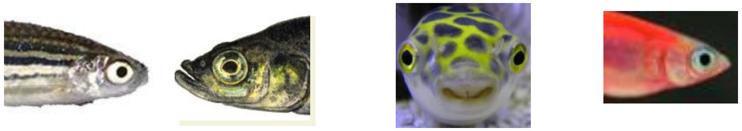
GENE TREE CORRECTION GUIDED BY ORTHOLOGY



Manuel Lafond[1], Magali Semeria[2], Krister M. Swenson[1,4], Eric Tannier[2,3] and Nadia El-Mabrouk[1]

[1]Université de Montréal

[2]Laboratoire de Biometrie et Biologie Evolutive

[3]INRIA Grenoble Rhône-Alpe

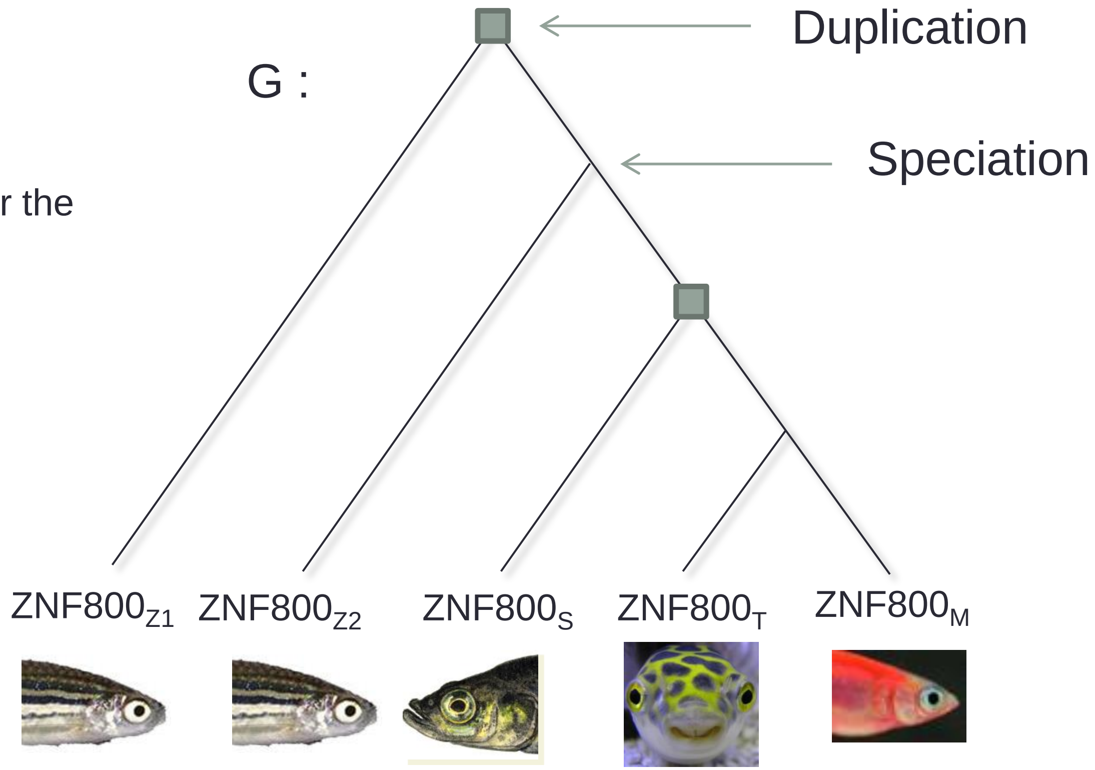
[4]McGill Center for Bioinformatics

Introduction

- **Gene trees** reflect the evolutionary history of a family of **homologous** genes
  - Ancestral genes may have undergone **duplication** or **speciation**

G : Gene tree of the Ensembl ZincFinger protein 800 gene, for the species

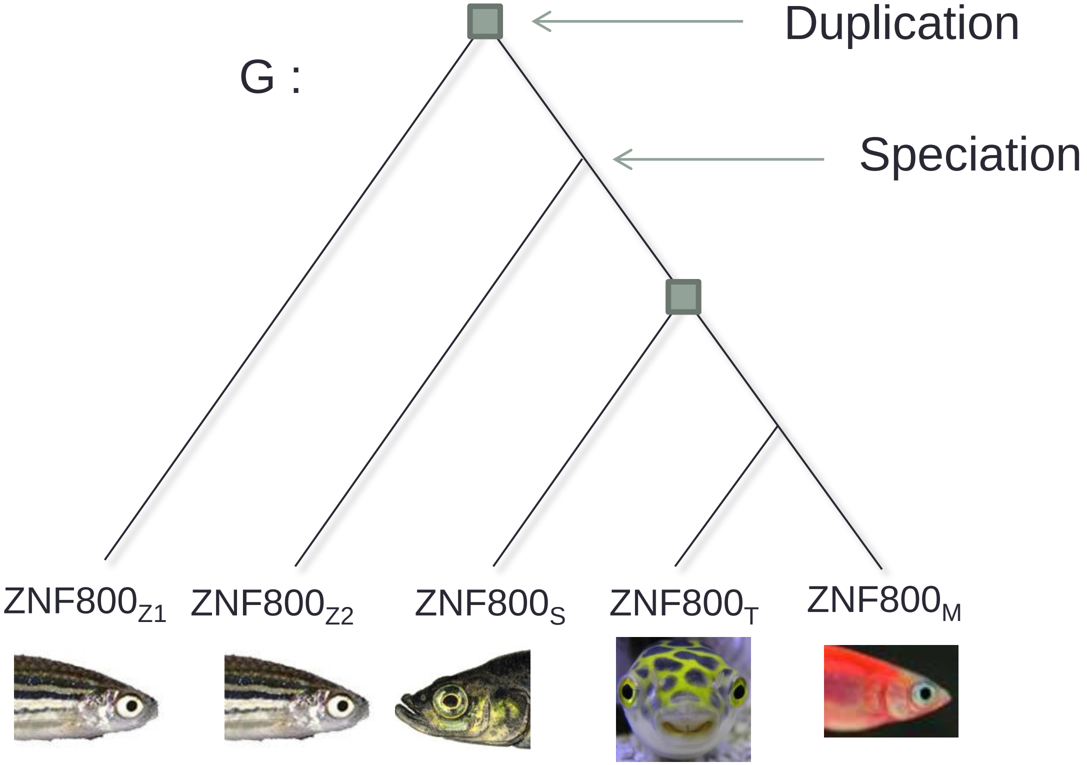
- Zebrafish
- Stickleback
- Medaka
- Tetraodon



Introduction

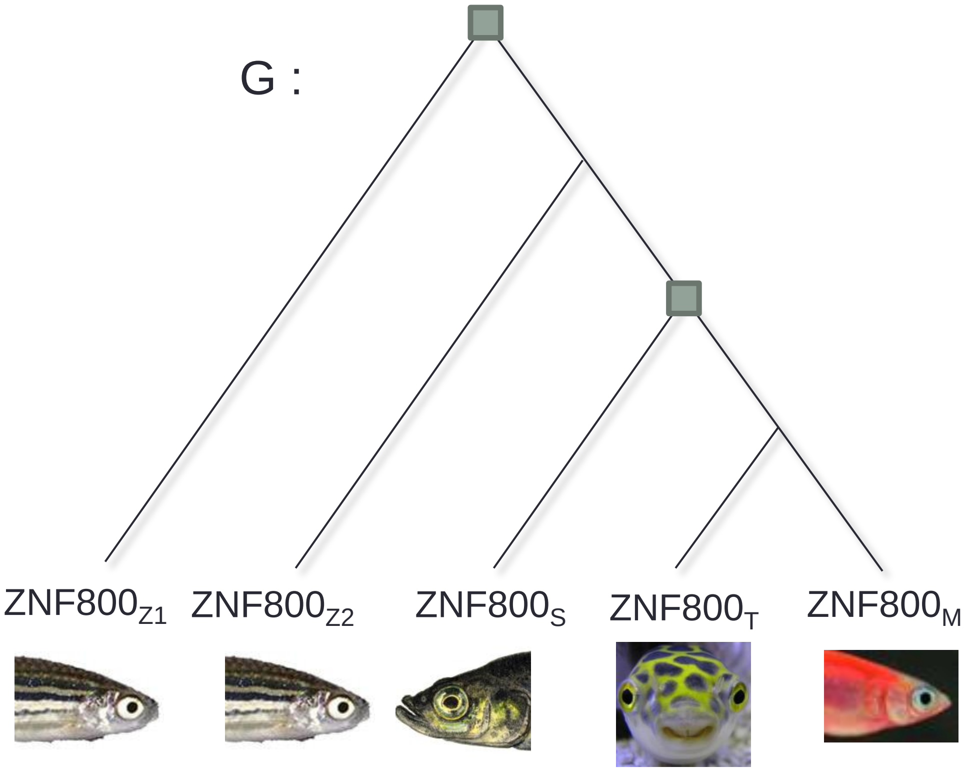
(LCA = Lowest Common Ancestor)

