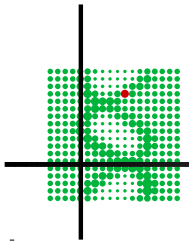


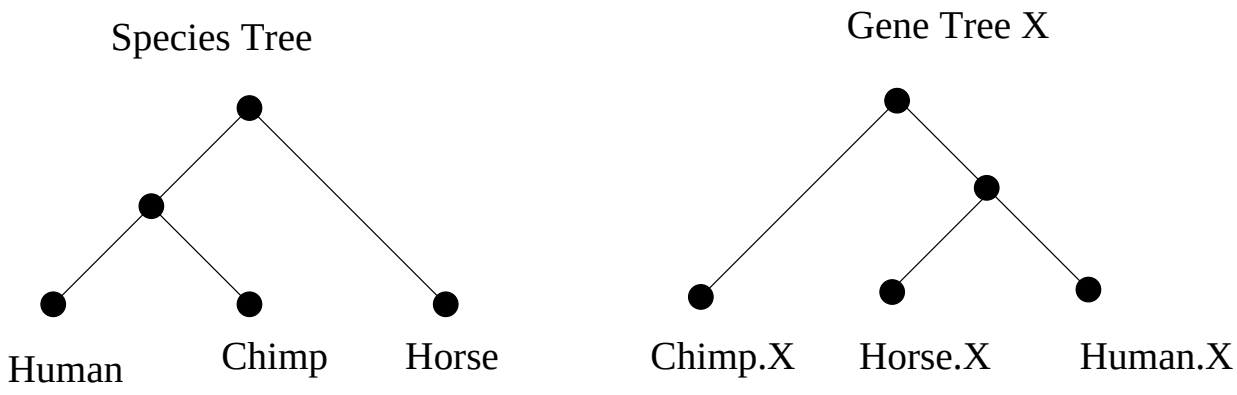
An Improved Algorithm for the Macro-evolutionary Phylogeny Problem

