How to compare arc-annotated sequences: the alignment hierarchy

G. Blin[1] H. Touzet[2]

[1]IGM-LabInfo UMR 8049, Université de Marne La Vallée
`gblin@univ-mlv.fr`

[2]LIFL, UMR CNRS 8022 Université des Sciences et Technologies de Lille
`touzet@lifl.fr`

Outline

Outline

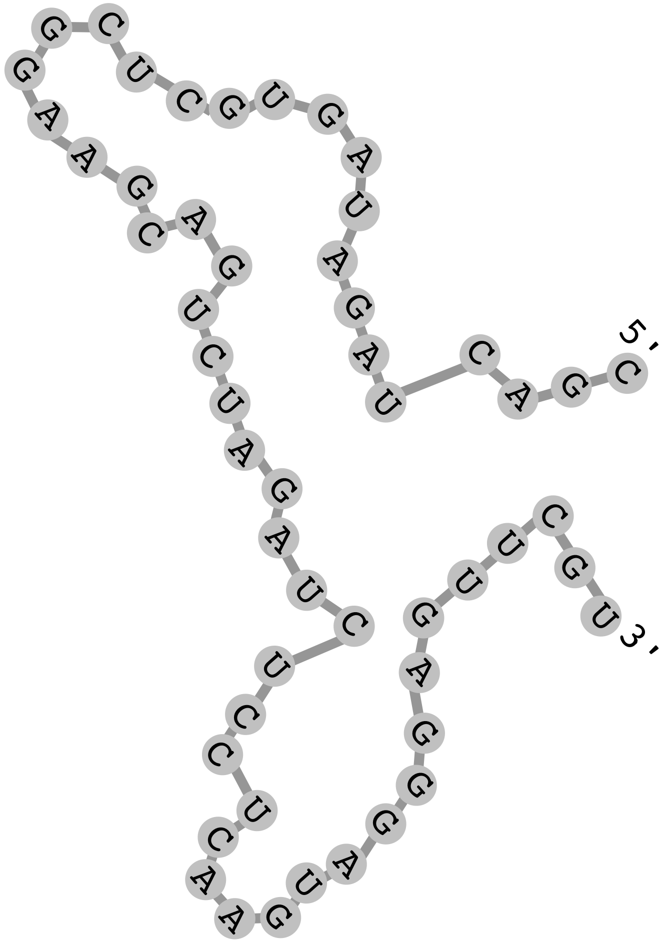
What is RNA ?

RiboNucleic Acid

What is RNA ?

RiboNucleic Acid

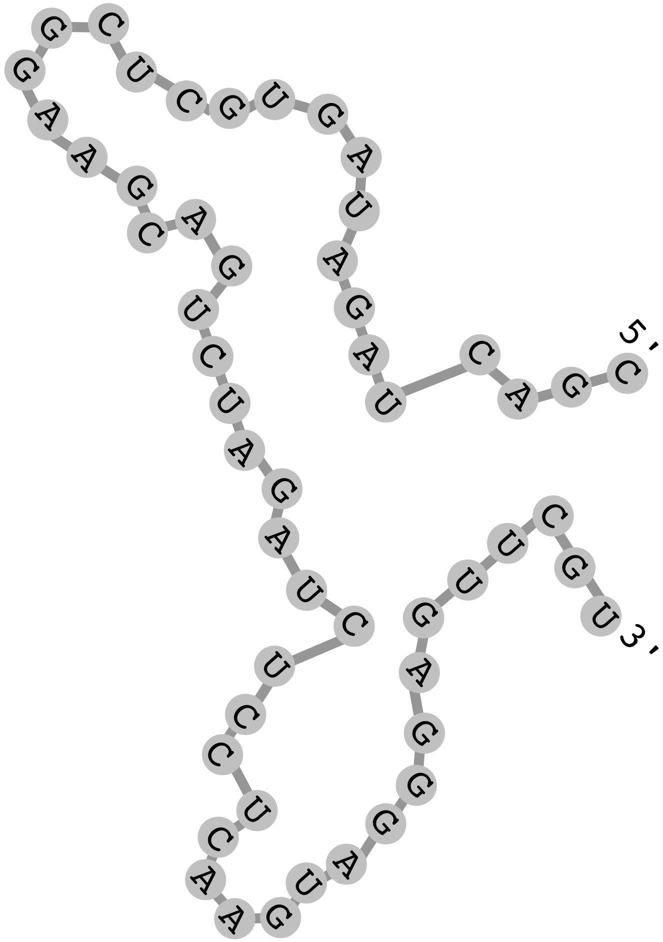
- One-Strand molecule



What is RNA ?

RiboNucleic Acid

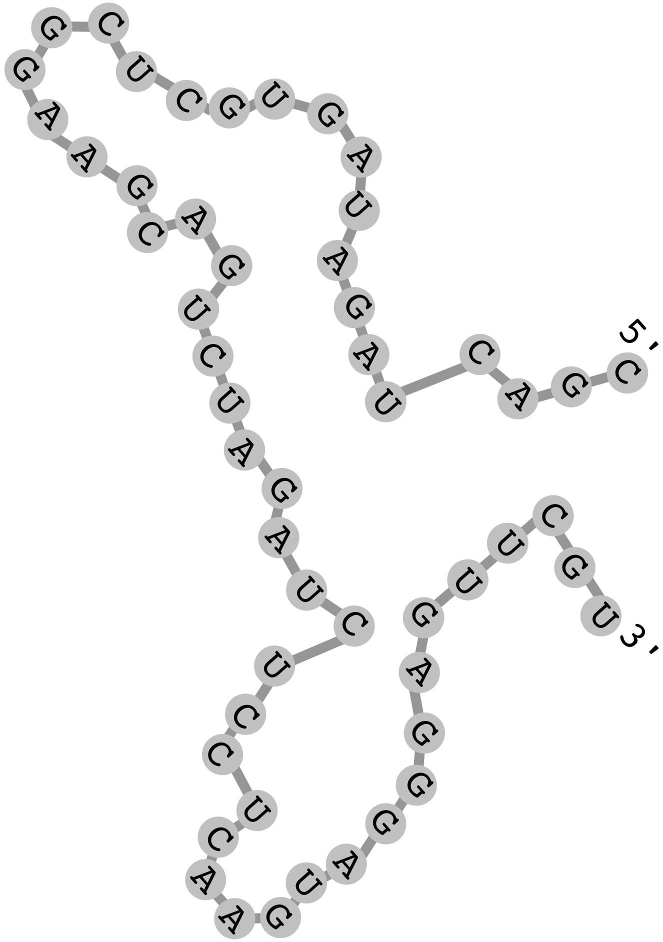
- One-Strand molecule
- Bases Adenine, Guanine, Cytosine and Uracil (instead of Thymine in DNA)



What is RNA ?

