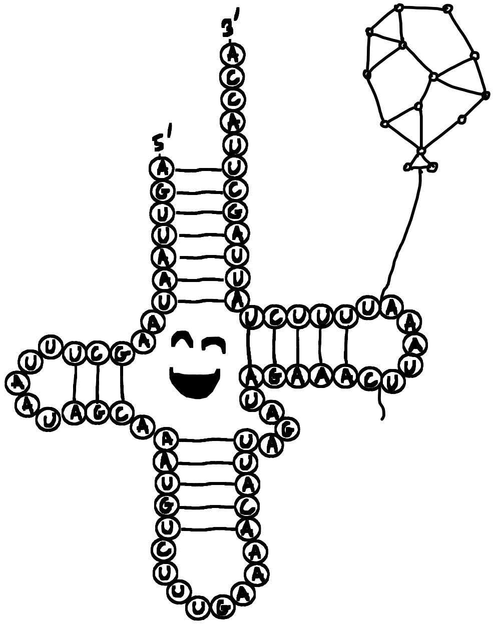
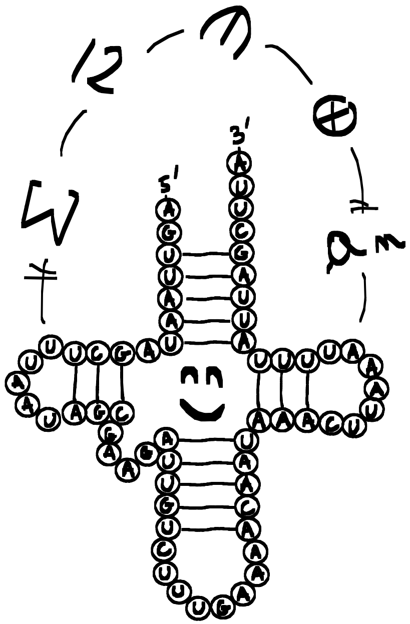
GENERATING TREE ALIGNMENTS

HOW COMBINATORICS CAN HELP BIOINFORMATICS

Julien COURTIEL (PIMS/Univ. of British Columbia, Vancouver)

2016 SFU Symposium on Mathematics and Computation

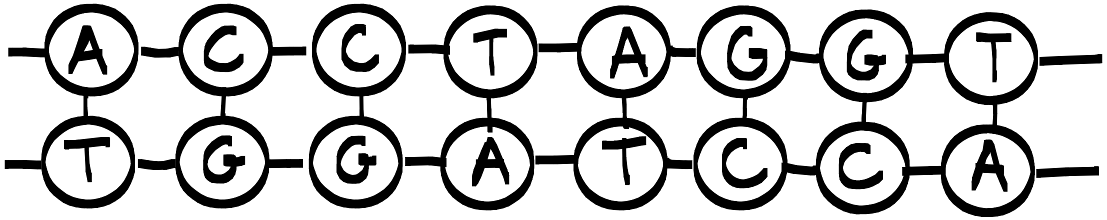




Co-authors: Cedric CHAUVE (Simon Fraser University, Vancouver)
Yann PONTY (CNRS/LIX, Ecole Polytechnique, Inria Saclay)

WHAT IS RNA?

DNA
the code

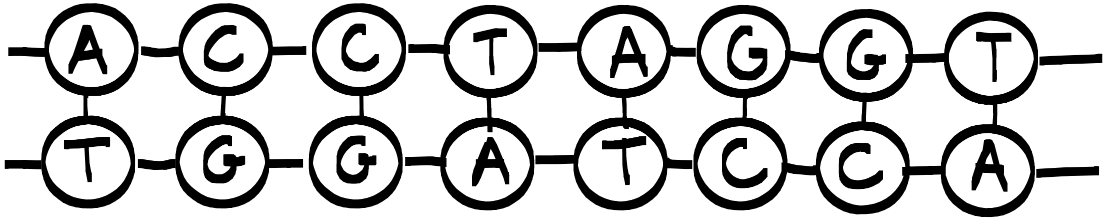


WHAT IS RNA?

DNA
the code

↓

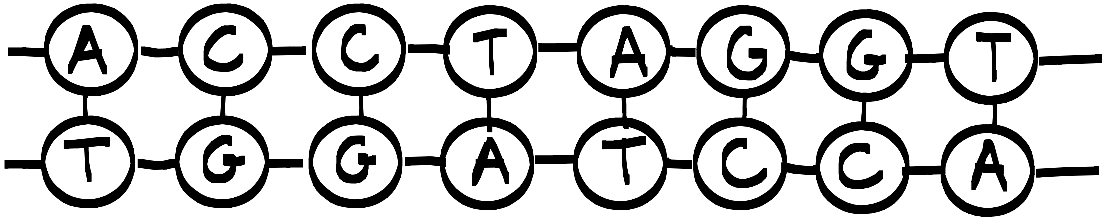
RNA



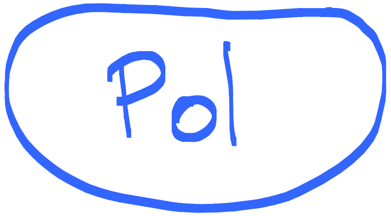
WHAT IS RNA?

BETTER
CALL POL

DNA
the code



↓



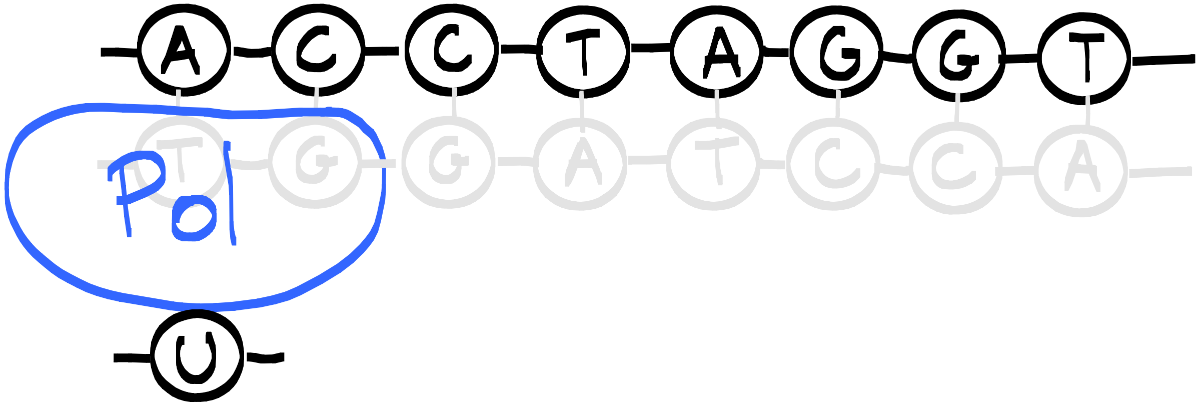
RNA

WHAT IS RNA?

DNA
the code

↓

RNA

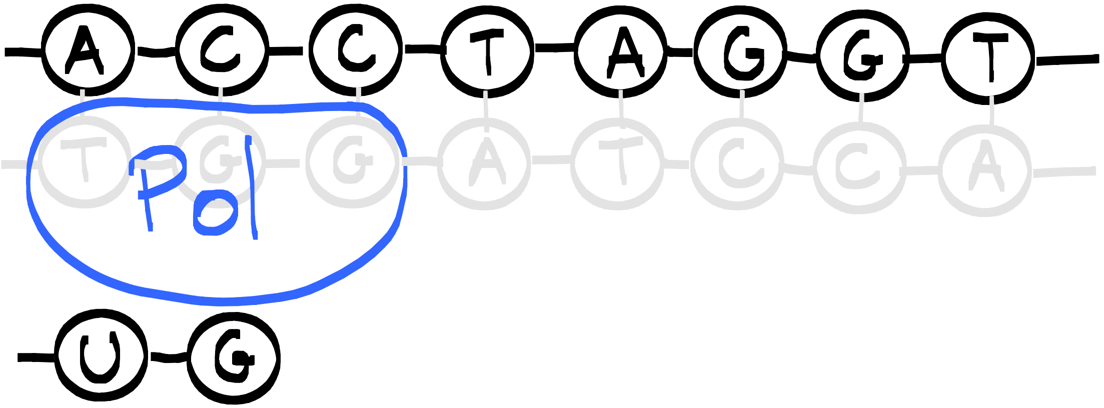


WHAT IS RNA?

DNA
the code

↓

RNA

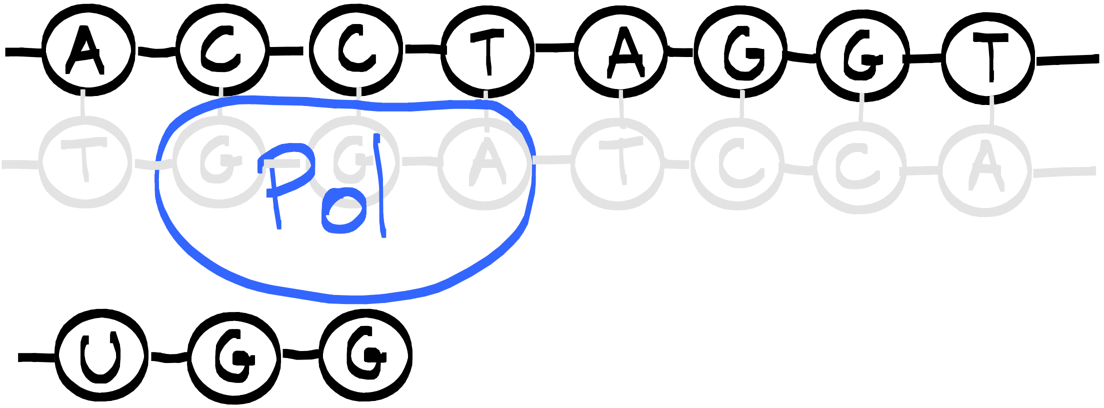


WHAT IS RNA?

DNA
the code

↓

RNA

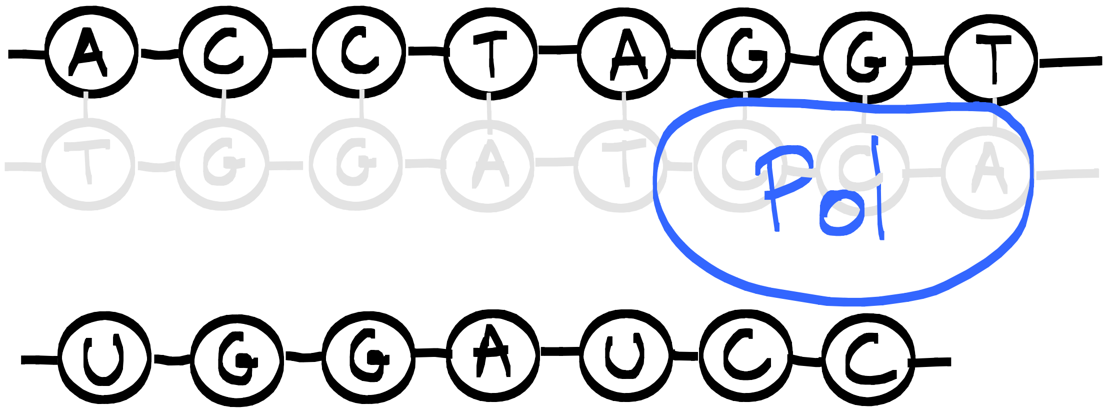


WHAT IS RNA?

DNA
the code

↓

RNA

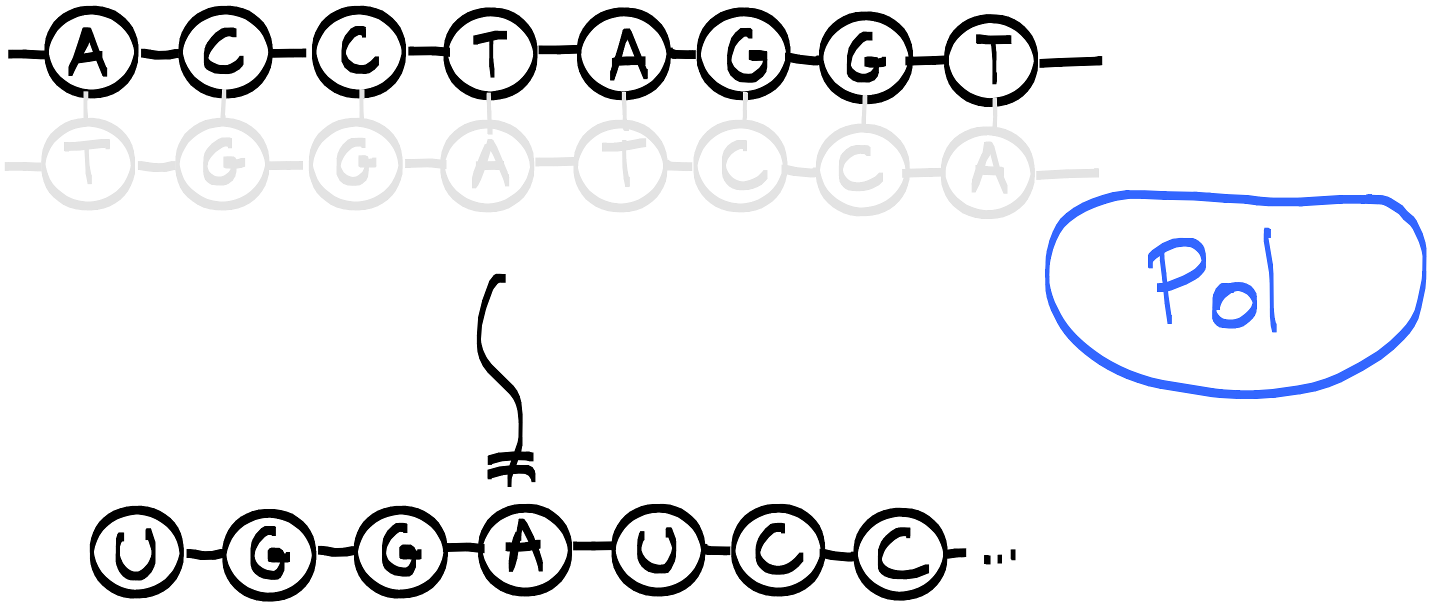


WHAT IS RNA?

DNA
the code

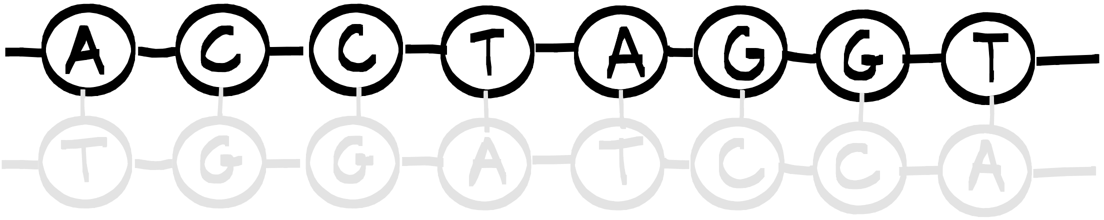
↓

RNA



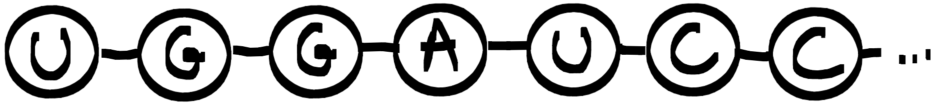
WHAT IS RNA?

DNA
the code



↓

RNA



↓

proteins

WHAT IS RNA?

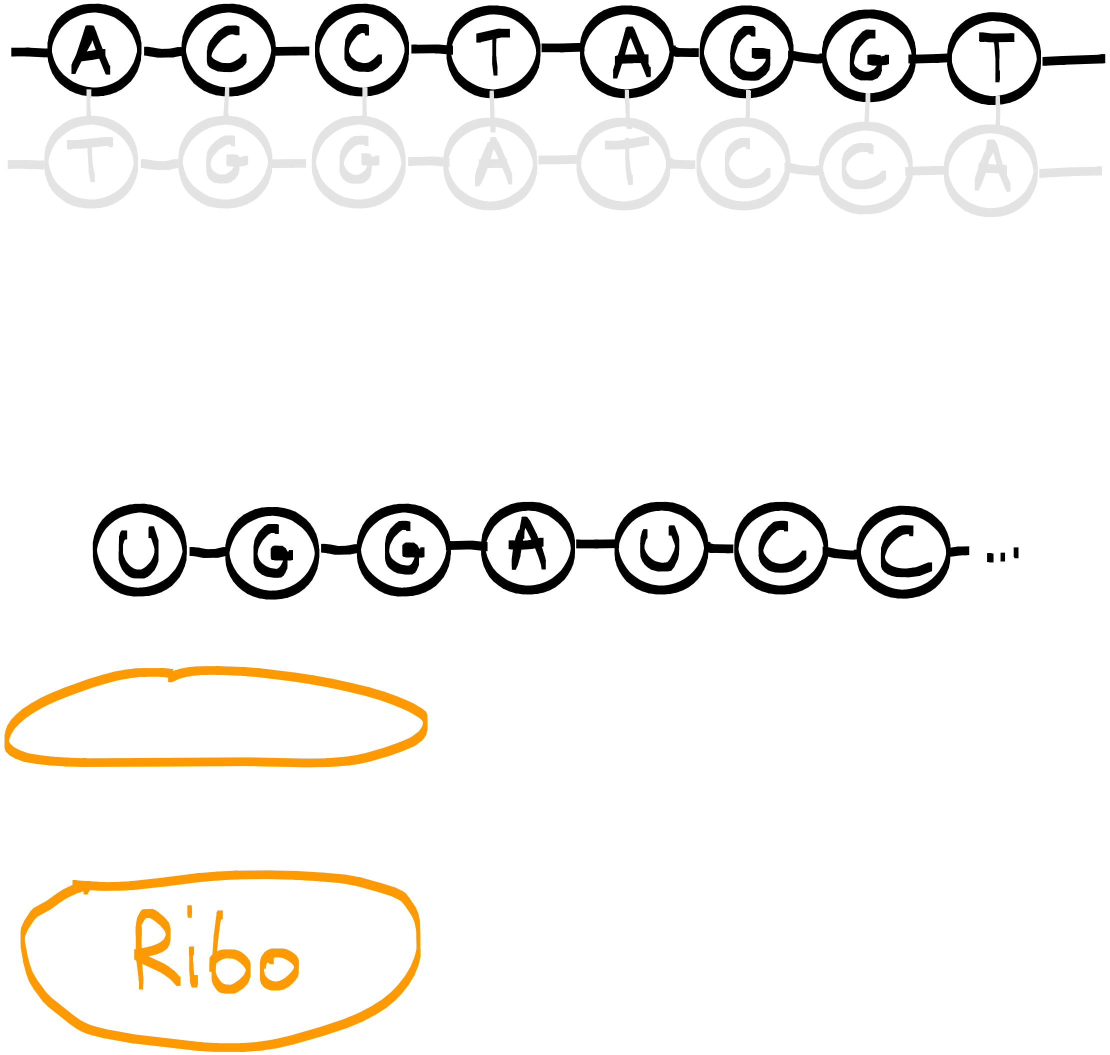
DNA
the code

↓

RNA

↓

proteins



WHAT IS RNA?

