



Stage co-encadré par :  
Sèverine Bérard et Éric Tannier

## Détection de co-évolution de gènes

Master 2 : Informatique à Finalité  
Professionnelle et Recherche Unifiée  
(IFPRU)

Parcours Ingénierie de l'Intelligence  
Artificielle (I2A)

# Présentation du sujet

- **Bioinformatique**
- **Gène** : portion d'ADN qui code une protéine.
- **Adjacence** : relation entre 2 gènes
- **Génome** : ensemble d'adjacences



# Présentation du sujet

- Arbres de gènes
- **Données** sur les adjacences de gènes des espèces actuelles
- **But** : histoire évolutive des adjacences
- **Intérêt** : reconstruction des génomes ancestraux

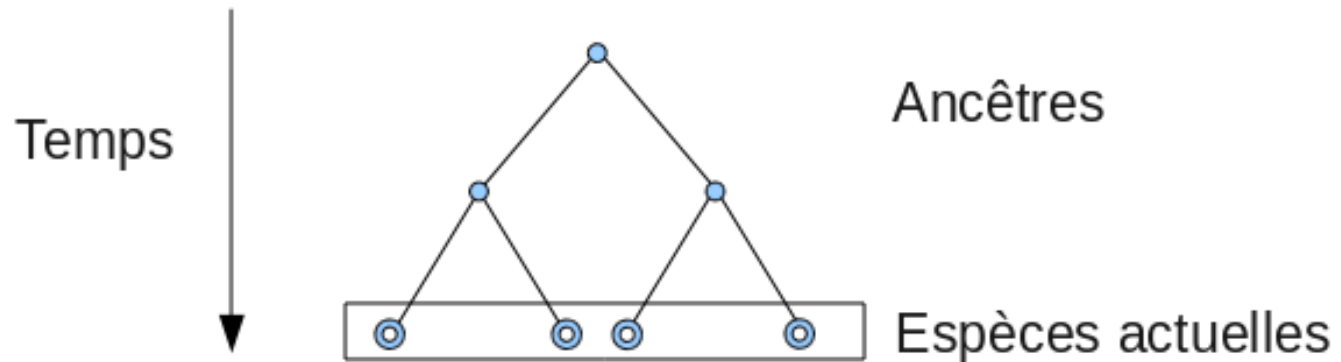


# Plan

1. **Présentation de la problématique au travers d'exemples**
2. Formalisation
3. Extraits de l'algorithme
4. Application à des données réelles
5. Conclusion et perspectives

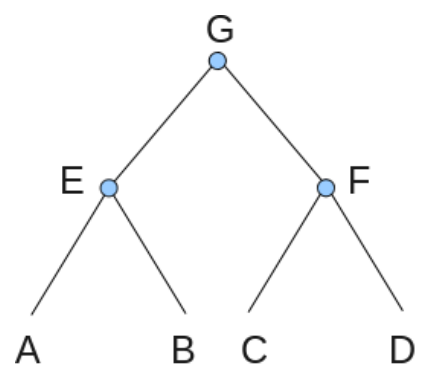
# Définitions

- Une **adjacence** entre 2 gènes  $A_1$  et  $A_2$  se note  $A_1 \sim A_2$  ou  $A_2 \sim A_1$  (symétrie)
- **Arbre phylogénétique** : graphe connexe non cyclique, orienté
  - Arbre de gènes
  - Arbre d'espèces
  - *Arbre d'adjacences*

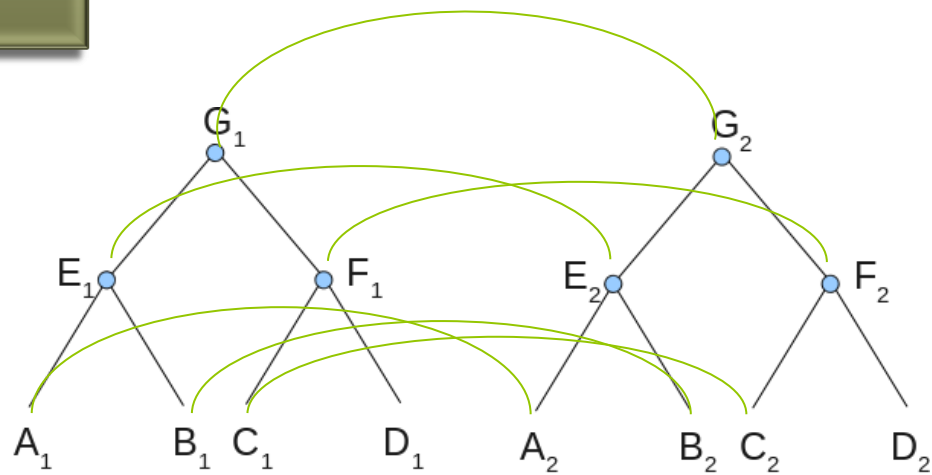


- **Forêt** : ensemble d'arbres

# Exemple 1

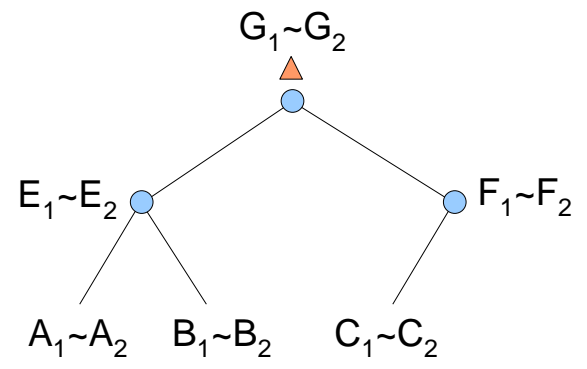


Arbre des espèces  $S$



Arbre des gènes  $G_1$

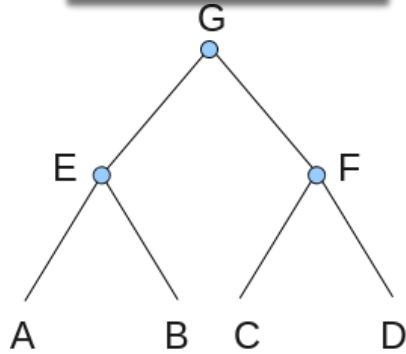
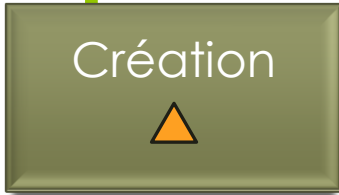
Arbre des gènes  $G_2$