- Pairwise extant genes relationships
  - **Orthologs :** LCA is a speciation (e.g. $ZNF800_{Z2}$, $ZNF800_{T}$)
  - **Paralogs :** LCA is a duplication (e.g. $ZNF800_{Z1}$, $ZNF800_{T}$)

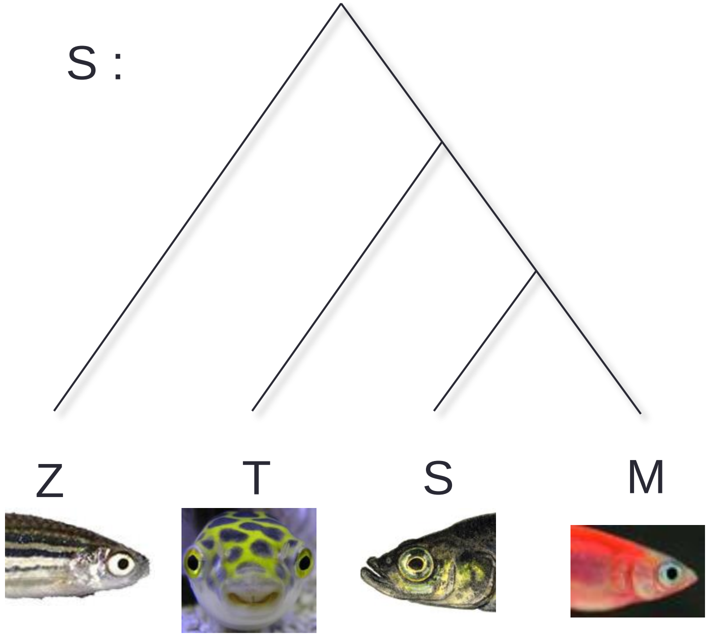


Introduction

- Each gene tree has an associated **species tree**
  - Each extant gene g is mapped to an extant species by a function s(g)

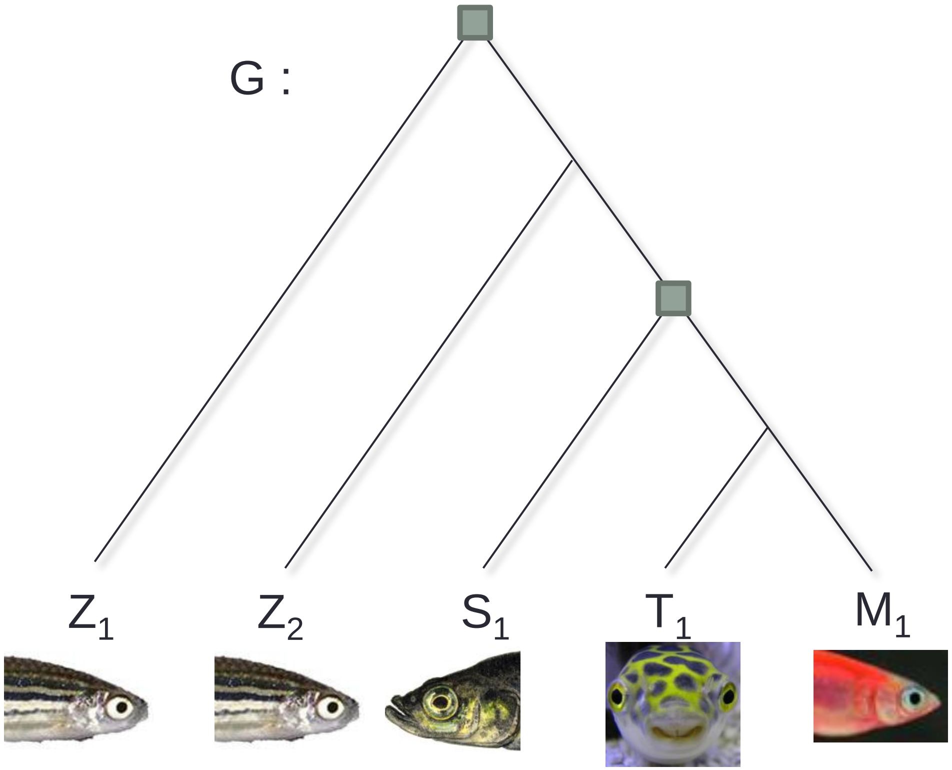


G : Gene tree for the ZincFinger protein 800

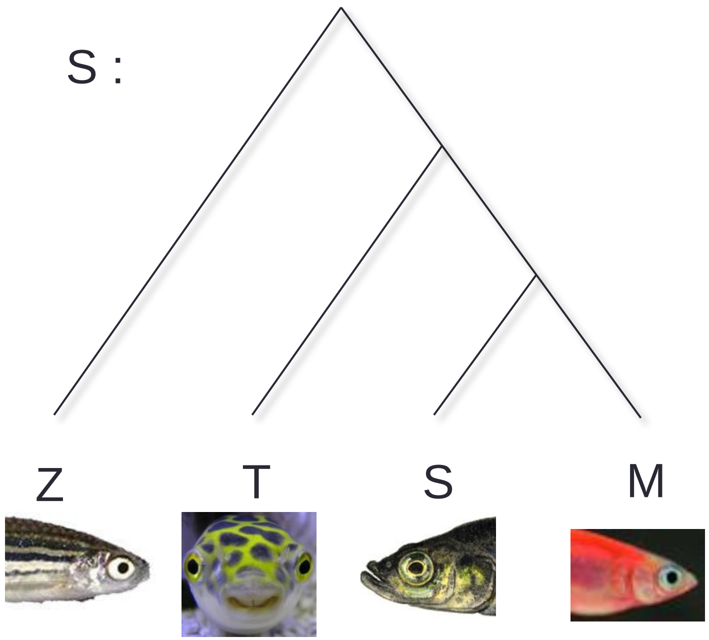


Introduction

- Each gene tree has an associated **species tree**
  - Each extant gene g is mapped to an extant species by a function s(g)
  - We use this mapping to ease up notation

