling matrices[2]
$\Rightarrow$ `UnaFold` [MZ08] finds the minimal free-energy (MFE) structure.

[2] Slightly altered contribution for internal/bulges...

Backtracking

While reconstructing the m.f.e. conformation:



$$\mathcal{M}(i,j) = \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1), b(k-1)\right) + \mathcal{M}^1(k,j)\right\}$$
$$\mathcal{M}^1(i,j) = \text{Min}\left\{b + \mathcal{M}^1(i,j-1), c + \mathcal{M}'(i,j)\right\}$$

$\mathcal{O}(n)$ potential contributors to the Min:
$\Rightarrow$ $\mathcal{O}(n^2)$ worst-case complexity for naive traceback
Keep track of best contribution to Min $\Rightarrow$ $\mathcal{O}(n)$ worst-case traceback

$\mathcal{O}(n^3)$ time complexity for filling matrices[2]
$\Rightarrow$ `UnaFold` [MZ08] finds the minimal free-energy (MFE) structure.

[2] Slightly altered contribution for internal/bulges...

Backtracking

While reconstructing the m.f.e. conformation:

$$\mathcal{M}'(i,j) = \text{Min}\left\{\begin{array}{r} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'(i+1,j-1) \\ \text{Min}(E_{BI}(i,i',j',j) + \mathcal{M}'(i',j')) \\ a + c + \text{Min}(\ \mathcal{M}'(i+1,k-1) + \mathcal{M}^1(k,j-1)\ ) \end{array}\right\}$$

$$\mathcal{M}(i,j) = \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1), b(k-1)\right) + \mathcal{M}^1(k,j)\right\}$$

$$\mathcal{M}^1(i,j) = \text{Min}\left\{b + \mathcal{M}^1(i,j-1), c + \mathcal{M}'(i,j)\right\}$$

$\mathcal{O}(n)$ potential contributors to the Min:
$\Rightarrow$ $\mathcal{O}(n^2)$ worst-case complexity for naive traceback
Keep track of best contribution to Min $\Rightarrow$ $\mathcal{O}(n)$ worst-case traceback

$\mathcal{O}(n^3)$ time complexity for filling matrices[2]
$\Rightarrow$ UnaFold [MZ08] finds the minimal free-energy (MFE) structure.

[2] Slightly altered contribution for internal/bulges...

Backtracking

While reconstructing the m.f.e. conformation:

$$\mathcal{M}'(i,j) = \text{Min}\left\{\begin{array}{r} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'(i+1,j-1) \\ \text{Min}(E_{BI}(i,i',j',j) + \mathcal{M}'(i',j')) \\ a + c + \text{Min}(\ \mathcal{M}'(i+1,k-1) + \mathcal{M}^1(k,j-1)\ ) \end{array}\right\}$$

$$\mathcal{M}(i,j) = \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1), b(k-1)\right) + \mathcal{M}^1(k,j)\right\}$$

$$\mathcal{M}^1(i,j) = \text{Min}\left\{b + \mathcal{M}^1(i,j-1), c + \mathcal{M}'(i,j)\right\}$$

$\mathcal{O}(n)$ potential contributors to the Min:

$\Rightarrow$ $\mathcal{O}(n^2)$ worst-case complexity for naive traceback

Keep track of best contribution to Min $\Rightarrow$ $\mathcal{O}(n)$ worst-case traceback

$\mathcal{O}(n^3)$ time complexity for filling matrices[2]

$\Rightarrow$ `UnaFold` [MZ08] finds the minimal free-energy (MFE) structure.

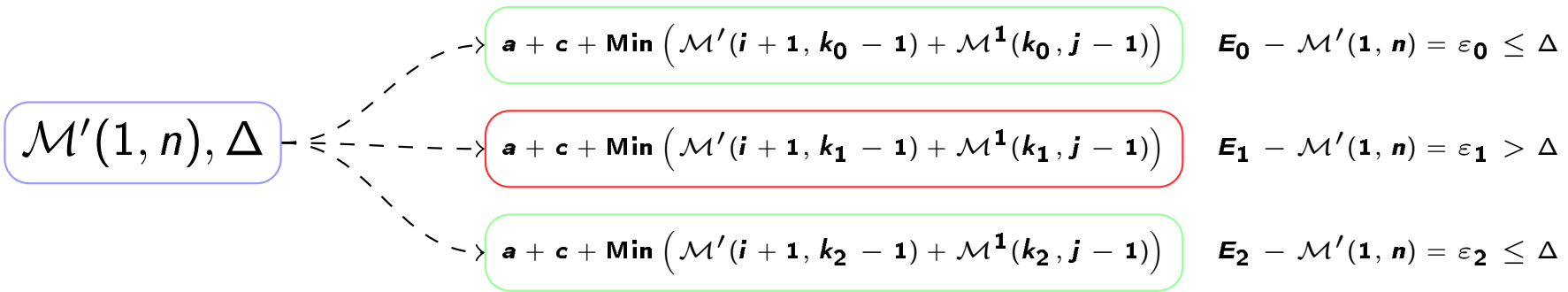
[2] Slightly altered contribution for internal/bulges...

Sampling suboptimal foldings

Prob.: Approximation of energy function (Pseudoknots, NC base-pairs), so the real (native) structure could be underestimated and ignored.

$\Rightarrow$ Generate suboptimal foldings (`RNASubopt` [WFHS99]), i.e. build all structures within Δ KCal.mol^{-1} of m.f.e.:

- Backtrack over contributors within Δ of m.f.e.
- Update bound Δ' s.t. further backtracks gives at least one structure
- Combine subsets while pruned (Sort or brute force)



Sampling suboptimal foldings

Prob.: Approximation of energy function (Pseudoknots, NC base-pairs), so the real (native) structure could be underestimated and ignored.

$\Rightarrow$ Generate suboptimal foldings (`RNASubopt` [WFHS99]), i.e. build all structures within Δ KCal.mol^{-1} of m.f.e.:

- Backtrack over contributors within Δ of m.f.e.
- Update bound Δ' s.t. further backtracks gives at least one structure
- Combine subsets while pruned (Sort or brute force)

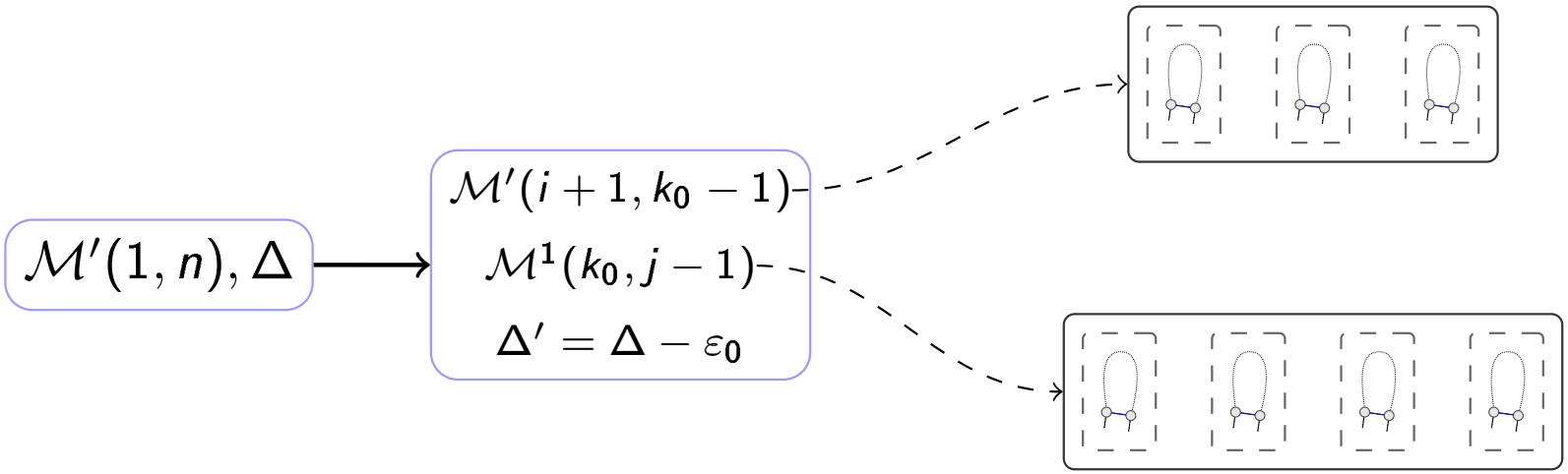
$$\mathcal{M}'(1,n),\Delta \longrightarrow \begin{array}{c} \mathcal{M}'(i+1,k_0-1) \\ \mathcal{M}^1(k_0,j-1) \\ \Delta' = \Delta - \varepsilon_0 \end{array}$$

Sampling suboptimal foldings

Prob.: Approximation of energy function (Pseudoknots, NC base-pairs), so the real (native) structure could be underestimated and ignored.

