GENERATING TREE ALIGNMENTS
HOW COMBINATORICS CAN HELP BIOINFORMATICS

Julien COURTIEL (PIMS/Univ. of British Columbia, Vancouver)
2016 SFU Symposium on Mathematics and Computation

Co-authors: Cedric CHAUVE (Simon Fraser University, Vancouver)
Yann PONTY (CNRS/LIX, Ecole Polytechnique, Inria Saclay)
WHAT IS RNA?

DNA
the code

ACCTAGGT
TGGAATTCGAA
WHAT IS RNA?

DNA
the code

RNA

ACCTAGGTT
TGGATTCAC
WHAT IS RNA?

DNA → the code → RNA

ACCTAGGTA
TGGATTCACA

Pol

BETTER CALL POL
WHAT IS RNA?

DNA

the code

RNA

ACCTAGGT

P彰

U-CAA
WHAT IS RNA?

DNA → RNA

the code

RNA

\[\text{ACCTAGGT}\]

\[\text{UG}

\text{Pol}
WHAT IS RNA?

DNA → the code → RNA

ACCTAGGT

Pol

UGGG
WHAT IS RNA?

DNA
the code
↓
RNA

ACCTAGGT
UGGAUCCC

Pol
WHAT IS RNA?

DNA
the code
↓
RNA

ACCTAGGTT
TGAGATCCCA
UGGAUCCC...

Pol
WHAT IS RNA?

DNA
the code

RNA

proteins

- DNA: ACGCTAGGGT
  - RNA: UGGAGAUCUC...
WHAT IS RNA?

DNA
the code

RNA

proteins

Ribosomal RNA (Ribo)
WHAT IS RNA?

DNA  RNA  proteins
the code

---

\[\text{ACCTAGGGT}\]  \[\text{UGGGAUCCC}\]  \[\text{Trp}\]
WHAT IS RNA?

DNA

the code

RNA

proteins

Trp

Gly

ACCTAGGT

UGGGAUCCC...

Ribosome
WHAT IS RNA?

DNA

the code

RNA

proteins
WHAT IS RNA?

DNA
the code

RNA

proteins

Trp Gly Asp Ile Ser
Classic dogma

DNA
the code

↓

RNA
the messenger

↓

proteins
the machine

WHAT IS RNA?

A C C T A G G T

T G G A T T C C A

U G G A U U C C

...
Classic dogma

DNA
the code

RNA
the messenger?

proteins
the machine

WHAT IS RNA?

\[\text{ACCGTAGGCT} \rightarrow \text{UGGGAUCC}\]

BOOORING

\[\text{Trp-Gly-Asp-Ile-Ser} \ldots\]
**WHAT IS RNA?**

- **DNA**
  - the code

  ↓

- **RNA**
  - the messenger

  ↓

- **proteins**
  - the machine

DNA sequence: ACCCTAGGGT

RNA sequence: UGGGAAUCC

- Actually also...
  - translator, enzyme, catalyst...

  ↑

  recent breakthrough

Protein sequence: Trp-Gly-Asp-Ile-Ser...
WHAT IS RNA?

RNA
the messenger

U-G-G-A-U-C-C...

actually also... translator, regulator, enzyme, catalyst...
So what is RNA?

RNA is a single-stranded molecule (chain of nucleotides)...

U-G-G-A-U-C-C-...
So what is RNA?

RNA is

a single-stranded molecule
(chain of nucleotides)...

...stabilized by hydrogen bonds...
So what is RNA?

RNA is

a single-stranded molecule (chain of nucleotides)...

...stabilized by hydrogen bonds...

...which folds onto itself.
So what is RNA?

RNA is

- a single-stranded molecule (chain of nucleotides)...
- ...stabilized by hydrogen bonds...
- ...which folds onto itself.

UGAUCUC...

primary structure

secondary structure

tertiary structure
**RNA COMPARISON**

**Interesting problem**: evaluating similarity between two RNAs.
RNA COMPARISON

Interesting problem: evaluating similarity between two RNAs.

