Algorithms for Exploring the Space of Gene Tree/Species Tree Reconciliations

Jean-Philippe Doyon[1] Cedric Chauve[2] Sylvie Hamel[1]

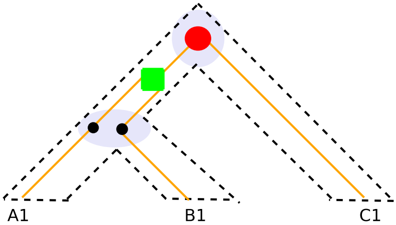
1- Département d'Informatique et de Recherche Opérationnelle, Université de Montréal
2- Department of Mathematics, Simon Fraser University

RECOMB Comparative Genomics
Paris, October 2008

Gene Family Evolution

The evolution of a genome is determined by

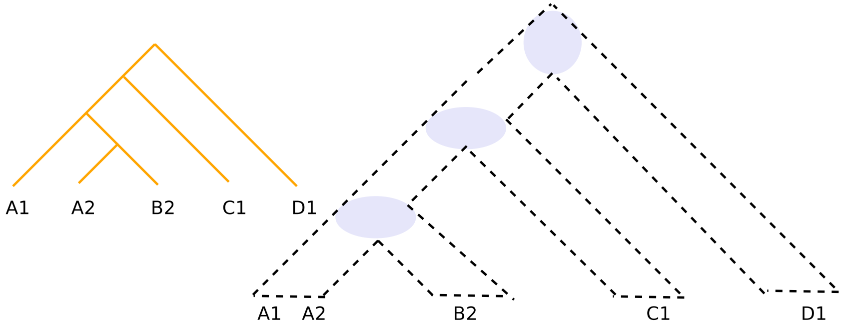
- speciation (●): new species are created;
- duplication (■): a gene is duplicated into two copies;
- loss (•): a gene has no function or is deleted from the genome.



Why it is important to study the evolution of homologous genes?

- Orthologous and paralogous genes
- Gene content of ancestral genomes
- Phylogenomic

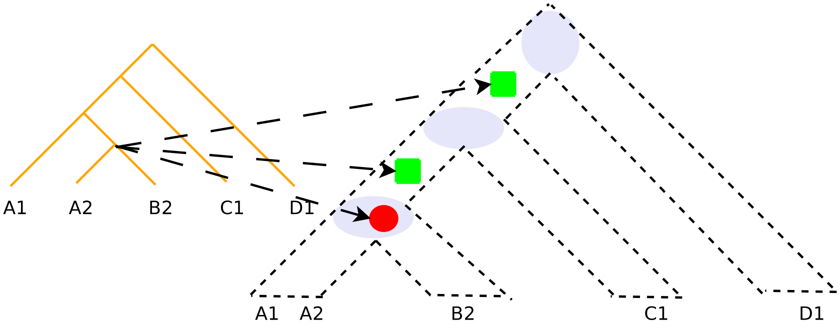
The Problem



The Main Question

Define the evolution of the gene tree (G) according to species tree (S) in term of speciation, duplication, and loss events.

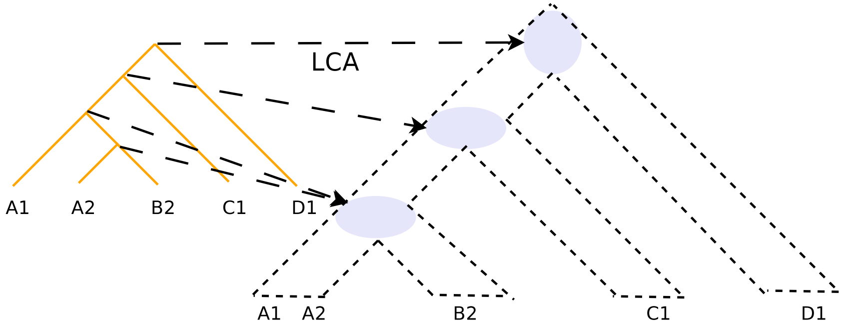
The Problem



The Main Question

Define the evolution of the gene tree (G) according to species tree (S) in term of speciation, duplication, and loss events.

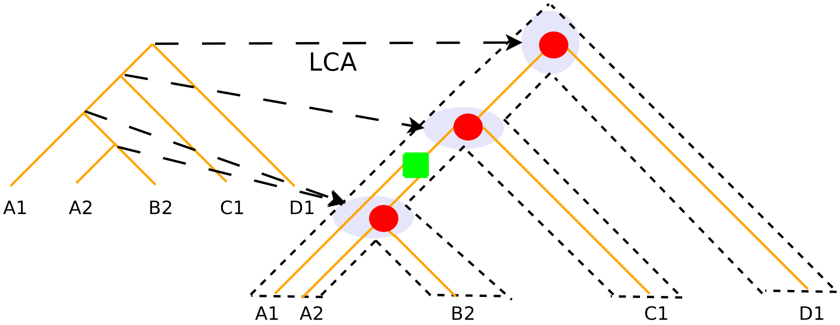
Definitions



The mapping $LCA : V(G) \to V(S)$

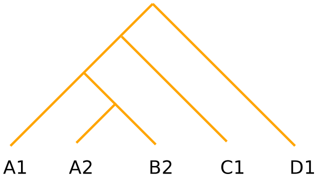
maps a gene u (of G) onto the **Last Common Ancestor** species (of S) of all species that contain a gene descendant of u.

