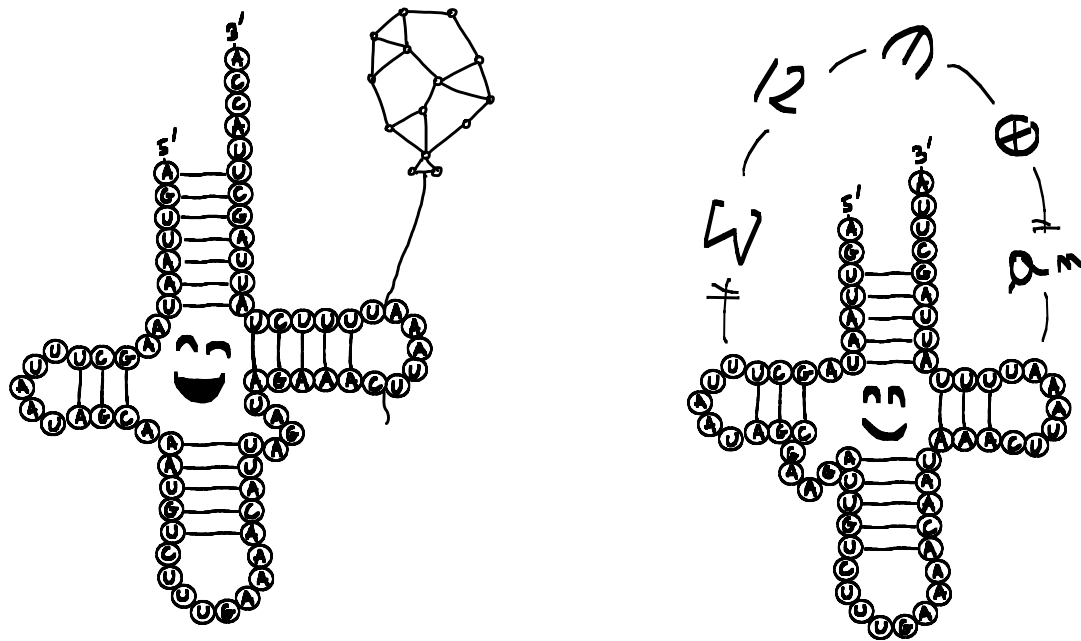


GENERATING TREE ALIGNMENTS

Julien COURTIEL (Université Paris 13)

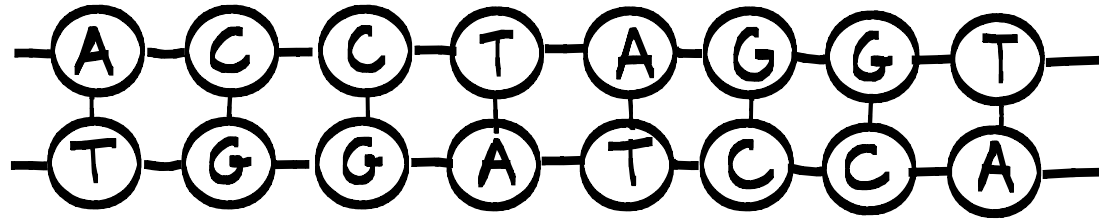
Séminaire CALIN



Co-authors: Cedric CHAUVÉ (Simon Fraser University, Vancouver)
Yann PONTY (CNRS/LIX, Ecole Polytechnique, Inria Saclay)

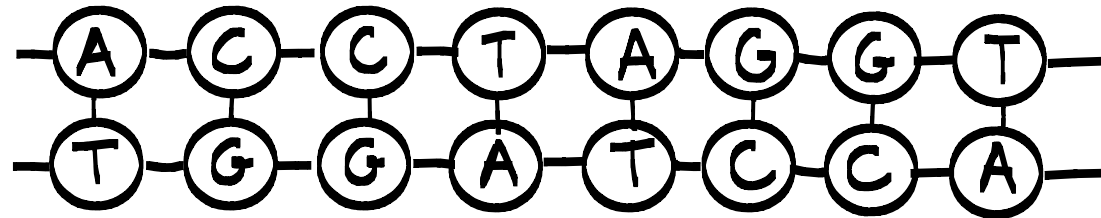
WHAT IS RNA?

DNA
the code



WHAT IS RNA?

DNA
the code

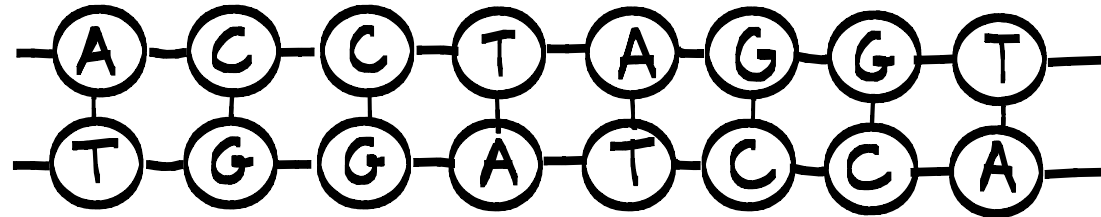


RNA

WHAT IS RNA?

BETTER
CALL POL

DNA
the code



Pol

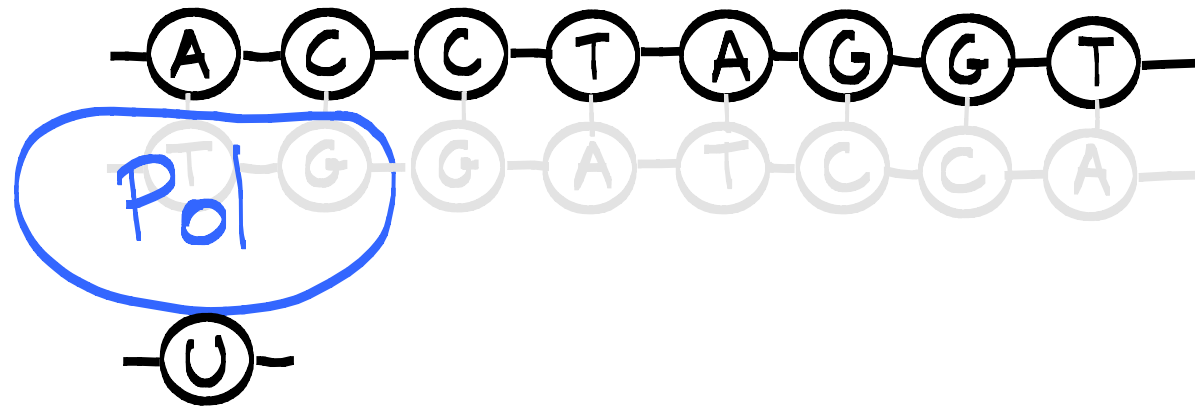
RNA

WHAT IS RNA?

DNA
the code



RNA

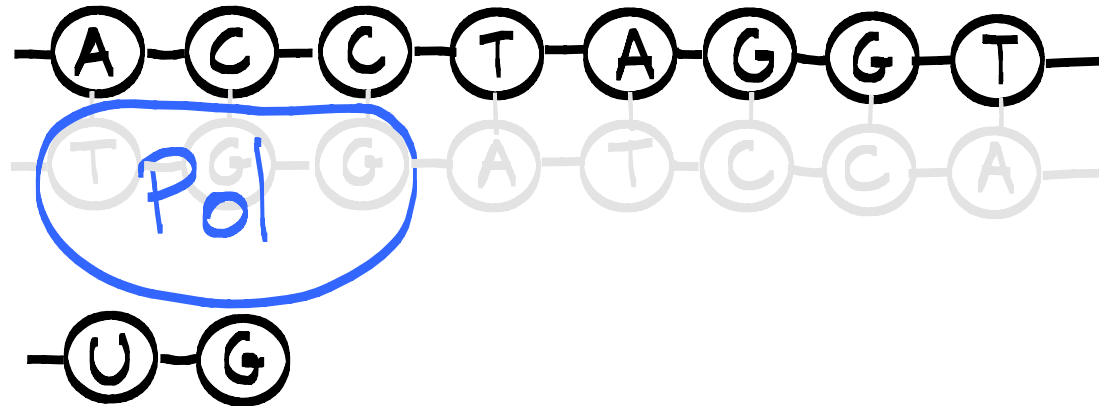


WHAT IS RNA?

DNA
the code



RNA

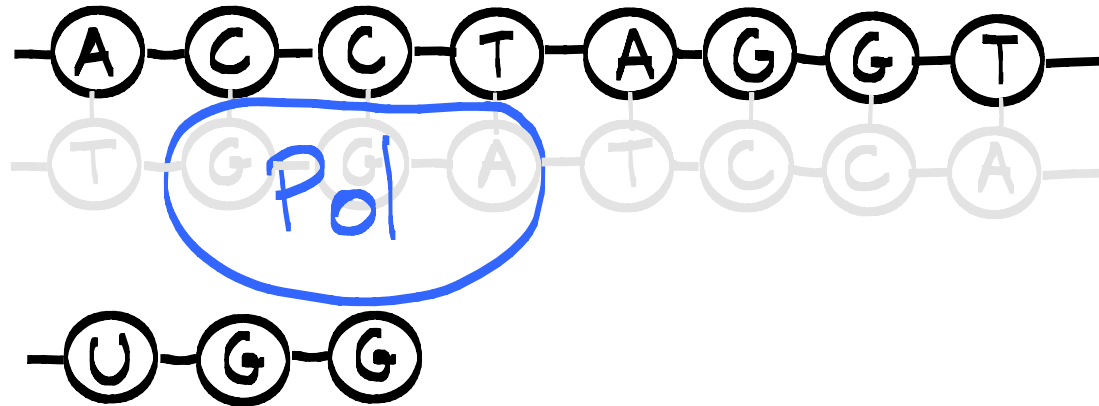


WHAT IS RNA?

DNA
the code



RNA

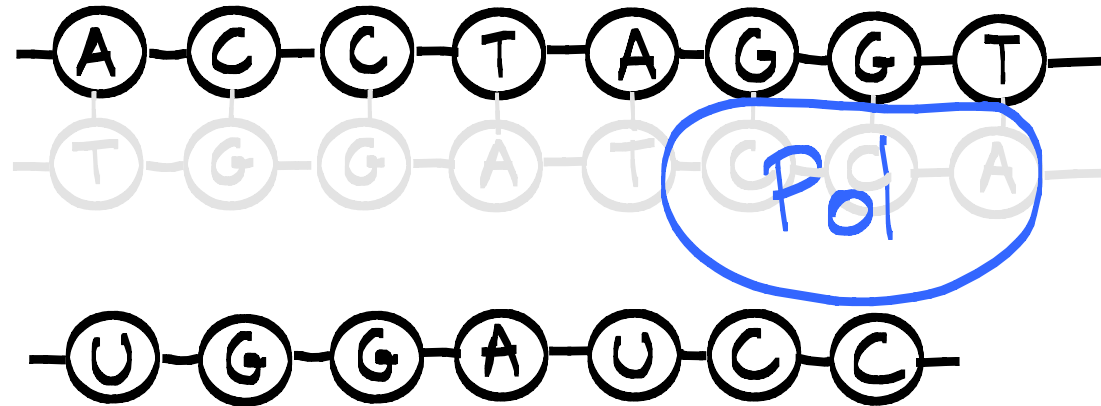


WHAT IS RNA?

DNA
the code



RNA

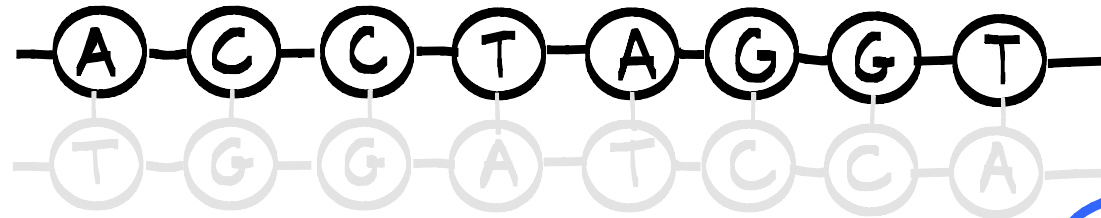


WHAT IS RNA?

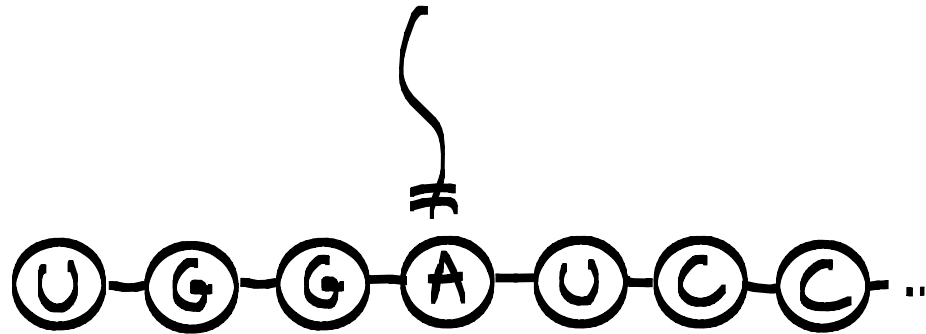
DNA
the code



RNA

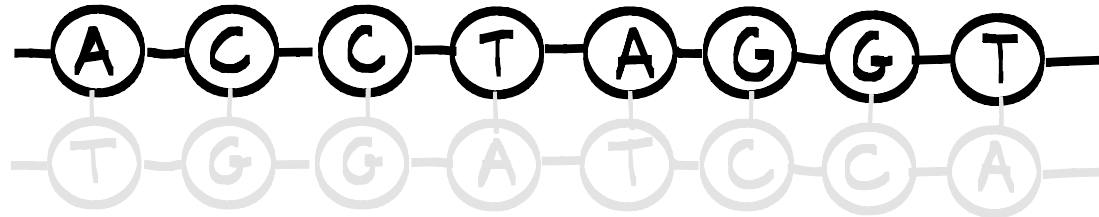


Pol

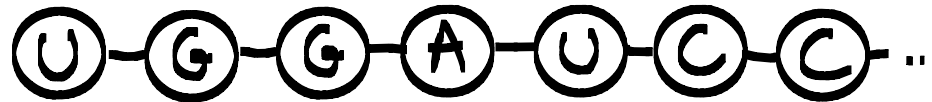


WHAT IS RNA?

DNA
the code



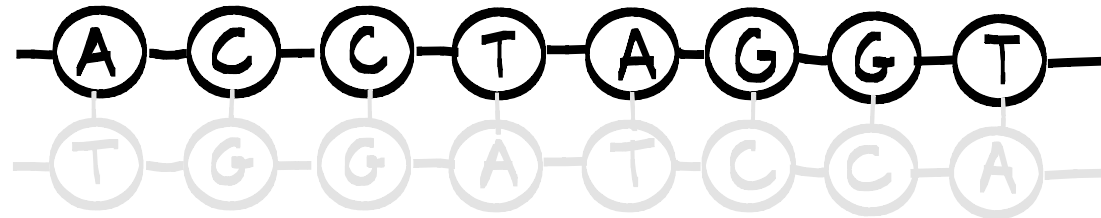
RNA



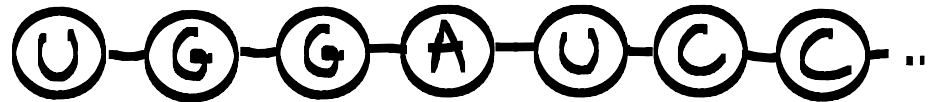
proteins

WHAT IS RNA?

DNA
the code



RNA



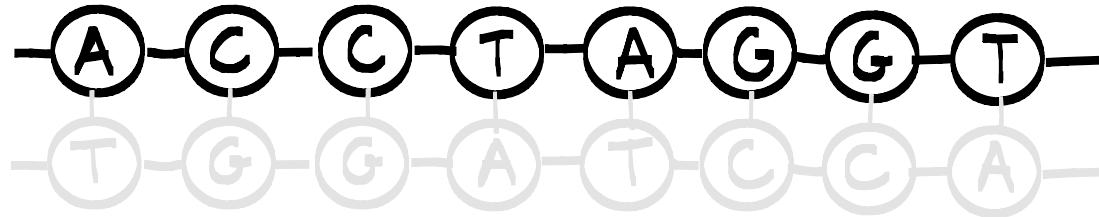
proteins



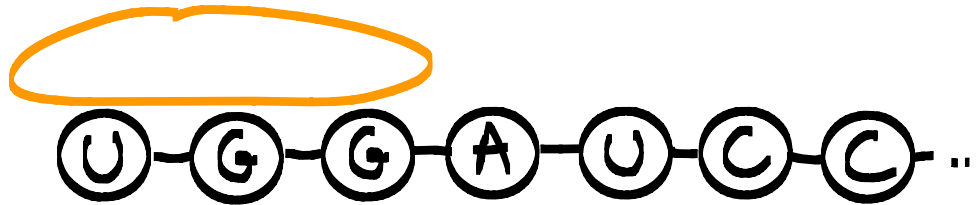
Ribo

WHAT IS RNA?