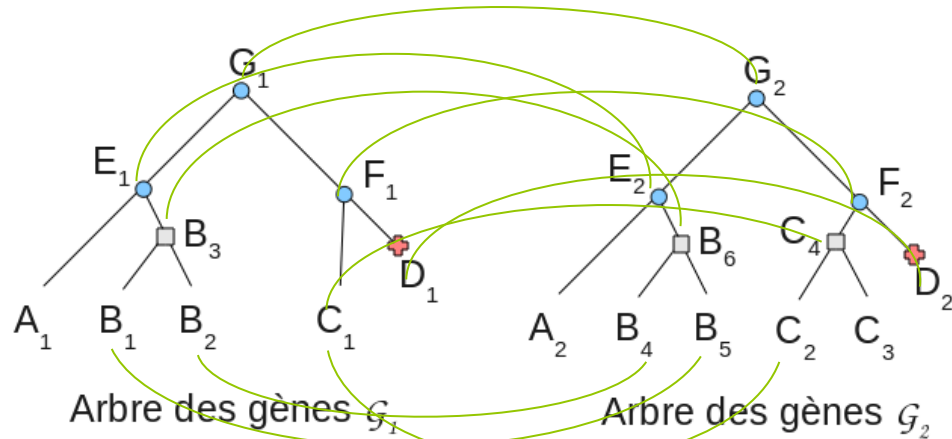
Liste d'adjacence  $\mathcal{L}$  :  $A_1 \sim A_2$ ,  $B_1 \sim B_2$  et  $C_1 \sim C_2$

[Fitch]

# Exemple 2

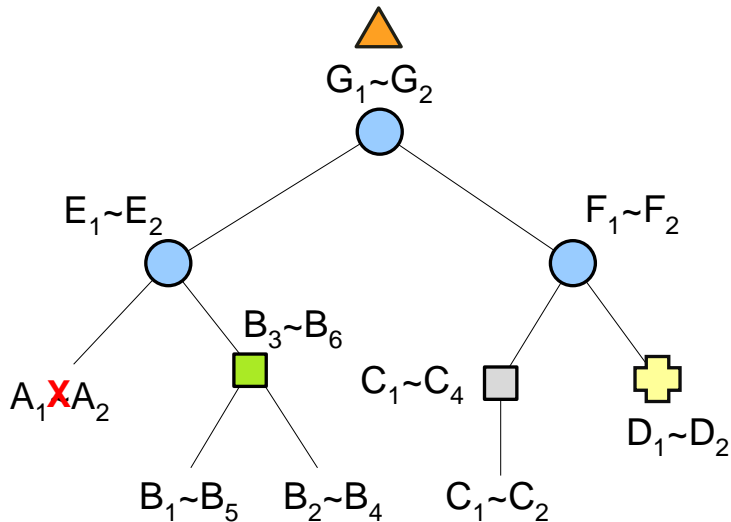


Arbre des espèces  $s$



Arbre des gènes  $G_1$

Arbre des gènes  $G_2$




Liste d'adjacence  $\mathcal{L}$  :  $B_1 \sim B_5$ ,  $B_2 \sim B_4$  et  $C_1 \sim C_2$

# Plan


1. Présentation de la problématique au travers d'exemples
2. **Formalisation**
3. Extraits de l'algorithme
4. Application à des données réelles
5. Conclusion et perspectives




# Événements évolutifs


Spéciation  
Coût : 0 


Espèces  
Gènes  
Adjacences


Duplication de gène  
Coût :  $D_G$  


Perte de gène  
Coût :  $P_G$  

Gènes  
Adjacences

Duplication d'adjacence  
Coût :  $D_A \leq 2 * D_G$  

Perte d'adjacence  
Coût :  $P_A \leq 2 * P_G$  

Création d'adjacence  
Coût :  $Cr$  

Cassure d'adjacence  
Coût :  $Ca$  

Adjacences

# Arbre d'adjacences

**Remarque :** un arbre d'adjacences (ou une forêt d'arbres d'adjacences) est associé(e) à un ou plusieurs arbres de gènes et à une liste d'adjacences.

- Feuilles :
  - Adjacence actuelle
  - Perte d'adjacence
  - Perte de gène
  - Cassure
- Nœuds internes :
  - Nœud de spéciation
  - Nœud de duplication d'adjacence
  - Nœud de duplication de gène
- Création d'adjacence