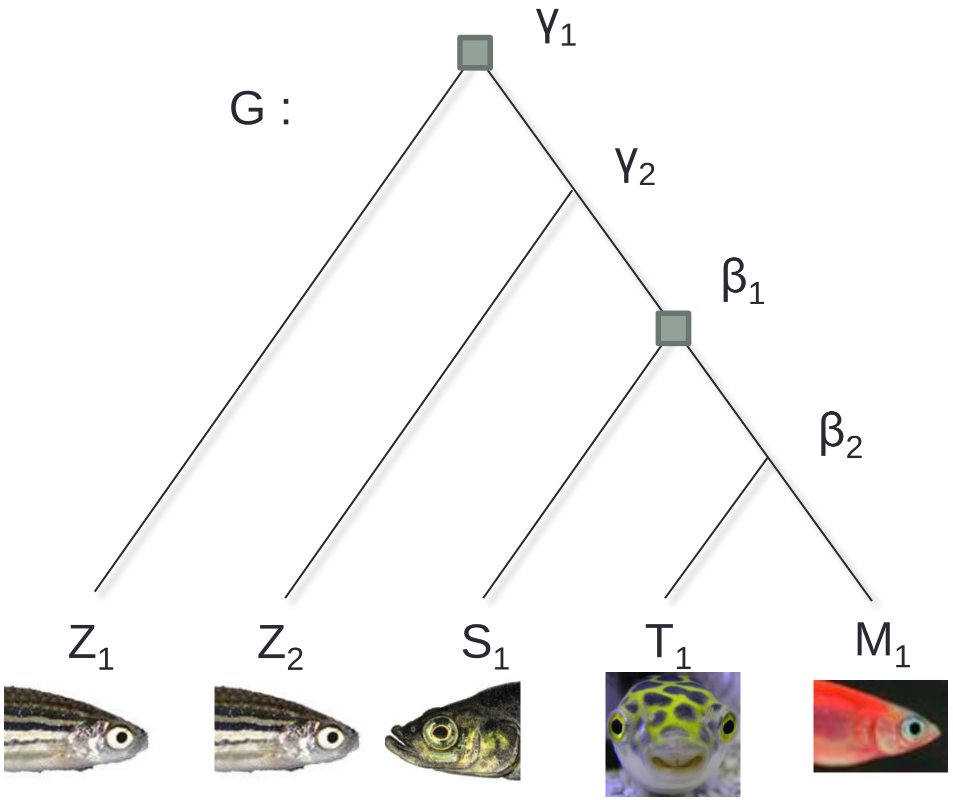


G : Gene tree for the ZincFinger protein 800

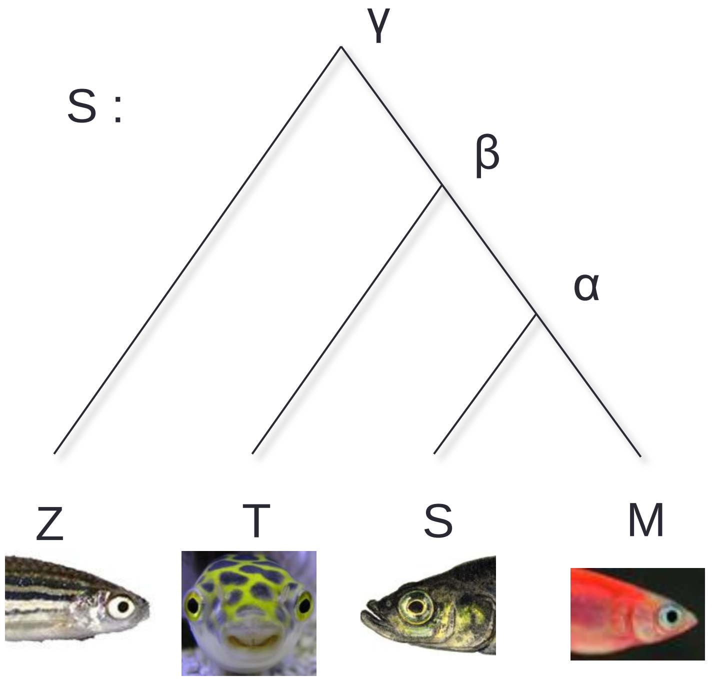


Introduction

- Each gene tree has an associated **species tree**
  - s(g) for ancestral genes : we use **LCA Mapping**, where each ancestral gene is mapped to the LCA of its descendants mappings in S