RiboNucleic Acid

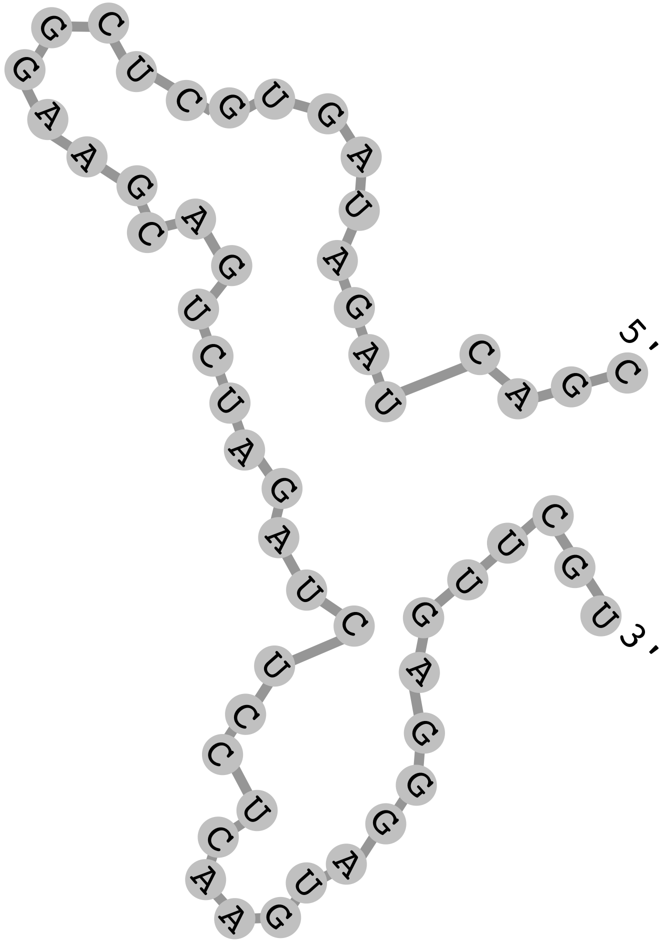
- One-Strand molecule
- Bases Adenine, Guanine, Cytosine and Uracil (instead of Thymine in DNA)
- Present in any living organism



What is RNA ?

RiboNucleic Acid

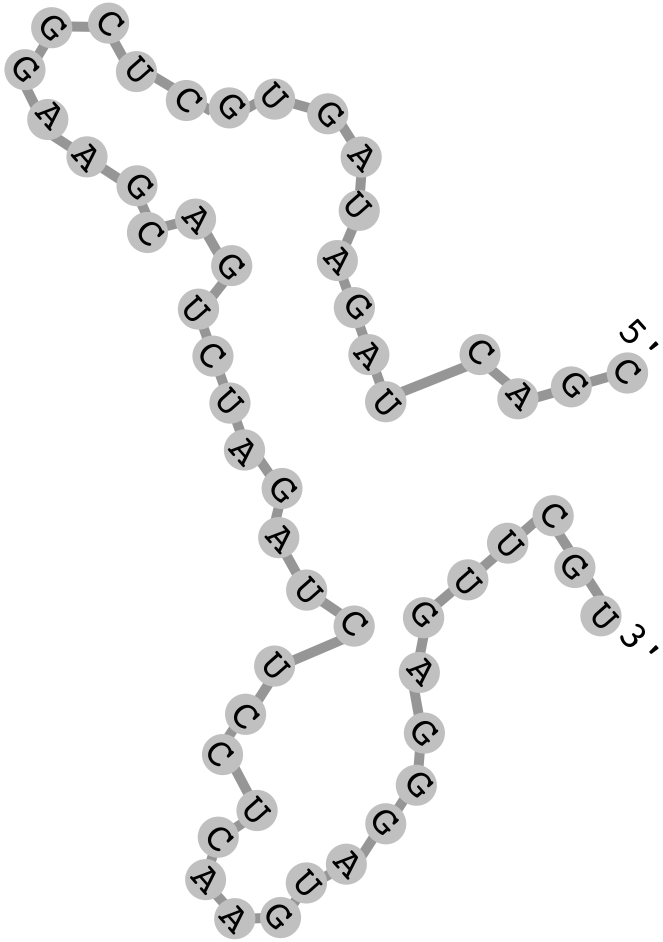
- One-Strand molecule
- Bases Adenine, Guanine, Cytosine and Uracil (instead of Thymine in DNA)
- Present in any living organism
- Essential for protein synthesis



What is RNA ?

RiboNucleic Acid

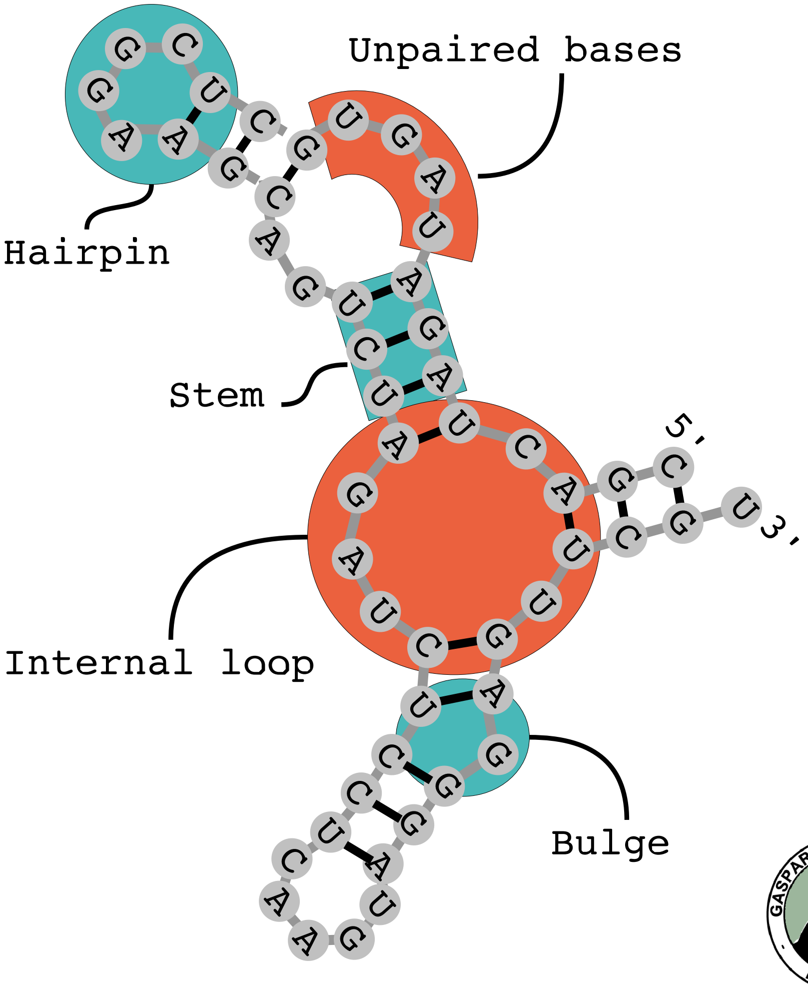
- One-Strand molecule
- Bases Adenine, Guanine, Cytosine and Uracil (instead of Thymine in DNA)
- Present in any living organism
- Essential for protein synthesis
- RNA study is essential !



How to study RNA ?