$\Rightarrow$ Generate suboptimal foldings (`RNASubopt` [WFHS99]),
i.e. build all structures within Δ KCal.mol^{-1} of m.f.e.:

- Backtrack over contributors within Δ of m.f.e.
- Update bound Δ' s.t. further backtracks gives at least one structure
- Combine subsets while pruned (Sort or brute force)

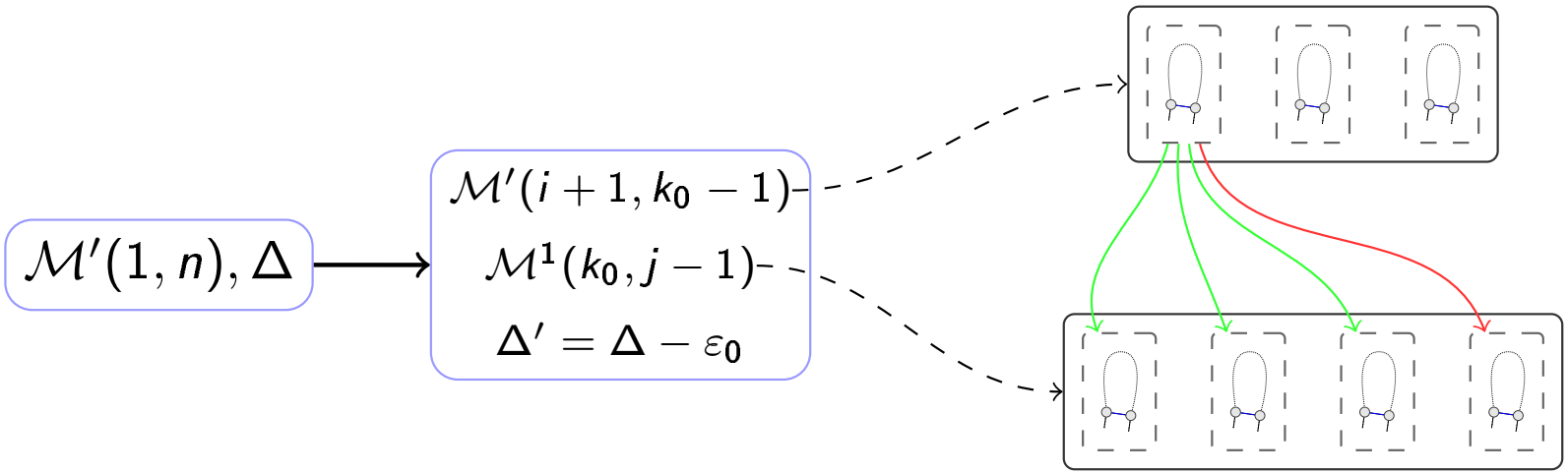


Sampling suboptimal foldings

Prob.: Approximation of energy function (Pseudoknots, NC base-pairs), so the real (native) structure could be underestimated and ignored.

$\Rightarrow$ Generate suboptimal foldings (`RNASubopt` [WFHS99]),
i.e. build all structures within Δ KCal.mol^{-1} of m.f.e.:

- Backtrack over contributors within Δ of m.f.e.
- Update bound Δ' s.t. further backtracks gives at least one structure
- Combine subsets while pruned (Sort or brute force)

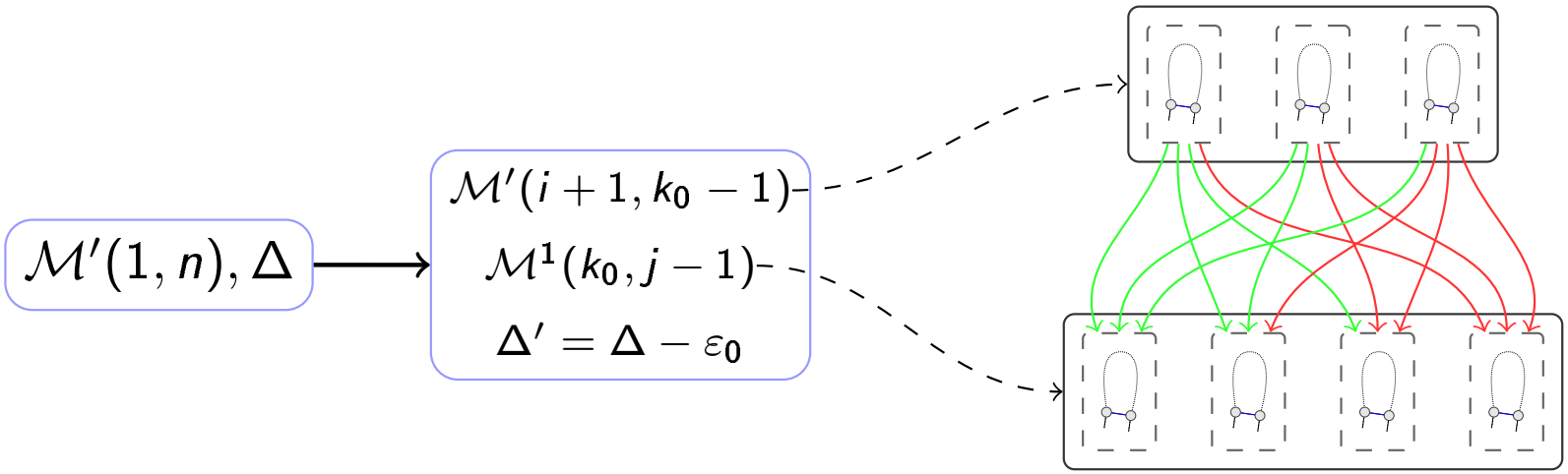


Sampling suboptimal foldings

Prob.: Approximation of energy function (Pseudoknots, NC base-pairs), so the real (native) structure could be underestimated and ignored.

$\Rightarrow$ Generate suboptimal foldings (`RNASubopt` [WFHS99]),
i.e. build all structures within Δ KCal.mol^{-1} of m.f.e.:

- Backtrack over contributors within Δ of m.f.e.
- Update bound Δ' s.t. further backtracks gives at least one structure
- Combine subsets while pruned (Sort or brute force)



Sampling suboptimal foldings

Prob.: Approximation of energy function (Pseudoknots, NC base-pairs), so the real (native) structure could be underestimated and ignored.

$\Rightarrow$ Generate suboptimal foldings (`RNASubopt` [WFHS99]),
i.e. build all structures within Δ KCal.mol^{-1} of m.f.e.:

- Backtrack over contributors within Δ of m.f.e.
- Update bound Δ' s.t. further backtracks gives at least one structure
- Combine subsets while pruned (Sort or brute force)

Subopts might need to be sorted afterward
$\Rightarrow \mathcal{O}(n^3 + nk\log(k))$ time complexity
(k grows exponentially on Δ, but hey!)

Open question A: Iterative generation of subopts

Assume we've already filled matrices, can we tweak the backtrack s.t. all subopts are generated in increasing order?

Sampling suboptimal foldings

Prob.: Approximation of energy function (Pseudoknots, NC base-pairs), so the real (native) structure could be underestimated and ignored.

$\Rightarrow$ Generate suboptimal foldings (`RNASubopt` [WFHS99]),
i.e. build all structures within Δ KCal.mol^{-1} of m.f.e.:

- Backtrack over contributors within Δ of m.f.e.
- Update bound Δ' s.t. further backtracks gives at least one structure
- Combine subsets while pruned (Sort or brute force)

Subopts might need to be sorted afterward
$\Rightarrow \mathcal{O}(n^3 + nk\log(k))$ time complexity
(k grows exponentially on Δ, but hey!)

Open question A: Iterative generation of subopts

Assume we've already filled matrices, can we tweak the backtrack s.t. all subopts are generated in increasing order?

Partition function: Beyond m.f.e. hypothesis

Partition function/Boltzmann probability

- Let ω be an RNA sequence,
- $\mathcal{S}_\omega$ be the set of sequences compatible with ω,

$$\textbf{Partition function} \quad \mathcal{Z}_\omega = \sum_{S \in \mathcal{S}_\omega} e^{\frac{-E_{S,\omega}}{RT}}$$

where T is temperature in Kelvin and R is the universal gas constant.

$$\textbf{Boltzmann probability} \quad P_{S,\omega} = \frac{e^{\frac{-E_{S,\omega}}{RT}}}{\mathcal{Z}_\omega}$$

$P_{S,\omega}$ is the probability of observing ω in conformation S.

$\Rightarrow$ Offers a more dynamic view of the folding process

$\Rightarrow$ Provides a model for computing various probabilities (BP, Motifs ...)

$\Rightarrow$ Very easy to embed into existing DP equations

Partition function: Beyond m.f.e. hypothesis

Partition function/Boltzmann probability

- Let ω be an RNA sequence,
- $\mathcal{S}_\omega$ be the set of sequences compatible with ω,

$$\textbf{Partition function} \quad \mathcal{Z}_\omega = \sum_{S \in \mathcal{S}_\omega} e^{\frac{-E_{S,\omega}}{RT}}$$

where T is temperature in Kelvin and R is the universal gas constant.

$$\textbf{Boltzmann probability} \quad P_{S,\omega} = \frac{e^{\frac{-E_{S,\omega}}{RT}}}{\mathcal{Z}_\omega}$$

$P_{S,\omega}$ is the probability of observing ω in conformation S.