DNA
the code



RNA



Ribo

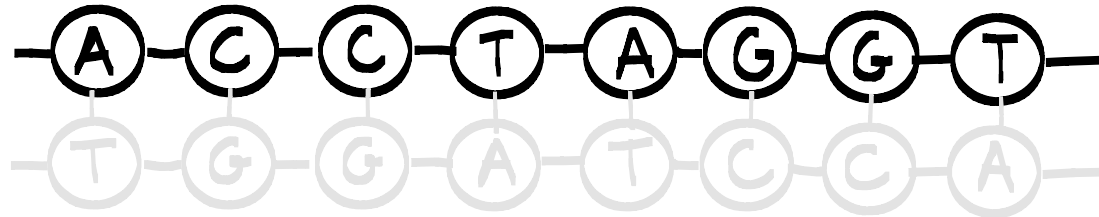
Trp



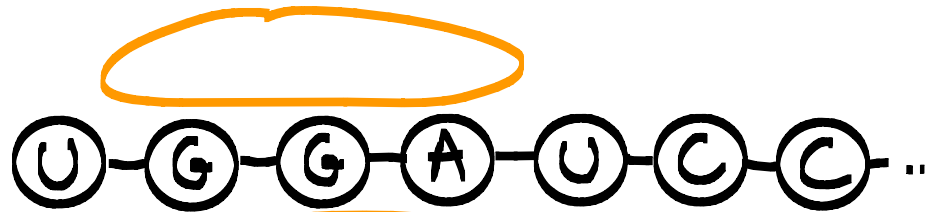
proteins

WHAT IS RNA?

DNA
the code



RNA



Ribo

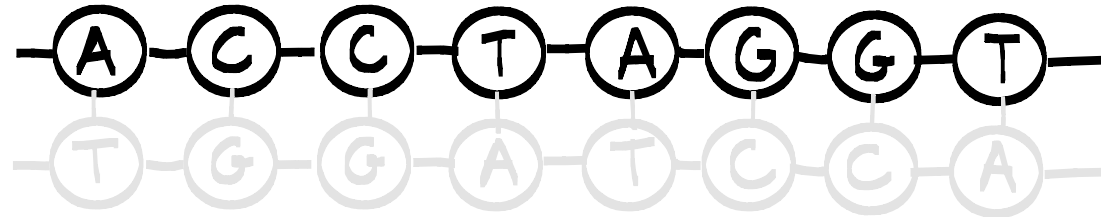


proteins

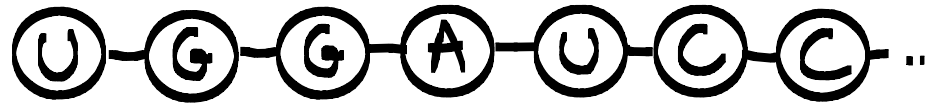


WHAT IS RNA?

DNA
the code



RNA



Ribo

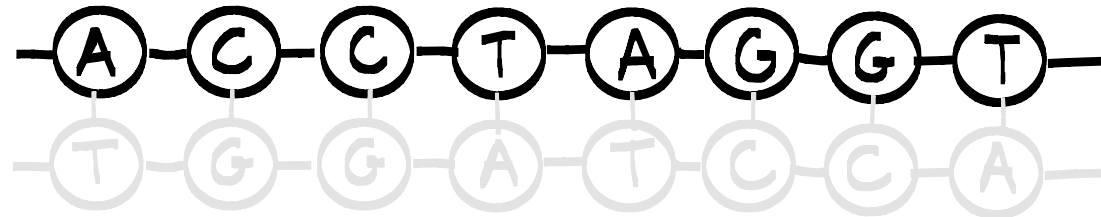


proteins

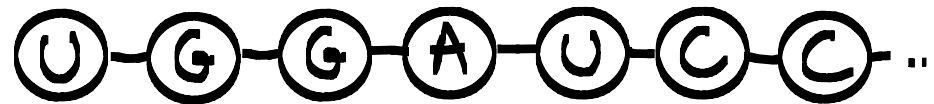


WHAT IS RNA?

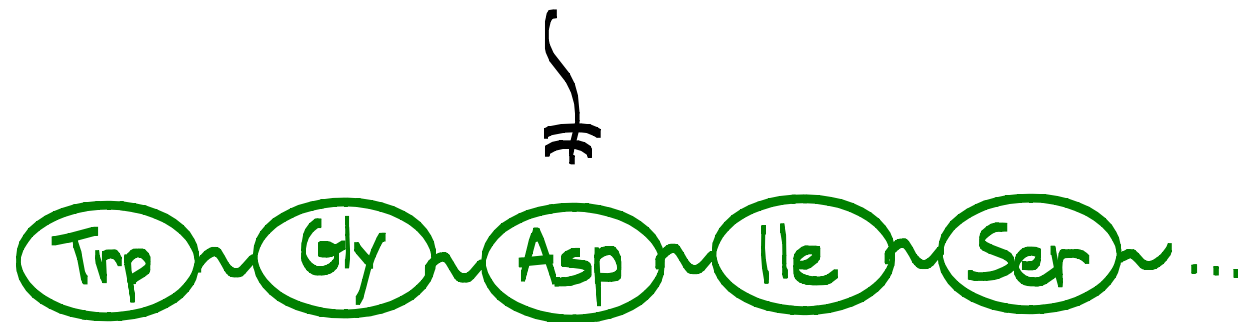
DNA
the code



RNA



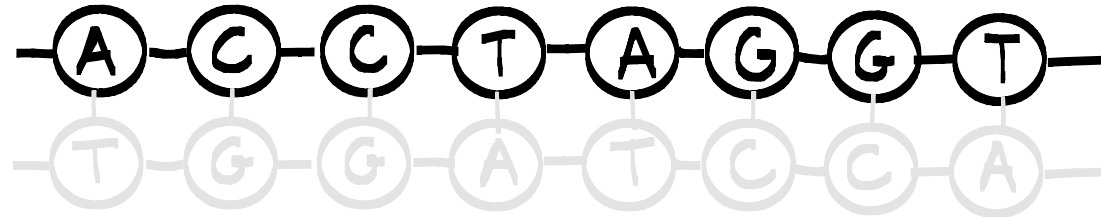
proteins



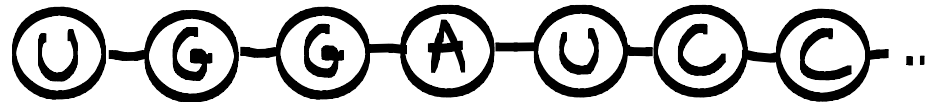
Classic dogma

WHAT IS RNA?

DNA
the code



RNA
the messenger



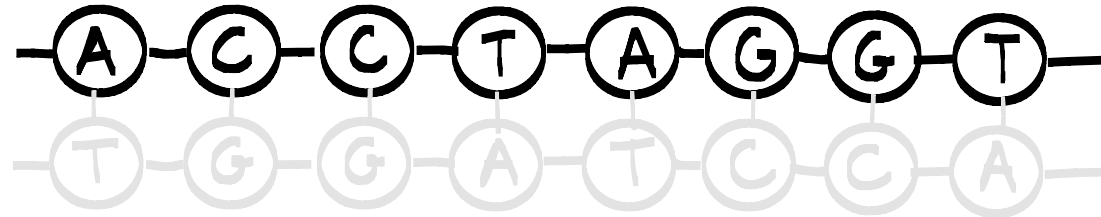
proteins
the machine



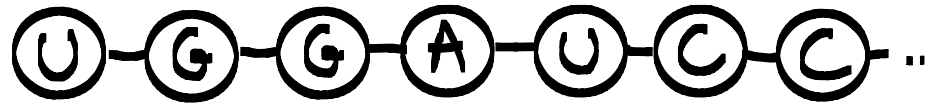
Classic dogma

WHAT IS RNA?

DNA
the code



RNA
the messenger?



BOO RING — (T)

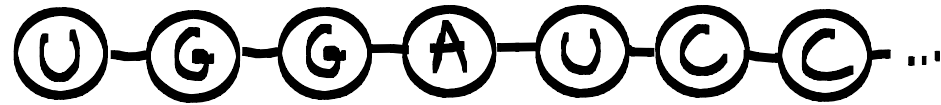


proteins
the machine



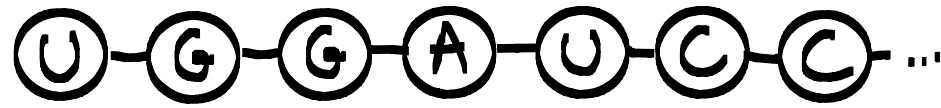
WHAT IS RNA?

RNA
the messenger

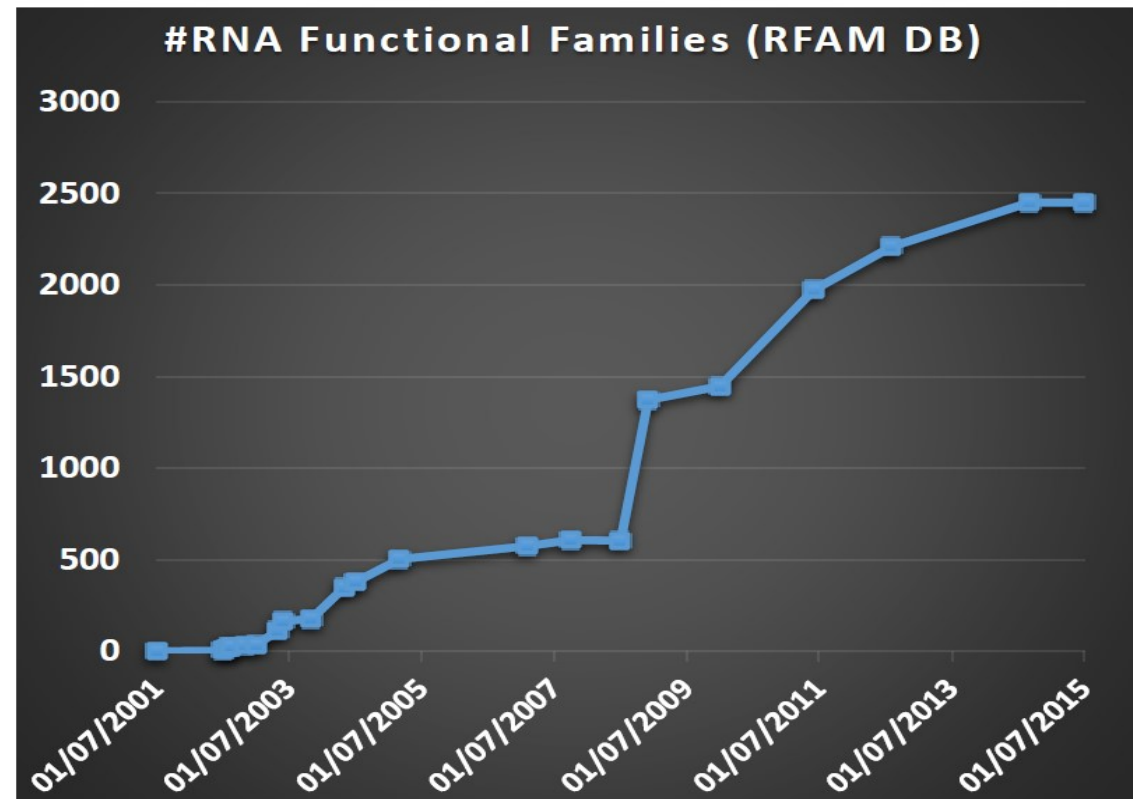


WHAT IS RNA?

RNA
the messenger



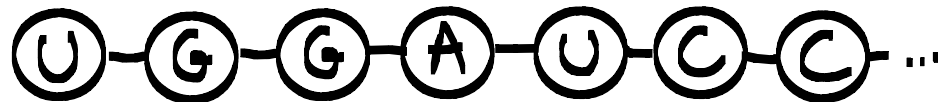
actually also... translator,
regulator, enzyme,
catalyst...



So what is RNA?

RNA is

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



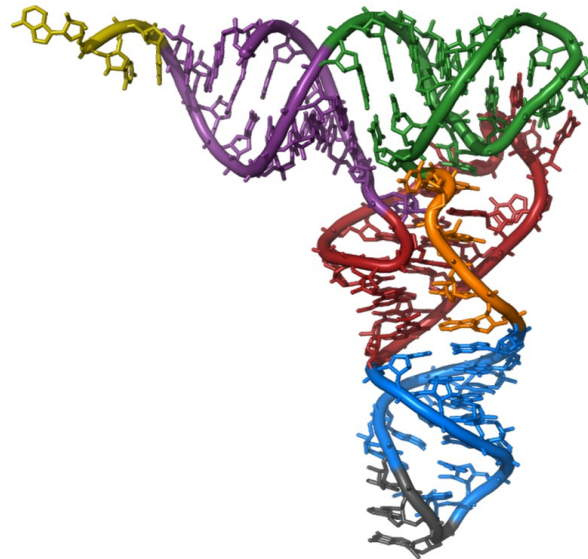
So what is RNA?

RNA is

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

... stabilized by
hydrogen bonds..

... which folds
onto itself.



So what is RNA?

RNA is

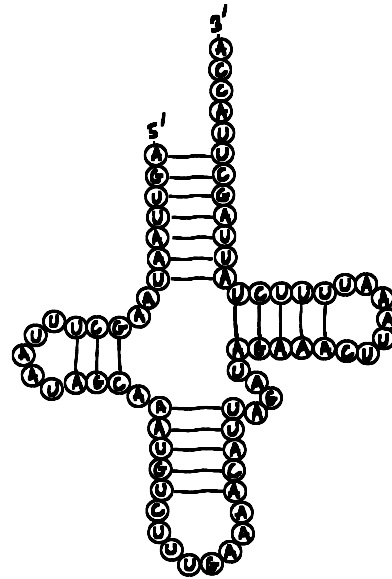
a single-stranded molecule
(chain of nucleotides)...

... stabilized by
hydrogen bonds..

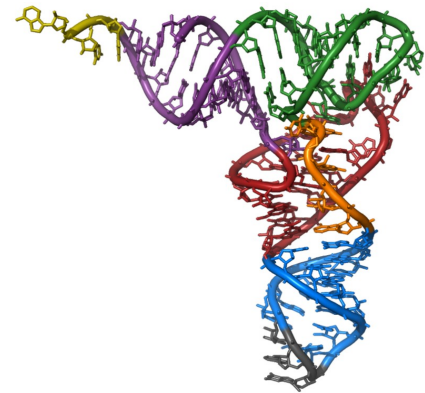
... which folds
onto itself.



primary structure



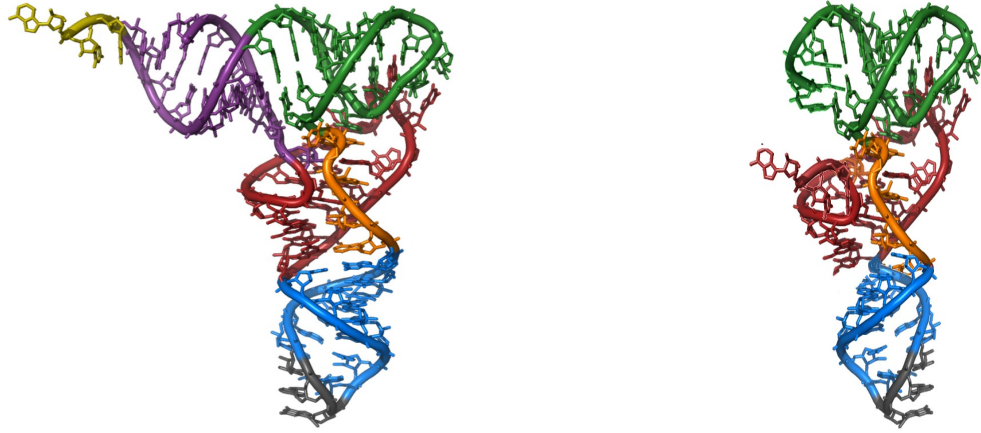
secondary structure



tertiary structure

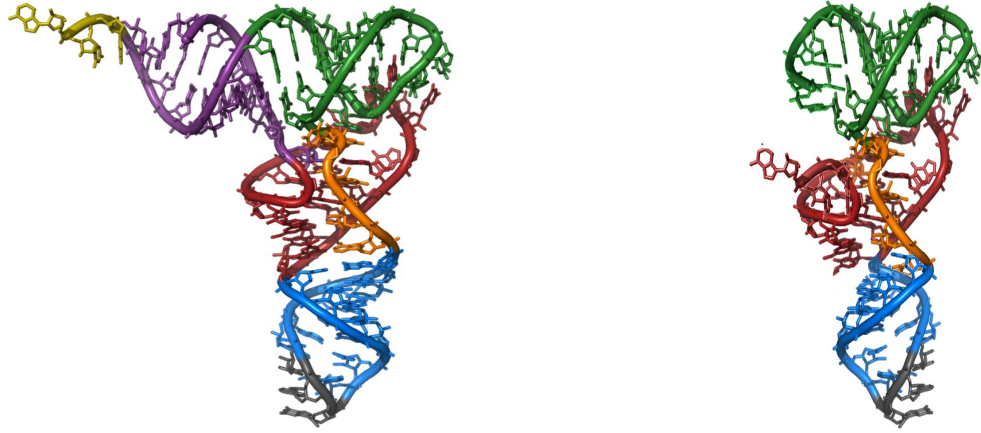
RNA COMPARISON

Interesting problem: evaluating similarity between two RNAs.