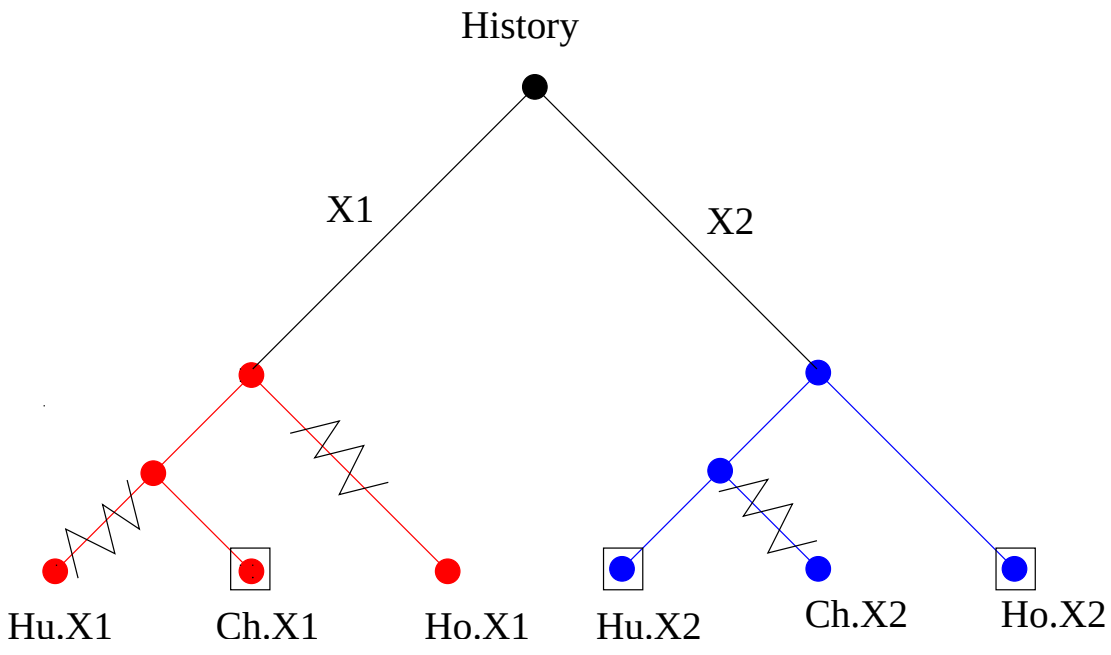
Behshad Behzadi and Martin Vingron

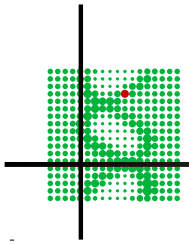
Max Planck Institute for Molecular Genetics

CPM 2006, Barcelona

Gene trees, species trees and gene duplications





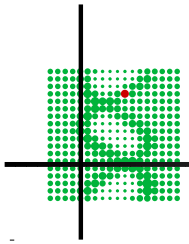


Gene trees, species trees and gene duplications

- Topology of gene trees and species trees are usually different.

- The evolutionary history of a gene family should be determined by:
  - **micro-evolutionary events** (sequence evolution)
  - **macro-evolutionary events** (gene duplication and loss)

- Tree Reconciliation Algorithm (Page 1994)



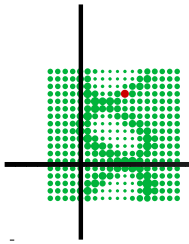
Durand et al. (RECOMB2005)

- A Hybrid Micro-Macroevolutionary Approach to Gene Tree Reconstruction

Two phase approach:
Phase 1: A gene tree based only on micro-evolutionary model is constructed.
Phase 2: Refining the tree w.r.t. a macroevolutionary model.

- regions with strong sequence support are left intact.
- other regions are rearranged w.r.t to the **D/L score**.

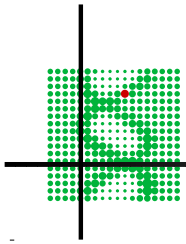
D/L score: $c_\lambda L + c_\delta D$, the weighted sum of the number of duplications, D, and the number of of losses, L, in the tree.



The Macro-Evolutionary Phylogeny Problem

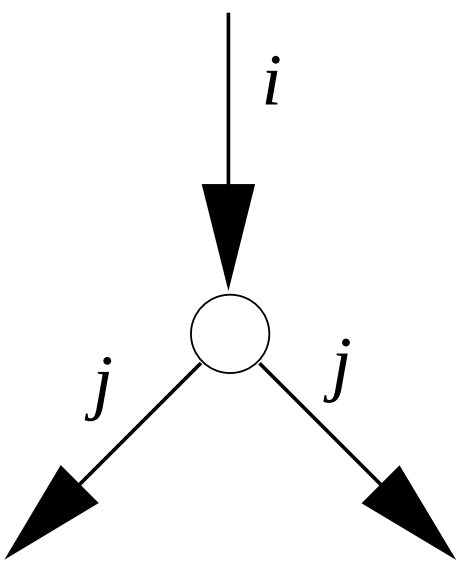
- **Input:** A rooted species tree, T_S with s leaves; a list of multiplicities $m_1, ..., m_s$, where m_l is the number of gene family members found in species l; weights c_λ and c_δ.

- **Output:** The set of all rooted gene trees $\{T_G\}$ with $\Sigma_{l=1}^{s} m_l$ leaves such that D/L Score of T_G is minimal.



