1 Introduction
   - Dynamic programming 101
   - Dynamic programming framework

2 Variations on RNA folding
   - Why RNA?
   - RNA folding
   - RNA Structure(s)
   - Some representations of RNA structure
   - Thermodynamics vs Kinetics

3 Free-energy minimization
   - Nussinov-style RNA folding
   - Turner energy model
   - MFold/Unafold
   - Performances and the comparative approach
   - Towards a 3D ab-initio prediction

4 Boltzmann ensemble
   - Nussinov: Minimisation ⇒ Counting
   - Computing the partition function
   - Statistical sampling
... 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.

\[
21 = \begin{array}{c} \text{20} \end{array} + \begin{array}{c} \text{1} \end{array}
\]

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.

\[
21 = 20 + 1
\]

\[
55 = 50 + 5 + 5 + 5 + 5 + 1 + 1
\]

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.

\[
21 = \hspace{0.5cm} + \\
55 = \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} \\
60 = \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} + \hspace{0.5cm} ??
\]
...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.

$$21 = \text{20} + \text{1}$$
$$55 = \text{50} + \text{1} + \text{1} + \text{1} + \text{1} + \text{1}$$
$$60 = \text{50} + \text{1} + \text{1} + \text{1} + \text{1} + \text{1} + \text{1} + \text{1} + \text{1} + \text{1} + \text{1} + \text{1}$$
$$= \text{50} + \text{20} + \text{1}$$

Problem *a priori* (?!) non-solvable using such a *greedy* approach, as a (simpler) problem is already NP-complete (thus Efficient solution ⇒ 1M$).
Strategy #2: Brute force enumeration $\rightarrow$ #Coins$^N$ (Ouch!)

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

\[
\text{Min}\#\text{Coins}(N) = \begin{cases} 
1 + \text{Min}\#\text{Coins}(N - 1) \\
1 + \text{Min}\#\text{Coins}(N - 20) \\
1 + \text{Min}\#\text{Coins}(N - 50)
\end{cases}
\]

With some memory ($N$ intermediate computations), the minimum number of coins can be obtained after $N \times \#\text{Coins}$ 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.
Strategy #2: Brute force enumeration → \( \#\text{Coins}^N \) (Ouch!)

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

\[
\text{Min}\#\text{Coins}(N) = \text{Min} \begin{cases} 
1 + \text{Min}\#\text{Coins}(N - 1) \\
1 + \text{Min}\#\text{Coins}(N - 20) \\
1 + \text{Min}\#\text{Coins}(N - 50)
\end{cases}
\]

With some memory (\( N \) intermediate computations), the minimum number of coins can be obtained after \( N \times \#\text{Coins} \) 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.
Dynamic programming = General optimization technique.

**Prerequisite:** Optimal solution for problem $P$ can be derived from solutions to strict sub-problems of $P$.

**Bioinformatics:**

- Discrete solution space (alignments, structures...)
- Additively-inherited objective function (cost, log-odd score, energy...)

$\Rightarrow$ Efficient dynamic programming scheme

**Example:** Local Alignment (Smith/Waterman)

\[
\begin{align*}
W(i, 0) &= 0 \\
W(0, j) &= 0 \\
W(i, j) &= \max \left\{ W(i-1, j-1) + m_{i,j}, W(i-1, j) + p_i, W(i, j-1) + p_d \right\}
\end{align*}
\]
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:

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

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

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

⇒ Under these properties, one can enumerate solution space.
Outline

1. Introduction
   - Dynamic programming 101
   - Dynamic programming framework

2. Variations on RNA folding
   - Why RNA?
   - RNA folding
   - RNA Structure(s)
   - Some representations of RNA structure
   - Thermodynamics vs Kinetics

3. Free-energy minimization
   - Nussinov-style RNA folding
   - Turner energy model
   - MFold/Unafold
   - Performances and the comparative approach
   - Towards a 3D ab-initio prediction

4. Boltzmann ensemble
   - Nussinov: Minimisation ⇒ Counting
   - Computing the partition function
   - Statistical sampling
Fundamental *dogma* of molecular biology

DNA

\{A, C, G, T\}*

\[\begin{array}{c}
\text{ATGGTTAACCCATT} \\
\text{TACCAATGGGTA} \\
\end{array}\]
Fundamental dogma of molecular biology

DNA
\{A, C, G, T\}^*

RNA
\{A, C, G, U\}^*

THE CODE (genes)

THE MACHINE (enzymes)

Pol

ATGGTTACCCAT

TAACCAATGGGTAT
Fundamental *dogma* of molecular biology

**DNA**
\{A, C, G, T\}*

**RNAs**
\{A, C, G, U\}*

**Proteins**
\{Ala, Arg, ..., Val\}*

THE CODE (genes)
THE MACHINE (enzymes)

\[\text{Pol} \quad \begin{array}{c}
C \\
C \\
A \\
A \\
T \\
G \\
G \\
T \\
A
\end{array}\]
Fundamental *dogma* of molecular biology

**DNA**
{A, C, G, T}*

**RNAs**
{A, C, G, U}*
Fundamental *dogma* of molecular biology

DNA
\{A, C, G, T\}*

RNAs
\{A, C, G, U\}*

THE CODE (genes)

THE MACHINE (enzymes)

\[
\begin{align*}
\text{ATG} & \rightarrow \text{AUG} \\
\text{TTA} & \rightarrow \text{UAA} \\
\text{ACC} & \rightarrow \text{UAA} \\
\text{CAA} & \rightarrow \text{UAA} \\
\text{ATG} & \rightarrow \text{AUG} \\
\text{TGT} & \rightarrow \text{CGU} \\
\end{align*}
\]
Fundamental *dogma* of molecular biology

- **DNA**: \{A, C, G, T\}*
- **RNAs**: \{A, C, G, U\}*

![Diagram showing DNA and RNAs with nucleotide bases and enzymes](image-url)
Fundamental *dogma* of molecular biology

DNA
\{A, C, G, T\}*

RNAs
\{A, C, G, U\}*

THE CODE (genes)
THE MACHINE (enzymes)
Fundamental *dogma* of molecular biology

- **DNA**
  \{A, C, G, T\}*

- **RNAs**
  \{A, C, G, U\}*

- **Proteins**
  \{Ala, Arg, . . . , Val\}*
  20+ Amino acids

---

THE CODE (genes)
THE MACHINE (enzymes)
MEH. . .
Fundamental *dogma* of molecular biology

- **DNA**
  - {A, C, G, T}*

- **RNAs**
  - {A, C, G, U}*

- **Proteins**
  - {Ala, Arg, . . . , Val}*

  - 20+ Amino acids

- **Ribosome**

- **mRNA**
  - ATGGTTACCCCAT

- **tRNA**
  - AUGCUGUAACCCCAU

- **Amino acids**
  - Met
Fundamental *dogma* of molecular biology

**DNA**
\{A, C, G, T\}

**RNAs**
\{A, C, G, U\}

**Proteins**
\{Ala, Arg, ..., Val\}

20+ Amino acids

---

Amino acids

{Ala, Arg, ... , Val}*

Ribosome

ATGGTGTACCCCAT

ATGGAACCCAU

AUGGGUUAACCCAU

RTsense

Met Val
Fundamental *dogma* of molecular biology

- **DNA** \{A, C, G, T\}*
- **RNAs** \{A, C, G, U\}*
- **Proteins** \{Ala, Arg, \ldots, Val\}*

- RNA codon sequence: AUG GGU UAU ACC CCA AU
- Corresponding amino acid sequence: Met Val Thr

Diagram showing the translation of DNA into proteins by the ribosome.
Fundamental *dogma* of molecular biology