RNA COMPARISON

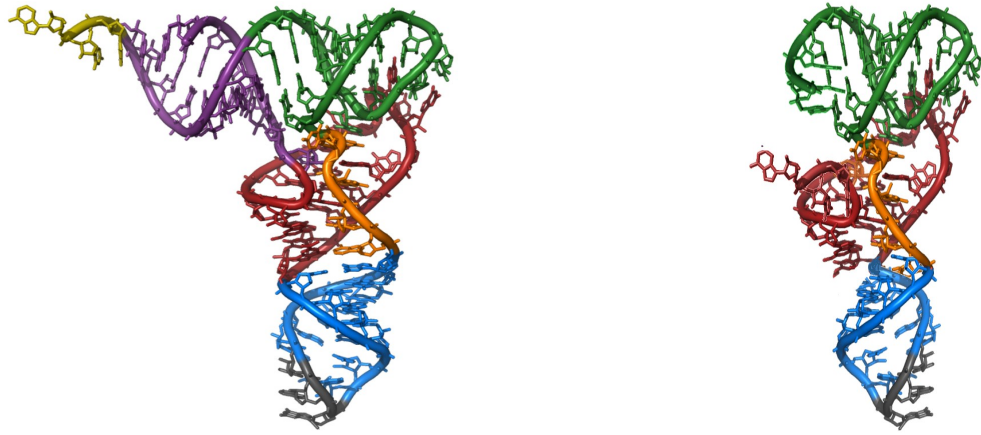
Interesting problem: evaluating similarity between two RNAs.



🤔 - Why?

RNA COMPARISON

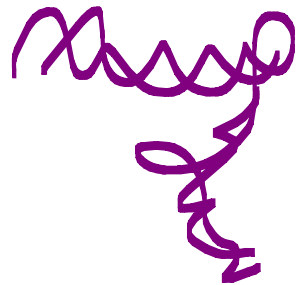
Interesting problem: evaluating similarity between two RNAs.



🤔 - Why?

Typical situation:

New
RNA



ACAGUACC...

large database

AUCCAG...

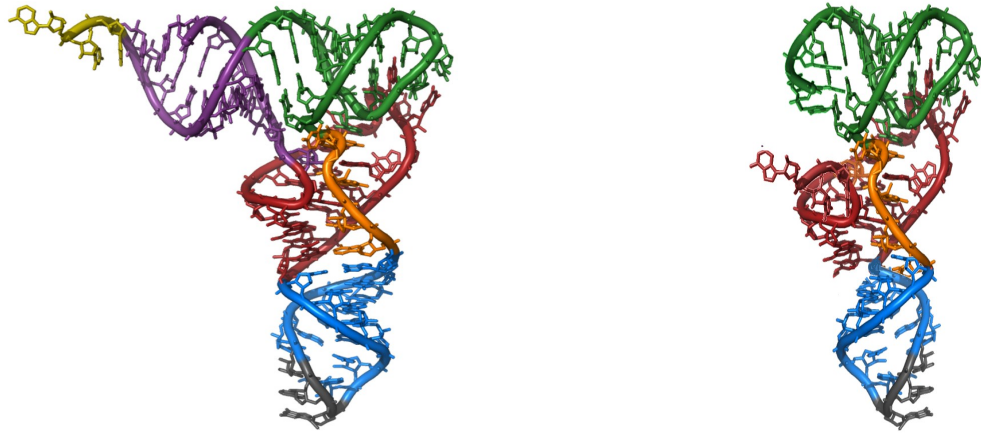
UUAGACC...

CCAGC...

AAAGU...

RNA COMPARISON

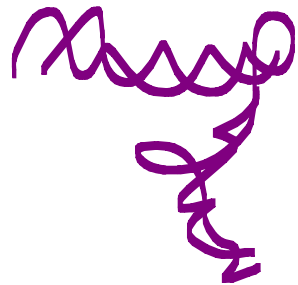
Interesting problem: evaluating similarity between two RNAs.



🤔 - Why?

Typical situation:

New RNA



ACAGUACC ...

finding similar RNAs



large database

AUCCAG ...



CCAGC ..

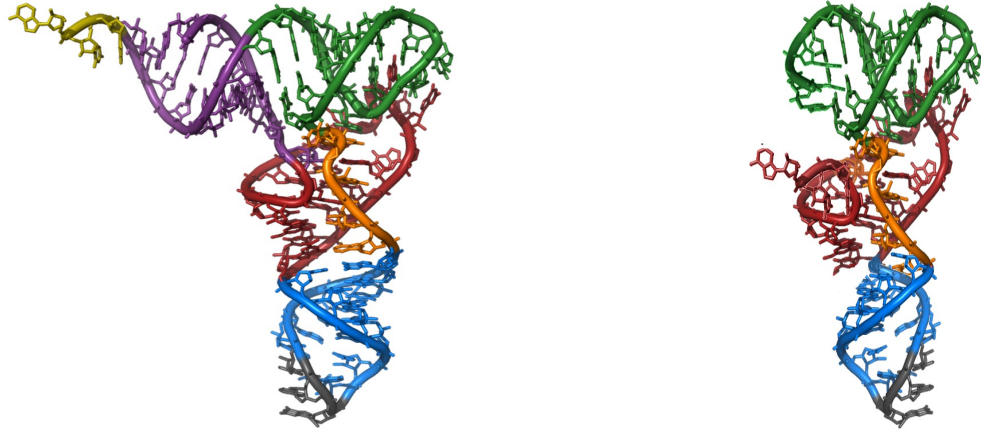
UUAGACC ...

AAAGU ...



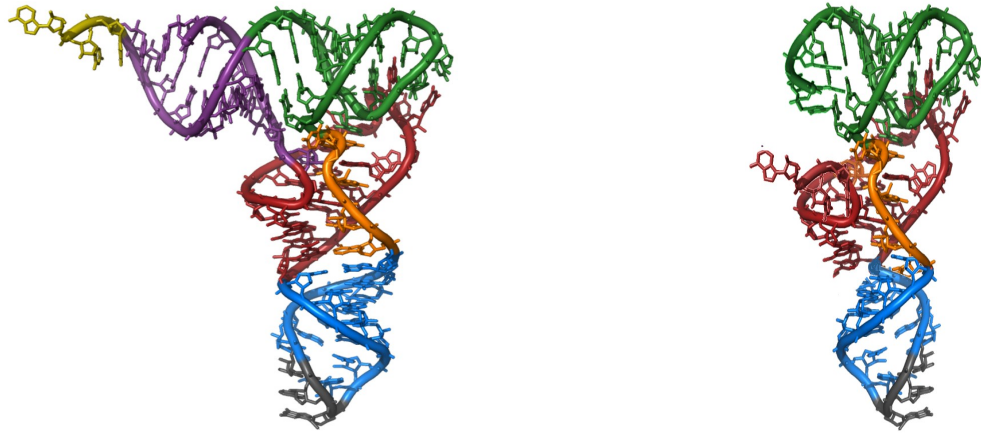
MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?



MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?



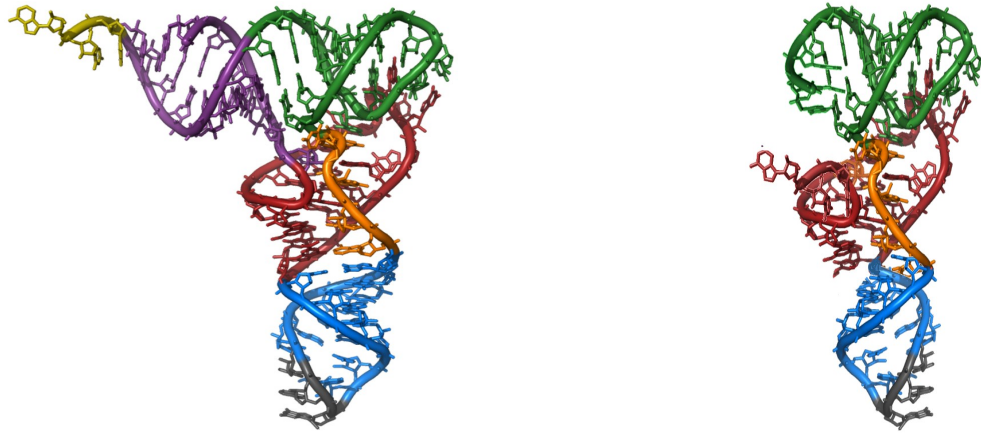
First idea: compare the primary structures.

RNA 1: AUUCGAUUA...

RNA 2: ACCAUGAUUA...

MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?



First idea: compare the primary structures.
→ sequence alignment

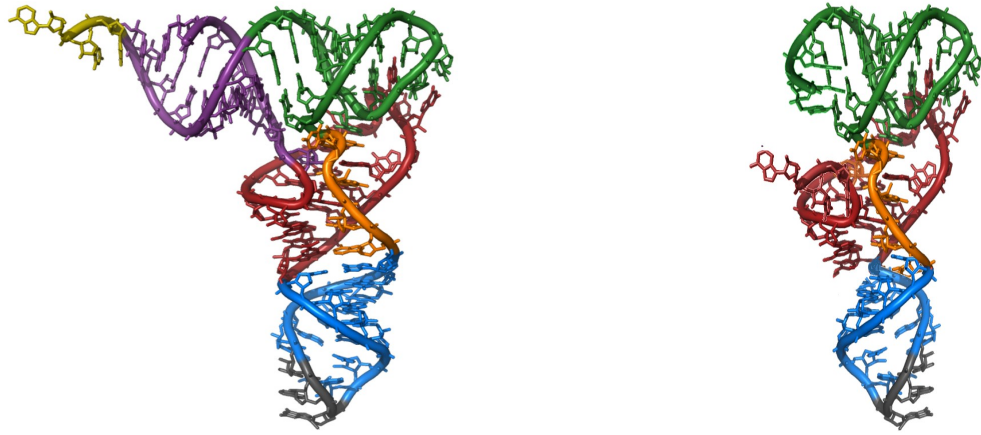
RNA 1: AUUCGAUUA...

RNA 2: ACCAUGAUUA...

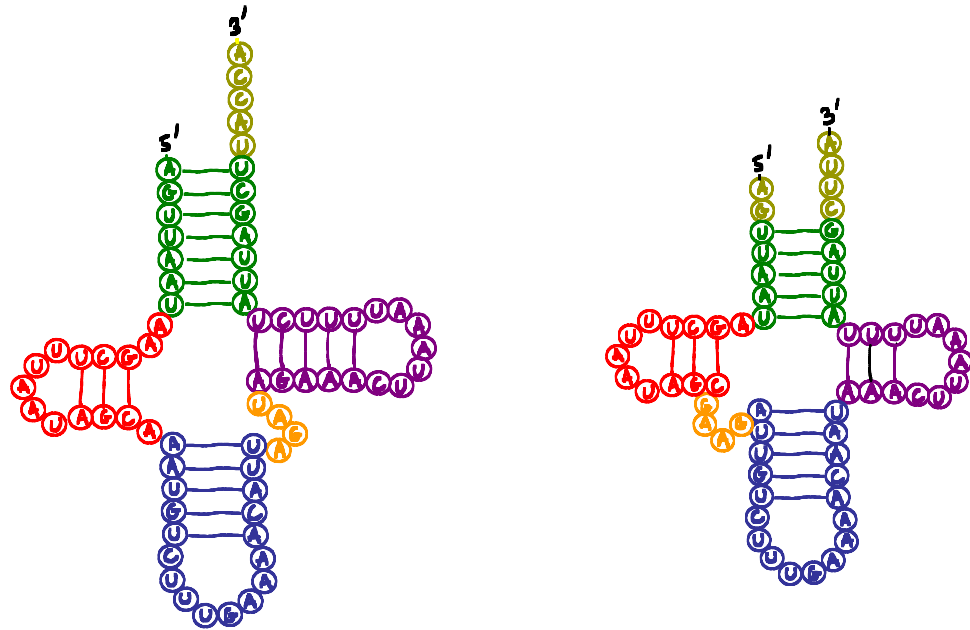
alignment: $\begin{pmatrix} A \\ A \end{pmatrix} \begin{pmatrix} U \\ - \end{pmatrix} \begin{pmatrix} - \\ C \end{pmatrix} \begin{pmatrix} U \\ - \end{pmatrix} \begin{pmatrix} C \\ C \end{pmatrix} \begin{pmatrix} - \\ A \end{pmatrix} \begin{pmatrix} - \\ U \end{pmatrix} \begin{pmatrix} G \\ G \end{pmatrix} \begin{pmatrix} A \\ A \end{pmatrix} \begin{pmatrix} U \\ U \end{pmatrix} \begin{pmatrix} U \\ U \end{pmatrix} \begin{pmatrix} A \\ A \end{pmatrix} \dots$

MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?

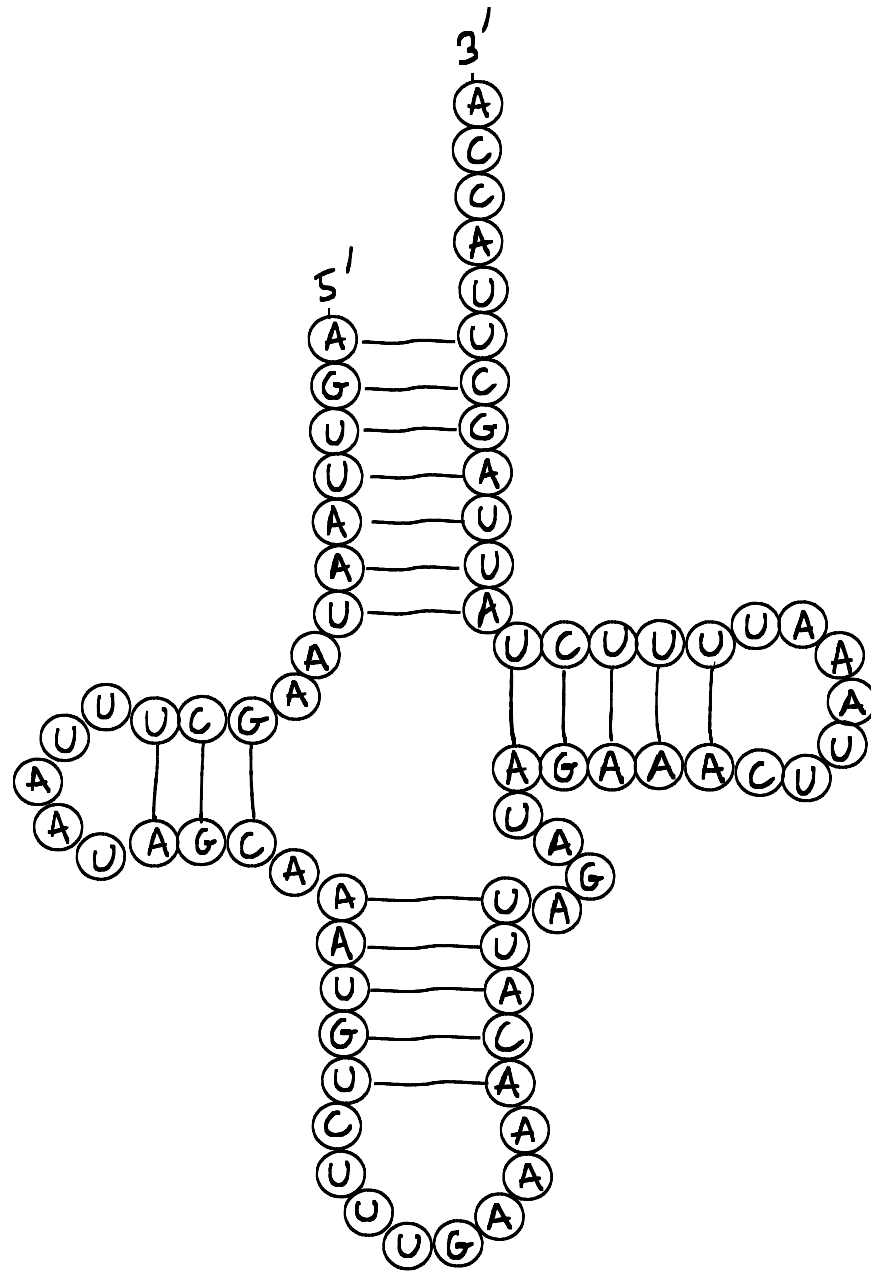


Second idea: compare secondary structures.

