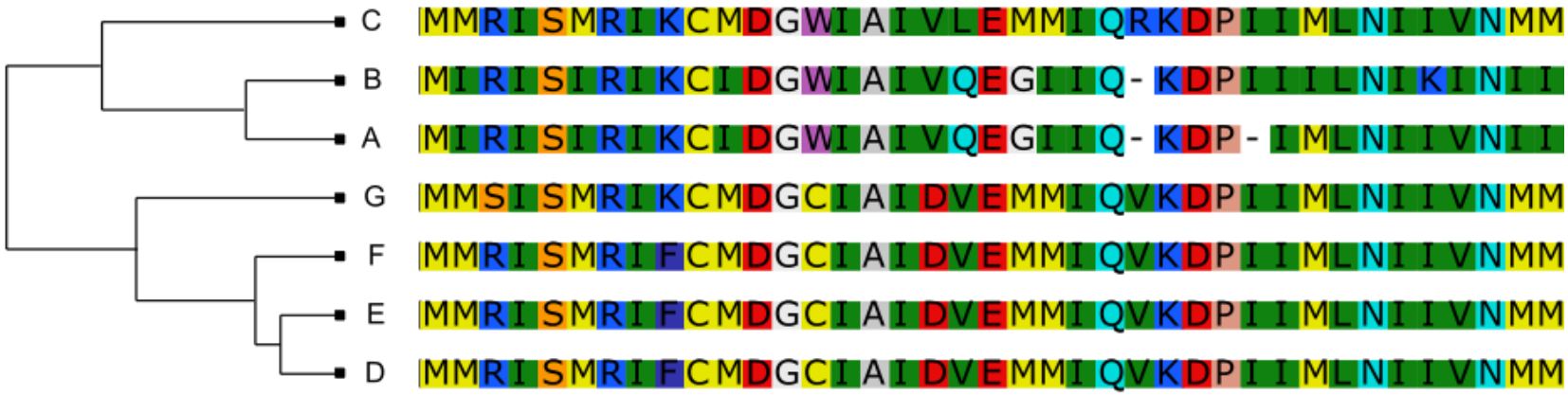
Efficient Non-binary Gene Tree Resolution with Weighted Reconciliation Cost

DIRO
Université de Montréal

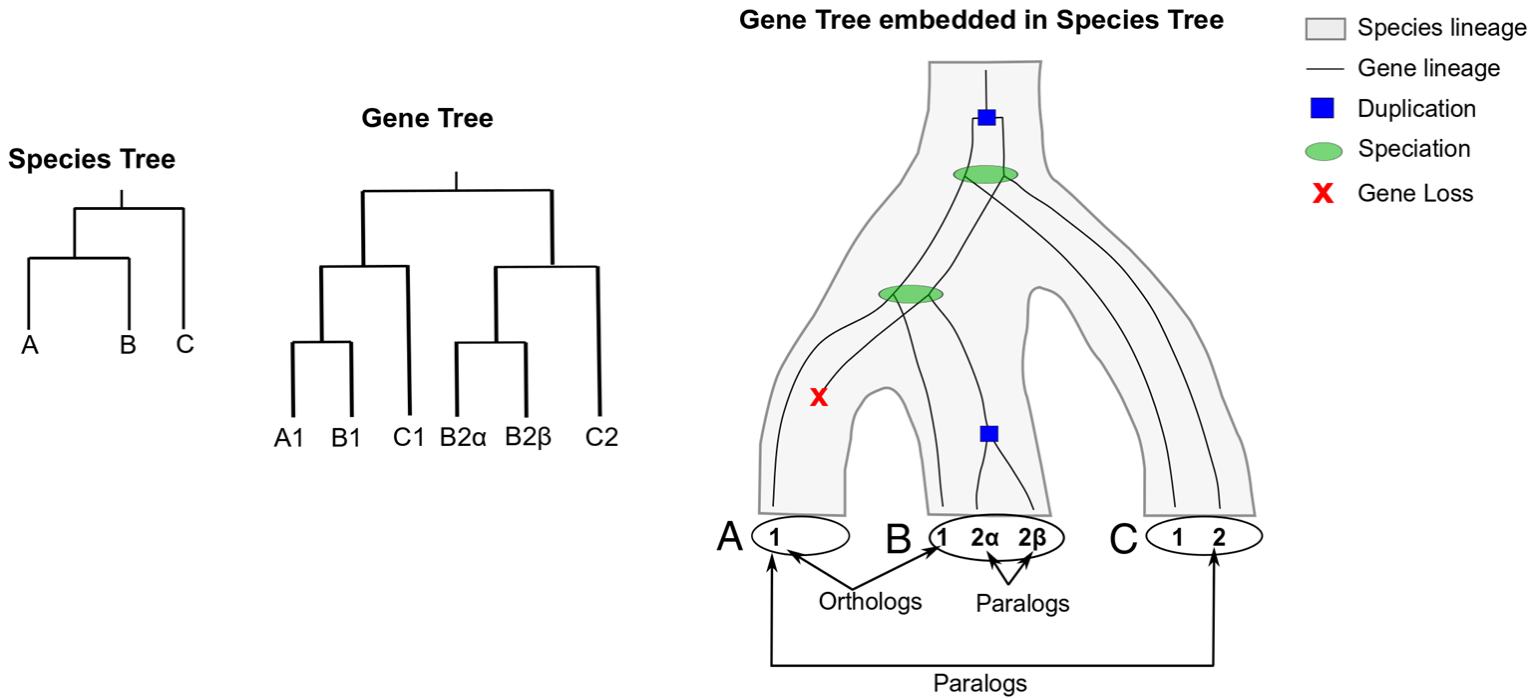
Manuel Lafond,
Emmanuel Noutahi
Nadia El-Mabrouk

Introduction

Gene Tree : representation of the evolutionary history of a family of **homologous** genes

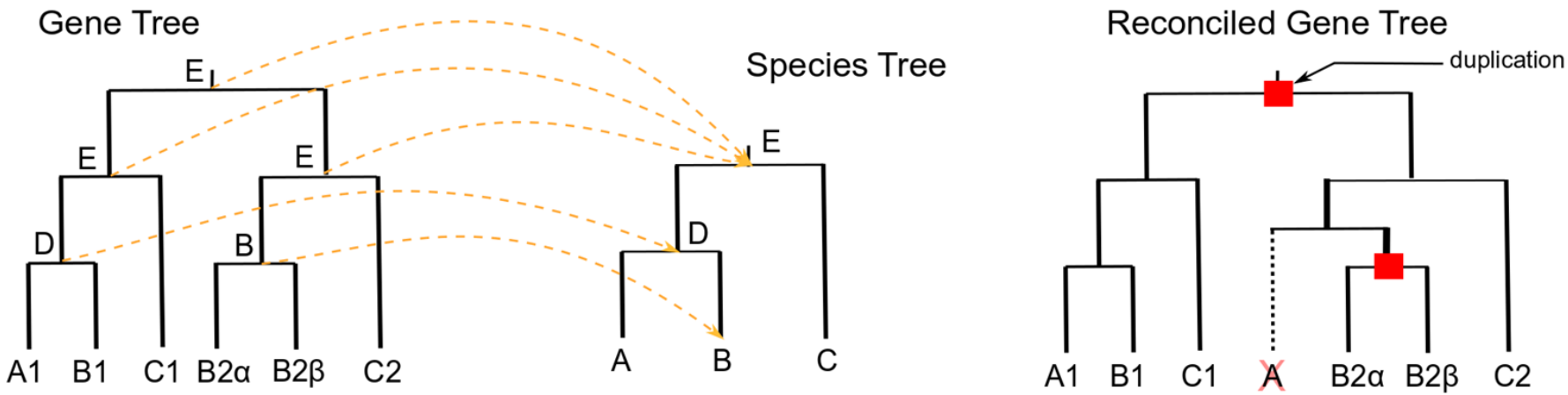


Gene Family history inference with reconciliation



- LCA == speciation node ⇒ orthologous gene
- LCA == duplication node ⇒ paralogous gene

