M2 BIM/STRUCT - Lecture 1

Folding RNA *in silico*

Yann Ponty

Bioinformatics Team
École Polytechnique/CNRS/INRIA AMIB – France

http://www.lix.polytechnique.fr/~ponty/index.php?page=bim2016

December 13rd, 2016

Outline

1. **Introduction**
   - Dynamic programming 101
   - Why RNA?
   - RNA folding
   - RNA Structure(s)
   - Some representations of RNA structure
2. **Some flavours of folding prediction**
   - Thermodynamics vs Kinetics
   - Dynamic programming: Reminder
3. **Free-energy minimization**
   - Nussinov-style RNA folding
   - Turner energy model
   - MFold/Unafold
   - Performances and the comparative approach
   - Towards a 3D ab-initio prediction

Foreword . . .

. . . or how to make a million bucks by giving change parsimoniously!!

Problem: You have access to unlimited amount of **1**, **20** and **50** cents coins. A client prefers to travel light, i.e. to **minimize the #coins**. How to give **N** cents back in change without losing a customer?

Strategy #1: Start with *heaviest* coins, and then complete/fill-up with coins of *decreasing* value.

$21 = ??$

55

60

Foreword . . .

. . . or how to make a million bucks by giving change parsimoniously!!

Problem: You have access to unlimited amount of **1**, **20** and **50** cents coins.
A client prefers to travel light, i.e. to **minimize the #coins**.
How to give **N** cents back in change without losing a customer?

Strategy #1: Start with *heaviest* coins, and then complete/fill-up with coins of *decreasing* value.

21 =  + 

55??

60

. . . or how to make a million bucks by giving change parsimoniously!!

Problem: You have access to unlimited amount of **1**, **20** and **50** cents coins.
A client prefers to travel light, i.e. to **minimize the #coins**.
How to give **N** cents back in change without losing a customer?

Strategy #1: Start with *heaviest* coins, and then complete/fill-up with coins of *decreasing* value.

21 =  + 

55 =  +  +  +  +  + 

60??

. . . or how to make a million bucks by giving change parsimoniously!!

Problem: You have access to unlimited amount of **1**, **20** and **50** cents coins.
A client prefers to travel light, i.e. to **minimize the #coins**.
How to give **N** cents back in change without losing a customer?

Strategy #1: Start with *heaviest* coins, and then complete/fill-up with coins of *decreasing* value.

$21 =$ 

$55 =$ 

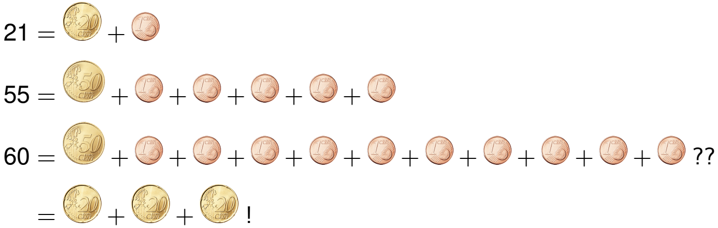
$60 =$  ??

Foreword . . .

. . . or how to make a million bucks by giving change parsimoniously!!

Problem: You have access to unlimited amount of **1**, **20** and **50** cents coins.
A client prefers to travel light, i.e. to **minimize the #coins**.
How to give **N** cents back in change without losing a customer?

Strategy #1: Start with *heaviest* coins, and then complete/fill-up with coins of *decreasing* value.



Problem *a priori (?!)* non-solvable using such a *greedy* approach, as a (simpler) problem is already NP-complete (thus Efficient solution $\Rightarrow$ 1M\$).

Strategy #2: Brute force enumeration $\rightarrow$ #Coins^N (Ouch!)

Strategy #3: The following recurrence gives the minimal number of coins:

$$Min\#Coins(N) = \text{Min}$$

 $\rightarrow \quad 1 + Min\#Coins(N-1)$

 $\rightarrow \quad 1 + Min\#Coins(N-20)$

 $\rightarrow \quad 1 + Min\#Coins(N-50)$

With some memory (N intermediate computations), the minimum number of coins can be obtained after $N\times$#Coins operations. An actual set of coins can be reconstructing by **tracing back** the choices performed at each stage, leading to the minimum.

Remark: We still haven't won the million, as N has **exponential value compared to the length of its encoding**, so the algorithm does not qualify as *efficient* (i.e. polynomial).

Still, this approach is much more efficient than a brute-force enumeration:

$\Rightarrow$ Dynamic programming.

Foreword

Strategy #2: Brute force enumeration $\rightarrow$ #Coins^N (Ouch!)

Strategy #3: The following recurrence gives the minimal number of coins:

$Min\#Coins(N) = \text{Min}$

-  $\rightarrow \quad 1 + Min\#Coins(N-1)$
-  $\rightarrow \quad 1 + Min\#Coins(N-20)$
-  $\rightarrow \quad 1 + Min\#Coins(N-50)$

With some memory (N intermediate computations), the minimum number of coins can be obtained after $N\times$#Coins operations. An actual set of coins can be reconstructing by **tracing back** the choices performed at each stage, leading to the minimum.

Remark: We still haven't won the million, as N has **exponential value compared to the length of its encoding**, so the algorithm does not qualify as *efficient* (i.e. polynomial).

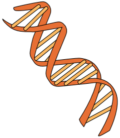
Still, this approach is much more efficient than a brute-force enumeration:

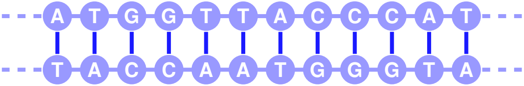
$\Rightarrow$ Dynamic programming.

Fundamental *dogma* of molecular biology

DNA

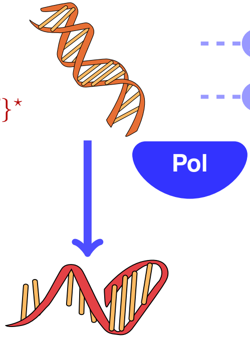
$\{A, C, G, T\}^\star$





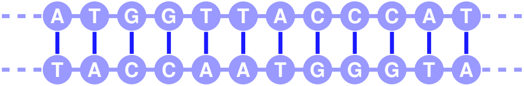
DNA

$\{A, C, G, T\}^\star$



RNAs

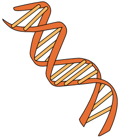
$\{A, C, G, U\}^\star$

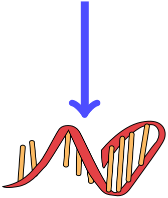


Fundamental *dogma* of molecular biology

DNA

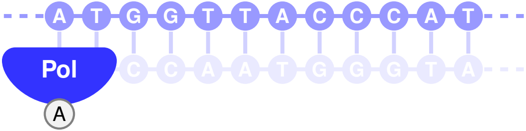
$\{A, C, G, T\}^\star$





RNAs

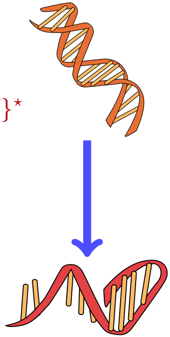
$\{A, C, G, U\}^\star$



Fundamental *dogma* of molecular biology

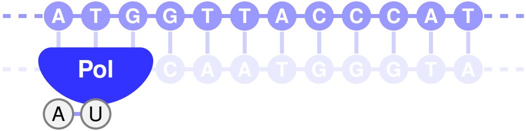
DNA

$\{A, C, G, T\}^\star$



RNAs

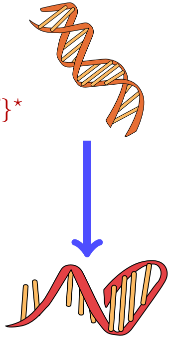
$\{A, C, G, U\}^\star$



Fundamental *dogma* of molecular biology

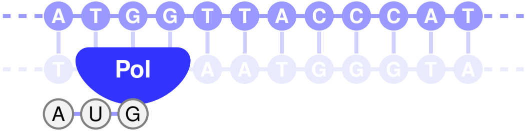
DNA

$\{A, C, G, T\}^\star$



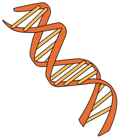
RNAs

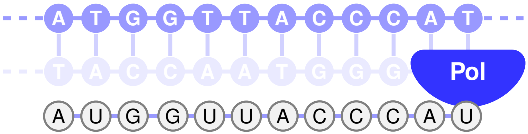
$\{A, C, G, U\}^\star$



DNA

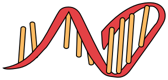
$\{A, C, G, T\}^\star$





RNAs

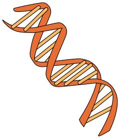
$\{A, C, G, U\}^\star$

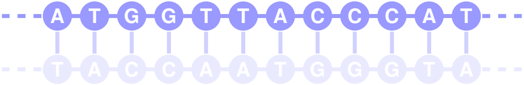


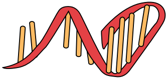
Fundamental *dogma* of molecular biology