😊? — Why?
RNA COMPARISON

Interesting problem: evaluating similarity between two RNAs.

Why?

Typical situation:

New RNA \[\rightarrow\] ACAGUACC...

large database

AUCCAG...
UUAGACC...
CCAGC...
AAAGU...
**RNA COMPARISON**

**Interesting problem:** evaluating similarity between two RNAs.

- 🤔 Why?

**Typical situation:**

New RNA → ACAGUACC ...

- Finding similar RNAs

- Large database
  - AUCCAG ...
  - CCAGC ...
  - UUAGACC ...
  - AAAGU ...
**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 the primary structures.

RNA 1: AUUUCGAAUA...  
RNA 2: ACCAUGAAUA...
Motivation: RNA Comparison

Question: how to measure similarity between two RNAs?

First idea: compare the primary structures $\rightarrow$ sequence alignment

RNA 1: AUUUCGAUUAA ...
RNA 2: ACCAUGAUAUA ...

**MOTIVATION: RNA COMPARISON**

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

Second idea: compare secondary structures.

→ 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

Objective: Align trees coming from RNA 2^ary structures
FROM SECONDARY STRUCTURES TO TREES

Objective: Align trees coming from RNA 2^ary structures
**SEQUENCE ALIGNMENT**

**super sequence** = word on $\Sigma \times \Sigma \oplus \Sigma \times \{-\} \oplus \{-\} \times \Sigma$


↑  
match  insertion  deletion  mismatch
**SEQUENCE ALIGNMENT**

**super sequence** = word on $\Sigma \times \Sigma \ominus \Sigma \times \{-\} \ominus \{-\} \times \Sigma$

\[ (A)(U)(C)(G) \quad \text{match} \quad \text{insertion} \quad \text{deletion} \quad \text{mismatch} \quad \text{projections} \]


\[ \text{ACCAUGAUUA} \]
SEQUENCE ALIGNMENT

super sequence = word on $\Sigma \times \Sigma \oplus \Sigma \times \{\text{-}\} \oplus \{\text{-}\} \times \Sigma$

\]

\[\text{match} \quad \text{insertion} \quad \text{deletion} \quad \text{mismatch} \quad \text{projections}\]

Given two sequences $S_1$ and $S_2$, alignment between $S_1$ and $S_2$ = supersequence with projections $S_1$ and $S_2$
**SEQUENCE ALIGNMENT**

super sequence = word on $\Sigma \times \Sigma \oplus \Sigma \times \{-\} \oplus \{-\} \times \Sigma$


$$\text{ACCAUGAUA \uparrow \text{projections}}$$

Given two sequences $S_1$ and $S_2$, alignment between $S_1$ and $S_2 = \text{supersequence with projections } S_1 \text{ and } S_2$

cost of an alignment = nb of insertions + deletions + mismatches
OPTIMAL ALIGNMENT

Classical problem: Given $S_1$ and $S_2$, find one optimal alignment between $S_1$ and $S_2$.

Solvable by Dynamic Programming:
- Needleman-Wunsch algorithm
- Smith-Waterman algorithm
- BLAST (heuristic)

Worst case and average time: $O(n^2)$
Trees are plane, rooted, and vertices are labeled by an alphabet $\Sigma$.

Supertree = tree with 3 types of vertices:
- $X\hat{Y}$ (mis)match
- $X-$ insertion
- $-Y$ deletion
TREE ALIGNMENTS

first projection

second projection
TREE ALIGNMENTS

first projection

A
 C
 C
 U

keep left letters

second projection

AA
 C-
 -G
 -U C-
 UA
TREE ALIGNMENTS

first projection

second projection

keep left letters
TREE ALIGNMENTS

first projection

keep left letters

second projection

keep right letters
TREE ALIGNMENTS

first projection

second projection

keep left letters

keep right letters
Given two trees $S$ and $T$, alignment between $S$ and $T = \text{supertree whose projections are } S \text{ and } T$. 
Given two trees $S$ and $T$, the alignment between $S$ and $T$ is the super-tree whose projections are $S$ and $T$.

