M2 BIM – STRUCT – Lecture 1
Folding RNA \textit{in silico}

Yann Ponty

AMIBio Team
CNRS LIX, École Polytechnique
Outline

Introduction
  Dynamic programming 101
  Why RNA?
  RNA folding
  RNA Structure(s)
  Some representations of RNA structure

Some flavours of folding prediction
  Thermodynamics vs Kinetics
  Dynamic programming: Reminder

Free-energy minimization
  Nussinov-style RNA folding
Foreword... 

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

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

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

21 =??

55

60
Foreword...

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

**Problem:** You have access to unlimited amount of 1, 20 and 50 cents coins. A client prefers to travel light, i.e. to minimize the #coins.

How to give N cents back in change without losing a customer?

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

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

\[
55??
\]

\[
60
\]
Foreword…

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

**Problem:** You have access to unlimited amount of 1, 20 and 50 cents coins. A client prefers to travel light, i.e. to minimize the #coins.

How to give N cents back in change without losing a customer?

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

\[
21 = \qquad + \qquad \\
55 = \qquad + \qquad + \qquad + \qquad + \qquad \\
60??
\]
Foreword... 

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

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

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

\[
21 = \text{??} + \text{??} + \text{??}
\]
\[
55 = \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??}
\]
\[
60 = \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??} + \text{??}
\]
Foreword…

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

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

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

21 = ⋅ + ⋅
55 = ⋅ + ⋅ + ⋅ + ⋅ + ⋅ + ⋅
60 = ⋅ + ⋅ + ⋅ + ⋅ + ⋅ + ⋅ + ⋅ + ⋅ + ⋅ + ⋅ ??

= ⋅ + ⋅ + ⋅ !

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

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

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

\[
\begin{align*}
\text{Min\#Coins}(N) &= \text{Min} \\
&\quad \rightarrow 1 + \text{Min\#Coins}(N - 1) \\
&\quad \rightarrow 1 + \text{Min\#Coins}(N - 20) \\
&\quad \rightarrow 1 + \text{Min\#Coins}(N - 50)
\end{align*}
\]

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.
Foreword...

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

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

$$Min\#\text{Coins}(N) = \min \left\{ \begin{array}{ll}
1 + Min\#\text{Coins}(N - 1) \\
1 + Min\#\text{Coins}(N - 20) \\
1 + Min\#\text{Coins}(N - 50) 
\end{array} \right\}$$

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.
Fundamental **dogma** of molecular biology

DNA

{A, C, G, T}*

\[ \text{RNA: } \{A, C, G, U\} \]

\[ \text{DNA: } \{A, C, G, T\} \]

\[ \text{Proteins: } \{\text{Ala, Arg, }\ldots, \text{Val}\} \]

\[ \text{The code (genes)} \]

\[ \text{The machine (enzymes)} \]

\[ \text{M.E.H.} \ldots \]
Fundamental *dogma* of molecular biology

**DNA**

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

**RNAs**

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

**THE CODE**

**THE MACHINE**

DNA → RNAs → Proteins → Amino acids

\[
\text{ATGGTTACCCAT} \\
\text{TAACAAATGGGTA}
\]
Fundamental *dogma* of molecular biology

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

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

\[\begin{array}{c}
\text{ATGCTTAACCCCAT} \\
\text{CCATGGGTA} \\
\text{Pol} \\
A
\end{array}\]
Fundamental **dogma** of molecular biology

- **DNA**
  - \( \{A, C, G, T\}^* \)

- **RNAs**
  - \( \{A, C, G, U\}^* \)

- **Proteins**
  - \( \{\text{Ala, Arg, ... , Val}\} \)

- **Amino acids**

**THE CODE**

**THE MACHINE**
Fundamental *dogma* of molecular biology

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

THE CODE (genes)

THE MACHINE (enzymes)

20 Amino acids
Fundamental *dogma* of molecular biology

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

THE CODE (genes)

THE MACHINE (enzymes)

```
ATGGTTACCCCAT
```

```
AUGGUAACCCAU
```
Fundamental *dogma* of molecular biology

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

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

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

- **DNA** \{A, C, G, T\}*
- **RNAs** \{A, C, G, U\}*
- **Proteins** \{Ala, Arg, . . . , Val\}*
  - 20+ Amino acids

![Diagram](image-url)

**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

**THE CODE** (genes)

**THE MACHINE** (enzymes)

\[
\begin{array}{c}
\text{ATG GTT AAC CCA AT CCA A T} \\
\text{TAC CAA A TG GGT A} \\
\text{AUGC UUA ACC CCA AU}
\end{array}
\]
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)

Ribosome

ATGGTTACCCCAT

ATACCAATGGGTA

AUUGUUACCCAU
Fundamental *dogma* of molecular biology

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

20+ Amino acids

---

**THE CODE**
**THE MACHINE**

Ribosome

- Met
- Val
- Thr
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)**

---

Ribosome

Met Val Thr His

---

\[ \begin{array}{cccc}
A & T & G & G \\
T & A & C & C \\
C & C & C & A \\
A & U & G & G \\
\end{array} \]
Fundamental *dogma* of molecular biology