$\Rightarrow$ Offers a more dynamic view of the folding process

$\Rightarrow$ Provides a model for computing various probabilities (BP, Motifs ...)

$\Rightarrow$ Very easy to embed into existing DP equations

Partition function: Beyond m.f.e. hypothesis

Partition function/Boltzmann probability

- Let ω be an RNA sequence,
- $\mathcal{S}_\omega$ be the set of sequences compatible with ω,

$$\textbf{Partition function} \quad \mathcal{Z}_\omega = \sum_{S \in \mathcal{S}_\omega} e^{\frac{-E_{S,\omega}}{RT}}$$

where T is temperature in Kelvin and R is the universal gas constant.

$$\textbf{Boltzmann probability} \quad P_{S,\omega} = \frac{e^{\frac{-E_{S,\omega}}{RT}}}{\mathcal{Z}_\omega}$$

$P_{S,\omega}$ is the probability of observing ω in conformation S.

$\Rightarrow$ Offers a more dynamic view of the folding process

$\Rightarrow$ Provides a model for computing various probabilities (BP, Motifs ...)

$\Rightarrow$ Very easy to embed into existing DP equations

Partition function: Beyond m.f.e. hypothesis

Partition function/Boltzmann probability

- Let ω be an RNA sequence,
- $\mathcal{S}_\omega$ be the set of sequences compatible with ω,

$$\textbf{Partition function} \quad \mathcal{Z}_\omega = \sum_{S \in \mathcal{S}_\omega} e^{\frac{-E_{S,\omega}}{RT}}$$

where T is temperature in Kelvin and R is the universal gas constant.

$$\textbf{Boltzmann probability} \quad P_{S,\omega} = \frac{e^{\frac{-E_{S,\omega}}{RT}}}{\mathcal{Z}_\omega}$$

$P_{S,\omega}$ is the probability of observing ω in conformation S.

$\Rightarrow$ Offers a more dynamic view of the folding process

$\Rightarrow$ Provides a model for computing various probabilities (BP, Motifs ...)

$\Rightarrow$ Very easy to embed into existing DP equations

Partition function: Beyond m.f.e. hypothesis

From m.f.e. folding to partition function [McC90]:

- Atomic energy increment $E \Rightarrow$ Boltzmann factor $e^{\frac{-E}{RT}}$
- Energies contr. move to the exponent:
  Sums $(+) \Rightarrow$ Products $(\times)$
- Summing instead of minimizing: Min => Sums $(\sum)$

$$
\begin{aligned}
\mathcal{M}'(i,j) &= \text{Min}\left\{\begin{array}{c} E_H(i,j) \\ E_S(i,j)+\mathcal{M}'(i+1,j-1) \\ \text{Min}(E_{BI}(i,i',j',j)+\mathcal{M}'(i',j')) \\ a+c+\text{Min}\left(\mathcal{M}'(i+1,k-1)+\mathcal{M}^1(k,j-1)\right) \end{array}\right\} \\
\mathcal{M}(i,j) &= \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1),b(k-1)\right)+\mathcal{M}^1(k,j)\right\} \\
\mathcal{M}^1(i,j) &= \text{Min}\left\{b+\mathcal{M}^1(i,j-1),c+\mathcal{M}'(i,j)\right\}
\end{aligned}
$$

Partition function: Beyond m.f.e. hypothesis

From m.f.e. folding to partition function [McC90]:

- Atomic energy increment $E \Rightarrow$ Boltzmann factor $e^{\frac{-E}{RT}}$
- Energies contr. move to the exponent:
  Sums $(+) \Rightarrow$ Products $(\times)$
- Summing instead of minimizing: Min => Sums $(\sum)$

$$\mathcal{M}'(i,j) = \text{Min}\left\{\begin{array}{c} e^{\frac{-E_H(i,j)}{RT}} \\ e^{\frac{-E_S(i,j)}{RT}} + \mathcal{M}'(i+1,j-1) \\ \text{Min}\left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} + \mathcal{M}'(i',j')\right) \\ e^{\frac{-(a+c)}{RT}} + \text{Min}\left(\mathcal{M}'(i+1,k-1) + \mathcal{M}^1(k,j-1)\right) \end{array}\right\}$$

$$\mathcal{M}(i,j) = \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1), e^{\frac{-b(k-1)}{RT}}\right) + \mathcal{M}^1(k,j)\right\}$$

$$\mathcal{M}^1(i,j) = \text{Min}\left\{e^{\frac{-b}{RT}} + \mathcal{M}^1(i,j-1), e^{\frac{-c}{RT}} + \mathcal{M}'(i,j)\right\}$$

Partition function: Beyond m.f.e. hypothesis

From m.f.e. folding to partition function [McC90]:

- Atomic energy increment $E \Rightarrow$ Boltzmann factor $e^{\frac{-E}{RT}}$
- Energies contr. move to the exponent:
  Sums $(+) \Rightarrow$ Products $(\times)$
- Summing instead of minimizing: Min => Sums $(\sum)$

$$\mathcal{M}'(i,j) = \text{Min}\left\{\begin{array}{c} e^{\frac{-E_H(i,j)}{RT}} \\ e^{\frac{-E_S(i,j)}{RT}}\mathcal{M}'(i+1,j-1) \\ \text{Min}\left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}}\mathcal{M}'(i',j')\right) \\ e^{\frac{-(a+c)}{RT}}\text{Min}\left(\mathcal{M}'(i+1,k-1)\mathcal{M}^1(k,j-1)\right) \end{array}\right\}$$

$$\mathcal{M}(i,j) = \text{Min}\left\{\text{Min}\left(\mathcal{M}(i,k-1), e^{\frac{-b(k-1)}{RT}}\right)\mathcal{M}^1(k,j)\right\}$$

$$\mathcal{M}^1(i,j) = \text{Min}\left\{e^{\frac{-b}{RT}}\mathcal{M}^1(i,j-1), e^{\frac{-c}{RT}}\mathcal{M}'(i,j)\right\}$$

Partition function: Beyond m.f.e. hypothesis

From m.f.e. folding to partition function [McC90]:

- Atomic energy increment $E \Rightarrow$ Boltzmann factor $e^{\frac{-E}{RT}}$
- Energies contr. move to the exponent:
  Sums $(+) \Rightarrow$ Products $(\times)$
- Summing instead of minimizing: Min => Sums $(\sum)$

$$
\begin{aligned}
\mathcal{Z}'(i,j) &= \sum \left\{ \begin{array}{c} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) \\ + \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) \\ + e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) \end{array} \right\} \\
\mathcal{Z}(i,j) &= \sum \left(\mathcal{Z}(i,k-1) + e^{\frac{-b(k-1)}{RT}} \right) \mathcal{Z}^1(k,j) \\
\mathcal{Z}^1(i,j) &= e^{\frac{-b}{RT}} \mathcal{Z}^1(i,j-1) + e^{\frac{-c}{RT}} \mathcal{Z}'(i,j)
\end{aligned}
$$

Partition function: Beyond m.f.e. hypothesis

From m.f.e. folding to partition function [McC90]:

- Atomic energy increment $E \Rightarrow$ Boltzmann factor $e^{\frac{-E}{RT}}$
- Energies contr. move to the exponent:
  Sums $(+) \Rightarrow$ Products $(\times)$
- Summing instead of minimizing: Min => Sums $(\sum)$

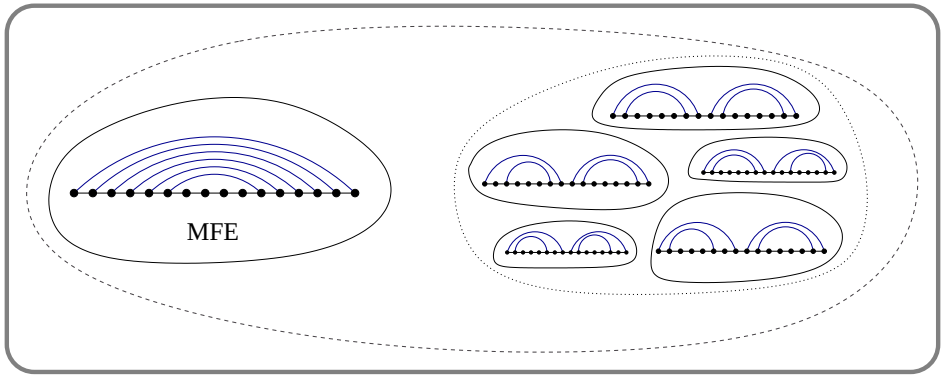
$$
\begin{aligned}
\mathcal{Z}'(i,j) &= \sum \left\{ \begin{array}{c} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) \\ + \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) \\ + e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) \end{array} \right\} \\
\mathcal{Z}(i,j) &= \sum \left(\mathcal{Z}(i,k-1) + e^{\frac{-b(k-1)}{RT}} \right) \mathcal{Z}^1(k,j) \\
\mathcal{Z}^1(i,j) &= e^{\frac{-b}{RT}} \mathcal{Z}^1(i,j-1) + e^{\frac{-c}{RT}} \mathcal{Z}'(i,j)
\end{aligned}
$$

Now, we can restrict the sums to compute unpaired/paired base probabilities, base-pair prob., hairpin loops prob. ...