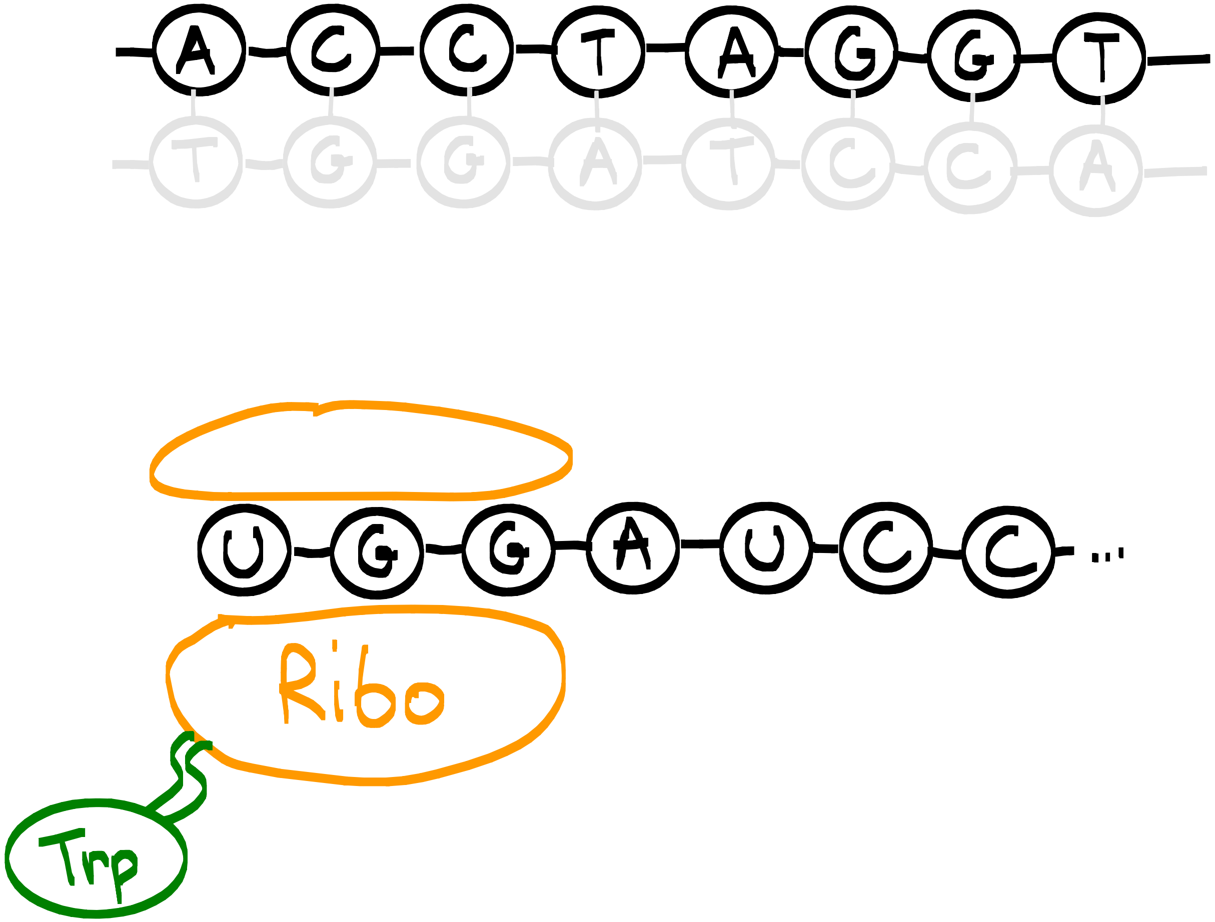
DNA
the code

↓

RNA

↓

proteins



WHAT IS RNA?

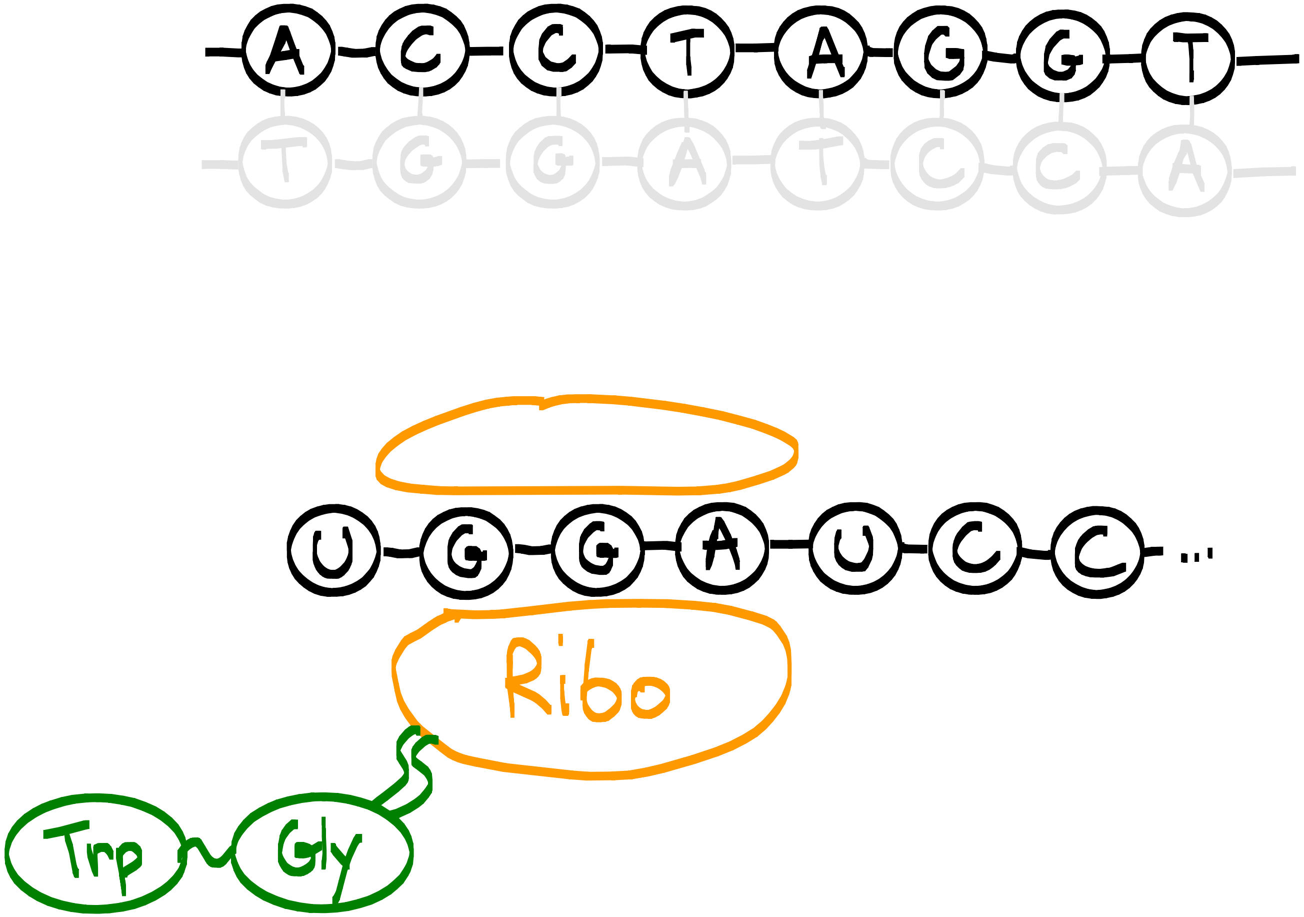
DNA
the code

↓

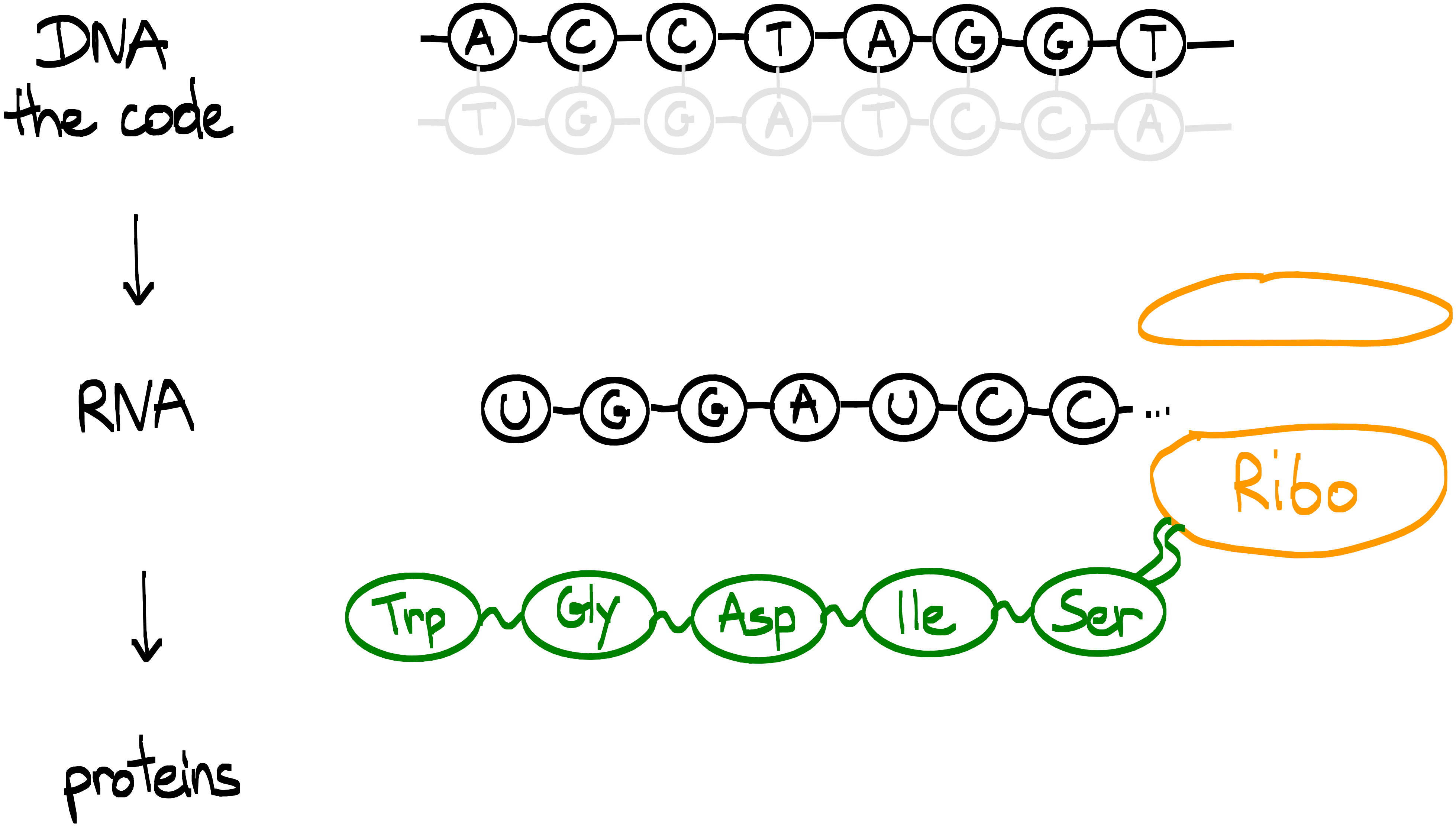
RNA

↓

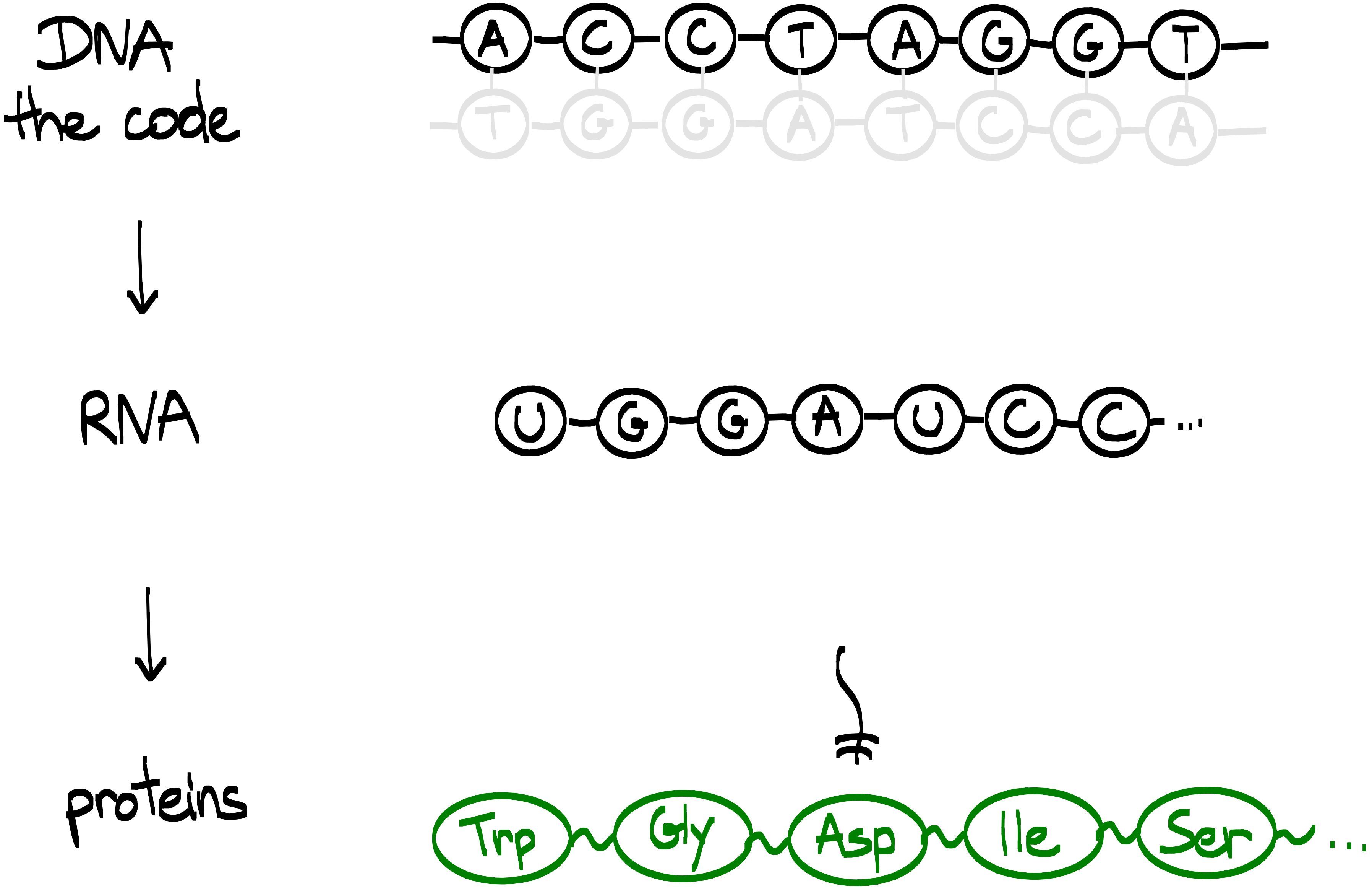
proteins



WHAT IS RNA?


DNA
the code
A C C T A G G T
T G G A T C C A
RNA
U G G A U C C ...
Ribo
Trp Gly Asp Ile Ser
proteins

WHAT IS RNA?


DNA
the code
A C C T A G G T
T G G A T C C A
RNA
U G G A U C C ...
proteins
Trp Gly Asp Ile Ser ...

Classic dogma

WHAT IS RNA?

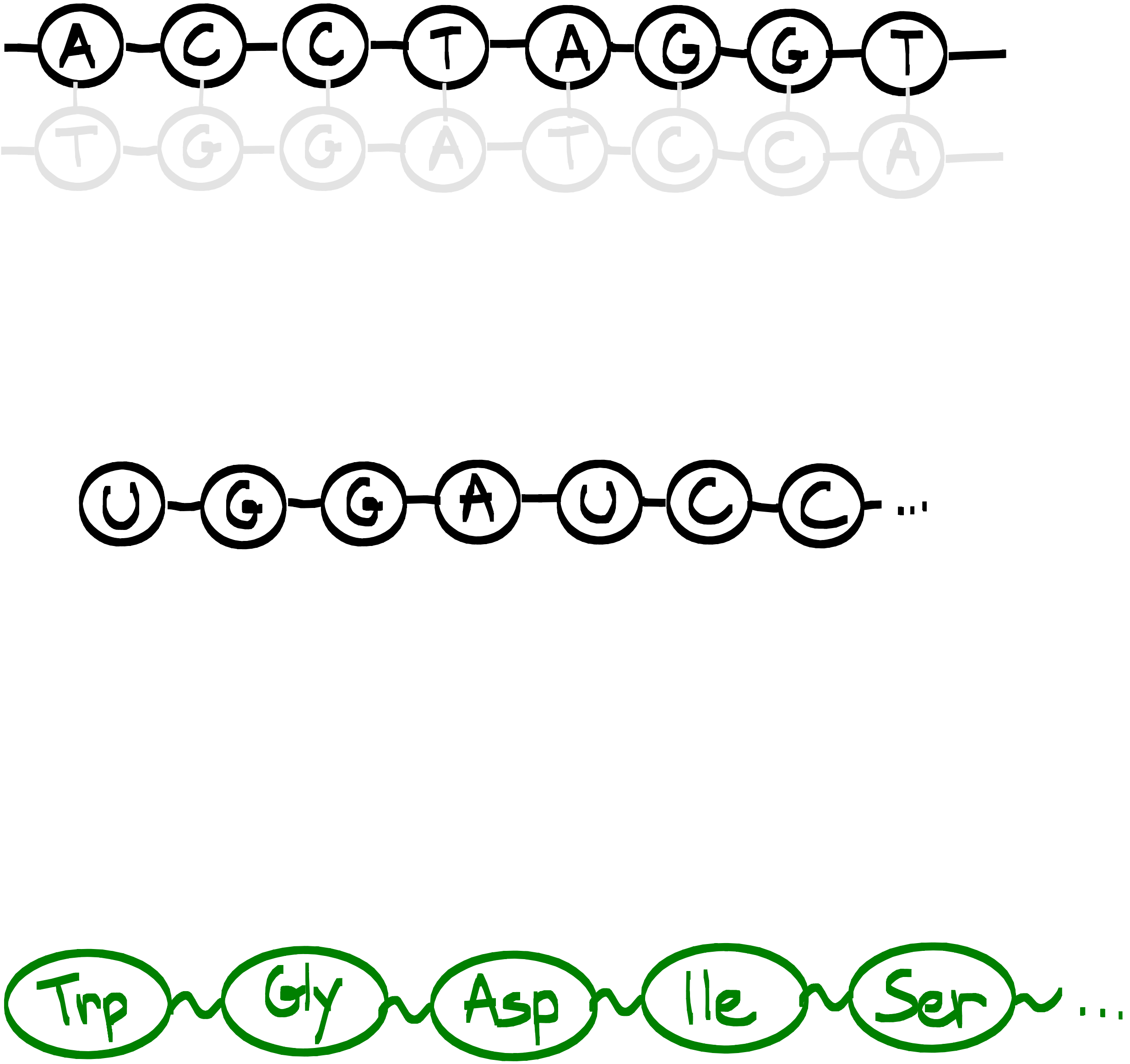
DNA
the code

↓

RNA
the messenger

↓

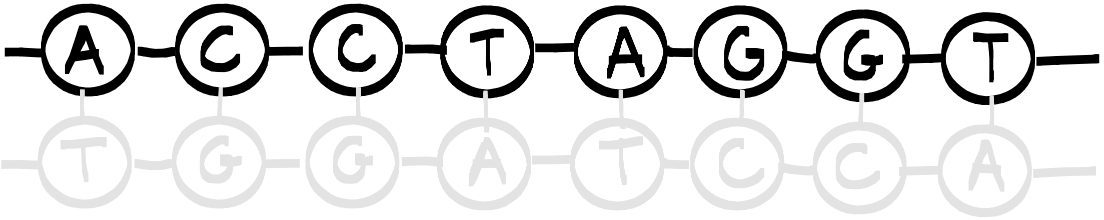
proteins
the machine



Classic dogma

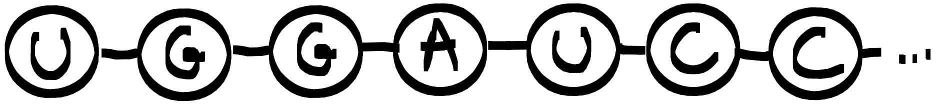
WHAT IS RNA?

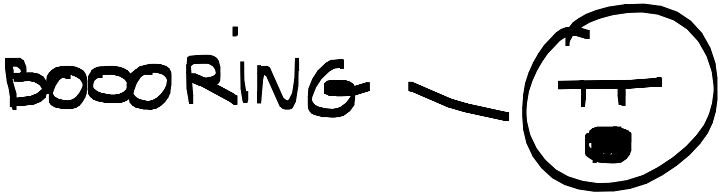
DNA
the code



↓

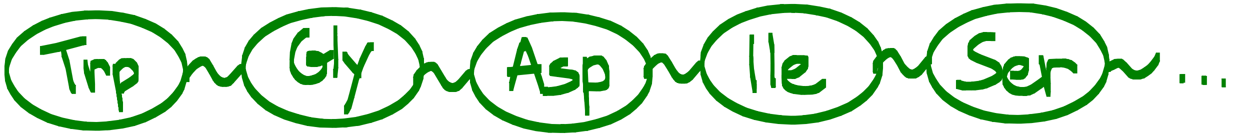
RNA
the messenger?





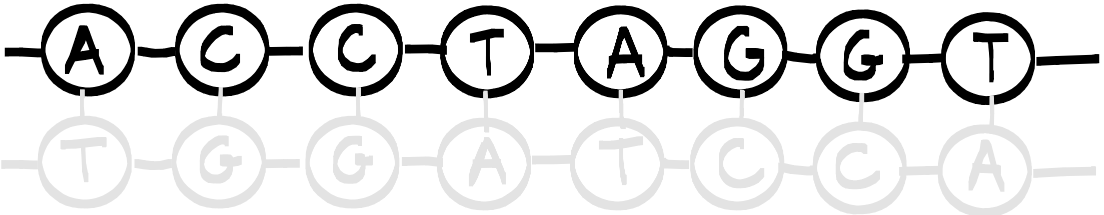
↓

proteins
the machine



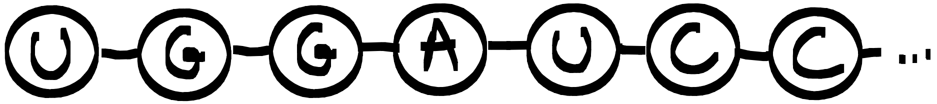
WHAT IS RNA?

