COUNTING, GENERATING AND SAMPLING TREE ALIGNMENTS

Cédric CHAUVE (Simon Fraser University, Vancouver)
Julien COURTIEL (PIMS/Univ. of British Columbia, Vancouver)
Yann PONTY (CNRS/LIX)

LRI, January 14th 2016
MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?
**Motivation: RNA comparison**

**Question:** how to measure similarity between two RNAs?

First idea: compare nucleic acid sequences.

RNA 1: AUUUCGAUUA...
RNA 2: ACCAUGAUUA...
Motivation: RNA comparison

Question: how to measure similarity between two RNAs?

First idea: compare nucleic acid sequences.

→ sequence alignment

RNA 1: AUUCGAAUUA...
RNA 2: ACCAUGCAGUUA...


First idea: compare nucleic acid sequences.

→ sequence alignment

RNA 1: AUUCGAAUUA...
RNA 2: ACCAUGCAGUUA...


Motivation: RNA Comparison

Question: how to measure similarity between two RNAs?

Second idea: compare secondary structures.

$\Rightarrow$ notion of tree alignment

[Jiang, Wang, Zhang]
FROM SECONDARY STRUCTURES TO TREES
FROM SECONDARY STRUCTURES TO TREES
FROM SECONDARY STRUCTURES TO TREES
FROM SECONDARY STRUCTURES TO TREES
FROM SECONDARY STRUCTURES TO TREES
FROM SECONDARY STRUCTURES TO TREES
FROM SECONDARY STRUCTURES TO TREES

Objective: Align trees coming from RNA secondary structures
FROM SECONDARY STRUCTURES TO TREES

Objective: Align trees coming from RNA 2^nd structures
**TREES AND SUPERTREES**

Trees are plane, rooted, and vertices are labeled by an alphabet $\Sigma$.

**Supertree** = tree with 3 types of vertices:

- $XY$ (mis)match
- $X -$ insertion
- $- Y$ deletion
Let $A$ be a supertree,

\[ \Pi_1(A) = \text{tree obtained by changing} \]
\[ \begin{align*}
XY & \rightarrow \times \\
X- & \rightarrow \times
\end{align*} \]

and removing $-Y$.

\[ \Pi_2(A) = \text{tree obtained by changing} \]
\[ \begin{align*}
XY & \rightarrow Y \\
-Y & \rightarrow Y
\end{align*} \]

and removing $X-$. 

\[ \text{Graph of } A \]

\[ \begin{array}{c}
A \\
\downarrow \\
C- \\
\downarrow \\
U \\
\downarrow \\
C- \\
\downarrow \\
UA
\end{array} \]
Let $A$ be a supertree,

$\mathcal{T}_1(A) =$ tree obtained by changing $XY \rightarrow X$ and removing $-Y$.

$\mathcal{T}_2(A) =$ tree obtained by changing $XY \rightarrow Y$ and removing $X-$. 

---

**Supertrees Induce Tree Alignments**
Let $A$ be a supertree,

$\Pi_1(A) =$ tree obtained by changing $\text{XY} \rightarrow \text{X}$

and removing $\text{Y}$.

$\Pi_2(A) =$ tree obtained by changing $\text{XY} \rightarrow \text{Y}$

$\text{-Y} \rightarrow \text{Y}$

and removing $\text{X-}$.
Let $A$ be a supertree,

$T_{T_1}(A) =$ tree obtained by changing

$\begin{align*}
\text{XY} &\rightarrow \text{X} \\
\text{X-} &\rightarrow \text{X}
\end{align*}$

and removing $\text{-Y}$.

$T_{T_2}(A) =$ tree obtained by changing

$\begin{align*}
\text{XY} &\rightarrow \text{Y} \\
\text{-Y} &\rightarrow \text{Y}
\end{align*}$

and removing $\text{X-}$.
Let $A$ be a supertree,

$T_{1}(A) = \text{tree obtained by changing } \begin{array}{c}
\text{XY} \\
\text{X-}
\end{array} \rightarrow \begin{array}{c}
\text{X} \\
\text{X}
\end{array}
\text{and removing } -Y$. 