- **Input:** rooted gene tree G, L(G) the set of gene and a species tree S
- **Reconciliation R(G,S)** : extension of the gene tree reflecting a history of evolutionary events (speciations, duplications, losses) in agreement with the species tree
- Optimize a criterion (duplication+loss cost)

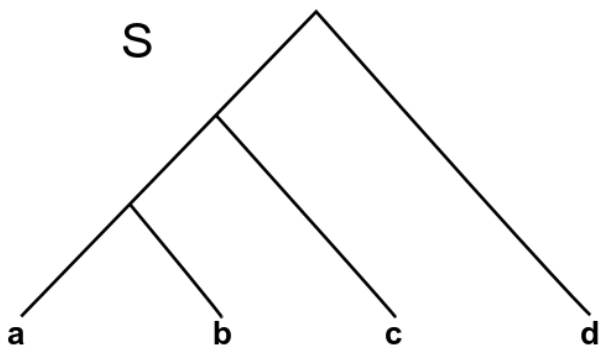


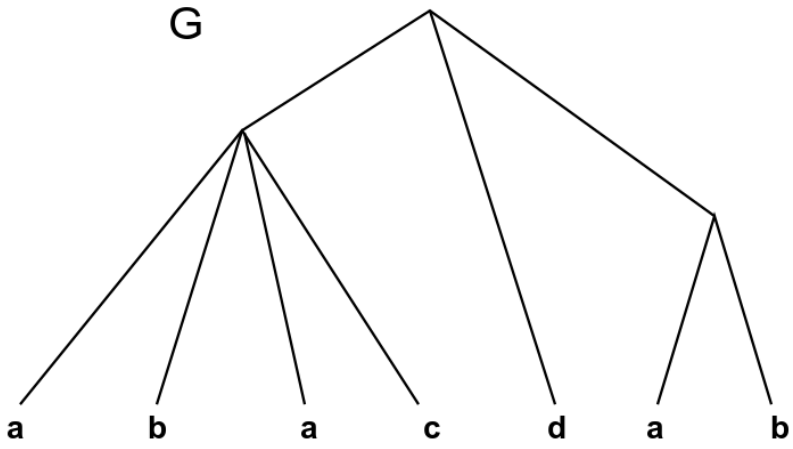
Reconciliation by LCA-Mapping

(Górecki & Tiuryn, 2006), (Chauve & El-Mabrouk, 2009)

Motivation

- Most reconciliation software works with binaries trees
  - Reconciled trees are binary
- Non binary trees in case of uncertainty
  - Methodological reasons
  - Lack of resolution between sequences (edge with weak support)



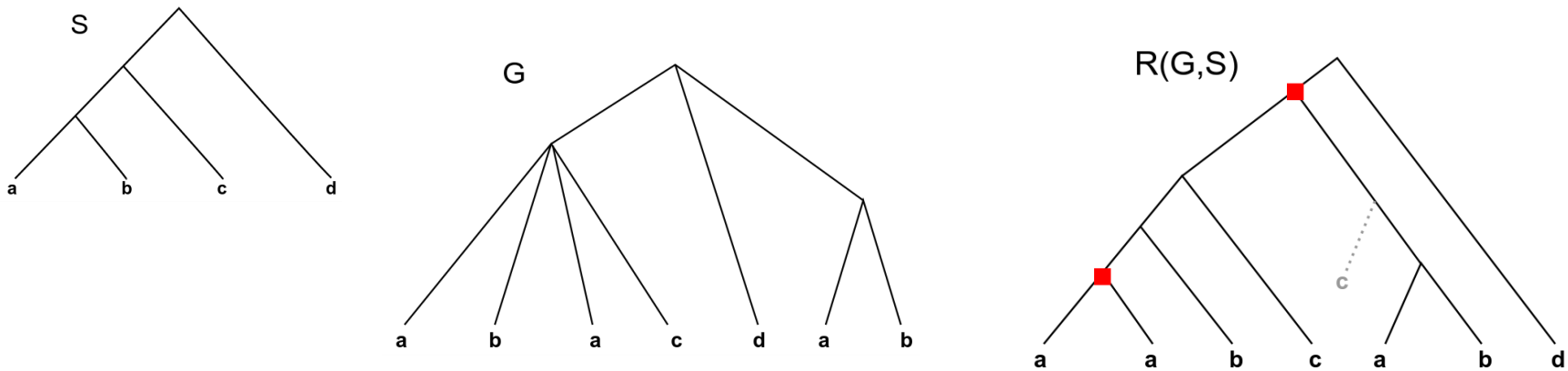


Problem statement

Minimum Resolution Problem (MRP)

Given : A binary species tree S and a non-binary gene tree G

Find: A binary resolution of G with minimum reconciliation cost (duplications + losses) with respect to S.



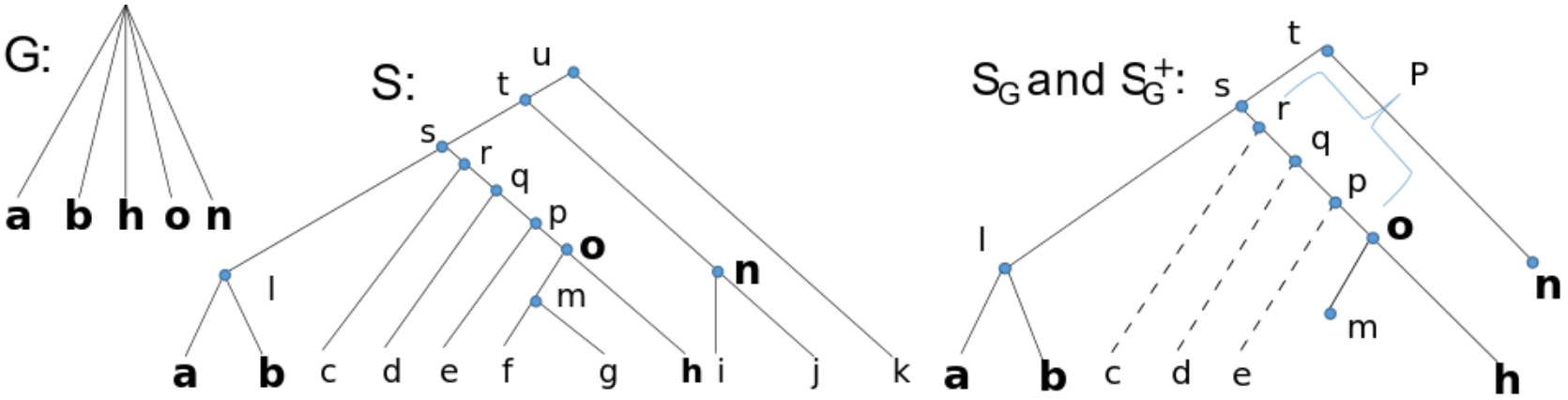
Previous work

- Chang & Eulenstein, 2006
  - **Polytomies can be resolved independently**
  - Cubic time **per polytomy**
- Durand et al., 2006
  - **General cost for duplication and loss**
  - Cubic time **per polytomy**
- Lafond et al., 2012
  - Quadratic time **per polytomy** (linear for unit cost)
- Zheng and Zhang, 2014
  - **Compressed species tree**
  - Linear time for **genetree (with the unit cost)**

