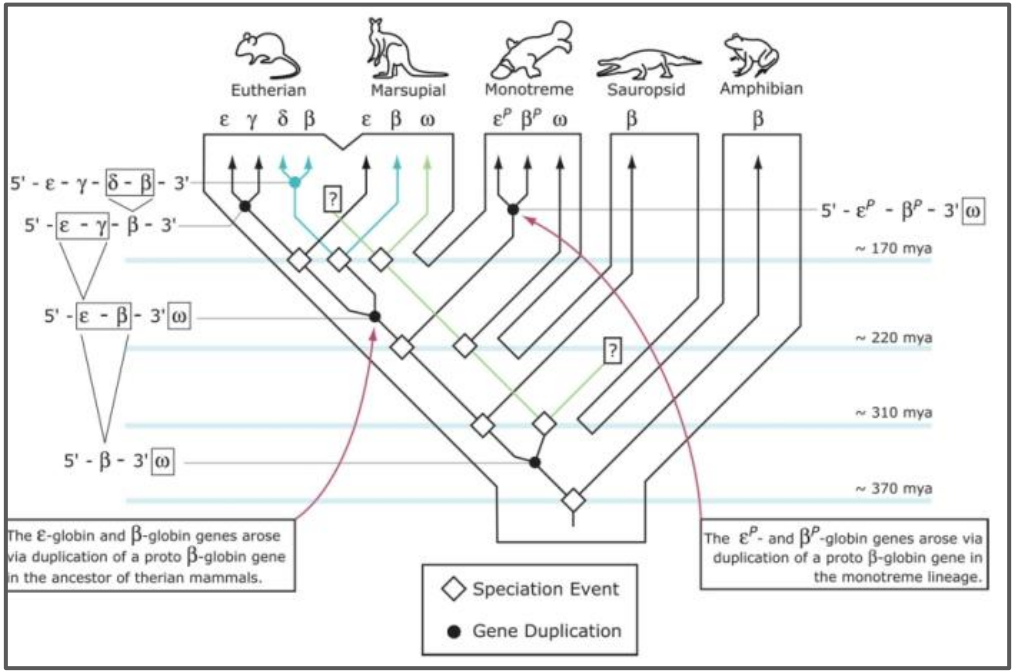
Gene tree / species tree reconciliation

Cedric Chauve
Department of Mathematics
Simon Fraser University

Homologous gene family

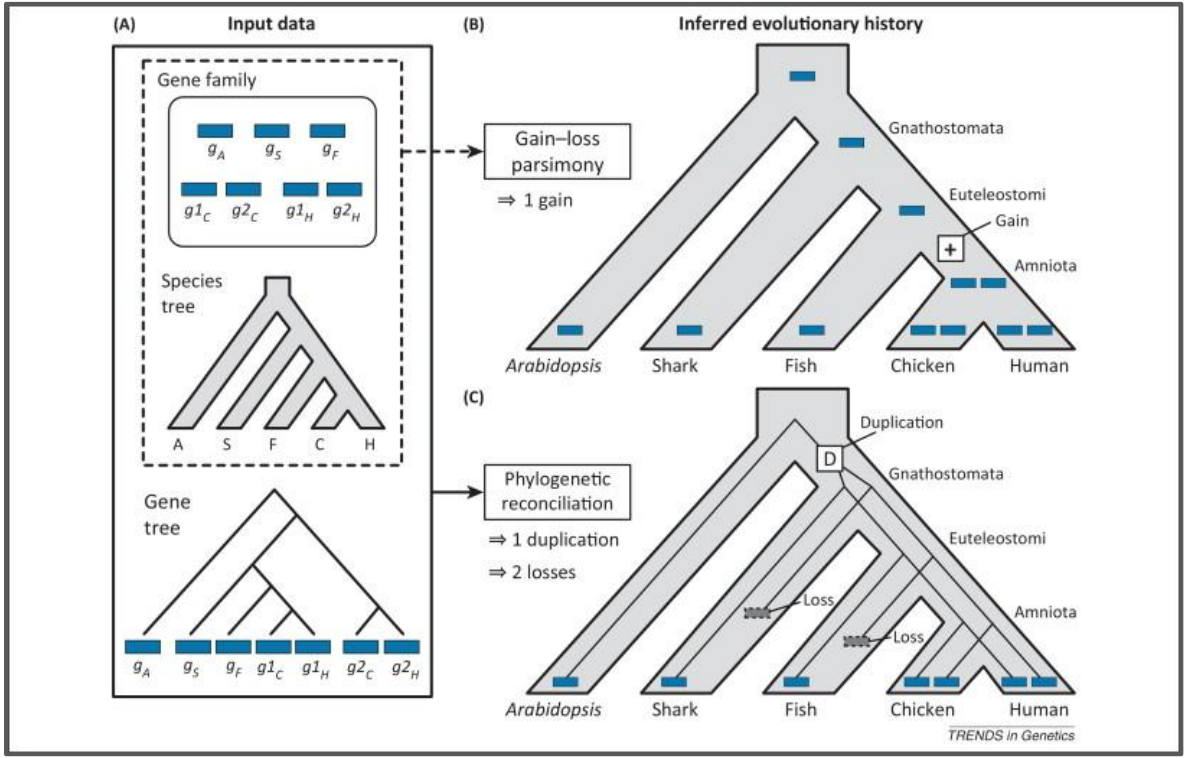
A gene family **within a group S of species** is a **group of genes** that have evolved from a unique ancestral gene from the **Last Common Ancestor of S**, through **speciation**, **gene duplication**, **gene loss**, **gene transfer, incomplete lineage sorting (ILS).**



A gene family is naturally associated to a **gene tree G** that describes the true evolution of the genes in the family.