DNA

$\{A, C, G, T\}^\star$

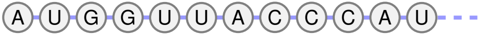






RNAs

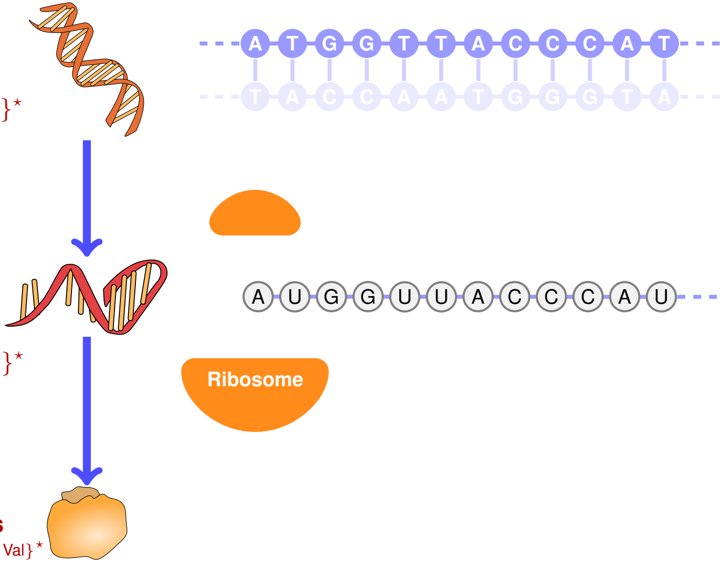
$\{A, C, G, U\}^\star$



Fundamental *dogma* of molecular biology

DNA

$\{A, C, G, T\}^\star$



RNAs

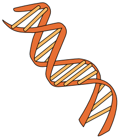
$\{A, C, G, U\}^\star$

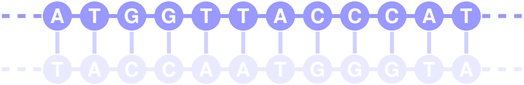
Proteins

$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^\star}_{20^+ \text{ Amino acids}}$

DNA

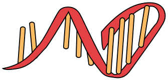
$\{A, C, G, T\}^{\star}$

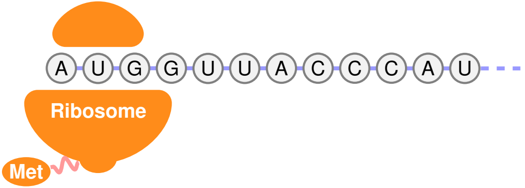




RNAs

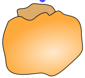
$\{A, C, G, U\}^{\star}$



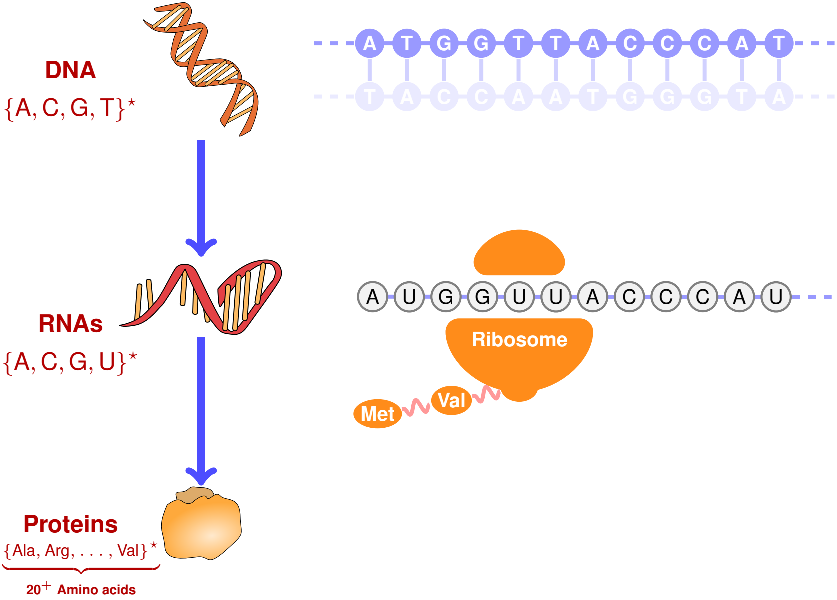


Proteins

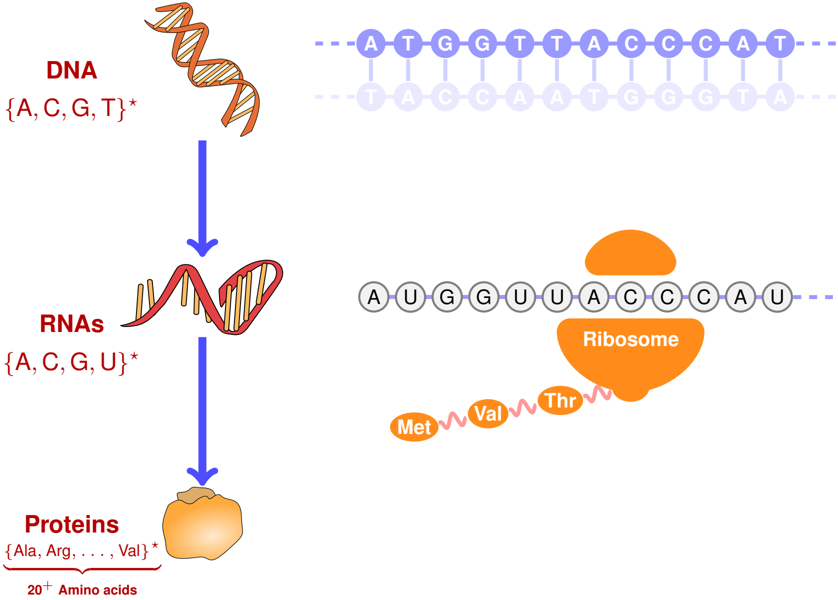
$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^{\star}}_{20^{+}\ \textbf{Amino acids}}$



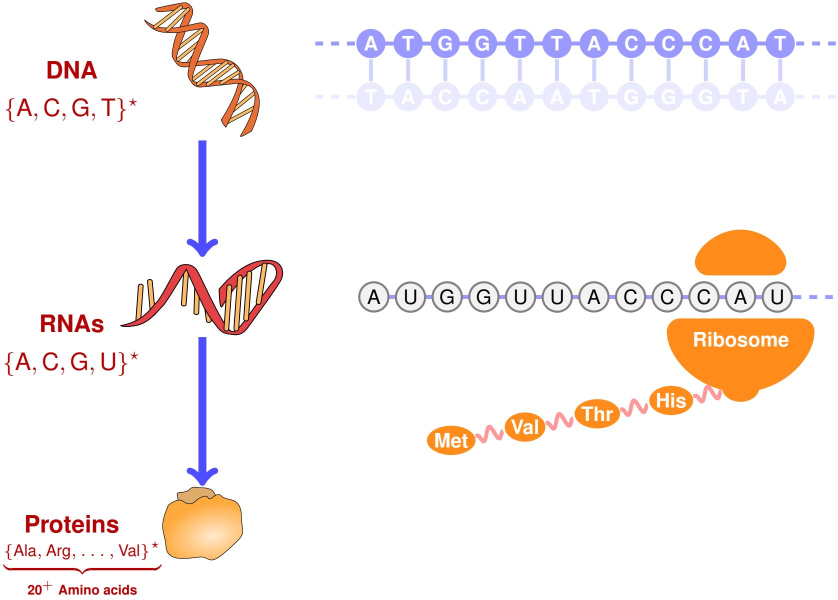
Fundamental *dogma* of molecular biology



Fundamental *dogma* of molecular biology

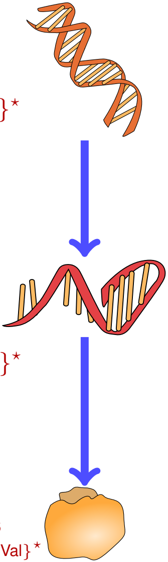


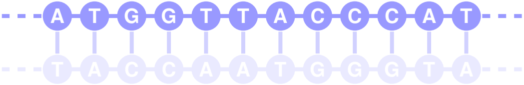
Fundamental *dogma* of molecular biology



DNA

$\{A, C, G, T\}^\star$





RNAs

$\{A, C, G, U\}^\star$

Proteins

$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^\star}_{}$

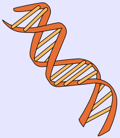
20^+ Amino acids

Fundamental *dogma* of molecular biology

THE CODE
(genes)

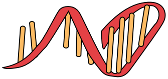
DNA

$\{A, C, G, T\}^\star$



A T G G T T A C C C A T

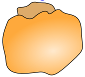
T A C C A A T G G G T A



RNAs

$\{A, C, G, U\}^\star$

A U G G U U A C C C A U



Proteins

$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^\star}$

20^+ Amino acids

Met Val Thr His Ile Leu His Asn

Fundamental *dogma* of molecular biology

THE CODE
(genes)

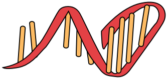
DNA

$\{A, C, G, T\}^\star$

A T G G T T A C C C A T
T A C C A A T G G G T A

RNAs

$\{A, C, G, U\}^\star$

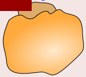


A U G G U U A C C C A U

THE MACHINE
(enzymes)