A new dynamic programming approach for the MRP problem

- Introduction of species-specific cost
- Best known complexity
- Output all optimal solutions

Ignoring part of the species tree

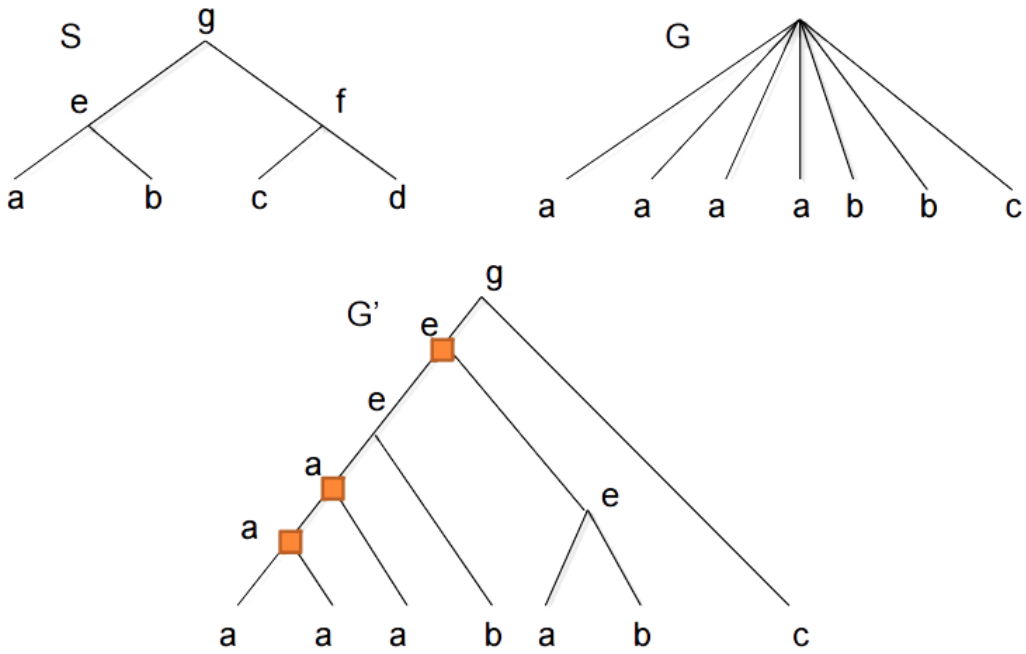


Reconciling a binary resolution of G with S or S^+_G will yield the same cost

(Lafond and al., 2012), (Zheng and Zhang, 2014)

Idea behind PolytomySolver

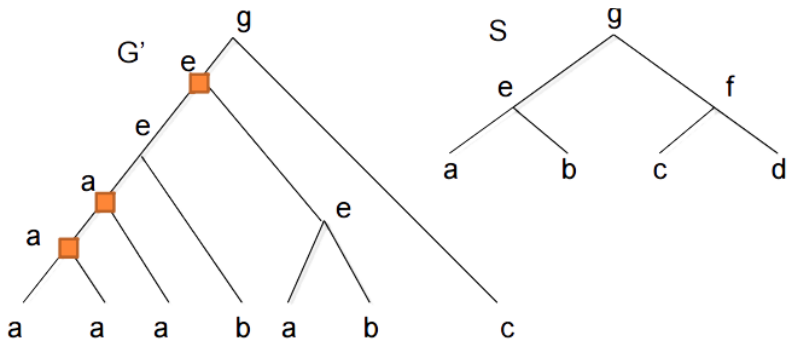
A minimal resolution contains **k** partial resolution rooted at each node of $S^+(G)$



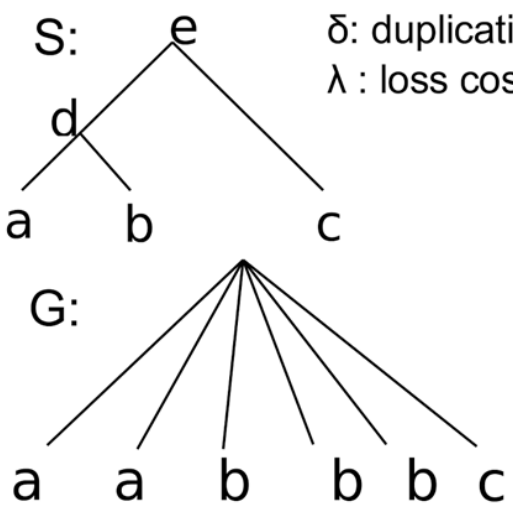
Idea behind PolytomySolver

M	1	2	3
a			
b			
c			
d			
e			
f			
g			

- For a species ***s***, M(s,k) represent the cost of having k-partial resolution rooted at ***s***
- M(root(S^+(G)), 1) is the cost of the resolution



Filling M(s,k) : leaf case



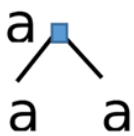
δ: duplication cost
λ : loss cost

M	1	2	3	4
a		0		
b				
c				
d				
e				

$M_{a,2} = 0$

a a

$M_{a,1} = \delta_a$



$M_{a,3} = \lambda_a$

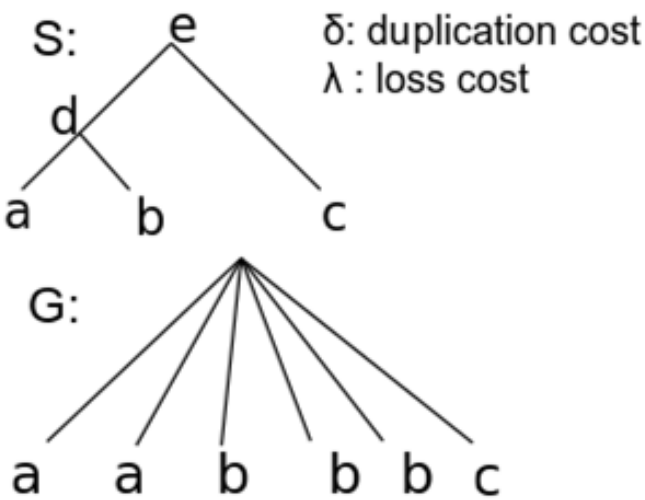
a a a

1. If k = m(s) ⇒ no cost
2. If k > m(s) ⇒ add new nodes (losses)
3. If k < m(s) ⇒ join nodes (duplication)

m(s) : multiplicity of s in G

Ex : m(a) = 2

Filling M(s,k) : leaf case



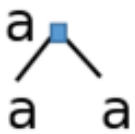
δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b				
c				
d				
e				