$T_{2}(A) = \text{tree obtained by changing } \begin{array}{c}
\text{XY} \\
\text{-Y}
\end{array} \rightarrow \begin{array}{c}
\text{Y} \\
\text{Y}
\end{array}
\text{and removing } X-$. 

A
Let $A$ be a supertree,

$\Pi_1(A) =$ tree obtained by changing

\[
\begin{array}{c}
XY \\
X- \\
\end{array} 
\rightarrow 
\begin{array}{c}
X \\
X \\
\end{array} 
\]

and removing $-Y$.

$\Pi_2(A) =$ tree obtained by changing

\[
\begin{array}{c}
XY \\
-\ Y \\
\end{array} 
\rightarrow 
\begin{array}{c}
Y \\
Y \\
\end{array} 
\]

and removing $X-$.
Let $A$ be a supertree,

$$\Pi_1(A) = \text{tree obtained by changing}$$

- $XY \rightarrow X$
- $X- \rightarrow X$

and removing $-Y$.

$$\Pi_2(A) = \text{tree obtained by changing}$$

- $XY \rightarrow Y$
- $-Y \rightarrow Y$

and removing $X-$.
Let $A$ be a supertree,

$$\Pi_1(A) = \text{tree obtained by changing } \begin{array}{c} XY \\ X- \end{array} \rightarrow \begin{array}{c} X \\ X \end{array} \text{ and removing } \begin{array}{c} -Y \end{array}. $$

$$\Pi_2(A) = \text{tree obtained by changing } \begin{array}{c} XY \\ -Y \end{array} \rightarrow \begin{array}{c} Y \\ Y \end{array} \text{ and removing } \begin{array}{c} X- \end{array}. $$
Let $A$ be a supertree,

$$\Pi_1(A) = \text{tree obtained by changing } \boxed{XY} \rightarrow \boxed{XX}$$

and removing $-Y$.

$$\Pi_2(A) = \text{tree obtained by changing } \boxed{XY} \rightarrow \boxed{YY}$$

$$\boxed{-Y} \rightarrow \boxed{YY}$$

and removing $X-$.
Let $A$ be a supertree,

$\Pi_1(A) =$ tree obtained by changing

$XY \rightarrow X \times$

$X- \rightarrow X$

and removing $-Y$.

$\Pi_2(A) =$ tree obtained by changing

$XY \rightarrow Y$

$-Y \rightarrow Y$

and removing $X-$.

$\Pi_1(A)$

$\Pi_2(A)$
Given two trees $S$ and $T$, a supertree $A$ defines an alignment between $S$ and $T$ if $TT_1(A) = S$ and $TT_2(A) = T$. 
Given two trees $S$ and $T$, a supertree $A$ defines an alignment between $S$ and $T$ if $TT_1(A) = S$ and $TT_2(A) = T$.

$cost(A) = \text{nb of insertions} + \text{deletions} + \text{mismatches}$ 
(can be changed for more complicated models)
Connection with sequence alignments

Tree alignments generalize sequence alignments.

AUUCG AUUA ... ACCAU UGAUUA ...

alignment:

alignment:
AAU - - C U - - C C - A - - U GG AA UU ...
Which alignment between A, C, C, U and A, G, U, A is the most likely?
Which alignment between and is the most likely?

Probability of an alignment $A$:

$$\alpha \propto e^{-\frac{\text{cost}(A)}{k}}$$

(Gibbs-Boltzmann distribution)
Which alignment between and is the most likely?