Statistical sampling of RNA

Apology for statistical sampling

The m.f.e. (Highest Boltzmann probability) $\mathcal{M}$ can be isolated and less probable than a set $\mathcal{B}$ of structurally similar suboptimals.
In this setting, native structure closer to $\mathcal{B}$ than to $\mathcal{M}$ [DCL05].



Strategy:

- Sample structures with Boltzmann probability
- Cluster structures
- Build and return a consensus structure from the best cluster

$\Rightarrow$ Relative improvements for specificity (+17.6%) and sensitivity (+21.74%, except for group II Introns)

Statistical sampling through stochastic traceback

Algorithm `SFold` [DL03]:

1. Generate a random number in $[0, \mathcal{Z}'(i,j))$
2. Subtract to r individual contributions to $\mathcal{Z}'(i,j)$, until $r < 0$
3. Recurse over substructures

$$\mathcal{Z}'(i,j) = \sum \left\{ \begin{array}{cc} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) & \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) & \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) & \text{C} \end{array} \right\}$$

Statistical sampling through stochastic traceback

Algorithm `SFold` [DL03]:

1. Generate a random number in $[0, \mathcal{Z}'(i,j))$
2. Subtract to r individual contributions to $\mathcal{Z}'(i,j)$, until $r < 0$
3. Recurse over substructures

$$\mathcal{Z}'(i,j) = \sum \text{???} \begin{cases} \text{A} & e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) \\ \text{B} & \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) \\ \text{C} & e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) \end{cases}$$

Statistical sampling through stochastic traceback

Algorithm `SFold` [DL03]:

1. Generate a random number in $[0, \mathcal{Z}'(i,j))$
2. Subtract to r individual contributions to $\mathcal{Z}'(i,j)$, until $r < 0$
3. Recurse over substructures

$$\mathcal{Z}'(i,j) = \sum \left\{ \begin{array}{cc} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) & \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) & \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) & \text{C} \end{array} \right\}$$

r

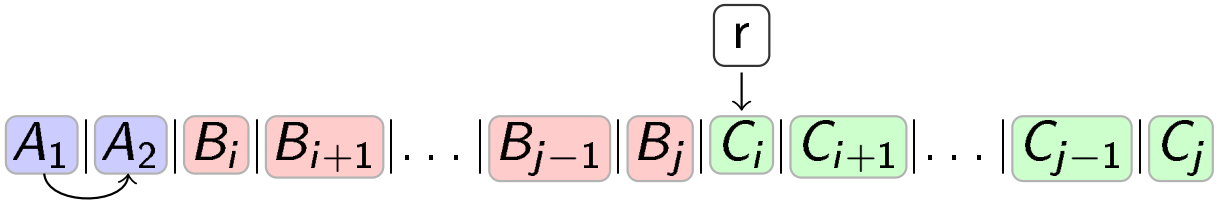
$A_1 | A_2 | B_i | B_{i+1} | \ldots | B_{j-1} | B_j | C_i | C_{i+1} | \ldots | C_{j-1} | C_j$

Statistical sampling through stochastic traceback

Algorithm `SFold` [DL03]:

1. Generate a random number in $[0, \mathcal{Z}'(i,j))$
2. Subtract to r individual contributions to $\mathcal{Z}'(i,j)$, until $r < 0$
3. Recurse over substructures

$$\mathcal{Z}'(i,j) = \sum \left\{ \begin{array}{cc} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) & \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) & \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) & \text{C} \end{array} \right\}$$

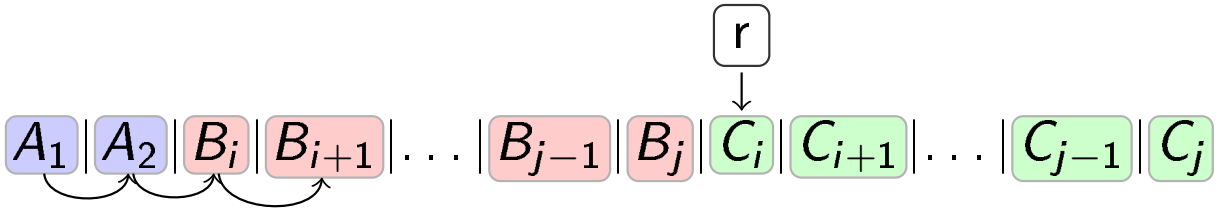


Statistical sampling through stochastic traceback

Algorithm `SFold` [DL03]:

1. Generate a random number in $[0, \mathcal{Z}'(i,j))$
2. Subtract to r individual contributions to $\mathcal{Z}'(i,j)$, until $r < 0$
3. Recurse over substructures

$$\mathcal{Z}'(i,j) = \sum \left\{ \begin{array}{cc} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) & \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) & \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) & \text{C} \end{array} \right\}$$

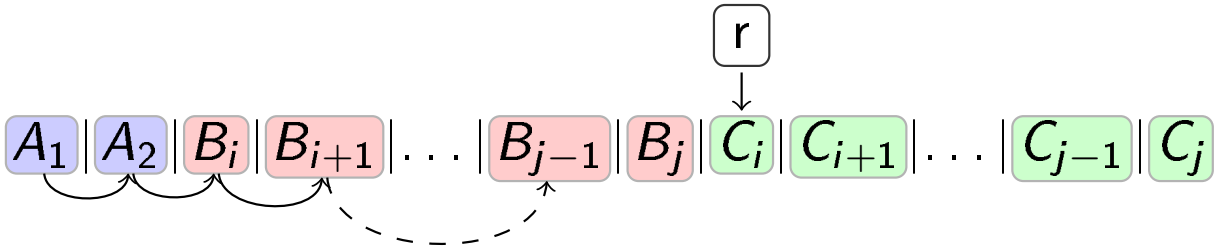


Statistical sampling through stochastic traceback

Algorithm `SFold` [DL03]:

1. Generate a random number in $[0, \mathcal{Z}'(i,j))$
2. Subtract to r individual contributions to $\mathcal{Z}'(i,j)$, until $r < 0$
3. Recurse over substructures

$$\mathcal{Z}'(i,j) = \sum \left\{ \begin{array}{cc} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) & \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) & \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) & \text{C} \end{array} \right\}$$

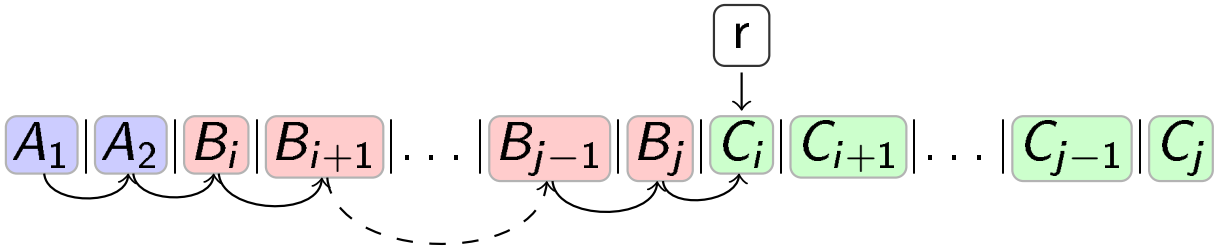


Statistical sampling through stochastic traceback

Algorithm `SFold` [DL03]:

1. Generate a random number in $[0, \mathcal{Z}'(i,j))$
2. Subtract to r individual contributions to $\mathcal{Z}'(i,j)$, until $r < 0$
3. Recurse over substructures

$$\mathcal{Z}'(i,j) = \sum \left\{ \begin{array}{cc} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) & \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) & \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) & \text{C} \end{array} \right\}$$

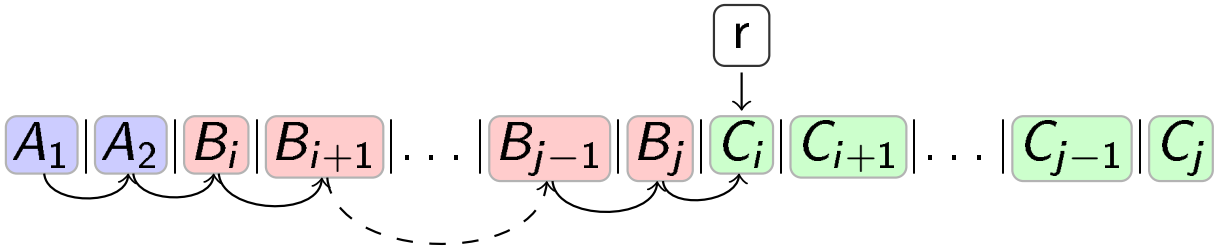


Statistical sampling through stochastic traceback

Algorithm `SFold` [DL03]:

1. Generate a random number in $[0, \mathcal{Z}'(i,j))$
2. Subtract to r individual contributions to $\mathcal{Z}'(i,j)$, until $r < 0$
3. Recurse over substructures

$$\mathcal{Z}'(i,j) = \sum \left\{ \begin{array}{cc} e^{\frac{-E_H(i,j)}{RT}} + e^{\frac{-E_S(i,j)}{RT}} \mathcal{Z}'(i+1,j-1) & \text{A} \\ \sum \left(e^{\frac{-E_{BI}(i,i',j',j)}{RT}} \mathcal{Z}'(i',j') \right) & \text{B} \\ e^{\frac{-(a+c)}{RT}} \sum \left(\mathcal{Z}'(i+1,k-1) \mathcal{Z}^1(k,j-1) \right) & \text{C} \end{array} \right\}$$



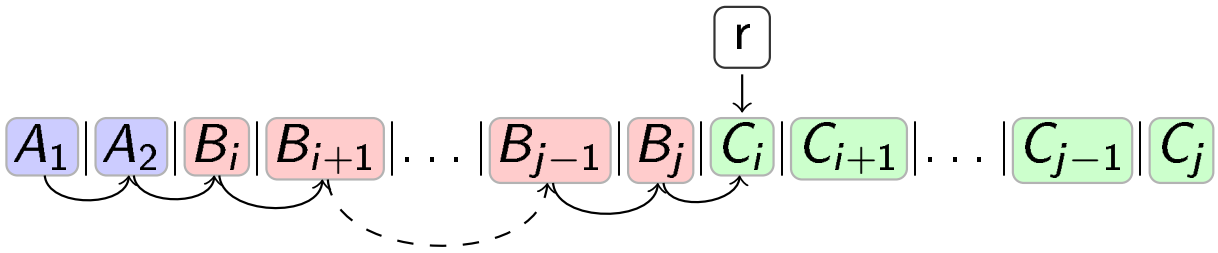
After $\Theta(n)$ operations, recurse over size $n-1$ interval
$\Rightarrow$ Worst-case time complexity for k samples in $\mathcal{O}(n^2 k)$

Remark: This is a weighted instance of the so-called recursive random generation of decomposable objects.

Efficient statistical sampling

How to improve statistical sampling?

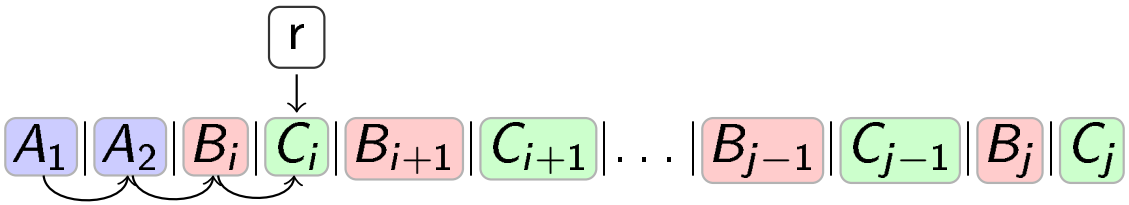
- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation



Efficient statistical sampling

How to improve statistical sampling?

- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation

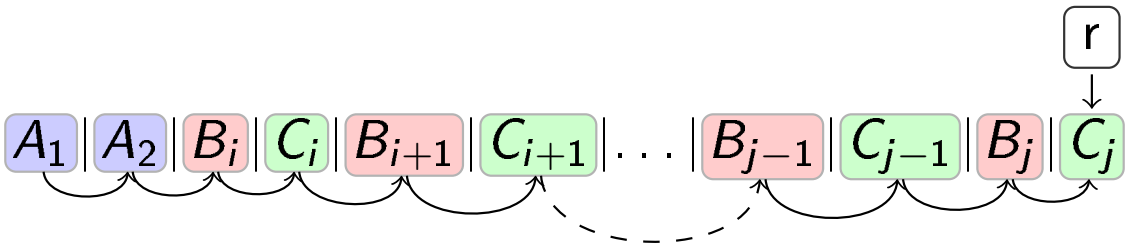


$\Rightarrow$ Some terms from B and C are reached in $\mathcal{O}(1)$

Efficient statistical sampling

How to improve statistical sampling?

- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation

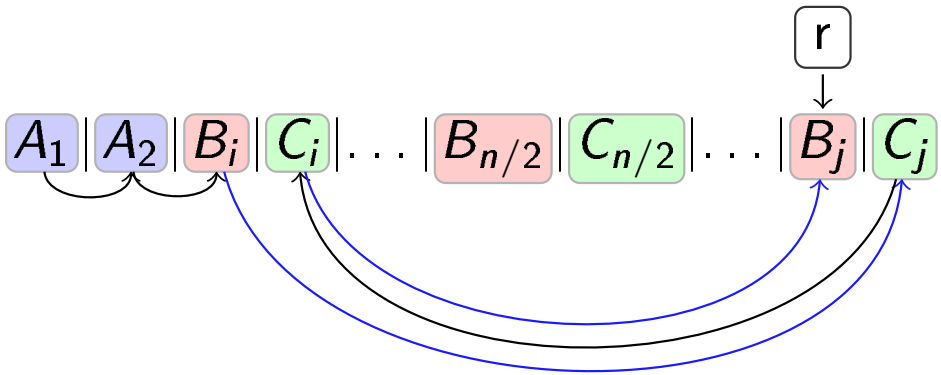


$\Rightarrow$ Some terms from B and C are reached in $\mathcal{O}(1)$
But still $\Theta(n^2)$, since $\mathcal{Z}'(i,j) \rightarrow (\mathcal{Z}'(i+1,k-1), \mathcal{Z}^1(k,j-1))$

Efficient statistical sampling

How to improve statistical sampling?

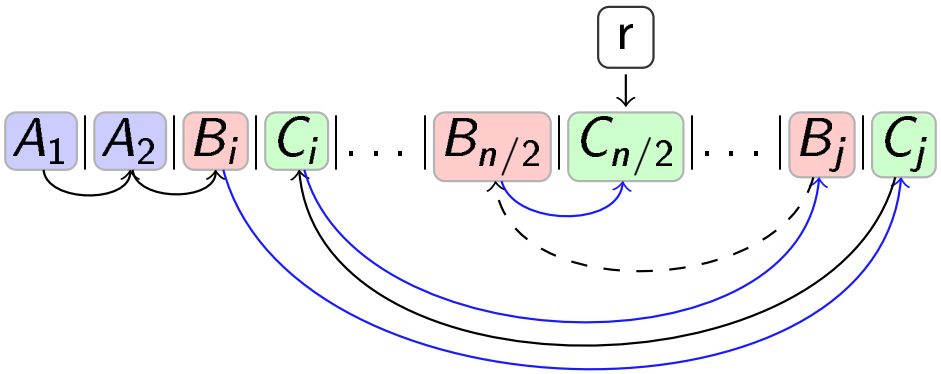
- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation



Efficient statistical sampling

How to improve statistical sampling?

- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94] $\Rightarrow \Theta(n \log(n))$ worst-case
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation

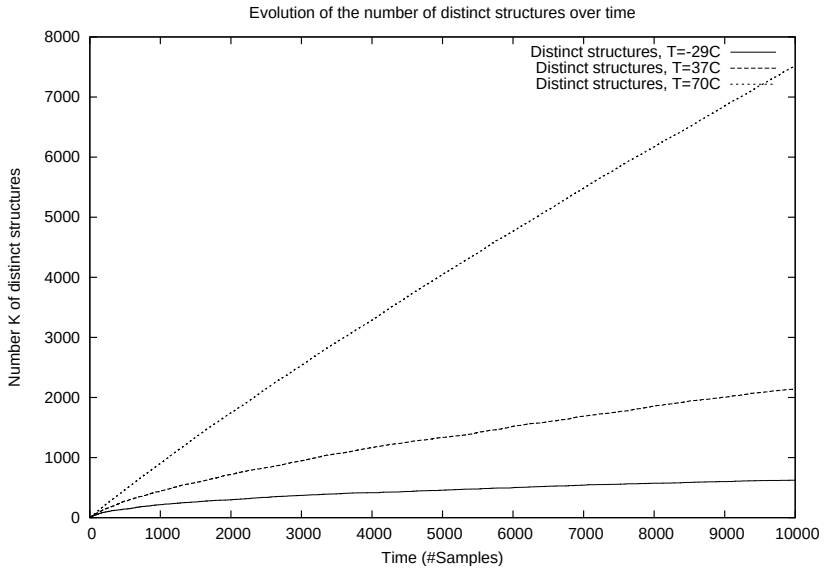


Worst-case: Divide exactly at each step [GK81] $\Rightarrow \Theta(n \log(n))$

Efficient statistical sampling

How to improve statistical sampling?