DNA

\{A, C, G, T\}*

RNAs

\{A, C, G, U\}*

Proteins

\{Ala, Arg, . . . , Val\} *

20+ Amino acids

\[ \text{Ribosome} \]

\[ \text{ATG GGT TTA CCA CAT} \]

\[ \text{AUG GGU UAA CCA CAU} \]

| Met | Val | Thr | His |
Fundamental *dogma* of molecular biology

- **DNA**
  \{A, C, G, T\}*

- **RNAs**
  \{A, C, G, U\}*

- **Proteins**
  \{Ala, Arg, ..., Val\}*

  20+ Amino acids

- **RNA Sequence:**
  \text{AUGGGUUAACCCCAAU}

- **DNA Sequence:**
  \text{ATGGTATTACCCCAT}

- **Protein Sequence:**
  Met Val Thr His Ile Leu His Asn
Fundamental *dogma* of molecular biology

THE CODE (genes)

DNA \{A, C, G, T\}^*

RNAs \{A, C, G, U\}^*

Proteins \{Ala, Arg, . . . , Val\}^*  \(20^+\) Amino acids

\[\begin{array}{cccccccccccc}
\text{ATGGTTTACCCAT} \\
\text{TAACCAATGGGTA} \\
\text{AUUGGUUACCCCAU} \\
\text{Met Val Thr His Ile Leu His Asn} \\
\end{array}\]
The fundamental dogma of molecular biology is illustrated in the diagram. It shows the relationship between the \textit{code} (DNA), \textit{machine} (enzymes), and \textit{products} (proteins).

- **THE CODE** (genes): DNA, represented by \{A, C, G, T\}.
- **THE MACHINE** (enzymes): Proteins, represented by \{Ala, Arg, . . . , Val\}.
- **THE PRODUCTS** (proteins): Amino acids, shown with their corresponding codons in RNA (\{A, C, G, U\}).

The diagram also shows the translation process: DNA sequences are transcribed into RNA, which is then translated into proteins. The specific sequence of DNA or RNA (e.g., \texttt{ATGGTATTACCCCAAT} for DNA and \texttt{AUGGUUACCCCAU} for RNA) corresponds to the start codon for translation, leading to the synthesis of a specific protein sequence (e.g., Met-Val-Thr-His-Ile-Leu-His-Asn).
Fundamental *dogma* of molecular biology

**THE CODE**
- Genes
- DNA
  \{A, C, G, T\}*

**RNA**
- RNAs
  \{A, C, G, U\}*

**THE MACHINE**
- Enzymes
- Proteins
  \{Ala, Arg, ..., Val\}*
  20+ Amino acids

**THE CODE** → **RNA** → **THE MACHINE**

- DNA sequence: ATGGTTATACCCCAT
- RNA sequence: AUGGUUACCCCAU
- Proteins: Met Val Thr His Ile Leu His Asn
Fundamental *dogma* of molecular biology

- DNA
- RNA
- Transcription
- Translation
- Proteins

RNA functions:
- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- . . .
Fundamental *dogma* of molecular biology

- DNA
- RNA
- Proteins

**RNA functions**
- Messenger
- Translation
- Regulation
- Enzyme
- Catalytic
- ...

- Carrier
- Transfer
- Participates
- Synthesis
- Maturation
- Regulation
- Transcription
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**.
Three\textsuperscript{1} levels of representation:

Primary structure

Secondary structure

Tertiary structure

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

\textsuperscript{1}Well, mostly...
Three\(^1\) levels of representation:

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) [LW01].

Pseudoknots (PKs)

Considering PKs may lead to better predictions, but:
- Some PK conformations are simply unfeasible;
- Folding \textit{in silico} with general pseudoknots is NP-complete [LP00];

Still, folding on restricted classes of conformations seems promising [CDR$^+$04].
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*

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

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*

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*

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

Non-crossing arc-annotated sequences*

Supporting intuitions
Different representations
Common combinatorial structure

*Additional steric constraints
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;
**Transcription**: RNA synthesized, supposedly without structure\(^2\)

![Diagram](image)

\(T = 0\)

**But** most mRNAs are degrade before 7h (Org.: Souris [SSN+09]).

\(^2\)Except for co-transcriptional folding...
**Transcription:** RNA synthesized, supposedly without structure\(^2\)

\[ T = 1\text{h} \]

**But** most mRNAs are degrade before 7h (Org.: Souris [SSN+09]).

\(^2\)Except for co-transcriptional folding...
Transcription: RNA synthesized, supposedly without structure\(^2\)

\[ T = 2h \]

But most mRNAs are degrade before 7h (Org.: Souris [SSN\(^+\)09]).

\(^2\)Except for co-transcriptional folding...
**Transcription:** RNA synthesized, supposedly without structure\(^2\)

\[ T = 5h \]

But most mRNAs are degrade before 7h (Org.: Souris [SSN\(^+\)09]).

\(^2\)Except for co-transcriptional folding...
Transcription: RNA synthesized, supposedly without structure$^2$

But most mRNAs are degrade before 7h (Org.: Souris [SSN$^+09$]).

$^2$Except for co-transcriptional folding...
**Transcription:** RNA synthesized, supposedly without structure\(^2\)

\[ T \rightarrow \infty \]

**But** most mRNAs are degrade before 7h (Org.: Souris [SSN\(^+\)09]).

\(^2\)Except for co-transcriptional folding...
**Transcription:** RNA synthesized, supposedly without structure\(^2\)

\[ T = 10h \]

**But** most mRNAs are degrade before 7h (Org.: Souris [SSN\(^+\)09]).

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 [MTSC09])

\(^2\)Except for co-transcriptional folding...
1 Introduction
- Dynamic programming 101
- Dynamic programming framework

2 Variations on RNA folding
- Why RNA?
- RNA folding
- RNA Structure(s)
- Some representations of RNA structure
- Thermodynamics vs Kinetics

3 Free-energy minimization
- Nussinov-style RNA folding
- Turner energy model
- MFold/Unafold
- Performances and the comparative approach
- Towards a 3D ab-initio prediction

4 Boltzmann ensemble
- Nussinov: Minimisation ⇒ Counting
- Computing the partition function
- Statistical sampling

Ab initio folding prediction =
Predict RNA structure from its sequence $\omega$ only.

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

\[ E_{\omega,S} = -\#Paires(S) \]

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

**Example:**

\[ UUUUCCCUAAAGG \]

Variant: Weight each pair with \(-\#\)Hydrogen bonds

\[ \Delta G(G\equiv C) = -3 \quad \Delta G(A=U) = -2 \quad \Delta G(G-U) = -1 \]
**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)

\[ E_{\omega,S} = -\#Paires(S) \]

Folding in NJ model \(\iff\) **Base-pair (weight) maximization**

**Example:**

```
UUUUC CCUA AAA GG
```

**Variant:** Weight each pair with \(-\#\)Hydrogen bonds

\[ \Delta G(G\equiv C) = -3 \quad \Delta G(A=U) = -2 \quad \Delta G(G-U) = -1 \]
\[ N_{i,t} = 0, \quad \forall t \in [i, i + \theta] \]

\[ N_{i,j} = \min \begin{cases} 
N_{i+1,j} & \text{i unpaired} \\
\min_{k=i+\theta+1} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j} & \text{i paired with } k
\end{cases} \]
Nussinov/Jacobson DP scheme

\[ i \quad j = i \quad i+1 \quad j + i \quad k \quad j \geq \theta \]

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

\[ N_{i,j} = \min \left\{ \begin{array}{ll}
N_{i+1,j} & \text{i unpaired} \\
\min_{k=i+\theta+1} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j} & \text{i paired with k}
\end{array} \right. \]