This gene tree is **conditioned** by the **species tree** encoding the species evolution.

Gene tree / species tree



We are given

- ❑ A gene family.
- ❑ A gene tree **G** for this family.
- ❑ A species tree **S**.

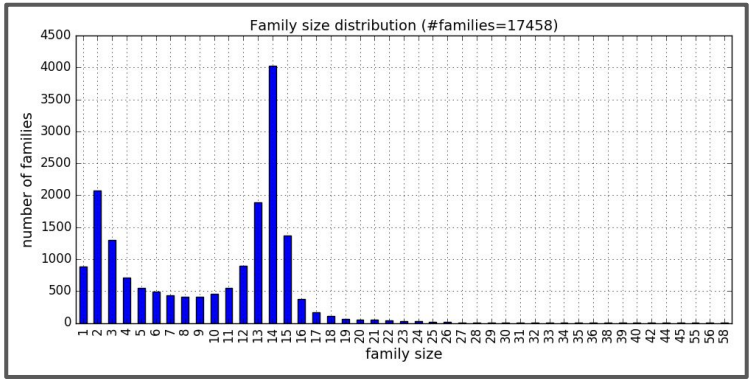
We want to **reconcile G with S**, i.e. explain how **G** could have occured knowing it is conditioned by **S** (bottom).
It is more refined than to explain how the number of copies of the gene observed in each species could have occured (top).

Why are-we interested in reconciliations?

Our goal down the road is to study genome rearrangements evolution for the group of mosquito genomes we have been discussing, including the reconstruction of ancestral gene orders.

We have gene families including a significant number with duplicated/lost genes.

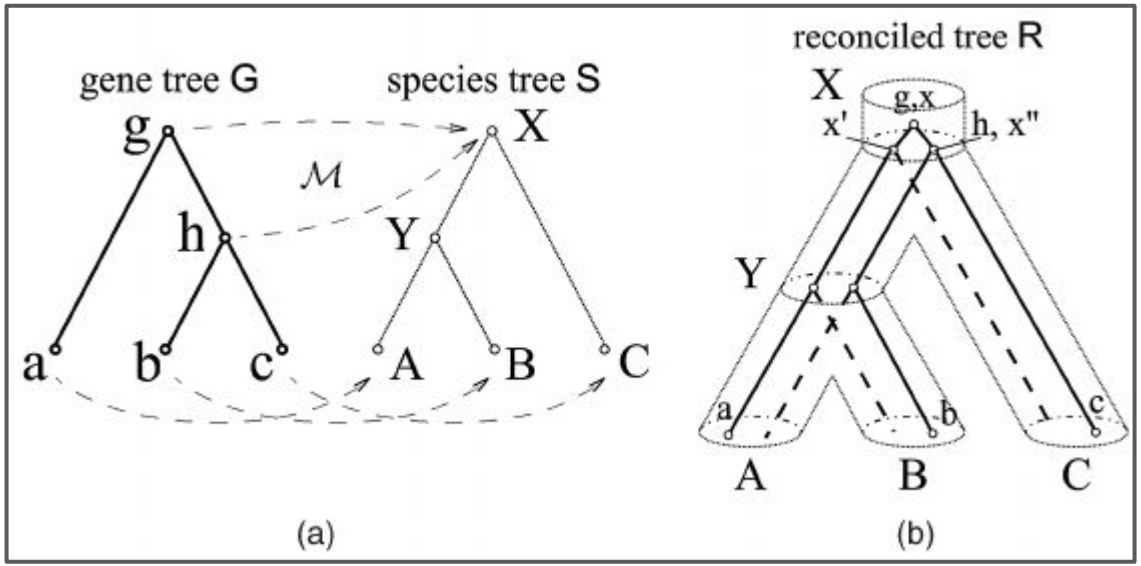
Reconciled gene trees provide the information of the **gene content of ancestral species** and form the basic input for some genome rearrangements algorithms that account for gene duplication.



Reconciliation: mapping genes to species

The basis of reconciliation methods, consists into

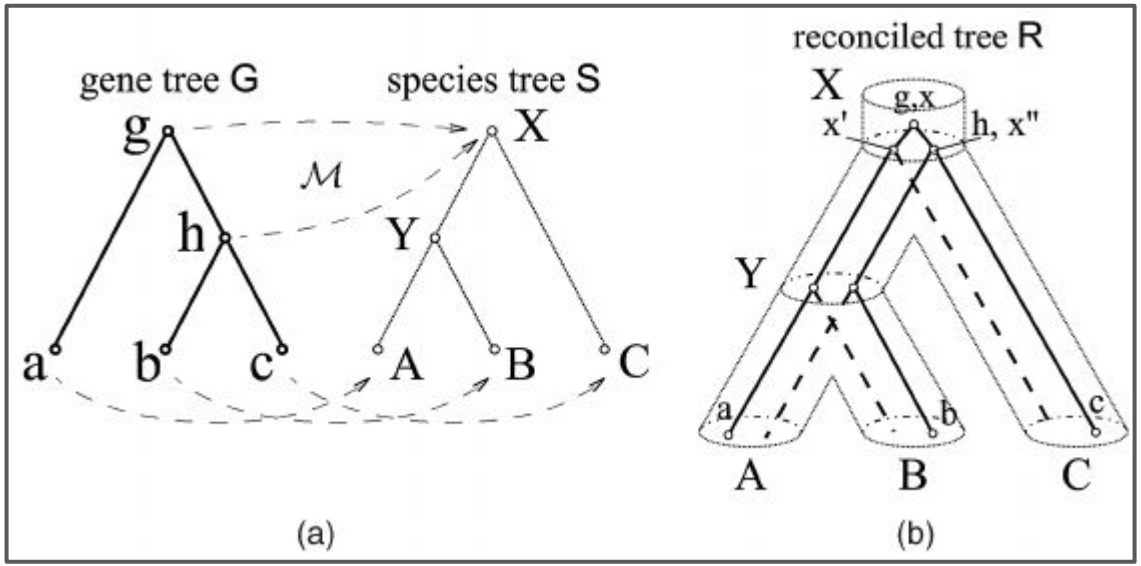
1. **Mapping** the nodes (genes, extant and ancestral) of **G** to the nodes of **S** (species), while **respecting ancestrality constraints**: if **g** is mapped to **X** and its descendant **h** is mapped to **Y**, **Y** can not be an ancestor of **X**.
2. Inferring **evolutionary events** from this mapping.