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

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

Proteins
\{Ala, Arg, \ldots, Val\}^*

20^+ Amino acids

\begin{align*}
\text{ATGGTTACCCCAT} & \quad \text{TACCAATGGGTA} \\
\text{AUGGUUACCCAU} & \quad \text{Met Val Thr His Ile Leu His Asn}
\end{align*}
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

THE MACHINE (enzymes)

```
ATGGTTACCCCAT
```

```
TACCAATGGGTA
```

```
AUGGUUACCCAU
```

```
Met Val Thr His Ile Leu His Asn
```
Fundamental *dogma* of molecular biology

**THE CODE** (genes)
DNA
{A, C, G, T}*

**DNA**

RNAs
{A, C, G, U}*

**RNAs**

**THE MACHINE** (enzymes)

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

20+ Amino acids

**THE MACHINE**

Proteins

Met Val Thr His Ile Leu His Asn
Fundamental *dogma* of molecular biology

**THE CODE** *(genes)*

DNA

{A, C, G, T} *

**MEH...**

RNAs

{A, C, G, U} *

**THE MACHINE** *(enzymes)*

Proteins

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

20+ Amino acids

---

**THE CODE** *(genes)*

DNA

{A, C, G, T} *

**MEH...**

RNAs

{A, C, G, U} *

**THE MACHINE** *(enzymes)*

Proteins

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

20+ Amino acids

---

RNA:

```
{A, C, G, U}
```

DNA:

```
{A, C, G, T}
```

Proteins:

```
{Ala, Arg, ... , Val}
```

THE CODE

```
ATGGTTACCCCAT
```

MEH...

```
AUGGUAACCCAU
```

THE MACHINE

```
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
- ...

- Transcription
- Translation
- Regulation
RiboNucleic Acids (RNAs) in Human biology/health: Friends and Foes!

RiboNucleic Acids (RNAs)

Encodes proteins
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)
RiboNucleic Acids (RNAs) in Human biology/health: Friends and Foes!

Targeting system for DNA Editing
CRISPR therapies
Sickle-cell anemia, β-thalassemia, Leber congenital amaurosis (LCA), cancers...

RiboNucleic Acids (RNAs)

Encodes proteins
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)
RiboNucleic Acids (RNAs) in Human biology/health: Friends and Foes!

Targeting system for DNA Editing
CRISPR therapies
Sickle-cell anemia, β-thalassamia, Leber congenital amaurosis (LCA), cancers...

Sensor of metabolites
Riboswitches

RiboNucleic Acids (RNAs)

Encodes proteins
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)

Hendel et al, 2015; Agrotis & Ketteler, 2015
Berens & Suess 2015

Hendel et al, 2015; Agrotis & Ketteler, 2015

Targeting system for DNA Editing
CRISPR therapies
Sickle-cell anemia, β-thalassamia, Leber congenital amaurosis (LCA), cancers...

Sensor of metabolites
Riboswitches

RiboNucleic Acids (RNAs)

Encodes proteins
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)
RiboNucleic Acids (RNAs) in Human biology/health: Friends and Foes!

**Targeting system for DNA Editing**
- CRISPR therapies
- Sickle-cell anemia, β-thalassamia, Leber congenital amaurosis (LCA), cancers...

**Quantitative expression**
- Transcriptomic signatures
- Cancer diagnosis/prognosis/relapse...

**Sensor of metabolites**
- Riboswitches

**Encodes proteins**
- mRNA Vaccines
- COVID-19, Malaria (Zika, CMV, Cancers?)

*Hendel et al, 2015; Agrotis & Ketteler, 2015
Berens & Suess 2015
[NGuyen et al, 2021]*
RiboNucleic Acids (RNAs) in Human biology/health: Friends **and** Foes!

**Targeting system for DNA Editing**
CRISPR therapies
Sickle-cell anemia, β-thalassamia, Leber congenital amaurosis (LCA), cancers...

**Quantitative expression**
Transcriptomic signatures
Cancer diagnosis/prognosis/relapse...

**Sensor of metabolites**
Riboswitches

**Non-coding mutations**
IncRNAs, miRNAs, structure-associated (RiboSnitches)
β-thalassemia, duchenne muscular dystrophy,
Cystic fibrosis, Rett syndrome...

**Encodes proteins**
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)

Hendel et al, 2015; Agrotis & Ketteler, 2015
Berens & Suess 2015
[NGuyen et al, 2021]
Solem et al, 2015

Encyclopedia Britannica, Inc 2013
Watts et al, Nature 2009

Genomic material for Human pathogens
HIV-1, SARS-CoV 2, HCoVs, MERS

Encodes proteins
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)
RiboNucleic Acids (RNAs) in Human biology/health: Friends and Foes!

Targeting system for DNA Editing
CRISPR therapies
Sickle-cell anemia, β-thalassemia, Leber congenital amaurosis (LCA), cancers...

Quantitative expression
Transcriptomic signatures
Cancer diagnosis/prognosis/relapse...

Non-coding mutations
IncRNAs, miRNAs, structure-associated (RiboSnitches)
β-thalassemia, duchenne muscular dystrophy, Cystic fibrosis, Rett syndrome...

Sensor of metabolites
Riboswitches

Genomic material for Human pathogens
HIV-1, SARS-CoV 2, HCoVs, MERS

Encodes proteins
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)

Watts et al, Nature 2009

Encyclopedia Britannica, Inc. 2013

Solem et al, 2015

NGuyen et al, 2021

Berens & Suess 2015

Hendel et al, 2015; Agrotis & Ketteler, 2015

Hendel et al, 2015; Agrotis & Ketteler, 2015