$M_{a,2} = 0$

a a

$M_{a,1} = \delta_a$



$M_{a,3} = \lambda_a$

a a a

1. If k = m(s) ⇒ no cost
2. If k > m(s) ⇒ add new nodes (losses)
3. If k < m(s) ⇒ join nodes (duplication)

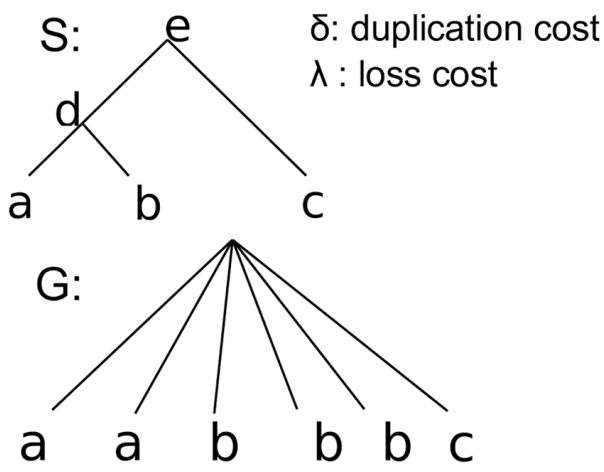
m(s) : multiplicity of s in G

Ex : m(a) = 2

Recurrence for a leaf

Lemma : For a leaf node *s*,

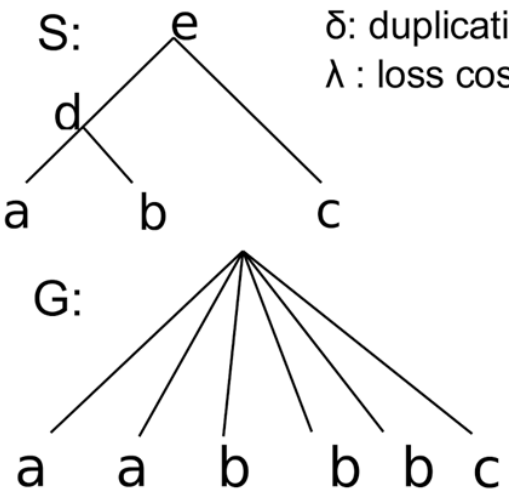
$$\begin{cases} M(s,k) = \lambda_s\,(k - m(s)) \text{ if } k > m(s) \\ M(s,k) = \delta_s\,(m(s) - k) \text{ else} \end{cases}$$



δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d				
e				

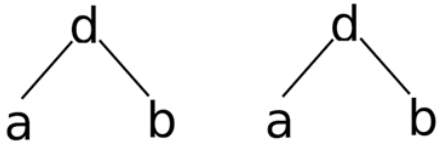
Filling M(s,k) : internal node case



δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d		1		
e				

$M_{d,2} = C_{d,2} = M_{a,2} + M_{b,2}$

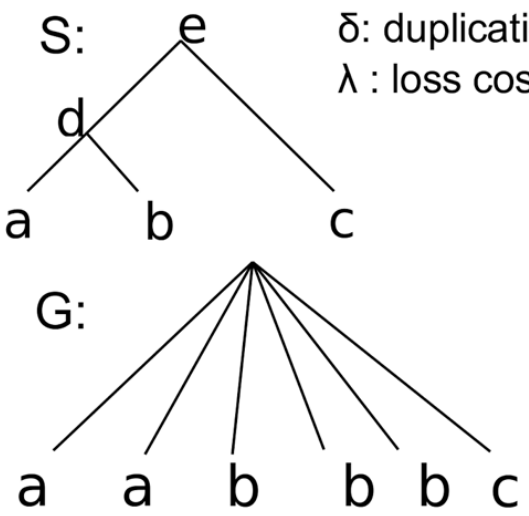


speciation
join a and b, until we have enough d

Speciation-only nodes

$$C_{s,k} = \begin{cases} M_{s_l,k-m(s)} + M_{s_r,k-m(s)} & \text{if } k > m(s) \\ +\infty & \text{otherwise} \end{cases}$$

Filling M(s,k) : internal node case



δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	X	1		
e				

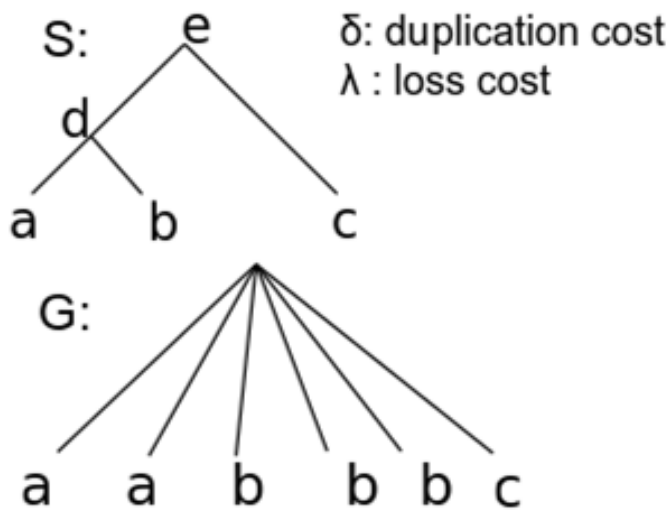
$M_{d,2} = M_{d,1} + \lambda_d$

d d

loss
add one d

Add a new node (loss) ⇒ $M(s, k) = M(s, k-1) + \lambda(s)$

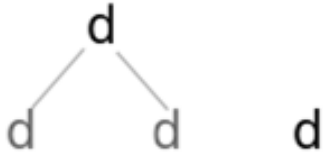
Filling M(s,k) : internal node case



δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d		1 ←	X	
e				

$M_{d,2} = M_{d,3} + \delta_d$



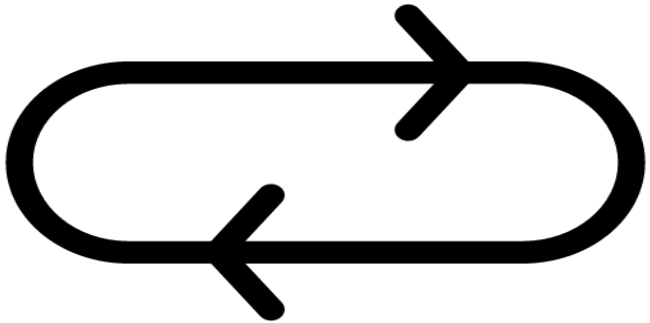
duplication
join two d

Join 2 nodes (duplication) ⇒ $M(s, k) = M(s, k+1) + \delta(s)$

Recurrence for internal node

Lemma : For an internal node *s* of S

$$M(s,k) = \min \left\{ M(s, k+1) + \delta_s,\ M(s, k-1) + \lambda_s,\ C(s,k) \right\}$$



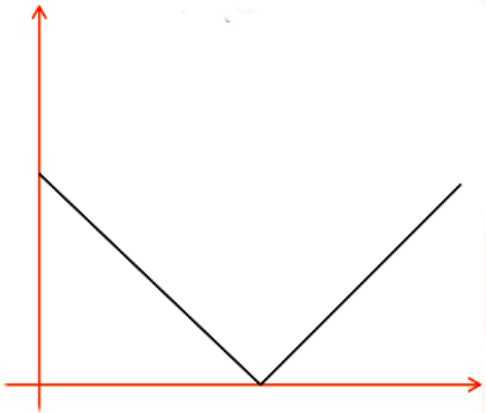
Convexity of M(s) and C(s)

Lemma : Both M(s) and C(s) are convex.