# Adjacence Actuelle

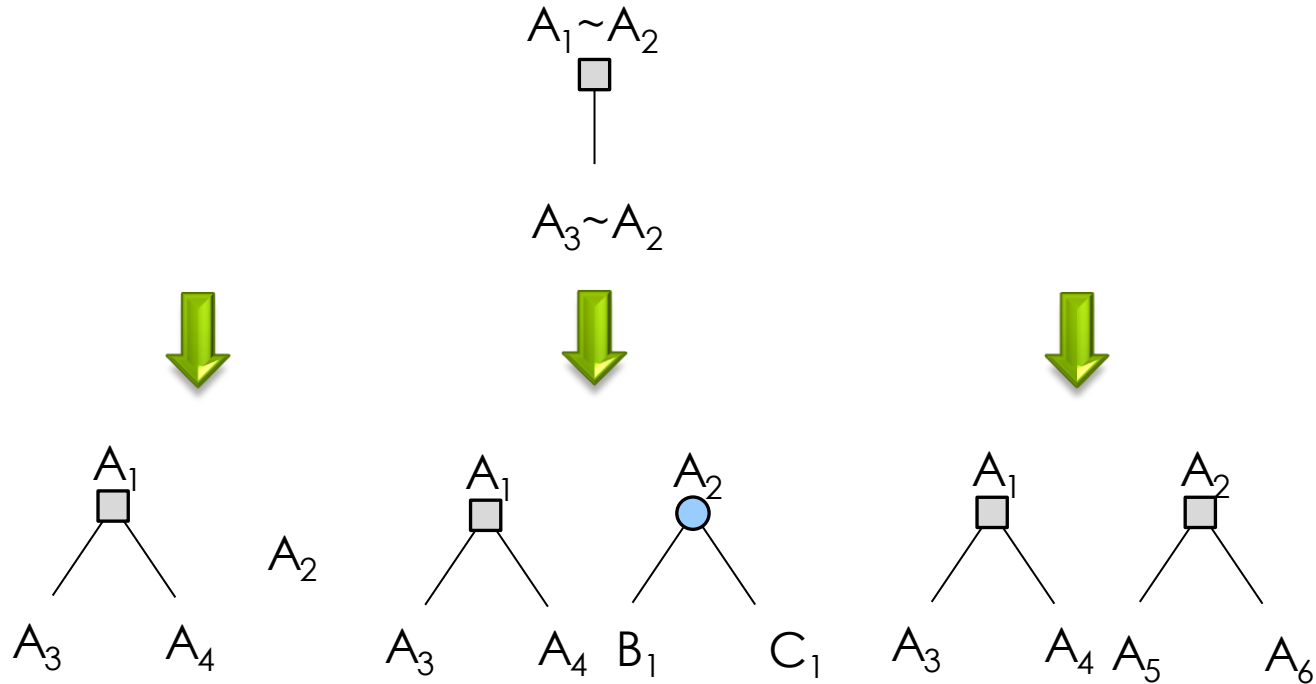
$A_1 \sim A_2$



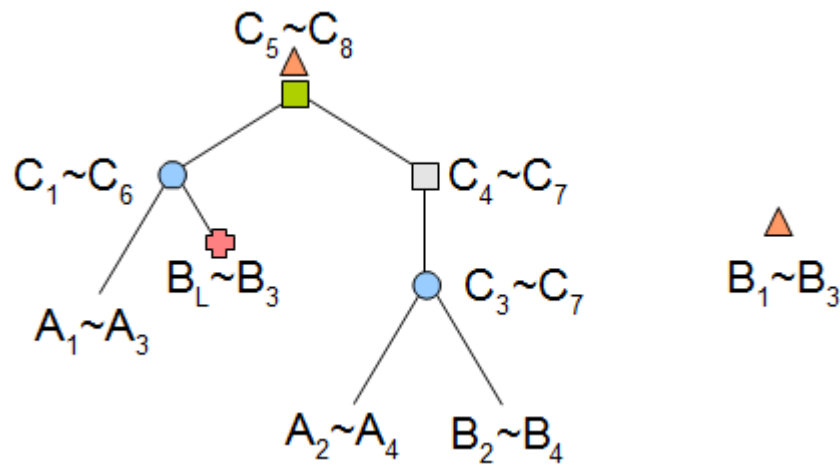
$A_1$

$A_2$

# Nœud de Duplication de Gène



# Nœud de Création

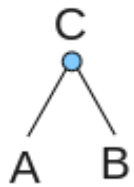


# Problématique

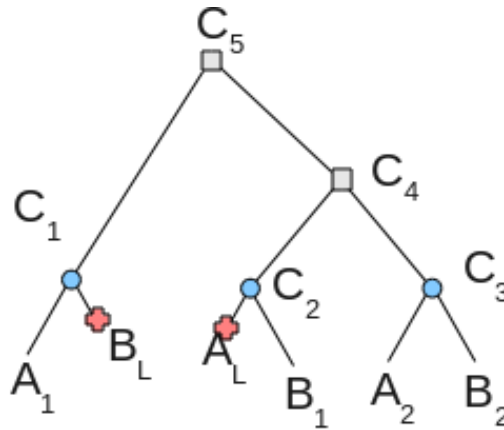
**Reconstruire l'histoire évolutive des adjacences à partir des arbres de gènes et des adjacences actuelles.**

- Limitation :
  - 2 arbres de gènes dont les racines sont de la même espèce
  - Adjacences entre 2 arbres de gènes différents
- Données :
  - 2 arbres de gènes  $G_1$  et  $G_2$
  - Une liste d'adjacences  $L$
  - Un arbre des espèces  $S$
- **Solution** : forêt d'arbres d'adjacences associés à  $G_1$ ,  $G_2$  et  $L$  de coût différentiel minimum.

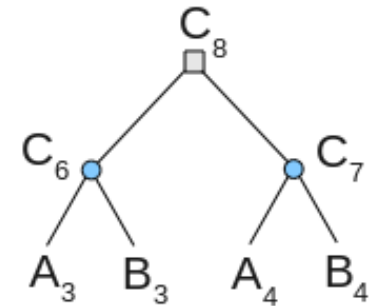
# Exemple



Arbre des espèces  $\mathcal{S}$



Arbre de gènes  $\mathcal{G}_1$

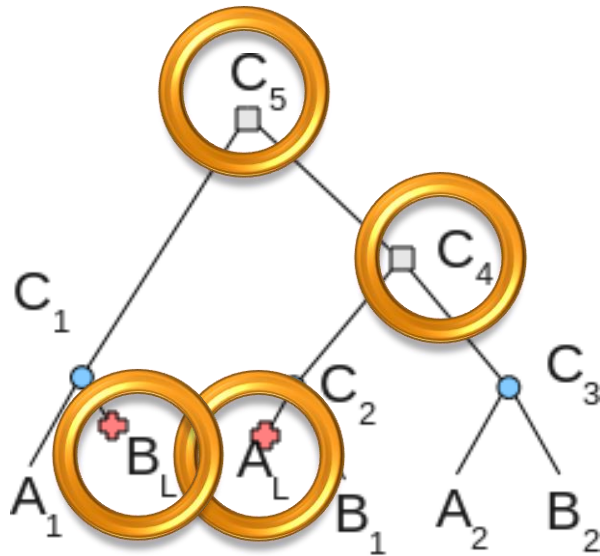


Arbre de gènes  $\mathcal{G}_2$

Liste des adjacences  $\mathcal{L}$  :  $(A_1 \sim A_3, B_1 \sim B_3, A_2 \sim A_4 \text{ et } B_2 \sim B_4)$

# Coûts

- **Coût d'un arbre** : somme des coûts des nœuds de l'arbre.