**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 MFE\(_{i,j}\) := Base-pairs of best struct. on \([i, j]\). Then first position \(i\) in MFE\(_{i,j}\) is either:
  - **Unpaired:** MFE\(_{i,j}\) = MFE\(_{i+1,j}\) \(\rightarrow\) free-energy = \(N_{i+1,j}\)
  - **Paired to** \(k\): MFE\(_{i,j}\) = \{(i, k)\} \cup MFE\(_{i+1,k-1}\) \cup 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

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

\[
N_{i,j} = \min \left\{ \begin{array}{ll}
N_{i+1,j} & \text{if } i \text{ unpaired} \\
\min_{k=i+\theta+1} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j} & \text{if } i \text{ paired with } k
\end{array} \right.
\]

**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 \(\text{MFE}_{i,j} := \text{Base-pairs of best struct. on } [i, j]\). Then first position \(i\) in \(\text{MFE}_{i,j} = \) is either:
  - **Unpaired:** \(\text{MFE}_{i,j} = \text{MFE}_{i+1,j}\) \(\rightarrow \text{free-energy} = N_{i+1,j}\)
  - **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))
    \(\rightarrow \text{free-energy} = \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j}\)
Nussinov/Jacobson DP scheme

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

**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 \(\text{MFE}_{i,j} := \text{Base-pairs of best struct. on } [i, j]\). Then first position \(i\) in \(\text{MFE}_{i,j} = \) is either:
  - **Unpaired:** \(\text{MFE}_{i,j} = \text{MFE}_{i+1,j} \rightarrow \text{free-energy} = N_{i+1,j}\)
  - **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 \((i, k)\))
    \(\rightarrow \text{free-energy} = \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j}\)
|   | 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 | 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 | 0 | 2 | 2 | 4 | 5 | 7 | 7 | 8 | 10 |   |
| A | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 8 | 8 |   |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 5 | 5 | 8 | 8 |   |
| U | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 6 | 7 |   |
| U | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 5 | 5 | 7 |   |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 5 | 5 |   |
| U | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 3 |   |
| U | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |   |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |
| G |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| A |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| C |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| G |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| A |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

\[ i \leq j \leq i + 1 \leq k \leq j \geq \theta \]
|   | 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 |   |
| 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 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 8 | 8 |     |   |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 5 | 5 | 8 | 8 |     |   |
| U | 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 | 3 | 5 | 5 |     |   |
| U | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 3 |     |   |
| U | 0 | 0 | 0 | 0 | 0 | 1 | 2 |     |   |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 0 |     |   |
| G | 0 | 0 | 0 | 0 | 0 | 0 | 0 |     |   |

\[ i \leq j \] and \( i \leq k \leq j \)
\[ i = i + 1 \]

\[ \geq \theta \]
\[(i, j) = (i, i+1) + \sum_{k=i}^{j} \geq \theta\]
\[
i_j = \begin{cases} 
0 & \text{if } i = j \\
0 & \text{if } j = i + 1 \\
\sum_{k=i}^{j-1} c_{k,j} & \text{otherwise}
\end{cases}
\geq \theta
\]
|   | 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 | 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 | 5 | 7 |   |   |   |   |   |
| C | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 5 | 5 |   |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 3 |   |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 0 | 1 | 2 |   |   |   |   |   |   |   |   |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |
| G | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |
| A | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |   |
| C | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |
| G | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |   |
| A | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |

\[ i \leq j \]

\[ i \leq k \leq j \]
<table>
<thead>
<tr>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>)</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>( i )</td>
<td>( j )</td>
<td>( i+j )</td>
<td>( j )</td>
<td>( i+k )</td>
<td>( j )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>( \geq \theta )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>G</td>
<td>G</td>
<td>A</td>
<td>U</td>
<td>A</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>A</td>
<td>G</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>A</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>=</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>i+1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>k</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( i \leq \theta )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\[
(i) = i_{i+1} + \sum_{k} \geq \theta
\]
<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ i \leq j \geq \theta \]
\[ i \leq j \leq i + 1 \]
\[ i \leq j = i \quad \text{and} \quad j = i + 1 \]
<table>
<thead>
<tr>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>. )</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( i = i + 1 \)

\( \geq \theta \)
\[
\begin{array}{cccccccccccccccc}
C & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 9 & 9 & 11 & 14 & 14 \\
G & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 7 & 9 & 11 & 11 & 11 \\
G & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 6 & 8 & 10 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 5 & 7 & 7 & 8 & 10 & 10 & 10 & 10 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 5 & 7 & 7 & 8 & 10 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 5 & 5 & 5 & 5 & 8 & 8 & 8 & 8 & 8 & 8 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 5 & 5 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 6 & 7 & 7 & 7 & 7 & 7 & 7 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 5 & 5 & 7 & 7 & 7 & 7 & 7 \\
C & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
U & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
i \leq j \leq k \]
<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>i</td>
<td>j = i</td>
<td>i+1</td>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>i</td>
<td>k</td>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ i \leq j \geq \theta \]
\[
\sum_{i \leq j} \max (0, M_{i+1 \sigma_{i+1}} + M_{i+2 \sigma_{i+2}} + \ldots + M_{j \sigma_{j}} + \theta)
\]
<table>
<thead>
<tr>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>(</td>
<td>(</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>)</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\[ i + 1 \geq \theta \]
<table>
<thead>
<tr>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>(</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>)</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>)</td>
<td>)</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>i</td>
<td>j = i</td>
<td>i+1</td>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>i</td>
<td>k</td>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ i + k \geq \theta \]
<p>|   | 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 | 0 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 6 | 8 | 10 | 10 | 10 |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 4 | 4 | 5 | 7 | 7 | 8 | 10 | 10 |
| U | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 4 | 5 | 7 | 7 | 8 | 10 |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 8 | 8 |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 5 | 5 | 8 | 8 |
| U | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 6 | 7 |
| U | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 5 | 7 |
| C | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 5 | 3 | 5 | 5 |
| U | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 3 |
| U | 0 | 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 | i | j = i | i+1 | j |
| G | i | k | j | \geq \theta |</p>
<table>
<thead>
<tr>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>(</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>)</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>G</td>
<td>A</td>
<td>U</td>
<td>A</td>
<td>C</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>A</td>
<td>G</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>G</td>
<td>G</td>
<td>A</td>
<td>U</td>
<td>A</td>
<td>C</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>A</td>
<td>G</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( (i, j) = (i+1, j) + \begin{array}{c}
\geq \theta
\end{array} \)
\begin{verbatim}
\begin{tabular}{cccccccccccccc}
  \hline
  & ( & ( & ( & . & . & . & ) & ) & ) & ) & ) & ) & ) & ) & & & \\
  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 & 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 & 10
  A & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 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 & 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 & 5 & 6 & 7 & & & &
  U & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 5 & 7 & & & & & &
  C & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 5 & 5 & & & & &
  U & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 3 & & & & & & &
  U & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & & & & & & & & &
  A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & &
  G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & &
  A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & &
  C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & &
  G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & &
  A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & &

\end{tabular}
\end{verbatim}

\begin{equation}
i \leq j \quad \text{for} \quad i \leq k \quad \text{and} \quad j \geq \theta
\end{equation}