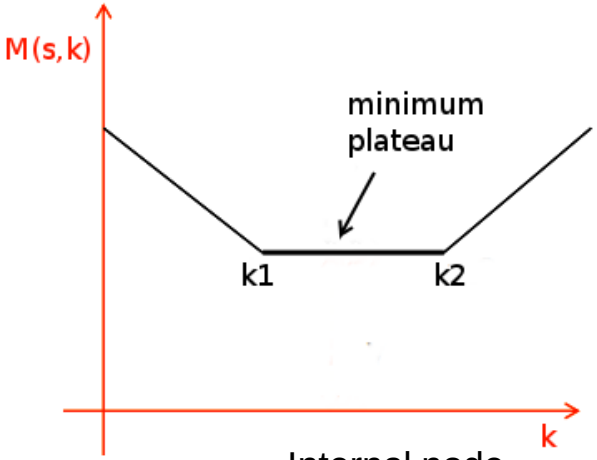
A function is convex iff :

$$\epsilon_1, \epsilon_2 > 0 \quad n > \epsilon_1, \ \mathrm{f(n - \epsilon_1)} + f(n + \epsilon_2) - 2f(n) \geq 0$$

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d				
e				



Leaf node



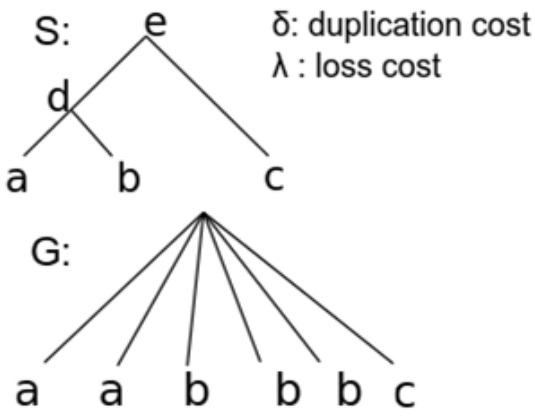
Internal node

$$C(s, k_1) = C(s, k_2) = \min_k C(s, k)$$

New recurrence

Theorem : Let k_1 and k_2 be the smallest and largest values, respectively, such that $C(s,k_1) = C(s,k_2) = \min_k C(s,k)$. Then,

$$M_{s,k} = \begin{cases} C_{s,k} & \text{if } k_1 \leq k \leq k_2 \\ \min(C_{s,k}, M_{s,k+1} + \delta_s) & \text{if } k < k_1 \\ \min(C_{s,k}, M_{s,k-1} + \lambda_s) & \text{if } k > k_2 \end{cases}$$

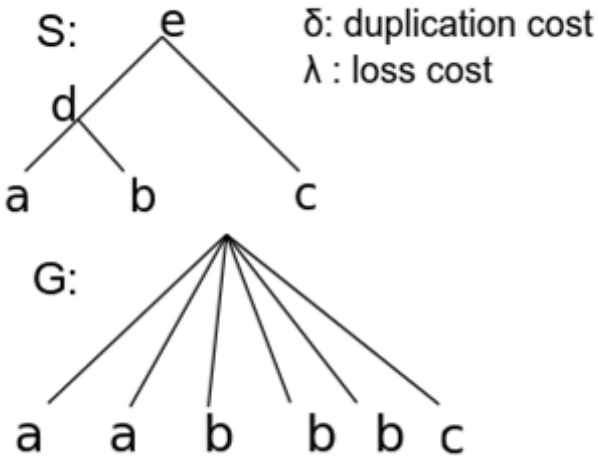


M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	3	1	1	3
e				

C(d, k)

New recurrence

$$M_{s,k} = \begin{cases} C_{s,k} & \text{if } k_1 \leq k \leq k_2 \\ \min(C_{s,k}, M_{s,k+1} + \delta_s) & \text{if } k < k_1 \\ \min(C_{s,k}, M_{s,k-1} + \lambda_s) & \text{if } k > k_2 \end{cases}$$



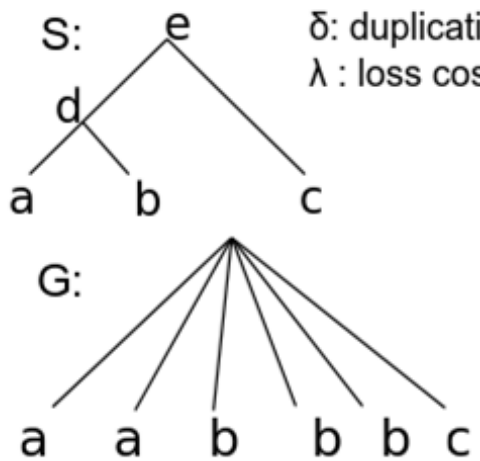
δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	3 2	1	1	3 2
e				

M(d, k)

New recurrence

$$M_{s,k} = \begin{cases} C_{s,k} & \text{if } k_1 \leq k \leq k_2 \\ \min(C_{s,k}, M_{s,k+1} + \delta_s) & \text{if } k < k_1 \\ \min(C_{s,k}, M_{s,k-1} + \lambda_s) & \text{if } k > k_2 \end{cases}$$

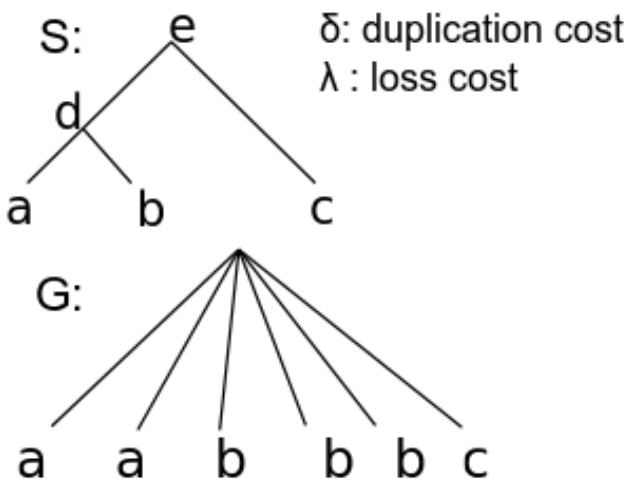


δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	2	1	1	2
e	2	2	3	4

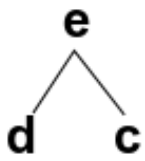
M(e, k)