DNA
the code



↓

RNA
the messenger

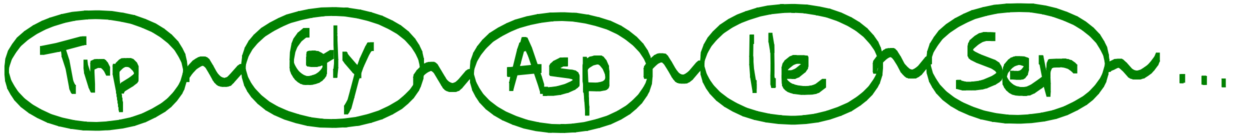


actually also... translator, regulator, enzyme, catalyst...

↑ recent breakthrough

↓

proteins
the machine

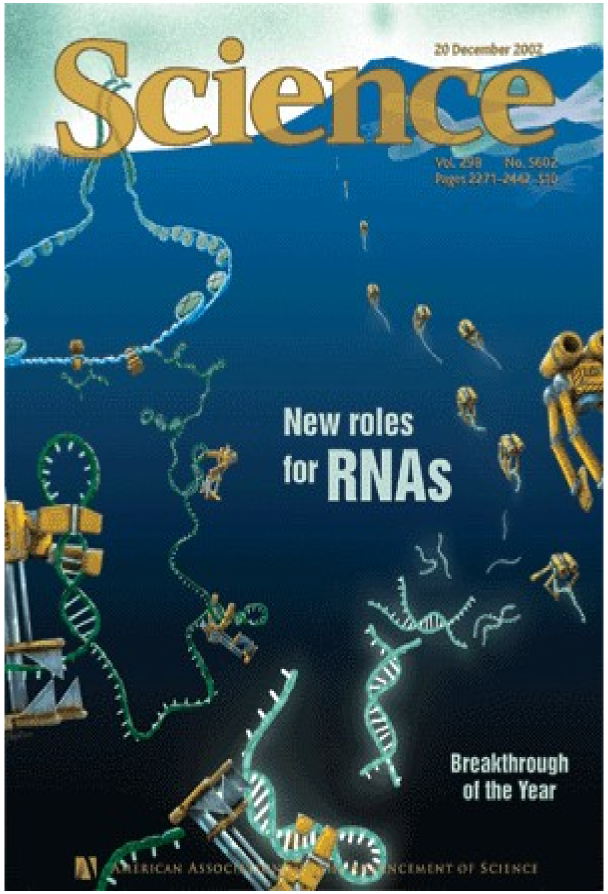


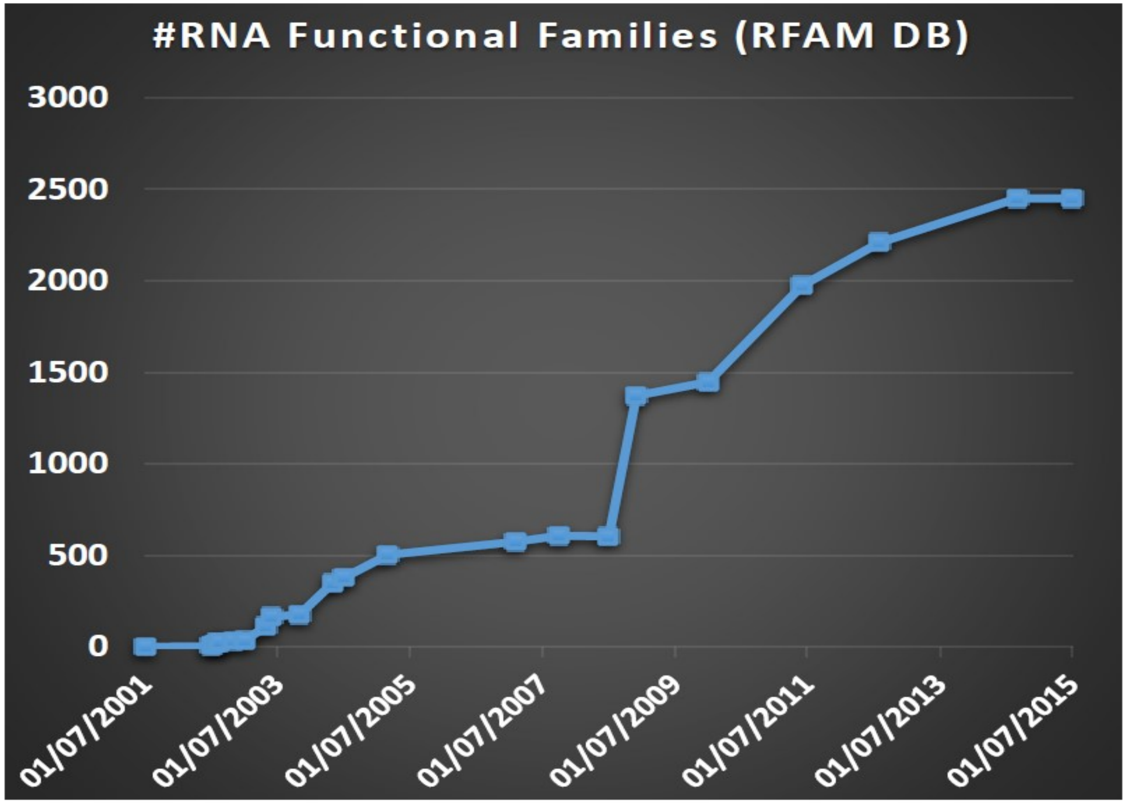
WHAT IS RNA?

RNA
the messenger

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

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

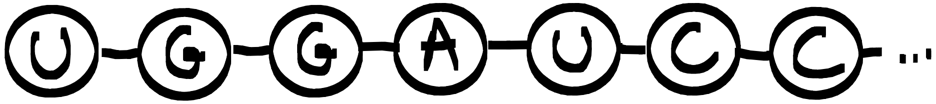




So what is RNA ?

RNA is

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

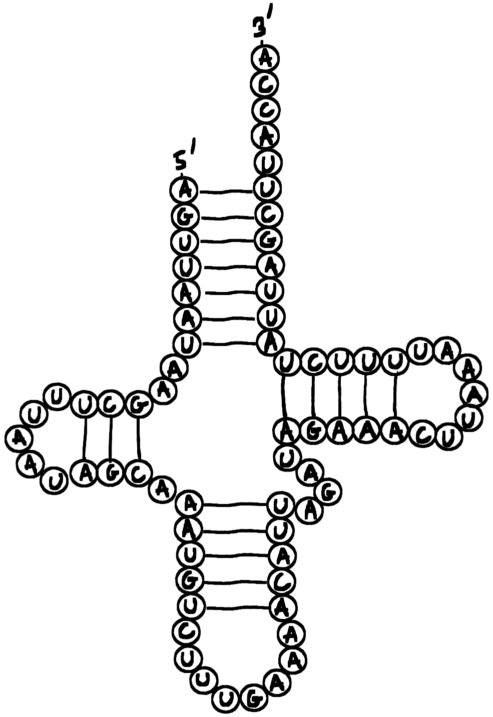


So what is RNA?

RNA is

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

... stabilized by hydrogen bonds..



3'
A C C A U U C G A U U A
5'
A G U U A A U
A A G C U U U A A U A G C A
A A U G U C U U U G A A A A C A U U
A U A G A G A A A A C U U A A U U U U C U

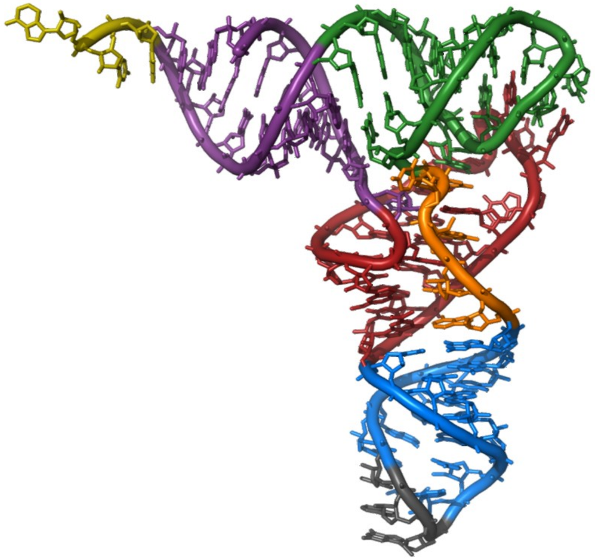
So what is RNA?

RNA is

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

... stabilized by hydrogen bonds..

...which folds onto itself.



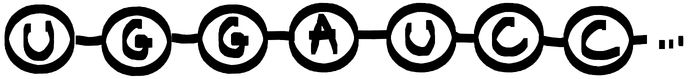
So what is RNA?

RNA is

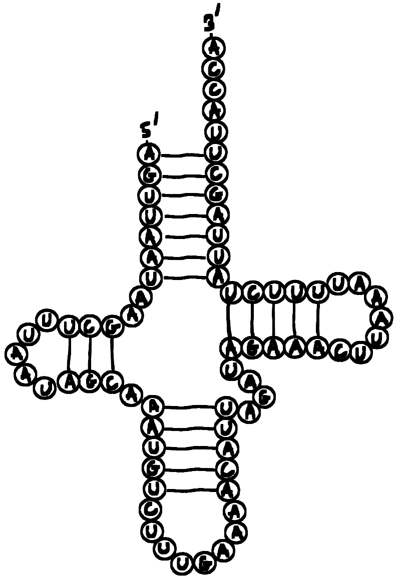
a single-stranded molecule (chain of nucleotides)...

... stabilized by hydrogen bonds..

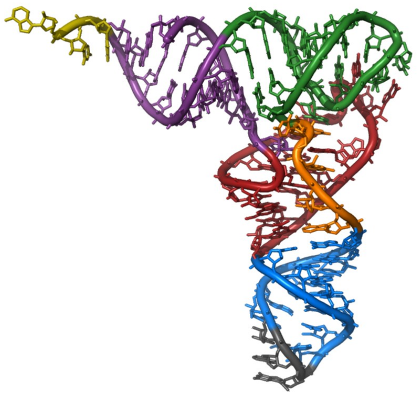
...which folds onto itself.



primary structure



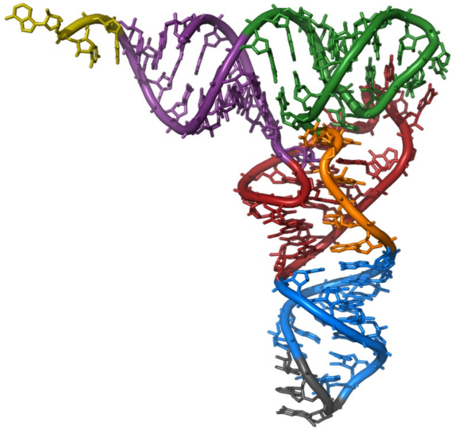
secondary structure

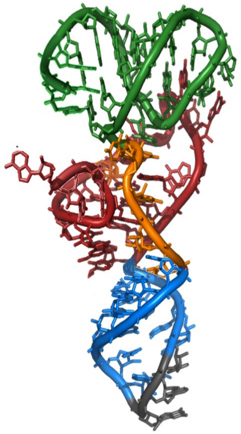


tertiary structure

RNA COMPARISON

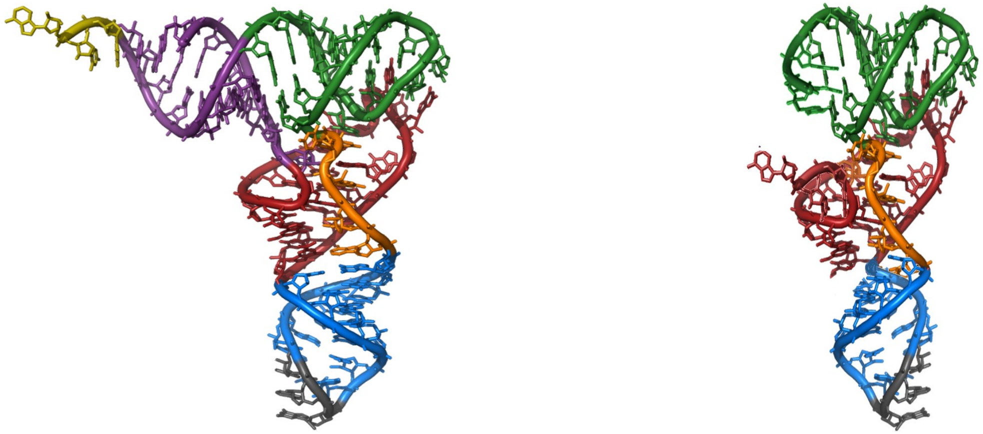
Interesting problem: evaluating similarity between two RNAs.

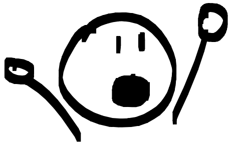




RNA COMPARISON

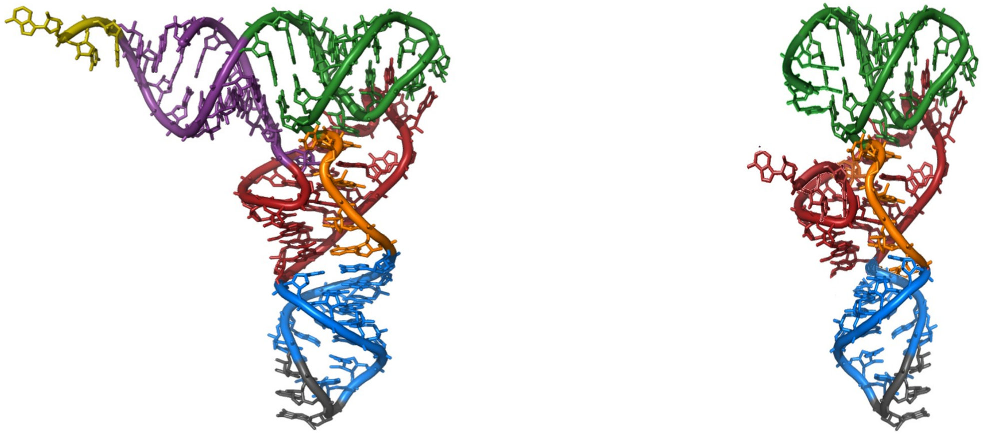
Interesting problem: evaluating similarity between two RNAs.

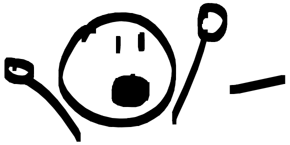


 – Why?

RNA COMPARISON

Interesting problem: evaluating similarity between two RNAs.




– Why?

Typical situation:

New RNA

ACAGUACC ...

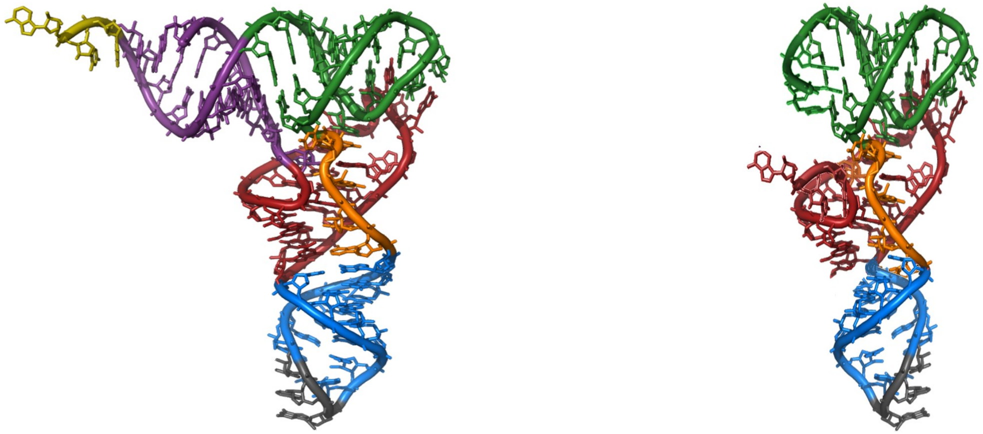
large database

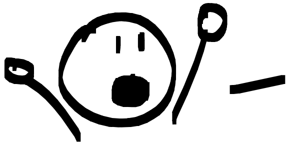
AUCCAG ... UUAGACC ...

CCAGC .. AAAGU ...

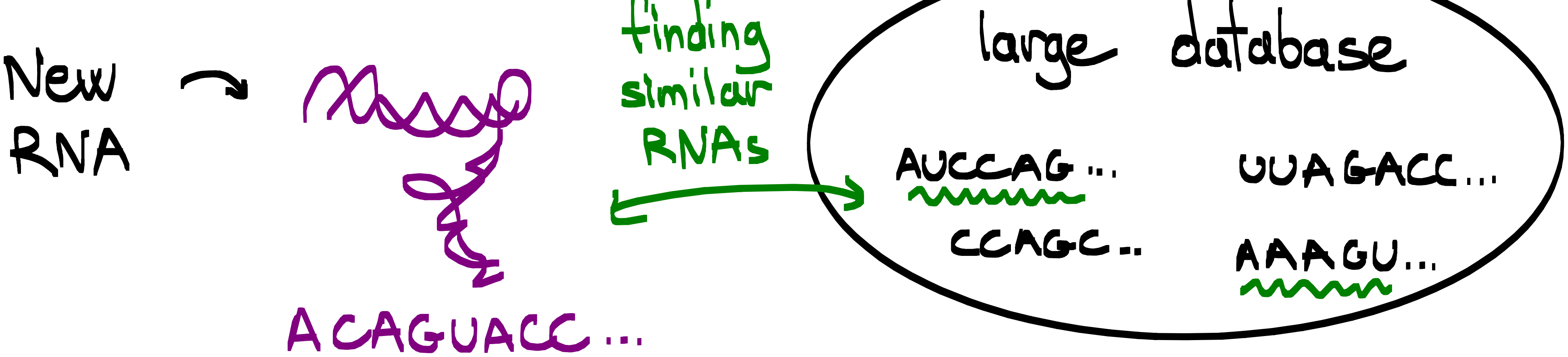
RNA COMPARISON

Interesting problem: evaluating similarity between two RNAs.



 – Why?

Typical situation:


New RNA

ACAGUACC...

finding similar RNAs

large database

AUCCAG...

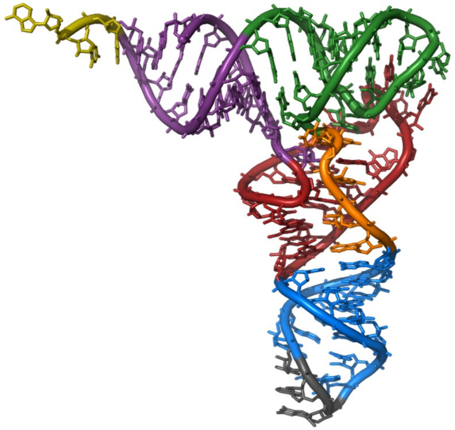
UUAGACC...

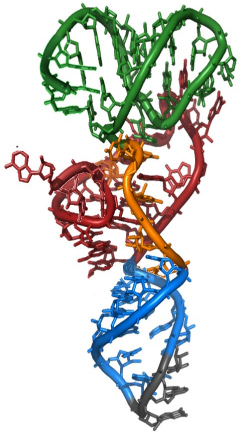
CCAGC..

AAAGU...

MOTIVATION: RNA COMPARISON

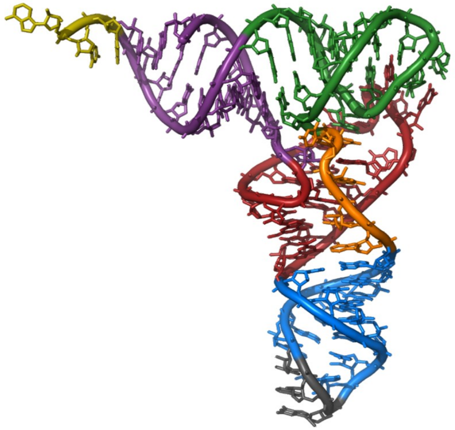
Question: how to measure similarity between two RNAs?

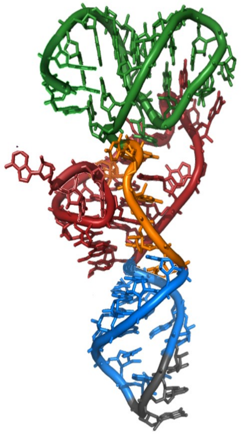




MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?