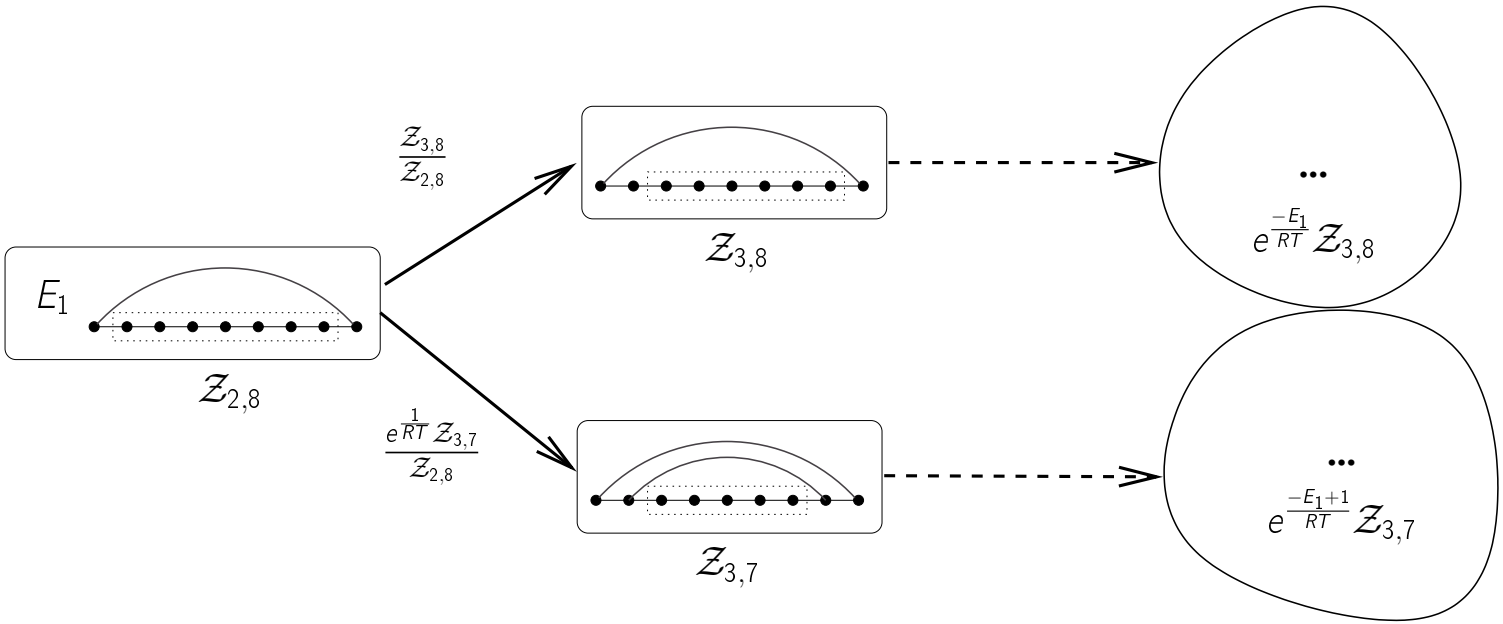
- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation



Efficient statistical sampling

How to improve statistical sampling?

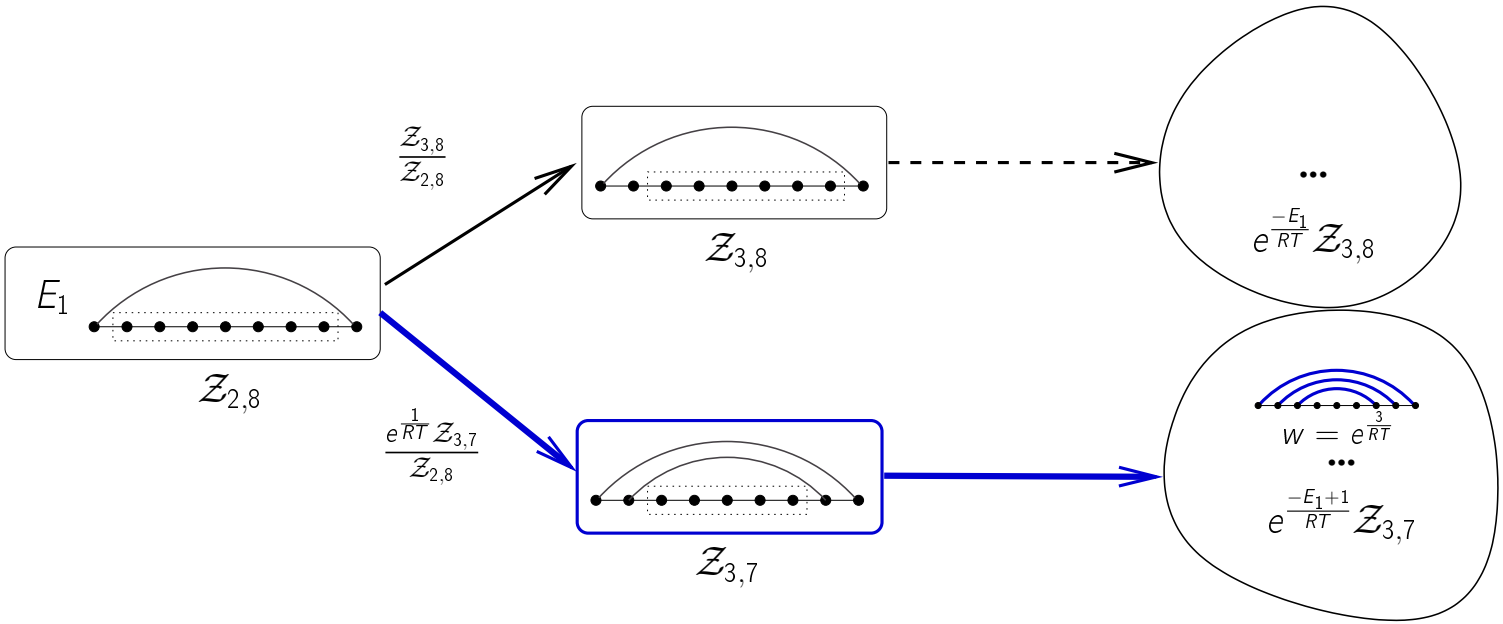
- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation



Efficient statistical sampling

How to improve statistical sampling?

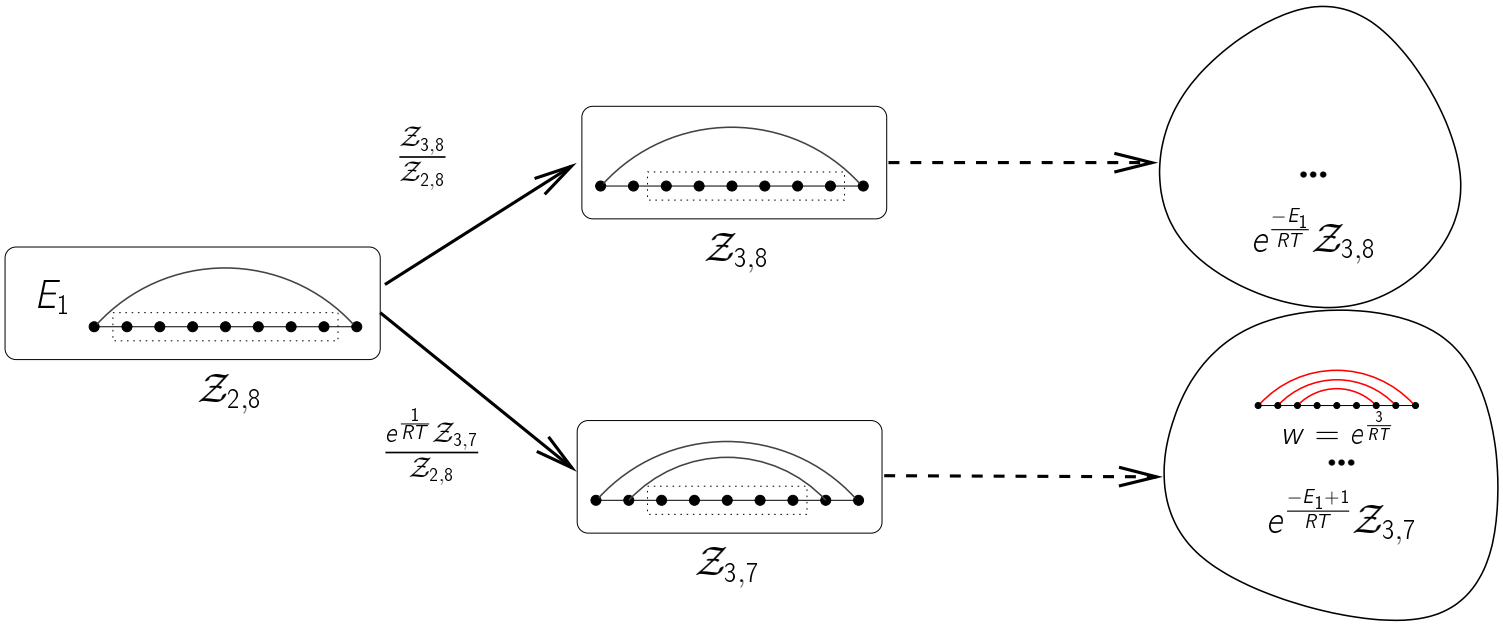
- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation



Efficient statistical sampling

How to improve statistical sampling?

