Minimum Leaf Removal for Reconciliation: Complexity and Algorithms

Riccardo Dondi [1] **Nadia El-Mabrouk** [2]

[1]Università di Bergamo, [2]Université de Montréal

CPM 2012, July 5, 2012

Outline

Outline

Biological Motivations

Genome evolution $\longrightarrow$ micro-evolutionary events and macro-evolutionary events

Macro-evolutionary events:

- **speciations**
- **duplications**
- losses

Understanding gene families evolution: fundamental in functional annotation, phylogenetic inference, comparative genomics

Gene tree - Species Tree

Gene families evolution:

- **Species tree** S: binary rooted tree; leaves: **uniquely leaf-labeled**
- **Gene tree** T: binary rooted tree; leaves: **not uniquely leaf-labeled**

Gene trees of different gene families (and species tree) may be **different**: duplications, losses, ...

Gene tree - Species Tree

Two main problems in gene families evolution:

- **Reconciliation** of a gene tree with a given species tree $\rightarrow$ inference of macro-evolutionary events
- **Inference** of a species tree from a set of given gene trees $\rightarrow$ inference of evolutionary history of the species

Errors in Gene Trees

A major problem in gene tree reconciliation: high sensitivity to error-prone gene trees

Strategies for preprocessing a gene tree T prior to reconciliation

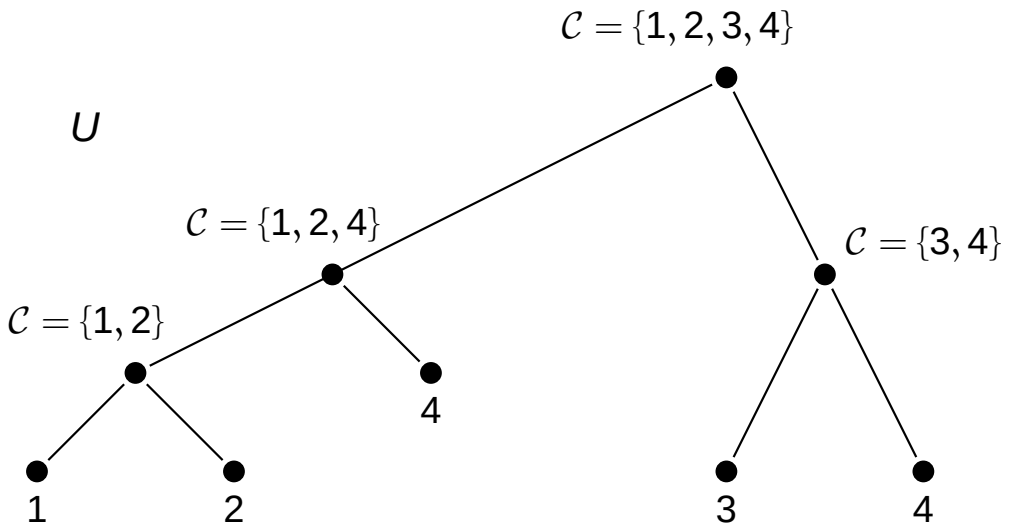
- local operations (NNIs) [`Chen et al, JCB 2007`; `Durant et al, JCB 2006`, `Eulenstein et al, ISBRA 2012`]
- **remove misplaced leaves** (gene copies)

LCA-mapping

Comparison of gene tree T and a species tree S: mapping between the nodes of T and $S \rightarrow$ **LCA-mapping** $\text{lca}_{T,S}$