Arbre de gènes  $G_1$

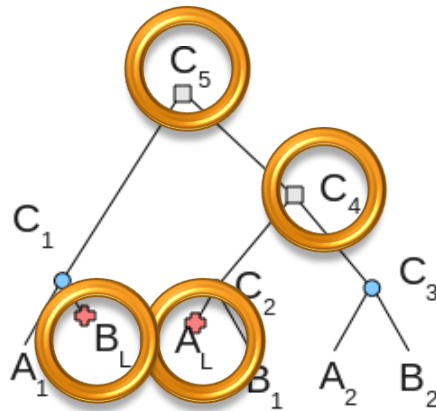
$$\begin{aligned} \text{Coût de } G_1 &= D_G + D_G + P_G + P_G \\ &= 2 * D_G + 2 * P_G \end{aligned}$$

- **Coût d'une forêt** : somme des coûts des arbres de la forêt.

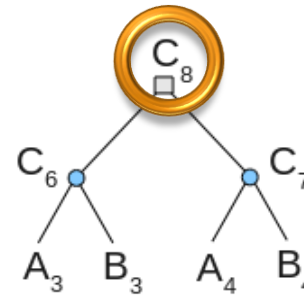


# Coûts

- Coût maximum** : somme des coûts des arbres de gènes  $G_1$  et  $G_2$  et du coût de création des adjacences de  $L$ .



Arbre de gènes  $G_1$



Arbre de gènes  $G_2$

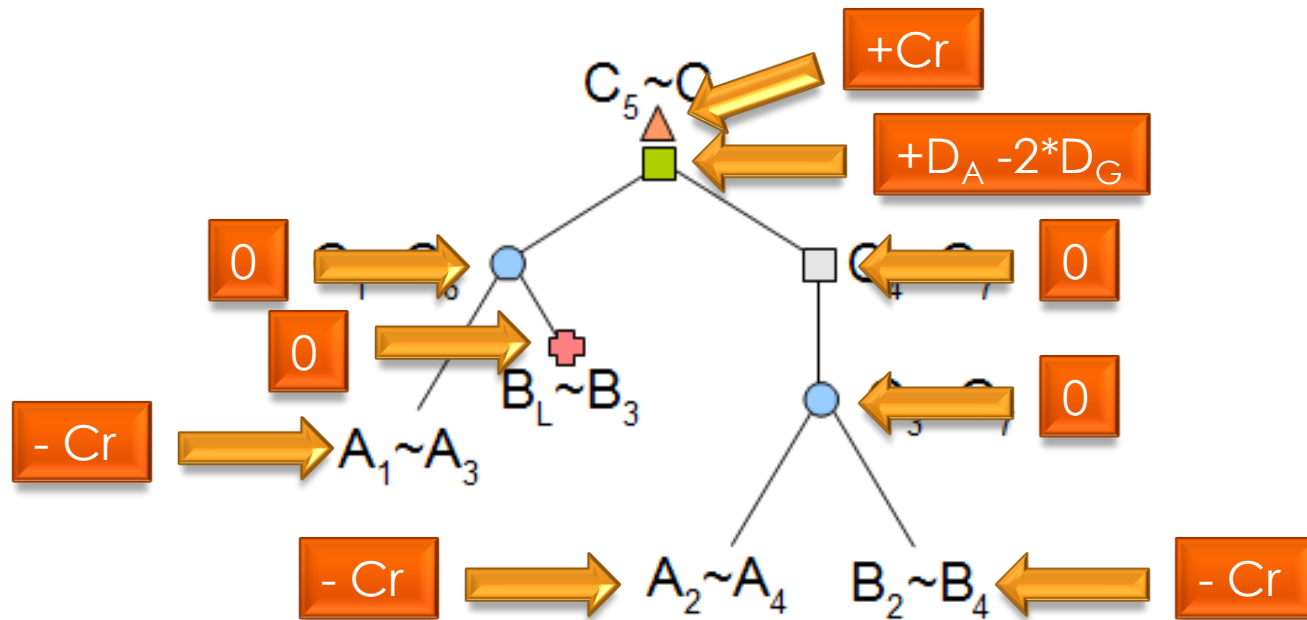
Liste des adjacences  $\mathcal{L}$   $(A_1 \sim A_3, B_1 \sim B_3, A_2 \sim A_4 \text{ et } B_2 \sim B_4)$

$$\begin{aligned}
 \text{Coût maximum} &= \text{Coût de } G_1 + \text{Coût de } G_2 + \text{Coût de } L \\
 &= (2 \cdot D_G + 2 \cdot P_G) + D_G + 4 \cdot Cr
 \end{aligned}$$

# Coûts

- **Coût d'un arbre d'adjacence** : tous les événements sur les adjacences + une partie des événements sur les gènes
- **Ce qu'on cherche à minimiser** : tous les événements sur les adjacences + tous les événements sur les gènes = coût différentiel + coût maximum
- **Coût différentiel d'un arbre d'adjacence** = somme des coûts différentiels des nœuds :
  - Spéciation : 0
  - Duplication de Gène : 0
  - Perte de Gène : 0
  - Duplication d'Adjacence :  $-2*D_G + D_A$
  - Perte d'adjacence :  $-2*P_G + P_A$
  - Création :  $+Cr$
  - Cassure :  $+Ca$
  - Adjacence actuelle :  $-Cr$