An example of reconciliation that gives as gene content 2 genes in root species **X**, two genes in **Y** and explains the evolution of **G** with one **gene duplication** and **three gene losses**.

Reconciliation: identifying evolutionary events (S)

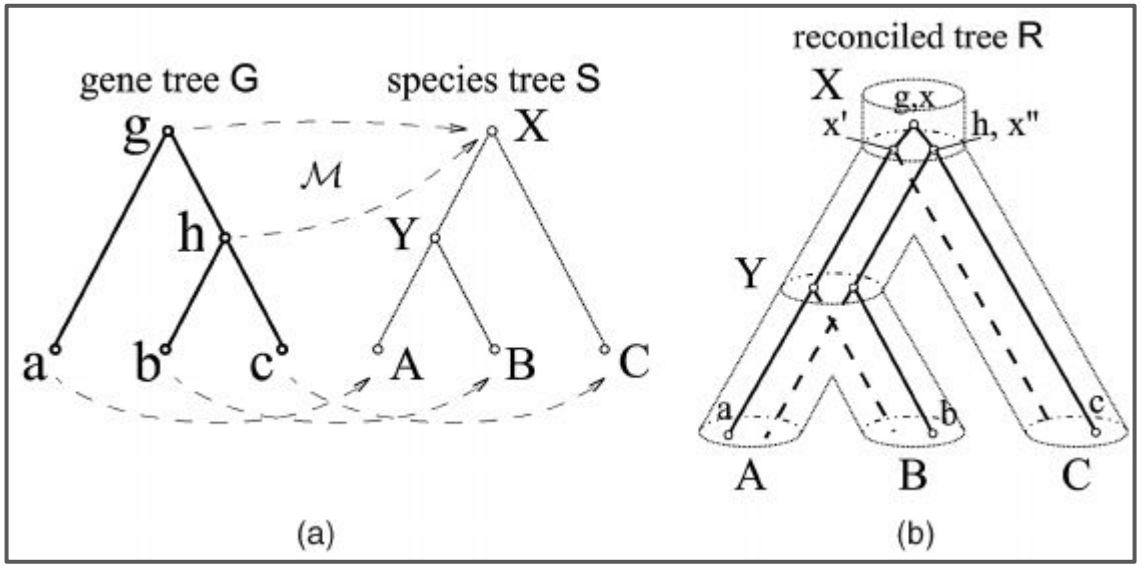
A **speciation** is indicated by a gene mapped to a species **X** while its two immediate descendants (children) are respectively mapped to species in the left subtree and right subtree of **X.**



An example of reconciliation gene **h** defines a speciation event.

Reconciliation: identifying evolutionary events (DL)

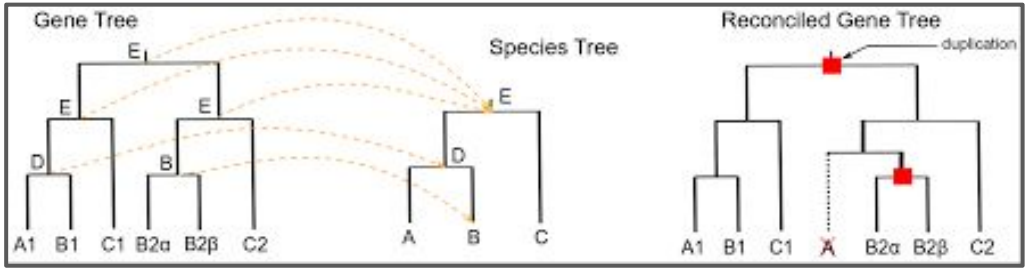
A **duplication** is indicated by a gene mapped to a species **X** and having (1) both of its descendants mapping in clade rooted at **X** and at least one of its immediate descendants mapping to the same species.
A **loss** is associated to any **species** on the path between a gene and one of its immediate descendant.



An example of reconciliation that gives as gene content 2 genes in root species **X**, two genes in **Y** and explains the evolution of **G** with one **gene duplication** at gene **g** and **three gene losses**.

Reconciliation algorithms: the DL model and parsimony

Theorem. There is a unique reconciliation that **minimizes the cumulated number of gene duplications and losses**. It can be computed in linear time using the **LCA algorithm**.



LCA algorithm:
If gene tree node **g** covers genes from species set **S'**, map it to **LCA(S')**.

There can be several reconciliations **minimizing the number of duplications only**, including the LCA reconciliation.These duplication-parsimonious reconciliations can be **sampled** under the uniform distribution in linear time after a **quadratic time dynamic programming** preprocessing (to be seen later).