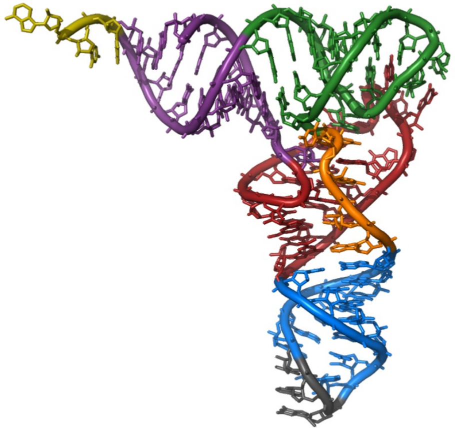
First idea: compare the primary structures.

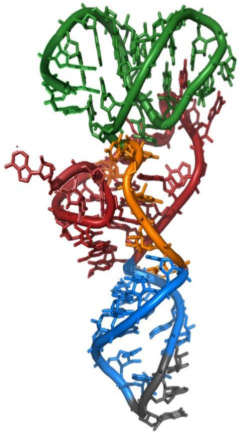
RNA 1: AUUCGAUUA...

RNA 2: ACCAUGAUUA...

MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?





First idea: compare the primary structures.

→ sequence alignment

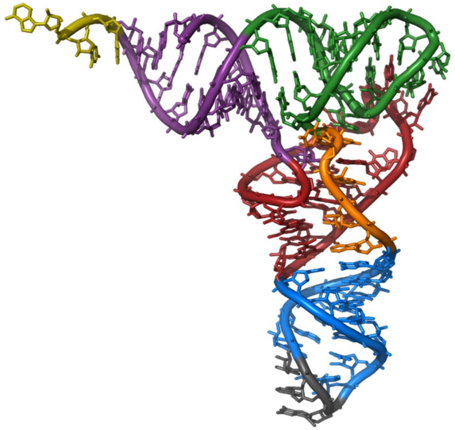
RNA 1: AUUCGAUUA...

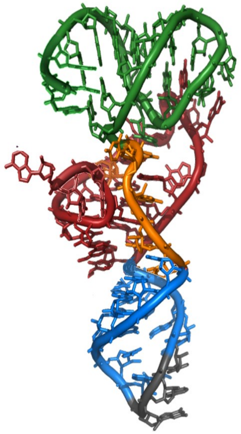
RNA 2: ACCAUGAUUA...

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

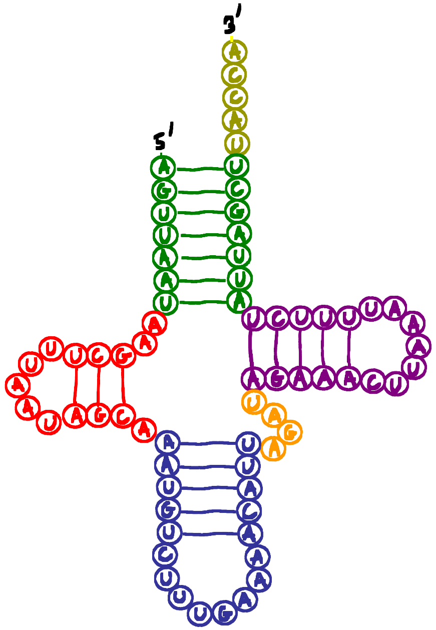
MOTIVATION: RNA COMPARISON

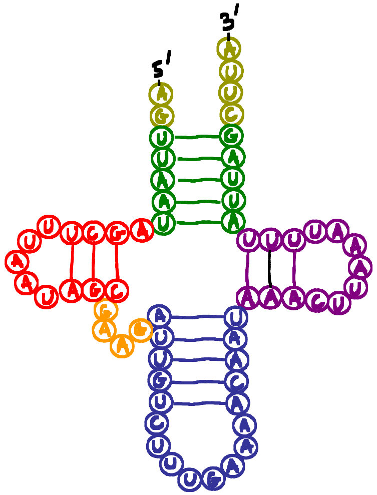
Question: how to measure similarity between two RNAs?





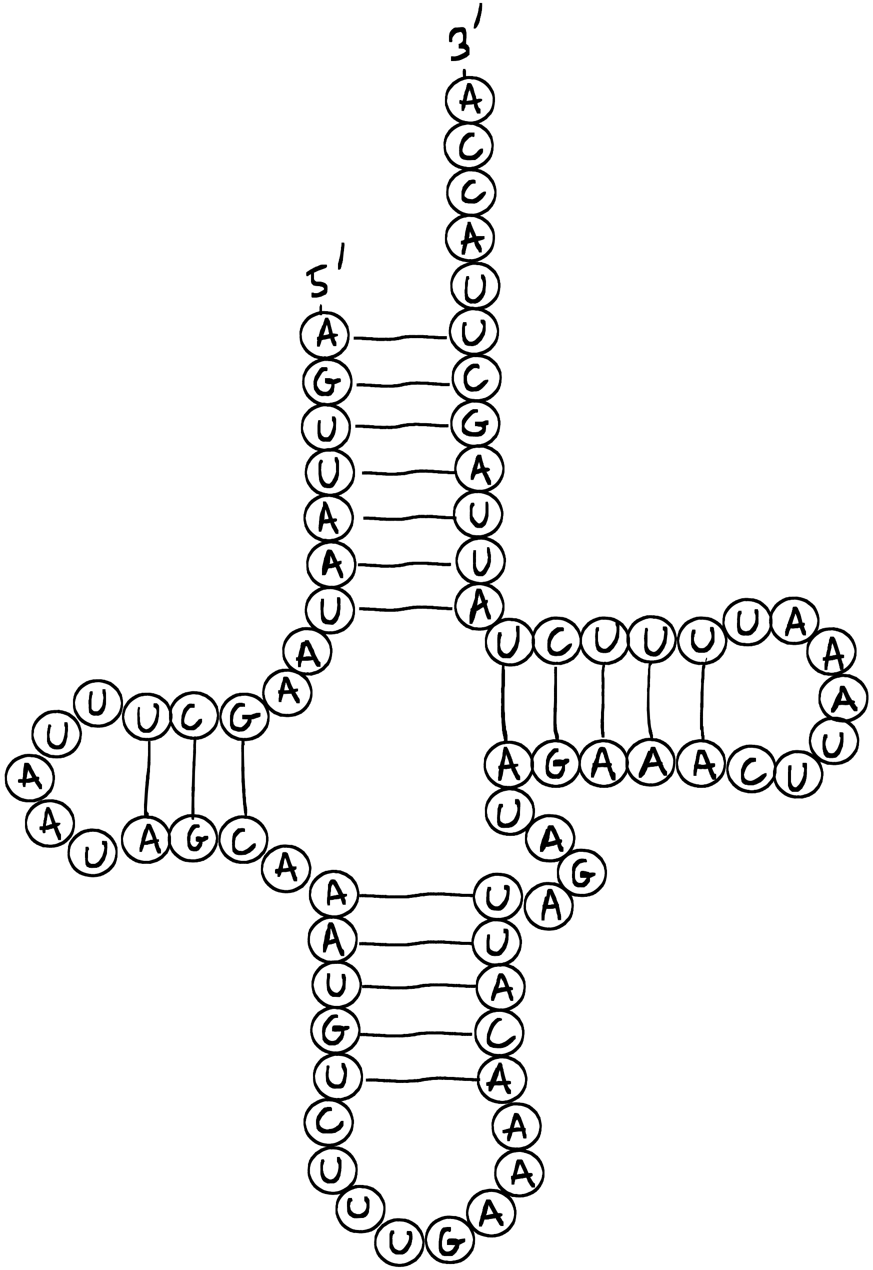
Second idea: compare secondary structures.



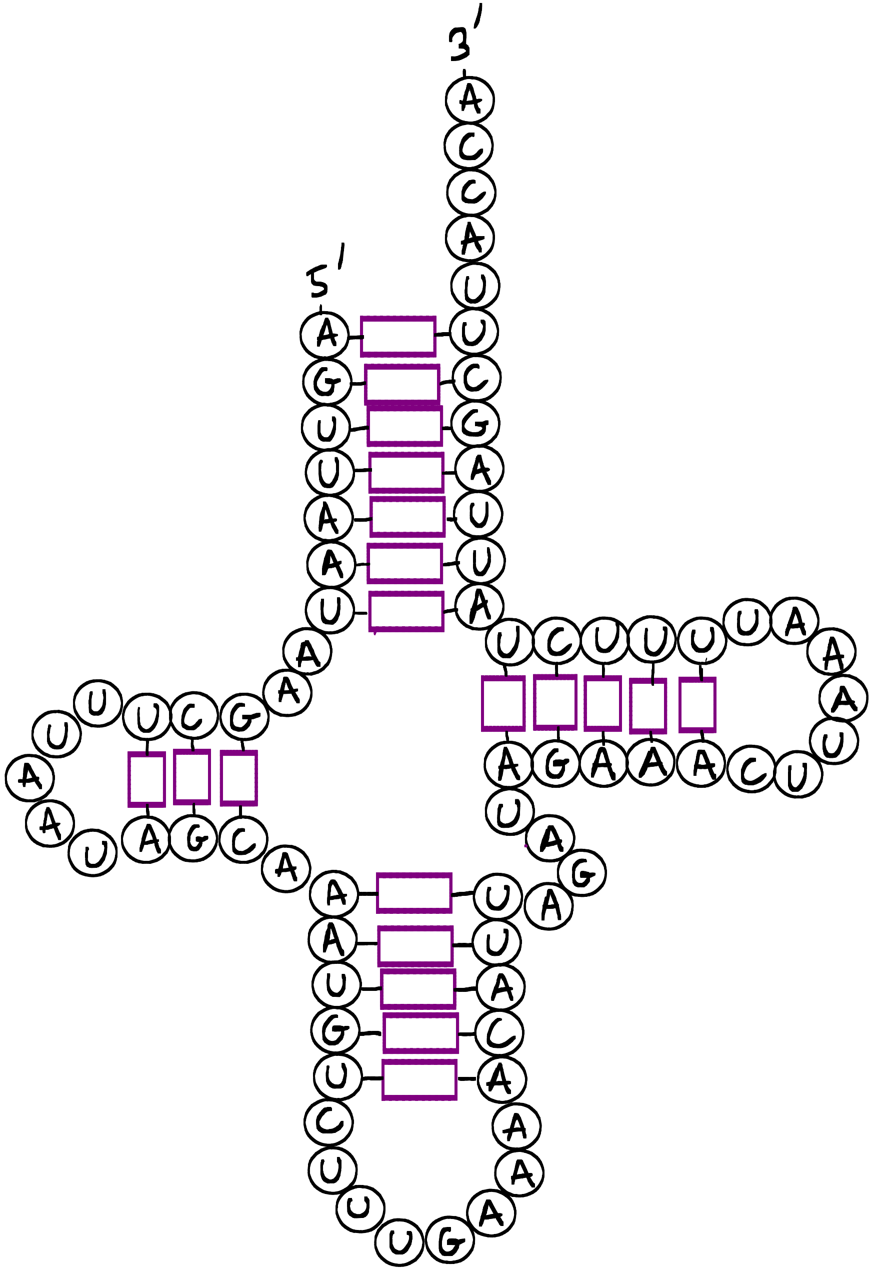


→ notion of tree alignment [Jiang, Wang, Zhang]

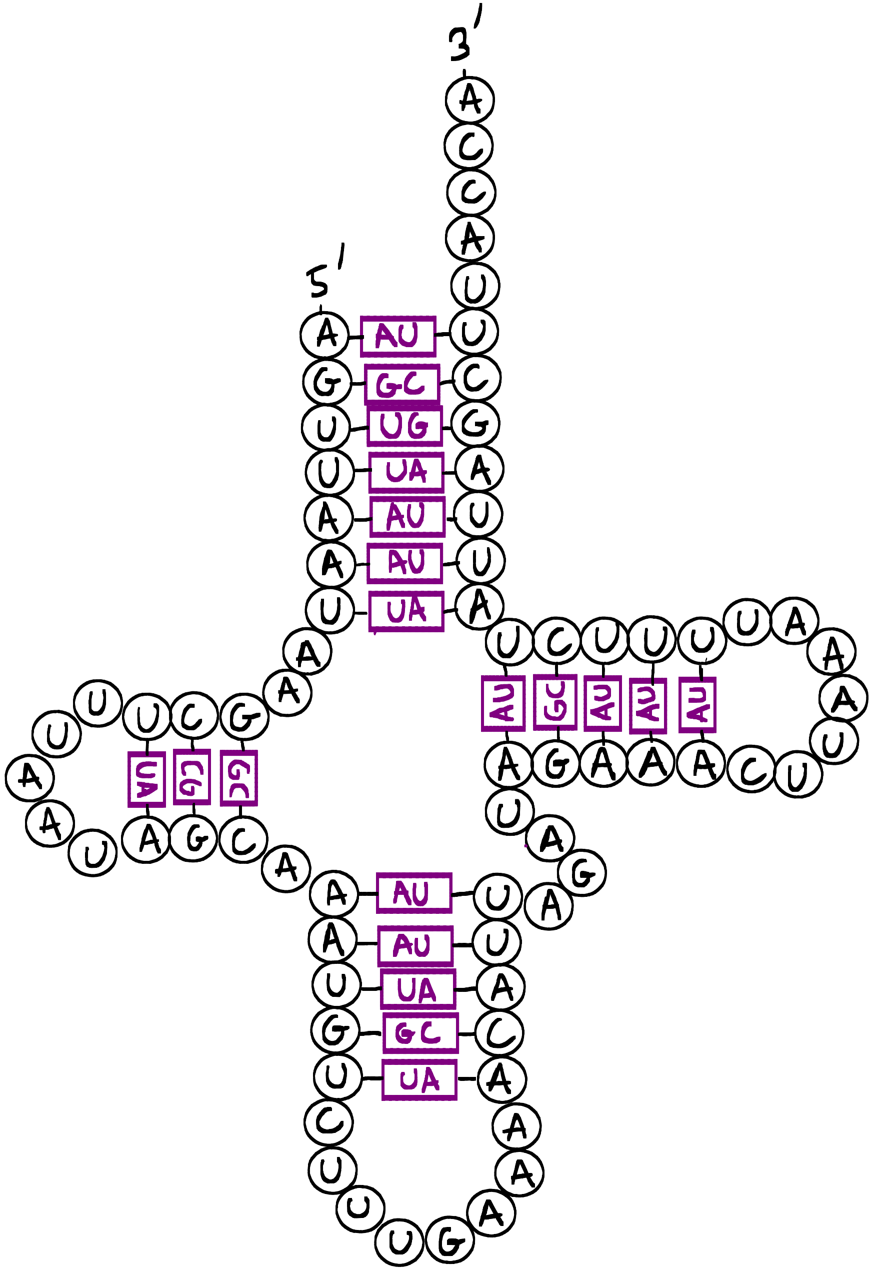
FROM SECONDARY STRUCTURES TO TREES


3'
5'
A G U U A A U A A G C U U U A A U A G C A A A U G U C U U U G A A A A C A U U A G A U A G A A A C U U A A A U U U C U A U U A G C U A C C A