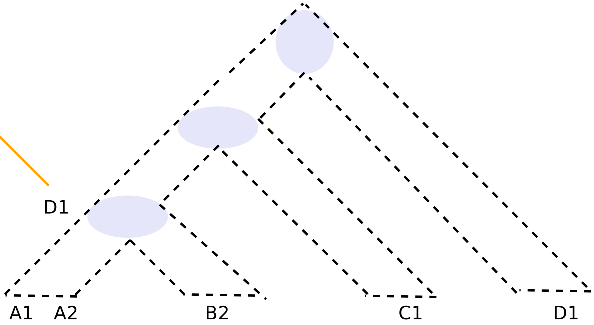
Definitions



- **LCA reconciliation**: the most parsimonious one (Chauve and El-Mabrouk (2009)).
- Arvestad et al. (2004), Bonizzoni et al. (2005), Górecki and Tiuryn (2006).

A more General Definition

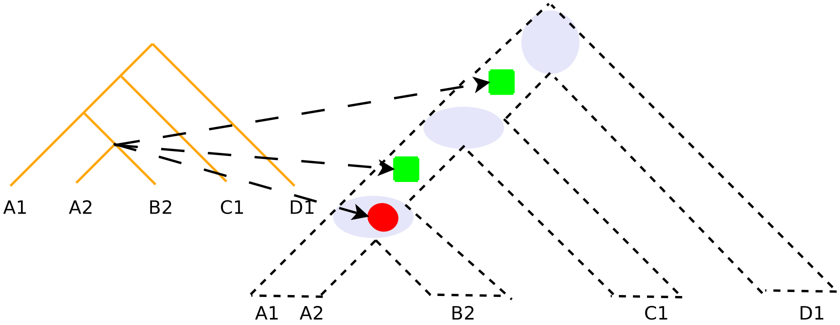




A Reconciliation between *G* and *S*

- Each leaf of *G* is mapped on the corresponding leaf of *S*.
- Each internal node is mapped either on the LCA or on an edge above.
- The descendance relationship of two nodes of *G* has to be respected in their mappings in *S*.

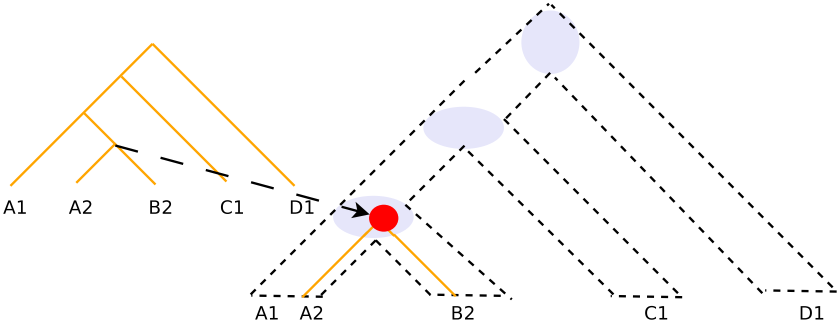
A more General Definition



A Reconciliation between *G* and *S*

- Each leaf of *G* is mapped on the corresponding leaf of *S*.
- Each internal node is mapped either on the LCA or on an edge above.
- The descendance relationship of two nodes of *G* has to be respected in their mappings in *S*.

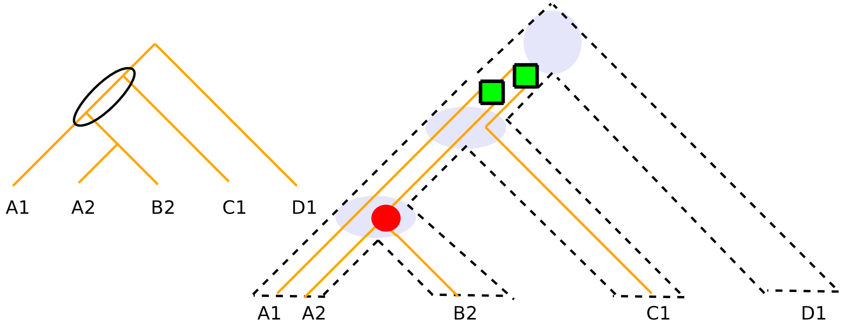
A more General Definition



A Reconciliation between *G* and *S*

- Each leaf of *G* is mapped on the corresponding leaf of *S*.
- Each internal node is mapped either on the LCA or on an edge above.
- The descendance relationship of two nodes of *G* has to be respected in their mappings in *S*.