Reconciliation algorithms: DP for the the DL model

DP algorithm for computing the cost of a parsimonious reconciliation:

C[u,x] = cost of reconciling gene subtree **G_u** with species subtree **S_x**
If **u** is an extant gene from species **x**,
    C[u,x]=0 and **C[u,y]=inf** for any species **y!=x.**
For every ancestral gene **u**, considered in bottom-up order
    For every species **x**, considered in bottom-up order
        C[u,x] = min {
            C[u_1,x_1]+C[u_2,x_2] if **u** is ancestral, **# Speciation**
            C[u_1,x_2]+C[u_2,x_1] if **u** is ancestral, **# Speciation**
            cost_loss + C[u,x_1] if **u** is ancestral, **# Speciation-Loss**
            cost_loss + C[u,x_2] if **u** is ancestral, **# Speciation-Loss**
            cost_dup + C[u_1,x] + C[u_2,x] **# Duplication**
            }

To extract an actual parsimonious reconciliation: backtrack from **C[root(G),root(S)]**.

Reconciliation algorithms: DP for the the DL model

DP algorithm for counting the number of reconciliations:

C[u,x] = cost of reconciling gene subtree **G_u** with species subtree **S_x**
A[u,x] = number of reconciliations between gene subtree **G_u** and species subtree **S_x**
If **u** is an extant gene from species **x**,
    A[u,x]=1 and **A[u,y]=0** for any species **y!=x.**
For every ancestral gene **u**, considered in bottom-up order
    For every species **x**, considered in bottom-up order
        A[u,x]=0
        A[u,x]+=A[u_1,x_1]xA[u_2,x_2]
        A[u,x]+=A[u_1,x_2]xA[u_2,x_1]
        A[u,x]+=A[u,x_1]
        A[u,x]+=A[u,x_2]
        A[u,x]+=A[u_1,x]xA[u_2,x]

The number of reconciliations between **G** and **S** is given by **A[root(G),root(S)]**.

Reconciliation algorithms: DP for the the DL model

DP algorithm for counting the number of parsimonious reconciliations:

C[u,x] = cost of reconciling gene subtree **G_u** with species subtree **S_x**
N[u,x] = number of parsimonious reconciliations between gene subtree **G_u** and species subtree **S_x**
If **u** is an extant gene from species **x**,
    N[u,x]=1 and **N[u,y]=0** for any species **y!=x.**
For every ancestral gene **u**, considered in bottom-up order
    For every species **x**, considered in bottom-up order
        N[u,x]=0
        If **C[u,x]==C[u_1,x_1]+C[u_2,x_2]** then **N[u,x]+=N[u_1,x_1]xN[u_2,x_2]**
        If **C[u,x]==C[u_1,x_2]+C[u_2,x_1]** then **N[u,x]+=N[u_1,x_2]xN[u_2,x_1]**
        If **C[u,x]==C[u_1,x_1]** then **N[u,x]+=N[u,x_1]**
        If **C[u,x]==C[u_1,x_2]** then **N[u,x]+=N[u,x_2]**
        If **C[u,x]==C[u_1,x]+ C[u_2,x]** then **N[u,x]+=N[u_1,x]xN[u_2,x]**

The number of parsimonious reconciliations between **G** and **S** is given by **N[root(G),root(S)]**.

Reconciliation algorithms: DP for the the DL model

DP algorithm for uniform sampling of parsimonious reconciliation:

C[u,x] = cost of reconciling gene subtree **G_u** with species subtree **S_x**
N[u,x] = number of parsimonious reconciliations between gene subtree **G_u** and species subtree **S_x**
To extract an actual parsimonious reconciliation: **stochastic backtrack** from **N[root(G),root(S)]**.

Example 1 of stochastic backtrack: extant genes

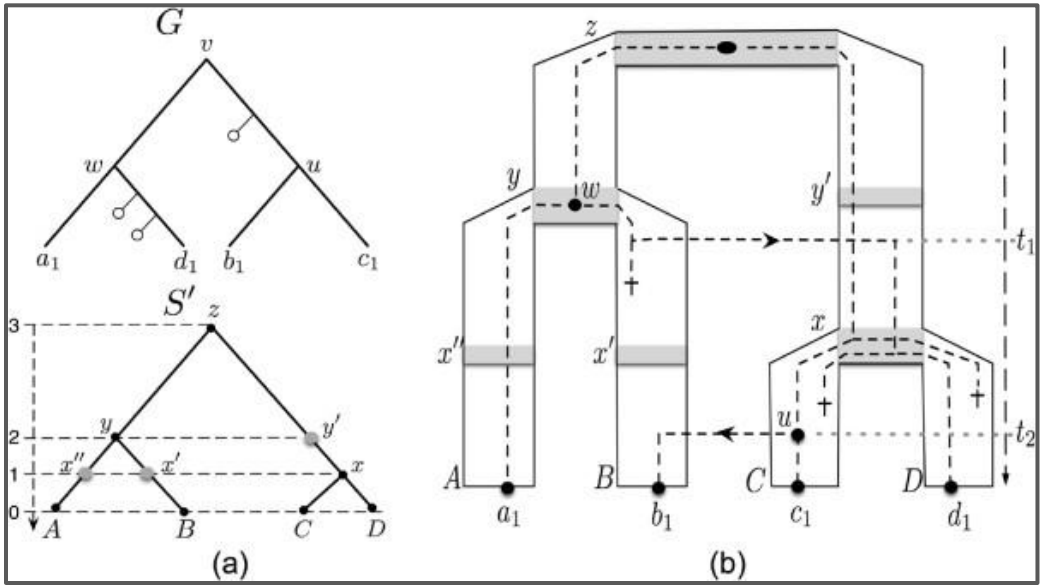
If **u** is an extant gene from species **x**, return (map **u** to **x**)
If **u** is an extant gene from another species than **x**, return (**None**)