RNA study

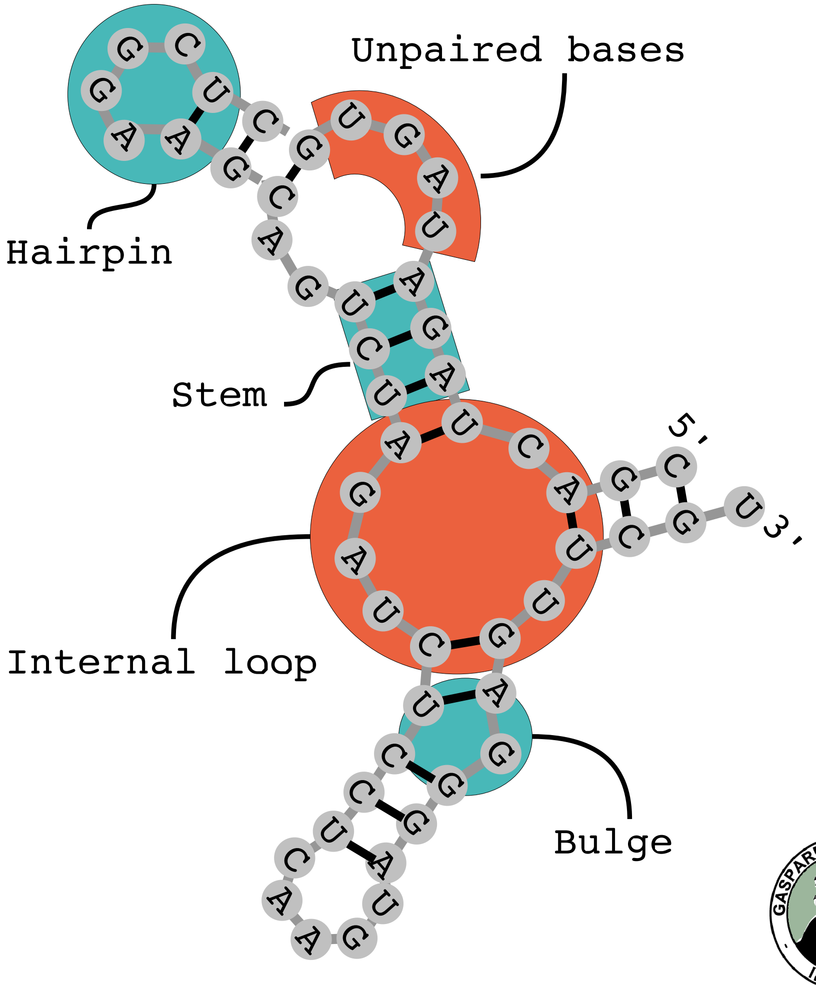
- RNA is folding in 3 dimensions
    - WC pairs
    - Wobble pairs



How to study RNA ?

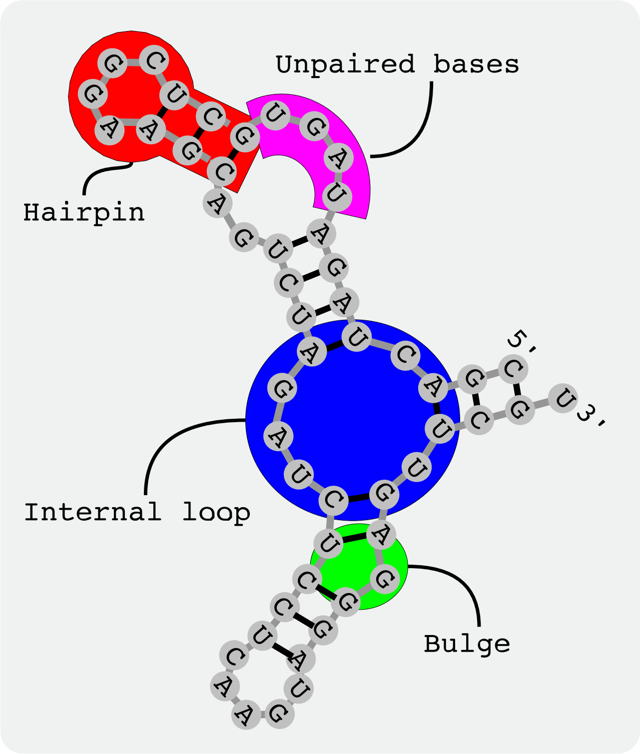
RNA study

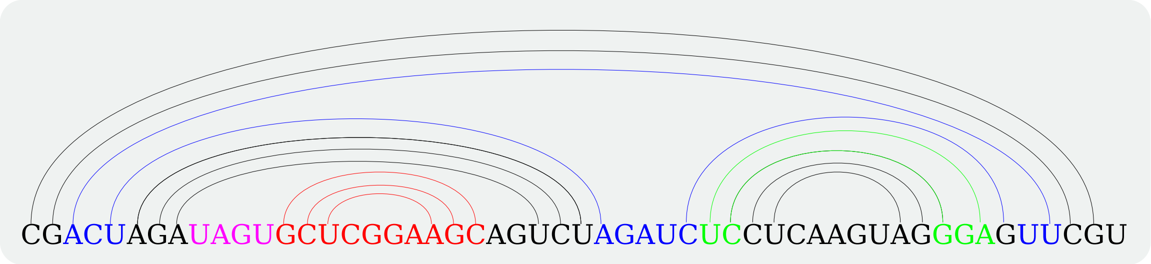
- RNA is folding in 3 dimensions
    - WC pairs
    - Wobble pairs
- Folding study (secondary or tertiary structure)
    - determines partially RNA function



Arc-Annotated Representation

RNA structure represented as an arc-annotated sequence : (S, P)



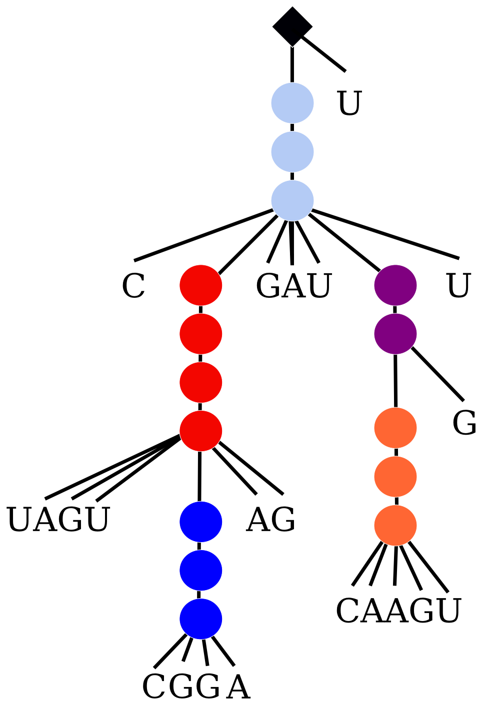


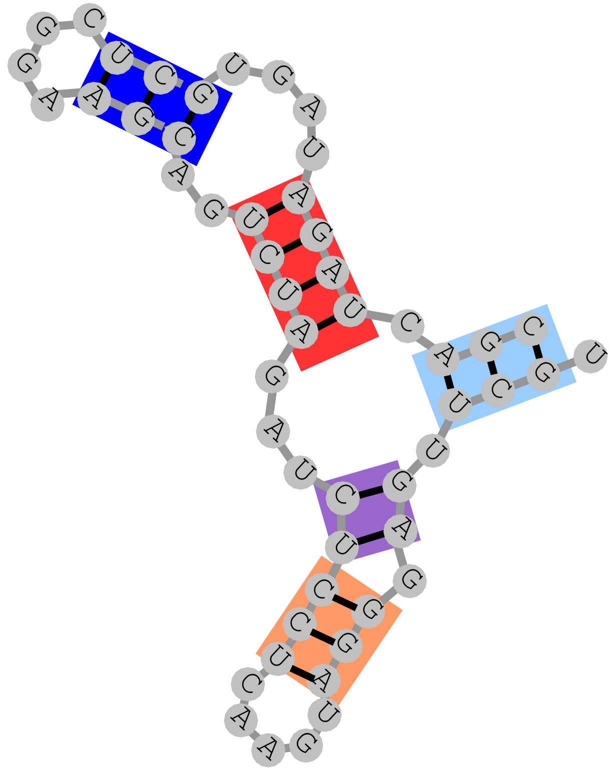
Outline

Arc-annotated sequences

Four main paradigms

- using tree representation (a base-pair≡internal node ; free-base≡leaf) :
- using a.a. sequences :