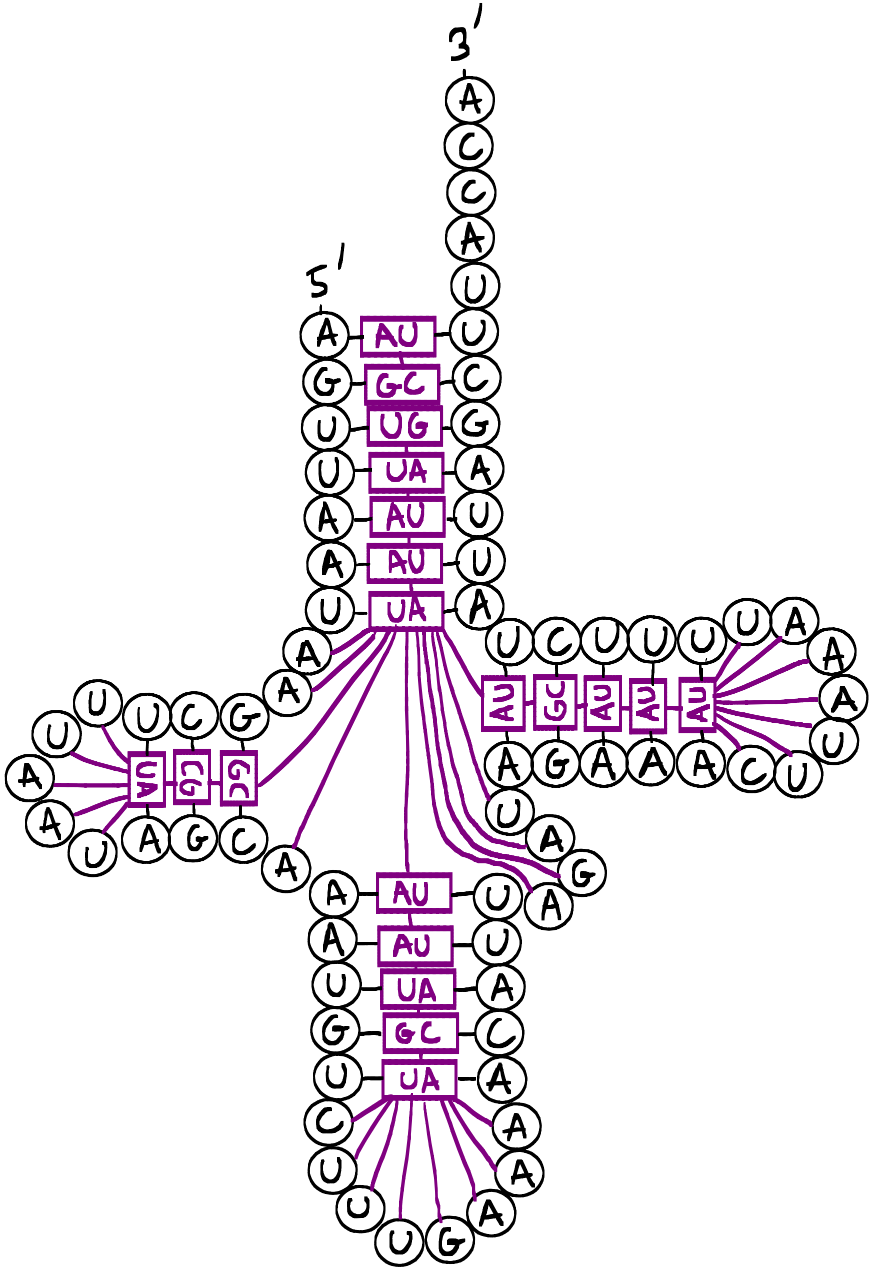
FROM SECONDARY STRUCTURES TO TREES



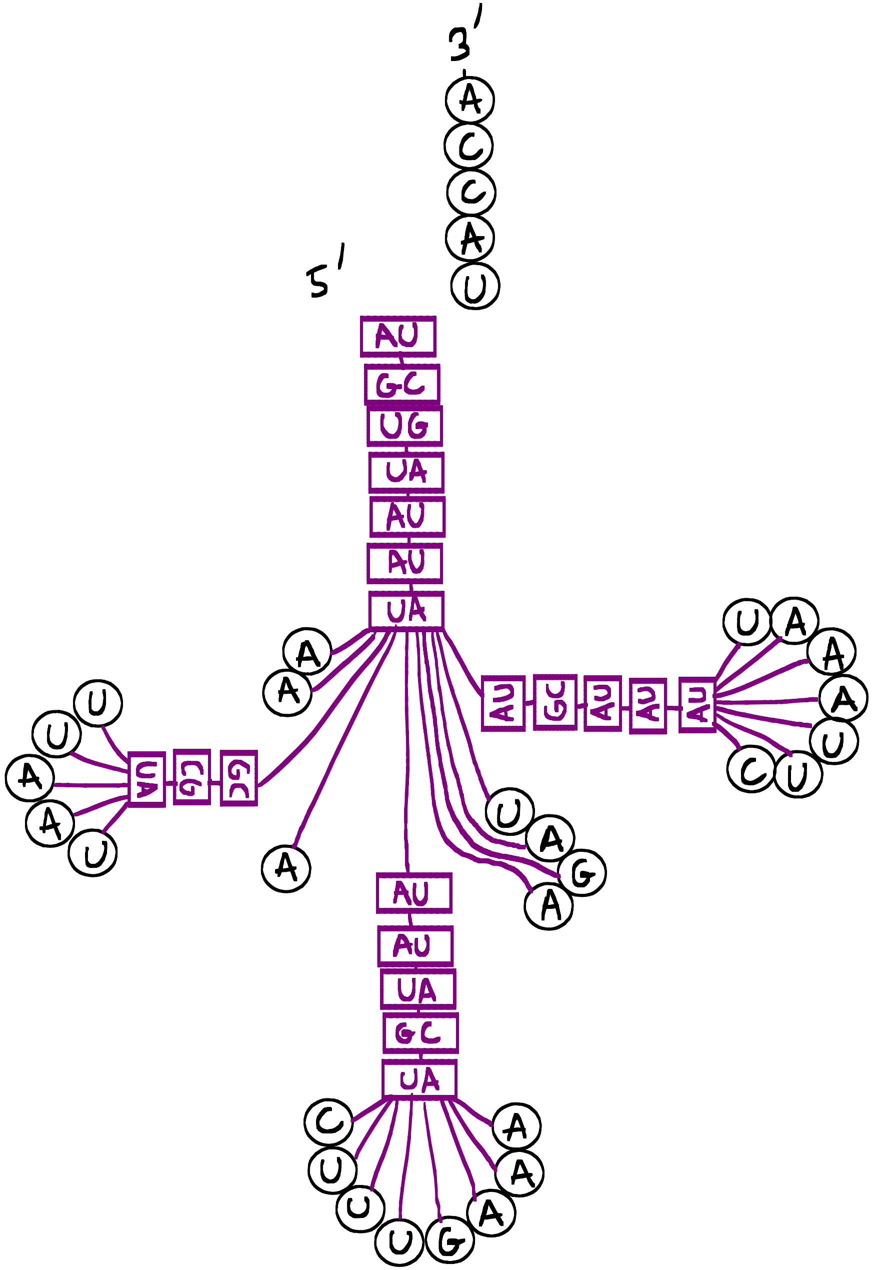
FROM SECONDARY STRUCTURES TO TREES



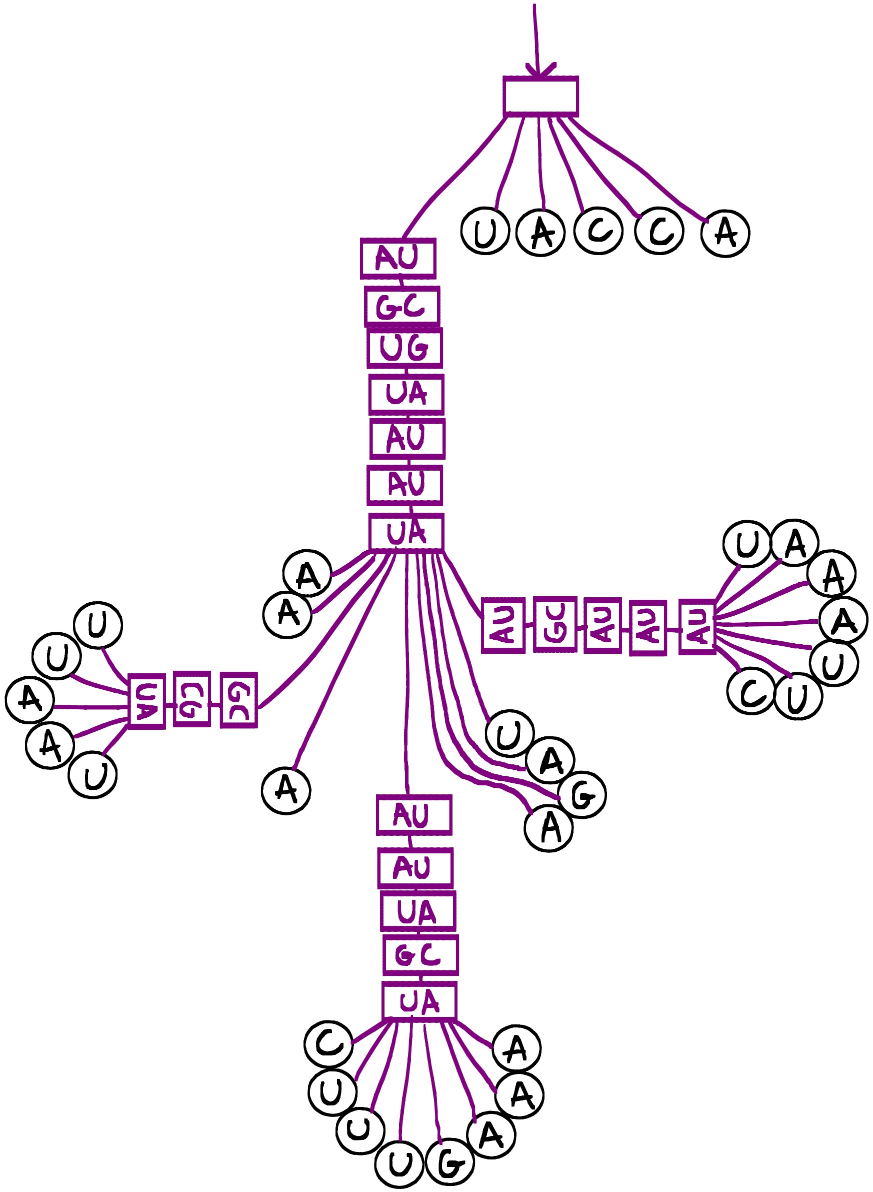
FROM SECONDARY STRUCTURES TO TREES



FROM SECONDARY STRUCTURES TO TREES

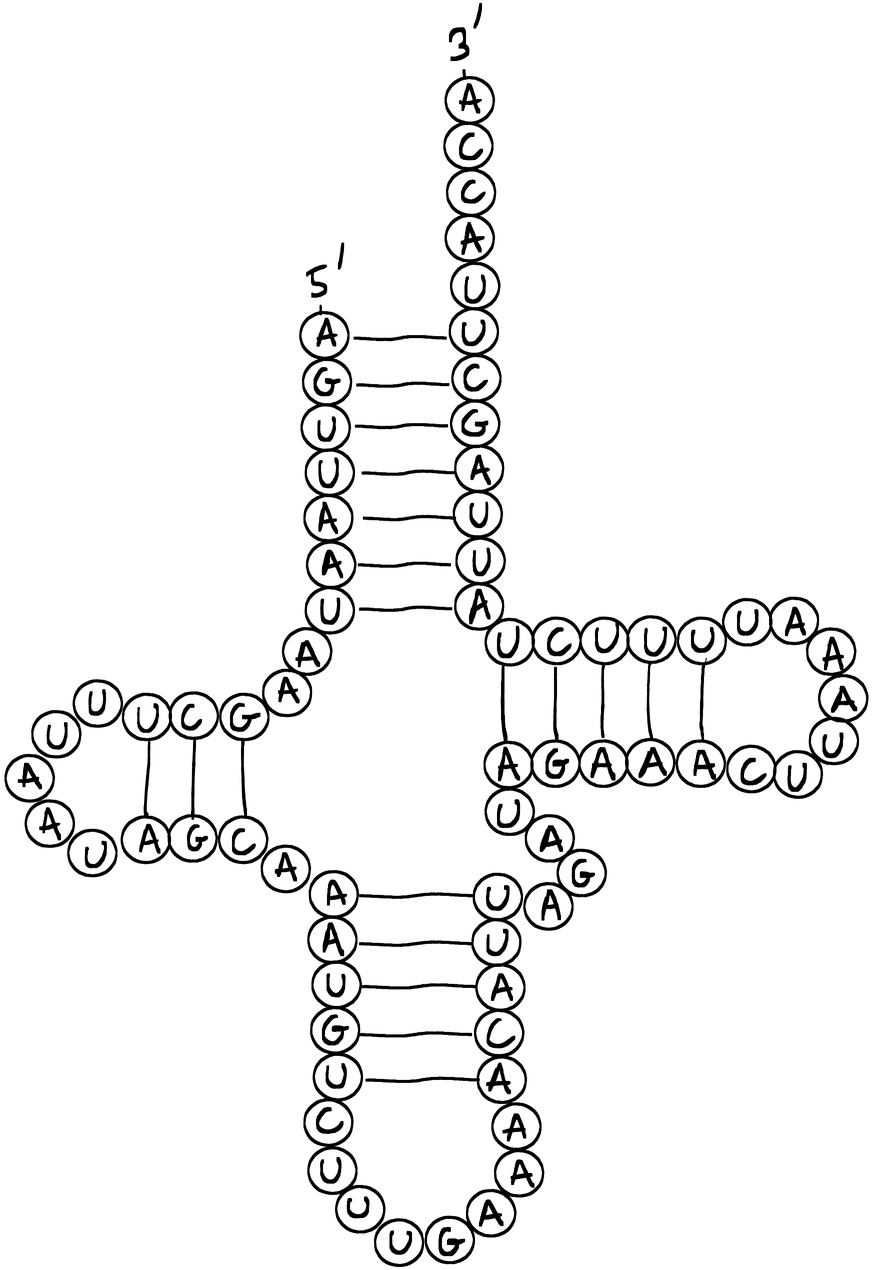


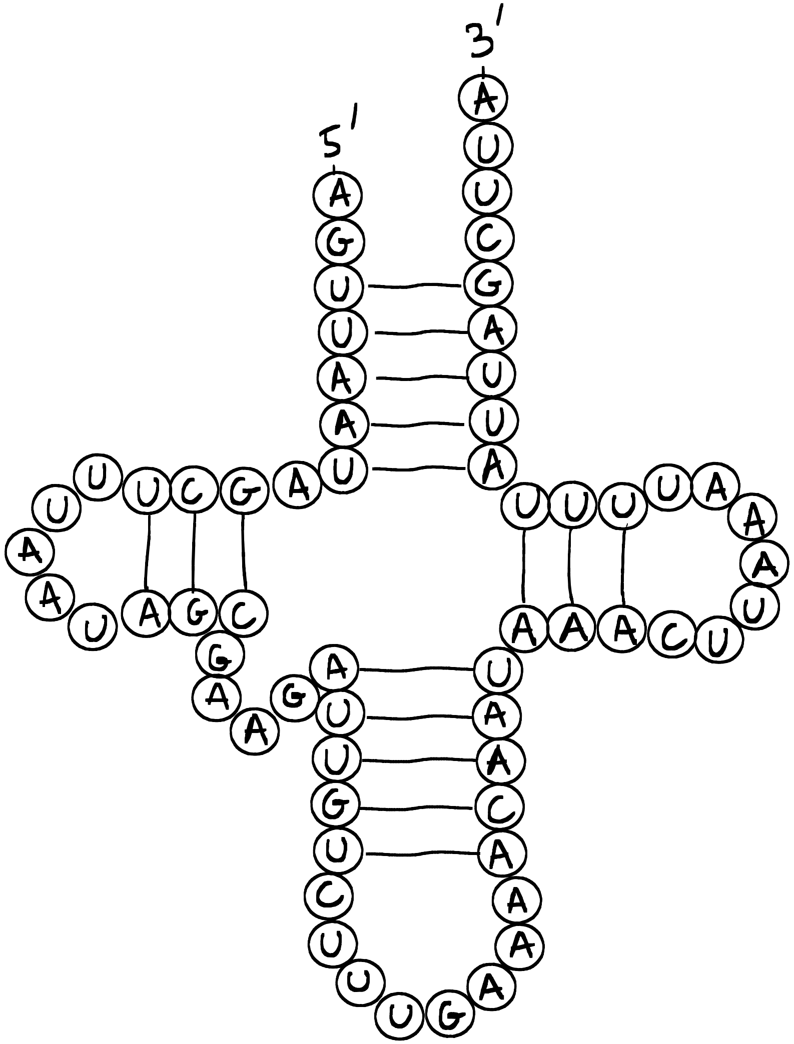
FROM SECONDARY STRUCTURES TO TREES



FROM SECONDARY STRUCTURES TO TREES

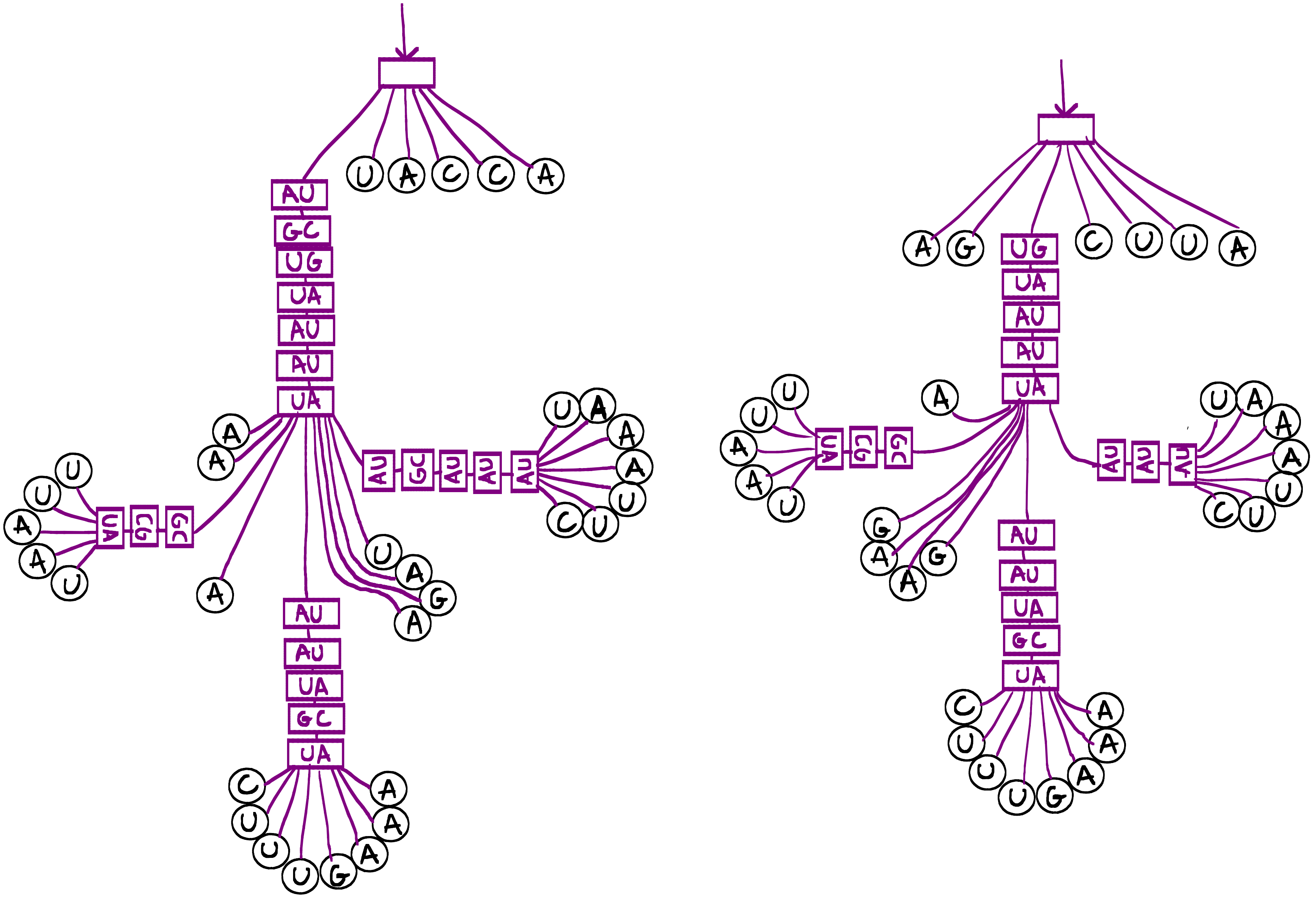
Objective: Align trees coming from RNA 2^{ary} structures





FROM SECONDARY STRUCTURES TO TREES

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



SEQUENCE ALIGNMENT

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

$$\underset{\substack{\uparrow\\ \text{match}}}{\binom{A}{A}}\binom{U}{-}\binom{-}{C}\underset{\substack{\uparrow\\ \text{insertion}}}{\binom{U}{-}}\binom{C}{C}\binom{-}{A}\underset{\substack{\uparrow\\ \text{deletion}}}{\binom{-}{U}}\binom{G}{G}\binom{A}{A}\binom{U}{U}\underbrace{\binom{A}{U}\binom{C}{A}}_{\substack{\uparrow\\ \text{mismatch}}}$$

SEQUENCE ALIGNMENT

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

$$\underset{\substack{\uparrow\\ \text{match}}}{\begin{pmatrix}A\\A\end{pmatrix}}\begin{pmatrix}U\\-\end{pmatrix}\begin{pmatrix}-\\C\end{pmatrix}\underset{\substack{\uparrow\\ \text{insertion}}}{\begin{pmatrix}U\\-\end{pmatrix}}\begin{pmatrix}C\\C\end{pmatrix}\begin{pmatrix}-\\A\end{pmatrix}\underset{\substack{\uparrow\\ \text{deletion}}}{\begin{pmatrix}-\\U\end{pmatrix}}\begin{pmatrix}G\\G\end{pmatrix}\begin{pmatrix}A\\A\end{pmatrix}\begin{pmatrix}U\\U\end{pmatrix}\underbrace{\begin{pmatrix}A\\U\end{pmatrix}\begin{pmatrix}C\\A\end{pmatrix}}_{\substack{\uparrow\\ \text{mismatch}}}\begin{matrix}\rightarrow \text{AUUCGAUAC}\\ \rightarrow \underset{\substack{\uparrow\\ \text{projections}}}{\text{ACCAUGAUUA}}\end{matrix}$$

SEQUENCE ALIGNMENT

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

$$\underset{\substack{\uparrow \\ \text{match}}}{\binom{A}{A}}\binom{U}{-}\binom{-}{C}\underset{\substack{\uparrow \\ \text{insertion}}}{\binom{U}{-}}\binom{C}{C}\binom{-}{A}\underset{\substack{\uparrow \\ \text{deletion}}}{\binom{-}{U}}\binom{G}{G}\binom{A}{A}\binom{U}{U}\underbrace{\binom{A}{U}\binom{C}{A}}_{\substack{\uparrow \\ \text{mismatch}}} \begin{array}{l} \rightarrow \text{AUUCGAUAC} \\ \rightarrow \text{ACCAUGAUUA} \\ \qquad \uparrow \\ \quad \text{projections} \end{array}$$

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 $\Sigma\times\Sigma \oplus \Sigma\times\{-\} \oplus \{-\}\times\Sigma$

$$\underset{\substack{\uparrow\\ \text{match}}}{\begin{pmatrix}A\\A\end{pmatrix}}\begin{pmatrix}U\\-\end{pmatrix}\begin{pmatrix}-\\C\end{pmatrix}\underset{\substack{\uparrow\\ \text{insertion}}}{\begin{pmatrix}U\\-\end{pmatrix}}\begin{pmatrix}C\\C\end{pmatrix}\begin{pmatrix}-\\A\end{pmatrix}\underset{\substack{\uparrow\\ \text{deletion}}}{\begin{pmatrix}-\\U\end{pmatrix}}\begin{pmatrix}G\\G\end{pmatrix}\begin{pmatrix}A\\A\end{pmatrix}\begin{pmatrix}U\\U\end{pmatrix}\underbrace{\begin{pmatrix}A\\U\end{pmatrix}\begin{pmatrix}C\\A\end{pmatrix}}_{\substack{\uparrow\\ \text{mismatch}}}$$

$\rightarrow$ AUUCGAUAC

$\rightarrow$ ACCAUGAUUA

$\uparrow$ projections

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, find one optimal alignment between S_1 and S_2.