![Diagrams showing different alignments]

**Probability of an alignment A**

\[
\alpha = \frac{-\text{cost}(A)}{\kappa}
\]

(Gibbs-Boltzmann distribution)
Why finding one optimal alignment may be inadequate:

- Co-optimal alignments can be very different. (see for instance [Vingron, Argos, 1990])

- Exploring the space of alignments enables the detection of high probability features.
Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

Probability of an alignment $A$:
$$\alpha \propto e^{-\frac{\text{cost}(A)}{K}}$$
(Gibbs-Boltzmann distribution)

Co-optimal alignments
**Objective:** Sampling alignments under the Gibbs-Boltzmann probability distribution.

\[
\alpha \in \frac{-\text{cost}(A)}{K} \quad \text{(Gibbs-Boltzmann distribution)}
\]
AMBIGUITY OF ALIGNMENTS

The two supertrees

induce the same alignment between the trees
AMBIGUITY OF ALIGNMENTS

The two supertrees

induce the same alignment between the trees
AMBIGUITY OF ALIGNMENTS

The two supertrees induce the same alignment between the trees.

They are the same!
AMBIGUITY OF ALIGNMENTS

The two supertrees

\[
\begin{array}{ccc}
\text{AA} & \text{CG} & \text{UA} \\
\text{CU} & \text{C} & \text{UA} \\
\text{C} & \text{U} & \text{G} \\
\end{array}
\]

\[\neq\]

\[
\begin{array}{ccc}
\text{AA} & \text{CG} & \text{UA} \\
\text{CU} & \text{C} & \text{UA} \\
\text{C} & \text{U} & \text{G} \\
\end{array}
\]

do not induce the same alignment between the trees
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow \begin{cases} \ v \ & | \\ x \ y \ & | \\ x \ - \ & | \\ - \ y \ & | \end{cases} \]

\[ F \leftarrow \begin{cases} \ e \ & | \\ x \ y \ & | \\ x \ - \ & | \\ - \ y \ & | \end{cases} \]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[
\begin{align*}
E & \leftarrow X \ Y | X - | - Y \\
F & \leftarrow \varepsilon | X \ Y | X - | - Y \\
\end{align*}
\]

Ex):

\[
e
\]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow F \]

\[ F \leftarrow \varepsilon \]

Ex: \[ e \]

[Jiang, Wang, Zhang]
**A Grammar for Alignments**

**Strategy:** Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ E \rightarrow X \cdot Y \quad | \quad X \cdot - \quad | \quad - \cdot Y \]

\[ F \rightarrow \varepsilon \quad | \quad X \cdot Y \quad | \quad X \cdot - \quad | \quad - \cdot Y \]

**Ex:**

\[ AA \rightarrow F \]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[
\begin{align*}
\mathcal{G} & \leftarrow \ \text{X} \text{Y} \ F \ F \ F \\
F & \leftarrow \varepsilon \ F \ F \ F \\
E & \leftarrow \AA \ F
\end{align*}
\]

[Jiang, Wang, Zhang]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow F \quad | \quad F \quad | \quad F \]

\[ F \leftarrow \varepsilon \quad | \quad F \quad | \quad F \quad | \quad F \]

Ex:
Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow \begin{align*}
  F \quad | \quad F \quad | \quad F
\end{align*} \]

\[ F \leftarrow \epsilon \quad | \quad F \quad | \quad F \quad | \quad F \]

Ex:
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

Ex:

[Reference: Jiang, Wang, Zhang]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \to F \mid F \mid F \]

\[ F \to \epsilon \mid F \mid F \mid F \]

Ex:

[Diagram of a tree structure with labels]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[
G \leftarrow \begin{cases} 
  XY & | \\
  F & | \\
  F & | \\
  F & |
\end{cases}
\]

\[
F \leftarrow \begin{cases} 
  \epsilon & | \\
  XY & | \\
  F & | \\
  X- & | \\
  F & | \\
  -Y & | \\
  F & |
\end{cases}
\]

Ex:

[Jiang, Wang, Zhang]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

Ex:
A grammar for alignments

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

Ex:
Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

Ex:
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow \begin{array}{c}
XY \\
F \\
F
\end{array} \quad \begin{array}{c}
x- \\
F \\
F
\end{array} \quad \begin{array}{c}
y- \\
F \\
F
\end{array} \]

\[ F \leftarrow \varepsilon \left| \begin{array}{c}
XY \\
F \\
F
\end{array} \quad \begin{array}{c}
x- \\
F \\
F
\end{array} \quad \begin{array}{c}
y- \\
F \\
F
\end{array} \]