Example 2 of stochastic backtrack: ancestral gene

Assume for the sake of exposition that parsimonious cases for **u,x** are
speciation(**u_1,x_1;u_2,x_2**), speciation-loss (**u,x_1**) and duplication
Let **r** be a **random integer** from **[1,N(u,x)]**.
If **r** in **[1,N[u_1,x_1]xN[u_2,x_2]]** return (map **u** to **x**, sp.) + **Sample(u_1,x_1) + Sample(u_2,x_2)**
Elif **r** in **[N[u_1,x_1]xN[u_2,x_2]+1,N[u_1,x_1]xN[u_2,x_2]+N[u,x_1]]** return **Sample(u,x_1)**
Else return (map **u** to **x**, dup) + **Sample(u_1,x) + Sample(u_2,x)**

Reconciliation: identifying evolutionary events (T)

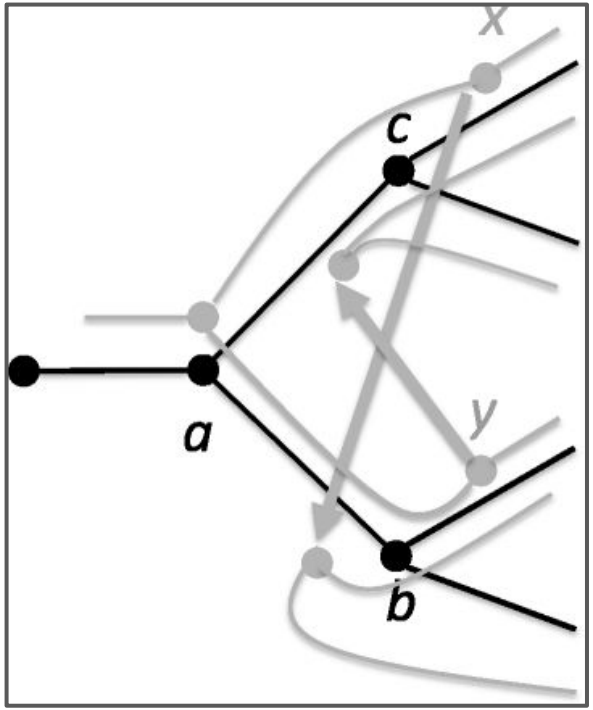
An **Horizontal Gene Transfer (HGT, T)** is defined by a gene **g** mapping to species **X** such that exactly one of its descendants does map to a species which is **not comparable** with **X**, i.e. is neither in the clade rooted at **X** nor an ancestor of **X**. It is similar to a speciation or duplication with one child moving in/sent to another clade in the species tree.



Correction 1.
HGTs can induce gene losses, as duplications: an HGT to a gene **g** mapped to species **X** such that the non-transfered child of **g** is not mapped to **X** induces a gene loss (gene **w** in the figure).

Algorithms: HGT and time-inconsistency

Under the previous definition, we can obtain evolutionary scenarios that are biologically unfeasible as they contain **time-inconsistent HGTs**.



The HGT from gene **x** implies that species **c** existed before species **b**.

The HGT from **y** implies conversely that species **b** existed before species **c**.

This is a time-inconsistency.

Algorithms: DLT-parsimony

Case 1. The given species tree does not contain ranking information.

- ❑ Computing a reconciliation that minimizes the number of duplications, losses and HGTs can be achieved in cubic time, through dynamic programming.

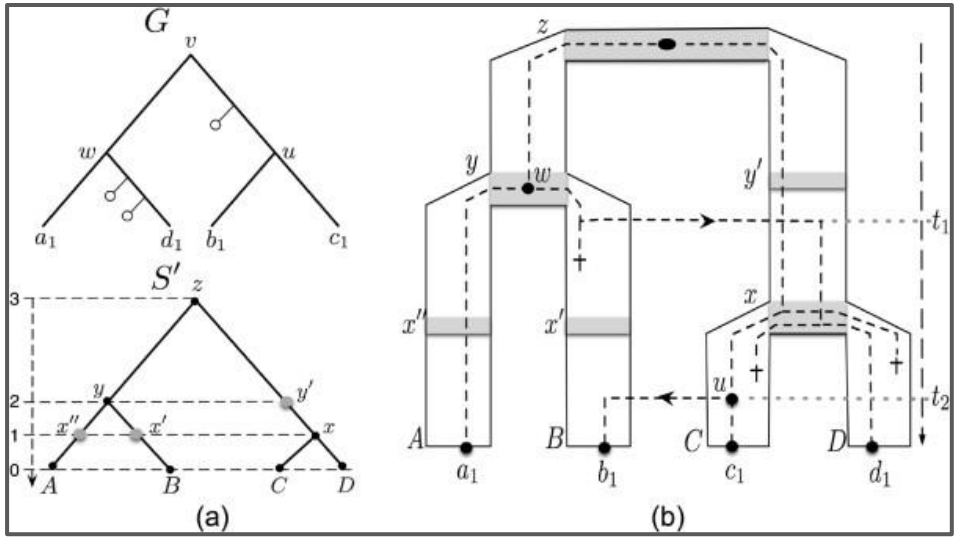
- ❑ Computing a reconciliation that minimizes the number of duplications, losses and HGTs while maintaining time-consistent HGTs is an NP-complete problem.

The difficulty stems from the fact that dynamic programming considers lineages as evolving independently from each others and can not account for the global constraint of time-consistency.

Algorithms: DLT-parsimony

Case 2. The given species tree is ranked.

Computing a reconciliation that minimizes the number of duplications, losses and HGTs and is time-consistent can be achieved in cubic time, through dynamic programming. Parsimonious reconciliations can be sampled under the uniform distribution in linear time.



The key element is that HGTs "**live within a time slice**" of the **extended species tree**, that contains only co-existing species. This ensures time-consistency.