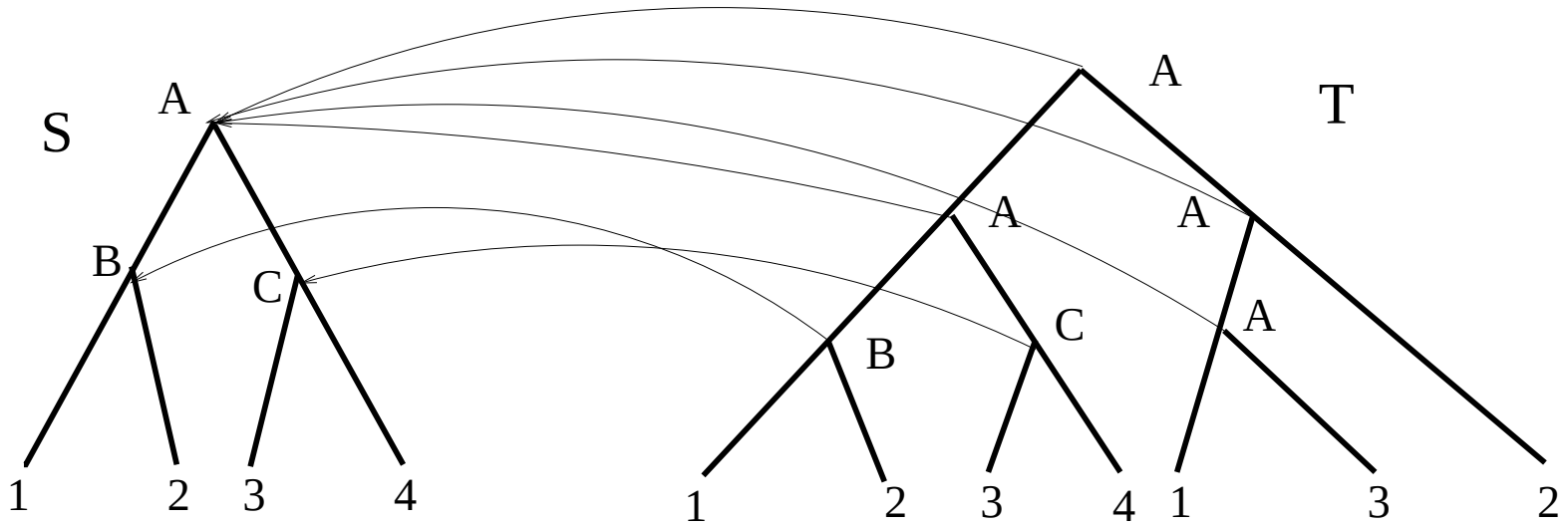
Definition (Cluster)

Given a tree U and a node x in U, $\mathcal{C}(x)$ leaf-labels of the subtree of U rooted at x.



LCA-mapping

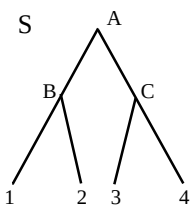
$\text{lca}_{T,S}$ maps every node x of T to the **Lowest Common Ancestor** (**LCA**) of $\mathcal{C}(x)$ in S.

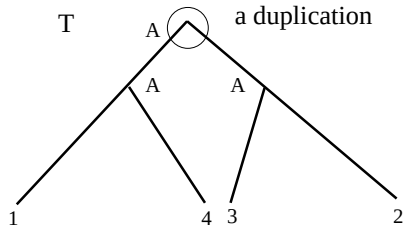


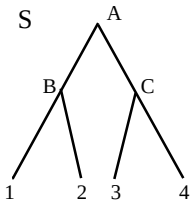
Duplications and Speciations

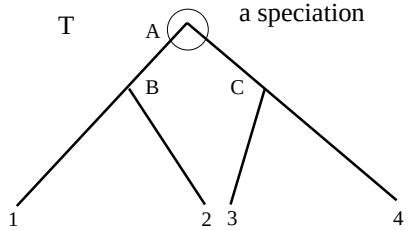
A node x of T is:

- a **duplication**: x and at least one of its children are mapped by $\text{lca}_{T,S}$ in the same node of S.
- a **speciation**: x and its children are mapped by $\text{lca}_{T,S}$ in different nodes of S.