Ex:

\[
\begin{array}{c}
AA \\
C- \\
-\text{C-} \\
-U \text{C-} \text{UA}
\end{array}
\]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow \begin{align*}
    F \quad | \quad F \\
\end{align*} \]

\[ F \leftarrow \begin{align*}
    F \quad | \quad F \\
\end{align*} \]

Ex:

\[ \begin{align*}
    &AA \\
    &\quad \begin{align*}
    &\text{C-} \\
    &\quad \begin{align*}
    &\text{U} \\
    &\quad \text{C-}
    \end{align*} \\
    &\quad \text{UA}
\end{align*} \]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \rightarrow X Y \mid x - \mid - y \]
\[ F \rightarrow \varepsilon \mid X Y \mid x - \mid - y \]

Ex:

[Diagram showing examples of alignments with corresponding grammatical representation.]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow \begin{array}{c|c|c}
XY & F & F \\
\hline
x- & F & F \\
\hline
-y & F & F \\
\end{array} \]

\[ F \leftarrow \begin{array}{c|c|c}
xy & F & F \\
\hline
x- & F & F \\
\hline
-y & F & F \\
\end{array} \]

Ex:

\[ \begin{array}{c}
\text{A-A} \\
\hline
\text{E-} \quad \text{E-} \\
\hline
\text{-U} \quad \text{E-} \quad \text{U-A} \\
\end{array} \]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow \begin{cases} XY \mid F \mid X- \mid F \mid -Y \mid F \end{cases} \]

\[ F \leftarrow \varepsilon \mid \begin{cases} XY \mid F \mid X- \mid F \mid -Y \mid F \end{cases} \]

Ex:

[Diagram showing examples of alignments]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

$$G \leftarrow \begin{cases} \text{XX} & | \text{XY} & | \text{XY} \\ \text{F} & | \text{F} & | \text{F} \end{cases}$$

$$F \leftarrow \begin{cases} \varepsilon & | \text{XX} & | \text{XY} & | \text{XY} \\ \text{F} & | \text{F} & | \text{F} & | \text{F} \end{cases}$$

Ex:

- $\begin{cases} \text{AA} & | \text{X-} & | \text{G} \\ \text{C-} & | \text{U} & | \text{C-} & | \text{U} \end{cases}$
- $\begin{cases} \text{C-} & | \text{G} & | \text{U} & | \text{C-} & | \text{U} \\ \text{F} & | \text{F} & \end{cases}$

[Jiang, Wang, Zhang]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow X \rightarrow Y \quad | \quad X \rightarrow \epsilon \quad | \quad X \rightarrow \cdot \]

Ex:

\[
\begin{align*}
G & \rightarrow AA \\
A & \rightarrow C - G \\
 & \rightarrow - U C - UA \\
C & \rightarrow C - UA \\
ge & \rightarrow - U 
\end{align*}
\]

[Jiang, Wang, Zhang]
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once.

An example of grammar that does not work:

\[ G \leftarrow \begin{align*}
F & \quad | \quad F \\
F & \leftarrow \varepsilon \\
\end{align*} \]

Ex:

- Ambiguous grammar
A GRAMMAR FOR ALIGNMENTS

**Theorem**: The set $A^*$ generated by the following grammar contains every tree alignment exactly once.
A GRAMMAR FOR ALIGNMENTS