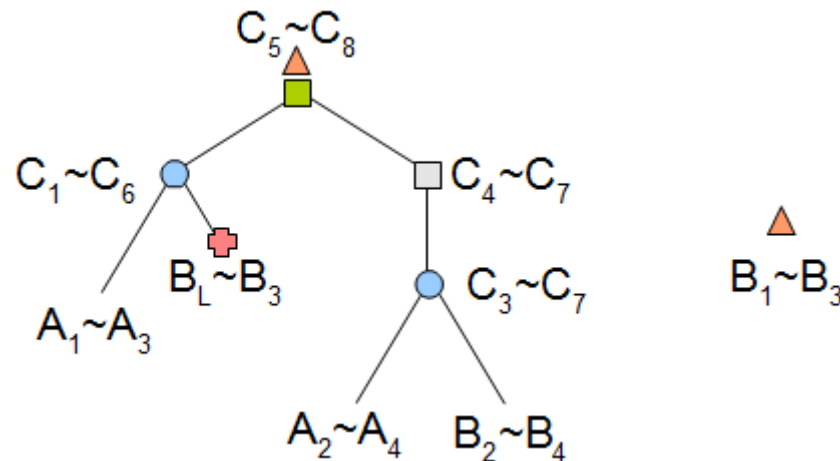
# Coûts



$$\begin{aligned} \text{Coût diff.} &= +Cr + D_A - 2 * D_G + 0 + 0 + 0 + 0 - Cr - Cr - Cr \\ &= D_A - 2 * D_G - 2 * Cr \end{aligned}$$

# Coûts

- Coût de la solution** : somme du coût maximum et du coût différentiel de la forêt d'arbres d'adjacences qui la compose.

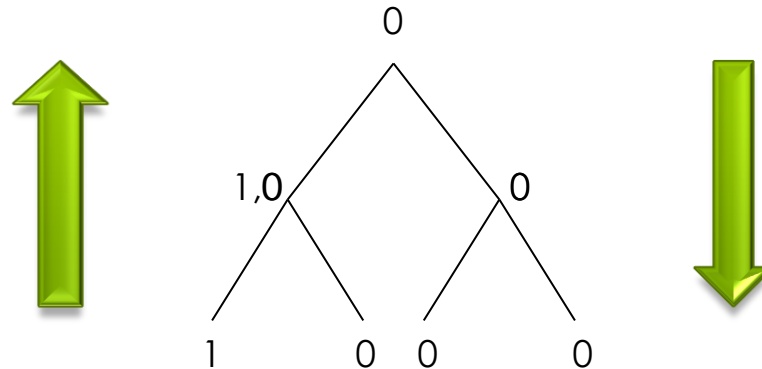


$$\begin{aligned}
 \text{Coût de la solution} &= \text{Coût max} + \text{Coût diff. de } G_1 + \text{Coût diff. de } G_2 \\
 &= (3 \cdot D_G + 2 \cdot P_G + 4 \cdot Cr) + (D_A - 2 \cdot D_G - 2 \cdot Cr) + (Cr - Cr) \\
 &= D_G + D_A + 2 \cdot P_G + 2 \cdot Cr
 \end{aligned}$$

# Plan

1. Présentation de la problématique au travers d'exemples
2. Formalisation
3. **Extraits de l'algorithme**
4. Application à des données réelles
5. Conclusion et perspectives

# Algorithme de Fitch



- Présence ou absence d'adjacence
- Fonction de coût  $c_1$  et  $c_0$
- Appel aux racines

# Algorithmes de calcul de coûts différentiels

- $A_1 \in G_1$  et  $A_2 \in G_2$
- $G_1(A_1)$ ,  $G_2(A_2)$  sous-arbre de racine  $A_1$  ou  $A_2$ .
- $L(A_1, A_2)$  adjacences entre descendants de  $A_1$  et  $A_2$ .
- $\mathbf{c}_1(A_1, A_2)$  calcule le coût différentiel minimum d'une forêt d'arbres d'adjacences associée à  $G_1(A_1)$ ,  $G_2(A_2)$  et  $L(A_1, A_2)$ , forêt dans laquelle **il existe** le nœud de création  $A_1 \sim A_2$  (- **Cr**)
- $\mathbf{c}_0(A_1, A_2)$  calcule le coût différentiel minimum d'une forêt d'arbres d'adjacences associée à  $G_1(A_1)$ ,  $G_2(A_2)$  et  $L(A_1, A_2)$ , forêt dans laquelle **il n'existe pas** le nœud de création  $A_1 \sim A_2$  (**sauf si**  $A_1 \sim A_2 \in L$ )

# Algorithmes de calcul de coûts

$c_1$  et  $c_0$  sont 2 algorithmes « répartiteurs »

$A_2 \backslash A_1$	GèneActuel	Perte	Duplication	Spéciation
GèneActuel	Cas A	Cas C	Cas D	X
Perte		Cas B	Cas C	Cas C
Duplication			Cas G	Cas F
Spéciation				Cas E



# Cas d'arrêt

## Cas A : Gène Actuel/Gène Actuel

- $c1GAGA(n1, n2) = Cr-Cr-Cr$  si  $n1 \sim n2 \in L$ ,  $Cr+Ca-Cr$  sinon
- $c0GAGA(n1, n2) = Cr-Cr$  si  $n1 \sim n2 \in L$ , 0 sinon

C	$n1 \sim n2 \in L$	$n1 \sim n2 \notin L$
• $C_1GAGA(n_1, n_2)$	▲ $n1 \sim n2$	▲ $n1 \times n2$
• $C_0GAGA(n_1, n_2)$	▲ $n1 \sim n2$	∅