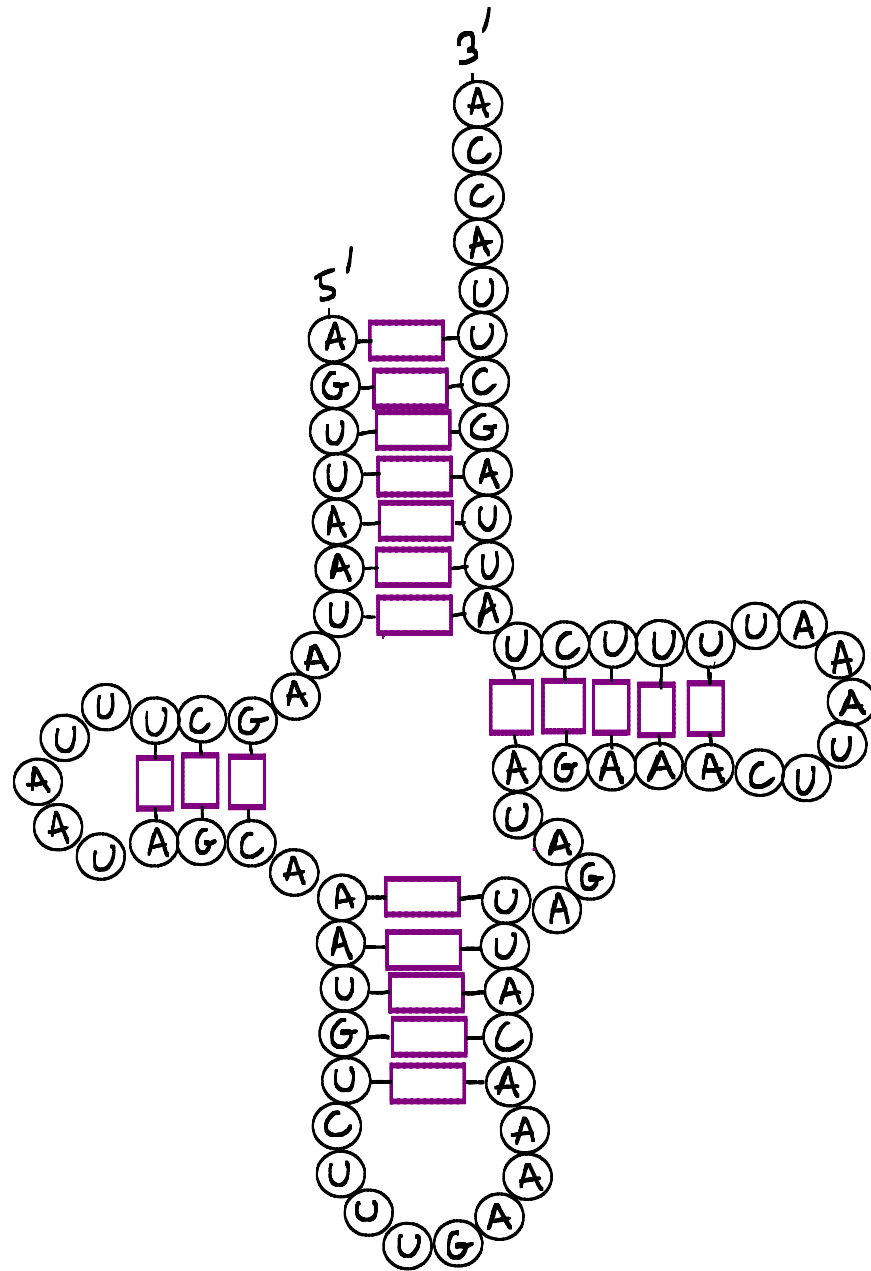


→ notion of
tree alignment
[Jiang, Wang,
Zhang]

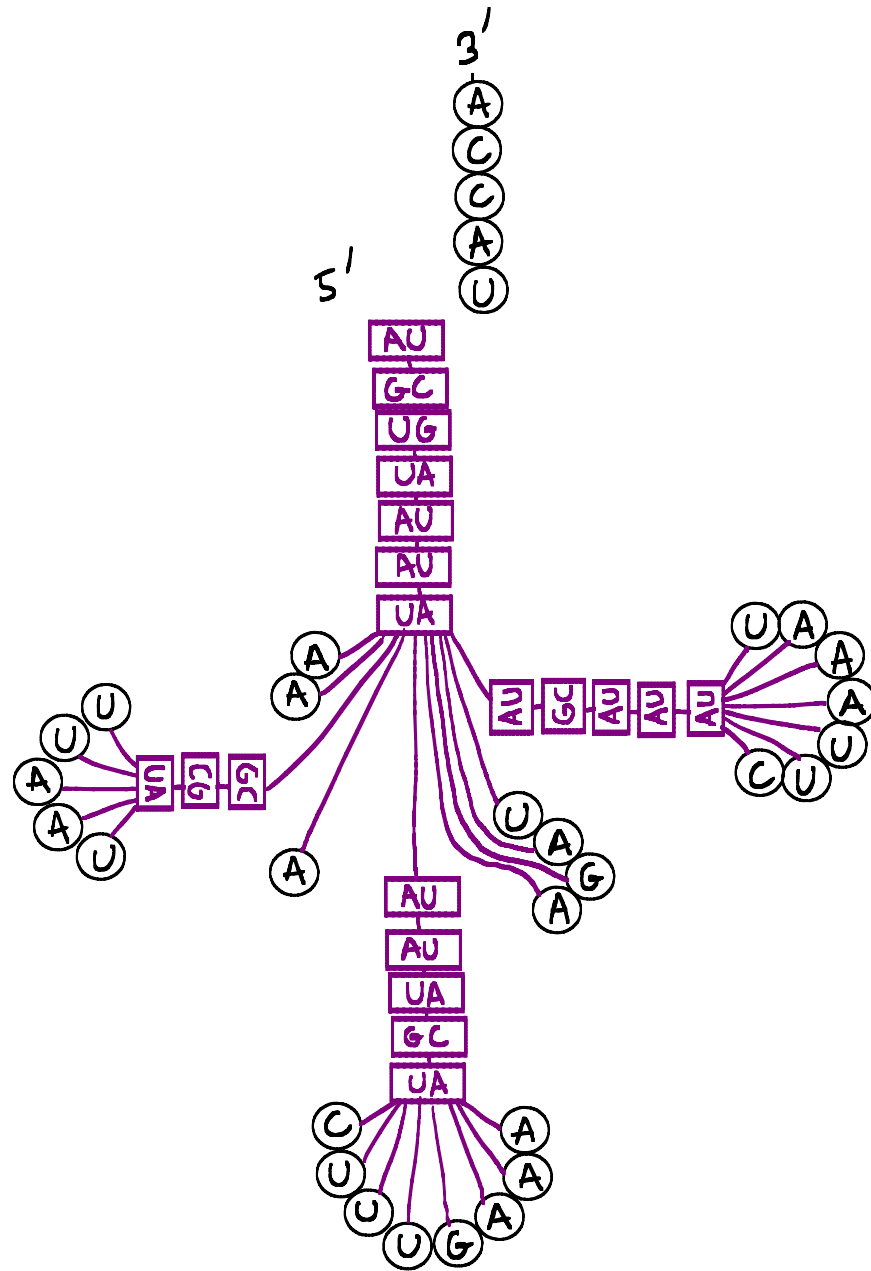
FROM SECONDARY STRUCTURES TO TREES



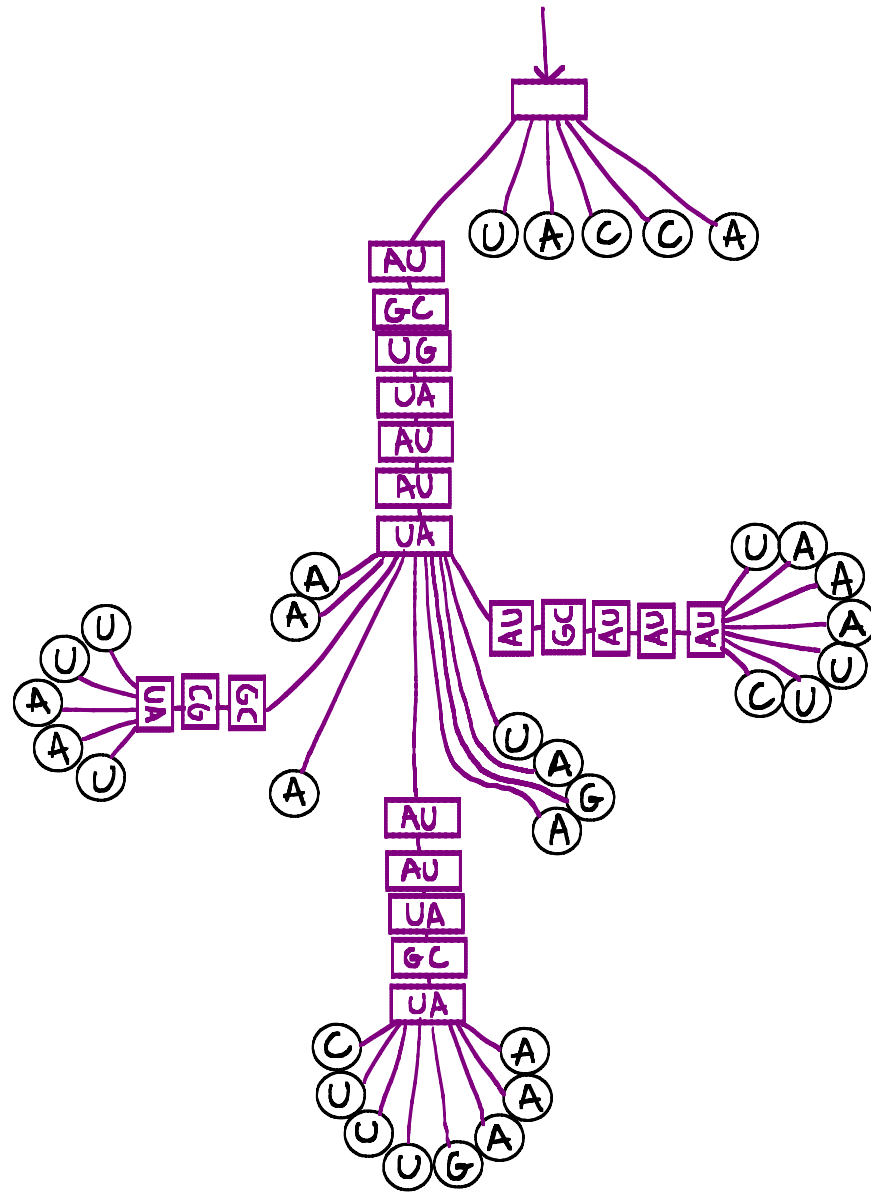
FROM SECONDARY STRUCTURES TO TREES



FROM SECONDARY STRUCTURES TO TREES

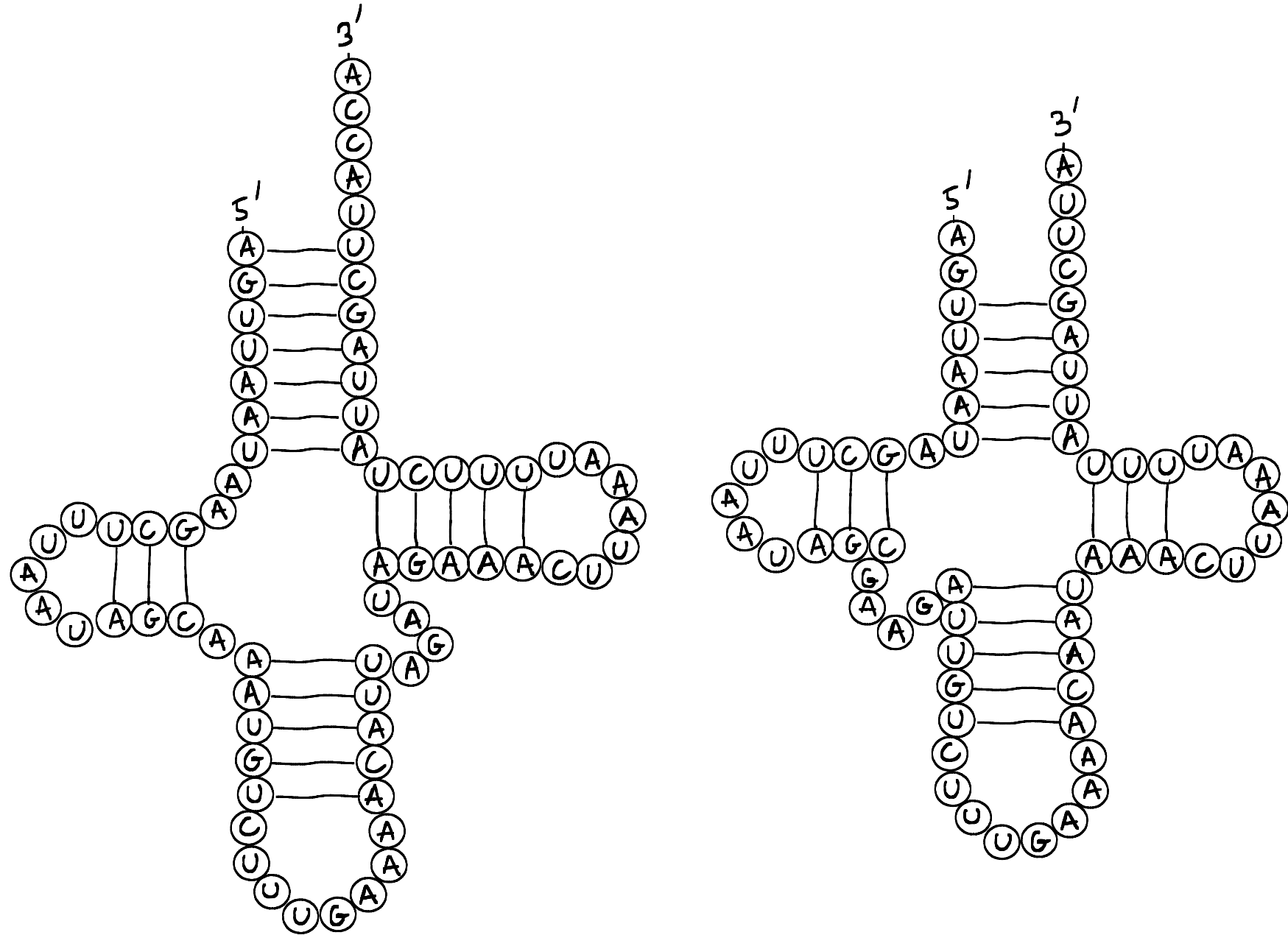


FROM SECONDARY STRUCTURES TO TREES



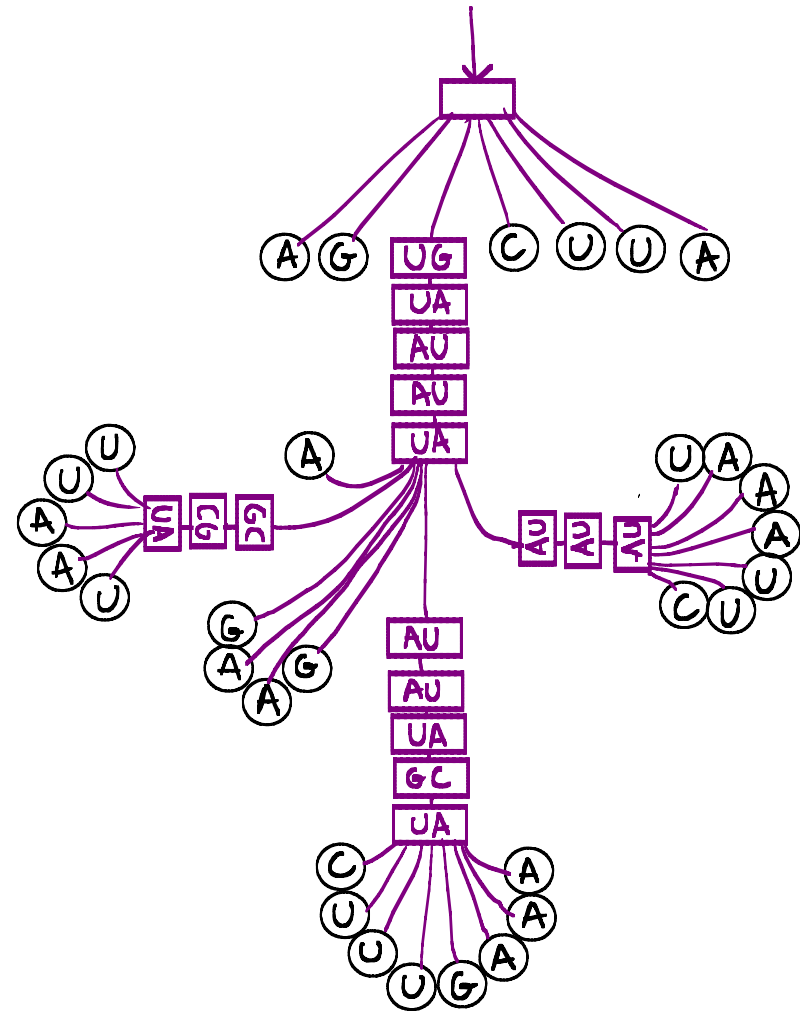
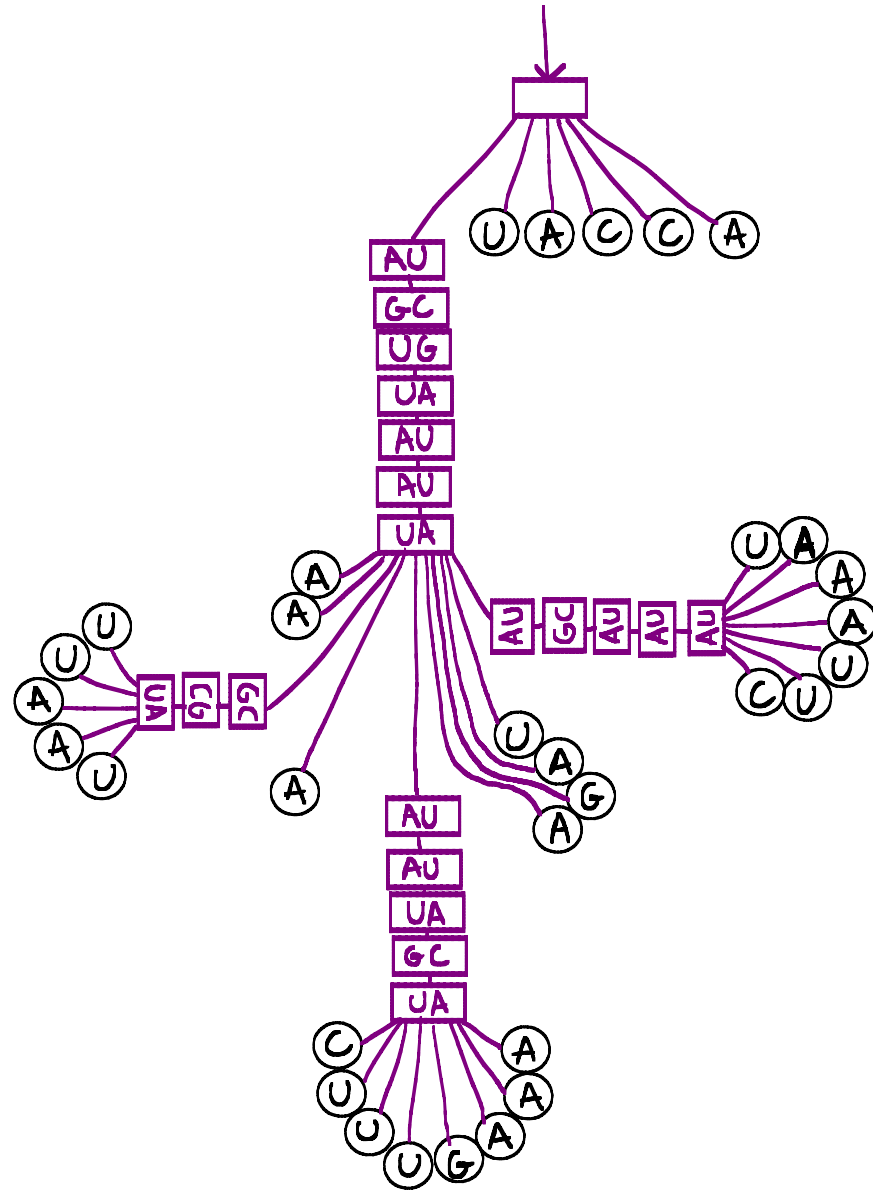
FROM SECONDARY STRUCTURES TO TREES