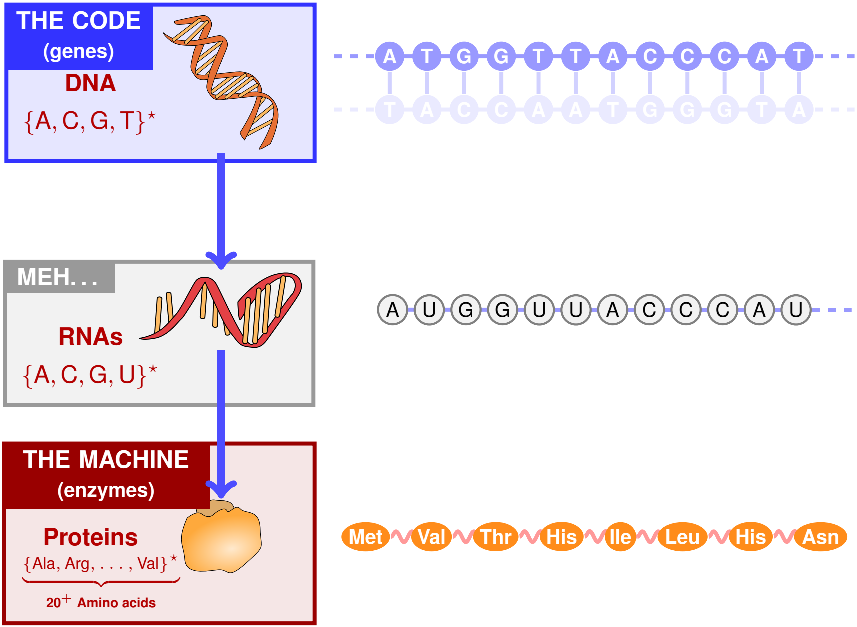
Proteins

$\underbrace{\{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}^\star}_{\textbf{20}^{+}\ \textbf{Amino acids}}$

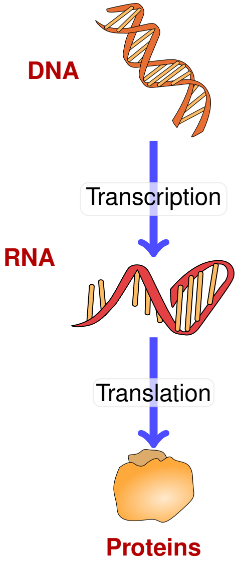


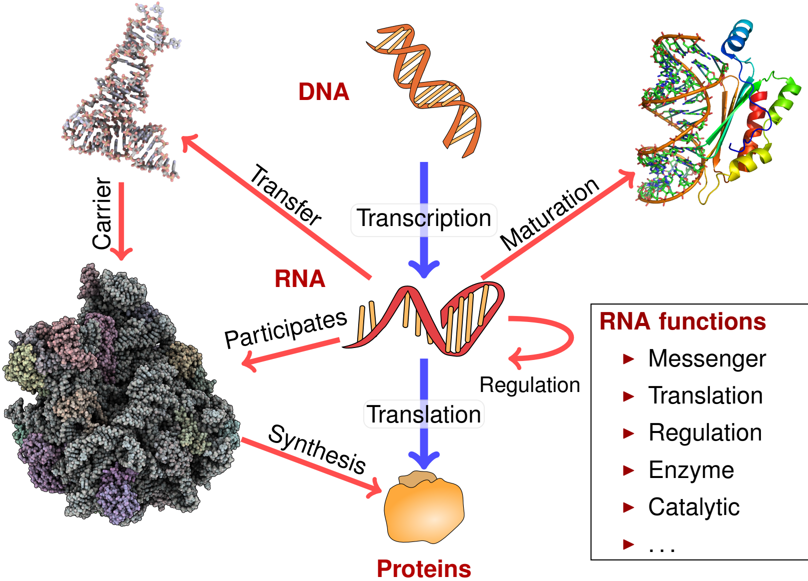
Met Val Thr His Ile Leu His Asn

Fundamental *dogma* of molecular biology



Fundamental *dogma* of molecular biology

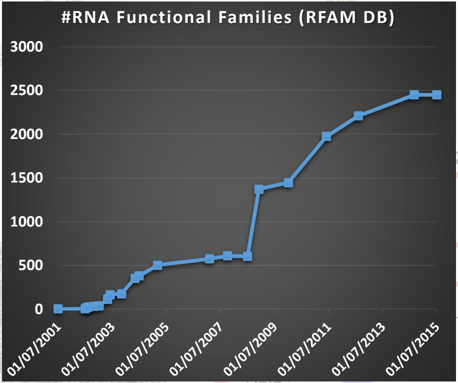




RNA functions

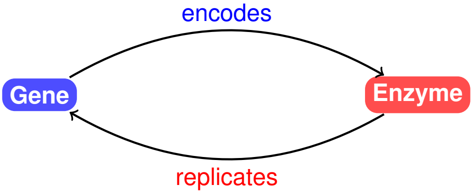
- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- ...

Fundamental *dogma* of molecular biology



RNA functions

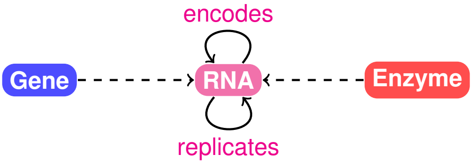
- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- . . .



A gene big enough to specify an enzyme would be too big to replicate accurately without the aid of an enzyme of the very kind that it is trying to specify. So the system *apparently cannot get started*.

[...] This is the RNA World. To see how plausible it is, we need to look at why proteins are good at being enzymes but bad at being replicators; at why DNA is good at replicating but bad at being an enzyme; and finally why *RNA might just be good enough at both roles to break out of the Catch-22.*

R. Dawkins. *The Ancestor's Tale: A Pilgrimage to the Dawn of Evolution*



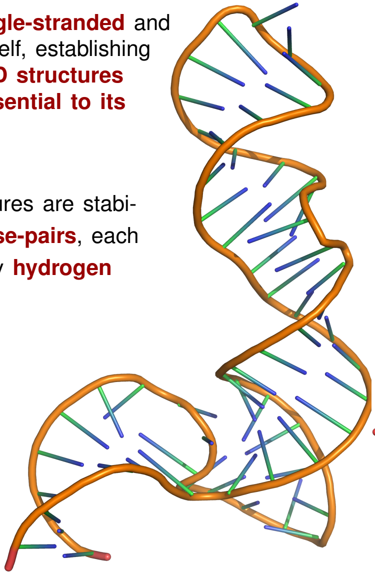
A gene big enough to specify an enzyme would be too big to replicate accurately without the aid of an enzyme of the very kind that it is trying to specify. So the system *apparently cannot get started*.

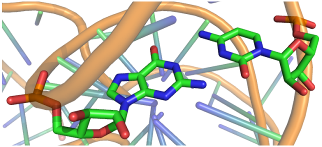
[...] This is the RNA World. To see how plausible it is, we need to look at why proteins are good at being enzymes but bad at being replicators; at why DNA is good at replicating but bad at being an enzyme; and finally why *RNA might just be good enough at both roles to break out of the Catch-22.*

R. Dawkins. *The Ancestor's Tale: A Pilgrimage to the Dawn of Evolution*

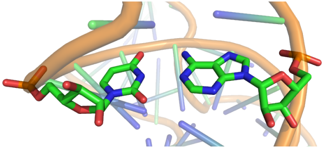
RNA is **single-stranded** and **folds** on itself, establishing **complex 3D structures** that are **essential to its function(s)**.

RNA structures are stabilized by **base-pairs**, each mediated by **hydrogen bonds**.



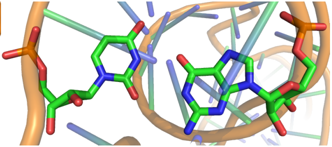


G/C



U/A

Watson/Crick base-pairs



U/G

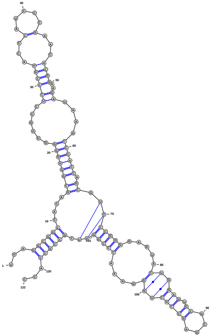
Wobble base-pair

Canonical base-pairs

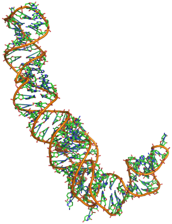
Three[1] levels of representation:

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

Primary structure



Secondary structure



Tertiary structure

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

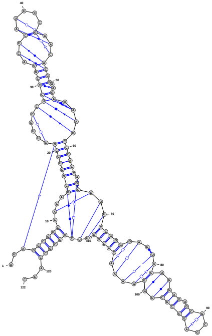
[1] Well, mostly...

RNA Structure(s)

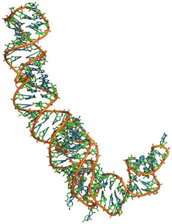
Three[1] levels of representation:

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Primary structure



Secondary^{+} structure

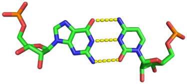


Tertiary structure

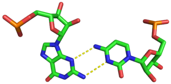
Source: 5s rRNA (PDB 1K73:B)

[1] Well, mostly...

- **Non-canonical base-pairs**
  Any base-pair other than {(A-U), (C-G), (G-U)}
  Or interacting on non-standard edge ($\neq$ WC/WC-Cis) [?].

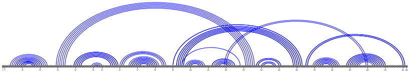


Canonique CG pair(WC/WC-Cis)



Non-canonique CG pair (Sugar/WC-Trans)

- **Pseudoknots (PKs)**



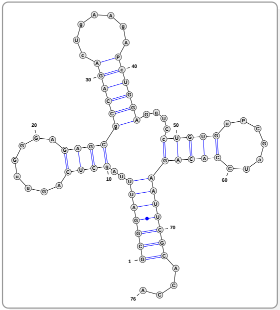
Pseudoknoted structure of group I ribozyme (PDBID: 1Y0Q:A)

Considering PKs may lead to better predictions, **but**:

- Some PK conformations are simply unfeasible;
- Folding *in silico* with general pseudoknots is NP-complete [?];

Still, folding on restricted classes of conformations seems promising [?].