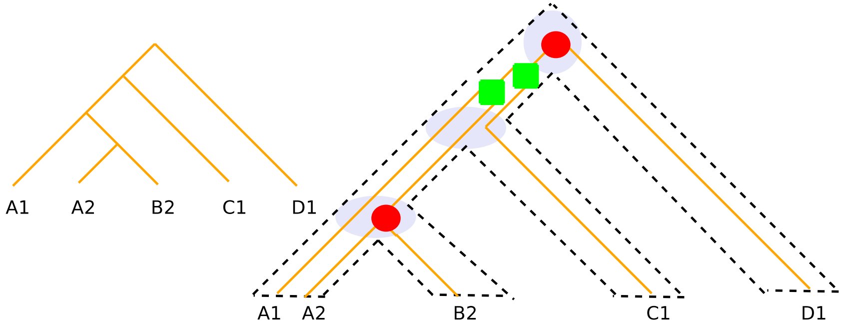
A more General Definition



A Reconciliation between G and S

- Each leaf of G is mapped on the corresponding leaf of S.
- Each internal node is mapped either on the LCA or on an edge above.
- The descendance relationship of two nodes of G has to be respected in their mappings in S.

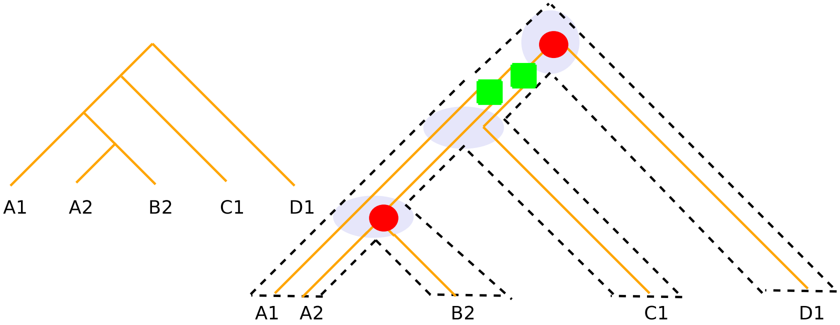
A more General Definition



A Reconciliation between *G* and *S*

- Each leaf of *G* is mapped on the corresponding leaf of *S*.
- Each internal node is mapped either on the LCA or on an edge above.
- The descendance relationship of two nodes of *G* has to be respected in their mappings in *S*.

A more General Definition



Properties

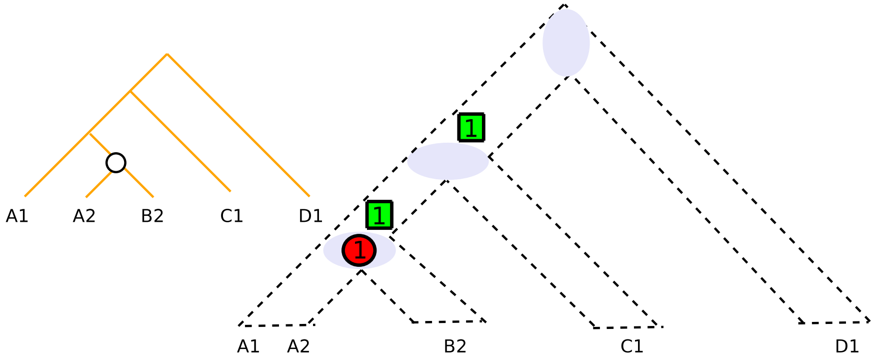
- Does not ONLY induce the LCA reconciliation.
- The number of reconciliations is finite, but can be exponential.

Reconciliation Space Exploration

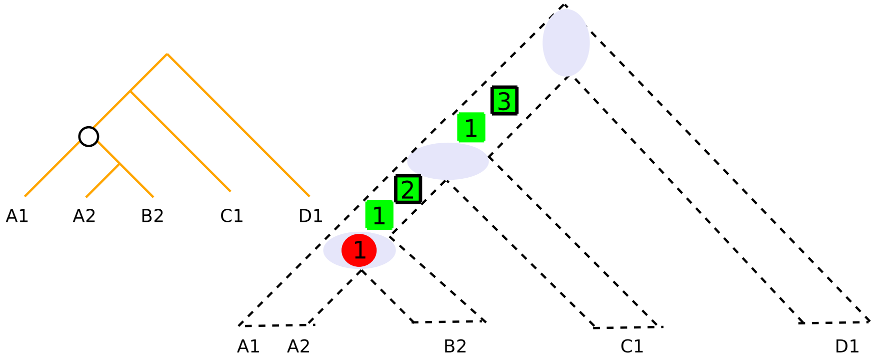
This simple definition allows

- Count the number of reconciliations.
- Generate randomly and uniformly a reconciliation.
- Define operators used to explore the whole space.
- Exhaustively explore the space.