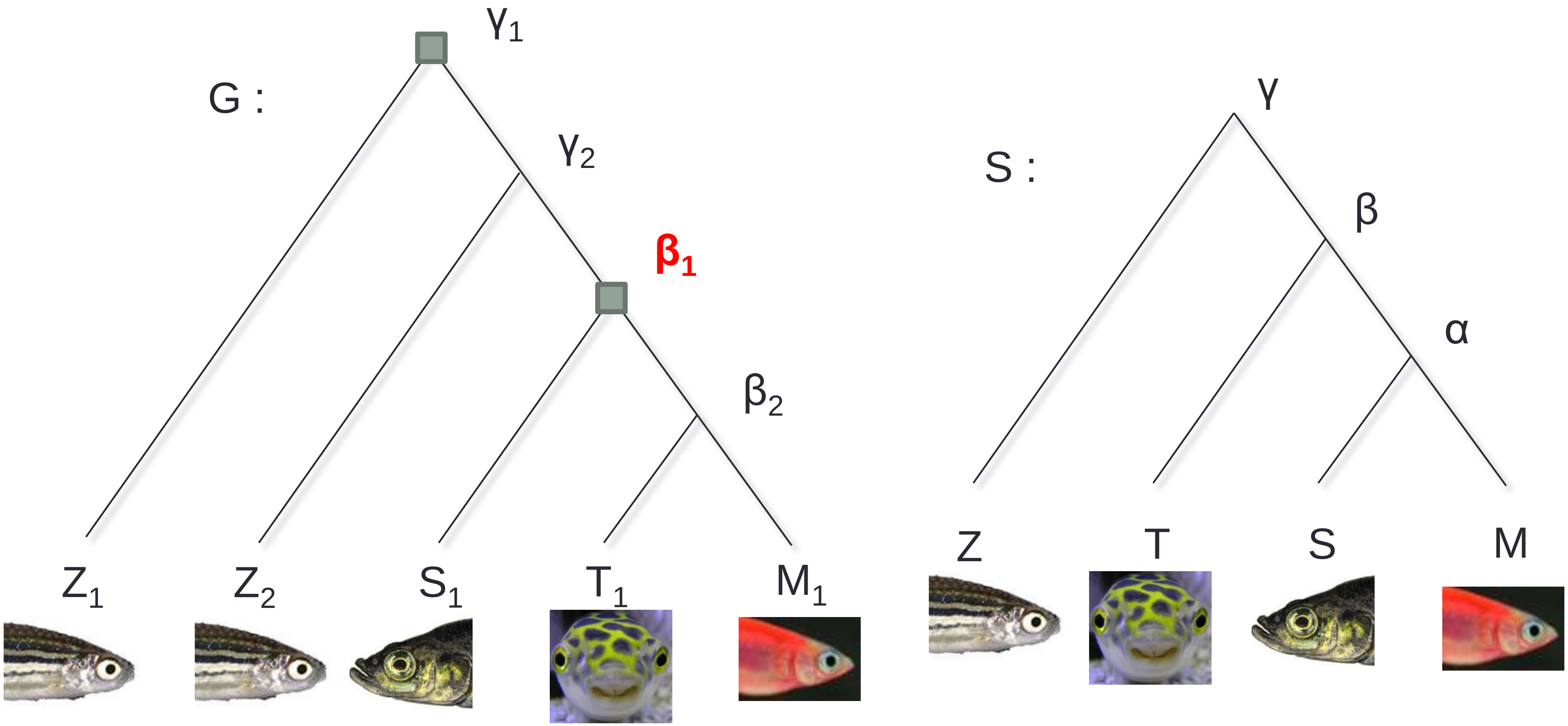


G : Gene tree for the ZincFinger protein 800



Introduction

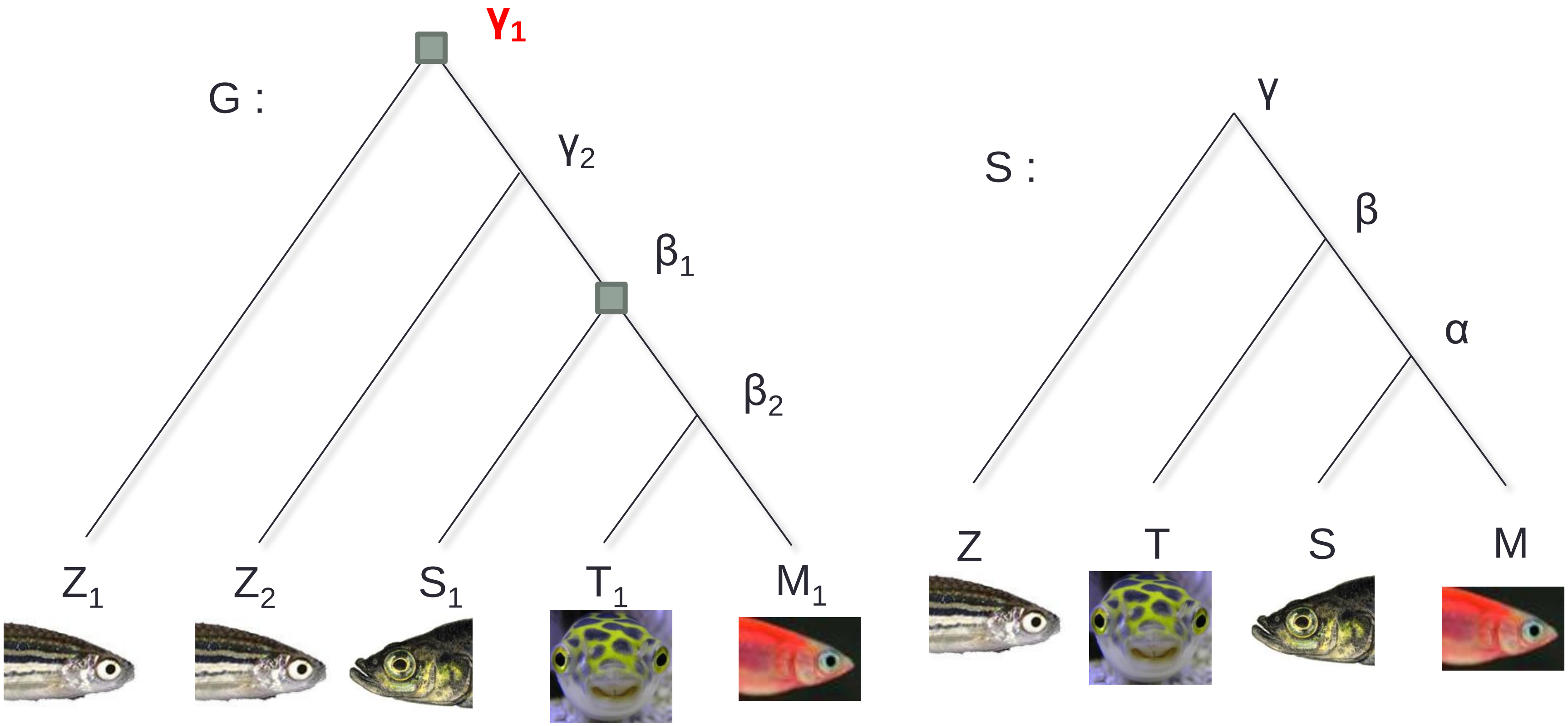
- **Reconciliation** infers speciation/duplication events
  - If g has the same mapping as one of its children, infer a duplication (otherwise, infer a speciation)



G : Gene tree for the ZincFinger protein 800

Introduction

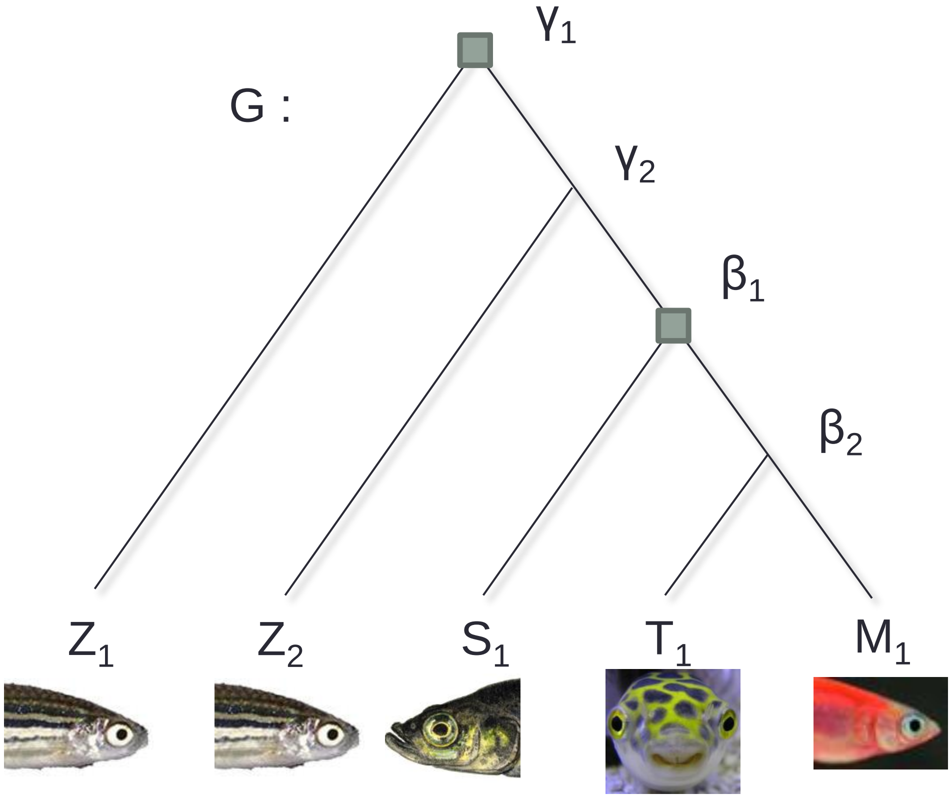
- **Reconciliation** infers speciation/duplication events
  - If g has the same mapping as one of its children, infer a duplication (otherwise, infer a speciation)



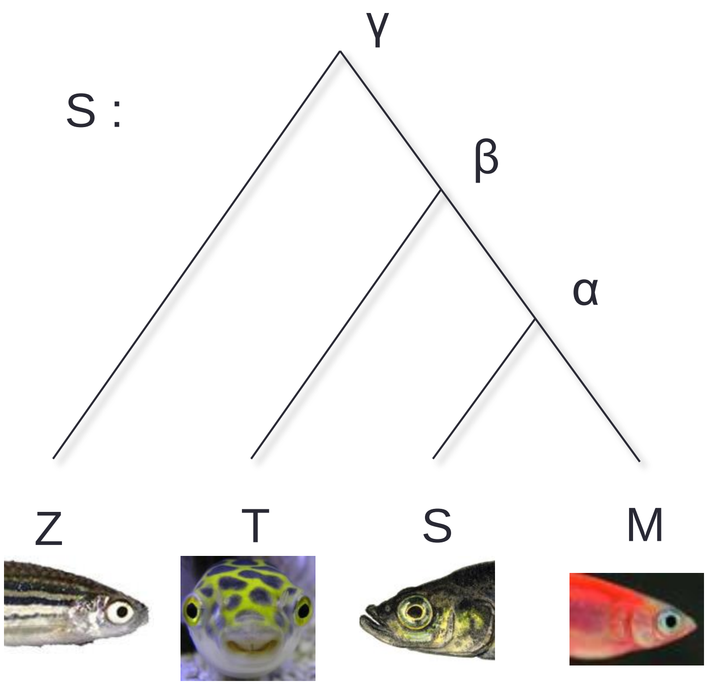
G : Gene tree for the ZincFinger protein 800

Introduction

- **Orthology** and **paralogy** are inferred given the gene tree.
- But instead, can we infer (or correct) parts of the gene tree, **given** orthology/paralogy relationships ?