![Diagram](attachment:image.png)
\[
\begin{array}{cccccccccccccccc}
\[
i = i + 1, j + 1, ik + j \geq \theta
\]
\[
\begin{array}{cccccccccccccccc}
( & ( & ( & . & . & . & ) & ) & ( & . & . & . & . & . & ) & ) & ) & ) & \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 9 & 9 & 11 & 14 & 14 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 7 & 9 & 11 & 11 & 11 \\
G & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 6 & 8 & 10 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 5 & 7 & 7 & 8 & 8 & 10 & 10 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 5 & 7 & 7 & 8 & 10 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 5 & 5 & 5 & 5 & 8 & 8 & 8 & 8 & 8 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 5 & 5 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 6 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 5 & 5 & 7 & 7 & 7 & 7 & 7 & 7 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 5 & 5 & 5 & 5 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
i = i^{+1} \geq \theta
\]
\[
\begin{array}{cccccccccccccccc}
\hline
( & ( & ( & . & . & . & ) & ) & ) & ) & ) & ) & ) & ) & ) & ) & ) & . \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 6 & 9 & 9 & 11 & 14 & 14 \\
G & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 7 & 9 & 11 & 11 & 11 \\
G & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 6 & 8 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 5 & 7 & 7 & 8 & 10 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 5 & 7 & 7 & 8 & 10 \\
A & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 5 & 5 & 5 & 8 & 8 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 5 & 5 & 8 & 8 \\
U & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 6 & 7 \\
U & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 5 & 7 \\
C & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 \\
U & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 3 \\
U & 0 & 0 & 0 & 0 & 1 & 2 \\
A & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 & 0 \\
C & i = i & i + 1 & j & j & i & i & k & j \\
G & \geq \theta \\
A & 0 \\
\end{array}
\]
<table>
<thead>
<tr>
<th>C</th>
<th>G</th>
<th>G</th>
<th>A</th>
<th>U</th>
<th>A</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>C</th>
<th>U</th>
<th>U</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>(</td>
<td>(</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
<td>)</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>i</td>
<td>j</td>
<td>=</td>
<td>i</td>
<td>i+1</td>
<td>j</td>
<td>+</td>
<td>i</td>
<td>k</td>
<td>j</td>
<td>≥ θ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Nussinov/Jacobson
\[
\begin{array}{cccccccccccccccc}
( & ( & ( & . & . & . & ) & ) & ) & ) & ) & ) & ) & ) & ) & ) & ) & . \\
\hline
C & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 6 & 9 & 9 & 11 & 14 & 14 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 7 & 9 & 11 & 11 & 11 \\
G & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 6 & 8 & 10 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 5 & 7 & 7 & 8 & 10 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 5 & 7 & 7 & 8 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 5 & 5 & 5 & 8 & 8 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 5 & 5 & 8 & 8 \\
U & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 6 & 7 \\
U & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 5 & 7 \\
C & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 5 & 5 \\
U & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 3 \\
U & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\
A & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 \\
C & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 \\
\end{array}
\]
Based on unambiguous decomposition of $2^\text{ary}$ structure into loops:

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

Free-energy $\Delta G$ 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 $\Delta G$ of a loop depend on bases, asymmetry, dangles …

Experimentally determined
+ Interpolated for larger loops.

Improved results by taking stacking into account.
Based on unambiguous decomposition of 2\textsuperscript{ary} structure into loops:

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

Free-energy $\Delta G$ 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\textsuperscript{ary} structure into loops:

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

Free-energy $\Delta G$ of a loop depend on bases, asymmetry, dangles ... 

Experimentally determined
+ Interpolated for larger loops.

Improved results by taking stacking into account.
Based on **unambiguous** decomposition of 2\textsuperscript{ary} structure into loops:

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

Free-energy $\Delta G$ of a loop depend on bases, asymmetry, dangles . . .

Experimentally determined
+ Interpolated for larger loops.

Improved results by taking stacking into account.
Based on unambiguous decomposition of 2\textsuperscript{ary} structure into loops:

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

Free-energy $\Delta G$ of a loop depend on bases, asymmetry, dangles \ldots

Experimentally determined
+ Interpolated for larger loops.

Improved results by taking stacking into account.
MFE DP equations

Stem loop = \{ Terminal loops, Stackings, Bulges/Internal loops, Multi loops (Sequence ≥2 helices) \}
MFE DP equations

\[
\begin{align*}
\text{Stem loop} & = \\
\text{Terminal loops} & \\
\text{Stackings} & \\
\text{Bulges/Internal loops} & \\
\text{Multi loops} & \quad \text{(Sequence } \geq 2 \text{ helices)} \\
\text{Helix sequence} & = \\
\text{Seq. + Helix} &
\end{align*}
\]
MFE DP equations

- Terminal loops
  - Stackings
  - Bulges/Internal loops
  - Multi loops (Sequence $\geq 2$ helices)

= Seq. + Helix
  - Helix
MFE DP equations

\[
\begin{align*}
\text{Stem loop} &= \\
&= \begin{cases} \\
\text{Terminal loops} \\
\text{Stackings} \\
\text{Bulges/Internal loops} \\
\text{Multi loops} \\
(\text{Sequence } \geq 2 \text{ helices})
\end{cases} \\
&= \\
&= \begin{cases} \\
\text{Seq. + Helix} \\
\text{Helix} \\
\text{Unpaired base} \\
\text{Stem Sloop}
\end{cases}
\end{align*}
\]
- $E_H(i, j)$: 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**

\[
M'_{i,j} = \min \left\{ \begin{array}{l}
E_H(i, j) \\
E_S(i, j) + M'_{i+1,j-1} \\
\text{Min}_{i', j'}(E_{BI}(i, i', j, j') + M'_{i', j'}) \\
a + \text{Min}_k(M_{i+1,k-1} + M'_{1,k-1})
\end{array} \right\}
\]

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

\[
M^1_{i,j} = \text{Min}_k \left\{ b + M^1_{i,j-1}, c + M'_{i,j} \right\}
\]
Backtracking to reconstruct MFE structure:

\[
\begin{align*}
\mathcal{M}'_{i,j} &= \min \left\{ 
E_H(i,j) \\
E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\
\min_{i',j'} (E_{BI}(i,i',j',j) + \mathcal{M}'_{i',j'}) \\
a + \min_k (\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1})
\right\}
\end{align*}
\]

\[
\begin{align*}
\mathcal{M}_{i,j} &= \min_k \left\{ \min (\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\} \\
\mathcal{M}^1_{i,j} &= \min_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}
\end{align*}
\]

**Complexity:**

For each \(\min\), \(O(n)\) potential contributors

\(\Rightarrow\) **Worst-case** complexity in \(O(n^2)\) for **naive backtrack**.

Keep best contributor for each \(\min\) \(\Rightarrow\) **Backtracking in** \(O(n)\)

\(\Rightarrow\) **UnaFold** [MZ08]/**RNAFold** [HFS+94] compute the MFE for the Turner model
in **overall**\(^3\) time/space complexities in \(O(n^3)/O(n^2)\)

\(^3\)Using a trick/restriction for internal loops...
Backtracking to reconstruct MFE structure:

\[ m'_{i,j} = \min \left\{ m_{i,k-1}, b(k-1) \right\} \]

\[ m_{i,j} = \min_k \left\{ m_{i+1,k-1} + m^1_{k,j} \right\} \]

\[ m^1_{i,j} = \min_k \left\{ b + m^1_{i,j-1}, c + m'_{i,j} \right\} \]

Complexity:

For each \( \min \), \( O(n) \) potential contributors

\( \Rightarrow \text{Worst-case complexity in } O(n^2) \) for \textbf{naive backtrack}.