Targeting system for DNA Editing
CRISPR therapies
Sickle-cell anemia, β-thalassemia, Leber congenital amaurosis (LCA), cancers...

Quantitative expression
Transcriptomic signatures
Cancer diagnosis/prognosis/relapse...

Non-coding mutations
IncRNAs, miRNAs, structure-associated (RiboSnitches)
β-thalassemia, duchenne muscular dystrophy, Cystic fibrosis, Rett syndrome...

Sensor of metabolites
Riboswitches

Genomic material for Human pathogens
HIV-1, SARS-CoV 2, HCoVs, MERS

Encodes proteins
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)

Watts et al, Nature 2009

Encyclopedia Britannica, Inc. 2013

Solem et al, 2015

NGuyen et al, 2021

Berens & Suess 2015

Hendel et al, 2015; Agrotis & Ketteler, 2015
RiboNucleic Acids (RNAs) in Human biology/health: Friends and Foes!

**Targeting system for DNA Editing**
CRISPR therapies
Sickle-cell anemia, β-thalassemia, Leber congenital amaurosis (LCA), cancers...

**Quantitative expression**
Transcriptomic signatures
Cancer diagnosis/prognosis/relapse...

**Sensor of metabolites**
Riboswitches
Berens & Suess 2015

**Non-coding mutations**
IncRNAs, miRNAs, structure-associated (RiboSnitches)
β-thalassemia, duchenne muscular dystrophy,
Cystic fibrosis, Rett syndrome...

**Regulation of gene expression**
RNA therapies (FDA approved)
Primary hyperoxaluria type 1 (PH1),
Hereditary transthyretin amyloidosis (ATTRv),
Acute hepatic porphyria (AHP)

**Genomic material for Human pathogens**
HIV-1, SARS-CoV 2, HCoVs, MERS

**Encodes proteins**
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)

**x 10...**
Watts et al, Nature 2009

Encyclopedia Britannica, Inc 2013
RNA functional diversity is (largely) enabled by deep structural diversity
RiboNucleic Acids (RNAs) in Human biology/health: Friends and Foes!

Rational design

Targeting system for DNA Editing
CRISPR therapies
Sickle-cell anemia, β-thalassaemia, Leber congenital amaurosis (LCA), cancers...

Quantitative expression
Transcriptomic signatures
Cancer diagnosis/prognosis/relapse...

Genomic material for Human pathogens
HIV-1, SARS-CoV 2, HCoVs, MERS

Regulation of gene expression
RNA therapies (FDA-approved)
Primary hyperoxaluria type 1 (PH1), Hereditary transthyretin amyloidosis (ATTRv), Acute hepatic porphyria (AHP)

Encodes proteins
mRNA Vaccines
COVID-19, Malaria (Zika, CMV, Cancers?)

Sensor of metabolites
Riboswitches

Non-coding mutations
IncRNAs, miRNAs, structure-associated (RiboSnitches)
β-thalassemia, duchenne muscular dystrophy, Cystic fibrosis, Rett syndrome...

Solem et al, 2015

[NGuyen et al, 2021]

[(2D) Structure Modeling]
RNA world: Resolving the *chicken vs egg* paradox at the origin of life...

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 world: Resolving the *chicken vs egg* paradox at the origin of life...

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 folding

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.
RNA Structure(s)

Three\(^1\) levels of representation:

- **Primary structure**
- **Secondary structure**
- **Tertiary structure**

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

\(^1\)Well, mostly...
Three¹ levels of representation:

UUAGGCGCCAACA GC
GGUGGGGUUGCCUC GC
CGUACCAUCCCGAA
CACGGAGAAAGCC
CACCAGCGUUCCGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC

Primary structure

Secondary+ structure

Tertiary structure

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

¹Well, mostly...
Ignored by secondary structure

- **Non-canonical base-pairs**
  - Any base-pair other than \{(A-U), (C-G), (G-U)\}
  - Or interacting on non-standard edge \(\neq \text{WC/WC-Cis}\) [LW01].

- **Pseudoknots (PKs)**

Considering PKs may lead to better predictions, but:
  - Some PK conformations are simply unfeasible;
  - Folding *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

- Dot plots
  - Adjacency matrices

- Non-crossing arc diagrams

Supporting intuitions

- Different representations
  - Common combinatorial structure

* Additional steric constraints
Various representations for a versatile biomolecule

- Outer-planar graphs
  - Hamiltonian-path, $\Delta(G) \leq 3$, 2-connected
- Motzkin words*
- Positive 1D meanders* over $S = \{+1, -1, 0\}$
- Non-crossing arc-annotated sequences*

**Supporting intuitions**

Different representations
Common combinatorial structure

* Additional steric constraints
Outline

Introduction
  Dynamic programming 101
  Why RNA?
  RNA folding
  RNA Structure(s)
  Some representations of RNA structure

Some flavours of folding prediction
  Thermodynamics vs Kinetics
  Dynamic programming: Reminder

Free-energy minimization
  Nussinov-style RNA folding
Thermodynamics *aparté*

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

Convergence towards a **stationary distribution** at the Boltzmann equilibrium, where the probability of a conformation only depends on its **free-energy**.