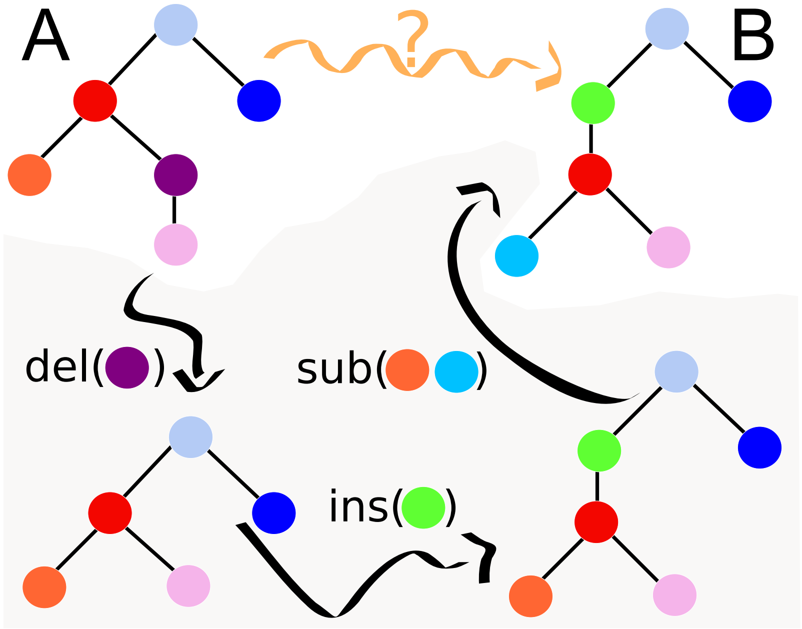




Arc-annotated sequences

Four main paradigms

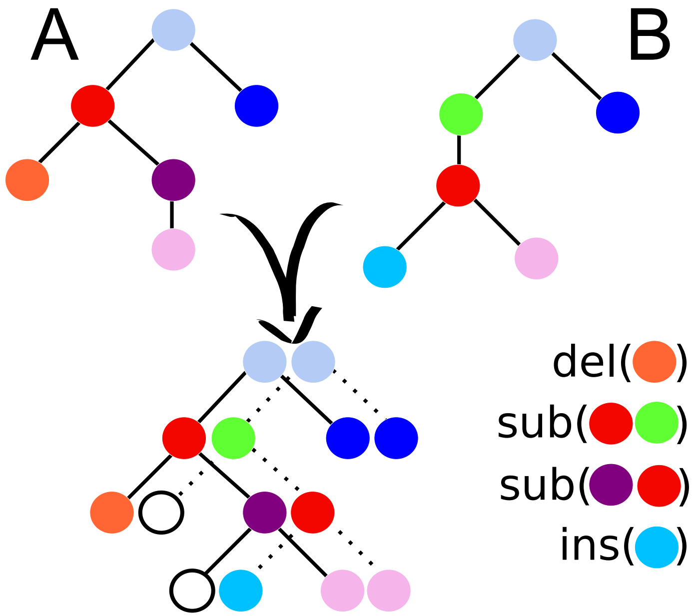
- using tree representation (a base-pair≡internal node ; free-base≡leaf) :
  - tree edit distance
- using a.a. sequences :



Arc-annotated sequences

Four main paradigms

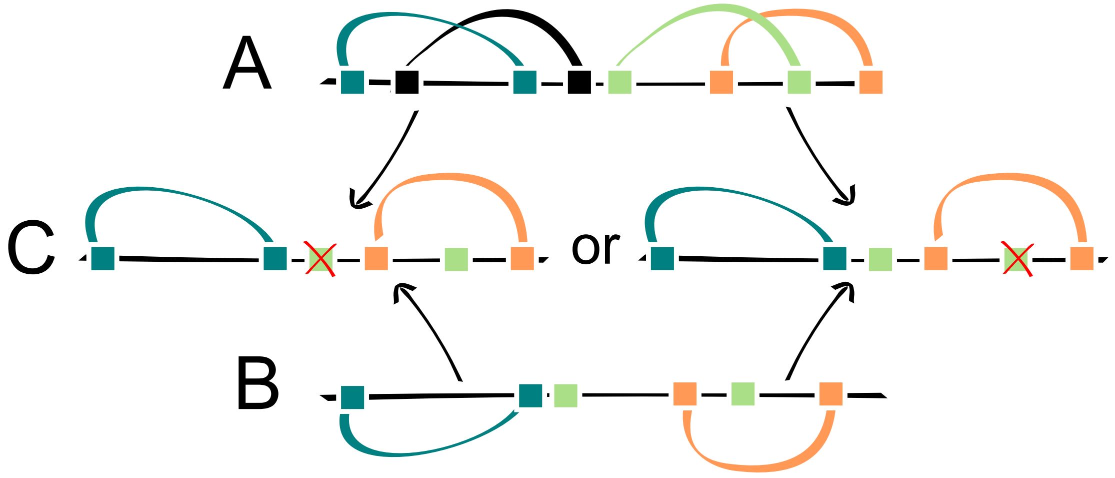
- using tree representation (a base-pair≡internal node ; free-base≡leaf) :
  - tree edit distance
  - tree alignment
- using a.a. sequences :



Arc-annotated sequences

Four main paradigms

- using tree representation (a base-pair≡internal node ; free-base≡leaf) :
  - tree edit distance
  - tree alignment
- using a.a. sequences :
  - longest arc-preserving common subsequence

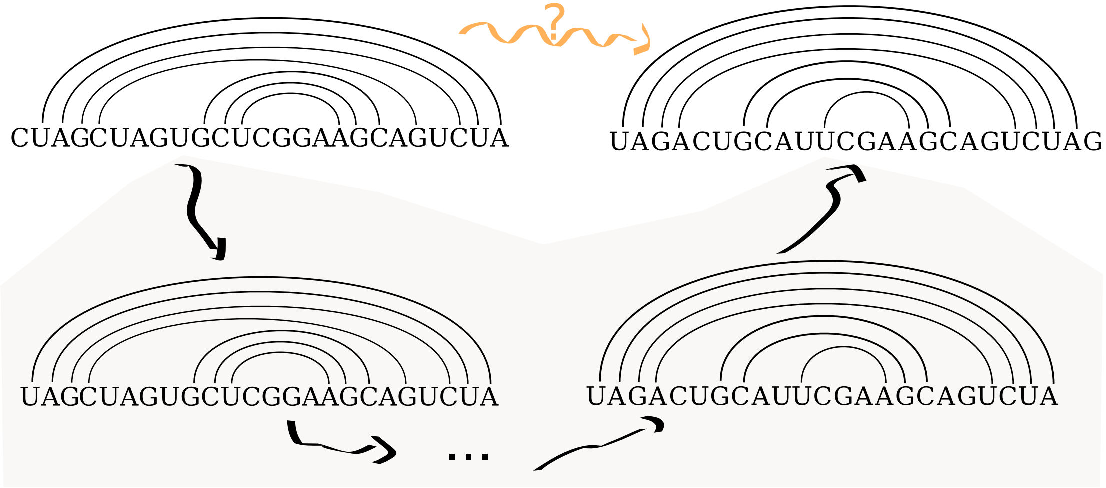


The 2 bases incident to an arc of A or B can be added to C only if the arc is preserved

Arc-annotated sequences

Four main paradigms

- using tree representation (a base-pair≡internal node ; free-base≡leaf) :
  - tree edit distance
  - tree alignment
- using a.a. sequences :
  - longest arc-preserving common subsequence
  - general edit distance