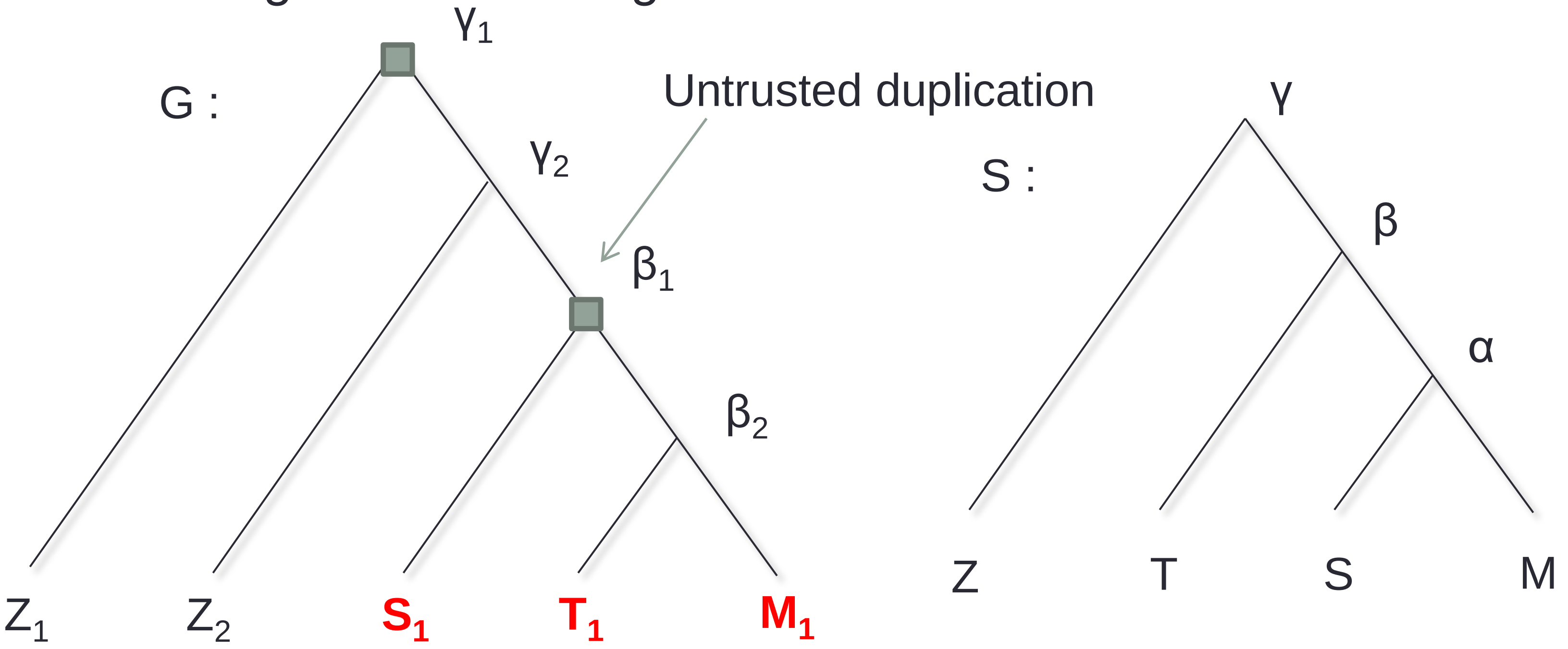


G : Gene tree for the ZincFinger protein 800



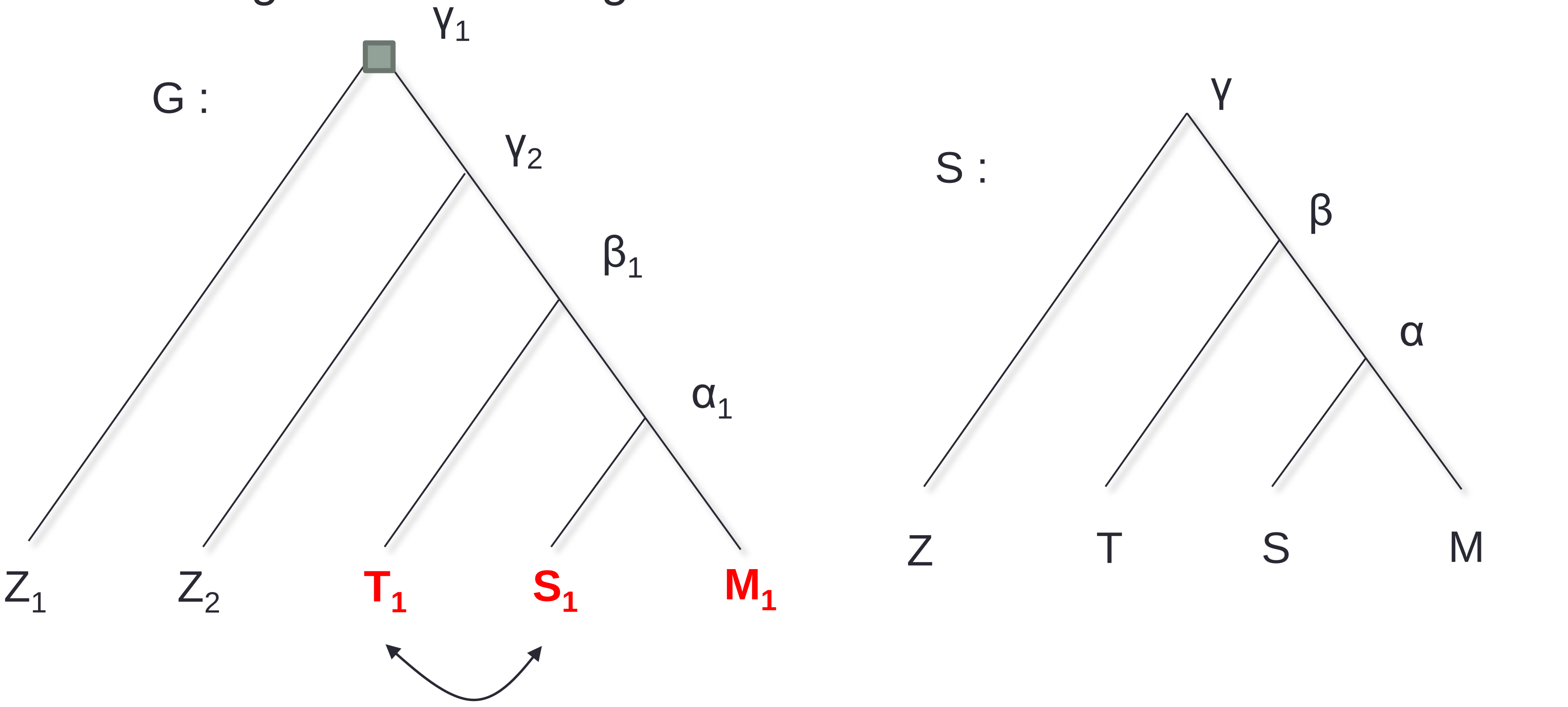
Introduction

- CASE 1 : Suppose we **KNOW** β_1 is a speciation, and we want to keep the β_1 clade (i.e. do not insert/remove leaves in the β_1 subtree)
  - Correct the gene tree making the minimum number of "moves"