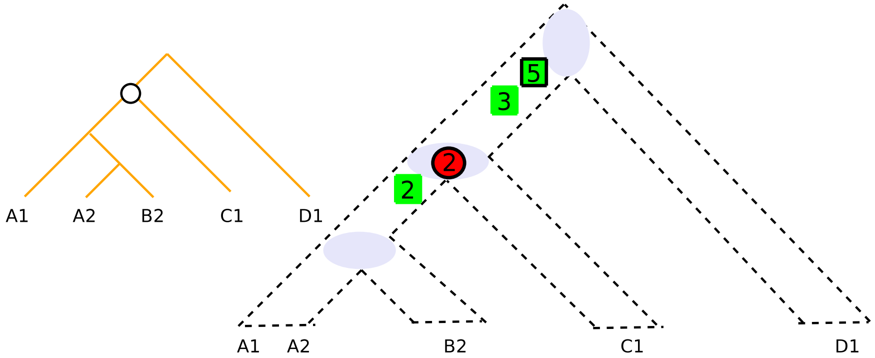
Counting the Number of Reconciliations



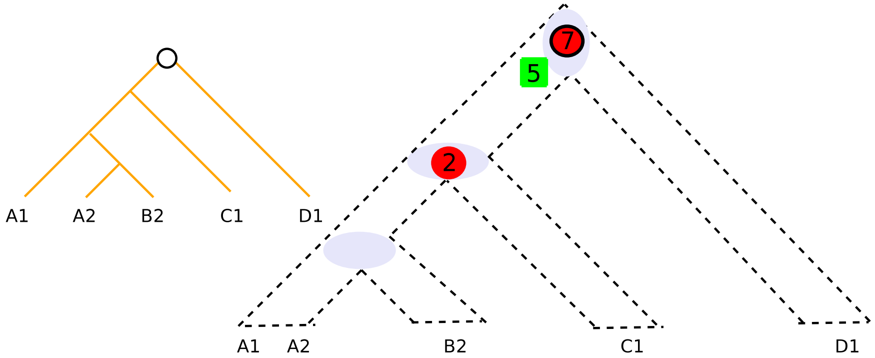
Counting the Number of Reconciliations



Counting the Number of Reconciliations



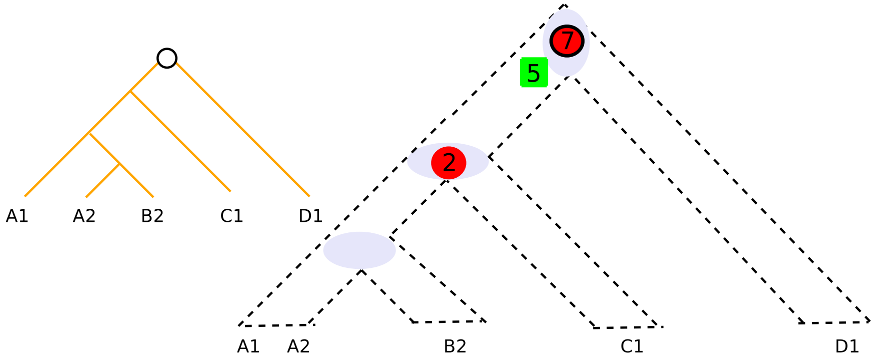
Counting the Number of Reconciliations



Propositions

- Dynamic programming algorithm in $O(|G||S|)$ time and space.
- Similar algorithm for the # of reconciliations that minimizes the duplication cost.

Counting the Number of Reconciliations



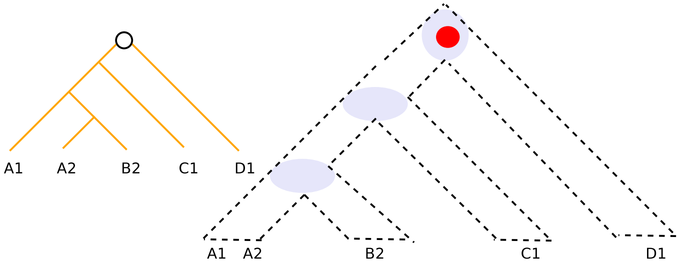
Propositions

- Dynamic programming algorithm in $O(|G||S|)$ time and space.
- Similar algorithm for the # of reconciliations that minimizes the duplication cost.