Outline

Purposes and results

A uniform framework to express comparison of a.a. sequences

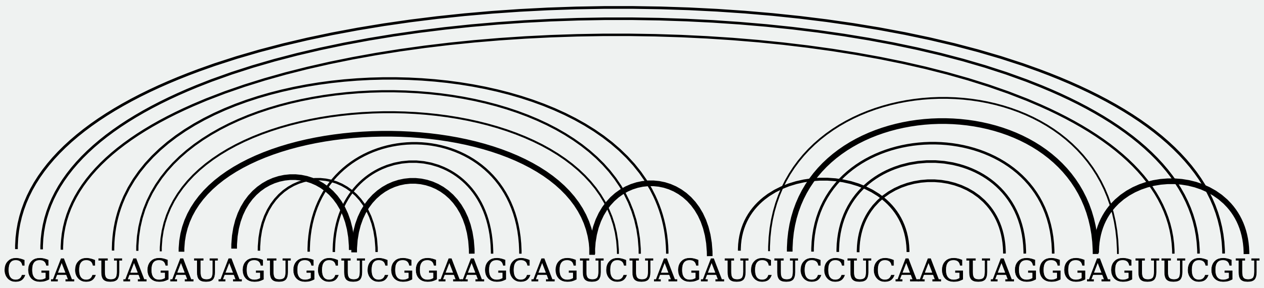
- Purposes :
    - Confront the different existing models
    - Find bridges between those last
- Results :
    - Theoretical : it brings together all previously mentioned comparison models
    - Practical : we design two poly-time algorithms for instances for which none can be found in other models.
      It allows us to compare biologically relevant instances considering a set of operations S whereas comparing such instances considering S with others comparison models is an **NP**-complete problem.

Preliminaries

Arc-structure levels

There are 4 levels (initially introduced by P.A. Evans) :

- UNLIMITED – no restriction at all,

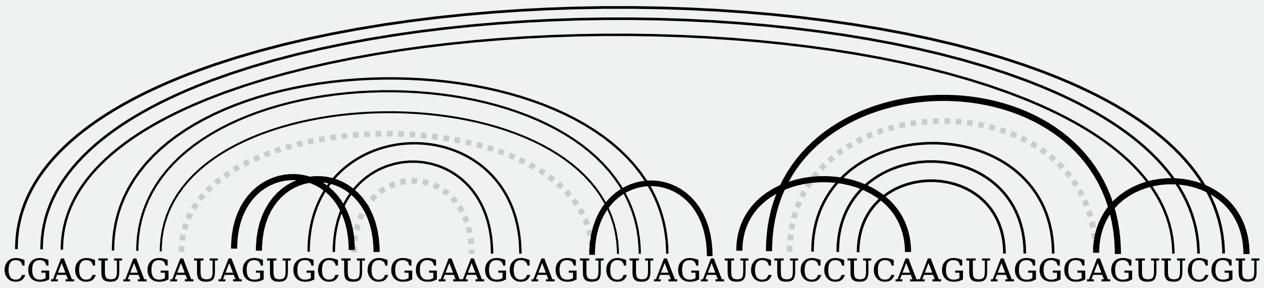


Preliminaries

Arc-structure levels

There are 4 levels (initially introduced by P.A. Evans) :

- UNLIMITED – no restriction at all,
- CROSSING – no base incident to more than one arc,

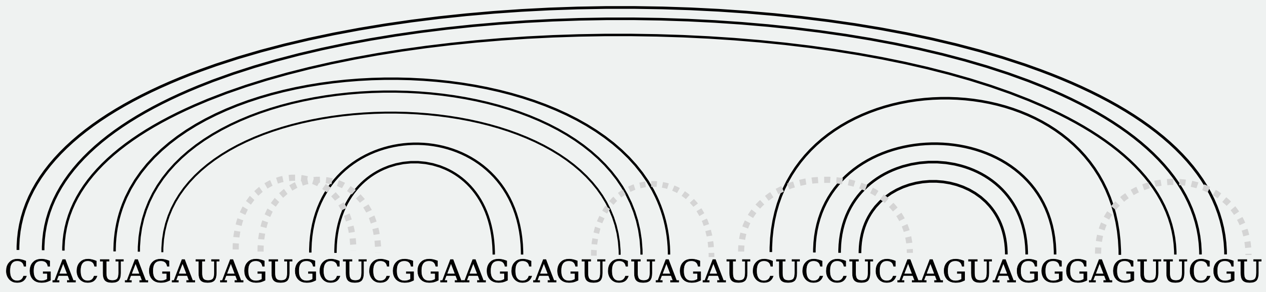


Preliminaries

Arc-structure levels

There are 4 levels (initially introduced by P.A. Evans) :

- UNLIMITED – no restriction at all,
- CROSSING – no base incident to more than one arc,
- NESTED – no base incident to more than one arc and no arcs are crossing,

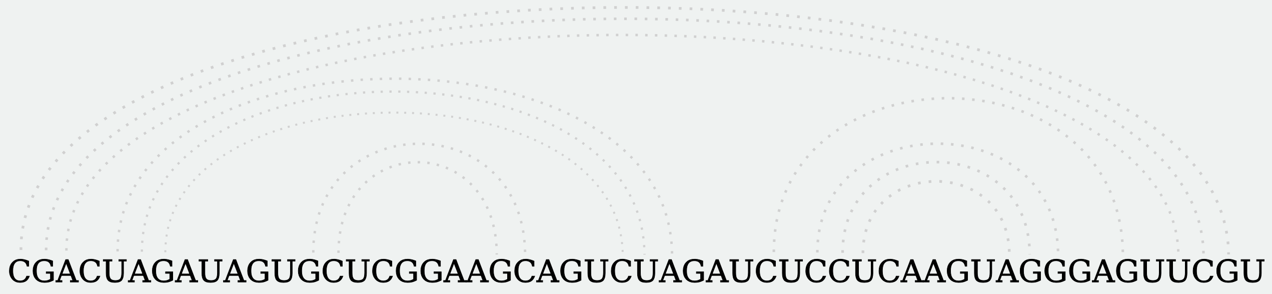


Preliminaries

Arc-structure levels

There are 4 levels (initially introduced by P.A. Evans) :

- UNLIMITED – no restriction at all,
- CROSSING – no base incident to more than one arc,
- NESTED – no base incident to more than one arc and no arcs are crossing,
- PLAIN – no arc (does not carry any structural information ⇒ not considered here).



Preliminaries

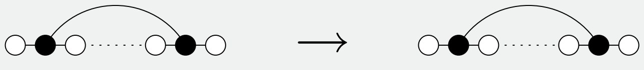
Edit operations

Initially introduced by Lin et al. :

Substitution operations, inducing renaming of bases :

base-(mis)match $(w_m : \Sigma^2 \rightarrow \mathbb{R})$

arc-(mis)match $(w_{am} : \Sigma^4 \rightarrow \mathbb{R})$



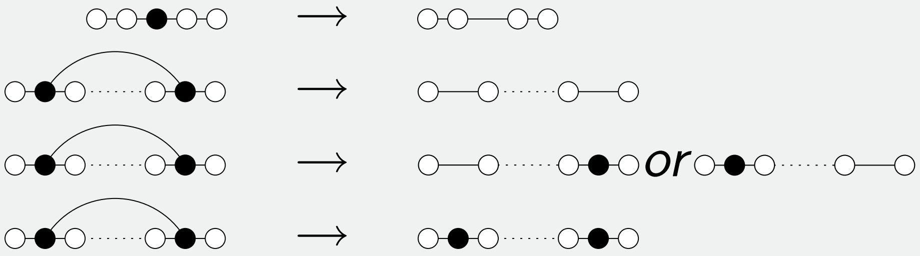
Deletion operations, inducing deletion of bases and/or of arcs :

base-deletion $(w_d : \Sigma \rightarrow \mathbb{R})$

arc-removing $(w_r : \Sigma^2 \rightarrow \mathbb{R})$

arc-altering $(w_a : \Sigma^3 \rightarrow \mathbb{R})$

arc-breaking $(w_b : \Sigma^4 \rightarrow \mathbb{R})$



Three edit models

- **I** : all substitution operations, base-deletions and arc-removings,
- **II** : the operations of model I and arc-alterings,
- **III** : the operations of model II and arc-breakings.