Algorithms: DLT-parsimony (ecceTERA, 2016)

If $t \notin h(x)$:

$$c(u,x,t), \bar{c}(u,x,t) := +\infty$$

Otherwise:

$$c(u,x,t) := \min \begin{cases} \bar{r}(u,x,t) + \lambda \times \theta_{x \neq x^d} & \{\mathbb{TL}\} \\ \bar{c}(u,x,t) & \{\textbf{no } \mathbb{TL} \textbf{ event}\} \end{cases}$$

$$\bar{c}(u,x,t) := \min \begin{cases} c(u_l, x_l, h_1(x_l)) + c(u_r, x_r, h_1(x_r)) & \text{if Binary}(u) \wedge \text{Binary}(x) \wedge t = h_1(x_l) = h_1(x_r) \quad \{\mathbb{S}\} \\ c(u_l, x_r, h_1(x_r)) + c(u_r, x_l, h_1(x_l)) & \\ c(u_l, x, t) + c(u_r, x, t) + \delta \times \theta_{x \neq x^d} & \text{if Binary}(u) \quad \{\mathbb{D}\} \\ c(u_l, x, t) + r(u_r, x, t) & \text{if Binary}(u) \quad \{\mathbb{T}\} \\ r(u_l, x, t) + c(u_r, x, t) & \\ c(u, x, t') & \text{where } t' \text{ is the smallest } t' > t \text{ such that } t' \in h(x) \ \{\textbf{no-event}\} \\ c(u, x_l, h_1(x_l)) + \lambda & \text{if Binary}(u) \wedge \text{Binary}(x) \wedge t = h_1(x_l) = h_1(x_r) \quad \{\mathbb{SL}\} \\ c(u, x_r, h_1(x_r)) + \lambda & \\ 0 & \text{if Leaf}(u) \wedge \text{Leaf}(x) \wedge s(u) = s(x) \quad \{\textbf{extant}\} \\ +\infty & \text{otherwise} \quad \{\textbf{Invalid}\} \end{cases}$$

$$r(u,x,t) = d^{\uparrow}(u,x,t)$$

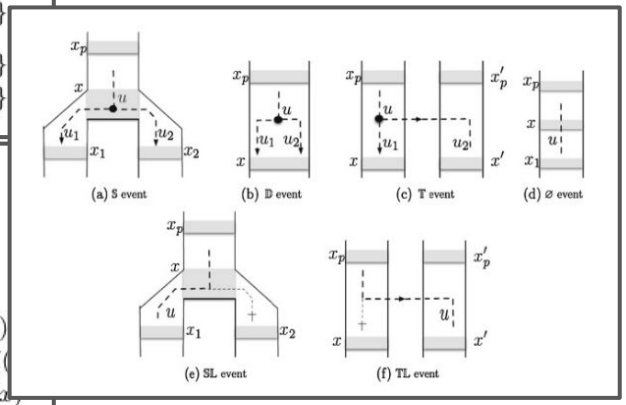
$$d^{\uparrow}(u,x,t) := \min \begin{cases} d^{\uparrow}(u, x_p, t) & \text{if } x \neq \text{Root} \\ d^{\downarrow}(u, x_s, t) & \text{if Binary}(x_p) \end{cases}$$

$$d^{\downarrow}(u,x,t) := \min \begin{cases} c(u,x,t) + \tau \times \theta_{x \neq x^d} & \text{if } t \in h(x) \\ d^{\downarrow}(u, x_l, t) & \text{if not Leaf}(x) \\ d^{\downarrow}(u, x_r, t) & \text{if Binary}(x) \end{cases}$$

$$\bar{r}(u,x,t) = \bar{d}^{\uparrow}(u,x,t)$$

$$\bar{d}^{\uparrow}(u,x,t) := \min \begin{cases} \bar{d}^{\uparrow}(u, x_p, t) & \text{if } x \neq \text{Root} \\ \bar{d}^{\downarrow}(u, x_s, t) & \text{if Binary}(x_p) \end{cases}$$

$$\bar{d}^{\downarrow}(u,x,t) := \min \begin{cases} \bar{c}(u,x,t) + \tau \times \theta_{x \neq x^d} & \text{if } t \in h(x) \\ \bar{d}^{\downarrow}(u, x_l, t) & \text{if not Leaf}(\\ \bar{d}^{\downarrow}(u, x_r, t) & \text{if Binary}(x \end{cases}$$

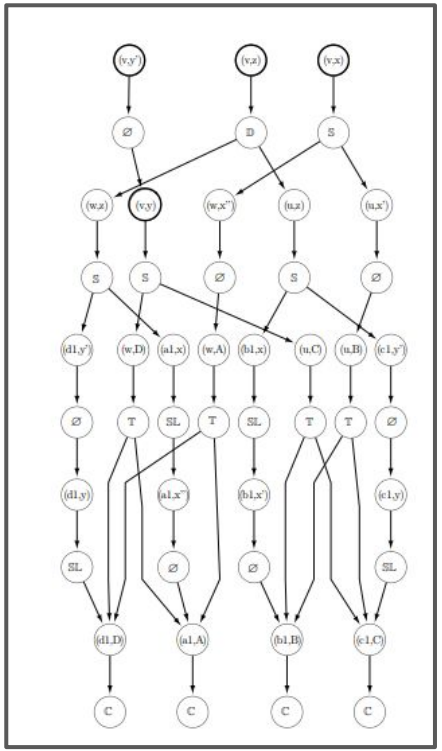


For illustration, the DP algorithm of the reconciliation tool ecceTERA, that can handle partially ranked species trees.

Algorithms: DLT-parsimony

All the algorithms described above are Dynamic Programming algorithms constrained by two trees (the gene tree and the species tree). They satisfy an important property.