Solvable by Dynamic Programming:

→ Needleman - Wunsch algorithm

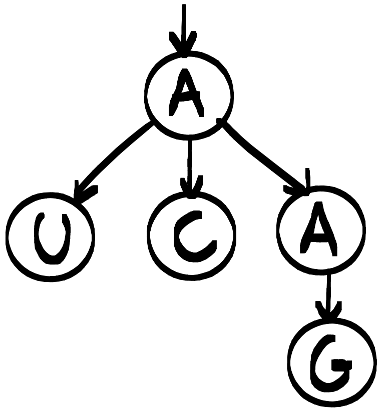
→ Smith - Waterman algorithm

→ BLAST (heuristic)

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

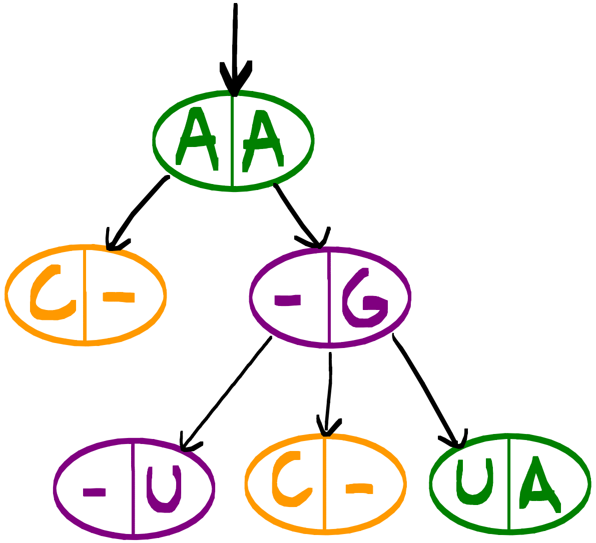
TREES AND SUPERTREES

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

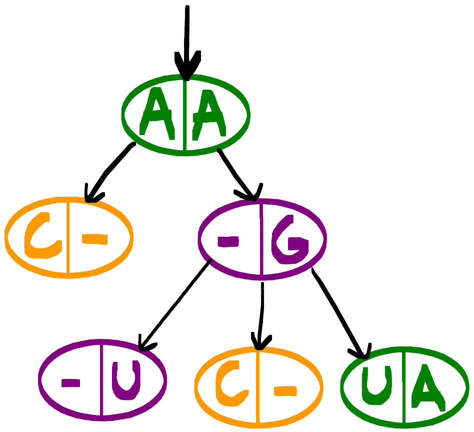


Supertree = tree with 3 types of vertices :

- X | Y (mis)match
- X | - insertion
- - | Y deletion



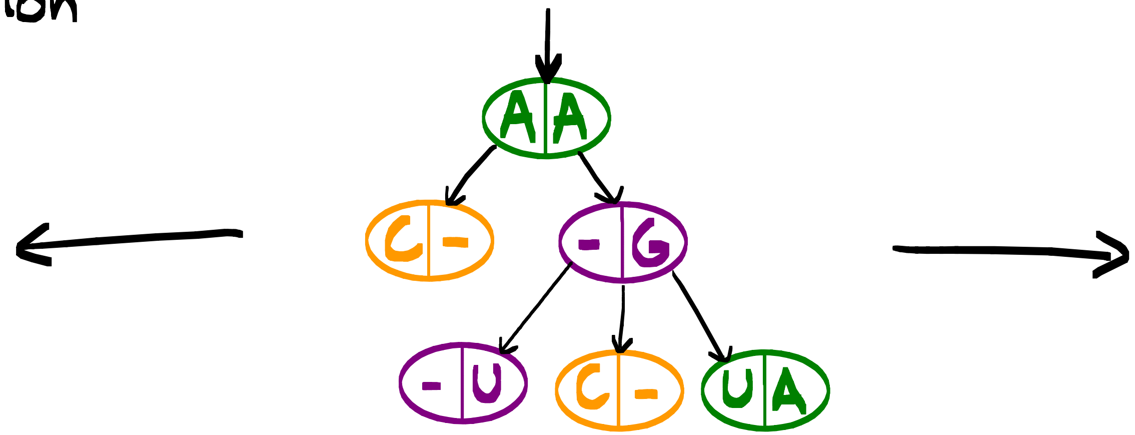
TREE ALIGNMENTS



TREE ALIGNMENTS

first projection

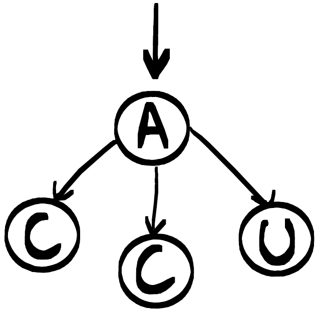
second projection

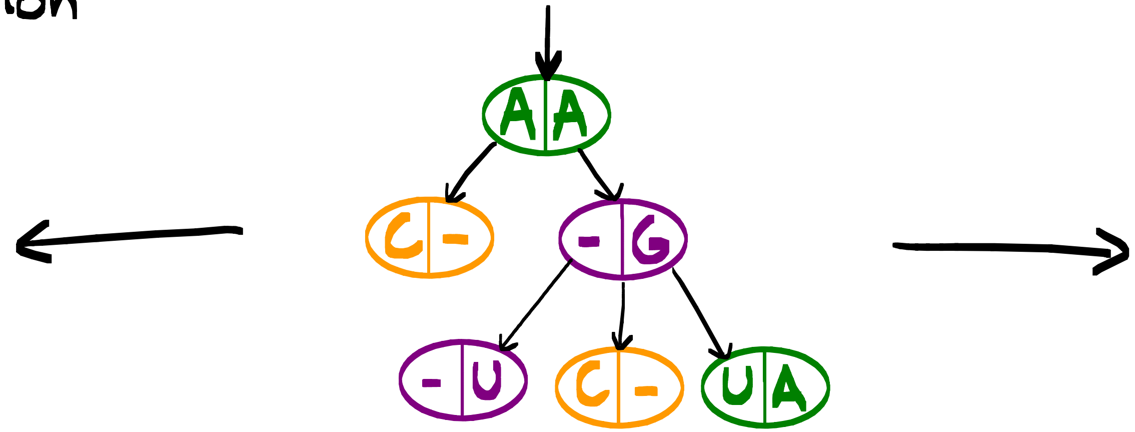


TREE ALIGNMENTS

first projection

second projection



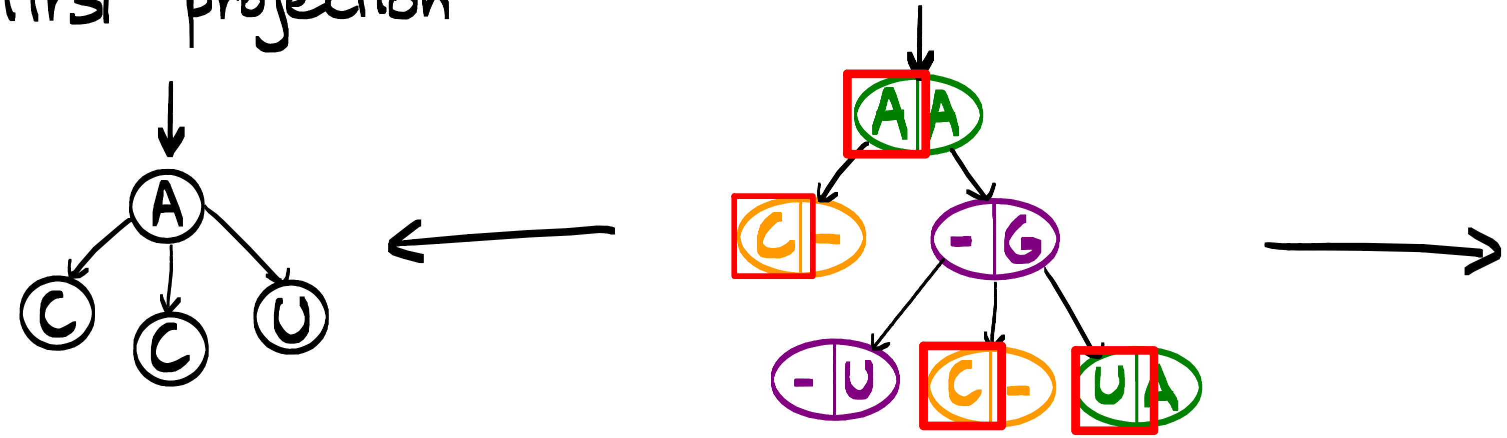


keep left letters

TREE ALIGNMENTS

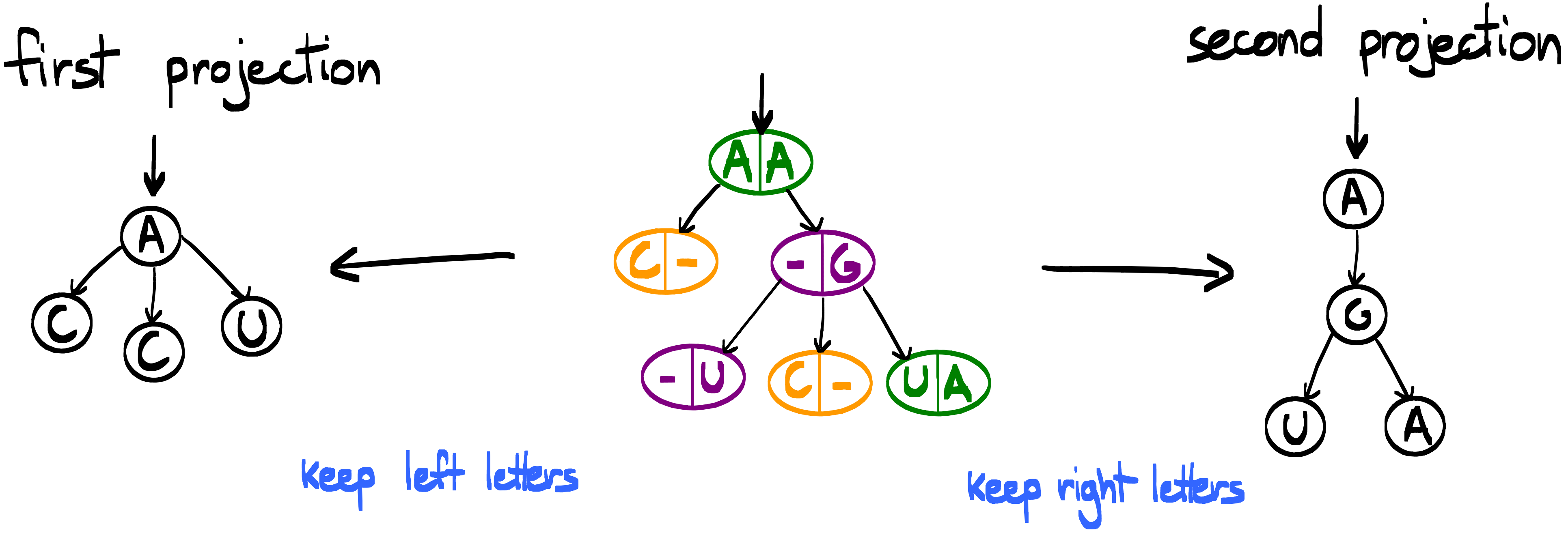
first projection

second projection

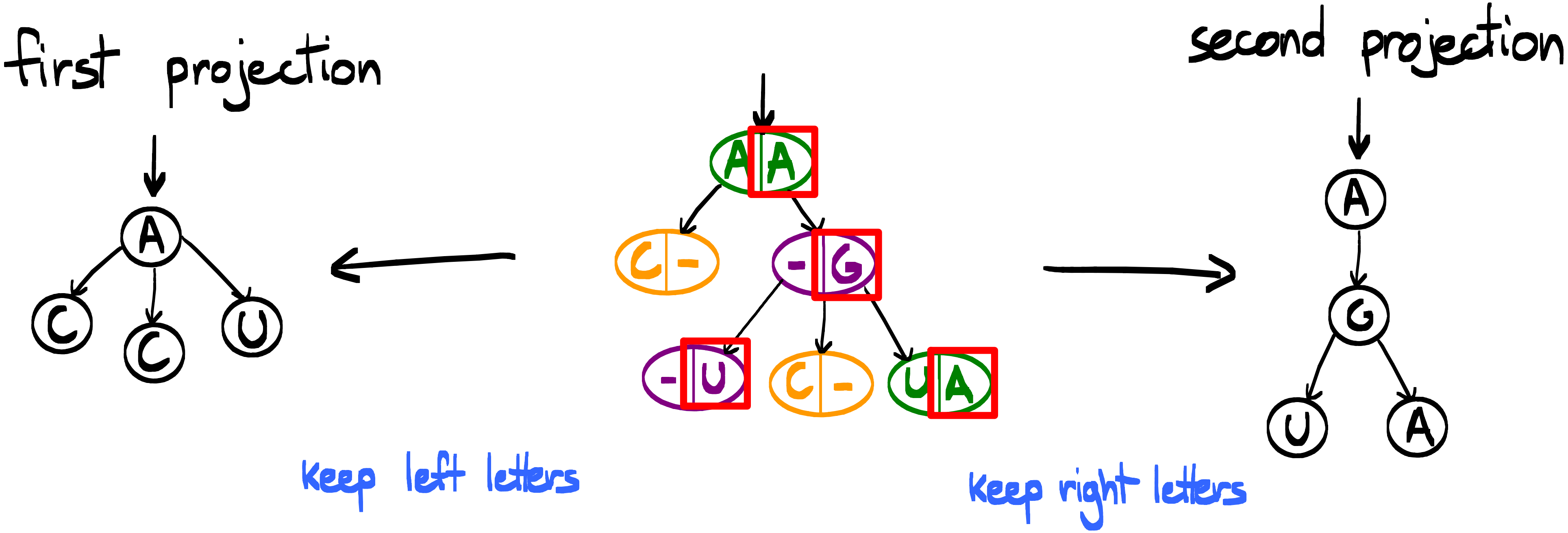


keep left letters

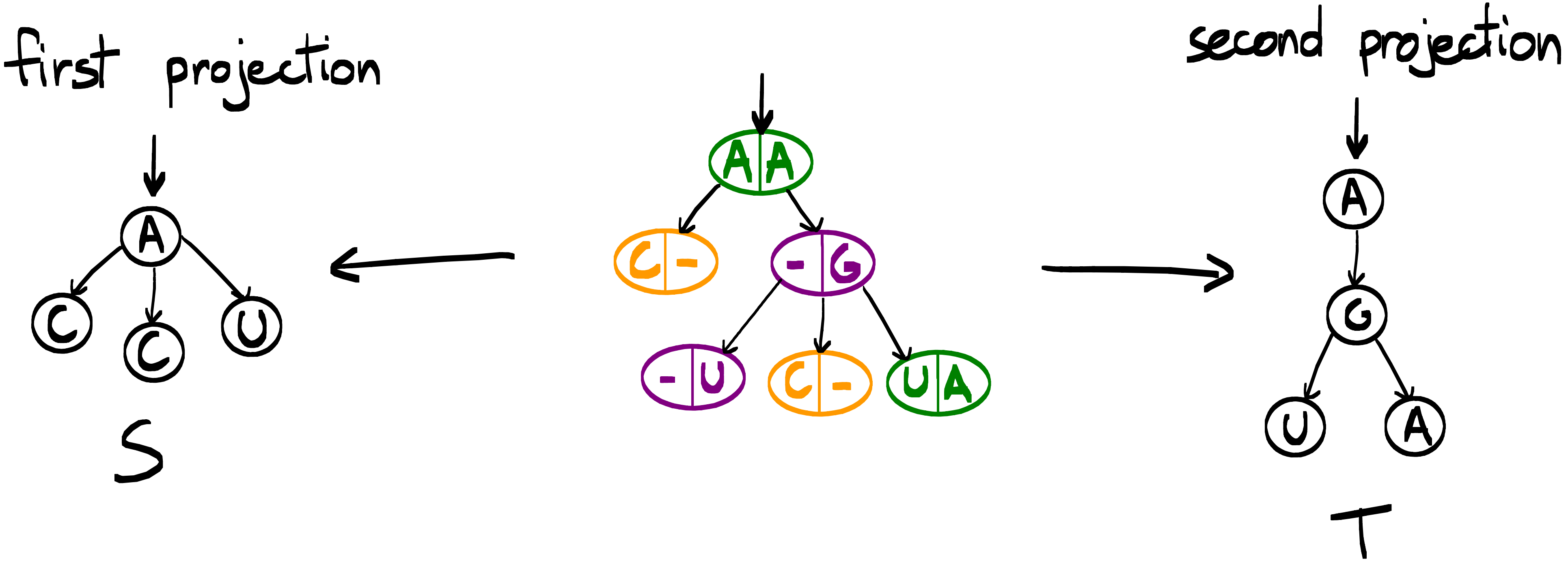
TREE ALIGNMENTS



TREE ALIGNMENTS



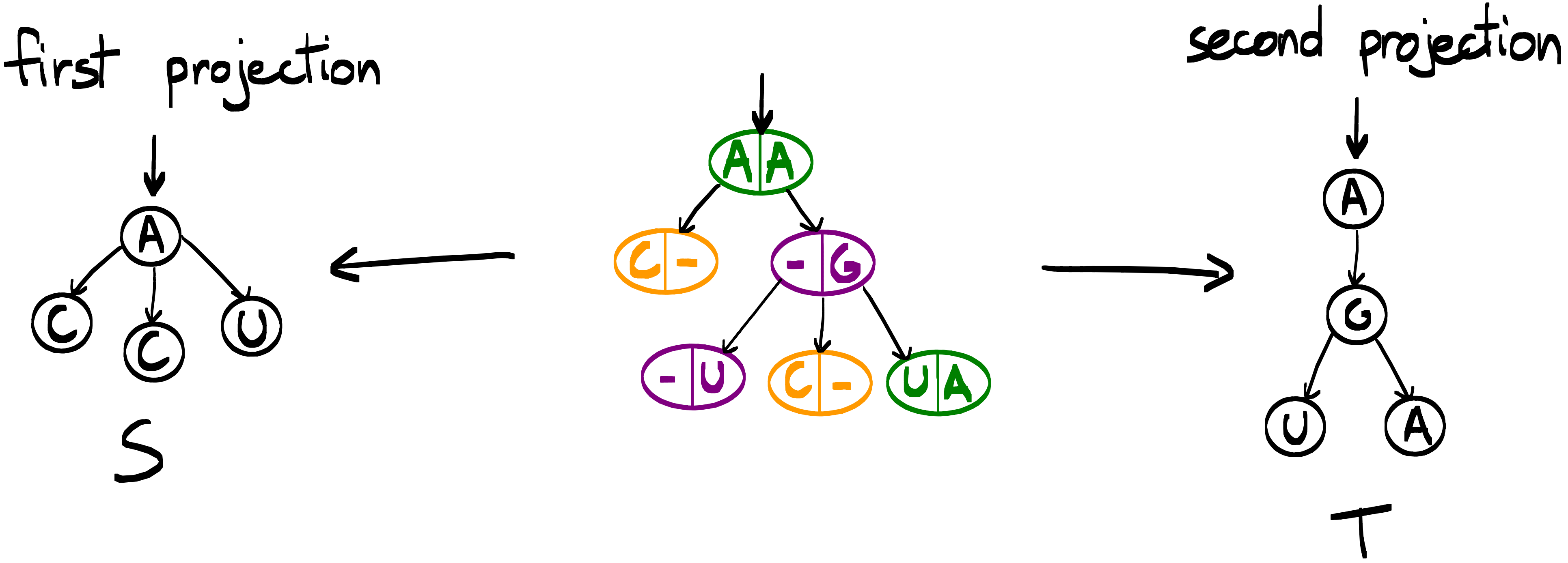
TREE ALIGNMENTS



Given two trees S and T,

<u>alignment</u> between S and T = supertree whose projections are S and T.

TREE ALIGNMENTS



Given two trees S and T,

<u>alignment</u> 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.

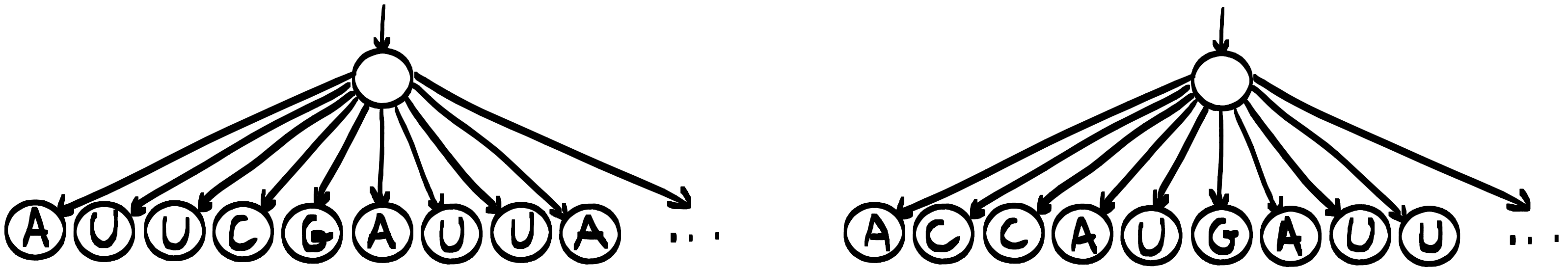
SEQUENCE

AUUCGAUUA...   ACCAUGAUUA...

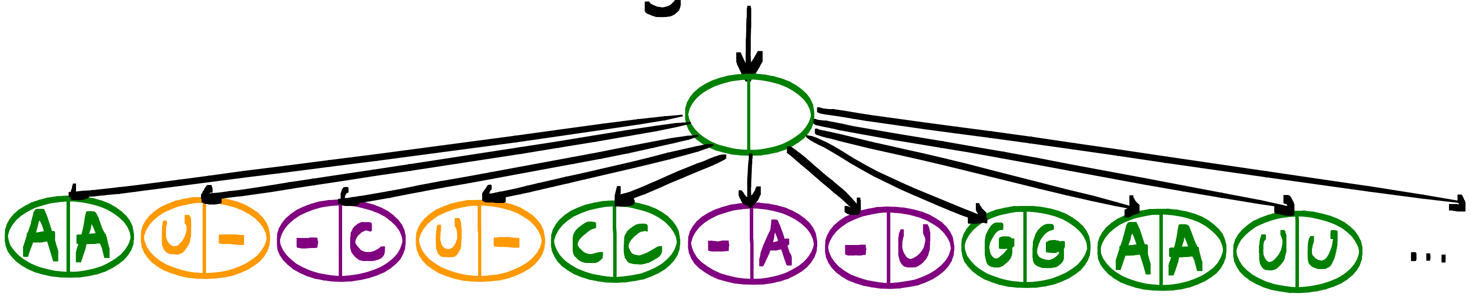
alignment :

$\binom{A}{A}\binom{U}{-}\binom{-}{C}\binom{U}{-}\binom{C}{C}\binom{-}{A}\binom{-}{U}\binom{G}{G}\binom{A}{A}\binom{U}{U}\binom{U}{U}\binom{A}{A}\ldots$

TREE



alignment :



OPTIMAL ALIGNMENT

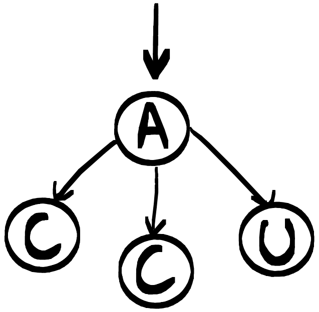
Classical problem: Given S and T, find one optimal alignment between S and T.