Backtrack to build the resolutions



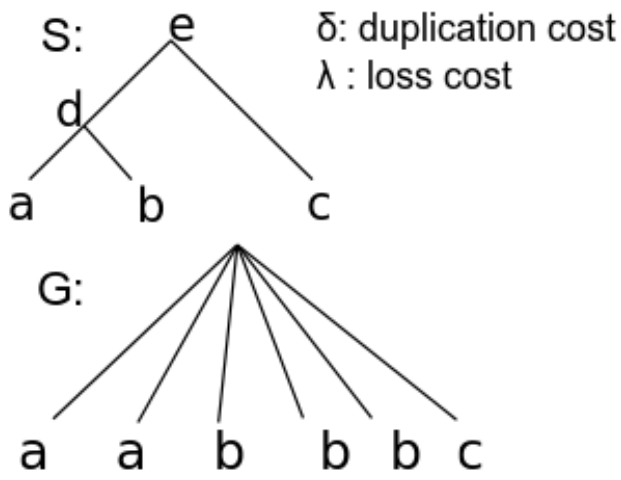
δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	2	1	1	2
e	2	2	3	4



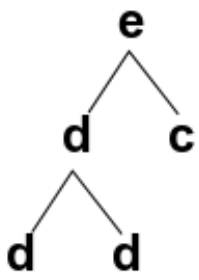
- From the root [M(e,1)] to the leaves
- 1 **e** obtained from joining (**c**,**d**)
- Keep pointers

Backtrack to build resolution



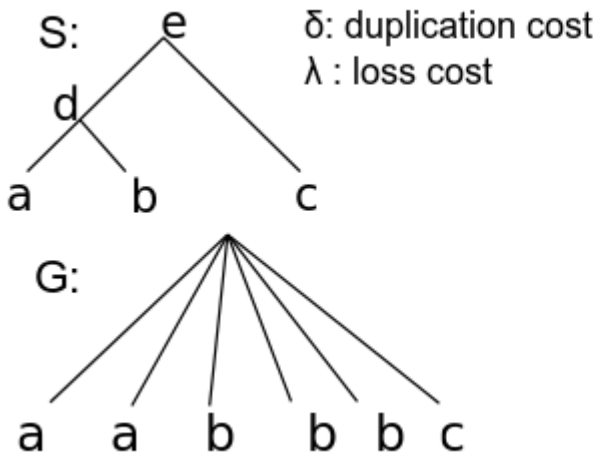
δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	2	1	1	2
e	2	2	3	4



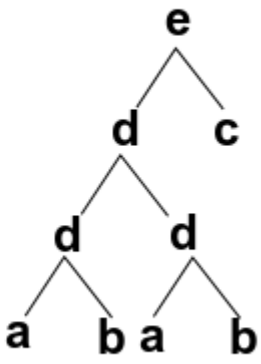
- One duplication in **d**
- Keep pointers

Backtrack to build resolution



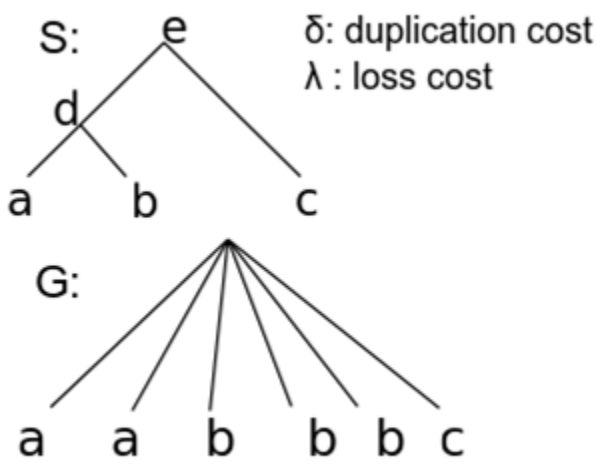
δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	2	1	1	2
e	2	2	3	4



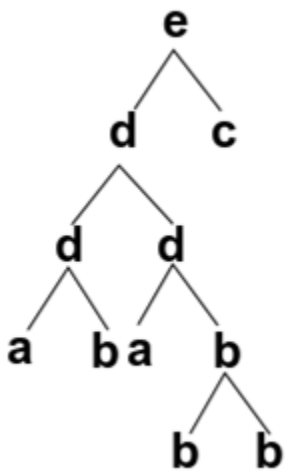
- 2 **d** obtained from joining (a,b), twice
- Keep pointers

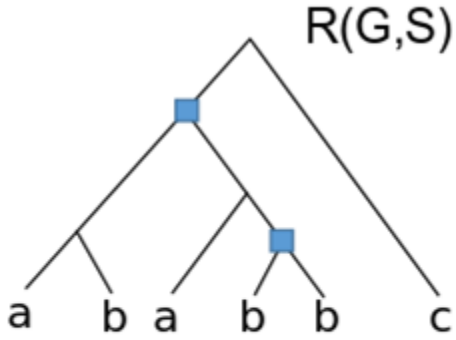
Backtrack to build resolution



δ: duplication cost
λ : loss cost

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	2	1	1	2
e	2	2	3	4





- duplication of b to get 2 copies from 3
- Keep pointers

Complexity analysis

Theorem : Only the values of M and C for columns k between 1 and |G| – 1 need to be computed

- Everybody thought that $\max(k) = \max_{s \in V(s)} m(s)$
- Table construction in **O(|G||S|)** for a polytomy G
- **O(p|S|Δ)** for a gene tree with *p* polytomies where **Δ** is the max degree of the nodes.

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	2	1	1	2
e	2	2	3	4

Complexity analysis

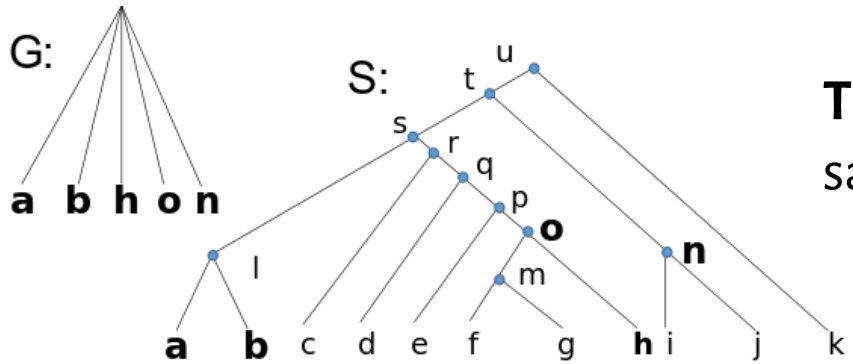
Theorem : Only the values of M and C for columns k between 1 and |G| – 1 need to be computed

- Everybody thought that $\max(k) = \max_{s \in V(s)} m(s)$
- Table construction in **O(|G||S|)** for a polytomy G
- **O(p|S|Δ)** for a gene tree with *p* polytomies where **Δ** is the max degree of the nodes.

Can we do better ???