Various representations for a versatile biomolecule



Outer-planar graphs

Hamiltonian-path, $\Delta(G) \leq 3$, 2-connected*

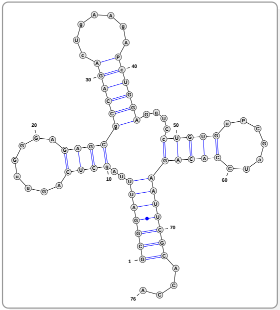
Supporting intuitions

Different representations

Common combinatorial structure

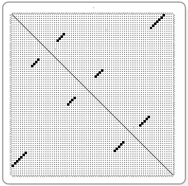
*Additional steric constraints

Various representations for a versatile biomolecule



Outer-planar graphs

Hamiltonian-path, $\Delta(G) \leq 3$, 2-connected*



Dot plots

Adjacency matrices*

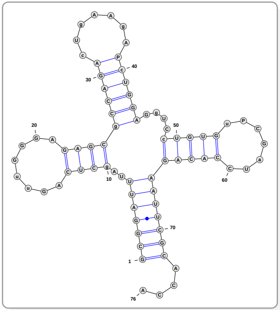
Supporting intuitions

Different representations

Common combinatorial structure

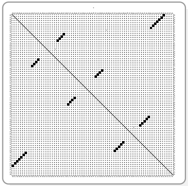
*Additional steric constraints

Various representations for a versatile biomolecule



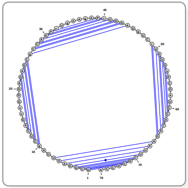
Outer-planar graphs

Hamiltonian-path, $\Delta(G) \leq 3$, 2-connected*



Dot plots

Adjacency matrices*



Non-crossing arc diagrams*

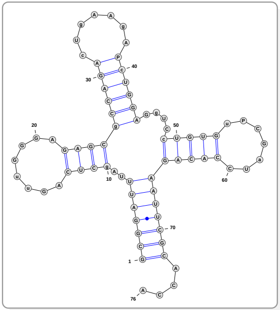
Supporting intuitions

Different representations

Common combinatorial structure

*Additional steric constraints

Various representations for a versatile biomolecule

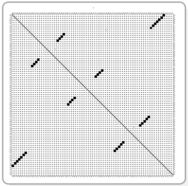


Outer-planar graphs

Hamiltonian-path, $\Delta(G) \leq 3$, 2-connected*

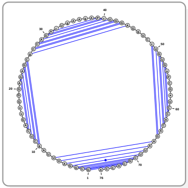
```
(((((((..((((........))))((((((.......))))))....(((((.......))))))))))))....
```

Motzkin words*



Dot plots

Adjacency matrices*



Non-crossing arc diagrams*

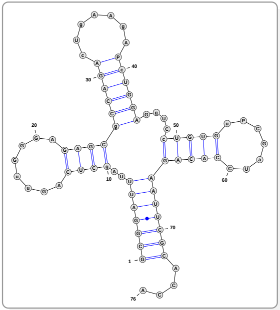
Supporting intuitions

Different representations

Common combinatorial structure

*Additional steric constraints

Various representations for a versatile biomolecule

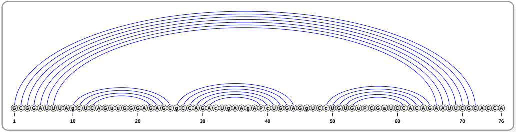


Outer-planar graphs

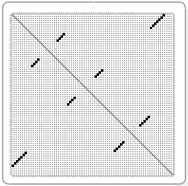
Hamiltonian-path, $\Delta(G) \leq 3$, 2-connected*

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

Motzkin words*

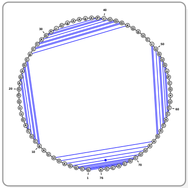


Non-crossing arc-annotated sequences*



Dot plots

Adjacency matrices*



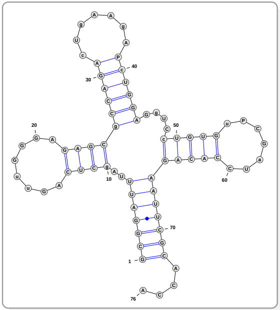
Non-crossing arc diagrams*

Supporting intuitions

Different representations

Common combinatorial structure

* Additional steric constraints

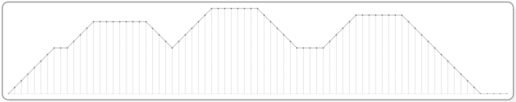


Outer-planar graphs

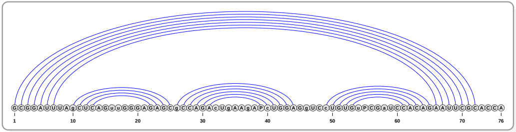
Hamiltonian-path, $\Delta(G) \leq 3$, 2-connected*

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

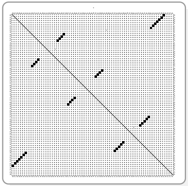
Motzkin words*



Positive 1D meanders* over $\mathcal{S} = \{+1, -1, 0\}$

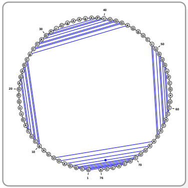


Non-crossing arc-annotated sequences*



Dot plots

Adjacency matrices*



Non-crossing arc diagrams*

Supporting intuitions

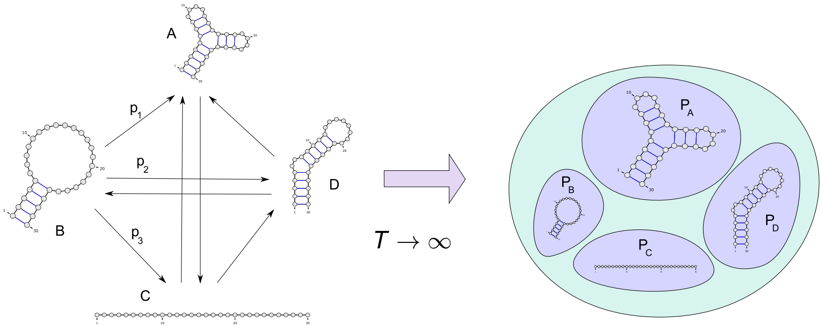
Different representations

Common combinatorial structure

* Additional steric constraints

Outline

At the nanoscopic scale, RNA structure *fluctuates* ($\approx$ Markov process).



Convergence towards a stationary distribution at the Boltzmann equilibrium, where the probability of a conformation only depends on its free-energy.
Corollary: Initial conformation does not matter.

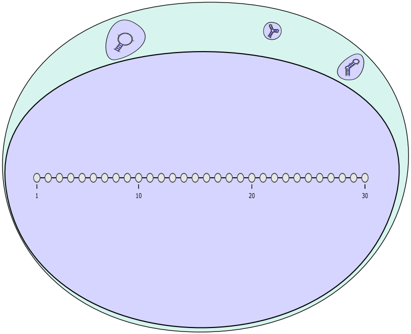
Questions: For a given **conformation space** and free-energy model:

A. Determine most stable (Minimum Free-Energy) structure at equilibrium;

B. Compute average properties of Boltzmann ensemble;

Away from equilibrium

Transcription: RNA synthesized, supposedly without structure[2]

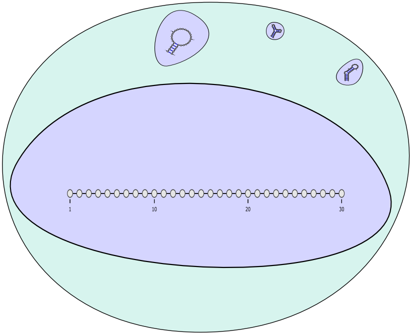


$T = 0$

But most mRNAs are degrade before 7h (Org.: Souris [?]).

[2] Except for co-transcriptional folding...

Transcription: RNA synthesized, supposedly without structure[2]

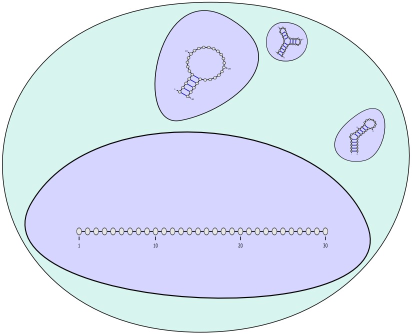


$T = 1h$

But most mRNAs are degrade before 7h (Org.: Souris [?]).

[2] Except for co-transcriptional folding...

Transcription: RNA synthesized, supposedly without structure[2]



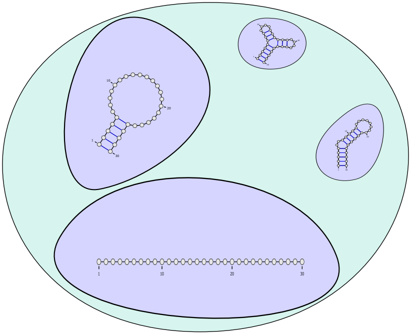
$$T = 2h$$

But most mRNAs are degrade before 7h (Org.: Souris [?]).

[2]Except for co-transcriptional folding...

Away from equilibrium

Transcription: RNA synthesized, supposedly without structure[2]



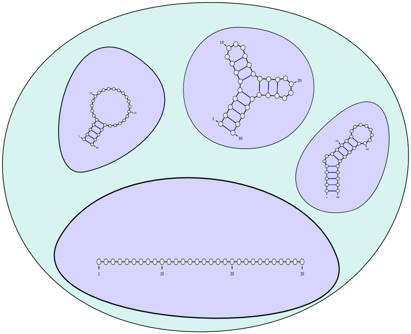
$$T = 5h$$

But most mRNAs are degrade before 7h (Org.: Souris [?]).