Keep best contributor for each \( \min \) \( \Rightarrow \textbf{Backtracking in } O(n) \)

\( \Rightarrow \textbf{UnaFold} [MZ08]/\textbf{RNAFold} [HFS+94] \) compute the MFE for the Turner model

in overall\(^3 \) time/space complexities in \( O(n^3)/O(n^2) \)

\(^3\)Using a trick/restriction for internal loops...
Backtracking to reconstruct MFE structure:

\[
\begin{align*}
\mathcal{M}'_{i,j} &= \text{Min} \left\{ \mathcal{M}_{i,k-1} + b(k-1), \mathcal{M}'_{i,j} \right\} \\
\mathcal{M}_{i,j} &= \text{Min}_k \left\{ \min (\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}'_{i,j} \right\} \\
\mathcal{M}^1_{i,j} &= \text{Min}_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}
\end{align*}
\]

**Complexity:**

For each \( \text{min} \), \( \mathcal{O}(n) \) 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** [MZ08]/**RNAFold** [HFS+94] 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:

\[ M'_{i,j} = \text{Min} \left\{ E_H(i,j), E_S(i,j) + M'_{i+1,j-1}, \min_{i',j'} (E_{BI}(i,i',j,j) + M'_{i',j'}) \right\} \]

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

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

**Complexity:**

For each \( \text{min} \), \( \mathcal{O}(n) \) 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 [MZ08]/RNAFold [HFS+94] 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:

\[ \mathcal{M}'_{i,j} = \min \left\{ \begin{array}{l}
E_H(i,j) \\
E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\
\min_{i',j'} (E_{BI}(i,i',j,j) + \mathcal{M}'_{i',j'}) \\
a + \min_k \left( \mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1} \right)
\end{array} \right\} \]

\[ \mathcal{M}_{i,j} \leftarrow \min_k \left\{ \min (\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\} \]

\[ \mathcal{M}^1_{i,j} \leftarrow \min_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\} \]

**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** [MZ08]/**RNAFold** [HFS⁺94] 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:

\[
\begin{align*}
\mathcal{M}'_{i,j} &= \min \left\{ E_H(i,j), \ E_S(i,j) + \mathcal{M}'_{i+1,j-1}, \ \min_{i',j'} (E_B(i,i',j',j) + \mathcal{M}'_{i',j'}) \right. \\
&\quad \left. + a + \min_k (\mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1}) \right\} \\
\mathcal{M}_{i,j} &= \min_k \left\{ \min (\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\} \\
\mathcal{M}^1_{i,j} &= \min_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}
\end{align*}
\]

**Complexity:**

For each \(\min\), \(O(n)\) potential contributors

\(\Rightarrow\) **Worst-case** complexity in \(O(n^2)\) for naive backtrack.

Keep best contributor for each \(\min\) \(\Rightarrow\) **Backtracking in** \(O(n)\)

\(\Rightarrow\) UnaFold [MZ08]/RNAFold [HFS+94] compute the MFE for the Turner model

in overall\(^3\) time/space complexities in \(O(n^3)/O(n^2)\)

\(^3\)Using a trick/restriction for internal loops...
Backtracking to reconstruct MFE structure:

\[
\mathcal{M}'_{i,j} = \min \left\{ \begin{array}{l}
E_H(i,j) \\
E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\
\min_{i',j'} \left( E_{BI}(i,i',j',j) + \mathcal{M}'_{i',j'} \right) \\
a + \min_k \left( \mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1} \right) \end{array} \right. 
\]

\[
\mathcal{M}_{i,j} = \min_k \left\{ \min (\mathcal{M}_{i,k-1}, b(k-1)) + \mathcal{M}^1_{k,j} \right\} 
\]

\[
\mathcal{M}^1_{i,j} = \min_k \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\} 
\]

**Complexity:**

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

⇒ UnaFold [MZ08]/RNAFold [HFS+94] 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
**Two main approaches**

<table>
<thead>
<tr>
<th>Definition (Ab initio folding)</th>
<th>Definition (Comparative approach)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting from sequence, find conformation that minimizes free-energy.</td>
<td>Starting from homologous sequences, postulate common structure and find best possible tradeoff between folding &amp; alignment.</td>
</tr>
</tbody>
</table>

**Advantages:**
- Mechanical nature allows the 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

**Advantages:**
- Better performances
- (Limited) cooperativity
- Self-improving

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

**sequence**
- UUAGGCGGCCACAGC
- GGUGGGGUUGCCUCC
- CGUACCCAUCCCGAA
- CACGGAAGAUAAGCC
- CACCAGCGUUCCGGG
- GAGUACUGGAGUGCG
- CGAGGCUUGGGAAA
- CCCGGGUUCGCCCGA
- CC

**taille**
- <700 : 70-73% [MSZT99, DCCG04]
- 16s, 23s : ~50%
- MCC : ~0.5 [GG04]

**thermodynamics**

**sens. spéc. mcc.**
- ~75%  ~75%  0.8
- Comparative [GG04]

Reminder:

\[ MCC = \frac{t^+ t^- - f^+ f^-}{\sqrt{(t^+ + f^+)(t^- + f^-)(t^- + f^+)(t^- + f^-)}} \]
Performances

Sequence

UUAGGCGGCCACAGCGGUUGGGGUUGCCUCGCCGUACCCAUCCCGAACACGGAAGAUAAGCCCACCAGCGUUCCGGGGAGUACUGGAGUGCGCGAGCCUCUGGGAAACCCGUUCGCCGCCACCC

Taille Sens.
<700 70-73%
[MSZT99, DCCG04]
16s, 23s ~50%
MCC ~ 0.5 [GG04]

Thermodynamics

Sens. Spé. MCC.
~75% ~75% 0.8
Comparative [GG04]

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

Sequence

Taille Sens.
<700 70-73%
[MSZT99, DCCG04]
16s,23s ~50%
MCC~ 0.5 [GG04]

Thermodynamics

Sens. Spé. MCC.
~75% ~75% 0.8
Comparative [GG04]

Reminder: \( MCC = \frac{t^+ t^- f^+ f^-}{\sqrt{(t^+ + f^+)(t^- + f^-)(t^+ + f^-)(t^- + f^-)}} \)
Goal: From sequence to all-atom/coarse grain 3D models!!!

- Comparative models + Molecular dynamics: RNA2D3D [SYKB07]
- Pipeline MC-Fold/MC-sym [PM08]
**Goal:** From sequence to all-atom/coarse grain 3D models!!!

- Comparative models + Molecular dynamics: RNA2D3D [SYKB07]
- Pipeline MC-Fold/MC-sym [PM08]
**Goal:** From sequence to all-atom/coarse grain 3D models!!!

- Comparative models + Molecular dynamics: RNA2D3D [SYKB07]
- Pipeline MC-Fold/MC-sym [PM08]
1. **Introduction**
   - Dynamic programming 101
   - Dynamic programming framework

2. **Variations on RNA folding**
   - Why RNA?
   - RNA folding
   - RNA Structure(s)
   - Some representations of RNA structure
   - Thermodynamics vs Kinetics

3. **Free-energy minimization**
   - Nussinov-style RNA folding
   - Turner energy model
   - MFold/Unafold
   - Performances and the comparative approach
   - Towards a 3D ab-initio prediction

4. **Boltzmann ensemble**
   - Nussinov: Minimisation $\Rightarrow$ Counting
   - Computing the partition function
   - Statistical sampling
RNA *breathes* ⇒ There is no more than a single conformation.

**New paradigm**

The conformations of an RNA **coexist** in the **Boltzmann distribution**.

![Image of RNA structures and MFE]

**Consequence:** The MFE probability can be arbitrarily small.

⇒ To understand how RNA acts, one must account for the set of alternative structures.

In particular, structurally close structures may **ally**, and become the most realistic candidate in the search for a functional conformation.
RNA *breathes* ⇒ There is no more than a single conformation.

**New paradigm**

The conformations of an RNA **coexist** in the **Boltzmann distribution**.

**Consequence:** The MFE probability can be arbitrarily small.  
⇒ To understand how RNA acts, one must account for the set of alternative structures.