Our (complicated) non-ambiguous grammar:

\[ \text{A} \leftarrow \text{\(\phi\)} \mid \text{C}_I \mid \text{C}_D \mid \text{F}_\text{I} \mid \text{F}_\text{D} \]

\[ \text{\(\phi\)} \leftarrow \text{\(\uparrow\)} \mid \text{F}_\text{AE} \]

\[ \text{C}_I \leftarrow \text{F}_\text{E} \]

\[ \text{F}_\text{I} \leftarrow \text{\(\varepsilon\)} \mid \text{F}_\text{E} \]

\[ \text{C}_D \leftarrow \text{F}_\text{D} \]

\[ \text{F}_\text{D} \leftarrow \text{\(\varepsilon\)} \mid \text{F}_\text{E} \]

\[ \text{\(\uparrow\)} \leftarrow \text{B}_{\text{ID}, \text{D}} \mid \text{F}_\text{D} \text{\(\uparrow\)} \text{F}_\text{D} \]

\[ \text{F}_\text{AE} \leftarrow \text{F}_\text{E} \mid \text{\(\uparrow\)} \mid \text{\(\uparrow\)} \]

For \( \text{J} \in \{\text{I}, \text{D}\}, (\text{M}, \text{M}') \in \{\text{\(\phi\)}, \Rightarrow, \leftrightarrow\}^2 \):

\[ \text{B}_{\text{ID}, \text{M}, \text{M}'} \leftarrow \text{\(\varepsilon\)} \mid \text{C}_I \text{B}_{\text{ID}, \text{M}, \text{M}'} \mid \text{C}_D \text{B}_{\text{ID}, \text{M}, \text{M}'} \mid \text{\(\uparrow\)} \text{B}_{\text{ID}, \text{M}, \text{M}'} \mid \text{\(\uparrow\)} \text{B}_{\text{ID}, \text{M}, \text{M}'} \mid \text{\(\uparrow\)} \text{B}_{\text{ID}, \text{M}, \text{M}'} \mid \text{\(\uparrow\)} \text{B}_{\text{ID}, \text{M}, \text{M}'} \mid \text{\(\uparrow\)} \text{B}_{\text{ID}, \text{M}, \text{M}'} \mid \text{\(\uparrow\)} \text{B}_{\text{ID}, \text{M}, \text{M}'} \mid \text{\(\uparrow\)} \text{B}_{\text{ID}, \text{M}, \text{M}'} \]

only if \( \text{J} \neq \text{D} \)
only if \( \text{\(\uparrow\)} \neq \text{\(\uparrow\)} \)
only if \( \text{M} \neq \text{\(\uparrow\)} \)

(no room for \( \text{B}_{\text{ID}, \text{M}, \text{M}'} \))
APPLICATION 1: COUNTING

\[ a_n = \text{number of tree alignments of size } n \]

Generating function:

\[ A(z) = \sum_{n \geq 0} a_n z^n \]
APPLICATION 1: COUNTING.

\[ a_n = \text{number of tree alignments of size } n \]

Generating function: \[ A(z) = \sum_{n \geq 0} a_n z^n \]

The principle on Jiang et al.'s grammar:

[Diagram of grammar rules]

\[ F \leftarrow E \]

\[ F \leftarrow E \]
APPLICATION 1: COUNTING

\[ a_n = \text{number of tree alignments of size } n \]

Generating function: \[ A(q) = \sum_{n \geq 0} a_n q^n \]

The principle on Jiang et al.'s grammar:

\[ E \leftarrow X Y \quad | \quad X - \quad | \quad - Y \]

\[ T(q) = F(q) + F(q) + F(q) \]

\[ F \leftarrow E \quad | \quad X Y \quad | \quad X - \quad | \quad - Y \]

\[ F(q) = 1 + F(q) \times F(q) + F(q) \times F(q) + F(q) \times F(q) \]
APPLICATION 1: COUNTING.

\[ a_m = \text{number of tree alignments of size } n \]

Generating function: \[ A(z) = \sum_{n \geq 0} a_m z^n \]