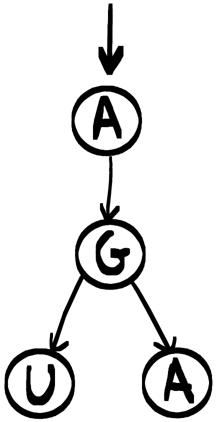
Solvable by Dynamic Programming:

Worst case time
$O(n^4)$
[Jiang, Wang, Zhang]

Average time
$O(n^2)$
[Herrbach, Denise, Dulucq]

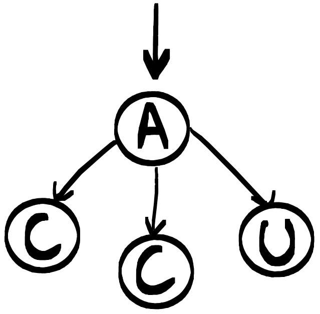
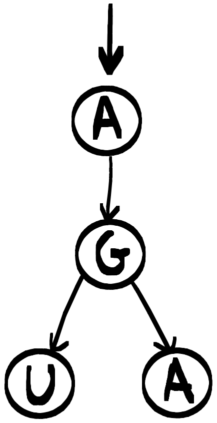
SPACE OF ALIGNMENTS

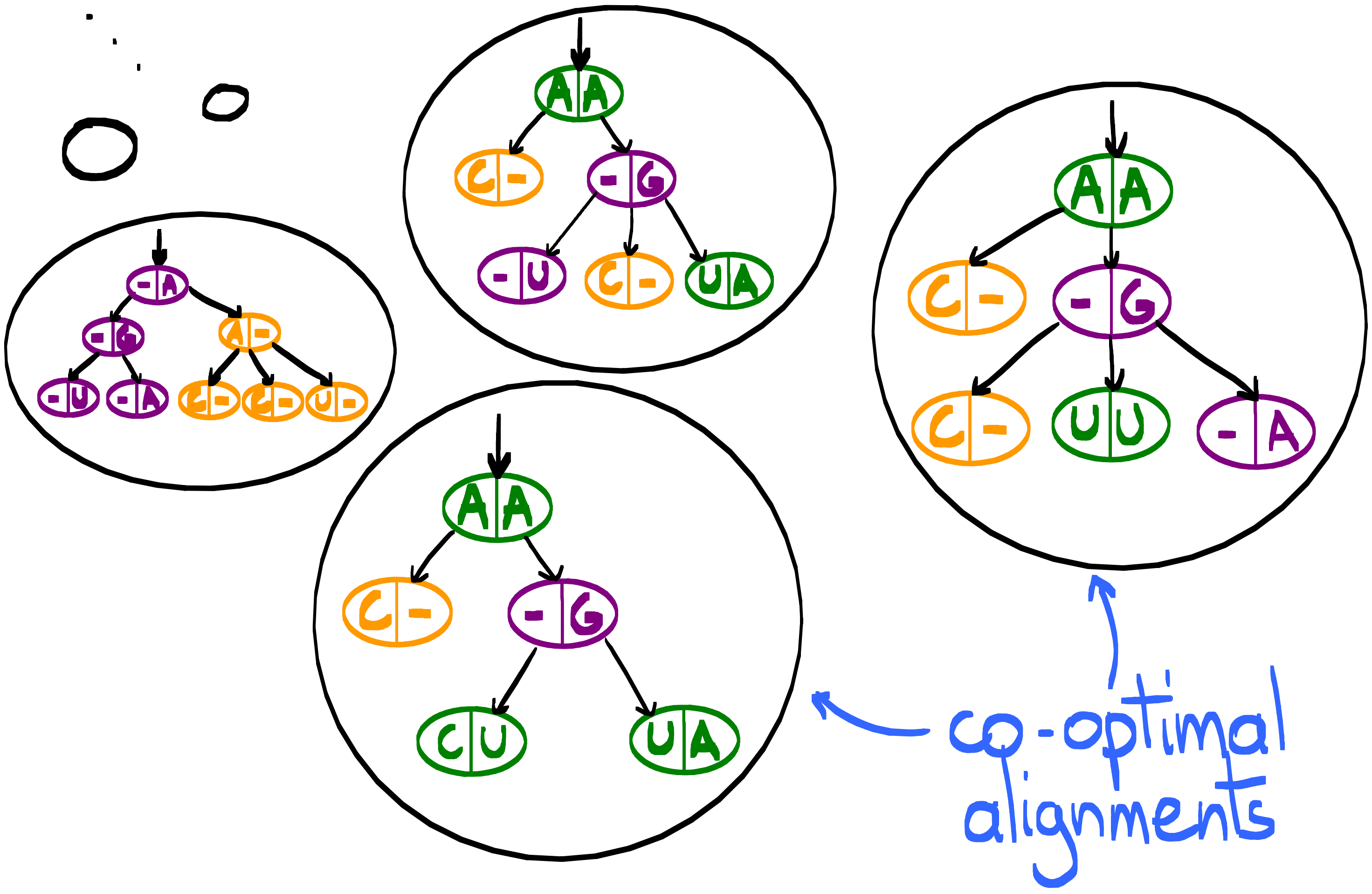
Which alignment between 

and 

is the most likely?

SPACE OF ALIGNMENTS

Which alignment between  and  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.

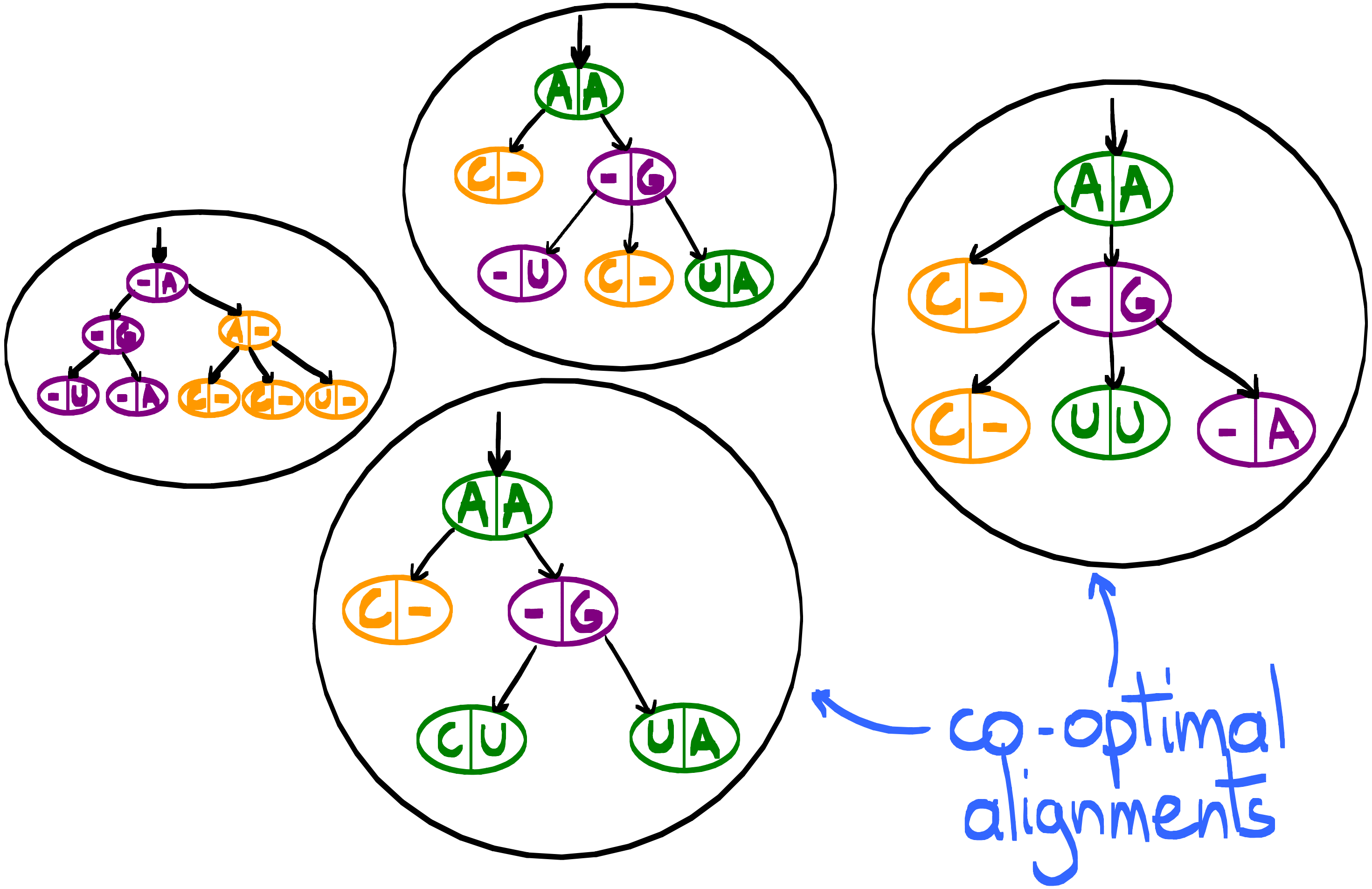
SPACE OF ALIGNMENTS

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

probability of an alignment A

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

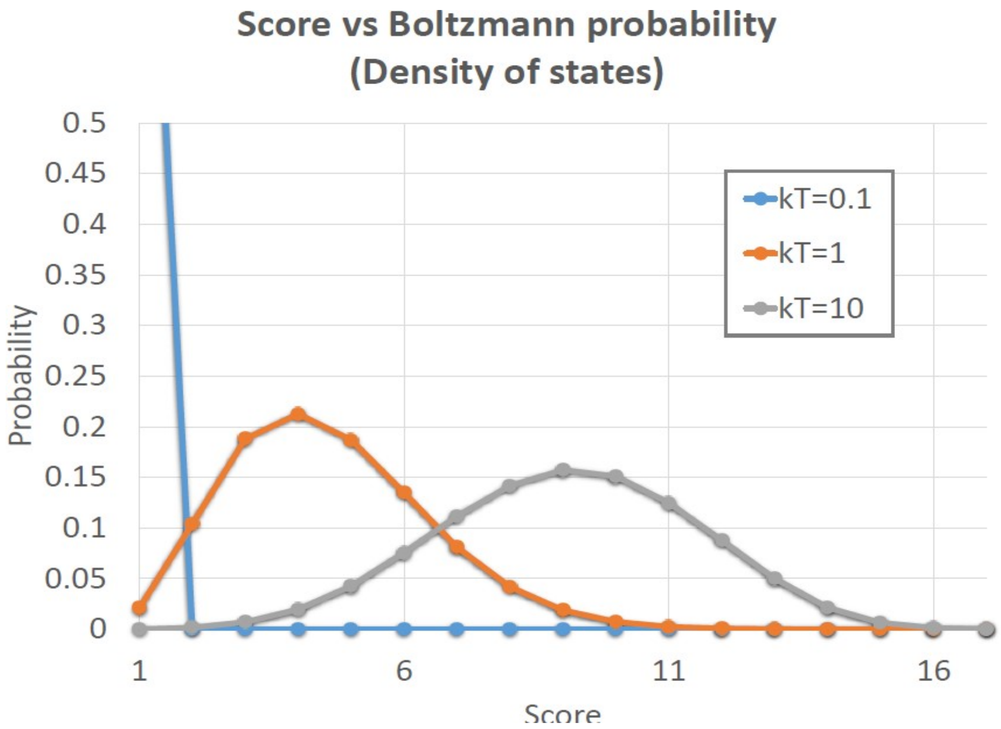
(Gibbs-Boltzmann distribution)



SPACE OF ALIGNMENTS

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

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.

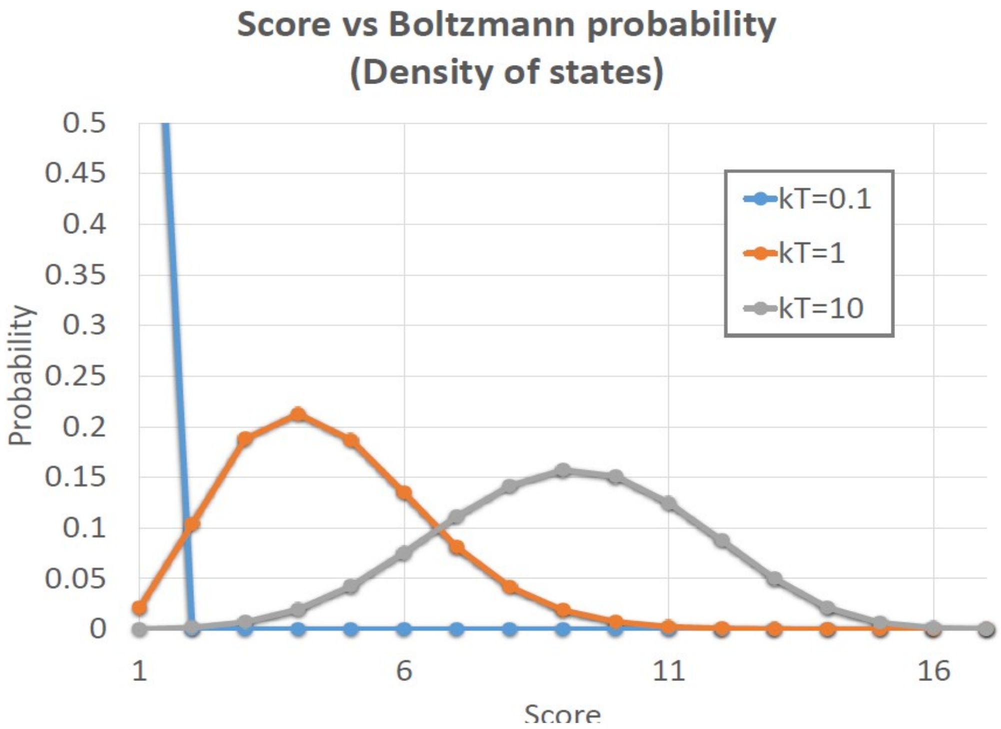
SPACE OF ALIGNMENTS

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

??. probability of an alignment A

$$\propto e^{-\frac{cost(A)}{K}}$$

(Gibbs-Boltzmann distribution)



K=0 : Uniform distribution over optimal alignments.

K=+∞ : Uniform distribution over all alignments.

AMBIGUITY OF ALIGNMENTS

For sequences,

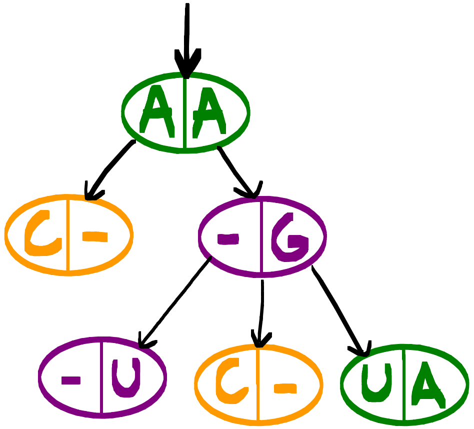
$$\begin{pmatrix}A\\A\end{pmatrix}\begin{pmatrix}U\\-\end{pmatrix}\begin{pmatrix}-\\C\end{pmatrix}\begin{pmatrix}U\\-\end{pmatrix}\begin{pmatrix}C\\C\end{pmatrix}\begin{pmatrix}-\\A\end{pmatrix}\begin{pmatrix}-\\U\end{pmatrix}\begin{pmatrix}G\\G\end{pmatrix}\begin{pmatrix}A\\A\end{pmatrix}\begin{pmatrix}U\\U\end{pmatrix}\begin{pmatrix}U\\U\end{pmatrix}\begin{pma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

is the same alignment as

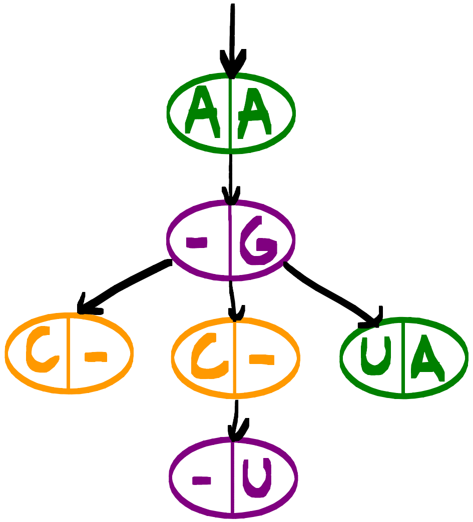
$$\begin{pmatrix}A\\A\end{pmatrix}\begin{pmatrix}-\\C\end{pmatrix}\begin{pmatrix}U\\-\end{pmatrix}\begin{pmatrix}U\\-\end{pmatrix}\begin{pmatrix}C\\C\end{pmatrix}\begin{pmatrix}-\\A\end{pmatrix}\begin{pmatrix}-\\U\end{pmatrix}\begin{pmatrix}G\\G\end{pmatrix}\begin{pmatrix}A\\A\end{pmatrix}\begin{pmatrix}U\\U\end{pmatrix}\begin{pmatrix}U\\U\end{pmatrix}\begin{pma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

AMBIGUITY OF ALIGNMENTS

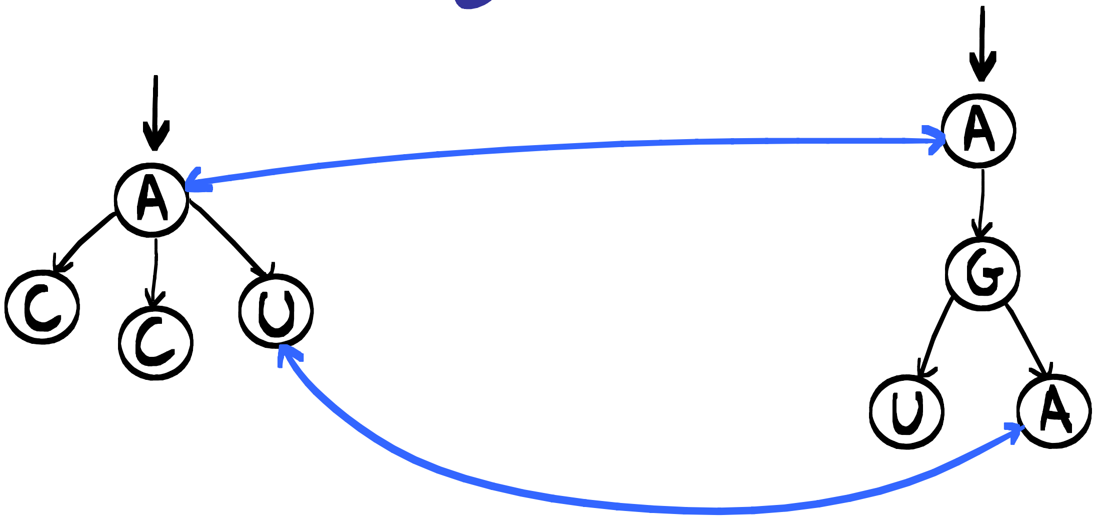
For trees,



and

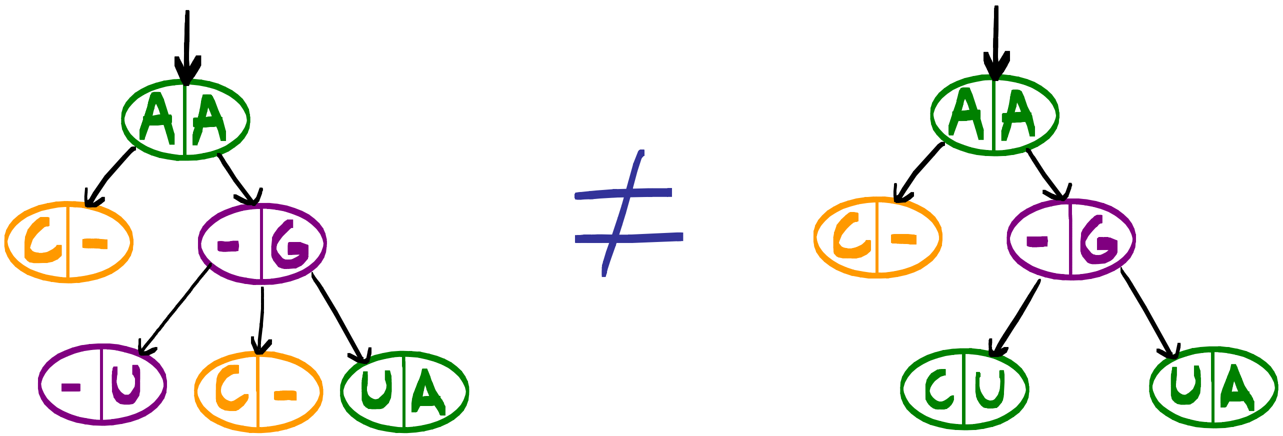


induce the same alignment between

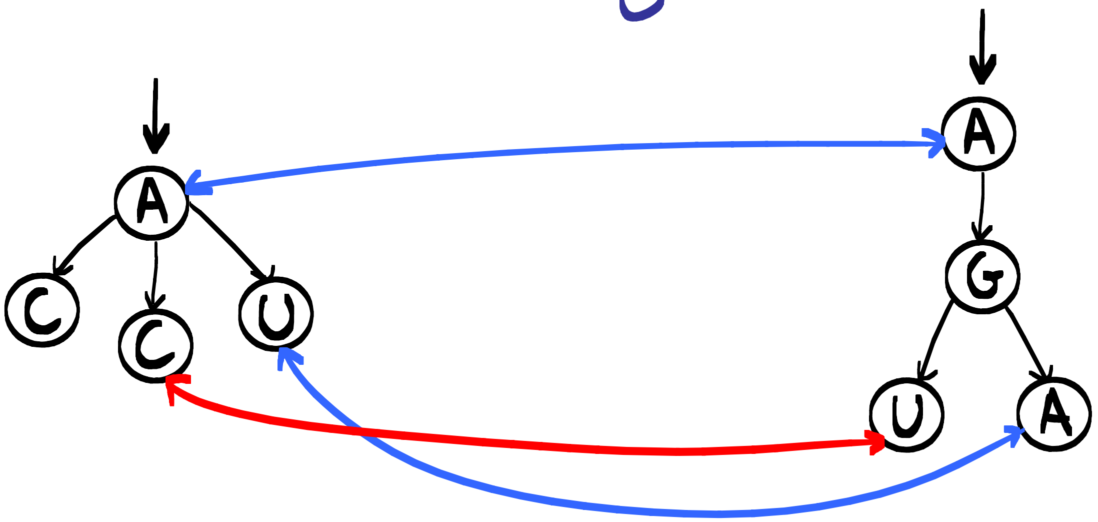


AMBIGUITY OF ALIGNMENTS

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 :

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

PROBLEM RAISED BY THE AMBIGUITY

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

Ex :

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

Insertions before Deletions.