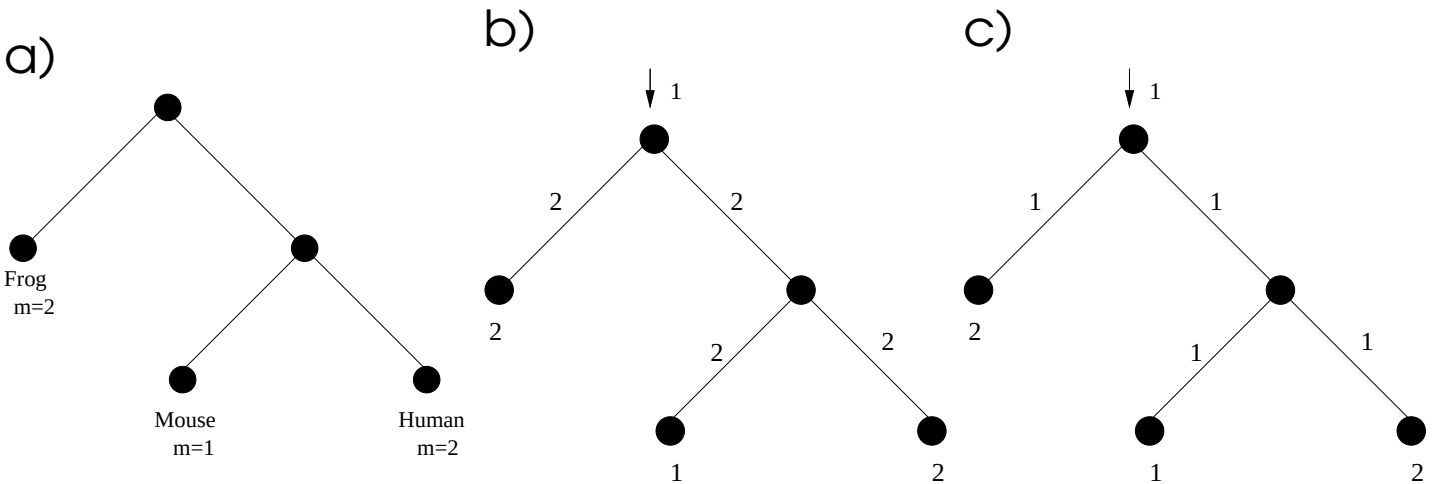
The Macro-Evolutionary Phylogeny Problem

- The output can be represented only by annotation of the species tree by the number of gene copies in different nodes.
- each duplication increases the number of gene copies by one.
- each loss decreases the number of the gene copies by one.
- the entering number of genes in root should be one.

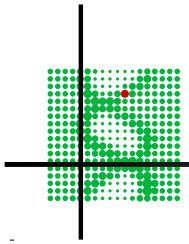


- if $i < j$ then $j - i$ duplications
- if $i = j$ then speciation
- if $i > j$ then $i - j$ losses

Minimal D/L score history



- a) species tree
- b) a history with one duplication and one loss
- c) a history with two duplications.