The principle on Jiang et al.'s grammar:

\[ G \leftarrow \begin{array}{c}
\begin{array}{c}
\text{XY} \\
\text{F}
\end{array} & \begin{array}{c}
\text{X} \quad - \\
\text{F}
\end{array} & \begin{array}{c}
\text{X} \quad - \\
\text{F}
\end{array}
\end{array} \]

\[ T(z) = F(z) + F(z) + F(z) \]

\[ F \leftarrow \varepsilon | \begin{array}{c}
\begin{array}{c}
\text{XY} \\
\text{F}
\end{array} & \begin{array}{c}
\text{F}
\end{array} & \begin{array}{c}
\text{XY} \\
\text{F}
\end{array} & \begin{array}{c}
\text{F}
\end{array}
\end{array} \]

\[ F(z) = 1 + F(z)^2 + F(z)^2 + F(z)^2 \]
APPLICATION 1: COUNTING

\[ a_n = \text{number of tree alignments of size } n \]

Generating function: \[ A(z) = \sum_{n \geq 0} a_n z^n \]

The principle on Jiang et al.'s grammar:

\[ C \leftarrow \begin{array}{c} F \end{array} \quad \begin{array}{c} X \end{array} \quad \begin{array}{c} Y \end{array} \]

\[ T(z) = F(z) + \frac{Xz}{1 + F(z)} + \frac{Yz}{1 + F(z)} \]

\[ F \leftarrow \begin{array}{c} E \end{array} \quad \begin{array}{c} X \end{array} \quad \begin{array}{c} Y \end{array} \]

\[ F(z) = 1 + F(z)^2 + \frac{Xz}{1 + F(z)^2} + \frac{Yz}{1 + F(z)^2} \]
APPLICATION 1: COUNTING

\[ a_m = \text{number of tree alignments of size } n \]

Generating function: \[ A(z) = \sum_{n \geq 0} a_n z^n \]

The principle on Jiang et al.'s grammar:

\[ G \leftarrow X \]

\[ T(z) = z^2 F(z) + z F(z) + z^2 F(z) \]

\[ F \leftarrow E \]

\[ F(z) = 1 + z^2 F(z)^2 + 2 z F(z)^2 + 2 z F(z)^2 \]
APPLICATION 1: COUNTING

\[ a_{m,k} = \text{number of tree alignments of size } n \text{ and } k \text{ matches} \]

Generating function: \[ A(z, u) = \sum_{n \geq 0} a_{m,k} z^m u^k \]

The principle on Jiang et al.'s grammar:

\[ T(z^2) = z^2 F(z) + z^2 F(z) + z^2 F(z) \]

\[ F(z) = 1 + z^2 F(z)^2 + z^2 F(z)^2 + z^2 F(z)^2 \]
**Theorem:** The generating function $A(z, \mu)$ of tree alignments satisfies

$$A(z, \mu) = \left( z^2 + z - \mu z^2 + \frac{z^2}{1 - 4z} \right) \cdot B(z, \mu)$$

where

$$(\mu z C(z)^2 - z C(z)^2 + 2z)B(z, \mu)^2 + (z^2 C^4(z) - 2z C(z)^2 - 1)B(z, \mu) + C(z) = 0$$

and

$$C(z) = \frac{1 - \sqrt{1 - 4z}}{2z} \quad \text{Catalan generating function}$$
SOME STATISTICAL PROPERTIES

Theorem: There are on average \( C \times 1.5^n \) alignments between two random trees of cumulative size \( n \)

where \( C = 0.299 \ldots \)

Corollary: A same alignment was repeated \( \sim 0.875 \times 1.412^n \) times on average in the previous ambiguous grammar.
Application 2 - Sampling

Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

\[
\text{probability of an alignment } A = \frac{-\text{cost}(A)}{K} \\
(\text{Gibbs-Boltzmann distribution})
\]
APPLICATION 2 - SAMPLING

Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

Strategy:
- Filter the grammar to obtain a new grammar that only generates alignments between two fixed trees S and T
- Use dynamic programming.
The principle on Jiang et al.'s grammar:

\[ G \leftarrow F \mid X \gamma \mid X \gamma \]

\[ F \leftarrow \varepsilon \mid F \mid F \mid F \]
The principle on Jiang et al.'s grammar:

We fix two trees $S$ and $T$.