In particular, structurally close structures may *ally*, and become the most realistic candidate in the search for a functional conformation.
For each structure $S$ compatible with an RNA $\omega$, the Boltzmann distribution associates a **Boltzmann factor** $B_{S,\omega} = e^{\frac{-E_{S,\omega}}{RT}}$, where:

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

To obtain a distribution, one simply renormalizes by the **partition function**

$$Z_{\omega} = \sum_{S \in S_{\omega}} e^{\frac{-E_{S,\omega}}{RT}}$$

where $S_{\omega}$ is the set of conformations that are compatibles with $\omega$.

The **Boltzmann probability** of a structure $S$ is simply given by

$$P_{S,\omega} = \frac{e^{\frac{-E_{S,\omega}}{RT}}}{Z_{\omega}}.$$
Nussinov/Jacobson DP scheme

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

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

**Ambiguity?** Consider \( i \): Either **unpaired**, or **paired** to \( k \).
Sets of structures generated in these two cases are clearly disjoint.
(also holds for various values of \( k \)) \( \Rightarrow \) **Unambiguous** decomposition

**Completeness?** True, since scheme explores every possible outcome for \( i \).
+ Induction on interval length \( \Rightarrow \) **Complete** decomposition
Recurrence for **minimal free-energy** of a fold:

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

\[
N_{i,j} = \min \left\{ \begin{array}{ll}
N_{i+1,j} & (i \text{ unpaired}) \\
\min_j \sum_{k=i+\theta+1}^{i+\theta} E_{i,k} + N_{i+1,k-1} + N_{k+1,j} & (i \text{ comp. with } k)
\end{array} \right.
\]

Recurrence for **counting compatible structures**:

\[
C_{i,t} = 1, \quad \forall t \in [i, i + \theta]
\]

\[
C_{i,j} = \sum \left\{ \begin{array}{ll}
C_{i+1,j} & (i \text{ unpaired}) \\
\sum_j \sum_{k=i+\theta+1}^{k+1} 1 \times C_{i+1,k-1} \times C_{k+1,j} & (i \text{ comp. with } k)
\end{array} \right.
\]

Decomposition matters, and the rest (MFE, count. . . ) follows!
Partition function = Weighted count over compatible structures

\[ Z_{i,t} = 1, \quad \forall t \in [i, i + \theta] \]

\[ Z_{i,j} = \sum \left\{ \sum_{k=i+\theta+1}^{j} 1 \times Z_{i+1,k-1} \times Z_{k+1,j} \right\} \]
Partition function = \textbf{Weighted count} over compatible structures

\[ Z_{i,t} = 1, \quad \forall t \in [i, i + \theta] \]

\[ Z_{i,j} = \sum \left\{ \sum_{k=i+\theta+1}^{j} \frac{e^{-E_{bp}(i,k)}}{RT} \times Z_{i+1,k-1} \times Z_{k+1,j} \right\} \]
Partition function = \textbf{Weighted count} over compatible structures

\[
\begin{align*}
\mathcal{M}'_{i,j} &= \text{Min} \left\{ \begin{array}{l}
E_H(i,j) \\
E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\
\text{Min}(E_B(i,i',j,j') + \mathcal{M}'_{i',j'}) \\
a + \text{Min} \left( \mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1} \right)
\end{array} \right. \\
\mathcal{M}_{i,j} &= \text{Min} \left\{ \text{Min} \left( \mathcal{M}_{i,k-1}, b(k - 1) \right) + \mathcal{M}^1_{k,j} \right\} \\
\mathcal{M}^1_{i,j} &= \text{Min} \left\{ b + \mathcal{M}^1_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}
\end{align*}
\]
Partition function = Weighted count over compatible structures

\[ \mathcal{M}_{i,j} = \min \left\{ \min \left( \mathcal{M}_{i-1,k} e^{-b(k-1)/RT} + \mathcal{M}^1_{k,j} \right) + \mathcal{M}^1_{i-1,j} e^{-c/RT} + \mathcal{M}'_{i,j} \right\} \]

\[ \mathcal{M}'_{i,j} = \min \left\{ e^{-E_H(i,j)/RT} + \mathcal{M}'_{i+1,j-1} + \min \left( \mathcal{M}_{i+1,k-1} + \mathcal{M}^1_{k,j-1} \right) \right\} \]
Partition function \( = \text{Weighted count} \) over compatible structures

\[
\mathcal{M}_i^1 = \text{Min} \left\{ e^{-\frac{b}{R T}} \mathcal{M}_i^1_{i-1, j} , e^{-c} \mathcal{M}_i^1 \right\}
\]

\[
\mathcal{M}_i,j = \text{Min} \left\{ \text{Min} \left( \mathcal{M}_i,k-1, e^{-\frac{b}{R T}} \right) \mathcal{M}_i^1_{k-1, j} \right\}
\]

\[
\mathcal{M}'_{i,j} = \text{Min} \left\{ \text{Min} \left( \mathcal{M}_{i+1,j-1}, e^{-\frac{E_H(i,j)}{R T}} \right) \mathcal{M}'_{i+1,j-1} \right\}
\]

\[
\text{Min} \left( e^{-\frac{E_S(i,j)}{R T}} \mathcal{M}'_{i,j}, e^{-\frac{E_{BI}(i',j',j)}{R T}} \mathcal{M}'_{i',j'} \right)
\]

\[
\text{Min} \left( \mathcal{M}_{i+1,k-1} \mathcal{M}_1^{1,k,j-1} \right)
\]
Partition function

\[ \text{Partition function} = \text{Weighted count over compatible structures} \]

\[
\begin{align*}
\mathcal{Z}'(i,j) &= \sum \left\{ \mathcal{Z}(i,k-1) e^{-\frac{b(k-1)}{RT}} + \mathcal{Z}'(i,j) \right\} \\
\mathcal{Z}(i,j) &= \left\{ \mathcal{Z}(i,k-1) e^{-\frac{b(k-1)}{RT}} + \mathcal{Z}'(i,j) \right\} \\
\mathcal{Z}'(i,j) &= e^{-\frac{c}{RT}} \mathcal{Z}'(i,j-1) + e^{-\frac{b}{RT}} \mathcal{Z}'(i,j)
\end{align*}
\]
Partition function $= \text{Weighted count}$ over compatible structures

\[
Z_{i,t} = 1, \quad \forall t \in [i, i + \theta]
\]

\[
Z_{i,j} = \sum \left\{ \sum_{k=i+\theta+1}^{j} e^{-E_{bp}(i,k)/RT} \times Z_{i+1,k-1} \times Z_{k+1,j} \right\}
\]

Validity of a partition function computation:

- Completeness/Unambiguity of decomposition scheme
- Correctness of Boltzmann factor

Weight induced by backtrack $= \text{Product of derivations weights}$

\[
e^{-E/RT} \rightarrow \text{Weight products} \Leftrightarrow \text{Summing energy terms}
\]

\[
e^{-E_{bp}(i,k)/RT} \times Z_{i+1,k-1} \times Z_{k+1,j} = \sum_x e^{-E(x)/RT} \cdot \sum_y e^{-E(y)/RT}
\]

\[
= \sum_{x,y} e^{-a/RT} \cdot e^{-E(x)/RT} \cdot e^{-E(y)/RT}
\]

\[
= \sum_{x,y} e^{-(E_{bp}(i,k)+E(x)+E(y))/RT}
\]
Partition function: 

\[ Z_{i,t} = 1, \quad \forall t \in [i, i + \theta] \]

\[ Z_{i,j} = \sum \left\{ \begin{array}{l}
Z_{i+1,j} \\
\sum_{k=i+\theta+1}^{j} e^{-\frac{E_{bp}(i,k)}{RT}} \times Z_{i+1,k-1} \times Z_{k+1,j}
\end{array} \right. \]

Validity of a partition function computation:

- Completeness/Unambiguity of decomposition scheme
- Correctness of Boltzmann factor

Weight induced by backtrack = Product of derivations weights

\[ e^{-E/RT} \rightarrow \text{Weight products} \Leftrightarrow \text{Summing energy terms} \]

\[ e^{-E_{bp}(i,k)/RT} \times Z_{i+1,k-1} \times Z_{k+1,j} = \sum_{x} e^{-E(x)/RT} \cdot \sum_{y} e^{-E(y)/RT} \]

\[ = \sum_{x,y} e^{-a/RT} \cdot e^{-E(x)/RT} \cdot e^{-E(y)/RT} \]

\[ = \sum_{x,y} e^{-(E_{bp}(i,k)+E(x)+E(y))/RT} \]
MFE (⇔ Max probability) may be **heavily dominated** by a set \( \mathcal{B} \) of **structurally similar** suboptimal structures.