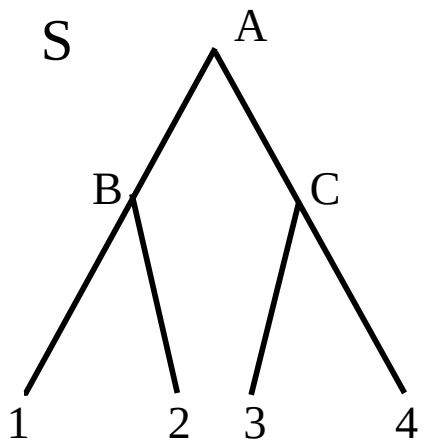


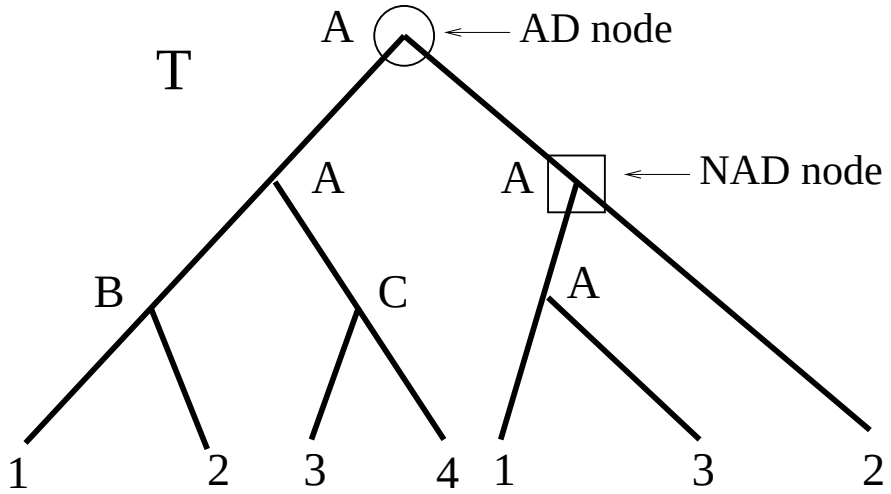
AD nodes and NAD nodes

The duplication nodes of T w. r. t. S

- *apparent duplications (AD nodes)*
- *non-apparent duplications (NAD) nodes*

NAD nodes: potentially resulting from misplaced of leaves in T
[Chauve and El-Mabrouk, RECOMB 2009]





Combinatorial Problem

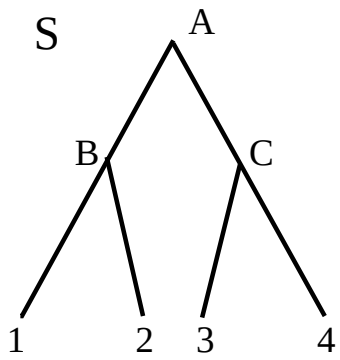
MinLeafRem: a combinatorial problem to remove misplaced leaves [Doroftei and El-Mabrouk, WABI 2011]

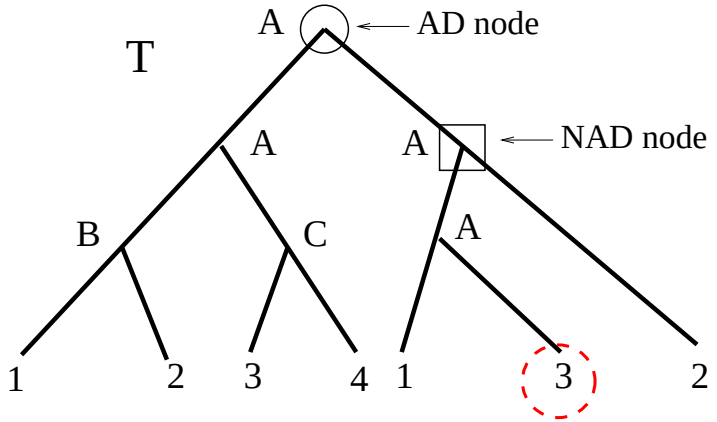
Problem

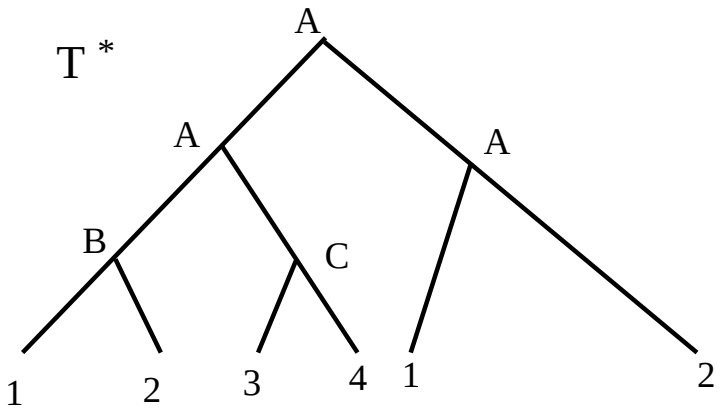
Minimum Leaf Removal Problem*[MinLeafRem]*
Input*: A gene tree T and a species tree S, both leaf-labelled by Γ.*
Output*: A tree T^* that contains only AD-nodes and speciation nodes and such that T^* is obtained from T by a minimum number of leaf removals.*

Combinatorial Problem

MinLeafRem - An example







MinLeafRem

Previous results [Doroftei and El-Mabrouk, WABI 2011]:

- Exact polynomial-time algorithm when the gene tree is uniquely leaf-labeled (see Maximum Agreement Subtree)
- Exact polynomial-time algorithm when there is no AD above NAD
- Polynomial-time heuristic for the general case

Outline

Computational Complexity of MinLeafRem

Theorem

MinLeafRem is APX-hard, even in the restricted case that each label is associated with at most two leaves of T.