Objective: Align trees coming from RNA 2^{ary} structures



FROM SECONDARY STRUCTURES TO TREES

Objective: Align trees coming from RNA 2^{ary} structures



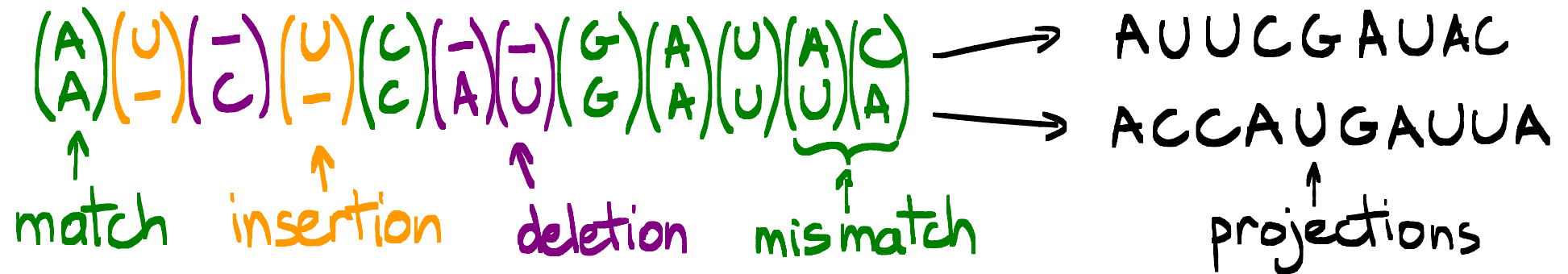
SEQUENCE ALIGNMENT

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

(A)(U)(-)(U)(C)(-)(-)(G)(A)(U)(A)(C)
(A)(-)(C)(-)(C)(A)(U)(G)(A)(U)(U)(A)
↑ match ↑ insertion ↑ deletion ↑ mismatch

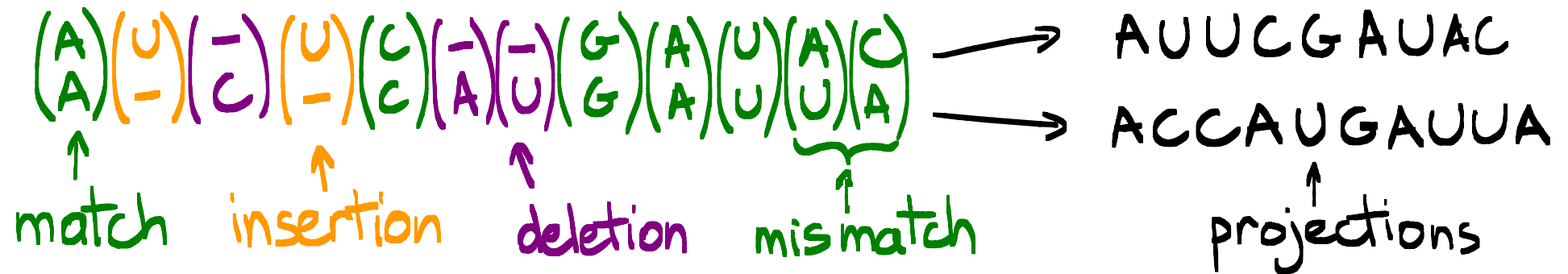
SEQUENCE ALIGNMENT

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



SEQUENCE ALIGNMENT

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

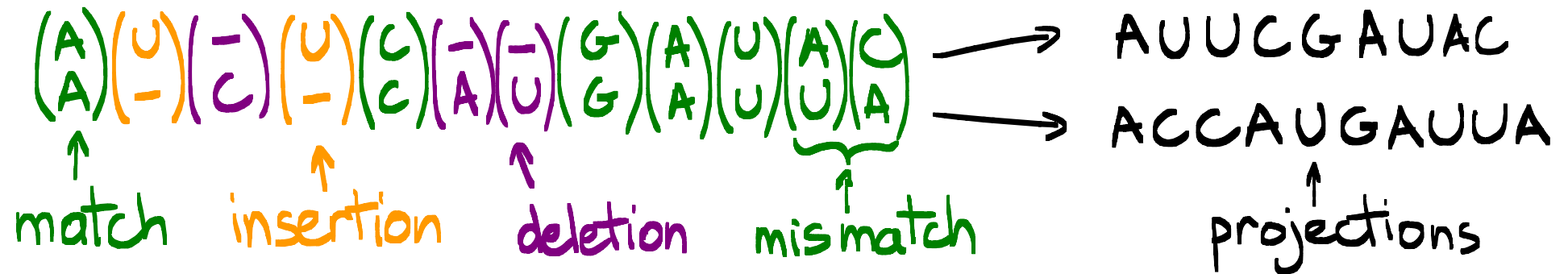


Given two sequences S_1 and S_2 ,

alignment between S_1 and S_2 = supersequence with projections S_1 and S_2

SEQUENCE ALIGNMENT

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



Given two sequences S_1 and S_2 ,

alignment between S_1 and S_2 = supersequence with projections S_1 and S_2

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

OPTIMAL ALIGNMENT

Classical problem: Given S_1 and S_2 each of length n , find one optimal alignment between S_1 and S_2 -

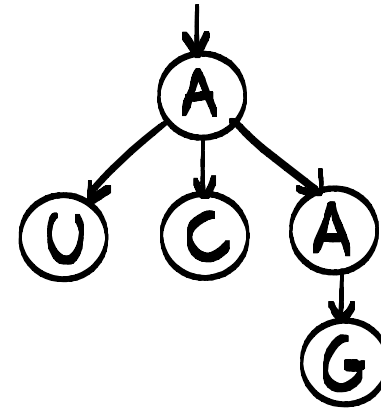
Solvable by Dynamic Programming:

- Needleman - Wunsch algorithm
- Smith - Waterman algorithm
- BLAST (heuristic)

Worst case and average time : $O(n^2)$

TREES AND SUPERTREES

Trees are plane, rooted, and vertices are labeled by an alphabet Σ .

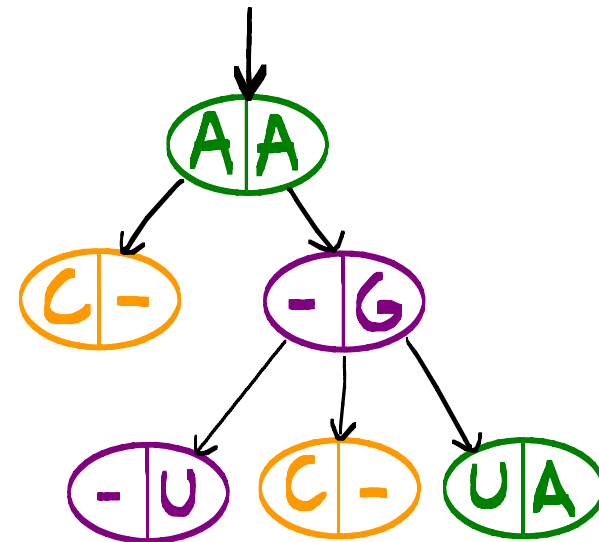


Supertree = tree with 3 types of vertices:

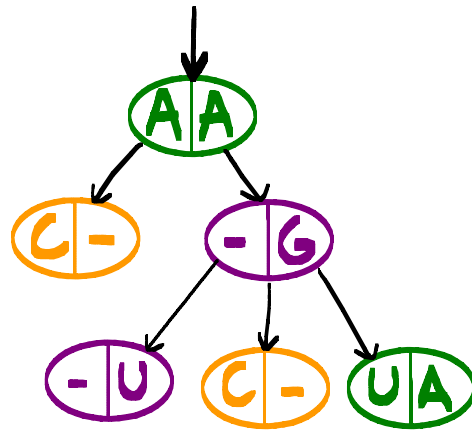
$(X|Y)$ (mis)match

$(X|-)$ insertion

$(-|Y)$ deletion



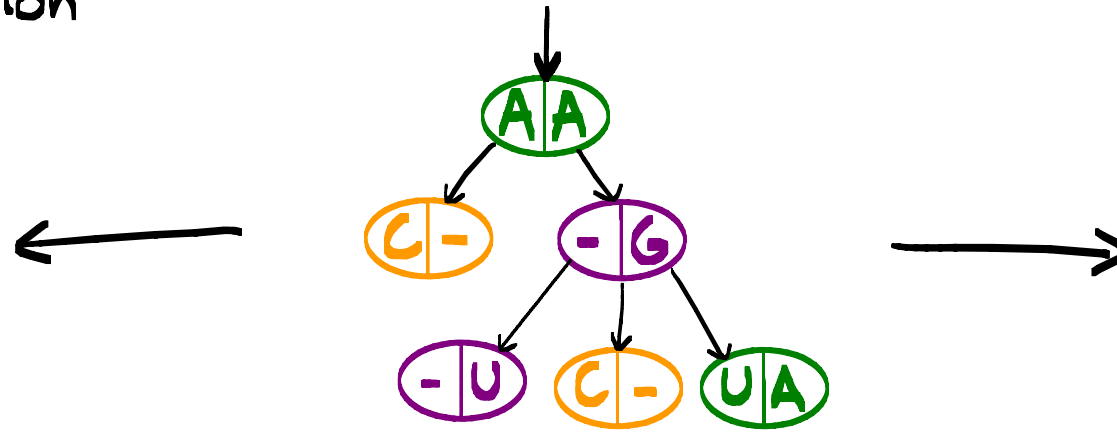
TREE ALIGNMENTS



TREE ALIGNMENTS

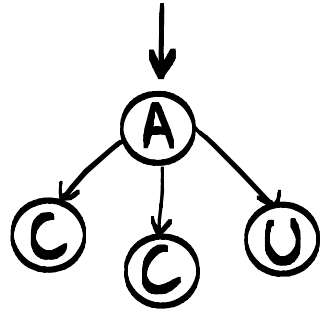
first projection

second projection

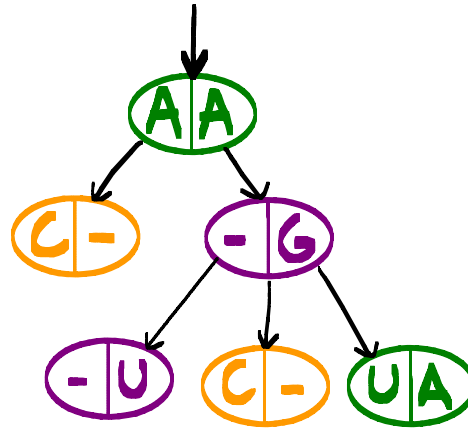


TREE ALIGNMENTS

first projection



keep left letters

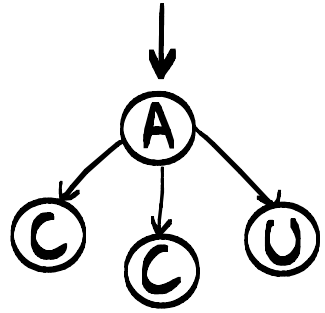


second projection



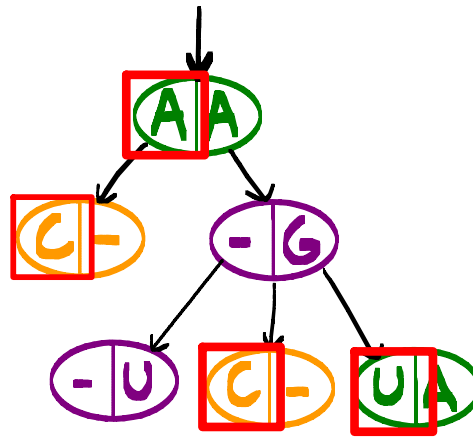
TREE ALIGNMENTS

first projection



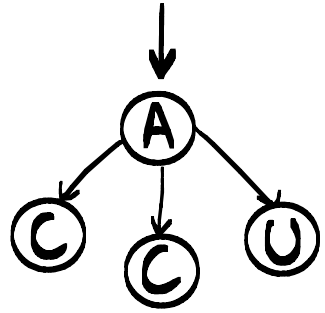
keep left letters

second projection



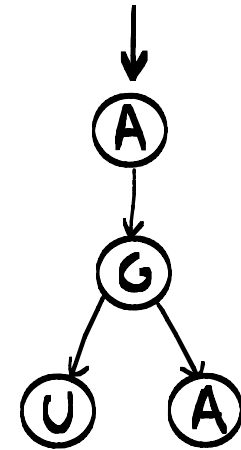
TREE ALIGNMENTS

first projection

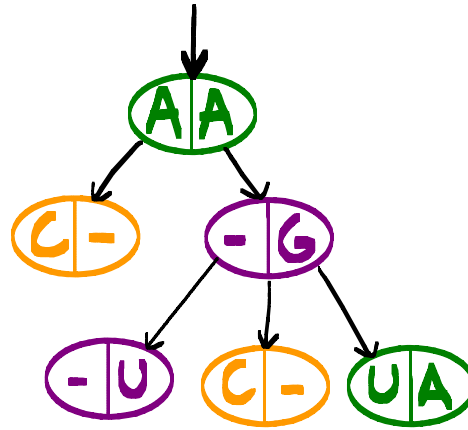


Keep left letters

second projection

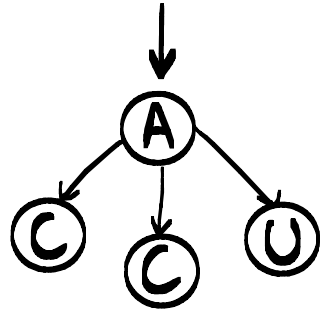


Keep right letters

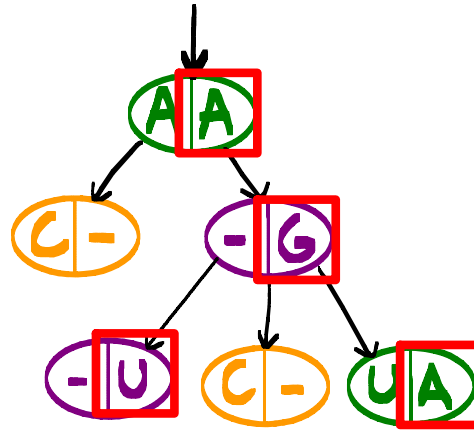


TREE ALIGNMENTS

first projection

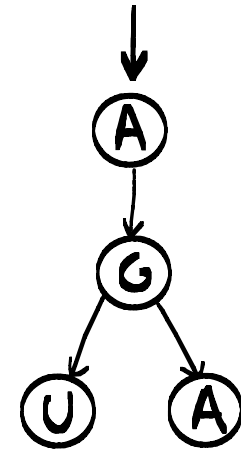


keep left letters



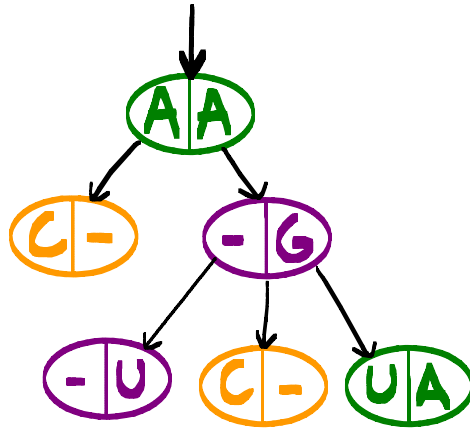
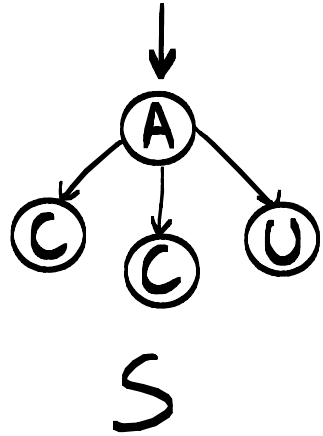
keep right letters

second projection

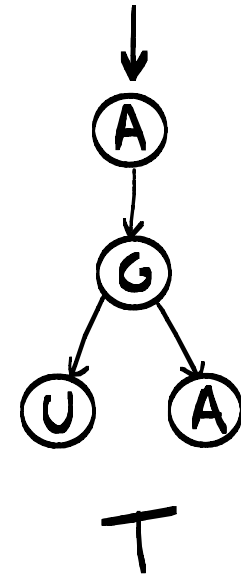


TREE ALIGNMENTS

first projection



second projection

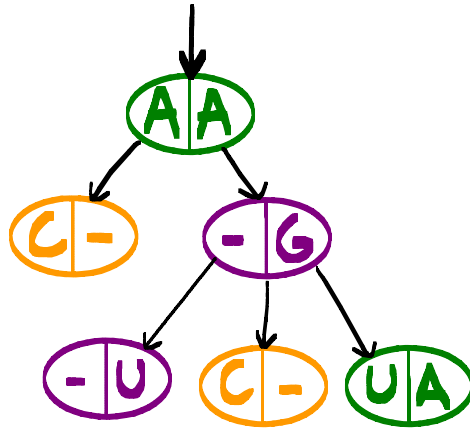
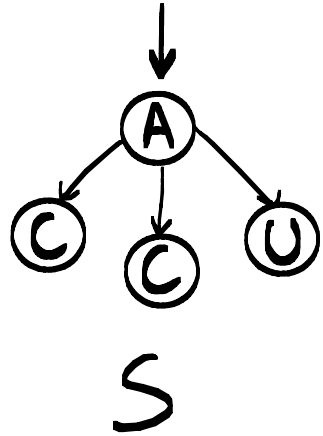


Given two trees S and T,

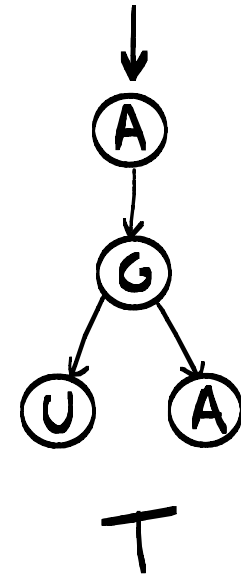
alignment between S and T = supertree whose projections are S and T.