Let $F$ be a subforest of $S$ and $G$ a subforest of $T$.

$$J[F, G] = \{ \text{alignments between } F \text{ and } G \}$$
The principle on Jiang et al.'s grammar:

We fix two trees $S$ and $T$.

Let $F$ be a subforest of $S$ and $G$ a subforest of $T$.

$J[F,G] = \{\text{alignments between } F \text{ and } G\}$

If $F = \emptyset$ and $G = \emptyset$, then

$J[F,G] \leftarrow \varepsilon$
The principle on Jiang et al.'s grammar:

We fix two trees $S$ and $T$.
Let $F$ be a subforest of $S$ and $G$ a subforest of $T$.

$J[F, G] = \sum$ alignments between $F$ and $G$

$G \leftarrow \begin{array}{c}
\begin{array}{c}
\times \times \\
\times \times \\
\end{array}
\begin{array}{c}
F \\
F \\
F \\
F \\
F \\
F \\
\end{array}
\end{array}$

$F \leftarrow E \mid \begin{array}{c}
\begin{array}{c}
\times \times \\
\times \times \\
\end{array}
\begin{array}{c}
F \\
F \\
F \\
F \\
F \\
F \\
\end{array}
\end{array}$

If $F = F_1 F_2$ and $G = \emptyset$, then

$J[F, G] \leftarrow \begin{array}{c}
\begin{array}{c}
\times \times \\
\times \times \\
\end{array}
\begin{array}{c}
J[F_1, \emptyset] \\
J[F_2, \emptyset] \\
J[F_1, \emptyset] \\
J[F_2, \emptyset] \\
J[F_1, \emptyset] \\
J[F_2, \emptyset] \\
\end{array}
\end{array}$
The principle on Jiang et al.'s grammar:

We fix two trees $S$ and $T$.

Let $F$ be a subforest of $S$ and $G$ a subforest of $T$.

$$J[F, G] = \# \text{alignments between } F \text{ and } G$$

If $F = \emptyset$ and $G = \begin{array}{c} \mathbb{G}_1 \\ \mathbb{G}_2 \end{array}$, then

$$J[F, G] \leftarrow \begin{array}{c} -y \end{array} \begin{array}{c} J[\emptyset, \mathbb{G}_1] \\ \mathbb{G}_2 \end{array}$$
The principle on Jiang et al.'s grammar:

We fix two trees $S$ and $T$.

Let $F$ be a subforest of $S$ and $G$ a subforest of $T$.

$$J[F, G] = \{\text{alignments between } F \text{ and } G\}$$

If $F = F_1 \circ F_2$ and $G = G_1 \circ G_2$, then

$$J[F, G] = J[F_1, G_1] \oplus J[F_2, G_2] \oplus J[F_1, G_2] \oplus J[F_2, G_1]$$
**Grammar of Alignments Between Two Fixed Trees**

- If $F = \emptyset$ and $G = \emptyset$, then $J[F,G] \leftarrow \epsilon$

- If $F = \overbrace{F_1}^{x_1}$ and $G = \emptyset$, then

  $J[F,G] \leftarrow J[F_1,\emptyset]$  
  $J[F_1,\emptyset] \leftarrow J[F_2,\emptyset]$  
  $J[F_2,\emptyset] \leftarrow J[F_3,\emptyset]$  
  $J[F_3,\emptyset] \leftarrow J[F_4,\emptyset]$  

- If $F = \emptyset$ and $G = \overbrace{G_1}^{x_1} G_2$, then

  $J[F,G] \leftarrow J[\emptyset,G_1] J[\emptyset,G_2]$  
  $J[\emptyset,G_1] \leftarrow J[\emptyset,F_1]$  
  $J[\emptyset,G_2] \leftarrow J[\emptyset,F_2]$  