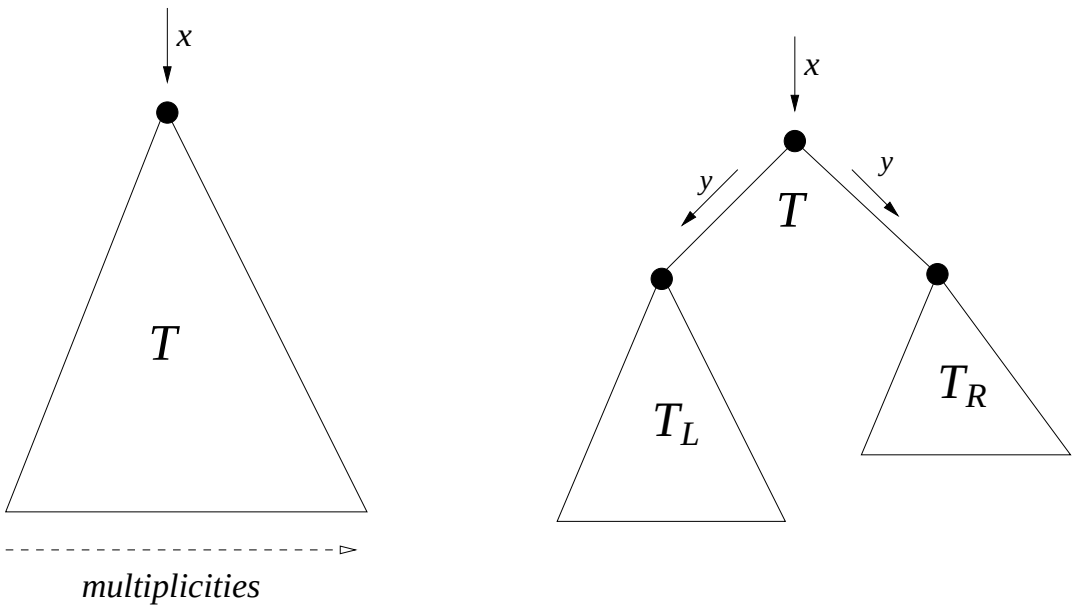
Past Work vs. Present Work

- Durand et al. (Recomb 2005)
  - Dynamic Programming for filling a table $COST[i, j, v]$
  - Finding the optimal solution using this table $COST$.

- This work
  - Dynamic Programming with a reduced dimension
  - Using combinatorial properties of optimal generation function $g(x, T)$.

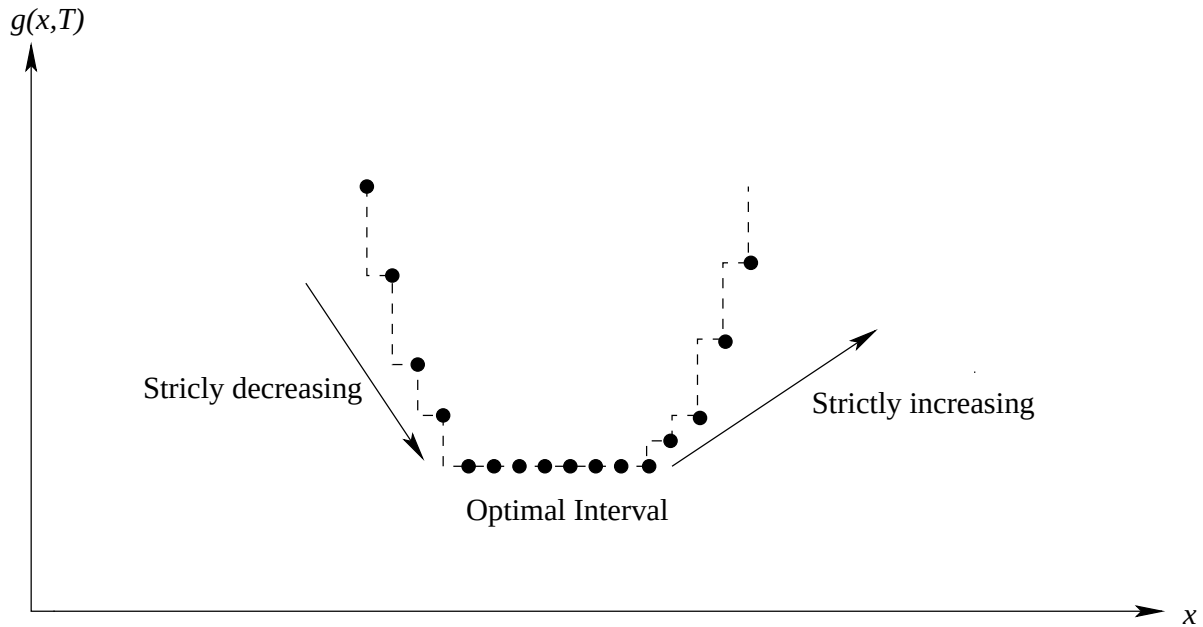
Definition of $g(x, \mathcal{T})$

- $g(x, \mathcal{T})$ the minimum D/L score of $\mathcal{T}$ where its root has x entering copies of genes.