[2] Except for co-transcriptional folding...

Away from equilibrium

Transcription: RNA synthesized, supposedly without structure[2]

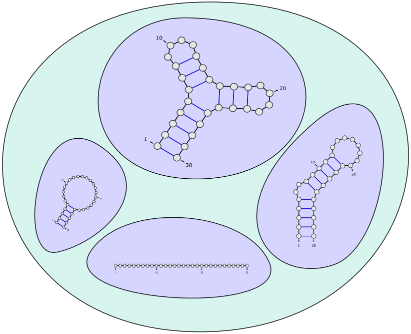


$$T = 10h$$

But most mRNAs are degrade before 7h (Org.: Souris [?]).

[2]Except for co-transcriptional folding...

Transcription: RNA synthesized, supposedly without structure[2]



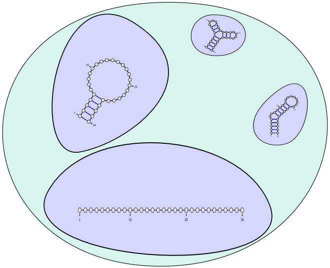
$T \to \infty$

But most mRNAs are degrade before 7h (Org.: Souris [?]).

[2] Except for co-transcriptional folding. . .

Away from equilibrium

Transcription: RNA synthesized, supposedly without structure[2]



$T = 10h$

But most mRNAs are degrade before 7h (Org.: Souris [?]).

A. Determine most stable (Minimum Free-Energy) structure at equilibrium;

B. Compute average properties of Boltzmann ensemble;

C. Determine most likely structure at finite time T.
(c.f. H. Isambert through simulation, NP-complete deterministically [?])

[2] Except for co-transcriptional folding...

Dynamic programming: General principle

Dynamic programming = General optimization technique.
Prerequisite: Optimal solution for problem P can be derived from solutions to strict sub-problems of P.

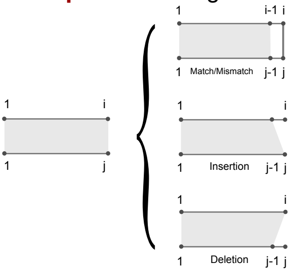
Bioinformatics :

Discete solution space (alignments, structures...)

+ Additively-inherited objective function (cost, log-odd score, energy...)

$\Rightarrow$ Efficient dynamic programming scheme

Example: Local Alignment(Smith/Waterman)



$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max\begin{cases} W(i-1,j-1)+m_{i,j}\\ W(i-1,j)+p_i\\ W(i,j-1)+p_d\end{cases}
\end{aligned}
$$

Algorithmic details

Dynamic programming scheme defines a space of (sub)problems and a recurrence that relates the score of a problem to that of smaller problems.

Given a scheme, two steps :

- **Matrix filling:** Computation and tabulation of best scores (Computed from smaller problems to larger ones).
- **Traceback:** Reconstruct best solution from contributing subproblems.

Complexity of algorithm depends on:

- **Cardinality** of sub-problem space
- **Number of alternatives** considers at each step (#Terms in recurrence)

Smith&Waterman example:

- i: $1 \to n+1 \Rightarrow \Theta(n)$
- j: $1 \to m+1 \Rightarrow \Theta(m)$
- 3 operations at each step

$\Rightarrow \Theta(m.n)$ time/memory

$$\begin{aligned} W(i,0) &= 0 \\ W(0,j) &= 0 \\ W(i,j) &= \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases} \end{aligned}$$

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max\begin{cases} W(i-1,j-1) + m_{i,j}\\ W(i-1,j) + p_i\\ W(i,j-1) + p_d\end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0								
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2							
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Délétion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1						
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max\begin{cases} W(i-1,j-1)+m_{i,j}\\ W(i-1,j)+p_i\\ W(i,j-1)+p_d\end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2					
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1				
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Délétion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0								
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$\begin{aligned} W(i,0) &= 0 \\ W(0,j) &= 0 \\ W(i,j) &= \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases} \end{aligned}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0								
A	0								
C	0								
A	0								
C	0								
A	0								

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max\begin{cases} W(i-1,j-1)+m_{i,j}\\ W(i-1,j)+p_i\\ W(i,j-1)+p_d\end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0								
C	0								
A	0								
C	0								
A	0								

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0								
A	0								
C	0								
A	0								

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max\begin{cases} W(i-1,j-1) + m_{i,j}\\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Délétion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max\begin{cases} W(i-1,j-1) + m_{i,j}\\ W(i-1,j) + p_i\\ W(i,j-1) + p_d \end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`

Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

A
A

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max \begin{cases} W(i-1,j-1) + m_{i,j}\\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

-	A
T	A

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Délétion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

C	-	A
C	T	A

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max\begin{cases} W(i-1,j-1)+m_{i,j}\\ W(i-1,j)+p_i\\ W(i,j-1)+p_d \end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

A	C	-	A
A	C	T	A

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`

Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

C	A	C	-	A
C	A	C	T	A

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Délétion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

A	C	A	C	-	A
A	C	A	C	T	A

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

C	A	C	A	C	-	A
C	A	C	A	C	T	A

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Déletion $p_i = p_j = -1$

$$W(i,0) = 0$$
$$W(0,j) = 0$$
$$W(i,j) = \max \begin{cases} W(i-1,j-1) + m_{i,j} \\ W(i-1,j) + p_i \\ W(i,j-1) + p_d \end{cases}$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

```
G C A C A C - A
- C A C A C T A
```

Complete example

Example: Local alignment of `AGCACACA` and `ACACACTA`
Costs: Match $m_{i,j} = +2$, Insertion/Délétion $p_i = p_j = -1$

$$
\begin{aligned}
W(i,0) &= 0\\
W(0,j) &= 0\\
W(i,j) &= \max\begin{cases} W(i-1,j-1) + m_{i,j}\\ W(i-1,j) + p_i\\ W(i,j-1) + p_d\end{cases}
\end{aligned}
$$

		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Best alignment

A	G	C	A	C	A	C	-	A
A	-	C	A	C	A	C	T	A

Properties of DP schemes

Necessary properties:

- **Correctness:** $\forall$ sub-problem, the computed value must indeed maximize the objective function .

Proofs usually inductive, and quite technical, but very systematic.

Desirable properties of DP schemes:

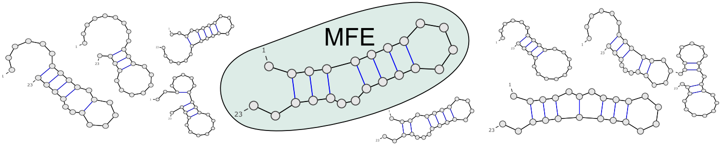
- **Completeness** of space of solutions generated by decomposition. Algorithmic tricks, by *cutting branches*, may violate this property.
- **Unambiguity:** Each solution is generated at most once.

$\Rightarrow$ Under these properties, one can **enumerate** solution space.

Folding by minimizing free-energy

Problem A: Determine Minimum Free-Energy structure (MFE).

***Ab* initio folding prediction =**
Predict RNA structure from its sequence ω only.



- **Conformations:** Set S_ω of secondary structures compatible (w.r.t. base-pairing constraints) with primary structure ω .
- **Free-Energy:** Function $E_{\omega,S}$ (KCal.mol^{-1}), additive on motifs occurring in any sequence/conformation couple (ω, S).
- **Native structure:** Functional conformation of the biomolecule.

  Remarks:
  - Not necessarily unique (Kinetics, or bi-stable structures);
  - In presence of PKs $\rightarrow$ Ambiguous: Which is the native conformation?

Nussinov/Jacobson model

Nussinov/Jacobson energy model (NJ)

Base-pair maximization (with a twist):

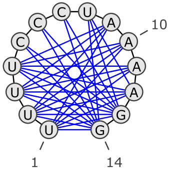
- Additive model on independently contributing base-pairs;
- Canonical base-pairs only: Watson/Crick (A/U,C/G) and Wobble (G/U)

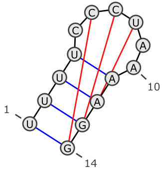
$$\Rightarrow E_{\omega,S} = -\#Paires(S)$$

Folding in NJ model $\Leftrightarrow$ Base-pair (weight) maximization

Example:

UUUUCCCUAAAAGG





Variant: Weight each pair with −#Hydrogen bonds

$\Delta G(G\equiv C) = -3$ $\Delta G(A{=}U) = -2$ $\Delta G(G{-}U) = -1$

Nussinov/Jacobson energy model (NJ)

Base-pair maximization (with a twist):

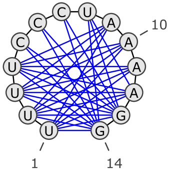
- Additive model on independently contributing base-pairs;
- Canonical base-pairs only: Watson/Crick (A/U,C/G) and Wobble (G/U)

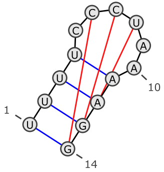
$$\Rightarrow E_{\omega,S} = -\#Paires(S)$$

Folding in NJ model $\Leftrightarrow$ Base-pair (weight) maximization

Example:

UUUUCCCUAAAAGG

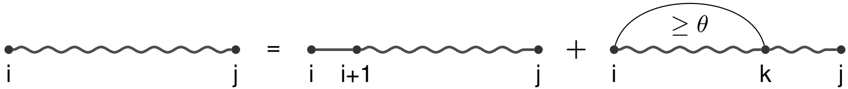




Variant: Weight each pair with −#Hydrogen bonds

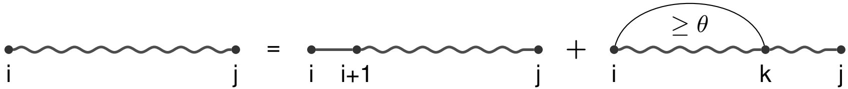
$\Delta G(G{\equiv}C) = -3$ $\Delta G(A{=}U) = -2$ $\Delta G(G{-}U) = -1$

Nussinov/Jacobson DP scheme



$$
\begin{aligned}
N_{i,t} &= 0, \quad \forall t \in [i, i+\theta] \\
N_{i,j} &= \min \begin{cases} N_{i+1,j} & i \text{ unpaired} \\ \min\limits_{k=i+\theta+1}^{j} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j} & i \text{ paired with } k \end{cases}
\end{aligned}
$$

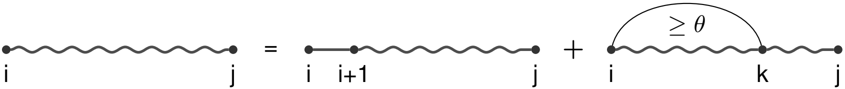
Nussinov/Jacobson DP scheme



$$
\begin{aligned}
N_{i,t} &= 0, \quad \forall t \in [i, i+\theta] \\
N_{i,j} &= \min \begin{cases} N_{i+1,j} & i \text{ unpaired} \\ \displaystyle\min_{k=i+\theta+1}^{j} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j} & i \text{ paired with } k \end{cases}
\end{aligned}
$$

Correctness. Goal = Show that MFE over interval $[i, j]$ is indeed found in $N_{i,j}$ after completing the computation. Proceed by induction:

- Assume that property holds for any $[i', j']$ such that $j' - i' < n$.
- Consider $[i, j], j - i = n$. Let $\text{MFE}_{i,j}$:= Base-pairs of best struct. on $[i, j]$. Then first position i in $\text{MFE}_{i,j}$ = is either:
  - **Unpaired:** $\text{MFE}_{i,j} = \text{MFE}_{i+1,j}$ $\rightarrow$ free-energy $= N_{i+1,j}$
  - **Paired to k:** $\text{MFE}_{i,j} = \{(i,k)\} \cup \text{MFE}_{i+1,k-1} \cup \text{MFE}_{k+1,j}$.
    (Indeed, any BP between $[i+1, k-1]$ and $[k+1, j]$ would cross (i, k))
    $\rightarrow$ free-energy $= \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j}$



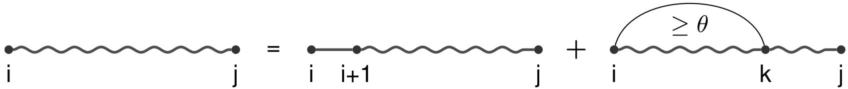
$$N_{i,t} = 0, \quad \forall t \in [i, i+\theta]$$

$$N_{i,j} = \min \begin{cases} N_{i+1,j} & i \text{ unpaired} \\ \min\limits_{k=i+\theta+1}^{j} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j} & i \text{ paired with } k \end{cases}$$

Correctness. Goal = Show that MFE over interval $[i, j]$ is indeed found in $N_{i,j}$ after completing the computation. Proceed by induction:

- Assume that property holds for any $[i', j']$ such that $j' - i' < n$.
- Consider $[i, j], j - i = n$. Let $\text{MFE}_{i,j}$:= Base-pairs of best struct. on $[i, j]$. Then first position i in $\text{MFE}_{i,j}$ = is either:
  - **Unpaired:** $\text{MFE}_{i,j} = \text{MFE}_{i+1,j}$ $\rightarrow$ free-energy $= N_{i+1,j}$
  - **Paired to k:** $\text{MFE}_{i,j} = \{(i,k)\} \cup \text{MFE}_{i+1,k-1} \cup \text{MFE}_{k+1,j}$.
    (Indeed, any BP between $[i+1, k-1]$ and $[k+1, j]$ would cross (i, k))
    $\rightarrow$ free-energy $= \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j}$

Nussinov/Jacobson DP scheme

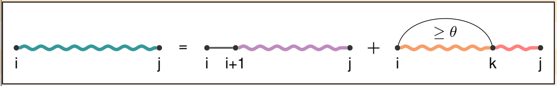


$$
\begin{aligned}
N_{i,t} &= 0, \quad \forall t \in [i, i+\theta] \\
N_{i,j} &= \min \begin{cases} N_{i+1,j} & i \text{ unpaired} \\ \min\limits_{k=i+\theta+1}^{j} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j} & i \text{ paired with } k \end{cases}
\end{aligned}
$$

Correctness. Goal = Show that MFE over interval $[i, j]$ is indeed found in $N_{i,j}$ after completing the computation. Proceed by induction:

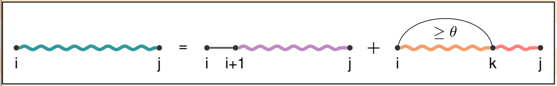
- Assume that property holds for any $[i', j']$ such that $j' - i' < n$.
- Consider $[i, j], j - i = n$. Let $\text{MFE}_{i,j} :=$ Base-pairs of best struct. on $[i, j]$. Then first position i in $\text{MFE}_{i,j} =$ is either:
  - **Unpaired:** $\text{MFE}_{i,j} = \text{MFE}_{i+1,j}$ $\rightarrow$ free-energy $= N_{i+1,j}$
  - **Paired to k:** $\text{MFE}_{i,j} = \{(i, k)\} \cup \text{MFE}_{i+1,k-1} \cup \text{MFE}_{k+1,j}$.
    (Indeed, any BP between $[i+1, k-1]$ and $[k+1, j]$ would cross (i, k))
    $\rightarrow$ free-energy $= \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j}$

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



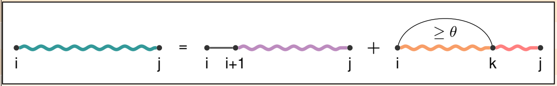
Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

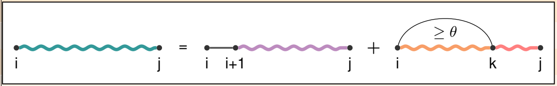


Nussinov/Jacobson

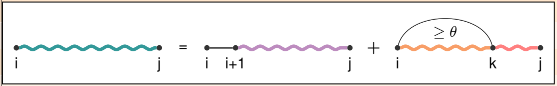
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

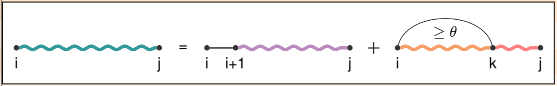


	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



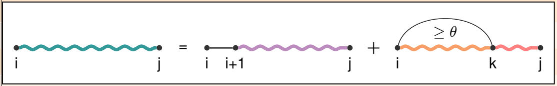
Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

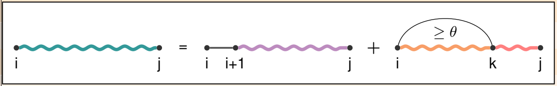


Nussinov/Jacobson

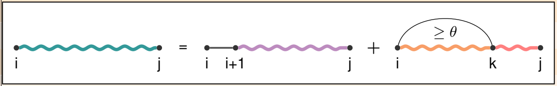
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

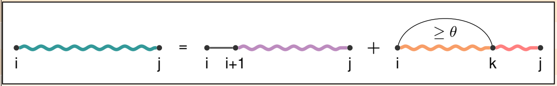


	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



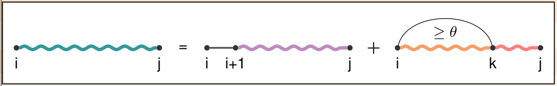
Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



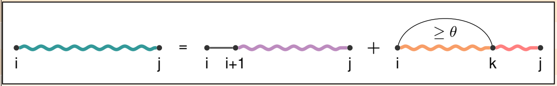
Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

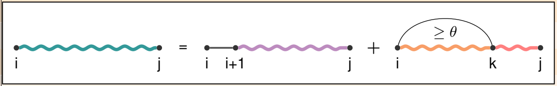


Nussinov/Jacobson

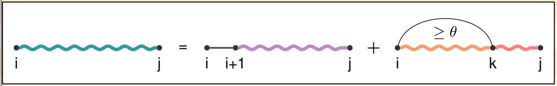
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



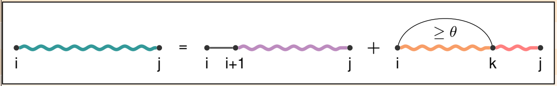
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



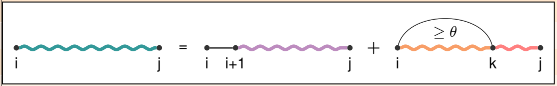
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	.	.	.	.	.	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



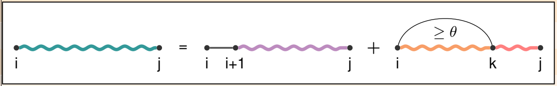
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	.	.	.	.	.	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



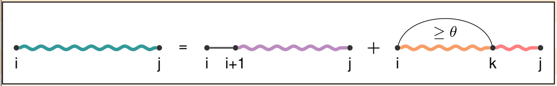
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	.	.	.	.	.	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