Uniform Random Generation

Algorithm

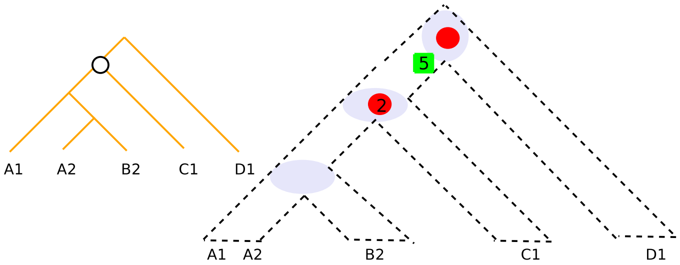
- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Uniform Random Generation

Algorithm

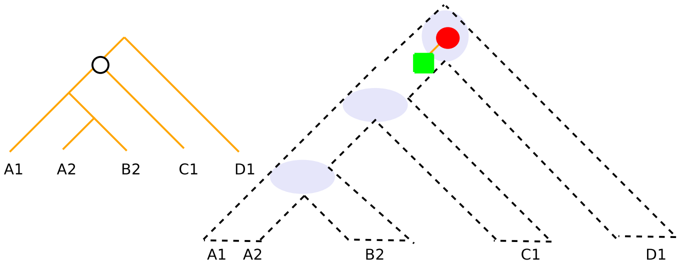
- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Uniform Random Generation

Algorithm

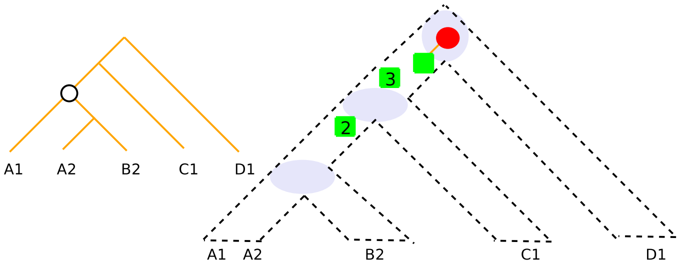
- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Uniform Random Generation

Algorithm

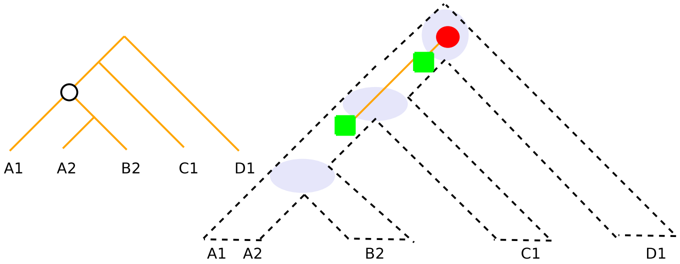
- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Uniform Random Generation

Algorithm

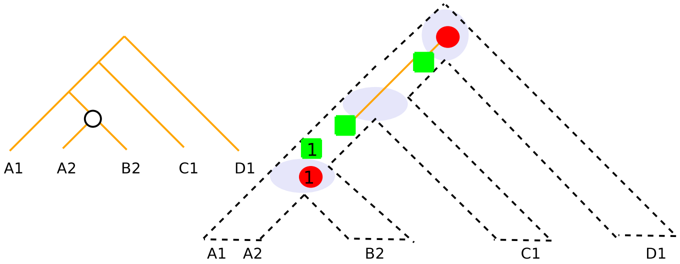
- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Uniform Random Generation

Algorithm

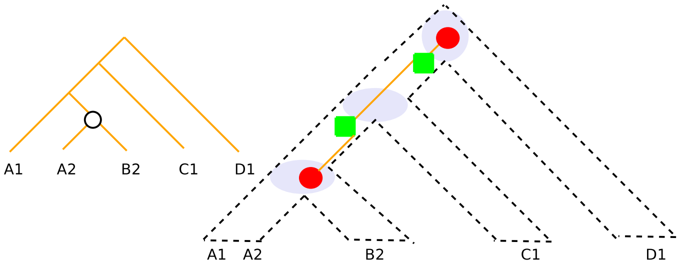
- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Uniform Random Generation

Algorithm

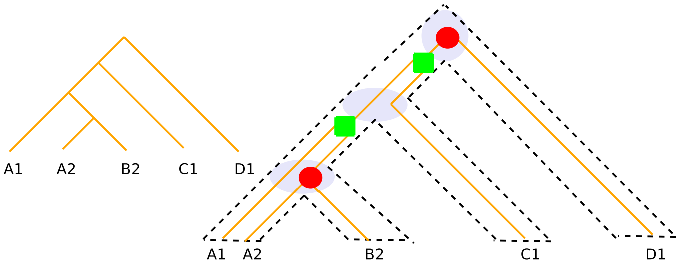
- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Uniform Random Generation