- $c0PGDS(n1, n2) = 0$

# Cas récursif (D)

## Pseudo cas d'arrêt

### Cas D : GèneActuel/Duplication

- $$C_1GAD(n_1, n_2) = \min(c_1(n_1, fg(n_2)) + c_0(n_1, fd(n_2)),$$

$$c_0(n_1, fg(n_2)) + c_1(n_1, fd(n_2)),$$

$$c_1(n_1, fg(n_2)) + c_1(n_1, fd(n_2)) + Cr,$$

$$c_0(n_1, fg(n_2)) + c_0(n_1, fd(n_2)) + Ca)$$
- $$C_0GAD(n_1, n_2) = \min(c_0(n_1, fg(n_2)) + c_0(n_1, fd(n_2)),$$

$$c_1(n_1, fg(n_2)) + c_0(n_1, fd(n_2)) + Cr,$$

$$c_0(n_1, fg(n_2)) + c_1(n_1, fd(n_2)) + Cr,$$

$$c_1(n_1, fg(n_2)) + c_1(n_1, fd(n_2)) + 2*Cr)$$

# Cas récursif (D)

## Pseudo cas d'arrêt

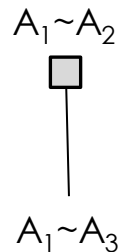
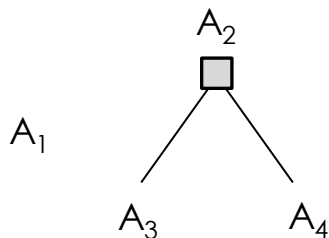
### Cas D : Gène Actuel/Duplication

- $$C_1GAD(A_1, A_2) = \min(c_1(A_1, A_3) + c_0(A_1, A_4)),$$

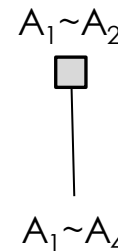
$$c_0(A_1, A_3) + c_1(A_1, A_4)),$$

$$c_1(A_1, A_3 + c_1(A_1, A_4)) + Cr,$$

$$c_0(A_1, A_3) + c_0(A_1, A_4)) + Ca)$$



ou



# Cas récurrents (E, F et G)

## Cas E : Spéciation/Spéciation

- $$C_1SS(n_1, n_2) = \min(c_1(fg(n_1), fg(n_2)) + c_1(fd(n_1), fd(n_2)),$$

$$c_1(fg(n_1), fg(n_2)) + c_0(fd(n_1), fd(n_2)) + Ca,$$

$$c_0(fg(n_1), fg(n_2)) + c_1(fd(n_1), fd(n_2)) + Ca,$$

$$c_0(fg(n_1), fg(n_2)) + c_0(fd(n_1), fd(n_2)) + 2*Ca)$$
- $$C_0SS(n_1, n_2) = \min(c_0(fg(n_1), fg(n_2)) + c_0(fd(n_1), fd(n_2)),$$

$$c_1(fg(n_1), fg(n_2)) + c_0(fd(n_1), fd(n_2)) + Cr,$$

$$c_0(fg(n_1), fg(n_2)) + c_1(fd(n_1), fd(n_2)) + Cr,$$

$$c_1(fg(n_1), fg(n_2)) + c_1(fd(n_1), fd(n_2)) + 2*Cr)$$

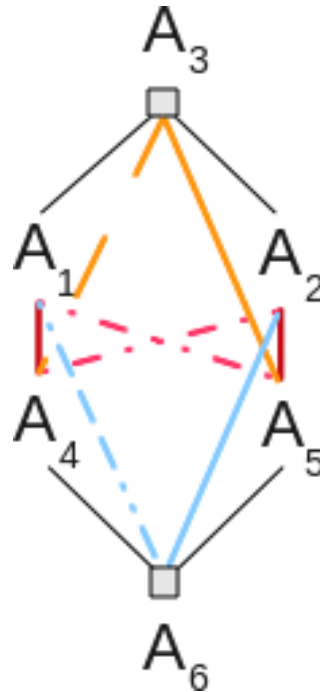
# Cas récurrents (E, F et G)

## Cas F : Spéciation/Duplication

- $C_1SD(n_1, n_2) = \min(c_1(n_1, fg(n_2)) + c_0(n_1, fd(n_2)),$   
 $c_0(n_1, fg(n_2)) + c_1(n_1, fd(n_2)),$   
 $c_1(n_1, fg(n_2)) + c_1(n_1, fd(n_2)) + Cr,$   
 $c_0(n_1, fg(n_2)) + c_0(n_1, fd(n_2)) + Ca)$
- $C_0SD(n_1, n_2) = \min(c_0(n_1, fg(n_2)) + c_0(n_1, fd(n_2)),$   
 $c_1(n_1, fg(n_2)) + c_0(n_1, fd(n_2)) + Cr,$   
 $c_0(n_1, fg(n_2)) + c_1(n_1, fd(n_2)) + Cr,$   
 $c_1(n_1, fg(n_2)) + c_1(n_1, fd(n_2)) + 2*Cr)$

# Cas récurrents (E, F et G)

## Cas G : Duplication/Duplication



# Preuve d'arrêt

## Propriété :

Pour tous couples de nœuds  $n_1 \in G_1$  et  $n_2 \in G_2$ , les algorithmes  $c_1$  et  $c_0$  s'arrêtent.

## Preuve :

Si  $c_1$  ou  $c_0$  fait appel à :

- cas A, B ou C : arrêt
- cas D, E, F ou G : récursivité sur un des 8 couples de nœuds suivants :
  - $(n_1, fg(n_2) \text{ ou } fd(n_2))$ ,
  - $(fg(n_1) \text{ ou } fd(n_1), n_2)$ ,
  - $(fg(n_1) \text{ ou } fd(n_1), fg(n_2) \text{ ou } fd(n_2))$

# Preuve d'optimalité

- Cas A, B et C : cas simples sur des feuilles
- Cas D (Gène Actuel / Duplication) :  
preuve par récurrence
- Cas E, F et G preuve par récurrence sur  
les 3 cas en même temps



# Algorithme DéCo

```
DéCo( $G_1, G_2, S, L$ )  
{  
  renvoyer(coût maximum +  
    min( $c_1(\text{racine}(G_1), \text{racine}(G_2)),$   
       $c_0(\text{racine}(G_1), \text{racine}(G_2))$ ))  
}
```