Proof.

L-reduction from Minimum Vertex Cover on Cubic Graphs. □

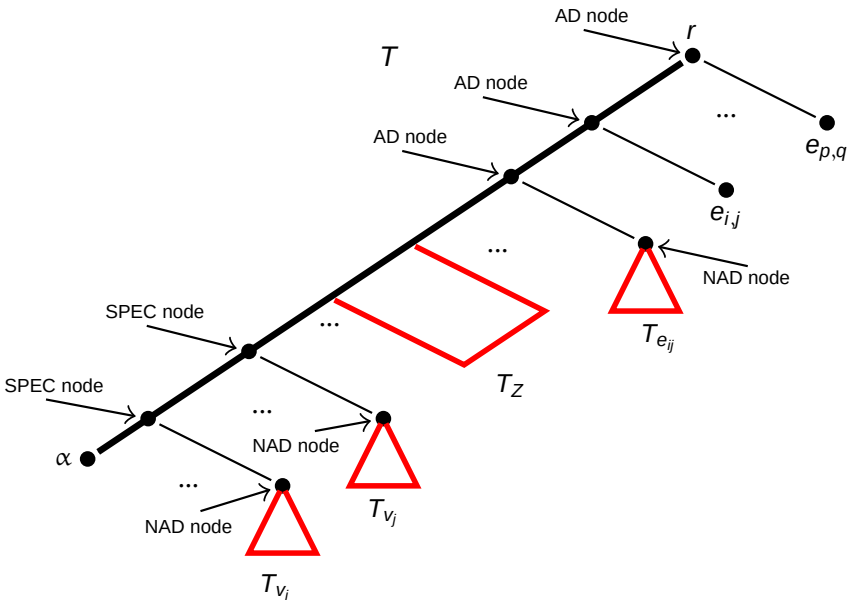
Computational Complexity of MinLeafRem

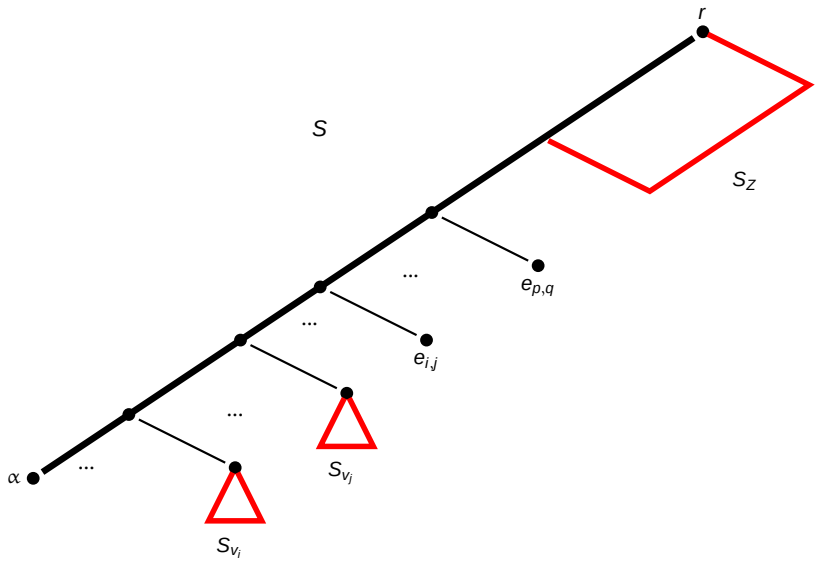
Problem

Minimum Vertex Cover Problem on Cubic graphs[MVCC]
Input: *A cubic graph* $G = (V, E)$ *where* $V = \{v_1, \ldots, v_n\}$ *is the set of vertices and* E *the set of edges of* G .
Output: *A minimum cardinality set* $V' \subseteq V$, *such that for each edge* $e_{i,j} = \{v_i, v_j\} \in E$, *at least one of* v_i, v_j *belongs to* V'.