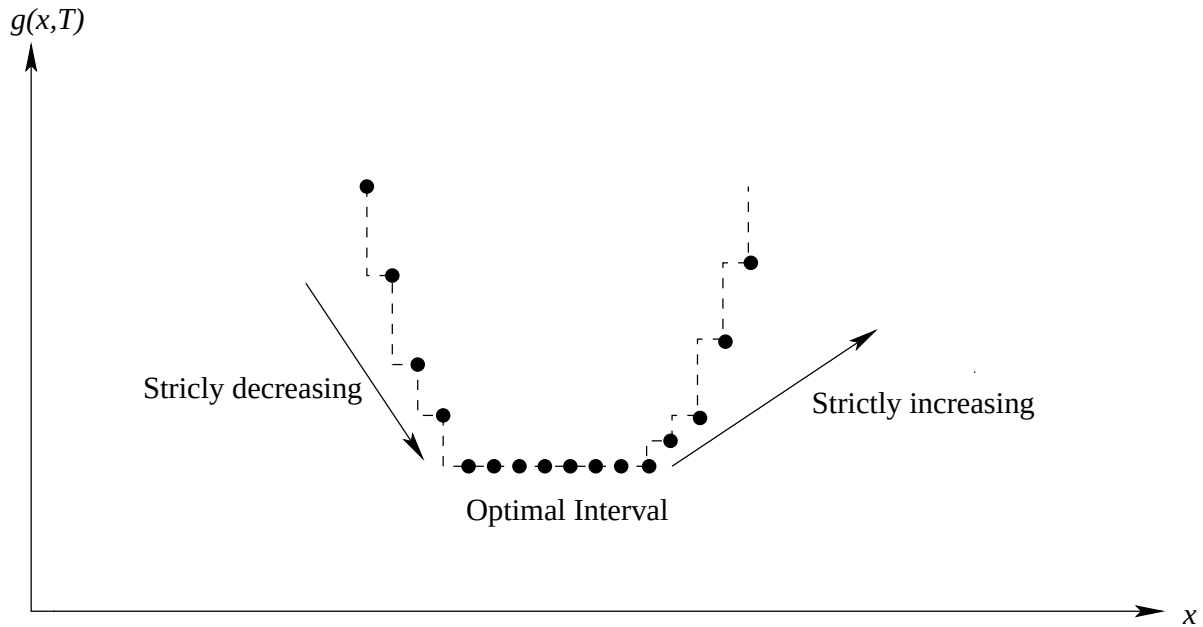


$$g(x, \mathcal{T}) = \min \begin{cases} g(x+1, \mathcal{T}) + c_\delta & \text{(Duplication)} \\ g(x-1, \mathcal{T}) + c_\lambda & \text{(Loss)} \\ g(x, \mathcal{T}_L) + g(x, \mathcal{T}_R) & \text{(Speciation)} \end{cases}$$

General Structure of $g(x, \mathcal{T})$



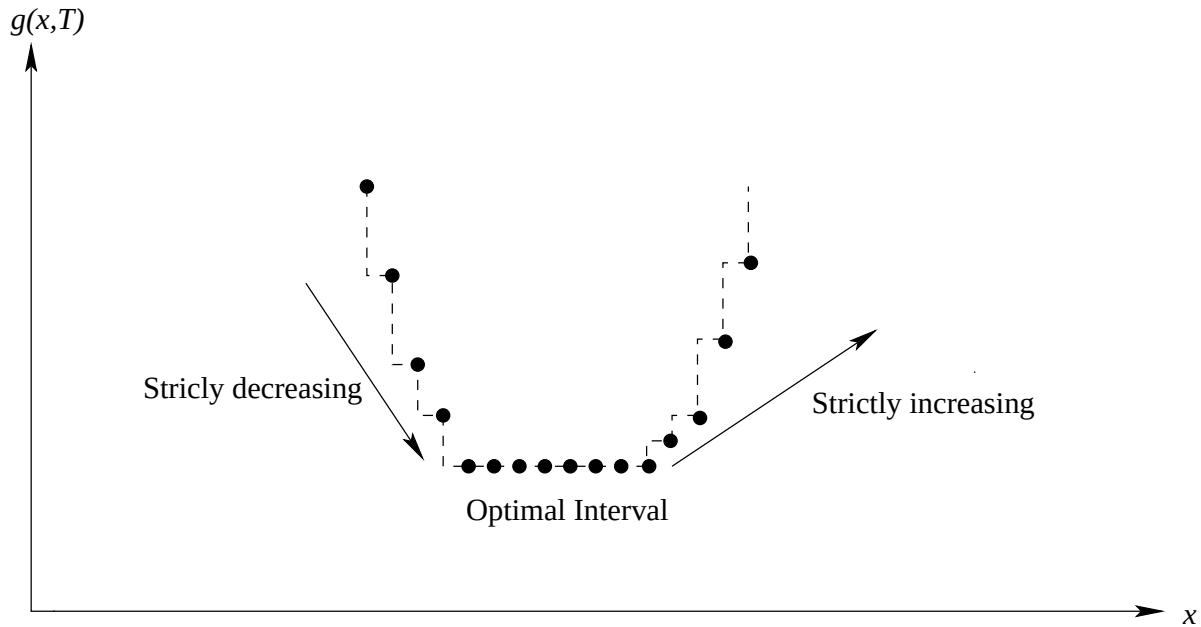
General Structure of $g(x, \mathcal{T})$