Algorithm

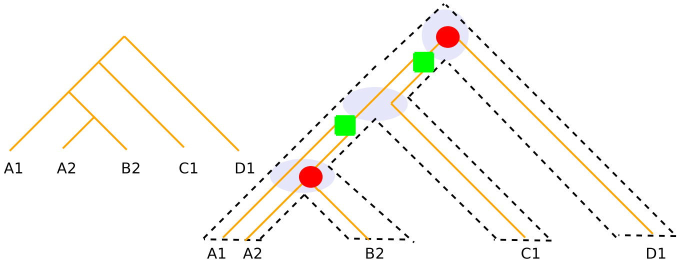
- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Uniform Random Generation

Algorithm

- Prefix traversal of G (u is the current node);
- Randomly select a node/edge c of S according to $Nb(u, c)$.



Theorem

- Uniform distribution over all reconciliations.
- Preprocessing of $O(|G||S|)$ for the counting.
- Worst case in $\Theta(|G||S|)$ and best case in $\Theta(|G|)$.
- Space in $O(|G||S|)$.

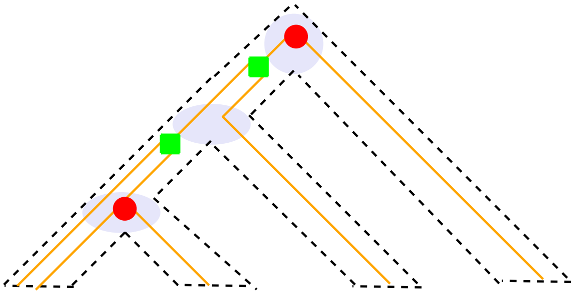
Nearest Mapping Change

Upward NMC

- Changes a speciation into a duplication.
- Moves a duplication upward.

Downward NMC

- Changes a duplication into a speciation.
- Moves a duplication downward.



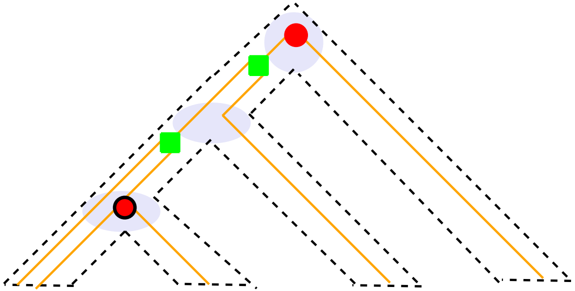
Nearest Mapping Change

Upward NMC

- Changes a speciation into a duplication.
- Moves a duplication upward.

Downward NMC

- Changes a duplication into a speciation.
- Moves a duplication downward.



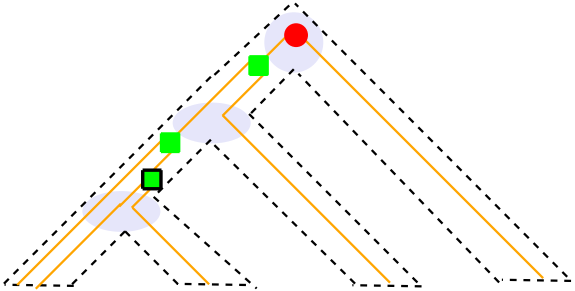
Nearest Mapping Change

Upward NMC

- Changes a speciation into a duplication.
- Moves a duplication upward.

Downward NMC

- Changes a duplication into a speciation.
- Moves a duplication downward.



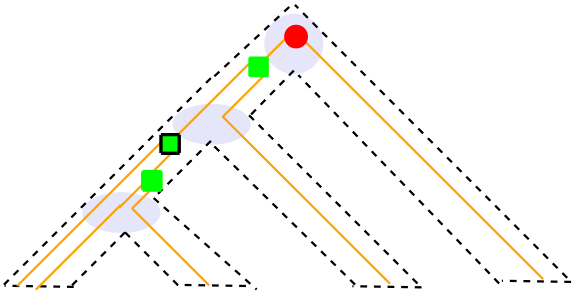
Nearest Mapping Change

Upward NMC

- Changes a speciation into a duplication.
- Moves a duplication upward.

Downward NMC

- Changes a duplication into a speciation.
- Moves a duplication downward.



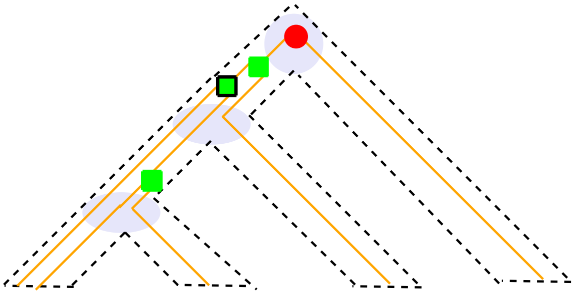
Nearest Mapping Change

Upward NMC

- Changes a speciation into a duplication.
- Moves a duplication upward.