**Corollary:** Initial conformation does not matter.

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

A. Determine most stable (Minimum Free-Energy) structure at equilibrium;
B. Compute average properties of Boltzmann ensemble;
Away from equilibrium

Transcription: RNA synthesized, supposedly without structure\(^2\)

\[ T = 0 \]

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

\(^2\)Except for co-transcriptional folding...
Away from equilibrium

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...
Away from equilibrium

**Transcription**: RNA synthesized, supposedly without structure

\[ T = 2h \]

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

\[ ^2 \text{Except for co-transcriptional folding...} \]
Away from equilibrium

**Transcription**: RNA synthesized, supposedly without structure\(^2\)

\[ T = 5h \]

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

\(^2\)Except for co-transcriptional folding...
Away from equilibrium

**Transcription:** RNA synthesized, supposedly without structure\(^2\)

\[ T = 10h \]

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

\(^2\)Except for co-transcriptional folding...
Away from equilibrium

**Transcription:** RNA synthesized, supposedly without structure\(^2\)

\[ T \to \infty \]

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

---

\(^2\)Except for co-transcriptional folding...
Away from equilibrium

**Transcription:** RNA synthesized, supposedly without structure

\[ T = 10h \]

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...
Dynamic programming: General principle

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

Bioinformatics:

- Discrete solution space (alignments, structures...)
- Additively-inherited objective function (cost, log-odd score, energy...)
⇒ Efficient dynamic programming scheme

Example: Local Alignment (Smith/Waterman)

$$
W(i, 0) = 0 \\
W(0, j) = 0 \\
W(i, j) = \max \left\{ \begin{array}{l}
W(i - 1, j - 1) + m_{i,j} \\
W(i - 1, j) + p_i \\
W(i, j - 1) + p_d
\end{array} \right.
$$
Algorithmic details

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

Given a scheme, two steps:

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

Complexity of algorithm depends on:

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

Smith&Waterman example:

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

\( W(i, 0) = 0 \)
\( W(0, j) = 0 \)
\( W(i, j) = \max \left\{ W(i - 1, j - 1) + m_{i,j}, W(i - 1, j) + p_i, W(i, j - 1) + p_d \right\} \)

\( \Rightarrow \Theta(m.n) \) time/memory
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

W(0,0) = 0

W(i,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}
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

$$
W(i, 0) = 0 \\
W(0, j) = 0 \\
W(i, j) = \max \left\{ \begin{array}{l}
W(i - 1, j - 1) + m_{i,j} \\
W(i - 1, j) + p_i \\
W(i, j - 1) + p_d
\end{array} \right.
$$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>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>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</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>
</tr>
<tr>
<td>C</td>
<td>0</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>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</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>
</tr>
<tr>
<td>C</td>
<td>0</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>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Complete example

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

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

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

\[
\begin{array}{cccccccc}
\text{A} & \text{C} & \text{A} & \text{C} & \text{A} & \text{C} & \text{T} & \text{A} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\text{A} & 0 & \text{2} & \rightarrow & 1 & \rightarrow & 2 \\
\text{G} & 0 \\
\text{C} & 0 \\
\text{A} & 0 \\
\text{C} & 0 \\
\text{A} & 0 \\
\text{C} & 0 \\
\text{A} & 0 \\
\end{array}
\]
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
W(i, 0) = 0 \\
W(0, j) = 0 \\
W(i, j) = \max \left\{ W(i-1, j-1) + m_{i,j}, W(i-1, j) + p_i, W(i, j-1) + p_d \right\}
\]

![Alignment Matrix](image)
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

\[
W(i, 0) = 0 \\
W(0, j) = 0 \\
W(i, j) = \max \left\{ W(i-1, j-1) + m_{i,j}, W(i-1, j) + p_i, W(i, j-1) + p_d \right\}
\]
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

Example: Local alignment of AGCACACA and ACACACTA

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

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

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</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>
</tr>
<tr>
<td>C</td>
<td>0</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>
</tr>
</tbody>
</table>
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

\[
W(i, 0) = 0 \\
W(0, j) = 0 \\
W(i, j) = \max \left\{ W(i-1, j-1) + m_{i,j}, W(i-1, j) + p_i, W(i, j-1) + p_d \right\}