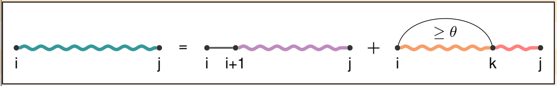


	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



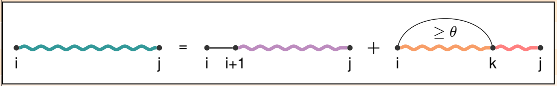
Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



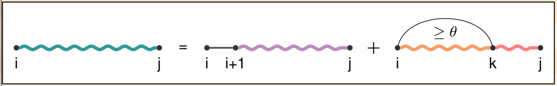
Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

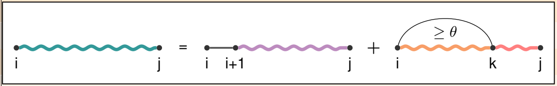


Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

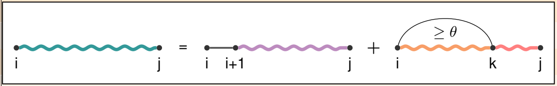


	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



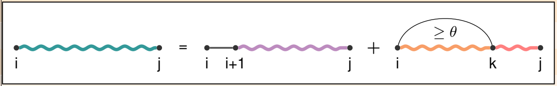
Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	.	.	.	.	.	.	.	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



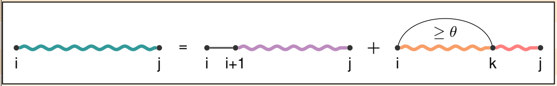
Nussinov/Jacobson

	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	(	.	.	.	.	.	)	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0

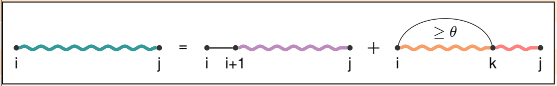


Nussinov/Jacobson

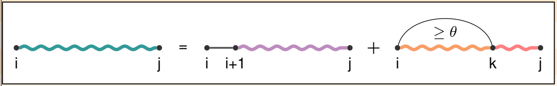
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	(	.	.	.	.	.	)	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



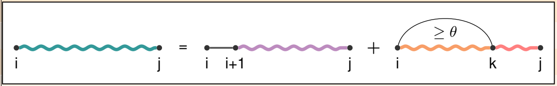
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	(	.	.	.	.	.	)	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	(	(	.	.	.	)	)	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



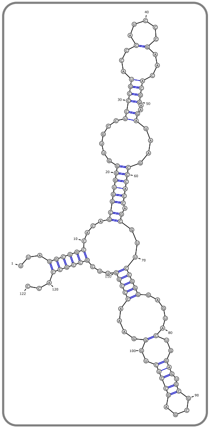
	C	G	G	A	U	A	C	U	U	C	U	U	A	G	A	C	G	A
	(	(	(	.	.	.	)	.	(	(	.	.	.	)	)	)	)	.
C	0	0	0	0	0	0	3	4	4	6	6	6	6	9	9	11	14	14
G		0	0	0	0	0	3	4	4	6	6	6	6	7	9	11	11	11
G			0	0	0	0	3	3	3	5	5	5	5	6	8	10	10	10
A				0	0	0	0	2	2	2	2	4	4	5	7	7	8	10
U					0	0	0	0	0	0	2	2	4	5	7	7	8	10
A						0	0	0	0	0	2	2	2	5	5	5	8	8
C							0	0	0	0	0	0	2	5	5	5	8	8
U								0	0	0	0	0	2	3	5	5	6	7
U									0	0	0	0	2	3	5	5	5	7
C										0	0	0	0	3	3	3	5	5
U											0	0	0	0	2	2	2	3
U												0	0	0	0	0	1	2
A													0	0	0	0	0	0
G														0	0	0	0	0
A															0	0	0	0
C																0	0	0
G																	0	0
A																		0



Turner energy model

Based on unambiguous decomposition of 2^{ary} structure into loops:

- Internal loops
- Bulges
- Terminal loops
- Multi loops
- Stackings



Free-energy Δ G of a loop depend on bases, assymmetry, dangles . . .

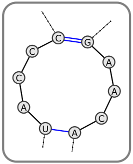
Experimentally determined
+ Interpolated for larger loops.

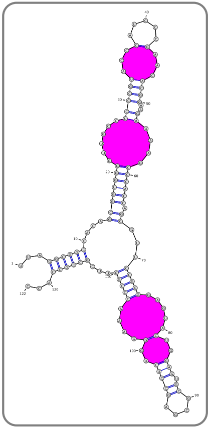
Improved results by taking stacking into account.

Turner energy model

Based on unambiguous decomposition of 2^{ary} structure into loops:

- Internal loops
- Bulges
- Terminal loops
- Multi loops
- Stackings





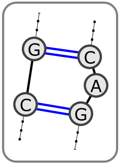
Free-energy Δ G of a loop depend on bases, assymmetry, dangles . . .

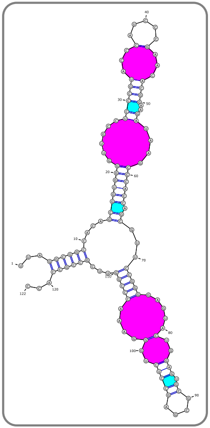
Experimentally determined
+ Interpolated for larger loops.

Improved results by taking stacking into account.

Based on unambiguous decomposition of 2^{ary} structure into loops:

- Internal loops
- Bulges
- Terminal loops
- Multi loops
- Stackings





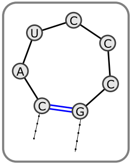
Free-energy Δ G of a loop depend on bases, assymmetry, dangles . . .

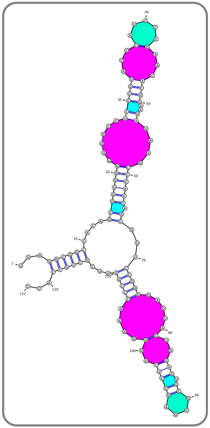
Experimentally determined
+ Interpolated for larger loops.

Improved results by taking stacking into account.

Based on unambiguous decomposition of 2^{ary} structure into loops:

- Internal loops
- Bulges
- Terminal loops
- Multi loops
- Stackings





Free-energy Δ G of a loop depend on bases, assymmetry, dangles . . .

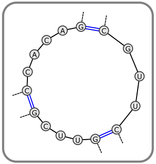
Experimentally determined
+ Interpolated for larger loops.

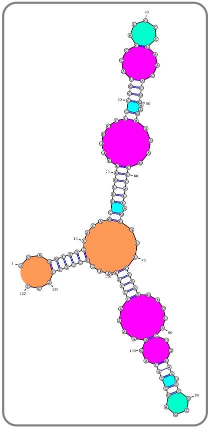
Improved results by taking stacking into account.

Turner energy model

Based on unambiguous decomposition of 2^{ary} structure into loops:

- Internal loops
- Bulges
- Terminal loops
- Multi loops
- Stackings





Free-energy Δ G of a loop depend on bases, assymmetry, dangles . . .

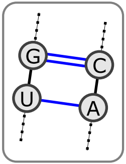
Experimentally determined
+ Interpolated for larger loops.

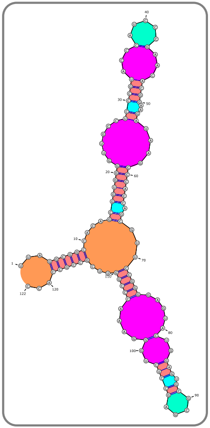
Improved results by taking stacking into account.

Turner energy model

Based on unambiguous decomposition of 2^ary^ structure into loops:

- Internal loops
- Bulges
- Terminal loops
- Multi loops
- Stackings



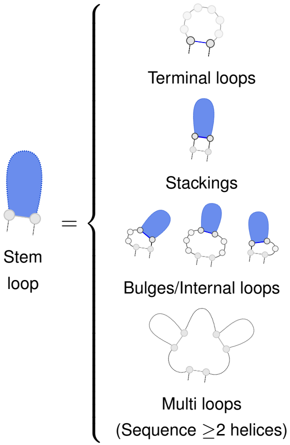


Free-energy Δ G of a loop depend on bases, assymmetry, dangles ...

Experimentally determined
+ Interpolated for larger loops.

Improved results by taking stacking into account.