TREE ALIGNMENTS

first projection



second projection



Given two trees S and T ,
alignment between S and T = supertree whose projections
are S and T .

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

CONNECTION WITH SEQUENCE ALIGNMENTS

Tree alignments generalize sequence alignments.

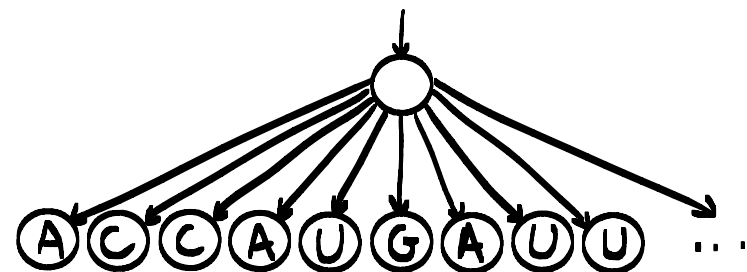
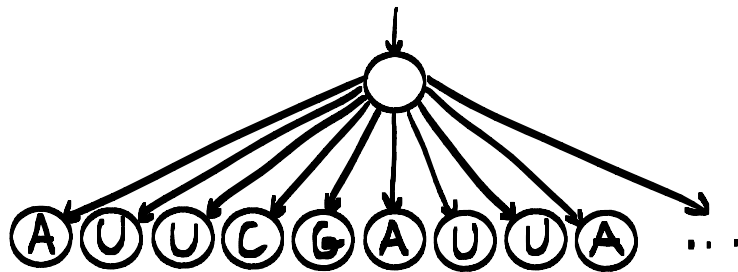
MCZMCDMS

AUUCGAUUA...

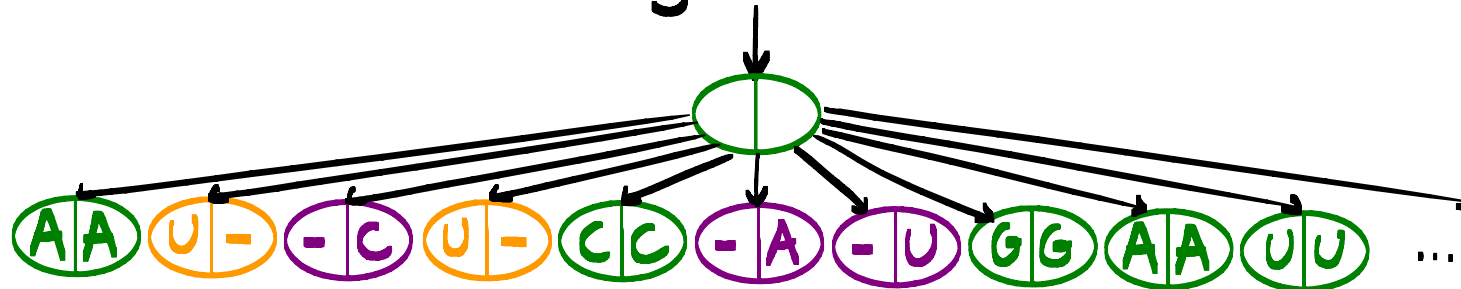
ACCAUGAUUA...

alignment:

(A)(U)(-)(U)(C)(-)(-)(G)(A)(U)(U)(A) ...
(A)(-)(C)(-)(C)(A)(U)(G)(A)(U)(U)(A) ...



alignment:



MMR-H

OPTIMAL ALIGNMENT

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

Solvable by Dynamic Programming:

Worst case time

$$O(n^4)$$

[Jiang, Wang, Zhang]

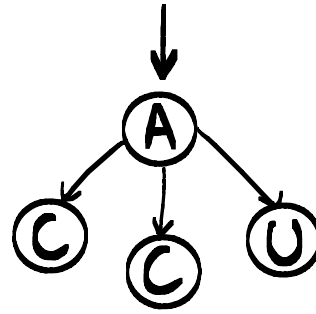
Average time

$$O(n^2)$$

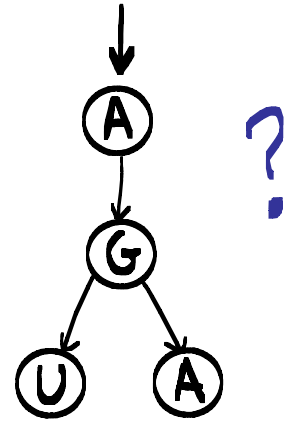
[Herrbach, Denise, Dulucq]

SPACE OF ALIGNMENTS

Is there only one optimal alignment between

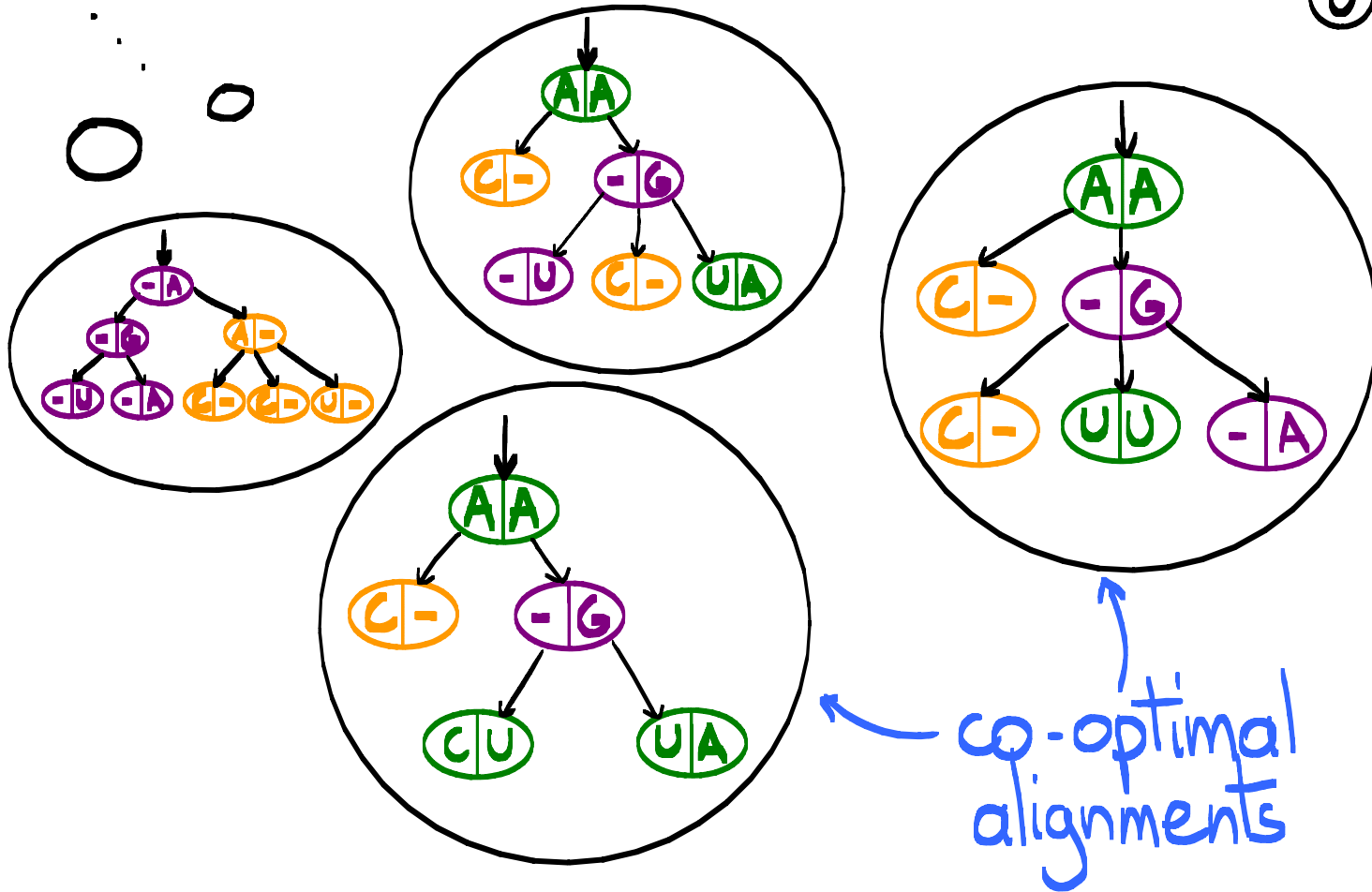
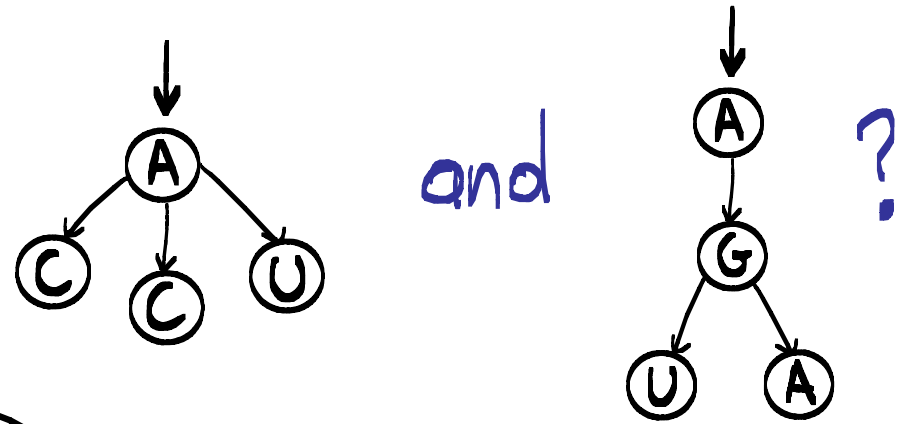


and



SPACE OF ALIGNMENTS

Is there only one optimal alignment between



SPACE OF ALIGNMENTS

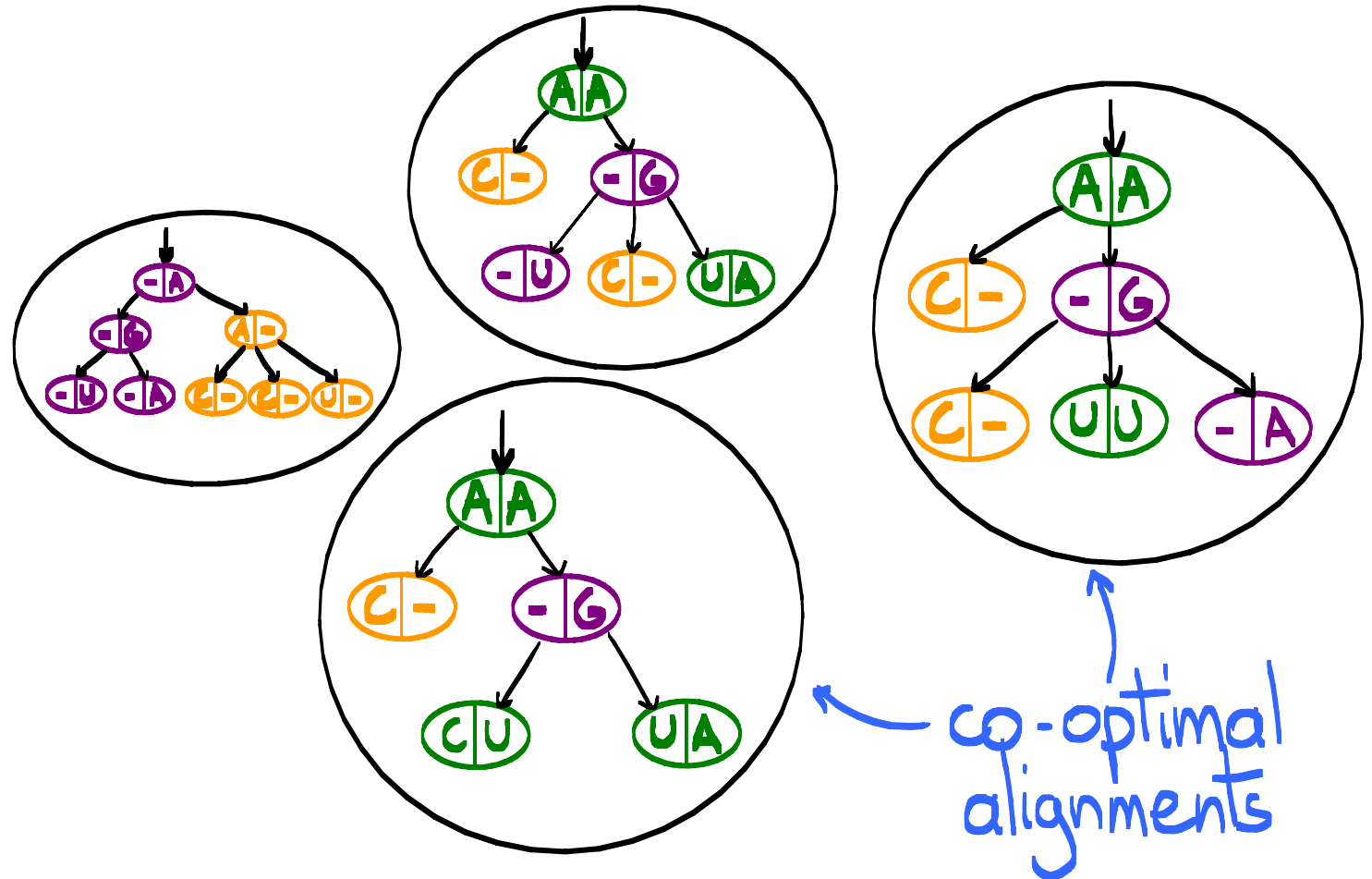
Why finding one optimal alignment may be inadequate:

- ▶ Co-optimal alignments can be very different.
- ▶ Exploring the space of alignments enables the detection of high probability features.

SPACE OF ALIGNMENTS

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

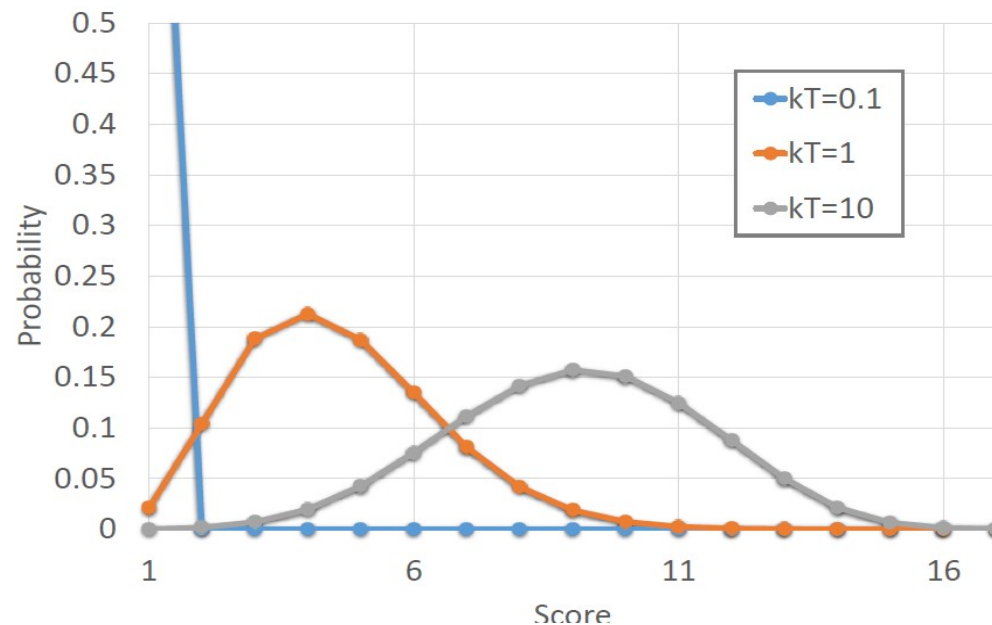
probability of
an alignment A
 $\propto e^{-\frac{\text{cost}(A)}{k}}$
(Gibbs-Boltzmann
distribution)



SPACE OF ALIGNMENTS

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

Score vs Boltzmann probability
(Density of states)



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

$K=0$: Uniform distribution over
optimal alignments.