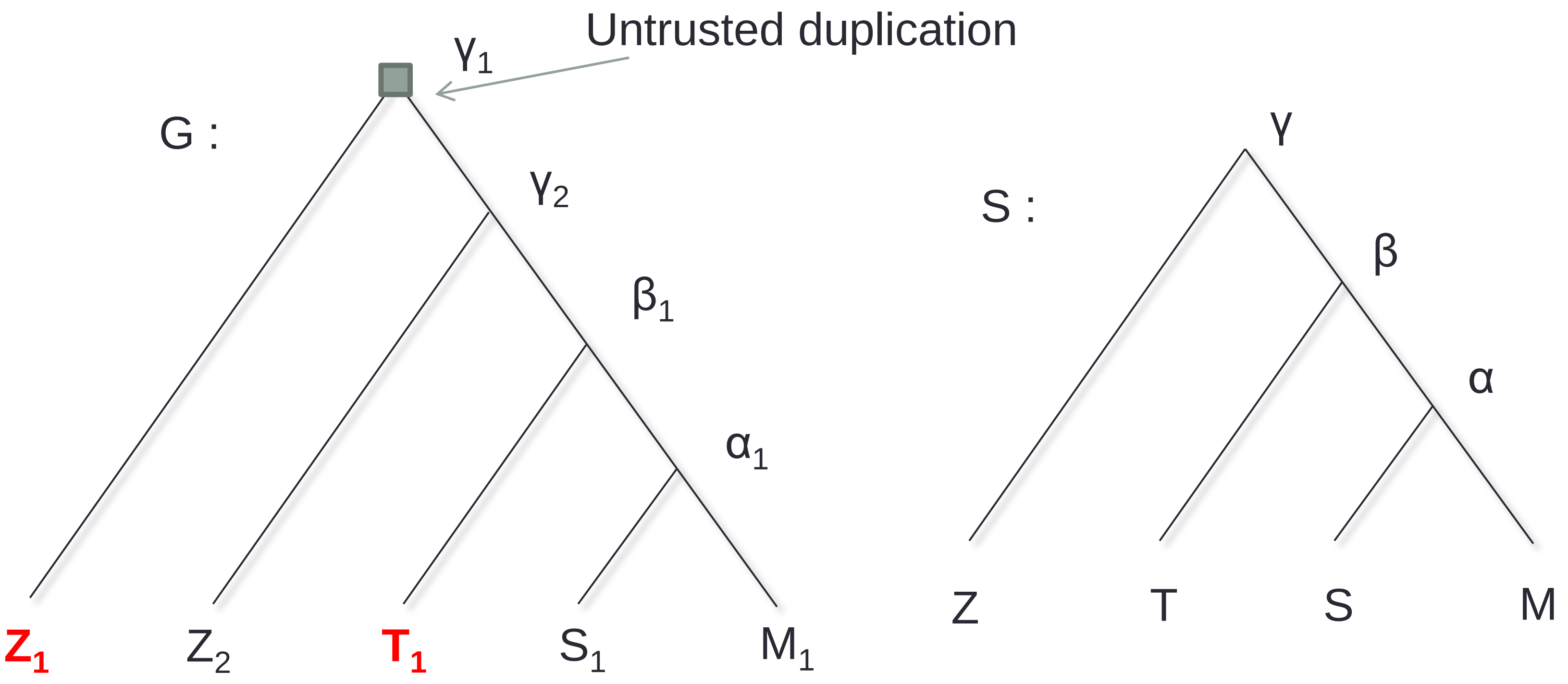
Introduction

- CASE 1 : Suppose we **KNOW** β_1 is a speciation, and we want to keep the β_1 clade (i.e. do not insert/remove leaves in the β_1 subtree)
    - Correct the gene tree making the minimum number of "moves"



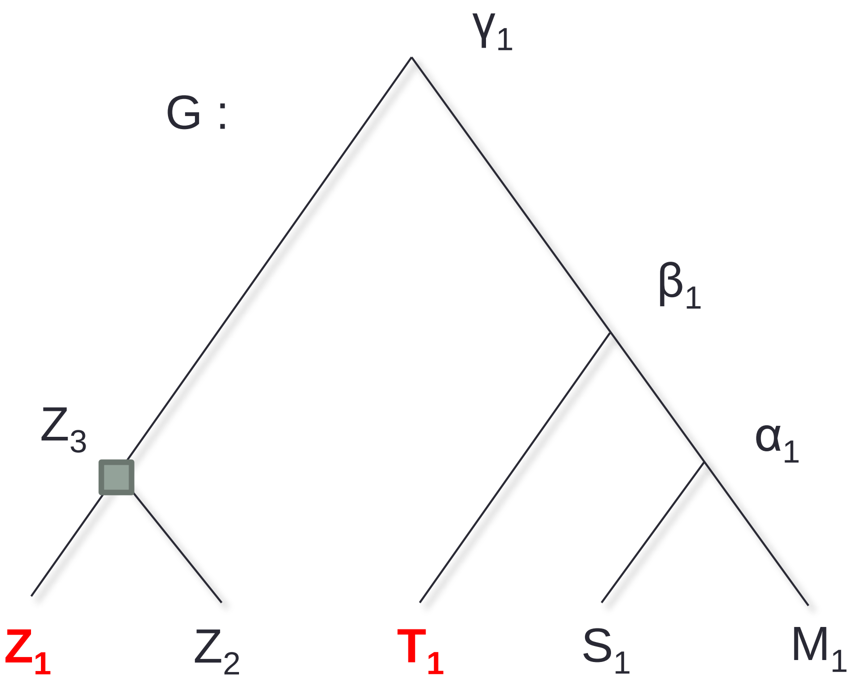
Introduction

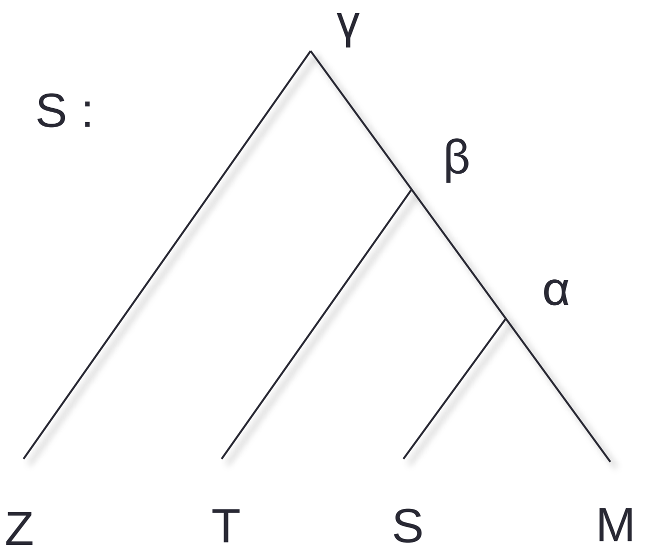
- CASE 2 : Suppose we **KNOW** Z_1 and T_1 are orthologous
  - Correct the gene tree making the minimum of “moves”



Introduction

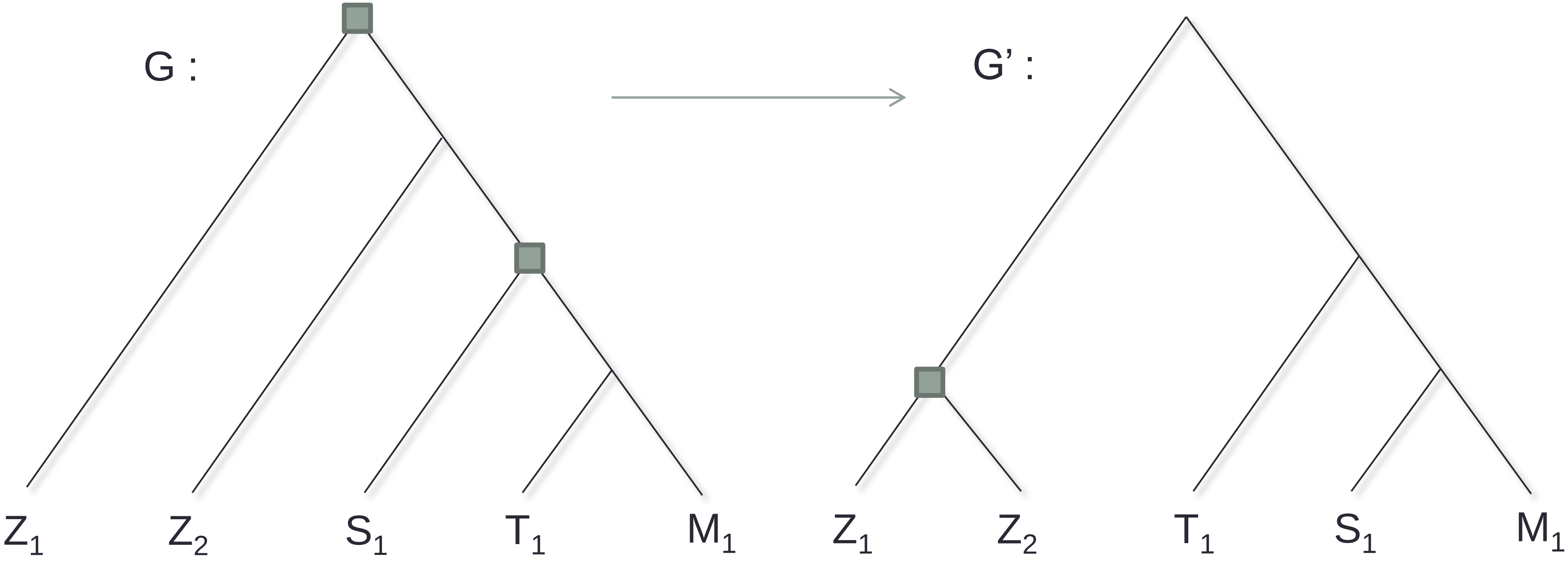
- CASE 2 : Suppose we **KNOW** Z_1 and T_1 are orthologous
  - Correct the gene tree making the minimum of “moves”