⇒ Functional conformation probably closer to \( \mathcal{B} \) than to MFE.

**Proof-of-concept:** [DCL05]

- Sample structures within Boltzmann probability
- Cluster structures
  - Build and return consensus structure of the heaviest cluster

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

**Problem**

How to sample from the Boltzmann ensemble?
Stochastic backtrack (adapted from SFold)

Goal [DL03]: From sequence $\omega$, draw $S$ with prob. $e^{-Es/RT}/Z$

Principle: Choose derivation with prob. prop. to its contribution to part. fun.

Precomputation: Compute part. fun. versions of matrices ($Z$, $Z'$, $Z^1$).

Stochastic backtrack:

1. Draw uniform random number $r \in [0, Z'(i, j))$
2. Subtract from $r$ the contributions of $Z'(i, j)$ until $r < 0$
3. Recurse over associated regions/matrices

$$Z'(i, j) \xrightarrow{???} e^{-\frac{E_H(i, j)}{RT}} + e^{-\frac{E_S(i, j)}{RT}} Z'(i + 1, j - 1)$$

$$\sum \left(e^{-\frac{E_B(i', i', j', j')}{RT}} Z'(i', j') \right)$$

$$\sum (Z(i + 1, k - 1) Z^1(k, j - 1))$$
**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_s/RT} / Z$

**Principle:** Choose derivation with prob. prop. to its contribution to part. fun.

**Precomputation:** Compute part. fun. versions of matrices ($Z, Z', Z^1$).

**Stochastic backtrack:**

1. Draw uniform random number $r \in [0, Z'(i,j))$
2. Subtract from $r$ the contributions of $Z'(i,j)$ until $r < 0$
3. Recurse over associated regions/matrices

\[
Z'(i,j) = \sum \left\{ \begin{array}{c}
\sum \left( e^{-E_{BI}(i',i',j',j) / RT} Z'(i',j') \right) \\
\sum (Z(i+1,k-1) Z^1(k,j-1))
\end{array} \right. 
\]

A

B

C
**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_s/RT} / Z$

**Principle:** Choose derivation with prob. prop. to its contribution to part. fun.

**Precomputation:** Compute part. fun. versions of matrices ($Z$, $Z'$, $Z^1$).

**Stochastic backtrack:**

1. Draw uniform random number $r \in [0, Z'(i, j))$
2. Subtract from $r$ the contributions of $Z'(i, j)$ until $r < 0$
3. Recurse over associated regions/matrices

\[
Z'(i, j) = \sum \left\{ \begin{array}{l}
e^{-E_H(i,j)} \frac{1}{RT} + e^{-E_S(i,j)} \frac{1}{RT} Z'(i + 1, j - 1) \\
\sum \left( e^{-E_B(i,i',j',j)} \frac{1}{RT} Z'(i', j') \right) \\
e^{-E_B(i',i',j',j')} \frac{1}{RT} \sum (Z(i + 1, k - 1) Z^1(k, j - 1)) \end{array} \right. 
\]

\[\begin{array}{c|c|c|c|c|c|c}
A_1 & A_2 & B_i & B_{i+1} & \ldots & B_{j-1} & B_j \\
A & B_i & C_i & C_{i+1} & \ldots & C_{j-1} & C_j
\end{array}\]
Stochastic backtrack (adapted from SFold)

**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_s/RT}/Z$

**Principle:** Choose derivation with prob. prop. to its contribution to part. fun.

**Precomputation:** Compute part. fun. versions of matrices ($Z, Z', Z^1$).

**Stochastic backtrack:**

1. Draw uniform random number $r \in [0, Z'(i, j))$
2. Subtract from $r$ the contributions of $Z'(i, j)$ until $r < 0$
3. Recurse over associated regions/matrices

$$Z'(i, j) = \sum \left\{ \begin{array}{l} e^{-E_H(i,j)/RT} + e^{-E_S(i,j)/RT} Z'(i + 1, j - 1) \\ \sum \left( e^{-E_B(i',i'',j',j)/RT} Z'(i', j') \right) \\ e^{-(a)/RT} \sum_r (Z(i + 1, k - 1) Z^1(k, j - 1)) \end{array} \right\}$$
Goal [DL03]: From sequence $\omega$, draw $S$ with prob. $e^{-E_S/RT}/Z$

Principle: Choose derivation with prob. prop. to its contribution to part. fun.

Precomputation: Compute part. fun. versions of matrices ($Z, Z', Z^1$).

Stochastic backtrack:

1. Draw uniform random number $r \in [0, Z'(i, j))$
2. Subtract from $r$ the contributions of $Z'(i, j)$ until $r < 0$
3. Recurse over associated regions/matrices

$$Z'(i, j) = \sum \left\{ \begin{array}{l} e^{-\frac{E_H(i,j)}{RT}} + e^{-\frac{E_S(i,j)}{RT}} Z'(i + 1, j - 1) \\ \sum \left( e^{-\frac{E_B(i', i', j', j')}{RT}} Z'(i', j') \right) \\ e^{-\frac{(a)}{RT}} \sum (Z(i + 1, k - 1) Z^1(k, j - 1)) \end{array} \right.$$

A

B

C

$A_1 | A_2 | B_i | B_{i+1} | \ldots | B_{j-1} | B_j | C_i | C_{i+1} | \ldots | C_{j-1} | C_j$
**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_s/RT}/Z$

**Principle:** Choose derivation with prob. prop. to its contribution to part. fun.

**Precomputation:** Compute part. fun. versions of matrices ($Z$, $Z'$, $Z^1$).

**Stochastic backtrack:**

1. Draw uniform random number $r \in [0, Z'(i,j))$
2. Subtract from $r$ the contributions of $Z'(i,j)$ until $r < 0$
3. Recurse over associated regions/matrices

\[
Z'(i,j) = \sum \begin{cases} 
  e^{-E_H(i,j)}/RT + e^{-E_S(i,j)}/RT \cdot Z'(i+1,j-1) \\
  \sum \left( e^{-E_B(i,i',j,j)}/RT \cdot Z'(i',j') \right) \\
  e^{-(a)}/RT \cdot \sum (Z(i+1,k-1) \cdot Z^1(k,j-1)) \end{cases}
\]
**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_S/RT}/Z$

**Principle:** Choose derivation with prob. prop. to its contribution to part. fun.

**Precomputation:** Compute part. fun. versions of matrices $(Z, Z', Z^1)$.

**Stochastic backtrack:**

1. Draw uniform random number $r \in [0, Z'(i, j))$
2. Subtract from $r$ the contributions of $Z'(i, j)$ until $r < 0$
3. Recurse over associated regions/matrices

\[
Z'(i, j) = \sum \begin{cases} 
    e \frac{-E_H(i,j)}{RT} + e \frac{-E_S(i,j)}{RT} Z'(i + 1, j - 1) & A \\
    e \frac{-E_B(i,i',j,j)}{RT} Z'(i', j') & B \\
    e^{-(a)} \sum (Z(i + 1, k - 1) Z^1(k, j - 1)) & C \\
\end{cases}
\]
Stochastic backtrack (adapted from SFold)