$g(x, T)$ in range $(-\infty, \infty)$:

- is firstly strictly decreasing,
- adapts its minimum on an interval,
- and then it is strictly increasing.

General Structure of $g(x, \mathcal{T})$

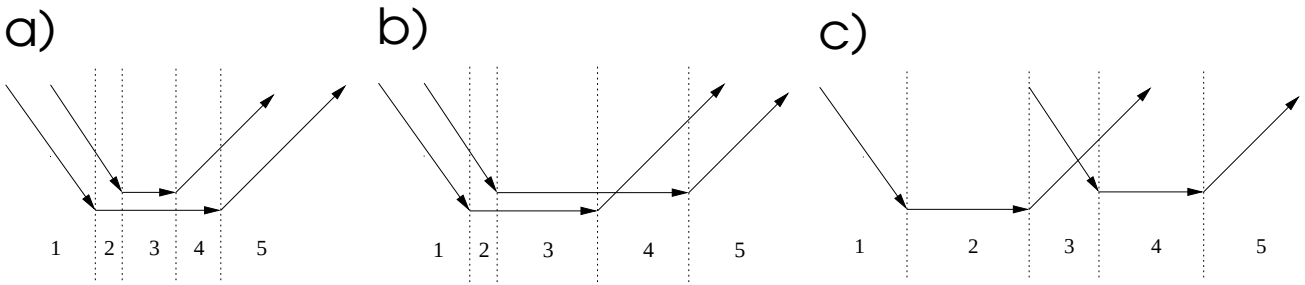


$g(x, T)$ in range $(-\infty, \infty)$:

- is firstly strictly decreasing,
- adapts its minimum on an interval,
- and then it is strictly increasing.

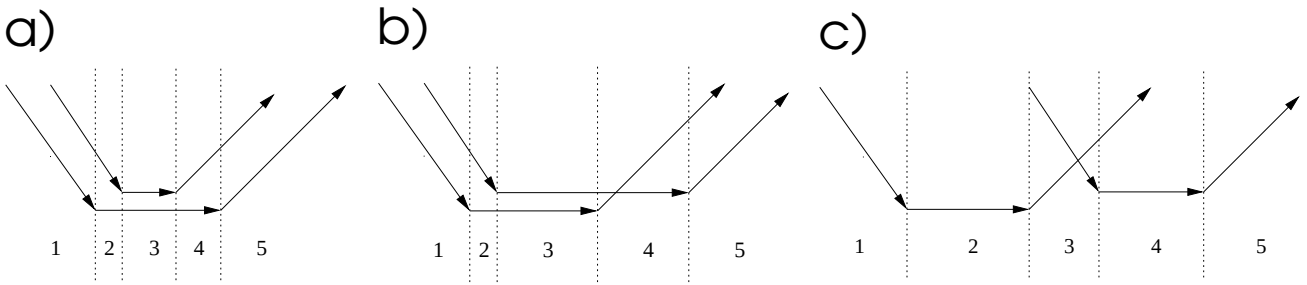
$g(x, T)$ is convex; $\Delta g(x, T) = g(x, T) - g(x-1, T)$ is increasing.
for very large x, $\Delta g(x, T) = c_\lambda$ and for very small x and $\Delta g(x, T) = -c_\delta$.