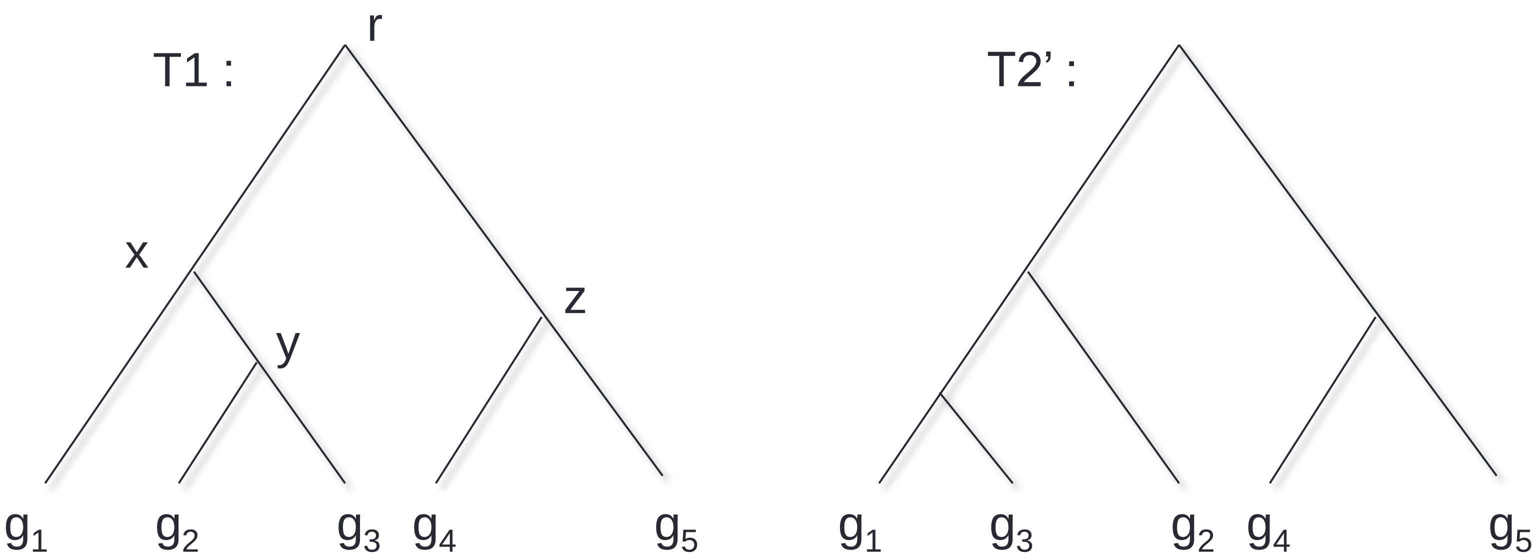
Two correction problems

- Case 1 and 2 give us speciation (orthology) constraints
  - Given G containing untrusted duplications, find a gene tree G' that satisfies the given constraints AND messes up G as least as possible
    - e.g. minimize the Robinson-Foulds distance



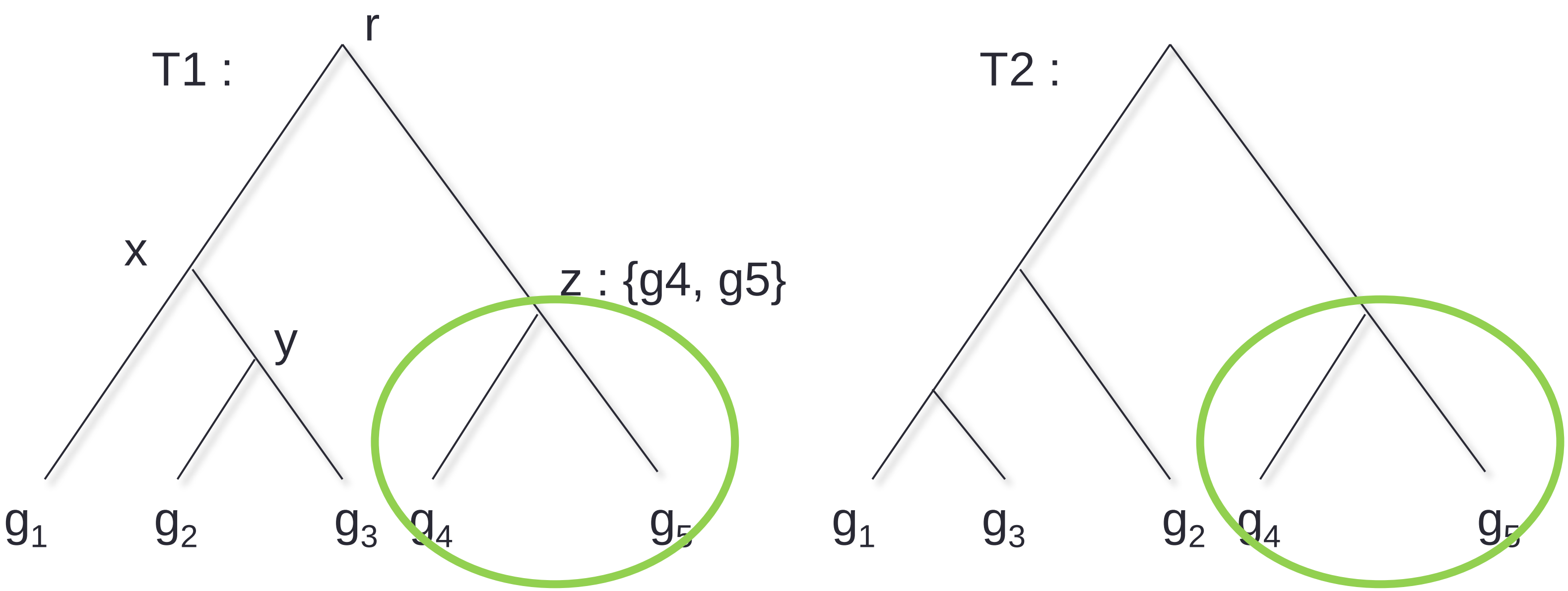
RF distance

- In the case of rooted binary trees T1, T2 with the same leaves :
- RFDist(T1, T2) is simply two times the number of **clades** in T1, but not in T2