Downward NMC

- Changes a duplication into a speciation.
- Moves a duplication downward.



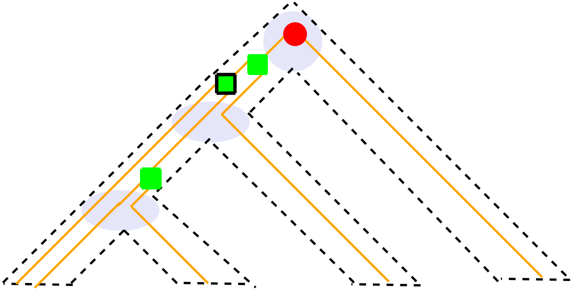
Nearest Mapping Change

Upward NMC

- Changes a speciation into a duplication.
- Moves a duplication upward.

Downward NMC

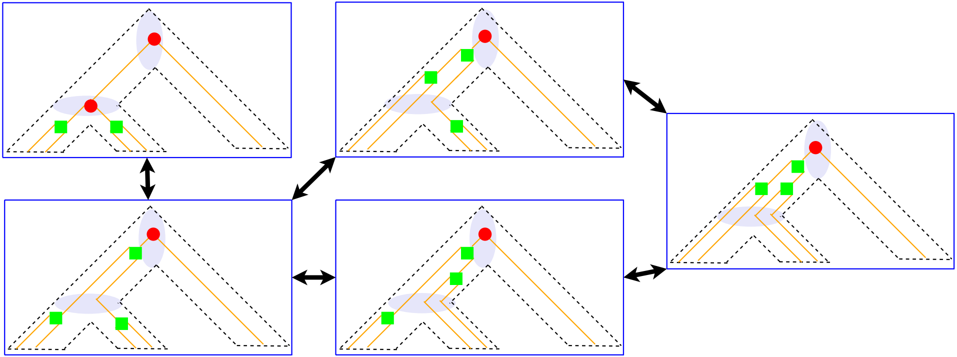
- Changes a duplication into a speciation.
- Moves a duplication downward.



Properties

- Sufficient to explore the whole space of reconciliations.
- The number of duplications and losses can be updated in constant time.

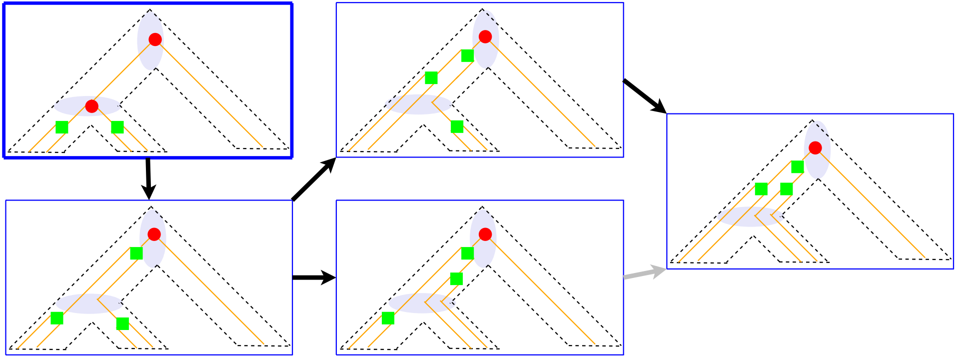
Exhaustive Exploration



Graph over the space

- $V(G) = \{\text{reconciliations}\}$
- $E(G) = \{\text{NMC operators}\}$

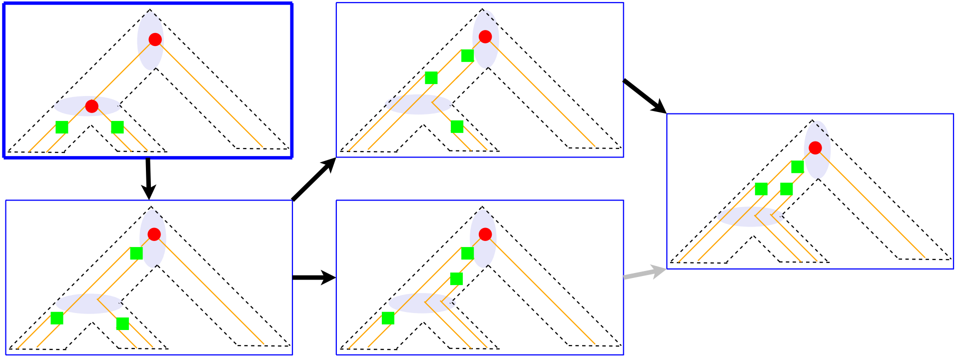
Exhaustive Exploration



Tree over the graph

- The root is the LCA reconciliation
- Upward NMCs only
- Lexicographic order constraint on the NMCs.