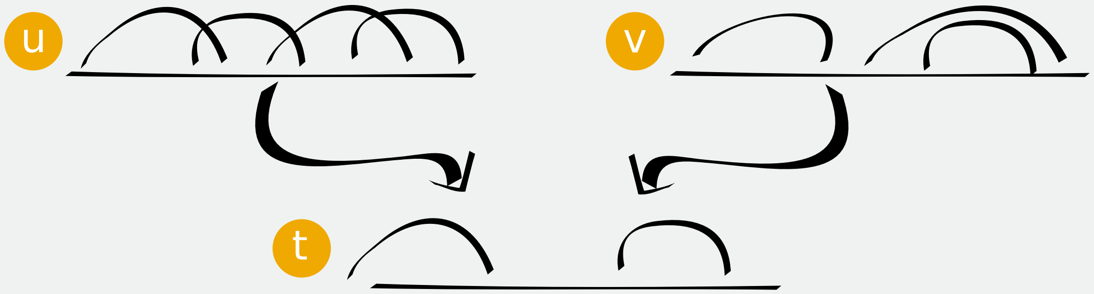
Preliminaries

Common K-subsequence – K-supersequence

For any model $K \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}\}$, given 3 sequences u, v and t,

- t is a common K-subsequence of u and v iff t can both be obtained from a series of edit operations of K from u and v

Common K-subsequence



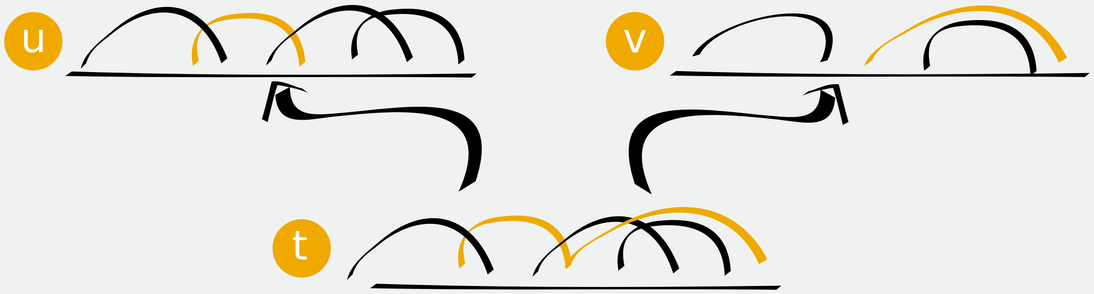
Preliminaries

Common K-subsequence – K-supersequence

For any model $K \in \{I, II, III\}$, given 3 sequences u, v and t,

- t is a common K-subsequence of u and v iff t can both be obtained from a series of edit operations of K from u and v
- t is a common K-supersequence of u and v iff both u and v can be obtained from a series of edit operations of K from t

Common K-supersequence



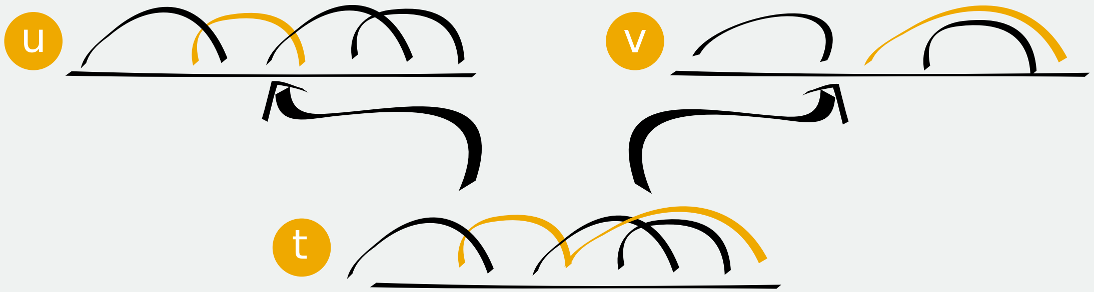
Preliminaries

Common K-subsequence – K-supersequence

For any model $K \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}\}$, given 3 sequences u, v and t,

- t is a common K-subsequence of u and v iff t can both be obtained from a series of edit operations of K from u and v
- t is a common K-supersequence of u and v iff both u and v can be obtained from a series of edit operations of K from t
- The cost of a common K-(super/sub)sequence is defined as the sum of the costs of the two series of edit operations

Common K-supersequence



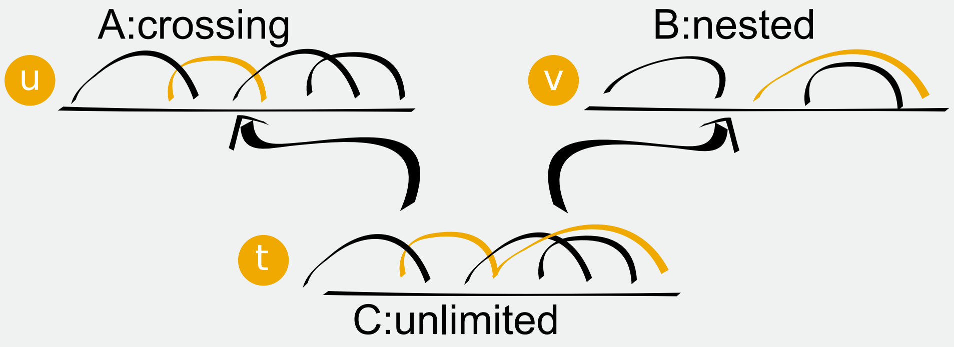
Problem definition

Arc-annotated sequence alignment

$\textsc{Align}(A, B, K) \rightarrow C$:

- Input : $K \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}\}$, $A, B, C \in \{\textsc{plain}, \textsc{nested}, \textsc{crossing}, \textsc{unlimited}\}$, two arc-annotated sequences u and v respectively of A-type and B-type
- Output : A common K-supersequence of u and v of $\underline{C}$-type and of minimal cost

$\textsc{Align}(A, B, K) \rightarrow C$



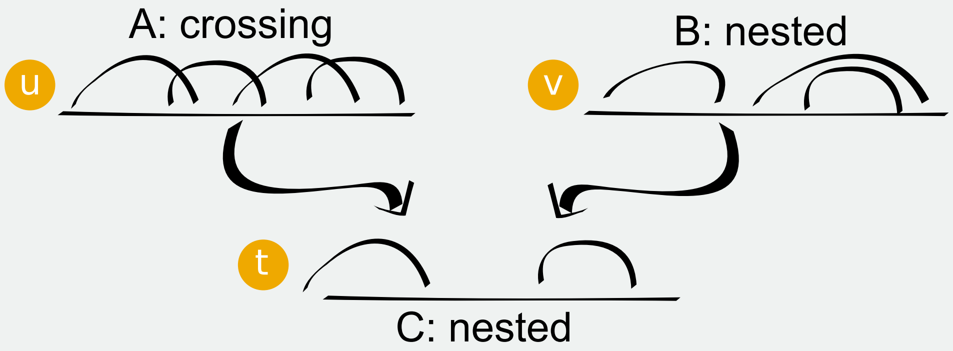
Problem definition

Edit distance

$\textsc{Edit}(A, B, K)$:

- Input : $K \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}\}$, $A, B \in \{\textsc{plain}, \textsc{nested}, \textsc{crossing}, \textsc{unlimited}\}$, two arc-annotated sequences u and v respectively of A-type and B-type
- Output : A common K-subsequence of u and v of $\textsc{unlimited}$-type and of minimal cost

$\textsc{Edit}(A, B, K)$



Problem definition

Edit distance

$\textsc{Edit}(A, B, K)$: No structural constraints on the common subsequence

- Input : $K \in \{\text{I}, \text{II}, \text{III}\}$, $A, B \in \{\textsc{plain}, \textsc{nested}, \textsc{crossing}, \textsc{unlimited}\}$, two arc-annotated sequences u and v respectively of A-type and B-type
- Output : A common K-subsequence of u and v of $\textsc{unlimited}$-type and of minimal cost

Arc-annotated sequence alignment

$\textsc{Align}(A, B, K) \rightarrow C$:

- Input : $K \in \{\text{I}, \text{II}, \text{III}\}$, $A, B, C \in \{\textsc{plain}, \textsc{nested}, \textsc{crossing}, \textsc{unlimited}\}$, two arc-annotated sequences u and v respectively of A-type and B-type
- Output : A common K-supersequence of u and v of C-type and of minimal cost

Outline

Ordered trees and the edit model I

Considering model I : all substitution, base-deletions and arc-removings

$A \times B \longrightarrow C$	Edit	model I
Nest × Nest → Nest		$O(n^4)$ – Jiang '95
Nest × Nest → Cros Nest × Nest → Unlim	×	$O(n^3 \log(n))$ – Klein '98
Cros × Nest → Cros Cros × Nest → Unlim	×	$O(n^3 \log(n))$ – Ma '01
Cros × Cros → Cros Cros × Cros → Unlim	×	**NP**-complete – Ma '01
Unlim × Nest → Unlim	×	$O(n^3 \log(n))$
Unlim × Cros → Unlim	×	**NP**-complete – Ma '01
Unlim × Unlim → Unlim	×	**NP**-complete – Ma '01

- Align$(A, B, K) \longrightarrow$ unlimited $\equiv$ Edit(A, B, K)

Ordered trees and the edit model I

Considering model I : all substitution, base-deletions and arc-removings

$A \times B \longrightarrow C$	EDIT	model I
NEST × NEST ⟶ NEST		$O(n^4)$ – Jiang '95
NEST × NEST ⟶ CROS NEST × NEST ⟶ UNLIM	×	$O(n^3 \log(n))$ – Klein '98
CROS × NEST ⟶ CROS CROS × NEST ⟶ UNLIM	×	$O(n^3 \log(n))$ – Ma '01
CROS × CROS ⟶ CROS CROS × CROS ⟶ UNLIM	×	**NP**-complete – Ma '01
UNLIM × NEST ⟶ UNLIM	×	$O(n^3 \log(n))$
UNLIM × CROS ⟶ UNLIM	×	**NP**-complete – Ma '01
UNLIM × UNLIM ⟶ UNLIM	×	**NP**-complete – Ma '01

- ALIGN(A, B, K) ⟶ UNLIMITED ≡ EDIT(A, B, K)
- ALIGN(A, B, I) ⟶ UNLIM ≡ ALIGN(A, B, I) ⟶ CROS. UNLIMITED is meaningless.
- ALIGN(UNLIM, B, I) ⟶ UNLIM ≡ ALIGN(CROS, B, I) ⟶ CROS

Ordered trees and the edit model I

Considering model I : all substitution, base-deletions and arc-removings

$A \times B \rightarrow C$	Edit	model I
Nest × Nest → Nest		$O(n^4)$ – Jiang '95
Nest × Nest → Cros Nest × Nest → Unlim	×	$O(n^3 \log(n))$ – Klein '98
Cros × Nest → Cros Cros × Nest → Unlim	×	$O(n^3 \log(n))$ – Ma '01
Cros × Cros → Cros Cros × Cros → Unlim	×	**NP**-complete – Ma '01
Unlim × Nest → Unlim	×	$O(n^3 \log(n))$
Unlim × Cros → Unlim	×	**NP**-complete – Ma '01
Unlim × Unlim → Unlim	×	**NP**-complete – Ma '01

- Align(A, B, K) → unlimited ≡ Edit(A, B, K)
- Align(A, B, I) → Unlim ≡ Align(A, B, I) → Cros. unlimited is meaningless.
- Align(Unlim, B, I) → Unlim ≡ Align(Cros, B, I) → Cros
- Comparing arc-annotated sequences of Nested types when considering the edit model I amounts to comparing ordered trees.

LAPCS and the edit model II

Considering model II : the operations of model I and arc-alterings

$A \times B \to C$	EDIT	model II
NEST × NEST → NEST		
NEST × NEST → CROS		
NEST × NEST → UNLIM	×	**NP**-complete – Lin '02
CROS × NEST → CROS		
CROS × NEST → UNLIM	×	**NP**-complete – Evans '99
CROS × CROS → CROS		
CROS × CROS → UNLIM	×	**NP**-complete – Evans '99
UNLIM × NEST → UNLIM	×	
UNLIM × CROS → UNLIM	×	
UNLIM × UNLIM → UNLIM	×	

- The LAPCS(A,B) problem is a specific case of EDIT(A, B, II), provided that the score system for edit operations is correctly chosen : $w_d = w_a = 1$ and $w_r = 2$.

Lapcs and the edit model II

Considering model II : the operations of model I and arc-alterings

$A \times B \longrightarrow C$	Edit	model II
Nest × Nest → Nest		
Nest × Nest → Cros		
Nest × Nest → Unlim	×	**NP**-complete – Lin '02
Cros × Nest → Cros		
Cros × Nest → Unlim	×	**NP**-complete – Evans '99
Cros × Cros → Cros		
Cros × Cros → Unlim	×	**NP**-complete – Evans '99
Unlim × Nest → Unlim	×	
Unlim × Cros → Unlim	×	
Unlim × Unlim → Unlim	×	

- The Lapcs(A,B) problem is a specific case of Edit(A, B, II), provided that the score system for edit operations is correctly chosen : $w_d = w_a = 1$ and $w_r = 2$.
- The Lapcs(Nest,Nest) problem is **NP**-complete. So does Edit(Nest, Nest, II).

Lapcs and the edit model II

Considering model II : the operations of model I and arc-alterings

$A \times B \rightarrow C$	Edit	model II
Nest × Nest → Nest		
Nest × Nest → Cros		
Nest × Nest → Unlim	×	**NP**-complete – Lin ’02
Cros × Nest → Cros		
Cros × Nest → Unlim	×	**NP**-complete – Evans ’99
Cros × Cros → Cros		
Cros × Cros → Unlim	×	**NP**-complete – Evans ’99
Unlim × Nest → Unlim	×	
Unlim × Cros → Unlim	×	
Unlim × Unlim → Unlim	×	

- The Lapcs(A,B) problem is a specific case of Edit(A, B, II), provided that the score system for edit operations is correctly chosen : $w_d = w_a = 1$ and $w_r = 2$.
- The Lapcs(Nest,Nest) problem is **NP**-complete. So does Edit(Nest, Nest, II).
- It remains four specific problems – we will answer to all of them.