- For each possible reconciliation (including non-parsimonious ones), there is **one and exactly one** derivation that can generate it: these algorithms are **complete** and **unambiguous**.



As a consequence once can, with little extra work:

- Count the total number of reconciliations.
- Count the total number of parsimonious reconciliations.
- Sample parsimonious reconciliations under the uniform distribution.
- **Encode compactly all parsimonious reconciliations.**
- **Compute the partition function.**
- **Sample reconciliations under the Boltzmann distribution.**

Algorithms: DLT-parsimony, Boltzmann sampling

Let **r** be a reconciliation of score **c(r)**. Its **Boltzmann score** is

$$b(r) = exp(-c(r)/kT)$$

where **kT** is a constant known as the Boltzmann temperature.

Let **R** be the set of all reconciliations (between a gene tree and a species tree).
The **partition function** of **R** is

$$Z = sum(b(r), for\ r\ in\ R)$$

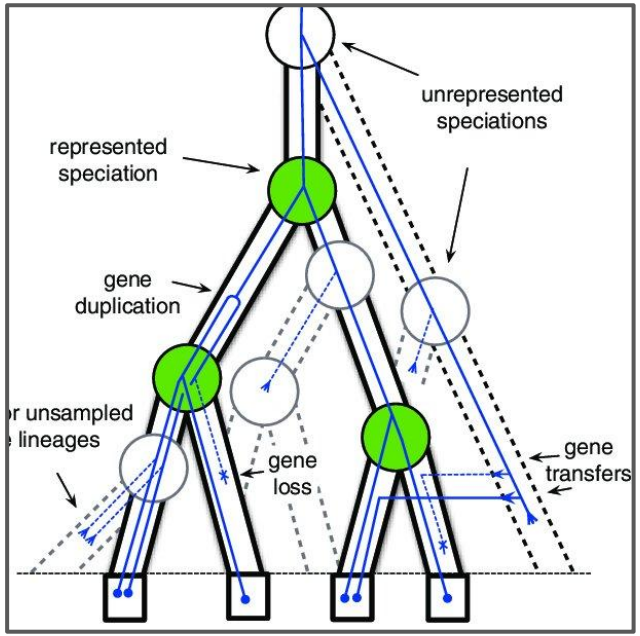
The **Boltzmann probability** of **r** is $b(r)/Z$.

The DP algorithms we did see before can be adapted to

- Compute the partition function for the set of reconciliations between **G** and **S**
- Sample reconciliations between **G** and **S** under the Boltzmann distribution:
  - If kT tends to zero, we sample parsimonious reconciliations uniformly
  - If kT tends to infinity, we sample all reconciliations under the uniform distribution

Algorithms: DLT-parsimony and unsampled species

What if we expect that some genes might appear in or leave a family through an HGT from/to an unsampled species (transfer from the deads)?



We can model this kind of events by adding an outgroup branch to **S** and assume that all HGT from/to the deads are HGTs from / to this branch.

This does not change the algorithms.

Amalgamation: principle

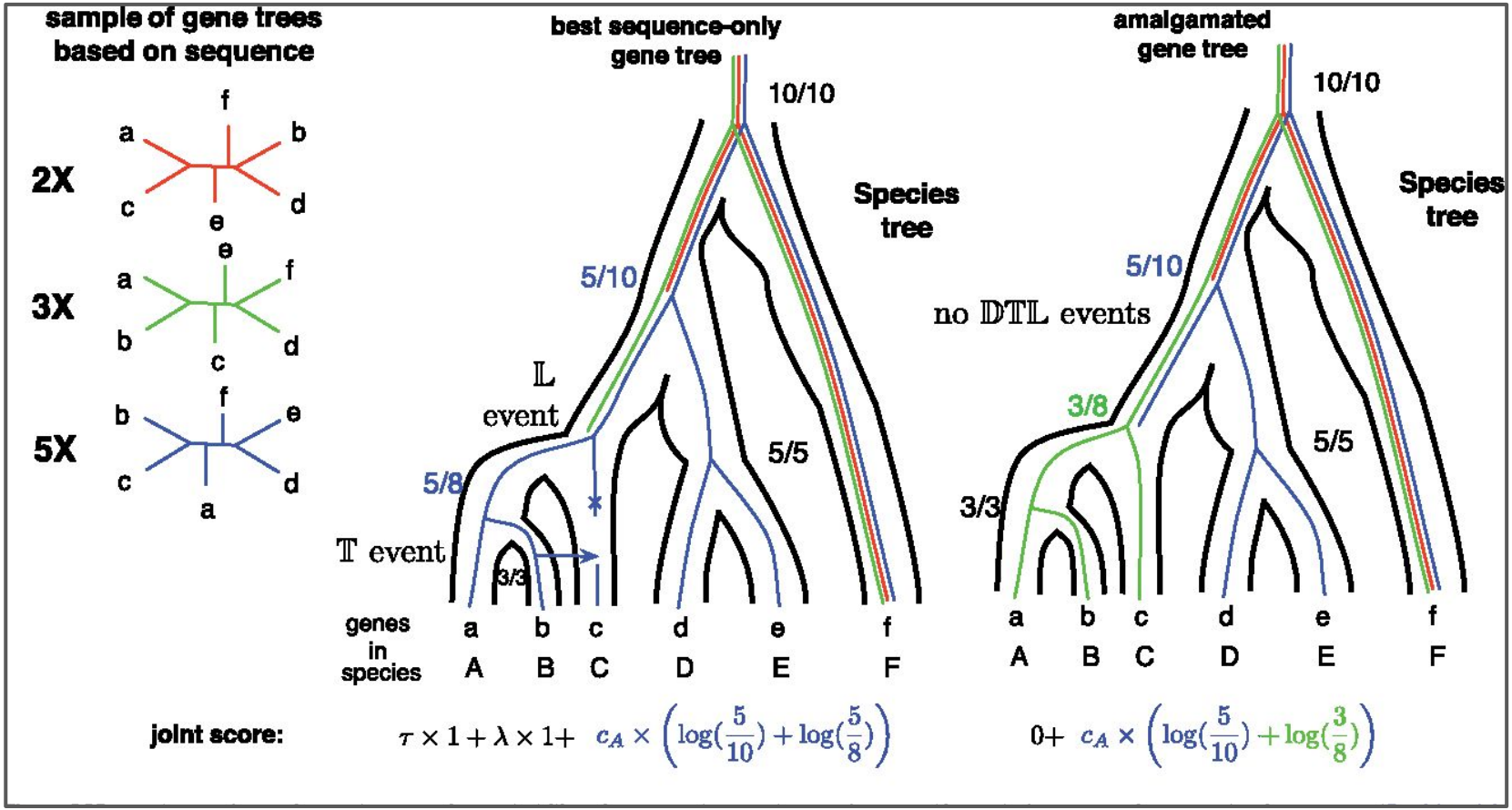
So far we have seen that we can reconcile a given gene tree (computed in general using an algorithm that optimizes some criterion, either parsimony or likelihood, …) and a given species tree.