**Goal [DL03]:** From sequence \( \omega \), draw \( S \) with prob. \( e^{-E_S/RT} / Z \)

**Principle:** Choose derivation with prob. prop. to its contribution to part. fun.

**Precomputation:** Compute part. fun. versions of matrices \( (Z, Z', Z^1) \).

**Stochastic backtrack:**

1. Draw uniform random number \( r \in [0, Z'(i,j)] \)
2. Subtract from \( r \) the contributions of \( Z'(i,j) \) until \( r < 0 \)
3. Recurse over associated regions/matrices

\[
Z'(i,j) = \sum \left\{ \begin{array}{l}
e^{-E_H(i,j) / RT} + e^{-E_S(i,j) / RT} Z'(i + 1, j - 1) \\
\sum \left( e^{-E_B(i,i',j',j)} / RT \right) Z'(i', j') \\
e^{- (a) RT} \sum (Z(i + 1, k - 1) Z^1(k, j - 1)) \end{array} \right. 
\]

**Correctness:** Each \( S \in S_\omega \) uniquely generated (DP scheme unambiguity)

Therefore the probability of generated \( S \) is

\[
p_S = \frac{B(E_1) \cdot B(E_2) \cdot B(E_3) \ldots B(\{S\})}{B(S_w) \cdot B(E_1) \cdot B(E_2) \ldots B(E_m)}
\]
Stochastic backtrack (adapted from SFold)

**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_S/RT}/Z$

**Principle:** Choose derivation with prob. prop. to its contribution to part. fun.

**Precomputation:** Compute part. fun. versions of matrices ($Z, Z', Z^1$).

**Stochastic backtrack:**

1. Draw uniform random number $r \in [0, Z'(i, j))$
2. Subtract from $r$ the contributions of $Z'(i, j)$ until $r < 0$
3. Recurse over associated regions/matrices

$$Z'(i, j) = \sum \left\{ \begin{array}{l}
  e^{-E_H(i,j)}_R + e^{-E_S(i,j)}_R Z'(i + 1, j - 1) \\
  \sum \left( e^{-E_B(i,i',j,j)}_R Z'(i', j') \right)
\end{array} \right. \quad \text{(A)}$$

$$e^{-R(a)} \sum (Z(i + 1, k - 1) Z^1(k, j - 1)) \quad \text{(B)}$$

**Correctness:** Each $S \in S_\omega$ uniquely generated (DP scheme unambiguity)

Therefore the probability of generated $S$ is

$$p_S = \frac{1}{B(S_w)} \cdot \frac{1}{1} \cdot \frac{1}{1} \cdots \frac{B(\{S\})}{1}$$
Stochastic backtrack (adapted from SFold)

**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_s/RT}/Z$

**Principle:** Choose derivation with prob. prop. to its contribution to part. fun.

**Precomputation:** Compute part. fun. versions of matrices ($Z$, $Z'$, $Z^1$).

**Stochastic backtrack:**
1. Draw uniform random number $r \in [0, Z'(i,j))$
2. Subtract from $r$ the contributions of $Z'(i,j)$ until $r < 0$
3. Recurse over associated regions/matrices

$$Z'(i,j) = \sum \left\{ \begin{array}{l} e^{-E_H(i,j)} \frac{R}{T} + e^{-E_S(i,j)} \frac{R}{T} Z'(i+1,j-1) \quad \text{(A)} \\ \sum \left( e^{-E_B(i,i',j',j)} \frac{R}{T} Z'(i',j') \right) \quad \text{(B)} \\ e^{-\frac{(a)}{R}} \sum \left( Z(i+1,k-1) Z^1(k,j-1) \right) \quad \text{(C)} \end{array} \right.$$  

**Correctness:** Each $S \in S_\omega$ uniquely generated (DP scheme unambiguity)

Therefore the probability of generated $S$ is

$$p_S = \frac{B(\{S\})}{B(S_\omega)} = \frac{e^{-E_s/RT}}{Z} = P_{S,\omega}$$
**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_s/RT}/Z$

**Stochastic backtrack:**

1. Draw uniform random number $r \in [0, Z'(i,j))$
2. Subtract from $r$ the contributions of $Z'(i,j)$ until $r < 0$
3. Recurse over associated regions/matrices

\[
Z'(i,j) = \begin{cases} 
    e^{-E_H(i',j')/RT} + e^{-E_S(i',j')/RT} Z'(i+1,j-1) \\
    \rightarrow \sum \left( e^{-E_B(i,i',j',j)/RT} Z'(i',j') \right) \\
    \rightarrow e^{-a/RT} \sum \left( Z(i+1,k-1) Z^1(k,j-1) \right)
\end{cases}
\]

Average-case complexity in $\Theta(k \times n\sqrt{n})$ (homopolymer model) [Pon08].

**Boustrophedon search** $\Rightarrow O(k \times n \log n)$ worst-case [Pon08].
**Goal [DL03]:** From sequence $\omega$, draw $S$ with prob. $e^{-E_S/RT}/Z$

**Stochastic backtrack:**

1. Draw uniform random number $r \in [0, Z'(i, j))$
2. Subtract from $r$ the contributions of $Z'(i, j)$ until $r < 0$
3. Recurse over associated regions/matrices

$$Z'(i, j) = \sum \left\{ e^{-E_H(i,j)/RT} + e^{-E_S(i,j)/RT} Z'(i + 1, j - 1) \right\}$$

$$\sum \left( e^{-E_B(i,i',j',j)/RT} Z'(i', j') \right)$$

$$e^{-\alpha RT} \sum (Z(i + 1, k - 1) Z^1(k, j - 1))$$

After $\Theta(n)$ operations, recurse over region of length $n - 1$

$\Rightarrow$ Worst-case complexity in $O(k \times n^2)$ for $k$ samples

Average-case complexity in $\Theta(k \times n\sqrt{n})$ (homopolymer model) [Pon08].

Boustrophedon search $\Rightarrow O(k \times n \log n)$ worst-case [Pon08].
Classifying RNA pseudoknotted structures.  

Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for rna secondary structure prediction.  

Y. Ding, C. Y. Chan, and C. E. Lawrence.  
RNA secondary structure prediction by centroids in a boltzmann weighted ensemble.  

Y. Ding and E. Lawrence.  
A statistical sampling algorithm for RNA secondary structure prediction.  

P. Gardner and R. Giegerich.  
A comprehensive comparison of comparative rna structure prediction approaches.  

Fast folding and comparison of RNA secondary structures.  

R. B. Lyngsø and C. N. S. Pedersen.  
RNA pseudoknot prediction in energy-based models.  

N. Leontis and E. Westhof.  
Geometric nomenclature and classification of RNA base pairs.  
Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure.

Jan Manuch, Chris Thachuk, Ladislav Stacho, and Anne Condon.
Np-completeness of the direct energy barrier problem without pseudoknots.

N. R. Markham and M. Zuker.

M. Parisien and F. Major.
The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data.

Y. Ponty.
Efficient sampling of RNA secondary structures from the boltzmann ensemble of low-energy: The boustrophedon method.

Lioudmila V Sharova, Alexei A Sharov, Timur Nedorezov, Yulan Piao, Nabeebi Shaik, and Minoru S H Ko.
Database for mrna half-life of 19 977 genes obtained by dna microarray analysis of pluripotent and differentiating mouse embryonic stem cells.

B. A. Shapiro, Y. G. Yingling, W. Kasprzak, and E. Bindewald.
Bridging the gap in rna structure prediction.