# Complexité

- Algorithmes de calcul des coûts (cas A à G) : programmation dynamique  
=>  $\mathcal{O}(n*m)$
- Algorithme DéCo a donc une complexité totale en  $\mathcal{O}(n*m)$  : complexité quadratique

# Plan

1. Présentation de la problématique au travers d'exemples
2. Formalisation
3. Extraits de l'algorithme
4. **Application aux données réelles**
5. Conclusion et perspectives

# Données réelles

```

test (~:/Documents/Projet) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
test x
1
2 SEQ physcomitrella_patens ESTEXT_FGENESH1_PG.C_870033 scaffold_87 412472 415063 1 ESTEXT_FGENESH1_PG.C_870033 XM_001766686.1
3 SEQ physcomitrella_patens FGENESH1_PG.SCAFFOLD_11000186 scaffold_11 2380140 2382460 1 FGENESH1_PG.SCAFFOLD_11000186 XM_001753862.1
4 SEQ physcomitrella_patens E_GW1.6.149.1 scaffold_6 3042228 3043431 -1 E_GW1.6.149.1 XM_001752872.1
5 SEQ physcomitrella_patens FGENESH1_PG.SCAFFOLD_5000102 scaffold_5 1970677 1972372 1 FGENESH1_PG.SCAFFOLD_5000102 XM_001752574.1
6 SEQ physcomitrella_patens FGENESH1_PG.SCAFFOLD_6000190 scaffold_6 3298723 3301564 -1 FGENESH1_PG.SCAFFOLD_6000190 XM_001752883.1
7 SEQ physcomitrella_patens FGENESH1_PG.SCAFFOLD_2000211 scaffold_2 2147390 2147800 -1 FGENESH1_PG.SCAFFOLD_2000211 XM_001751912.1
8 SEQ physcomitrella_patens GW1.309.29.1 scaffold_309 338293 339188 1 GW1.309.29.1 XM_001781626.1
9 DATA
10 (((GW1.309.29.1:0.0953,E_GW1.6.149.1:0.4460):0.1221,FGENESH1_PG.SCAFFOLD_2000211:1.1438):0.1977,
    ((ESTEXT_FGENESH1_PG.C_870033:0.1448,FGENESH1_PG.SCAFFOLD_11000186:0.2114):0.0849,
    (FGENESH1_PG.SCAFFOLD_5000102:0.1769,FGENESH1_PG.SCAFFOLD_6000190:0.1891):0.1768):0.9755):0.0000;
11 //
Plain Text Tab Width: 4 Ln 10, Col 178 INS

```

# Données réelles

Tulip

File Edit Algorithm Graph View Options Windows Help

Graph Editor D:/StageCIRAD/Algo/DeCo/Adjacences/magnoliophyta.tlp \*

Property Nodes Edges

	Id	viewAdj
1	0	355
2	1	745
3	2	1156
4	3	517
5	4	1224
6	5	710
7	6	1198

selected only To labels Set all

Displayed properties:  All  User  View

Name	Type	Range
colNoeuds	Color	Local
viewAdj	String	Local
viewBorderColor	Color	Local
viewBorderWidth	Metric	Local
viewColor	Color	Local
viewFont	String	Local
viewFontSize	Integer	Local
viewLabel	String	Local
viewLabelColor	Color	Local
viewLabelPosition	Integer	Local

Remove Import CSV Data Copy New

Graph Editor View Editor

Node Link Diagram view : unnamed\_1

nodes: 7, edges: 4

# Plan

1. Présentation de la problématique au travers d'exemples
2. Formalisation
3. Extraits de l'algorithme
4. Application aux données réelles
5. **Conclusion et perspectives**

# Conclusion

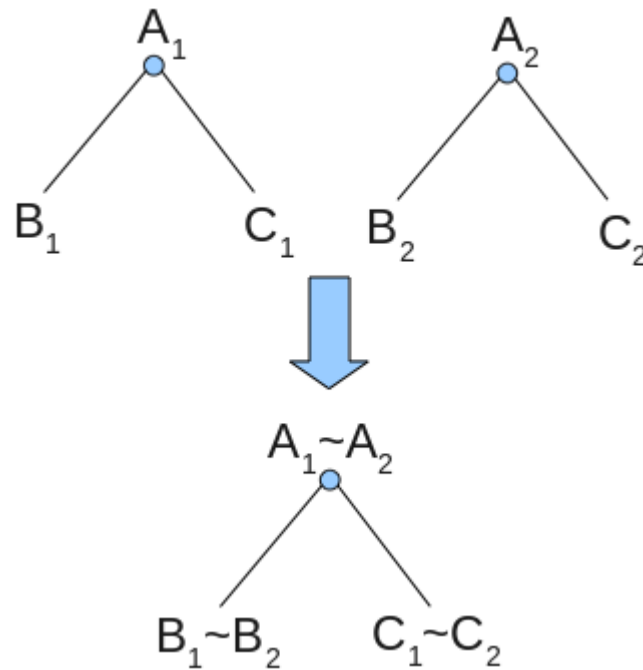
- Bilan
  - Appropriation du sujet, bibliographie
  - Formalisation et propriétés
  - Algorithme DéCo sur papier
  - Test sur des données construites
  - Code
  - Preuves de certaines propriétés