Could we compute in one step a reconciled gene tree, without having to go through an intermediate step of first computing a gene tree?

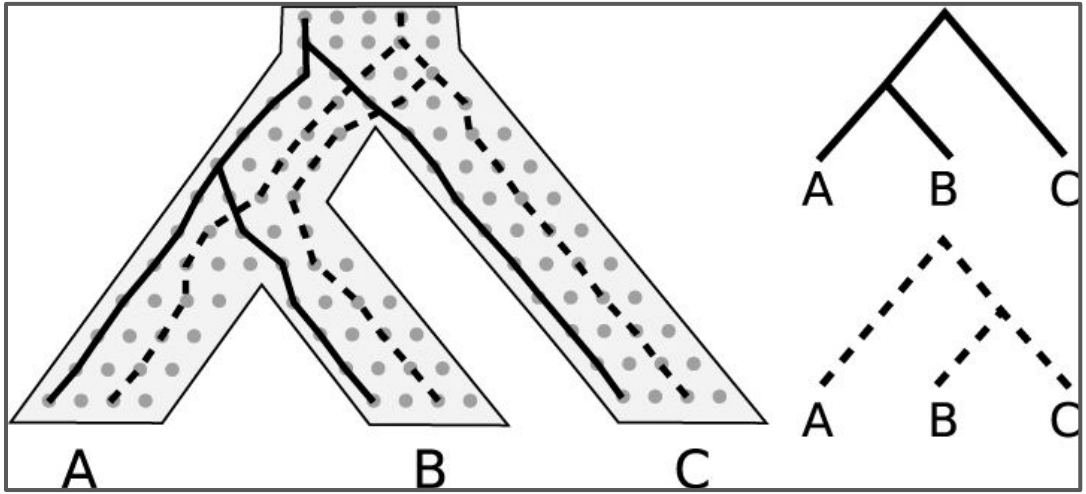
This can be done through the technique of **amalgamation**:

- We first sample unreconciled gene trees (e.g. MrBayes)
- We record the frequency of occurrence of each clade observed in the sampled trees
- We compute a gene tree formed of some of the observed clades that optimizes a linear combination of the a score associated to the clades frequencies and of the reconciliation score.
- This can be integrated to the DP reconciliation algorithms by adding an extra loop over the set of observed clades.

Amalgamation: illustration



Reconciliation: identifying evolutionary events (ILS)



ILS events do not fit well with the mapping technique described above, as extra copies of a gene can exist along a branch while not being due to a gene duplication: ILS is characterized by a speciation with two gene copies living in the same species.

The problem of computing reconciliations while accounting for Duplications, Losses (HGT) and ILS is difficult, and only few algorithms exist at the time, although it is a very active research area.

Overview of existing methods in 2016

Table 1. Comparison of selected features of the most commonly used parsimony reconciliation programs in the DTL model; all considered programs compute a parsimonious DTL reconciliation

Program	A Handles $\mathbb{TL}$ events	B Handles $\mathbb{T}$ from the dead	C (Un) rooted gene trees?	D Performs Amalga-mation	E Rearranges gene trees	F Computes event supports	G Handles ILS	H (Un)dated species trees?	I Only feasible solutions (dated)	J Only feasible solutions (undated)	K All (feasible) co-optimal solutions	L GUI	M Source code
RANGER-DTL 1.0	○	○	R,U	○	○	○	○	F,U	○	○	○	○	○ C++
Notung 2.8	○	○	R,U	○	•	○	•	U	○	•/○	•	•	○ Java
Mowgli	•	•/○	R	○	•	○	○	F	•	○	○	○	• C++
EUCALYPT	○	○	R	○	○	○	○	U	○	•/○	•	○	• Java
AngST	○	○	U	•	○	○	○	F,U	•	○	○	○	• Python
Jane 4	○	○	R	○	○	•	○	F,U,P	○	○	○	•	• Java
ecceTERA	•/○	•/○	R,U	•	○	•	○	F,U,P	•	•/○	•	○	• C++

Advanced topics

- ❑ Probabilistic methods
- ❑ Handling non-binary gene trees or species trees
- ❑ Correcting gene trees using the reconciliation score
- ❑ Handling protein domains evolution
- ❑ Reconciliation of a gene tree with a species network
- ❑ ...