Possible Configurations of $g(x, \mathcal{T}_L)$ and $g(x, \mathcal{T}_R)$

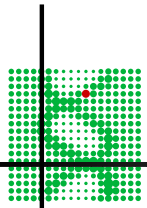


- a) Interval 3 is the optimal interval of $g(x, \mathcal{T})$.
- b) Interval 3 is the optimal interval of $g(x, \mathcal{T})$.
- c) The optimal interval of $g(x, \mathcal{T})$ is included in 3.

Possible Configurations of $g(x, \mathcal{T}_L)$ and $g(x, \mathcal{T}_R)$



- In interval (2,3,4), $g(x, \mathcal{T}) = g(x, \mathcal{T}_L) + g(x, \mathcal{T}_R)$.
- In interval (1), $g(x, \mathcal{T}) = \min\{g(x, \mathcal{T}_L) + g(x, \mathcal{T}_R), g(x+1, \mathcal{T}) + c_\delta\}$
- In interval (5), $g(x, \mathcal{T}) = \min\{g(x, \mathcal{T}_L) + g(x, \mathcal{T}_R), g(x-1, \mathcal{T}) + c_\lambda\}$



Algorithm

Gencost(Tree T)

1. **if** T *is a leaf* **then**

  1.1 **for** $i \leftarrow 1$ **to** m **do**

    1.1.1 **if** $i \geq label(T)$ **then** $g[i,T] \leftarrow (i - label(T)) \times c_\lambda$

    1.1.2 **if** $i < label(T)$ **then** $g[i,T] \leftarrow (label(T) - i) \times c_\delta$

  1.2 **exit**

2. *GenCost(T_L); GenCost(T_R);*

3. $[l_1, l_2] \leftarrow OPT(T_L)$; $[r_1, r_2] \leftarrow OPT(T_R)$

4. $t_1 \leftarrow \min\{l_1, r_1\}$; $t_2 \leftarrow \min\{l_2, r_2\}$

5. **for** $i \leftarrow t_1$ **to** t_2 **do** $g[i,T] \leftarrow g[i,T_L] + g[i,T_R]$

6. **for** $i \leftarrow t_2 + 1$ **to** m **do** $g[i,T] \leftarrow \min\{g[i-1,T] + c_\lambda, g[i,T_L] + g[i,T_R]\}$

7. **for** $i \leftarrow t_1 - 1$ **downto** 1 **do** $g[i,T] \leftarrow \min\{g[i+1,T] + c_\delta, g[i,T_L] + g[i,T_R]\}$

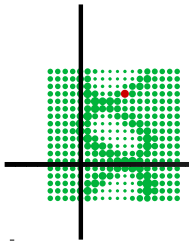


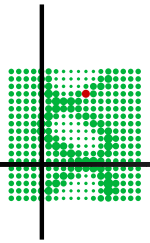
Table of Time Complexities

	One optimal answer	k optimal answer
Durand et al. 2005	$O(nm^2)$	$O(nm^2 + nmk)$
This work	$O(nm)$	$O(nm + nk)$
This work (unit costs)	$O(n)$	$O(nk)$

n the size of the species tree.

m the maximum number of gene copies in a species.

$-c_\delta \leq \Delta g(x, T) \leq c_\lambda$, so m can be replaced by $\min\{m, c_\delta + c_\lambda\}$



Summary and Future Work

- We proposed an algorithm running $O(m)$ times faster than the previous algorithm. In some gene families like kinases m can be a large number (several hundreds).

- We are currently working on including horizontal gene transfers into the model. should be considered in the macro-evolutionary problem.

ET CE SOIR ..... **ALLEZ LES BLEUS**