# Perspectives

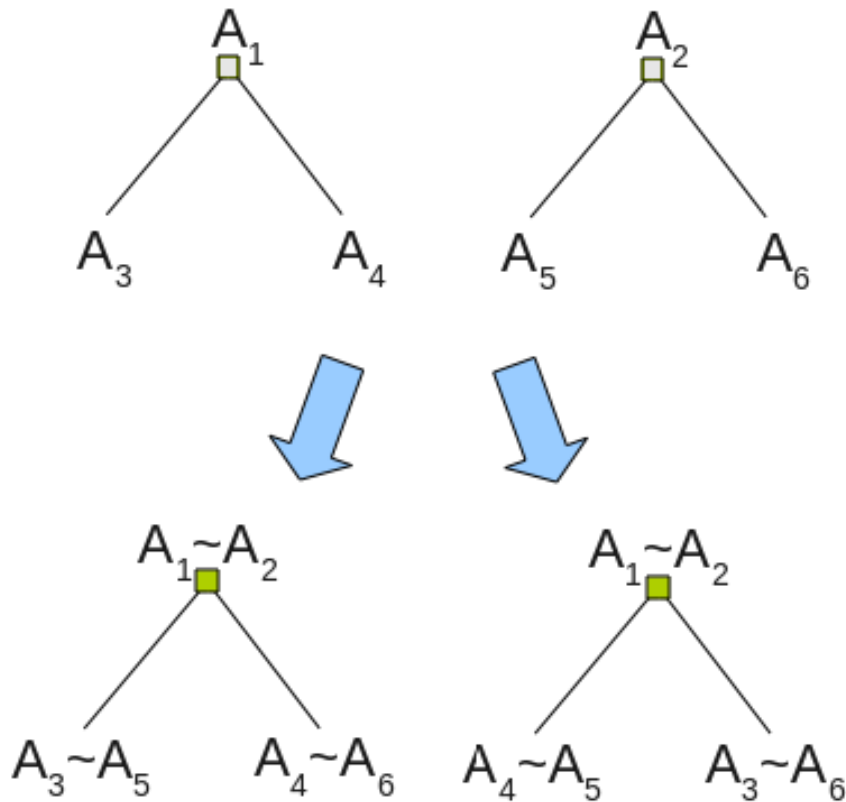
- Prendre en compte plus de 2 arbres de gènes
- Discriminer parmi les solutions de coût optimal celles qui sont biologiquement plus réalistes



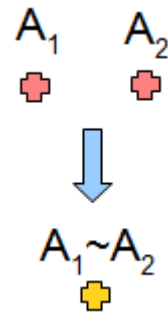
# Nœud de Spéciation



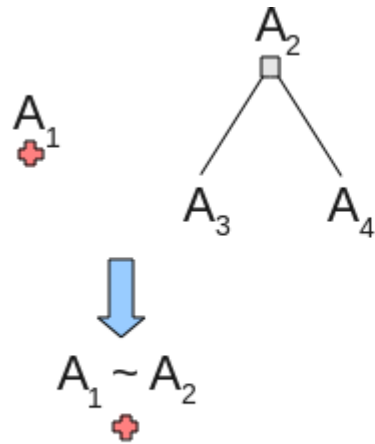
# Nœud de Duplication d'Adjacence



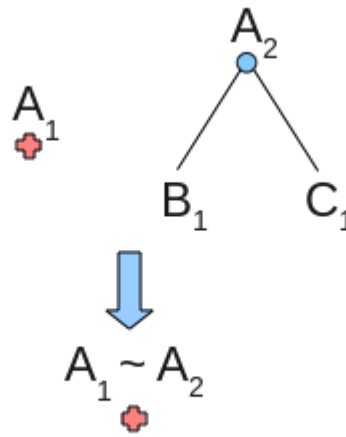
# Perte d'Adjacence



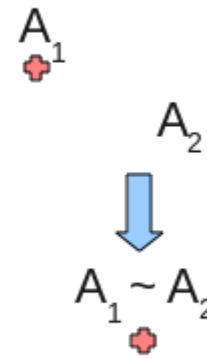
# Perte de Gène



Perte/Duplication

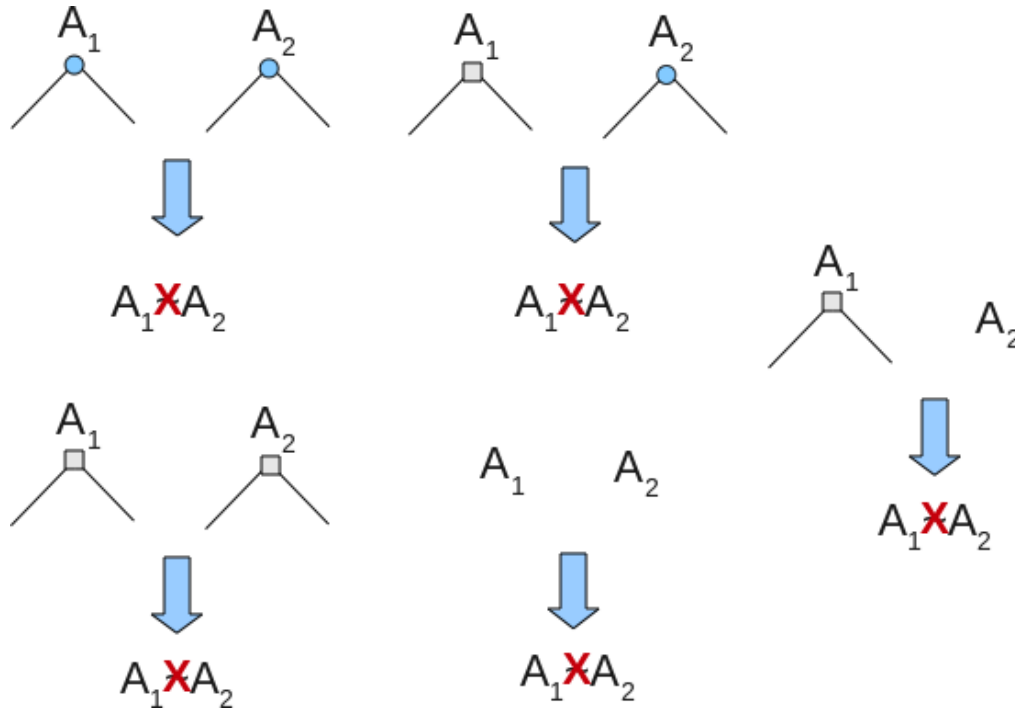


Perte/Spéciation



Perte/Gène Actuel

# Cassure



# Prétraitement

- Parser le fichier de données
- Réconcilier les arbres de gènes avec l'arbre des espèces
- Calculer le coût maximum