- If $F = \overbrace{F_1}^{x_1} F_2$ and $G = \overbrace{G_1}^{x_1} G_2$, then

  $J[F,G] \leftarrow J[F_1,G_1] J[F_2,G_2]$  
  $J[F_1,G_1] \leftarrow J[F_3,G_1]$  
  $J[F_2,G_2] \leftarrow J[F_3,G_2]$  
  $J[F_3,G_1] \leftarrow J[F_2,F_3]$  
  $J[F_3,G_2] \leftarrow J[F_2,F_3]$  
  $J[F_2,F_3] \leftarrow J[F_2,F_3]$  

GIBBS-BOLTZMANN DISTRIBUTION

\[ P(\text{alignment } A \text{ between } S \text{ and } T) = \frac{e^{-\frac{\text{cost}(A)}{K}}}{Z_{S,T}} \]

\[ Z_{S,T} = \sum_{A' \text{ alignment between } S \text{ and } T} e^{-\frac{\text{cost}(A')}{K}} \]
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

- If $F = \emptyset$ and $G = \emptyset$, then $J[F, G] \leftarrow \varepsilon$

- If $F = F_1 F_2$ and $G = \emptyset$, then $J[F, G] \leftarrow J[F_2, \emptyset]
  \quad J[F_1, \emptyset]

- If $F = \emptyset$ and $G = G_1 G_2$, then $J[F, G] \leftarrow J[\emptyset, G_2]
  \quad J[F, \emptyset]

- If $F = F_1 F_2$ and $G = G_1 G_2$, then $J[F, G] \leftarrow J[F_2 G_1]
  \quad J[F_1 G_2]
  \quad J[F, G_1]
  \quad J[F_2, G_2]
  \quad J[F, \emptyset]
  \quad G = G' G''$
PARTITION FUNCTION

- If $F = \emptyset$ and $G = \emptyset$, then $Z_{F,G} \leftarrow 0$

- If $F = \begin{array}{|c|c|}
\hline
F_1 & F_2 \\
\hline
\end{array}$ and $G = \emptyset$, then

$$Z_{F,G} \leftarrow e^{-\frac{1}{K}} \times Z_{F_1,\emptyset} \times Z_{F_2,\emptyset}$$

- If $F = \emptyset$ and $G = \begin{array}{|c|c|}
\hline
G_1 & G_2 \\
\hline
\end{array}$, then

$$Z_{F,G} \leftarrow e^{-\frac{1}{K}} \times Z_{\emptyset,G_1} \times Z_{\emptyset,G_2}$$

- If $F = \begin{array}{|c|c|}
\hline
F_1 & F_2 \\
\hline
\end{array}$ and $G = \begin{array}{|c|c|}
\hline
G_1 & G_2 \\
\hline
\end{array}$, then

$$Z_{F,G} \leftarrow Z_{F_1,G_1} \times Z_{F_2,G_2} + \sum_{G = G_{G'}G''} e^{-\frac{1}{K}} Z_{F_1,G'} \times Z_{F_2,G''} + \sum_{F = F'F''} e^{-\frac{1}{K}} Z_{F,G_1} \times Z_{F,G_2}$$
Theorem Let $S$ and $T$ be two trees of size $n_1$ and $n_2$. Sampling alignments between $S$ and $T$ under the Gibbs-Boltzmann distribution can be done with worst-case time and space complexities $O(n_1 n_2 (n_1 + n_2)^2)$ and with average-case time and space complexities $O(n_1 n_2)$. 
Theorem Let $S$ and $T$ be two trees of size $n_1$ and $n_2$. Sampling alignments between $S$ and $T$ under the Gibbs-Boltzmann distribution can be done with worst-case time and space complexities $O(n_1 n_2 (n_1 + n_2)^2)$ and with average-case time and space complexities $O(n_1 n_2)$.

Proof inspired by [Herrbach, Denise, Dulucq]
CONCLUSION

→ We are using our grammar and adapted dynamic programming algorithms to revisit the 3D alignments of RNA structures.

→ Simpler decomposition for tree alignments?