\]

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>0</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>
</tr>
<tr>
<td>C</td>
<td>0</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>
</tr>
</tbody>
</table>
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

Example: Local alignment of AGCACACA and ACACACTA

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

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

Example: Local alignment of AGCACACA and ACACACTA

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

$$W(i, 0) = 0$$
$$W(0, j) = 0$$
$$W(i, j) = \max \left\{ W(i-1, j-1) + m_{i,j}, W(i-1, j) + p_i, W(i, j-1) + p_d \right\}$$

Best alignment
Complete example

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

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

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

Best alignment

\[
\begin{array}{cccccccc}
A & C & A & C & A & C & T & A \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 2 \\
G & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
C & 0 & 0 & 3 & 2 & 3 & 2 & 3 & 2 & 1 \\
A & 0 & 2 & 2 & 5 & 4 & 5 & 4 & 3 & 4 \\
C & 0 & 1 & 1 & 4 & 7 & 6 & 7 & 6 & 5 \\
A & 0 & 2 & 3 & 6 & 9 & 8 & 8 & 7 & 8 \\
C & 0 & 1 & 4 & 5 & 8 & 8 & 11 & 10 & 9 \\
A & 0 & 2 & 3 & 6 & 7 & 10 & 10 & 10 & 12
\end{array}
\]
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

\[
W(i, 0) = 0,
W(0, j) = 0,
W(i, j) = \max\left\{ W(i-1, j-1) + m_{i,j}, W(i-1, j) + p_i, W(i, j-1) + p_d \right\}
\]

Best alignment
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

Best alignment

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>

Best alignment
Complete example

Example: Local alignment of \text{AGCACACA} and \text{ACACACTA}

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

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

Best alignment

\[
\begin{array}{ccccccc}
A & C & A & C & A & C & T & A \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 2 \\
G & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
C & 0 & 0 & 3 & 2 & 3 & 2 & 3 & 2 & 1 \\
A & 0 & 2 & 2 & 5 & 4 & 5 & 4 & 3 & 4 \\
C & 0 & 1 & 4 & 4 & 7 & 6 & 7 & 6 & 5 \\
A & 0 & 2 & 3 & 6 & 6 & 9 & 8 & 7 & 8 \\
C & 0 & 1 & 4 & 5 & 8 & 8 & 11 & 10 & 9 \\
A & 0 & 2 & 3 & 6 & 7 & 10 & 10 & 10 & 12 
\end{array}
\]
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

Best alignment

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

Best alignment

\[
\begin{array}{cccccccc}
A & C & A & C & - & A \\
A & C & A & C & T & A \\
\end{array}
\]
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

Best alignment

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

W($i$, 0) = 0
W(0, $j$) = 0
W($i$, $j$) = max

W($i$, $j$) = W($i-1$, $j-1$) + $m_{i,j}$
W($i$, $j$) = W($i-1$, $j$) + $p_i$
W($i$, $j$) = W($i$, $j-1$) + $p_d$
Complete example

Example: Local alignment of AGCACACA and ACACACTA

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

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

Best alignment

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

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}
**Complete example**

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

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

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

**Best alignment**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>C</th>
<th>T</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

W(i, 0) = 0 
W(0, j) = 0 
W(i, j) = max \left\{ \begin{array}{l}
W(i-1, j-1) + m_{i,j} \\
W(i-1, j) + p_i \\
W(i, j-1) + p_d
\end{array} \right. 

Best alignment

A  G  C  A  C  A  C  -  A  
A  -  C  A  C  A  C  T  A
Properties of DP schemes

Necessary properties:
▶ **Correctness**: $\forall$ sub-problem, the computed value must indeed maximize the objective function.
Proofs usually inductive, and quite technical, but very systematic.

Desirable properties of DP schemes:
▶ **Completeness** of space of solutions generated by decomposition.
  Algorithmic tricks, by *cutting branches*, may violate this property.
▶ **Unambiguity**: Each solution is generated at most once.
⇒ Under these properties, one can enumerate solution space.
Outline

Introduction
- Dynamic programming 101
- Why RNA?
- RNA folding
- RNA Structure(s)
- Some representations of RNA structure

Some flavours of folding prediction
- Thermodynamics vs Kinetics
- Dynamic programming: Reminder

Free-energy minimization
- Nussinov-style RNA folding
Folding by minimizing free-energy

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

**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 model

Nussinov/Jacobson energy model (NJ)

Base-pair maximization (with a twist):

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

⇒ \( E_{\omega,S} = -\# \text{Paires}(S) \)

Folding in NJ model ⇔ Base-pair (weight) maximization

Example:

```
UUUUCUAAAAGG
```

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 model

Nussinov/Jacobson energy model (NJ)

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

⇒ \( E_{\omega,S} = -\#\text{Paires}(S) \)

Folding in NJ model ⇔ Base-pair (weight) maximization

Example:

UUUUCUCUAAGG

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 DP scheme

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

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

\[ \begin{array}{c}
\text{i} \quad \ldots \quad \text{j} \\
\text{i} \quad \ldots \quad \text{i+1} \quad \text{j} \\
\text{i} \quad \text{k} \quad \text{j}
\end{array} \]

\[
N_{i,t} = \begin{cases}
0, & \forall t \in [i, i + \theta] \\
N_{i+1,j} & \text{i unpaired}
\end{cases}
\]

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

Corrrectness. 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: \(\text{MFE}_{i,j} = \text{MFE}_{i+1,j}\) \(\rightarrow\) free-energy = \(N_{i+1,j}\)
  - Paired to \(k\): \(\text{MFE}_{i,j} = \{(i, k)\} \cup \text{MFE}_{i+1,k-1} \cup \text{MFE}_{k+1,j}\)
    (Indeed, any BP between \([i+1, k-1]\) and \([k+1, j]\) would cross \((i, k)\))
    \(\rightarrow\) free-energy = \(\Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j}\)
Nussinov/Jacobson DP scheme

\[ i \rightarrow \ldots \rightarrow j = i \rightarrow \ldots \rightarrow i+1 + \begin{cases} \geq \theta \end{cases} \]

\[
N_{i,t} = 0, \quad \forall t \in [i, i + \theta] \\
N_{i,j} = \min \left\{ \begin{array}{ll}
N_{i+1,j} & \text{i unpaired} \\
\min_{k=i+\theta+1} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j} & \text{i paired with k}
\end{array} \right.
\]

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 \(\text{MFE}_{i,j} := \text{Base-pairs of best struct. on} \ [i, j]\). 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

\[ i \underset{\geq \theta}{\overset{\rightarrow}{\cdots}} j = i \overset{\rightarrow}{\cdots} i+1 \quad j + \quad i \overset{\rightarrow}{\cdots} i \quad k \quad j \]

\[
N_{i,t} = 0, \quad \forall t \in [i, i + \theta] \\
N_{i,j} = \min \left\{ \begin{array}{l}
N_{i+1,j} \\
\min_{k=i+\theta+1} \Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j}
\end{array} \right\}
\]

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: \(\text{MFE}_{i,j} = \text{MFE}_{i+1,j}\) \(\rightarrow\) free-energy = \(N_{i+1,j}\)
  - Paired to \(k\): \(\text{MFE}_{i,j} = \{ (i, k) \} \cup \text{MFE}_{i+1,k-1} \cup \text{MFE}_{k+1,j}\).
    (Indeed, any BP between \([i + 1, k - 1]\) and \([k + 1, j]\) would cross \((i, k)\))
    \(\rightarrow\) free-energy = \(\Delta G_{i,k} + N_{i+1,k-1} + N_{k+1,j}\)
### Nussinov/Jacobson

#### Table

<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>A</th>
<th>C</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>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>6</td>
<td>8</td>
<td>10</td>
<td>10</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>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>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>
</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>8</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>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>
</tbody>
</table>
| A | 0

#### Diagram

\[ i \leq j \leq i+1 \]

\[ i \leq k \leq j \]

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

\[
i \leq j \\
\text{i} \quad \text{j} = \text{i} \quad \text{i+1} \quad \text{j} + \text{i} \quad \text{k} \quad \text{j} \geq \theta
\]

\[
Nussinov/Jacobson
\]
Nussinov/Jacobson

|   | C | G | G | A | U | A | C | U | U | C | U | U | A | G | A | C | G | A |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 4 | 4 | 6 | 6 | 6 | 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 | 2 | 2 | 2 | 2 | 4 | 4 | 5 | 7 | 7 | 8 | 8 | 10 |
| U | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 4 | 5 | 7 | 7 | 8 | 8 | 10 |
| A | 0 | 0 | 0 | 0 | 0 | 2 | 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 | 0 | 2 | 3 | 5 | 5 | 5 | 6 | 7 |
| U | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 5 | 7 |
| C | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 5 | 5 |
| U | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 3 |
| U | 0 | 0 | 0 | 0 | 1 | 2 |
| A | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 0 | 0 | 0 |
| C | = i \rightarrow i+1 \rightarrow j + i \rightarrow k \rightarrow j \geq \theta |
| G | 0 | 0 | 0 | 0 |
| A | 0 | 0 | 0 | 0 | 0 |

\[ i \leq \theta \]
|   | 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 | 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 | 2 | 4 | 5 | 7 | 7 | 8 | 10 |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 8 | 8 |
| C | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 5 | 5 | 5 | 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 | 3 | 3 | 3 | 5 | 5 |
| U | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 3 |
| U | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| A | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 | 0 |

\[ i \leq j = i + 1 \]

\[ i + 1 \leq k \]

\[ \theta \]
<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>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>
</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>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>
</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>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>
</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>2</td>
<td>3</td>
<td>5</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>3</td>
<td>5</td>
<td>5</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>1</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></td>
<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>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>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>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>

\[
i \leq \theta \leq j \geq i + 1\]
<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>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>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>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>3</td>