M	1	2	3	4
a	1	0	1	2
b	2	1	0	1
c	0	1	2	3
d	2	1	1	2
e	2	2	3	4

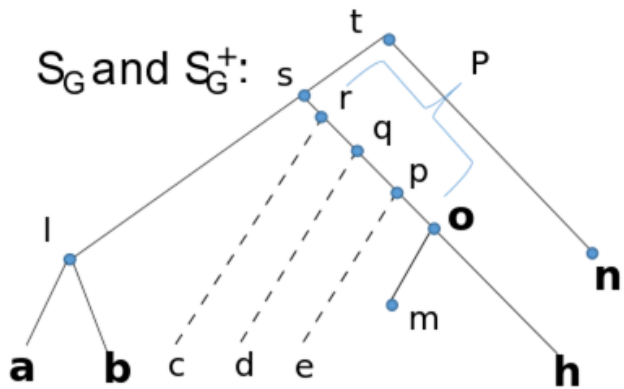
Faster algorithm for general cost (δ != λ) with species tree compression

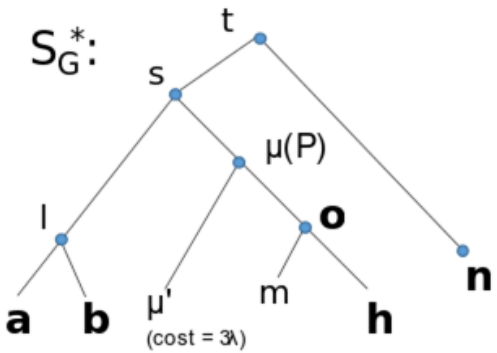


Theorem : A binary refinement of G have the same reconciliation cost with $S^{+}(G)$ or $S^{*}(G)$

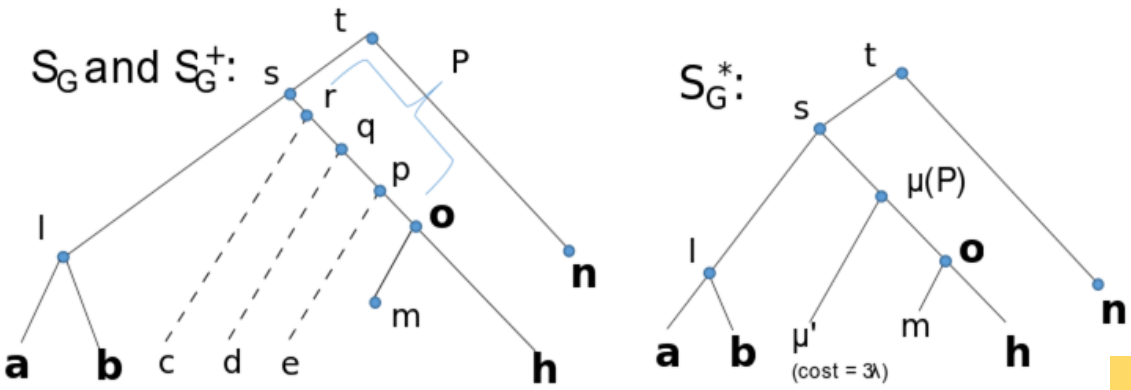
Idea from Zheng and Zhang

- **limited to unit cost**
- **only one solution**





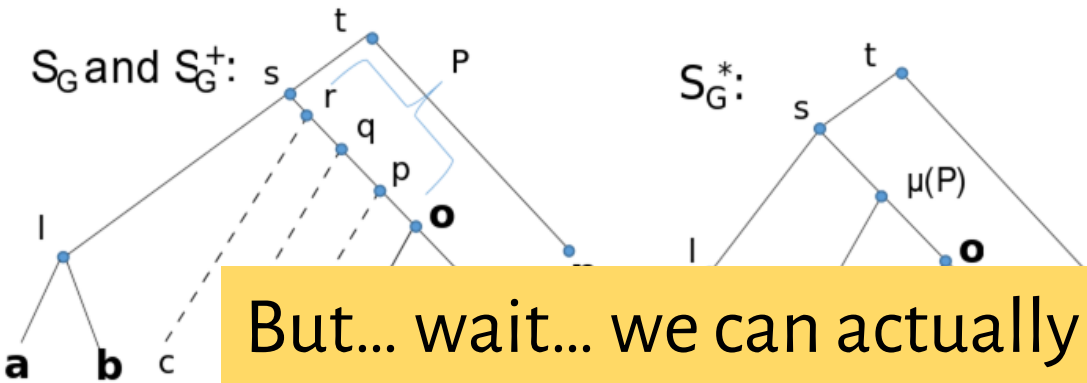
Faster algorithm for general cost (δ != λ) with species tree compression



We can construct S*(G) in O(|G|)

- $O(|S^*||G| + |G|) = O(|G|^2)$ for a polytomy G
- $\sum_{h \in V(H)} c \cdot \deg(h)^2 \leq c \cdot \Delta \sum_{h \in V(H)} \deg(h) \in O(\Delta |H|)$ for a gene tree H

Faster algorithm for general cost (δ != λ) with species tree compression

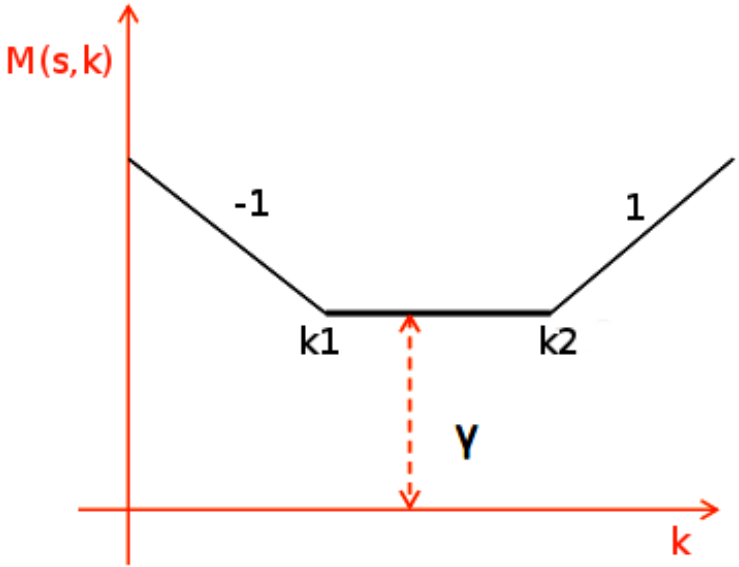


But... wait... we can actually do **BETTER** under the **unit cost** model ($\delta = \lambda$) !!

- $O(|S^*||G| + |G|) = O(|G|^2)$ for a polytomy G
- $\sum_{h \in V(H)} c \cdot \deg(h)^2 \leq c \cdot \Delta \sum_{h \in V(H)} \deg(h) \in O(\Delta |H|)$ for a gene tree H

Unit cost model ($\delta = \lambda$)

- Only 3 values needed to represent M(s) : k1, k2 and $\gamma \Rightarrow$ **O(1)** per row
- Still need to account for leaf with different cost
- **O(H)** per gene tree
- **Can output all solutions**