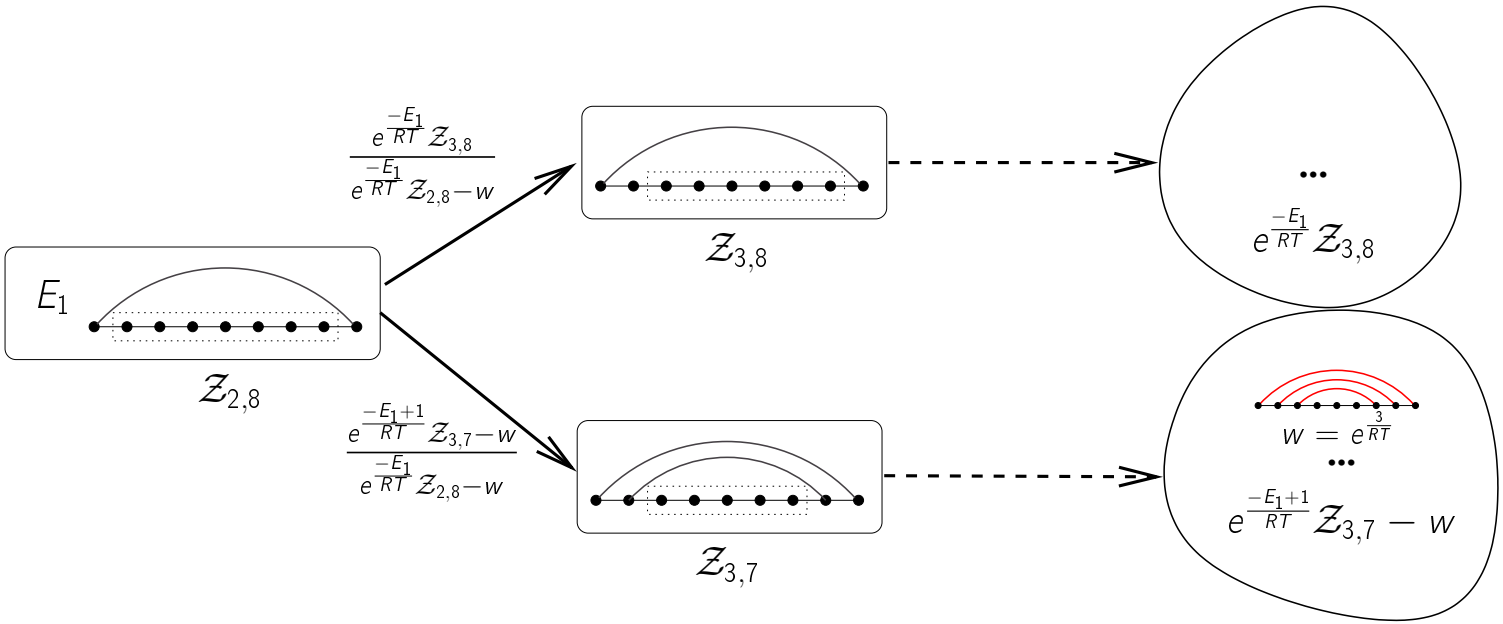
- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation



Efficient statistical sampling

How to improve statistical sampling?

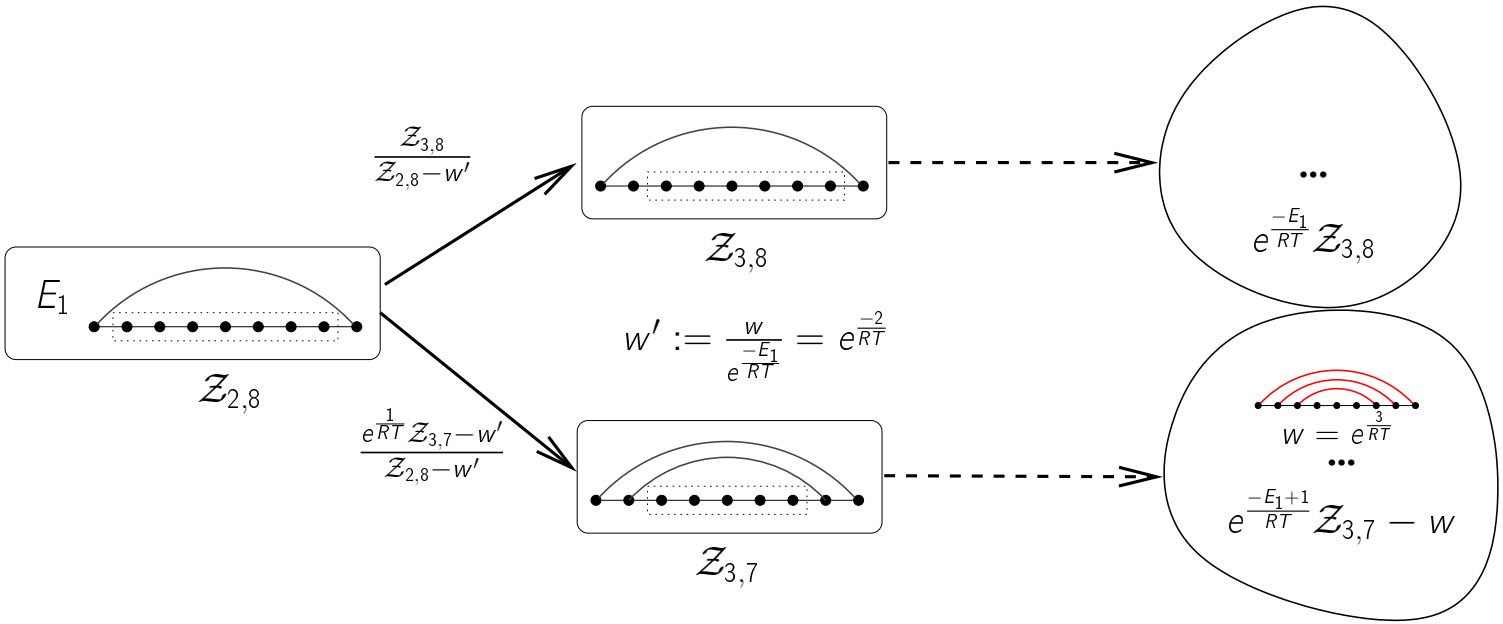
- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation



Efficient statistical sampling

How to improve statistical sampling?

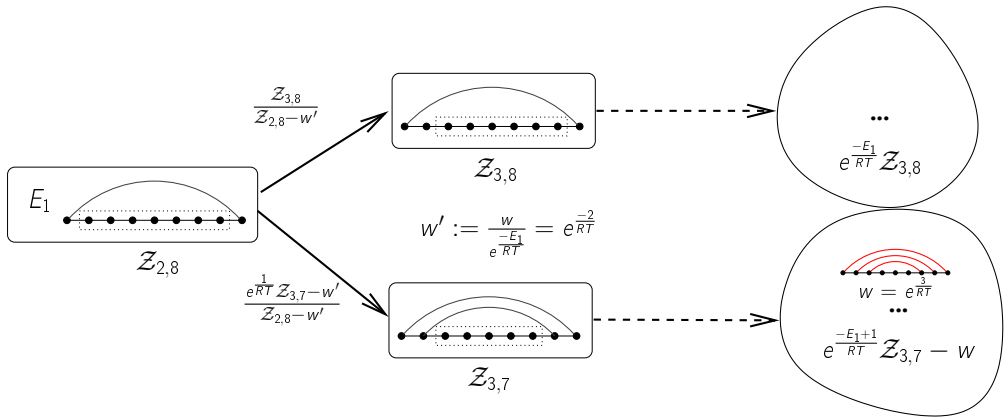
- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation



Efficient statistical sampling

How to improve statistical sampling?