<td>5</td>
<td>5</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></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>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>C</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>C</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>

\[ i \leq j \leq i + 1 \]

\[ i \leq k \leq j \]

\[ \geq \theta \]
\begin{align*}
\text{C} & \quad \text{G} & \quad \text{G} & \quad \text{A} & \quad \text{U} & \quad \text{A} & \quad \text{C} & \quad \text{U} & \quad \text{U} & \quad \text{C} & \quad \text{U} & \quad \text{U} & \quad \text{A} & \quad \text{G} & \quad \text{A} & \quad \text{C} & \quad \text{G} & \quad \text{A} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 3 & \quad 4 & \quad 4 & \quad 6 & \quad 6 & \quad 6 & \quad 6 & \quad 6 & \quad 9 & \quad 9 & \quad 11 & \quad 14 & \quad 14 \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 3 & \quad 4 & \quad 4 & \quad 6 & \quad 6 & \quad 6 & \quad 6 & \quad 7 & \quad 9 & \quad \text{11} & \quad 11 & \quad 11 \\
0 & \quad 0 & \quad 0 & \quad \text{0} & \quad 0 & \quad 3 & \quad 3 & \quad 3 & \quad 5 & \quad 5 & \quad 5 & \quad 5 & \quad 6 & \quad 8 & \quad \text{10} & \quad \text{10} & \quad \text{10} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 2 & \quad 2 & \quad 2 & \quad 2 & \quad 4 & \quad 4 & \quad 5 & \quad 7 & \quad \text{7} & \quad \text{7} & \quad \text{8} & \quad \text{10} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 2 & \quad 2 & \quad 2 & \quad 5 & \quad 5 & \quad 5 & \quad 5 & \quad 8 & \quad \text{8} & \quad \text{8} & \quad \text{8} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 2 & \quad 3 & \quad 5 & \quad \text{5} & \quad \text{6} & \quad \text{7} & \quad \text{5} & \quad \text{5} & \quad \text{5} & \quad \text{7} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 2 & \quad 3 & \quad 5 & \quad 5 & \quad 5 & \quad 3 & \quad \text{3} & \quad \text{5} & \quad \text{5} & \quad \text{5} & \quad \text{5} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 2 & \quad 2 & \quad 2 & \quad 2 & \quad 3 & \quad \text{3} & \quad \text{5} & \quad \text{5} & \quad \text{5} & \quad \text{5} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 1 & \quad \text{2} & \quad \text{3} & \quad \text{3} & \quad \text{5} & \quad \text{5} & \quad \text{5} & \quad \text{5} & \quad \text{5} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} \\
0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} & \quad \text{0} \\
\end{align*}

\[ i \leq j + 1 \]

\[ i \leq k \]

\[ \geq \theta \]
|   | C | G | G | A | U | A | C | U | U | C | U | U | A | G | A | C | G | A |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

(Nussinov/Jacobson)

\[ i, j = i, i+1 \] + \[ j, k \geq \theta \]
\[ i = \sum_{j=i}^{j+1} \sum_{i}^{k} \geq \theta \]
\[
i \leq \theta
\]
|   | 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 | 0 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 6 | 8 | 10 | 10 | 10 |   |   |   |   |   |   |   |   |
| A | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 4 | 4 | 5 | 7 | 7 | 8 | 10 |   |   |   |   |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 4 | 5 | 7 | 7 | 8 | 10 |   |   |   |   |   |   |   |   |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 8 | 8 |   |   |   |   |   |   |   |   |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 5 | 5 | 5 | 8 | 8 |   |   |   |   |   |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 5 | 5 | 6 | 7 |   |   |   |   |   |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 5 | 5 | 7 |   |   |   |   |   |   |   |   |   |   |   |
| C | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 5 | 5 | 5 | 5 |   |   |   |   |   |   |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 3 |   |   |   |   |   |   |   |   |   |   |   |
| U | 0 | 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 |   |   |   |   |   |   |   |   |   |   |

\[
i \leq j \quad \Rightarrow \quad C_{ij} + C_{i,k+1} \geq \theta_{ij}
\]
\[ i \leq j \leq i + 1 \geq \theta \]
### Nussinov/Jacobson

\[
i, j = i, i+1 + j, k \geq \theta
\]
\[ i \leq j \leq i+k \geq \theta \]
Nussinov/Jacobson

\[ i \leq j \leq i+1 \]

\[ \geq \theta \]
\[
i_{ij} = i_{i+1} + i_{k}\quad \text{if} \quad i_{ij} \geq \theta
\]
Nussinov/Jacobson

\[ C \ G \ G \ A \ U \ A \ C \ U \ U \ C \ U \ U \ A \ G \ A \ C \ G \ A \]

\[
\begin{array}{cccccccccccc}
C & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 9 & 9 & 11 & 14 & 14 \\
G & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 7 & 9 & 11 & 11 & 11 \\
G & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 6 & 8 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 5 & 7 & 7 & 8 & 8 & 10 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 5 & 7 & 7 & 8 & 8 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 5 & 5 & 5 & 5 & 8 & 8 & 8 & 8 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 5 & 5 & 8 & 8 & 8 & 8 & 8 & 8 \\
U & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 3 & 5 & 5 & 6 & 7 & 7 & 7 & 7 & 7 \\
U & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 5 & 5 & 7 & 7 & 7 & 7 & 7 & 7 \\
C & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
U & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
U & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A & \]

\[
\begin{array}{cccc}
C & \end{array}
\begin{array}{cccc}
G & \end{array}
\begin{array}{cccc}
A & \end{array}
\begin{array}{cccc}
A & \end{array}
\begin{array}{cccc}
C & \end{array}
\end{array}
\]

\[\begin{array}{cccccccc}
i & j & = & i & i+1 & j & + & \geq \theta & j & & & & & & & & \end{array}\]
\[
i \leq n - \theta \leq j
\]
## Nussinov/Jacobson

### Table

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

### Diagram

\[ i \leq j, \quad i = i+1 \quad \geq \theta \quad i \quad k \quad j \]
Nussinov/Jacobson

\[
i_j = i_{i+1} + \begin{cases} \geq \theta & \text{if } G \text{ at } j \\ 0 & \text{otherwise} \end{cases}
\]
<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>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>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>