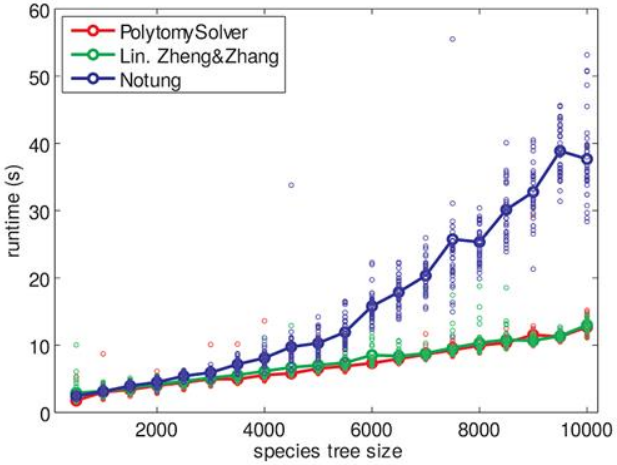


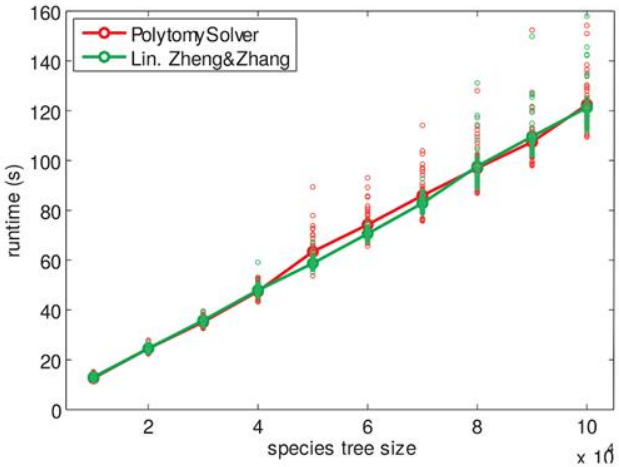
Complexity comparison

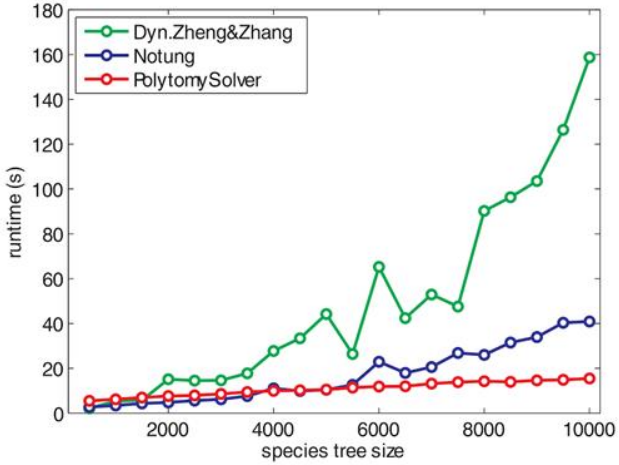
	$\delta = \lambda = 1$	$(\delta, \lambda) \in \mathcal{R}_{>0} \times \mathcal{R}_{>0}$	$\{(\delta_s, \lambda_s)\}_{s \in V(S)}$
NOTUNG	$O(\lvert S\rvert\lvert G\rvert\Delta^2)$	$O(\lvert S\rvert\lvert G\rvert\Delta^2)$	
Lafond	$O(\lvert S\rvert\lvert G\rvert)$	$O(\lvert S\rvert\lvert G\rvert\Delta)$	
Zheng & Zhang	$O(\lvert G\rvert)$	$O(\lvert G\rvert\Delta^2)$	
PolytomySolver	$O(\lvert G\rvert)$	$O(\lvert G\rvert\Delta)$	$O(\lvert G\rvert\lvert S\rvert\Delta)$

Best theoretical complexity to date, but in practice ???

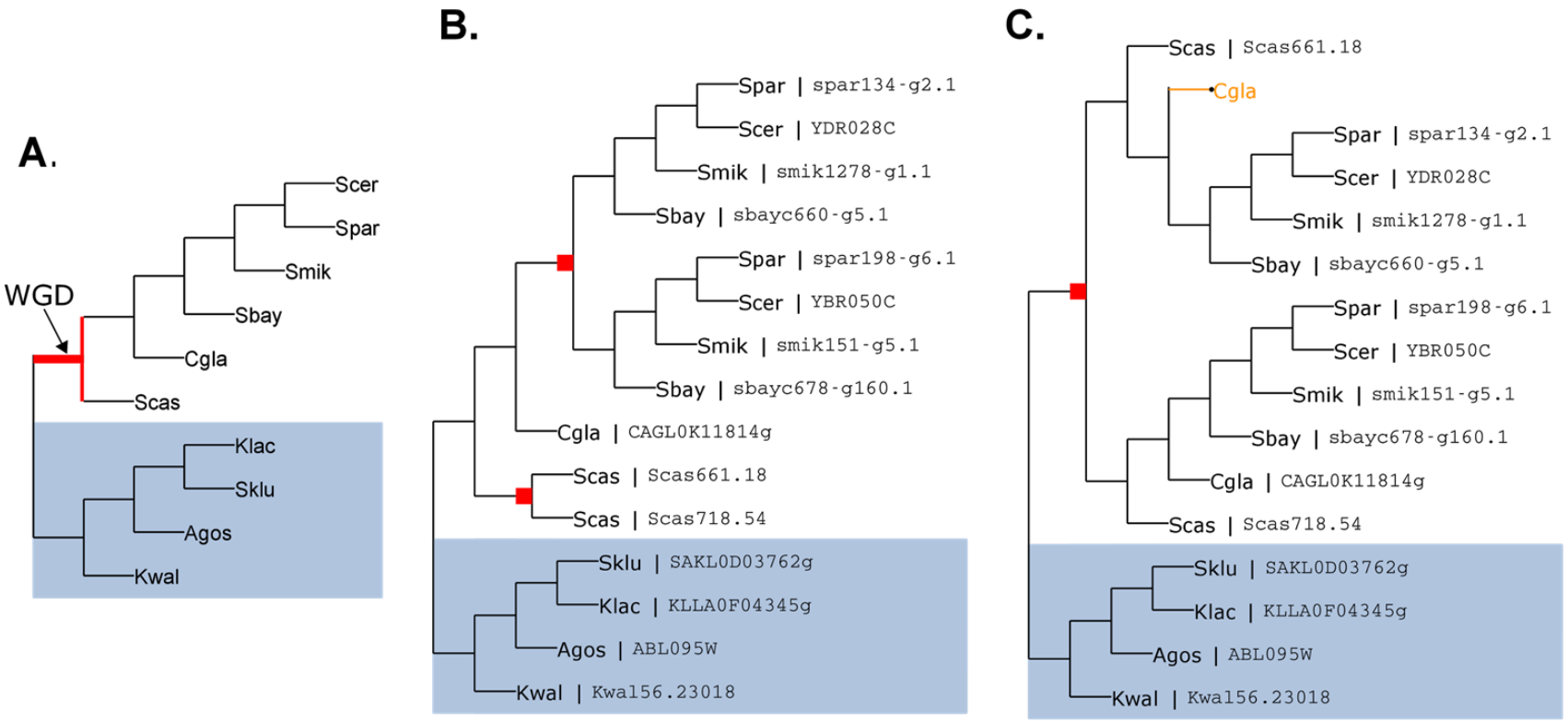
Running time on simulated dataset







- Advantage become more evident on large dataset
- Large gene family (ex: olfactory receptor (OR) gene superfamily)



- Missing data (lower loss cost)
- Availability of biological evidence (rate of gene duplication/loss)

Species-specific has actually some practical use

Conclusion

- Species-specific cost has practical advantage
- Fast algorithm for the resolution of polytomy
  - Can output all optimal solutions
  - LGT not included
- Base for the development of new algorithms for gene trees correction
  - ProfileNJ (NJ at joining step)

Merci!