PROBLEM RAISED BY THE AMBIGUITY

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

Ex :

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

Insertions before Deletions.

For trees, it is much more complicated!

PROBLEM RAISED BY THE AMBIGUITY

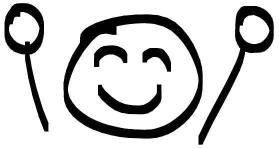
For sequences, we can deal with the ambiguity by defining canonical alignments.

Ex :

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

Insertions before Deletions.

For trees, it is much more complicated!

Strategy: COMBINATORICS ! 

PROBLEM RAISED BY THE AMBIGUITY

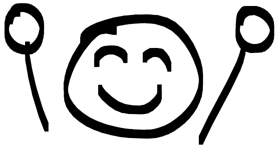
For sequences, we can deal with the ambiguity by defining canonical alignments.

Ex :

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

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:

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

Non-ambiguous grammar:

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

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

A GRAMMAR FOR ALIGNMENTS

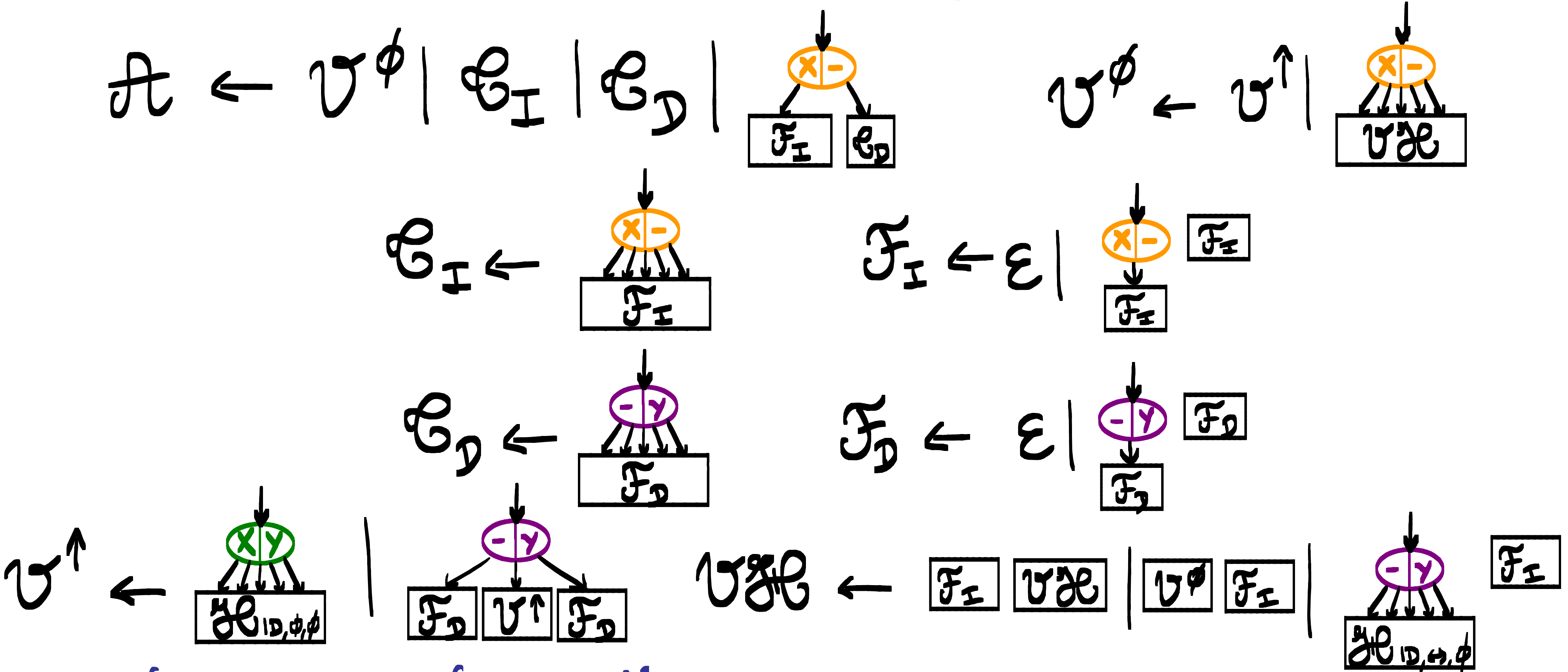
For trees, an ambiguous grammar can be derived from [Jiang, Wang, Zhang].

Our result:

Theorem: The set of generated by the following grammar contains every tree alignment exactly once.

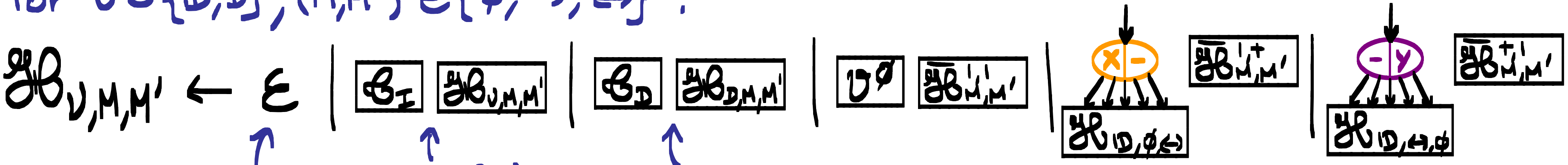
A GRAMMAR FOR ALIGNMENTS

Our (complicated) non-ambiguous grammar:



$\mathcal{A} \leftarrow \mathcal{V}^{\emptyset} \mid \mathcal{C}_I \mid \mathcal{C}_D \mid$
x|-
$\mathcal{F}_I$
$\mathcal{C}_D$
$\mathcal{V}^{\emptyset} \leftarrow \mathcal{V}^{\uparrow} \mid$
x|-
$\mathcal{VH}$
$\mathcal{C}_I \leftarrow$
x|-
$\mathcal{F}_I$
$\mathcal{F}_I \leftarrow \varepsilon \mid$
x|-
$\mathcal{F}_I$
$\mathcal{F}_I$
$\mathcal{C}_D \leftarrow$
-|y
$\mathcal{F}_D$
$\mathcal{F}_D \leftarrow \varepsilon \mid$
-|y
$\mathcal{F}_D$
$\mathcal{F}_D$
$\mathcal{V}^{\uparrow} \leftarrow$
x|y
$\mathcal{H}_{\mathbb{D},\emptyset,\emptyset}$
-|y
$\mathcal{F}_D$
$\mathcal{V}^{\uparrow}$
$\mathcal{F}_D$
$\mathcal{VH} \leftarrow$
$\mathcal{F}_I$
$\mathcal{VH}$
$\mathcal{V}^{\emptyset}$
$\mathcal{F}_I$
-|y
$\mathcal{F}_I$
$\mathcal{H}_{\mathbb{D},\leftrightarrow,\emptyset}$

For $\mathcal{D} \in \{\mathbb{D}, D\}$, $(M,M') \in \{\emptyset, \rightarrow, \leftrightarrow\}^2$:



$\mathcal{GB}_{\mathcal{D},M,M'} \leftarrow \varepsilon \mid$
$\mathcal{C}_I$
$\mathcal{GB}_{\mathcal{D},M,M'}$
$\mathcal{C}_D$
$\mathcal{GB}_{D,M,M'}$
$\mathcal{V}^{\emptyset}$
$\overline{\mathcal{GB}}^{\,\mathrm{l},\mathrm{l}}_{M,M'}$
x|-
$\overline{\mathcal{GB}}^{\,\mathrm{l},+}_{M,M'}$
$\mathcal{H}_{\mathbb{D},\emptyset,\leftrightarrow}$
-|y
$\overline{\mathcal{GB}}^{\,+,\mathrm{l}}_{M,M'}$
$\mathcal{H}_{\mathbb{D},\leftrightarrow,\emptyset}$

only if $(M,M')=(\emptyset,\emptyset)$

only if $\mathcal{D} \neq \mathbb{D}$ and $M \neq \leftrightarrow$

only if $M' \neq \leftrightarrow$

no room for $\overline{\mathcal{GB}}^{\,i,j}_{M,M'}$...

APPLICATION 1. COUNTING.

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

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

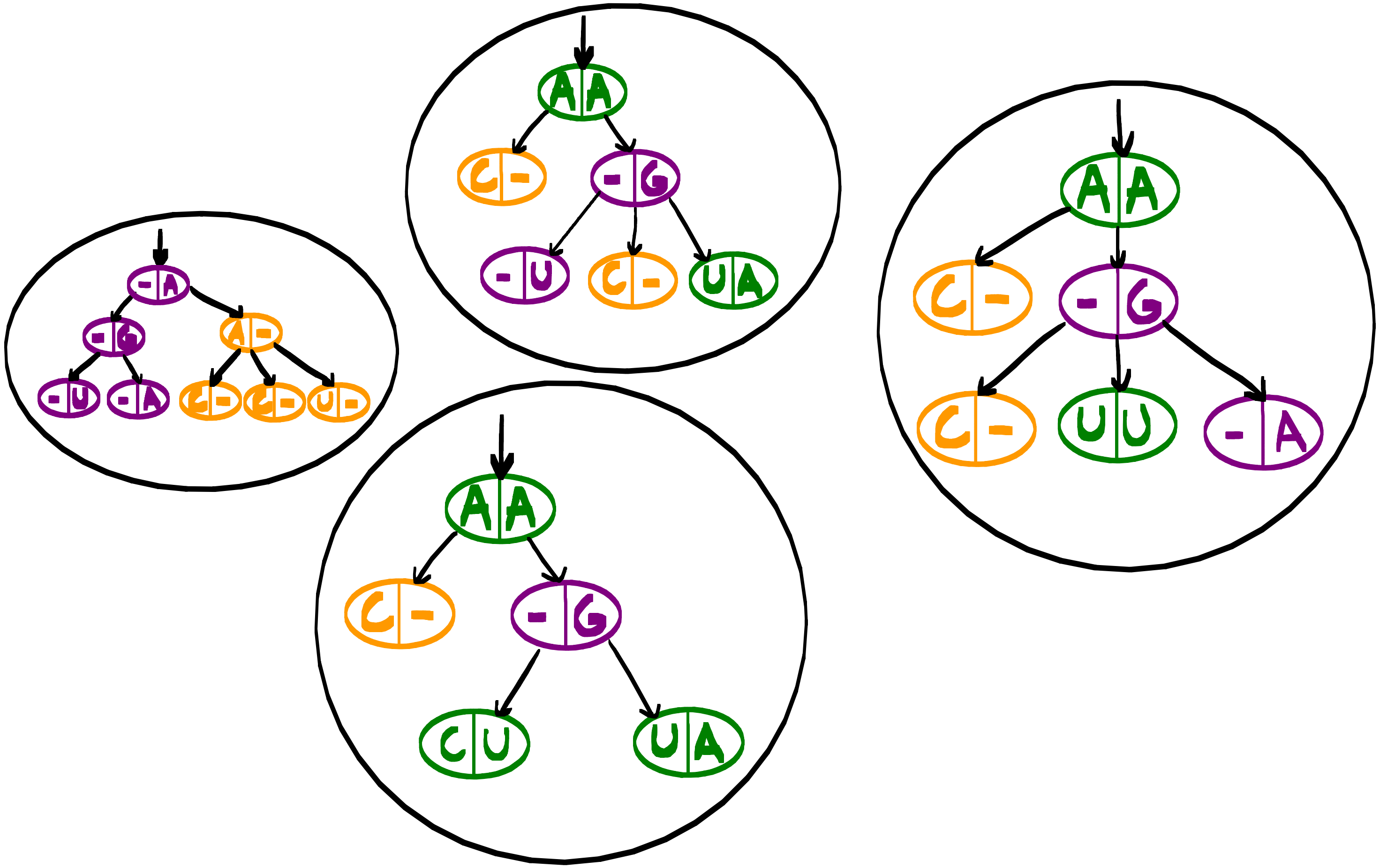
APPLICATION 2. SAMPLING

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.

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 $O(n_1 n_2)$.

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

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

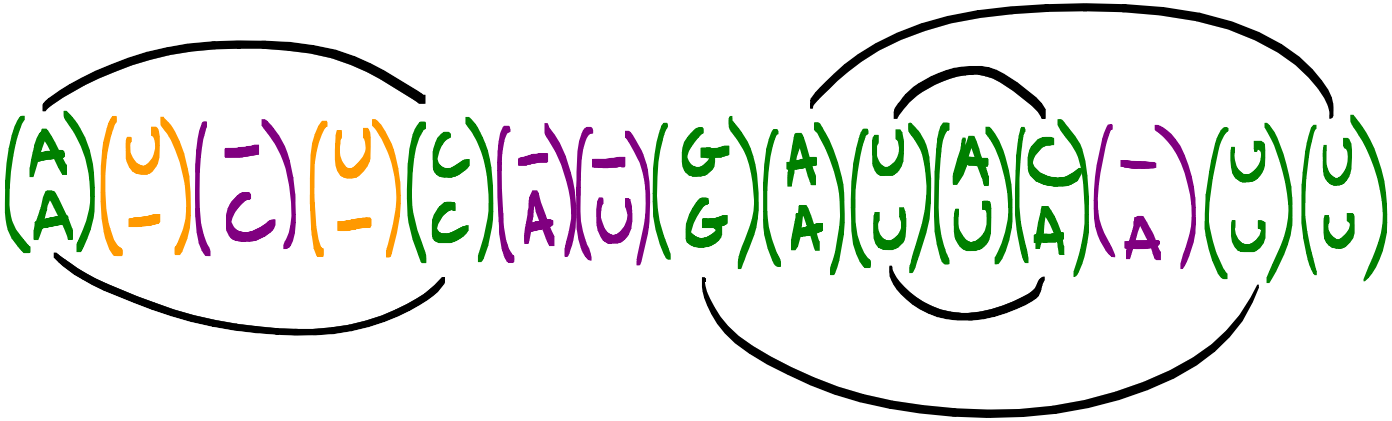
CONCLUSION

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

Open questions:

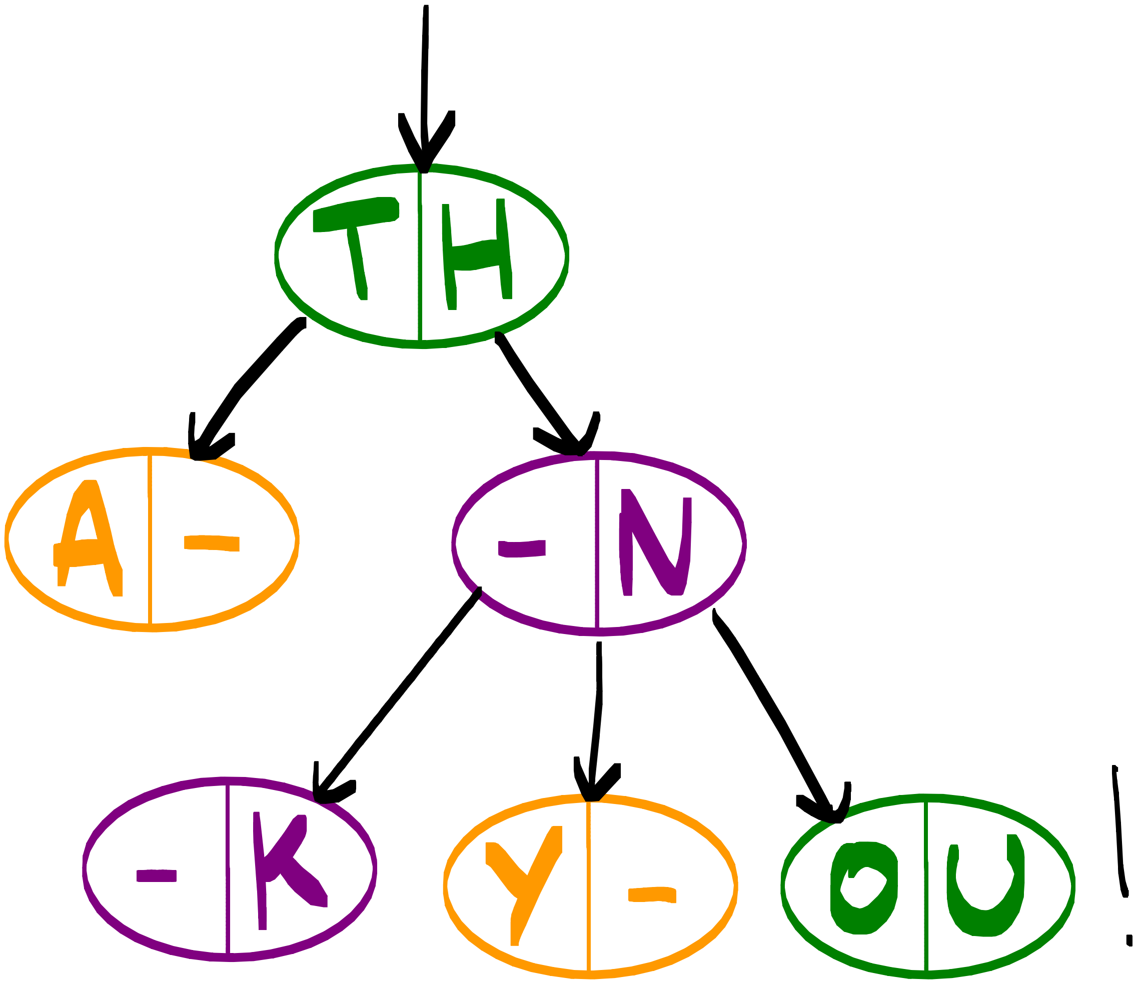
→ Existence of easier decompositions?

→ Alignment problem for arc-annoted sequences?



A U – U C – – G A U A C – U U

A – C – C A U G A U U A A U U


T | H
A | -
- | N
- | K
Y | -
O | U
!

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

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

and

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