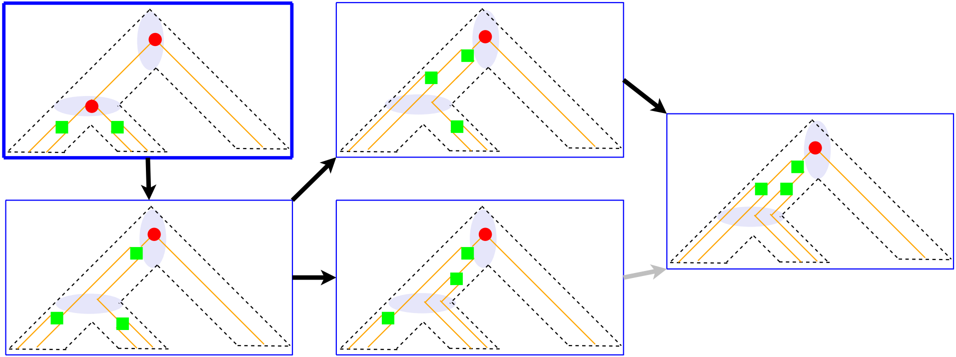
Exhaustive Exploration



Proposition

The tree is a spanning tree of the graph

Exhaustive Exploration



Theorem

The algorithm is in Constant Amortized Time $\Theta(\#rec)$ and space $O(|G||S|)$.

Experimental Results

Generation of synthetic gene trees

- Phylogenetic tree of 12 *Drosophila* with valued branches (Hahn, 2007).
- 1000 gene trees were generated using the birth-and-death process.
- 249 gene trees were unique (6 to 22 leaves).

Distribution of the duplication cost

- 237 gene trees have 1 optimal reconciliation.
- 12 gene trees have 2 optimal rec., always seperated by 1 NMC.
- The LCA reconciliation is equal or at 1 NMC to the true one.
- $\#dup(\alpha) - \#dup(\alpha^*)$ is proportional both to ($\alpha^* =$ optimal rec.)
  - the NMC distance between α and α^*;
  - how much α differs from the true reconciliation.

Experimental Results

Generation of synthetic gene trees

- Phylogenetic tree of 12 *Drosophila* with valued branches (Hahn, 2007).
- 1000 gene trees were generated using the birth-and-death process.
- 249 gene trees were unique (6 to 22 leaves).

Distribution of the duplication cost

- 237 gene trees have 1 optimal reconciliation.
- 12 gene trees have 2 optimal rec., always seperated by 1 NMC.
- The LCA reconciliation is equal or at 1 NMC to the true one.
- $\#dup(\alpha) - \#dup(\alpha^*)$ is proportional both to ($\alpha^* =$ optimal rec.)
  - the NMC distance between α and α^*;
  - how much α differs from the true reconciliation.

Experimental Results

Generation of synthetic gene trees

- Phylogenetic tree of 12 *Drosophila* with valued branches (Hahn, 2007).
- 1000 gene trees were generated using the birth-and-death process.
- 249 gene trees were unique (6 to 22 leaves).

Distribution of the duplication cost

- 237 gene trees have 1 optimal reconciliation.
- 12 gene trees have 2 optimal rec., always seperated by 1 NMC.
- The LCA reconciliation is equal or at 1 NMC to the true one.
- $\#dup(\alpha) - \#dup(\alpha^*)$ is proportional both to (α^* = optimal rec.)
  - the NMC distance between α and α^*;
  - how much α differs from the true reconciliation.

Experimental Results

Generation of synthetic gene trees

- Phylogenetic tree of 12 *Drosophila* with valued branches (Hahn, 2007).
- 1000 gene trees were generated using the birth-and-death process.
- 249 gene trees were unique (6 to 22 leaves).

Distribution of the duplication cost

- 237 gene trees have 1 optimal reconciliation.
- 12 gene trees have 2 optimal rec., always seperated by 1 NMC.
- The LCA reconciliation is equal or at 1 NMC to the true one.
- $\#dup(\alpha) - \#dup(\alpha^*)$ is proportional both to ($\alpha^* =$ optimal rec.)
  - the NMC distance between α and α^*;
  - how much α differs from the true reconciliation.

Conclusion

Theorical Contributions

- MCMC can use the random algorithm, and the NMCs.
- Exploration algorithm is optimal in time.

Experimental Contributions

- Large number of reconciliatons (1 000 000), even with low rates.
- Few (near) optimal reconciliations, always close to the LCA one.
- Parsimony can be a good criteria to find the true reconciliation.

Future Work

- Higher duplication and loss rates.
- Local neighbourhood exploration.
- Maximum likelihood cost.
- Non binary trees and multiple gene duplications.