The cost of an alignment is the number of insertions + deletions + mismatches.
Connection with sequence alignments.

Tree alignments generalize sequence alignments.

AUUCGAAUUA...

ACCAUGAUUU...

alignment:


alignment:

AA U - C U - C C - A - U G G A A U U...

alignment:
**OPTIMAL ALIGNMENT**

Classical problem: Given $S$ and $T$, find one optimal alignment between $S$ and $T$.

Solvable by Dynamic Programming:
- Worst case time: $O(n^4)$
- Average time: $O(n^2)$

[Jiang, Wang, Zhang] [Herrbach, Denise, Dulucq]
Which alignment between and is the most likely?
Which alignment between A and G is the most likely?
SPACE OF ALIGNMENTS

Why finding one optimal alignment may be inadequate:

- Co-optimal alignments can be very different.
- 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 $A$:
$$\alpha = \frac{e^{-\frac{\text{cost}(A)}{k}}}{K}$$
(Gibbs-Boltzmann distribution)
Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

\[ \text{probability of an alignment } A = \frac{e^{-\frac{\text{cost}(A)}{k}}}{k} \] (Gibbs-Boltzmann distribution)

\[ K=0 \quad \text{Uniform optimal distribution over alignments.} \]
\[ K=+\infty \quad \text{Uniform all distribution over alignments.} \]
**Objective:** Sampling alignments under the Gibbs-Boltzmann probability distribution.

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

(Gibbs-Boltzmann distribution)

\[ K = 0 : \text{Uniform optimal distribution over alignments.} \]

\[ K > 0 : \text{Uniform all distribution over alignments.} \]
AMBIGUITY OF ALIGNMENTS

For sequences,


is the same alignment as

AMBIGUITY OF ALIGNMENTS

For trees,

induce the same alignment between
AMBIGUITY OF ALIGNMENTS

The two supertrees

\[
\begin{align*}
&\text{AA} \\
&\quad \Downarrow \\
&\quad \text{C-} \quad \text{G} \\
&\quad \\
&\quad \text{U} \quad \text{C-} \quad \text{UA}
\end{align*}
\]

\[
\begin{align*}
&\text{AA} \\
&\quad \Downarrow \\
&\quad \text{C-} \quad \text{G} \\
&\quad \\
&\quad \text{CU} \quad \text{UA}
\end{align*}
\]

\[\neq\]

do not induce the same alignment between the trees
For sequences, we can deal with the ambiguity by defining canonical alignments.

PROBLEM RAISED BY THE AMBIGUITY

For sequences, we can deal with the ambiguity by defining canonical alignments.

Ex: \[(A)(u)(u)(-)c(c)\left(-\right)(G)(A)(u)(u)(A)\]

Insertions before Deletions.
PROBLEM RAISED BY THE AMBIGUITY

For sequences, we can deal with the ambiguity by defining canonical alignments.

Ex:

\[(A)^{-}(U)^{-}(U)^{-}(C)^{-}(C)^{-}(A)^{-}(U)^{-}(G)^{-}(A)^{-}(U)^{-}(U)^{-}(A)\]

Insertions before Deletions.

For trees, it is much more complicated!
PROBLEM RAISED BY THE AMBIGUITY

For sequences, we can deal with the ambiguity by defining canonical alignments.


Insertions before Deletions.

For trees, it is much more complicated!

Strategy: COMBINATORICS! 😊
Problem raised by the ambiguity

For sequences, we can deal with the ambiguity by defining canonical alignments.


Insertions before Deletions.

For trees, it is much more complicated!

Strategy: COMBINATORICS! 😊

Build a context-free grammar that generates every alignment exactly once.
GRAMMARS FOR SEQUENCE ALIGNMENTS

Ambiguous grammar:

\[ \gamma \leftarrow (\chi) \ [ \gamma ] \oplus (\chi) \ [ \gamma ] \oplus (\gamma) \ [ \gamma ] \oplus \varepsilon \]

Non-ambiguous grammar:

\[ \gamma \leftarrow (\chi) \ [ \gamma ] \oplus (\chi) \ [ \gamma ] \oplus (\gamma) \ [ \gamma ] \oplus \varepsilon \]

\[ \gamma \leftarrow (\chi) \ [ \gamma ] \oplus (\gamma) \ [ \gamma ] \oplus \varepsilon \]
A GRAMMAR FOR ALIGNMENTS

For trees, an ambiguous grammar can be derived from [Jiang, Wang, Zhang].