</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>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>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>
</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>
</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>
</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>
</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>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>
</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>
</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>
| A | 0 | 0 | 0 | 0
\[ i \leq j \leq i + 1 \]

\[ C \quad G \quad G \quad A \quad U \quad A \quad C \quad U \quad U \quad C \quad U \quad U \quad A \quad G \quad A \quad C \quad G \quad A \]

\[
\begin{array}{cccccccccccccccccc}
C & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 6 & 9 & 9 & 11 & 14 & 14 \\
G & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 6 & 6 & 6 & 6 & 6 & 7 & 9 & 11 & 11 & 11 \\
G & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 6 & 8 & 10 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 5 & 7 & 7 & 8 & 10 & 10 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 4 & 5 & 7 & 7 & 8 & 8 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 8 & 8 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 5 & 5 & 5 & 8 & 8 \\
U & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 6 & 7 \\
U & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 5 & 5 & 7 \\
C & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 5 & 5 \\
U & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\
U & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\
A & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 \\
C & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 \\
A & 0 & 0 & 0 & 0 \\
\end{array}
\]
\[
i_j = i_{i+1} + i_k \geq \theta
\]
Nussinov/Jacobson

\[
i = i \rightarrow i + 1
\]
\begin{align*}
&C \quad G \quad G \quad A \quad U \quad A \quad C \quad U \quad C \quad U \quad A \quad G \quad A \quad C \quad G \quad A \\
&\begin{pmatrix}
C & 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 & 6 & 7 & 9 & 11 & 11 & 11 \\
G & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 6 & 8 & 8 & 10 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 4 & 5 & 7 & 7 & 8 & 8 & 10 & 10 & 10 & 10 \\
U & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 5 & 7 & 7 & 8 & 8 & 10 & 10 & 10 & 10 & 10 & 10 \\
A & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 5 & 5 & 5 & 5 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
C & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 5 & 5 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
U & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 3 & 5 & 5 & 5 & 6 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
U & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 & 5 & 5 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
C & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 5 & 5 & 5 & 5 & 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 & 3 & 3 & 3 \\
U & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 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 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 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 & 0 & 0 & 0 \\
C & i \quad j = i \quad i+1 \quad j & + & i \quad k \quad j & \geq \theta \\
G & \end{pmatrix}
\end{align*}
### Table

|   | C | G | G | A | U | A | C | U | U | C | U | U | A | G | A | C | G | A |
| C | 0 | 0 | 0 | 0 | 0 | 3 | 4 | 4 | 6 | 6 | 6 | 6 | 9 | 9 | 11 | 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 | 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 | 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 | 5 | 6 | 7 |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 5 | 5 | 5 | 7 |   |   |   |   |   |
| C | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 5 |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 3 |   |   |   |   |   |   |   |
| U | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |   |   |   |   |   |   |   |   |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |
| G | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |
| A | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |

### Diagram

![Diagram](image-url)

- \( i \) and \( j \) are positions in the sequence.
- \( i \) and \( i+1 \) are consecutive positions.
- \( k \) is a position that is \( \geq \theta \) from \( i \).

\( i \) and \( j \) are connected if \( i \) and \( j \) are not consecutive and \( j \) is \( \geq \theta \) from \( i \).
|   | 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 | 11 |
| G | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 5 | 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 | 8 | 10 | 10 | 10 | 10 |
| A | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 | 8 | 8 | 8 | 8 |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 5 | 5 | 5 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| U | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 5 | 5 | 6 | 7 | 7 | 7 | 7 | 7 | 7 |
| U | 0 | 0 | 0 | 0 | 2 | 3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 7 | 7 | 7 | 7 | 7 |
| C | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 5 | 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 | 1 | 2 | 2 | 2 | 2 | 2 | 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 | 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 |

\[ i \leq j \]

\[ n \]

\[ i \leq k \]

\[ j \]

\[ \geq \theta \]
### RNA Structure Energy Calculation

The energy of an RNA structure can be calculated using the following formula:

\[
i, j = i \rightarrow i+1, j \bigoplus \sum_{k=0}^{3} \geq \theta
\]

where the energy of a base pair is given by:

- C-G

\[
C=6
\]

- G-A

\[
G=4
\]

- A-U

\[
A=10
\]

The total energy can be calculated by summing the energies of all possible base pairs in the RNA structure.
Classifying RNA pseudoknotted 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.

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

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.