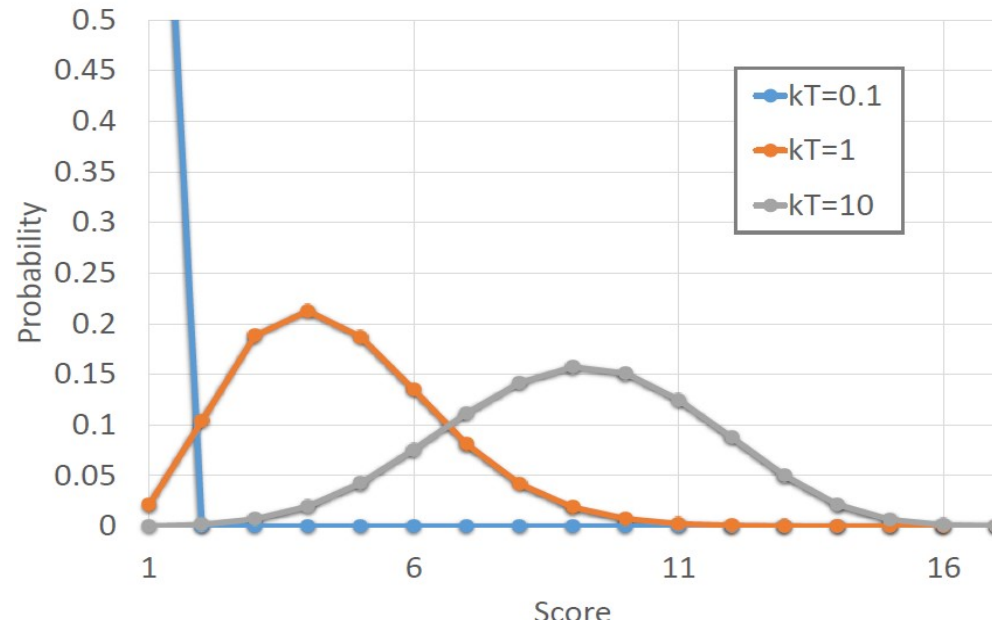
$K=+\infty$: Uniform distribution over
all alignments.

SPACE OF ALIGNMENTS

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

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

Score vs Boltzmann probability
(Density of states)



K=0 : Uniform distribution over optimal alignments.

K=+∞ : Uniform distribution over all alignments.

AMBIGUITY OF ALIGNMENTS

For sequences,

(A)(U)(-)(U)(C)(-)(-)(G)(A)(U)(U)(A)
(A)(-)(C)(-)(C)(A)(U)(G)(A)(U)(U)(A)

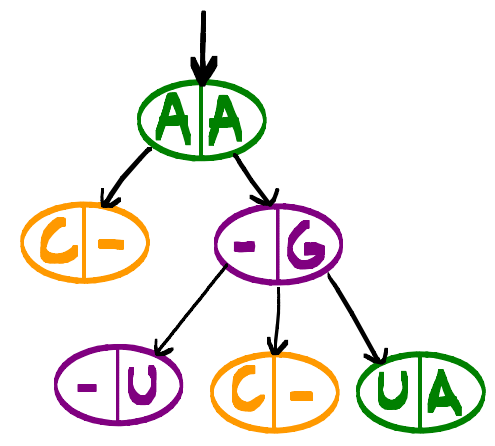
is the same
alignment as

(A)(-)(U)(U)(C)(-)(-)(G)(A)(U)(U)(A)
(A)(C)(-)(-)(C)(A)(U)(G)(A)(U)(U)(A)

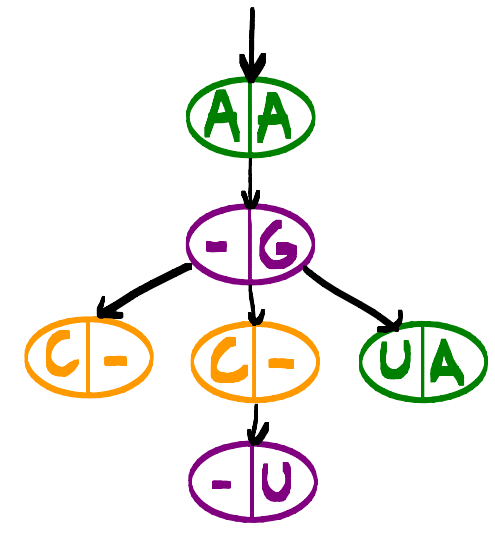


AMBIGUITY OF ALIGNMENTS

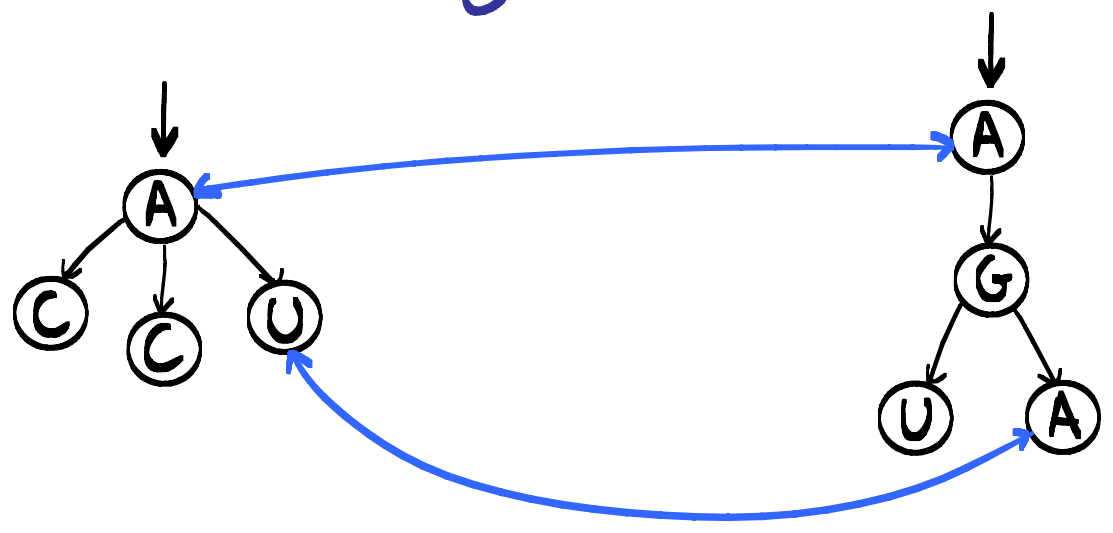
For trees,



and

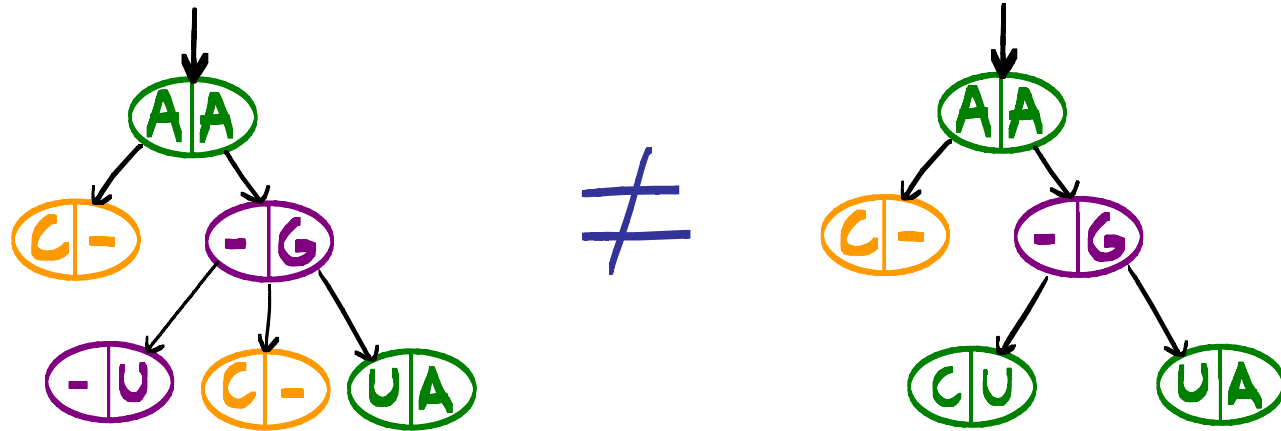


induce the same alignment between

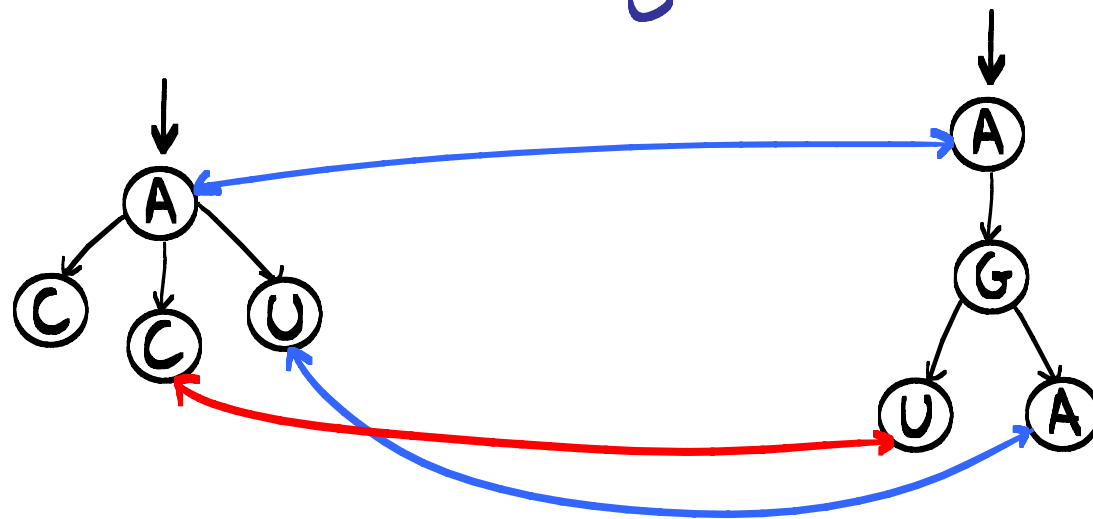


AMBIGUITY OF ALIGNMENTS

The two supertrees



do not induce the same alignment between the trees



PROBLEM RAISED BY THE AMBIGUITY

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

Ex:

(A)(U)(-)(U)(C)(-)(-)(G)(A)(U)(U)(A)
(A)(-)(C)(-)(C)(A)(U)(G)(A)(U)(U)(A)

PROBLEM RAISED BY THE AMBIGUITY

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

Ex:

(A)(U)(U)(-)(C)(-)(-)(G)(A)(U)(U)(A)
(A)(-)(-)(C)(C)(A)(U)(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)(-)(-)(G)(A)(U)(U)(A)
(A)(-)(-)(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.

Ex:
$$\begin{array}{cccccccccccc} (A) & (U) & (U) & (-) & (C) & (-) & (-) & (G) & (A) & (U) & (U) & (A) \\ (A) & (-) & (-) & (C) & (C) & (A) & (U) & (G) & (A) & (U) & (U) & (A) \end{array}$$

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.

Ex:
$$\begin{array}{cccccccccccc} (A) & (U) & (U) & (-) & (C) & (-) & (-) & (G) & (A) & (U) & (U) & (A) \\ (A) & (-) & (-) & (C) & (C) & (A) & (U) & (G) & (A) & (U) & (U) & (A) \end{array}$$

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:

$$\mathcal{S} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \mathcal{E}$$

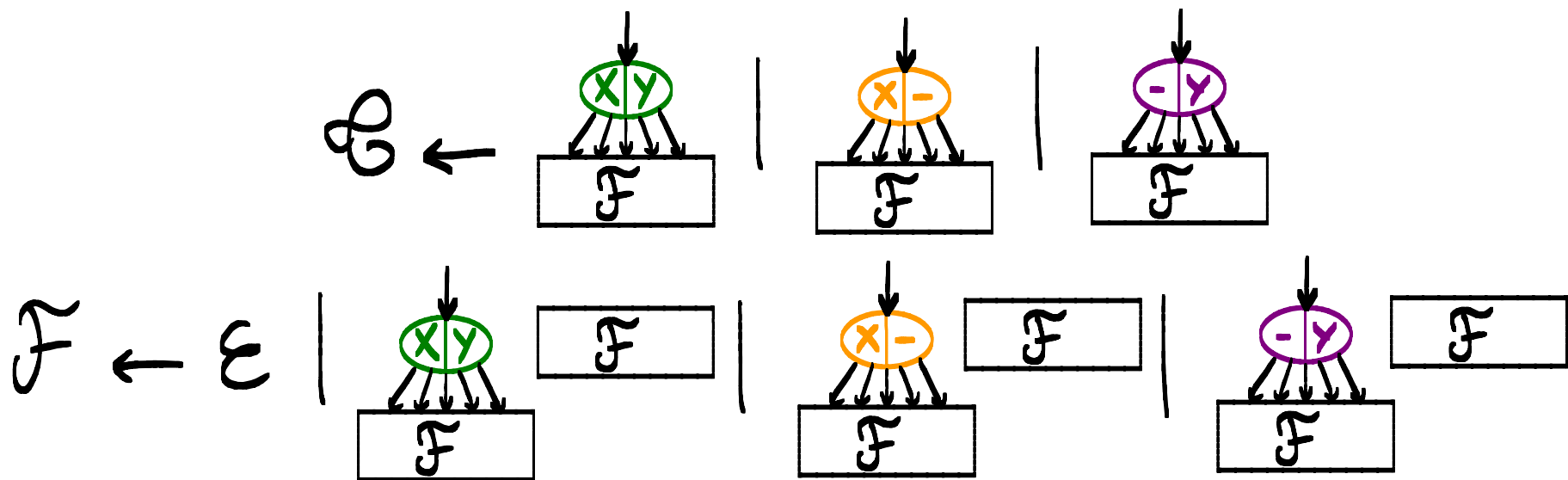
Non-ambiguous grammar:

$$\mathcal{S} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \mathcal{E}$$

$$\mathcal{S}^D \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \mathcal{E}$$

A GRAMMAR FOR ALIGNMENTS

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



A GRAMMAR FOR ALIGNMENTS

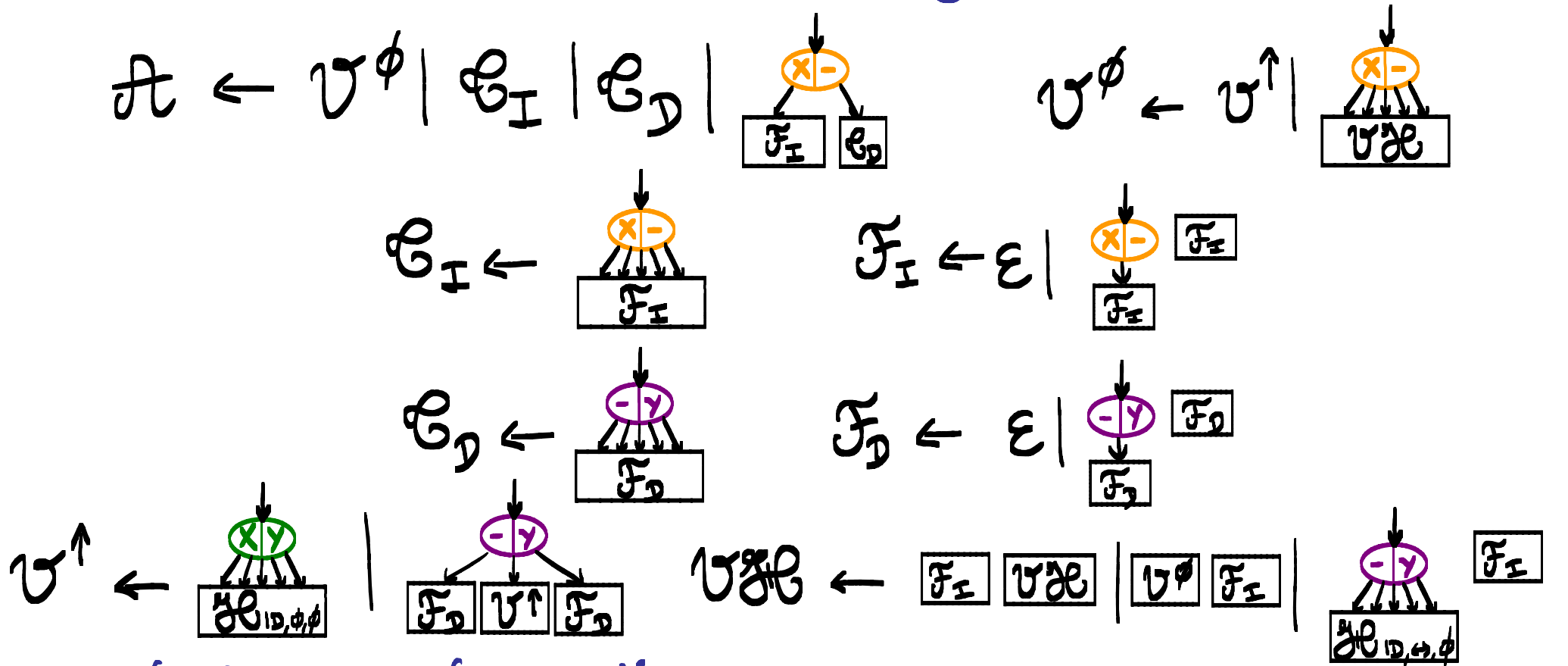
For trees, an ambiguous grammar can be derived from [Jiang, Wang, Zhang].