Our result:

**Theorem**: The set $A$ generated by the following grammar contains every tree alignment exactly once.
A Grammar for Alignments

Our (complicated) non-ambiguous grammar:

\[ A \rightarrow \Upsilon \phi | \mathcal{G}_I | \mathcal{G}_D | X- \]

\[ \mathcal{G}_I \leftarrow F_I \]

\[ \mathcal{G}_D \leftarrow F_D \]

\[ \Upsilon^\prime \leftarrow \mathcal{G}_I \Upsilon^\prime \mathcal{G}_D \]

\[ \Upsilon \phi \leftarrow \Upsilon^\prime \]

\[ \Upsilon^\prime \leftarrow F_I \Upsilon^\prime F_D \]

\[ \mathcal{G}_D \leftarrow F_\mathcal{D}_2 \]

\[ \Upsilon \mathcal{G}_E \leftarrow F_I \Upsilon \mathcal{G}_E | \Upsilon^\prime F_I \]

For \( J \in \{D, D'\}, (M, M') \in \{\phi, \rightarrow, \epsilon\}^2 \):

\[ \mathcal{G}_I, M, M' \leftarrow \epsilon | \mathcal{G}_I \mathcal{G}_I, M, M' | \mathcal{G}_D \mathcal{G}_D, M, M' | \Upsilon^\prime \mathcal{G}_D, M, M' | X- \mathcal{G}_D, M, M' | \mathcal{G}_I, M, M' \]

\[ \mathcal{G}_D, M, M' \leftarrow \mathcal{G}_I, M, M' \]

\[ \Upsilon \mathcal{G}_E \leftarrow \mathcal{G}_I \Upsilon \mathcal{G}_E | \Upsilon^\prime F_I \]

only if \( J \neq D \)

only if \( \Upsilon \neq \mathcal{D} \)

only if \( M' \neq \epsilon \)

\( (M, M') = (\phi, \phi) \) and \( M \neq \epsilon \)

no room for \( \mathcal{G}_D, M, M' \)
**Theorem** There are on average \( C \times 1.5^n \) alignments between two random trees of cumulative size \( n \) where \( C = 0.289 \ldots \)

**Corollary:** A same alignment was repeated \( \sim 0.875 \times 1.412^n \) times on average in Jiang et al.’s ambiguous grammar.
OBJECTIVE 2 - SAMPLING

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

\[
\text{probability of an alignment } A = \frac{-\text{cost}(A)}{\sum_k} \\
\text{(Gibbs-Boltzmann distribution)}
\]
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.
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 $O(n_1 n_2)$.

**Upsides:**
- No additional complexity cost (except constants, moderate)
- Flexibility of the sampling algorithm
- Already implemented

**Downside**
- Complicated DP scheme
CONCLUSION

“Combinatorics is a powerful tool to solve algorithmic problems.”
"Combinatorics is a powerful tool to solve algorithmic problems."

Open questions:

→ Existence of easier decompositions?

→ Alignment problem for arc-annotated sequences?
TH

A-

-N

-K

Y-

OU!
Theorem: The generating function $A(g, u)$ of tree alignments satisfies

$$A(g, u) = \left( g^2 + 3 - u g^2 + \frac{g}{\sqrt{1-4u}} \right) \times B(g, u)$$

where

$$(u g C(g)^2 - g^2 C(g)^2 + 2g) B(g, u)^2 + (g^2 C^4(g) - 2g C(g)^2 - 1) B(g, u) + C(g) = 0$$

and

$$C(g) = \frac{1 - \sqrt{1-4u}}{2g} \quad \text{Catalan generating function}$$