Given a cubic graph $G \rightarrow$ corresponding instance (T, S) of MinLeafRem

Computational Complexity of MinLeafRem





Computational Complexity of MinLeafRem

Main properties of the reduction:

- Leaves removed from $T_{v_i} \rightarrow v_i$ in Vertex Cover/Independent Set
- no leaf of T_Z is removed
- Leaf removed from $T_{e_{ij}} \rightarrow$ no AD node is affected

Outline

Parameterizations

We focus on some possible natural parameterizations:

- **size of the solution** (the number of leaves removed)
- number of labels in Γ associated with **multiple leaves** of T
- maximum number of leaves in T associated with a **single label** of $\Gamma \Rightarrow$ already APX-hard

Parameter: Number of Leaves Removed

Theorem

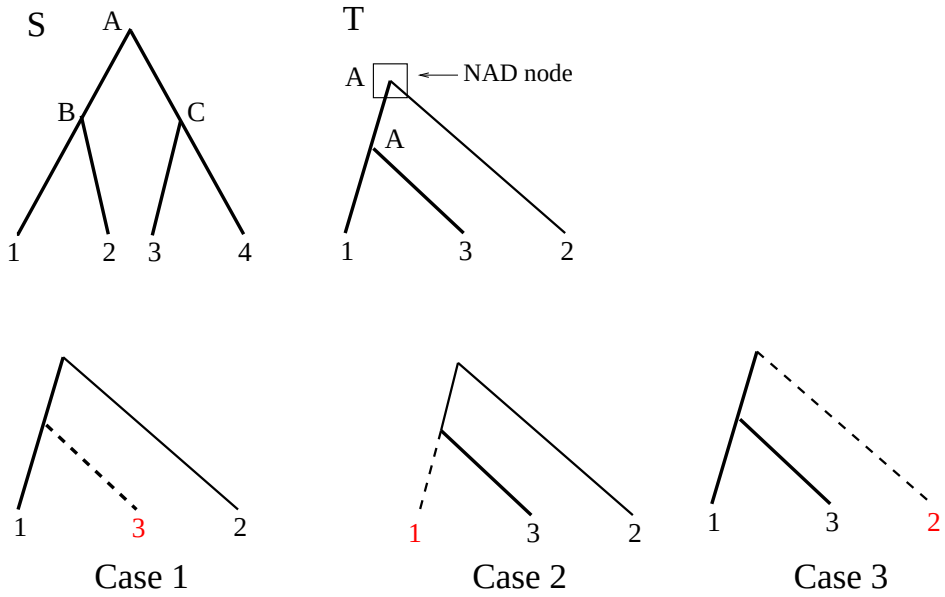
There is a fixed-parameter algorithm for for MinLeafRem of time complexity $O(3^c poly\ (|V(T)|, |V(S)|))$, where c is the number of leaves removed.

Depth-bounded search tree technique

Parameter: Number of Leaves Removed

Depth-bounded search tree algorithm:

- If T does not contain NAD nodes $\rightarrow$ no leaf removal
- for a NAD node x of T:
  - at least one leaf has to be removed
  - 3 possible sets of leaves has to be removed



Parameter: Number of Labels with Multiple Copies

$\Gamma_D \subseteq \Gamma$: set of labels associated with **multiple leaves** of T.

Theorem

An optimal solution of MinLeaf over instance (T, S) can be computed in time $O(4^{|\Gamma_D|} poly(|V(T)||V(S)|))$.

Proof.

Dynamic programming algorithm □

Parameter: Number of Labels with Multiple Copies

Dynamic programming recurrence:

$$M[T(x), S(y), \Gamma'_D] = \min_{\substack{\Gamma'_{1,D} \subseteq \Gamma'_D, \\ \Gamma'_{2,D} \subseteq \Gamma'_D, \\ \Gamma'_{1,D} \cup \Gamma'_{2,D} = \Gamma'_D}} \begin{cases} M[T(x_l), S(y_l), \Gamma'_{1,D}] + M[T(x_r), S(y_r), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset, \\ M[T(x_l), S(y_r), \Gamma'_{1,D}] + M[T(x_r), S(y_l), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset, \\ M[T(x_l), S(y), \Gamma'_{1,D}] + M[T(x_r), S(y), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} \neq \emptyset \\ M[T(x_l), S(y), \Gamma'_D] + |L(T(x_r))| \\ M[T(x_r), S(y), \Gamma'_D] + |L(T(x_l))| \\ M[T(x), S(y_l), \Gamma'_D] \\ M[T(x), S(y_r), \Gamma'_D] \end{cases}$$

Outline

Open Problems

Open Problems:

- Fixed-Parameter Tractability
  - improve the complexity of the fixed-parameter algorithms
  - **kernelization**
- study the problem under new biological meaningful parameters

Thank you!