The general edit distance and the edit model III

Considering model III : the operations of model II and arc-breakings

$A \times B \rightarrow C$	EDIT	model III
NEST × NEST → NEST		
NEST × NEST → CROS		
NEST × NEST → UNLIM	×	**NP**-complete – Blin '03
CROS × NEST → CROS		
CROS × NEST → UNLIM	×	Max SNP-hard – Jiang '02
CROS × CROS → CROS		
CROS × CROS → UNLIM	×	Max SNP-hard – Jiang '02
UNLIM × NEST → UNLIM	×	
UNLIM × CROS → UNLIM	×	
UNLIM × UNLIM → UNLIM	×	

- ALIGN$(A, B, III) \rightarrow$ UNLIM $\equiv$ EDIT(A, B, III)

The general edit distance and the edit model III

Considering model III : the operations of model II and arc-breakings

$A \times B \longrightarrow C$	Edit	model III
Nest × Nest → Nest		
Nest × Nest → Cros		
Nest × Nest → Unlim	×	**NP**-complete – Blin ’03
Cros × Nest → Cros		
Cros × Nest → Unlim	×	Max SNP-hard – Jiang ’02
Cros × Cros → Cros		
Cros × Cros → Unlim	×	Max SNP-hard – Jiang ’02
Unlim × Nest → Unlim	×	
Unlim × Cros → Unlim	×	
Unlim × Unlim → Unlim	×	

- $\text{Align}(A, B, III) \longrightarrow \text{Unlim} \equiv \text{Edit}(A, B, III)$
- It remains four specific problems – we will answer one of them.

Derived results from existing model

On the whole

$A \times B \longrightarrow C$	model I	model II	model III
Nest × Nest → Nest	P	?	?
Nest × Nest → Cros			
Nest × Nest → Unlim		NPC	NPC
Cros × Nest → Cros		?	?
Cros × Nest → Unlim		NPC	M-SNP H
Cros × Cros → Cros	NPC	?	?
Cros × Cros → Unlim		NPC	M-SNP H
Unlim × Nest → Unlim	P		
Unlim × Cros → Unlim	NPC		
Unlim × Unlim → Unlim			

Outline

1. Algorithmic of RNA
2. Arc-annotated sequences
3. The ALIGN hierarchy
4. Derived results from existing model
5. New results
6. Conclusion

Two polynomial time algorithms

Align(Nested, Nested, $\{II, III\}$) → Nested are polynomial

- We propose to resolve those problems with dynamic programming by extending the algorithm proposed by Jiang et al in 95 – dealing with tree alignment – with the operations of arc-breaking and arc-altering

$$2. A(\alpha(u)\circ v, \beta(w)\circ x) \;=\; \min \begin{cases} \ldots \\ w_a(\alpha, b) + \min\{A(u,y) + A(v,z) \mid y\circ b\circ z = \beta(w)\circ x\} & \text{– arc-altering} \\ w_a(\beta, b) + \min\{A(y,w) + A(z,x) \mid y\circ b\circ z = \alpha(u)\circ v\} & \text{– arc-altering} \end{cases}$$

$$3. A(b\circ v, \beta(w)\circ x) \;=\; \min \begin{cases} \ldots \\ w_a(\beta, b) + \min\{A(y,w) + A(z,x) \mid y\circ z = v\} & \text{– arc-altering} \\ w_a(\beta, b_2) + \min\{A(y,w) + A(z,x) \mid y\circ b_2\circ z = b\circ v\} & \text{– arc-altering} \\ w_b(\beta, b, b_2) + \min\{A(y,w) + A(z,x) \mid x\circ b_2\circ z = v\} & \text{– arc-breaking} \end{cases}$$

$$4. A(\alpha(u)\circ v, b\circ x) \;=\; \min \begin{cases} \ldots \\ w_a(\alpha, b) + \min\{A(u,y) + A(v,z) \mid y\circ z = x\} & \text{– arc-altering} \\ w_a(\alpha, b_2) + \min\{A(u,y) + A(v,z) \mid y\circ b_2\circ z = b\circ x\} & \text{– arc-altering} \\ w_b(\alpha, b, b_2) + \min\{A(u,y) + A(v,z) \mid y\circ b_2\circ z = x\} & \text{– arc-breaking} \end{cases}$$

Two polynomial time algorithms

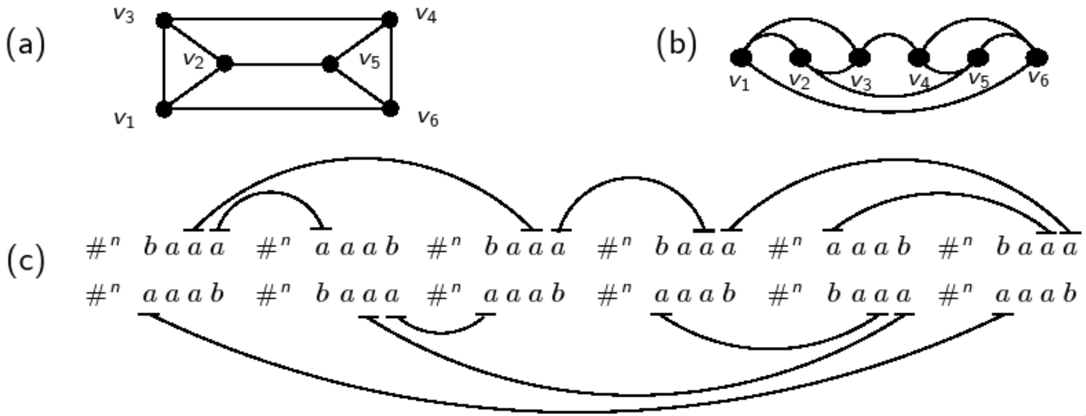
$\textsc{Align}(\textsc{Nested}, \textsc{Nested}, \{\mathrm{II}, \mathrm{III}\}) \rightarrow \textsc{Nested}$ are polynomial

- We prove that the complexity remains unchanged : $O(n^4)$ where n is the length of the longest sequence
- This result is somehow unexpected since the associate edit problems $\textsc{Edit}(\textsc{Nested}, \textsc{Nested}, II)$ and $\textsc{Edit}(\textsc{Nested}, \textsc{Nested}, III)$ are **NP**-complete
- It shows that imposing structural constraints on the type of the common supersequence is an adequate way for lower complexity of untractable problem

An hardness result

Align(Nested, Nested, II) → Crossing is **NP**-complete

- We show that relaxing the constraints on crossing arcs in the common supersequence makes the problem difficult
- We propose a reduction from the Maximum Independent Set problem on a specific class of cubic graphs
- A 2-page book embedding can be found in poly-time for them
- Quite technical proof mainely using that no arc-breaking is allowed – therefore it is not easily extensible to the model III



Outline

Conclusion

Our contribution

- We proposed a new framework for comparing arc-annotated sequences, namely the ALIGN hierarchy
- Its study is relevant both from a practical and theoretical perspective
- We provided two poly-time algorithms and an hardness result
- This results shed a new light on the border between tractability and untractability when dealing with arc-annotated sequences – especially of CROSSING type
- It remains 3 open cases that we conjecture to be **NP**-complete

Questions on "How to compare arc-annotated sequences : the alignment hierarchy"

G. Blin[1] H. Touzet[2]

[1]IGM-LabInfo UMR 8049, Université de Marne La Vallée
gblin@univ-mlv.fr

[2]LIFL, UMR CNRS 8022 Université des Sciences et Technologies de Lille
touzet@lifl.fr