- Improve time complexity:
  Average-case time complexity in $\Theta(kn\sqrt{n})$ [Pon08]
  ($\Theta(n^2)$ arises from recursing on $n - \mathcal{O}(1)$ after $\Theta(n)$ ops)
  - Interleaving Bulges (B) and Multiloops (C) contributions
  - Boustrophedon [FZV94]
    Investigate uneven decompositions first, then even ones !
- Non-redundant generation
  - Build prefix tree for parse traces, storing in each node the contributions $K = \sum_{S \in \mathcal{R}} e^{\frac{-E_S}{RT}}$ of already sampled structures $\mathcal{R}$
  - During traceback, modify contributions of terms using K [Pon08]



(Partial?) Conclusion

In structural biology, the following conditions:

- Additivity of energy function
- Sweetly enumerable conformational space (No $-$ or min)

allowed for an exhaustive (polynomial) exploration through:

- Generate suboptimal foldings (`RNASubopt`)
- Compute partition function (`McCaskill`)
- Partition conformation landscape (`RNAMutants`, `RNABor`, `RNAShapes`)
- Perform statistical sampling in the Boltzmann ensemble (`SFold`)
- Simulate simple hybridization (`hybrid`)

Additional motivations for enriching the conformational space:

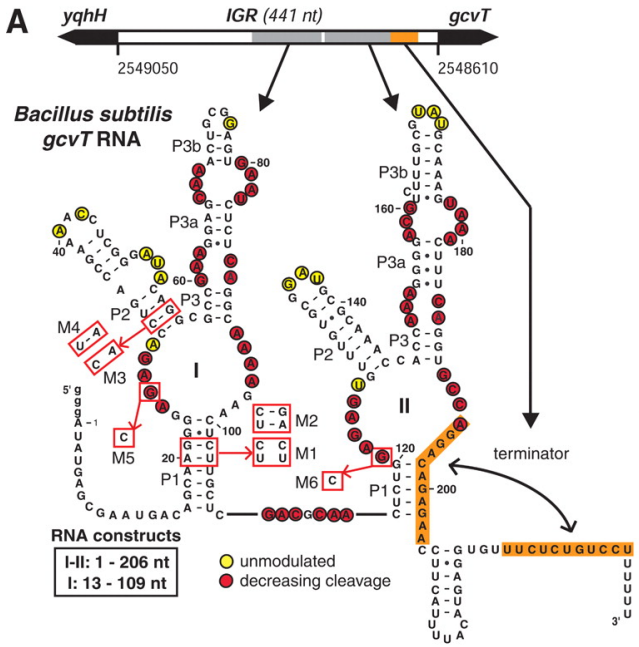
$\Rightarrow$ Better predictions (PK, non-canonical)

$\Rightarrow$ Predict interactions

Toward RNA design

Use our understanding of folding mechanisms to design

- Small interfering RNA engineering
- RNA switches, bistable RNAs
- Self-assembly and nanostructures



Science-Mandal *et al*-2004

Toward RNA design

RNA inverse folding problem

Given a target structure S, and a predictive algorithm P find an RNA sequence ω such $P(\omega) = S$.

Existing approaches:

- Local search approaches [AFH+04, AHHC07, BB06]
- No or few constraints
- Connectivity of the sequence space?

Open question B: Complexity of RNA design

What is the theoretical complexity of RNA-design?

Seems there is a hole in inverse-optimization theory...

Open question B': Stable RNA design

Given a target structure S, find an RNA sequence ω that both satisfies the RNA design problem, and has energy $E_{S,\omega} > E^{(2)}_{S,\omega} + \Delta$.

Toward RNA design

RNA inverse folding problem

Given a target structure S, and a predictive algorithm P find an RNA sequence ω such $P(\omega) = S$.

Existing approaches:

- Local search approaches [AFH+04, AHHC07, BB06]
- No or few constraints
- Connectivity of the sequence space?

Open question B: Complexity of RNA design

What is the theoretical complexity of RNA-design?

Seems there is a hole in inverse-optimization theory...

Open question B': Stable RNA design

Given a target structure S, find an RNA sequence ω that both satisfies the RNA design problem, and has energy $E_{S,\omega} > E^{(2)}_{S,\omega} + \Delta$.

Toward RNA design

RNA inverse folding problem

Given a target structure S, and a predictive algorithm P find an RNA sequence ω such $P(\omega) = S$.

Existing approaches:

- Local search approaches [AFH+04, AHHC07, BB06]
- No or few constraints
- Connectivity of the sequence space?

Open question B: Complexity of RNA design

What is the theoretical complexity of RNA-design?

Seems there is a hole in inverse-optimization theory...

Open question B': Stable RNA design

Given a target structure S, find an RNA sequence ω that both satisfies the RNA design problem, and has energy $E_{S,\omega} > E^{(2)}_{S,\omega} + \Delta$.

References I

M. Andronescu, A. P. Fejes, F. Hutter, H. H. Hoos, and A. Condon.

A New Algorithm for RNA Secondary Structure Design.
Journal of Molecular Biology, 336(3):607–624, 2004.

R. Aguirre-Hernandez, H. H. Hoos, and A. Condon.

Computational RNA secondary structure design: empirical complexity and improved methods.
BMC Bioinformatics, 8(1):34, 2007.

A. Busch and R. Backofen.

INFO-RNA–a fast approach to inverse RNA folding.
Bioinformatics, 22(15):1823–1831, 2006.

A. Condon, B. Davy, B. Rastegari, S. Zhao, and F. Tarrant.

Classifying RNA pseudoknotted structures.
Theoretical Computer Science, 320(1):35–50, 2004.

K. Doshi, J. J. Cannone, C. Cobaugh, and R. R. Gutell.

Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for rna secondary structure prediction.
BMC Bioinformatics, 5(1):105, 2004.

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.

References II

P. Flajolet, P. Zimmermann, and B. Van Cutsem.

Calculus for the random generation of labelled combinatorial structures.
Theoretical Computer Science, 132:1–35, 1994.

P. Gardner and R. Giegerich.

A comprehensive comparison of comparative rna structure prediction approaches.
BMC Bioinformatics, 5(1):140, 2004.

D. H. Greene and D. E. Knuth.

Mathematics for the Analysis of Algorithms.
Birkhauser Boston, 1981.

R. B. Lyngsø and C. N. S. Pedersen.

RNA pseudoknot prediction in energy-based models.
Journal of Computational Biology, 7(3-4):409–427, 2000.

N. Leontis and E. Westhof.

Geometric nomenclature and classification of RNA base pairs.
RNA, 7:499–512, 2001.

J.S. McCaskill.

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

D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner.

Expanded sequence dependence of thermodynamic parameters improves prediction of rna secondary structure.
Journal of Molecular Biology, 288(5):911–940, May 1999.

N. R. Markham and M. Zuker.

Bioinformatics, chapter UNAFold, pages 3–31.
Springer, 2008.

References III

M. Parisien and F. Major.

The mc-fold and mc-sym pipeline infers rna structure from sequence data.
Nature, 452(7183):51–55, 2008.

Y. Ponty.

Efficient sampling of RNA secondary structures from the boltzmann ensemble of low-energy: The boustrophedon method.
Journal of Mathematical Biology, 56(1-2):107–127, Jan 2008.

B. A. Shapiro, Y. G. Yingling, W. Kasprzak, and E. Bindewald.

Bridging the gap in rna structure prediction.
Curr Opin Struct Biol, 17(2):157–165, Apr 2007.

M. S. Waterman.

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

S. Wuchty, W. Fontana, I.L. Hofacker, and P. Schuster.

Complete suboptimal folding of RNA and the stability of secondary structures.
Biopolymers, 49:145–164, 1999.

M. Zuker and D. Sankoff.

Rna secondary structures and their prediction.
Bull Math Bio, 46:591–621, 1984.

Proof of average-case complexity

Theorem

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

Proof.

Homodimer model: All pair of positions can form a base-pair.
Boltzmann distribution: Based on Nussinov model.

Then the generating function $C(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} c(\omega) z^{|S|}$ holding the (unnormalized) average cost of a sampling scenario can be expressed in term of the **partition function** generating function $P_f(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} z^{|S|}$

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

□

Proof of average-case complexity

Theorem

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

Proof.

Homodimer model: All pair of positions can form a base-pair.
Boltzmann distribution: Based on Nussinov model.

Then the generating function $C(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} c(\omega) z^{|S|}$ holding the (unnormalized) average cost of a sampling scenario can be expressed in term of the **partition function** generating function $P_f(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} z^{|S|}$

Moreover, $P_f(z)$ is solution of a system of algebraic equations induced by Waterman's context-free grammar for RNA secondary structures.

$$\begin{cases} P_f(z) &= z^2 e^{\frac{1}{RT}} P_f^{\geq\theta}(z) P_f(z) + z P_f(z) + 1 \\ P_f^{\geq\theta}(z) &= z^2 e^{\frac{1}{RT}} P_f^{\geq\theta}(z) P_f(z) + z P_f(z) + z^\theta. \end{cases}$$

□

Proof of average-case complexity

Theorem

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

Proof.

Homodimer model: All pair of positions can form a base-pair.
Boltzmann distribution: Based on Nussinov model.

Then the generating function $C(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} c(\omega) z^{|S|}$ holding the (unnormalized) average cost of a sampling scenario can be expressed in term of the **partition function** generating function $P_f(z) = \sum_{S \in \mathcal{S}} e^{\frac{bp(S)}{RT}} z^{|S|}$

Moreover, $P_f(z)$ is solution of a system of algebraic equations induced by Waterman's context-free grammar for RNA secondary structures.

Extracting $A_n := [z^n]C(z)$ and $B_n := [z^n]P_f(z)$ using singularity analysis yields an average-case complexity $A_n/B_n \in \Theta(n\sqrt{n})$. □