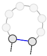
MFE DP equations



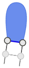
MFE DP equations

Stem loop

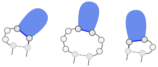
=



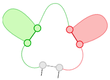
Terminal loops



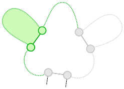
Stackings



Bulges/Internal loops

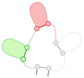


Multi loops
(Sequence $\geq$2 helices)



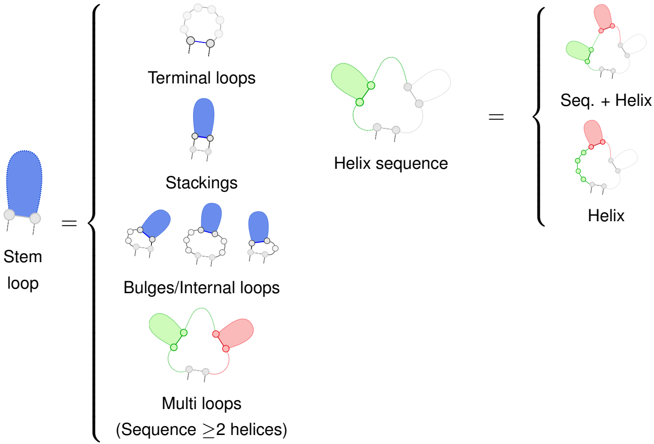
Helix sequence

=

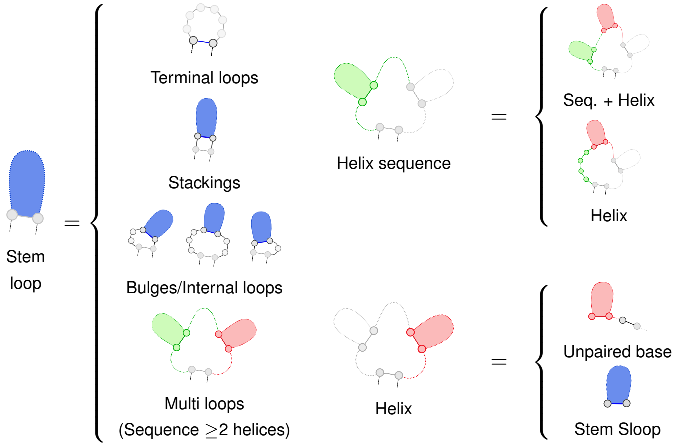


Seq. + Helix

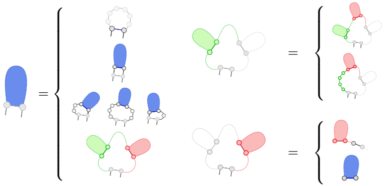
MFE DP equations



MFE DP equations



- $E_H(i,j)$: Energy of terminal loop *enclosed by* (i,j) pair
- $E_{BI}(i,j)$: Energy of bulge or internal loop *enclosed by* (i,j) pair
- $E_S(i,j)$: Energy of stacking $(i,j)/(i+1,j-1)$
- Penalty for multi loop (a), and occurrences of unpaired base (b) and helix (c) in multi loops.



DP recurrence

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

Backtracking to reconstruct MFE structure:

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

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

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

Complexity:

For each min, $\mathcal{O}(n)$ potential contributors
$\Rightarrow$ **Worst-case** complexity in $\mathcal{O}(n^2)$ for **naive backtrack**.
Keep best contributor for each Min $\Rightarrow$ **Backtracking in $\mathcal{O}(n)$**

$\Rightarrow$ `UnaFold` [?]/`RNAFold` [?] compute the MFE for the Turner model in **overall**[3] time/space complexities in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$

[3] Using a trick/restriction for internal loops...

Backtracking to reconstruct MFE structure:

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

Complexity:

For each min, $\mathcal{O}(n)$ potential contributors
$\Rightarrow$ **Worst-case** complexity in $\mathcal{O}(n^2)$ for **naive backtrack**.
Keep best contributor for each Min $\Rightarrow$ **Backtracking in $\mathcal{O}(n)$**

$\Rightarrow$ `UnaFold` [?]/`RNAFold` [?] compute the MFE for the Turner model in **overall**[3] time/space complexities in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$

[3] Using a trick/restriction for internal loops...

Backtracking to reconstruct MFE structure:

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

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

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

Complexity:

For each min, $\mathcal{O}(n)$ potential contributors
$\Rightarrow$ **Worst-case** complexity in $\mathcal{O}(n^2)$ for **naive backtrack**.
Keep best contributor for each Min $\Rightarrow$ **Backtracking in $\mathcal{O}(n)$**

$\Rightarrow$ `UnaFold` [?]/`RNAFold` [?] compute the MFE for the Turner model in **overall**[3] time/space complexities in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$

[3] Using a trick/restriction for internal loops...

Backtracking to reconstruct MFE structure:

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

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

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

Complexity:

For each min, $\mathcal{O}(n)$ potential contributors
$\Rightarrow$ **Worst-case** complexity in $\mathcal{O}(n^2)$ for **naive backtrack**.
Keep best contributor for each Min $\Rightarrow$ **Backtracking in $\mathcal{O}(n)$**

$\Rightarrow$ `UnaFold` [?]/`RNAFold` [?] compute the MFE for the Turner model in **overall**[3] time/space complexities in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$

[3] Using a trick/restriction for internal loops...

Remontée (Backtracking)

Backtracking to reconstruct MFE structure:

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

Complexity:

For each min, $\mathcal{O}(n)$ potential contributors
$\Rightarrow$ **Worst-case** complexity in $\mathcal{O}(n^2)$ for **naive backtrack**.
Keep best contributor for each Min $\Rightarrow$ **Backtracking in $\mathcal{O}(n)$**

$\Rightarrow$ `UnaFold` [?]/`RNAFold` [?] compute the MFE for the Turner model in **overall**[3] time/space complexities in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$

[3] Using a trick/restriction for internal loops...

Remontée (Backtracking)

Backtracking to reconstruct MFE structure:

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

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

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

Complexity:

For each min, $\mathcal{O}(n)$ potential contributors
$\Rightarrow$ **Worst-case** complexity in $\mathcal{O}(n^2)$ for **naive backtrack**.
Keep best contributor for each Min $\Rightarrow$ **Backtracking in $\mathcal{O}(n)$**

$\Rightarrow$ `UnaFold` [?]/`RNAFold` [?] compute the MFE for the Turner model in **overall**[3] time/space complexities in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$

[3] Using a trick/restriction for internal loops...

Backtracking to reconstruct MFE structure:

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

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

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

Complexity:

For each min, $\mathcal{O}(n)$ potential contributors
$\Rightarrow$ **Worst-case** complexity in $\mathcal{O}(n^2)$ for **naive backtrack**.
Keep best contributor for each Min $\Rightarrow$ **Backtracking in $\mathcal{O}(n)$**

$\Rightarrow$ `UnaFold` [?]/`RNAFold` [?] compute the MFE for the Turner model in **overall**[3] time/space complexities in $\mathcal{O}(n^3)/\mathcal{O}(n^2)$

[3] Using a trick/restriction for internal loops...

Two main approaches

Definition (Ab initio folding)

Starting from sequence, find conformation that minimizes free-energy.

Advantages:

- Mechanical nature allows the (in)validation of models
- Reasonable complexity $\mathcal{O}(n^3)/\mathcal{O}(n^2)$ time/space
- *Exhaustive* nature

Limitations:

- Hard to include PKs
- Highly dependent on energy model
- No cooperativity
- Limited performances

Definition (Comparative approach)

Starting from homologous sequences, postulate common structure and find best possible tradeoff between folding & alignment.

Avantages :

- Better performances
- (Limited) cooperativity
- Self-improving

Limitations

- Easily unreasonable complexity
- Non exhaustive search
- Captures *transient* structures

Definition (Ab initio folding)

Starting from sequence, find conformation that minimizes free-energy.

Advantages:

- Mechanical nature allows the (in)validation of models
- Reasonable complexity $\mathcal{O}(n^3)/\mathcal{O}(n^2)$ time/space
- *Exhaustive* nature

Limitations:

- Hard to include PKs
- Highly dependent on energy model
- No cooperativity
- Limited performances

Definition (Comparative approach)

Starting from homologous sequences, postulate common structure and find best possible tradeoff between folding & alignment.

Avantages :

- Better performances
- (Limited) cooperativity
- Self-improving

Limitations

- Easily unreasonable complexity
- Non exhaustive search
- Captures *transient* structures

Performances

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Sequence

Taille	Sens.
<700	70-73%

[?, ?]

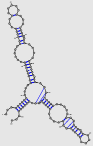
16s,23s	∼50%

MCC∼ 0.5 [?]

Thermodynamics

Sens.	Spé.	MCC.
∼75%	∼75%	0.8

Comparative [?]



2^{ary} Structure

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

Performances

Taille <700 — Sens. 70-73% [?, ?]

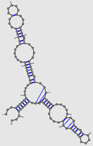
16s,23s — ~50%

MCC~ 0.5 [?]

Thermodynamics

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Sequence



2^{ary} Structure

Sens.	Spé.	MCC.
~75%	~75%	0.8

Comparative [?]

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

Performances

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

Sequence

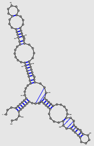
Taille	Sens.
<700	70-73%

[?, ?]

16s,23s	$\sim$50%

MCC$\sim$ 0.5 [?]

Thermodynamics



2^{ary} Structure

Sens.	Spé.	MCC.
$\sim$75%	$\sim$75%	0.8

Comparative [?]

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

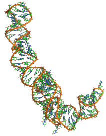
Towards a 3D ab-initio prediction

Goal: From sequence to all-atom/coarse grain 3D models!!!

- Comparative models + Molecular dynamics: `RNA2D3D` [?]
- Pipeline `MC-Fold`/`MC-sym` [?]

UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Séquence



3^{ary} Structure

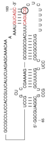
Goal: From sequence to all-atom/coarse grain 3D models!!!

- Comparative models + Molecular dynamics: `RNA2D3D` [?]
- Pipeline `MC-Fold/MC-sym` [?]

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

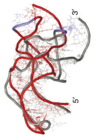
Séquence

Comparative `KNetFold`



2^{ary} Structure + Pseudoknots

Molecular dynamics



3^{ary} Structure

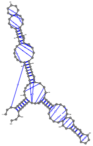
Goal: From sequence to all-atom/coarse grain 3D models!!!

- Comparative models + Molecular dynamics: `RNA2D3D` [?]
- Pipeline `MC-Fold/MC-sym` [?]

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

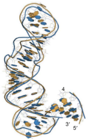
Séquence

→ `MCFold` →



2^{ary+} Structure

→ `MCSym` →



3^{ary} Structure

References I

Exercise: Parsing/folding RNAs (Python)

http://www.lix.polytechnique.fr/~ponty/enseignement/2015-2016-BIM-TP1.pdf