Our result:

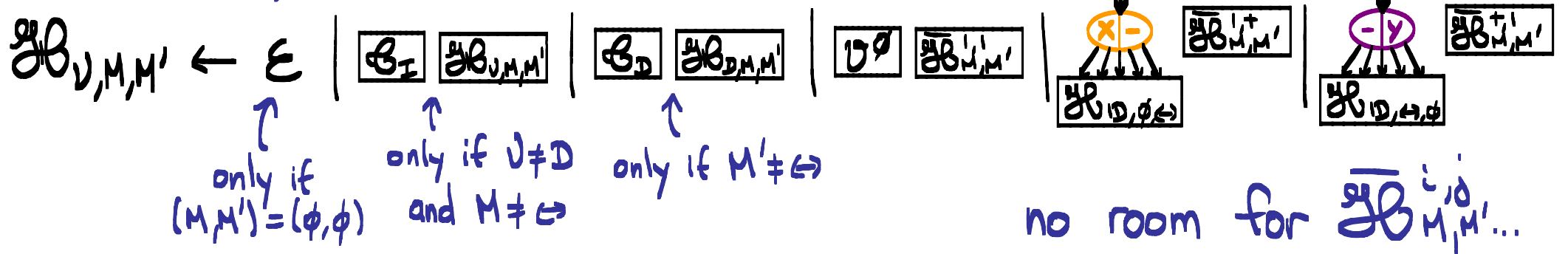
Theorem: The set \mathcal{A} generated by the following grammar contains every tree alignment exactly once.

A GRAMMAR FOR ALIGNMENTS

Our (complicated) non-ambiguous grammar:



For $v \in \{D, D^c\}, (M, M') \in \{\phi, \rightarrow, \Leftrightarrow\}^2$:



APPLICATION 1: COUNTING.

Theorem: The generating function $A(z, u)$ of tree alignments satisfies

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

where

$$(uzC(z)^2 - z^2C(z)^2 + 2z)B(z, u)^2 + (z^2C^4(z) - 2zC(z)^2 - 1)B(z, u) + C^2(z) = 0$$

and

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

APPLICATION 1. COUNTING.

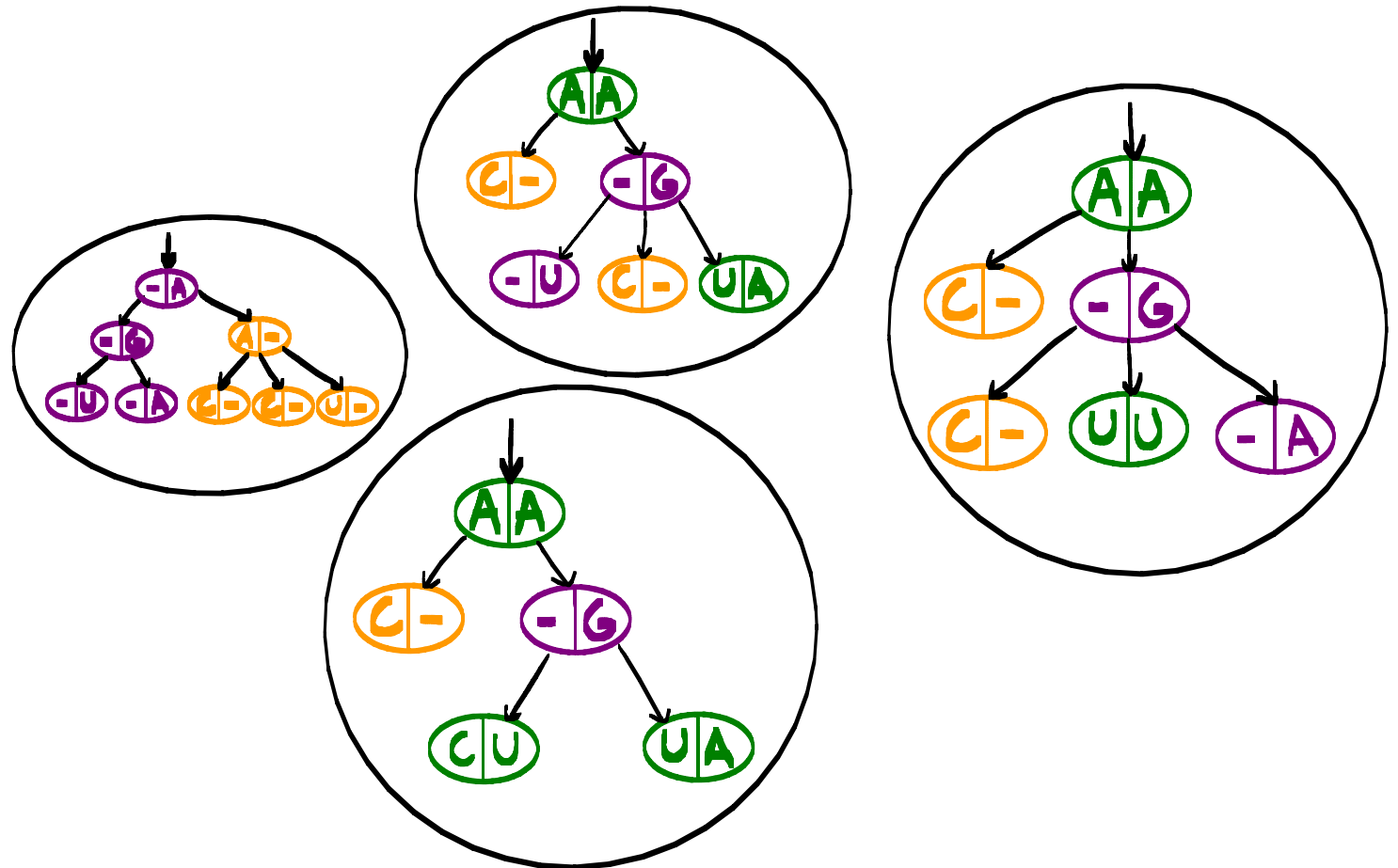
Theorem There are on average
 $C \times 1.5^n$ alignments
between two random trees of cumulative size n
where $C = 0.299\dots$

Corollary: A same alignment was repeated
 $\sim 0.875 \times 1.412^n$ times on
average in Jiang et al.'s
ambiguous grammar.

APPLICATION 2. SAMPLING

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

probability of
an alignment A
 $\propto e^{-\frac{\text{cost}(A)}{k}}$
(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.

APPLICATION 2. SAMPLING

FILTERING THE GRAMMAR (for sequence alignments)

$$\begin{aligned} \mathcal{S} &\leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \mathcal{E} \\ \mathcal{S}^D &\leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \mathcal{E} \end{aligned}$$

APPLICATION 2. SAMPLING

FILTERING THE GRAMMAR (for sequence alignments)

$$\begin{aligned} \mathcal{S} &\leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \varepsilon \\ \mathcal{S}^D &\leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \varepsilon \end{aligned}$$

Fix two sequences S_1 and S_2 -
 $\mathcal{A}[S_1, S_2] :=$ set of alignments between S_1 and S_2

APPLICATION 2. SAMPLING

FILTERING THE GRAMMAR (for sequence alignments)

$$\begin{aligned} \mathcal{S} &\leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \varepsilon \\ \mathcal{S}^D &\leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \varepsilon \end{aligned}$$

Fix two sequences S_1 and S_2 -
 $\mathcal{A}[S_1, S_2] :=$ set of alignments between S_1 and S_2 ??

APPLICATION 2. SAMPLING

FILTERING THE GRAMMAR (for sequence alignments)

$$\mathcal{S} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \varepsilon$$

$$\mathcal{S}^D \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \varepsilon$$

Fix two sequences S_1 and S_2 -
 $\mathcal{A}[S_1, S_2] :=$ set of alignments between S_1 and S_2 ??

If $S_1 = \varepsilon$ and $S_2 = \varepsilon$,
then

$$\mathcal{A}[S_1, S_2] \leftarrow \varepsilon \qquad \mathcal{A}^D[S_1, S_2] \leftarrow \varepsilon$$

APPLICATION 2. SAMPLING

FILTERING THE GRAMMAR (for sequence alignments)

$$\mathcal{G} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{G}} \oplus \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{G}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{G}^D} \oplus \varepsilon$$

$$\mathcal{G}^D \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{G}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{G}^D} \oplus \varepsilon$$

Fix two sequences S_1 and S_2 -
 $\mathcal{A}[S_1, S_2] :=$ set of alignments between S_1 and S_2 ??

If $S_1 = \varepsilon$ and $S_2 = Y S'_2$,
then

$$\mathcal{A}[S_1, S_2] \leftarrow \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{A}^D[\varepsilon, S'_2]} \quad \mathcal{A}^D[S_1, S_2] \leftarrow \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{A}^D[\varepsilon, S'_2]}$$

APPLICATION 2. SAMPLING

FILTERING THE GRAMMAR (for sequence alignments)

$$\mathcal{G} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{G}} \oplus \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{G}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{G}^D} \oplus \varepsilon$$

$$\mathcal{G}^D \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} \boxed{\mathcal{G}} \oplus \begin{pmatrix} - \\ y \end{pmatrix} \boxed{\mathcal{G}^D} \oplus \varepsilon$$

Fix two sequences S_1 and S_2 -
 $\mathcal{A}[S_1, S_2] :=$ set of alignments between S_1 and S_2 ??

If $S_1 = XS'$ and $S_2 = \varepsilon$,
then

$$\mathcal{A}[S_1, S_2] \leftarrow \begin{pmatrix} x \\ - \end{pmatrix} \boxed{\mathcal{A}[S', \varepsilon]} \quad \mathcal{A}^D[S_1, S_2] \leftarrow \emptyset$$

APPLICATION 2. SAMPLING

FILTERING THE GRAMMAR (for sequence alignments)

$$\mathcal{S} \leftarrow \begin{pmatrix} X \\ Y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} X \\ - \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ Y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \varepsilon$$

$$\mathcal{S}^D \leftarrow \begin{pmatrix} X \\ Y \end{pmatrix} \boxed{\mathcal{S}} \oplus \begin{pmatrix} - \\ Y \end{pmatrix} \boxed{\mathcal{S}^D} \oplus \varepsilon$$

Fix two sequences S_1 and S_2 -
 $\mathcal{A}[S_1, S_2] :=$ set of alignments between S_1 and S_2 ??

If $S_1 = X S_1'$ and $S_2 = Y S_2'$,

then

$$\mathcal{A}[S_1, S_2] \leftarrow \begin{pmatrix} X \\ Y \end{pmatrix} \boxed{\mathcal{A}[S_1', S_2']} \oplus \begin{pmatrix} X \\ - \end{pmatrix} \boxed{\mathcal{A}[S_1', S_2]} \oplus \begin{pmatrix} - \\ Y \end{pmatrix} \boxed{\mathcal{A}^D[S_1, S_2']}$$

$$\mathcal{A}^D[S_1, S_2] \leftarrow \begin{pmatrix} X \\ Y \end{pmatrix} \boxed{\mathcal{A}[S_1', S_2']} \oplus \begin{pmatrix} - \\ Y \end{pmatrix} \boxed{\mathcal{A}^D[S_1, S_2']}$$

APPLICATION 2. SAMPLING

FILTERING THE GRAMMAR (for sequence alignments)

If $S_1 = \varepsilon$ and $S_2 = \varepsilon$, then $\mathcal{A}[S_1, S_2] \leftarrow \varepsilon$; $\mathcal{A}^D[S_1, S_2] \leftarrow \varepsilon$

If $S_1 = XS'_1$ and $S_2 = \varepsilon$,
then $\mathcal{A}[S_1, S_2] \leftarrow \begin{pmatrix} X \\ - \end{pmatrix} \boxed{\mathcal{A}[S'_1, \varepsilon]}$ $\mathcal{A}^D[S_1, S_2] \leftarrow \phi$

If $S_1 = \varepsilon$ and $S_2 = YS'_2$,
then $\mathcal{A}[S_1, S_2] \leftarrow \begin{pmatrix} - \\ Y \end{pmatrix} \boxed{\mathcal{A}^D[\varepsilon, S'_2]}$ $\mathcal{A}^D[S_1, S_2] \leftarrow \begin{pmatrix} - \\ Y \end{pmatrix} \boxed{\mathcal{A}^D[\varepsilon, S'_2]}$

If $S_1 = XS'_1$ and $S_2 = YS'_2$,
then $\mathcal{A}[S_1, S_2] \leftarrow \begin{pmatrix} X \\ Y \end{pmatrix} \boxed{\mathcal{A}[S'_1, S'_2]} \oplus \begin{pmatrix} X \\ - \end{pmatrix} \boxed{\mathcal{A}[S'_1, S_2]} \oplus \begin{pmatrix} - \\ Y \end{pmatrix} \boxed{\mathcal{A}^D[S_1, S'_2]}$
 $\mathcal{A}^D[S_1, S_2] \leftarrow \begin{pmatrix} X \\ Y \end{pmatrix} \boxed{\mathcal{A}[S'_1, S'_2]} \oplus \begin{pmatrix} - \\ Y \end{pmatrix} \boxed{\mathcal{A}^D[S_1, S'_2]}$

APPLICATION 2. SAMPLING

COMPUTING THE PARTITION FUNCTION (dyn. prog.)

If $S_1 = \varepsilon$ and $S_2 = \varepsilon$, then $Z[S_1, S_2] \leftarrow 1$; $Z^D[S_1, S_2] \leftarrow 1$

If $S_1 = XS'_1$ and $S_2 = \varepsilon$,
then $Z[S_1, S_2] \leftarrow e^{-\frac{1}{k_x}} Z[S'_1, \varepsilon]$ $Z^D[S_1, S_2] \leftarrow 0$

If $S_1 = \varepsilon$ and $S_2 = YS'_2$,
then $Z[S_1, S_2] \leftarrow e^{-\frac{1}{k_x}} Z^D[\varepsilon, S'_2]$ $Z^D[S_1, S_2] \leftarrow e^{-\frac{1}{k_x}} Z^D[\varepsilon, S'_2]$

If $S_1 = XS'_1$ and $S_2 = YS'_2$,
then $Z[S_1, S_2] \leftarrow 1 \times Z[S'_1, S'_2] + e^{-\frac{1}{k_x}} Z[S'_1, S_2] + e^{-\frac{1}{k_x}} Z^D[S_1, S'_2]$

$Z[S_1, S_2] \leftarrow 1 \times Z[S'_1, S'_2] + e^{-\frac{1}{k_x}} Z^D[S_1, S'_2]$

APPLICATION 2. SAMPLING

FINALLY CLASSIC BOLTZMANN GENERATION

If $S_1 = \varepsilon$ and $S_2 = \varepsilon$, then $\mathcal{P}[S_1, S_2] \leftarrow \varepsilon$; $\mathcal{P}^D[S_1, S_2] \leftarrow \varepsilon$

If $S_1 = X S'_1$ and $S_2 = \varepsilon$,
then $\mathcal{P}[S_1, S_2] \leftarrow \binom{X}{-} \boxed{A[S'_1, \varepsilon]}$ $\mathcal{P}^D[S_1, S_2] \leftarrow \phi$

If $S_1 = \varepsilon$ and $S_2 = Y S'_2$,
then $\mathcal{P}[S_1, S_2] \leftarrow \binom{-}{Y} \boxed{A^D[\varepsilon, S'_2]}$ $\mathcal{P}^D[S_1, S_2] \leftarrow \binom{-}{Y} \boxed{A^D[\varepsilon, S'_2]}$

If $S_1 = X S'_1$ and $S_2 = Y S'_2$,
then $\mathcal{P}[S_1, S_2] \leftarrow \binom{X}{Y} \boxed{A[S'_1, S'_2]} \oplus \binom{X}{-} \boxed{A[S'_1, S_2]} \oplus \binom{-}{Y} \boxed{A^D[S_1, S'_2]}$
 $\mathcal{P}^D[S_1, S_2] \leftarrow \binom{X}{Y} \boxed{A[S'_1, S'_2]} \oplus \binom{-}{Y} \boxed{A^D[S_1, S'_2]}$

SAMPLING

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 $O(\bar{n}_1 \bar{n}_2)$.

SAMPLING

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

- Sampling a neighborhood of a tree?
- Existence of easier decompositions?
- Alignment problem for arc-annotated sequences?