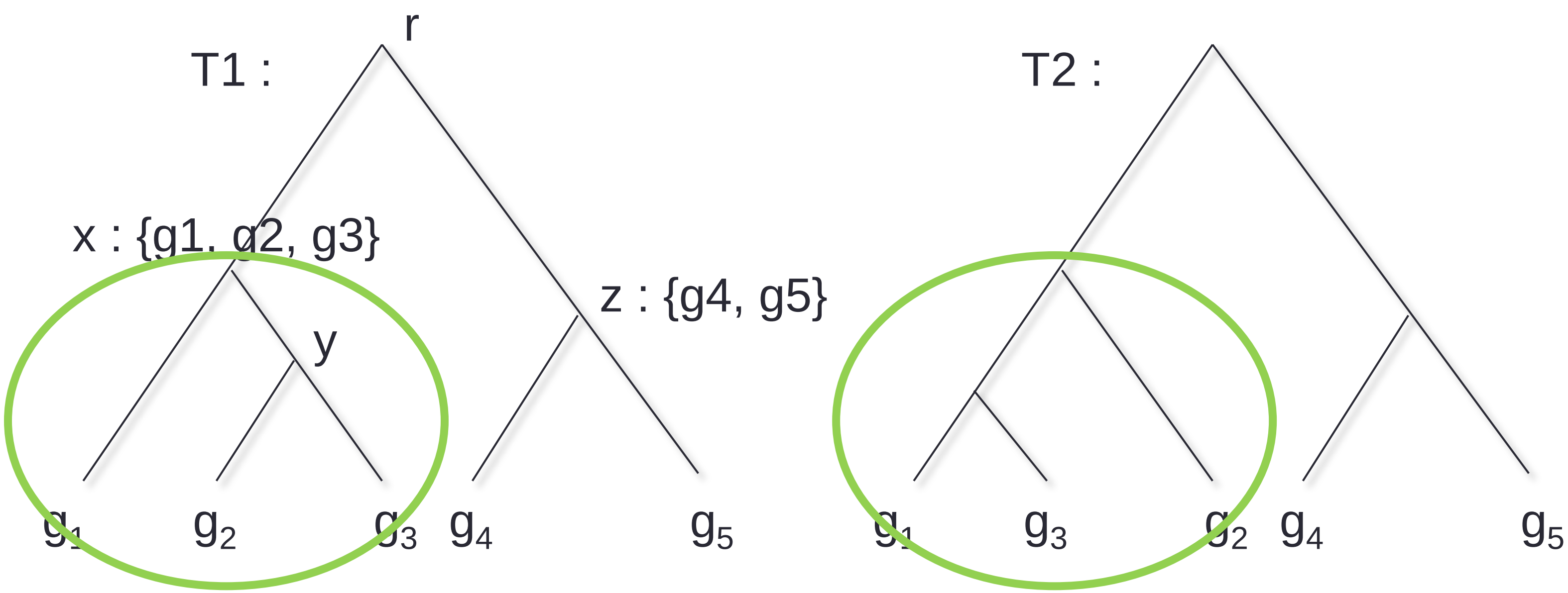
RF distance

- In the case of rooted binary trees T1, T2 with the same leaves :
- RFDist(T1, T2) is simply two times the number of **clades** in T1, but not in T2



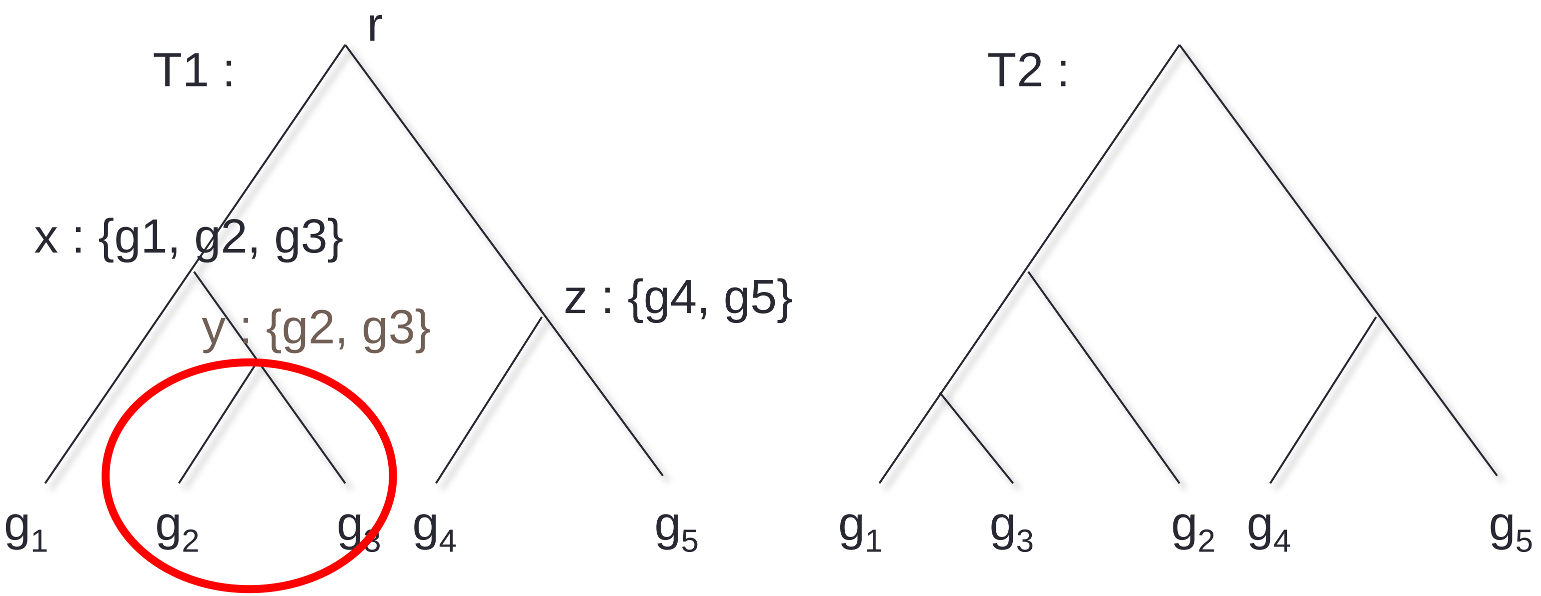
RF distance

- In the case of rooted binary trees T1, T2 with the same leaves :
- RFDist(T1, T2) is simply two times the number of **clades** in T1, but not in T2



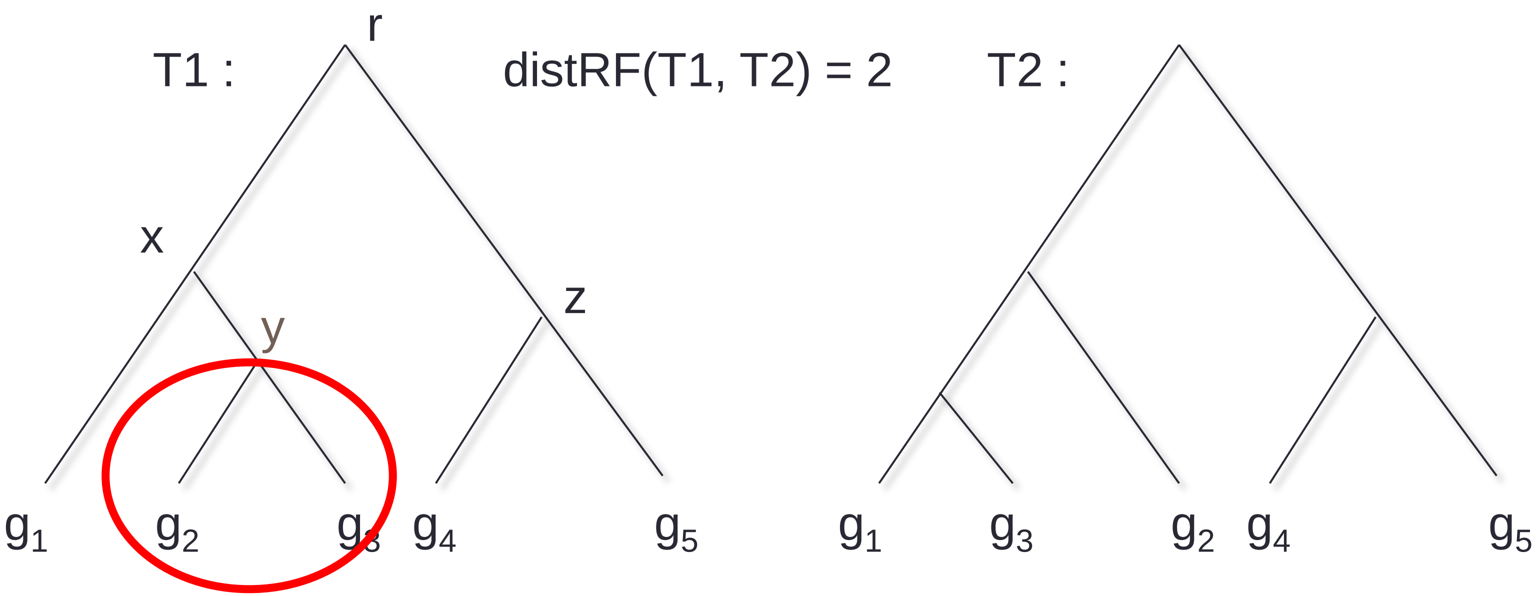
RF distance

- In the case of rooted binary trees T1, T2 with the same leaves :
- RFDist(T1, T2) is simply two times the number of **clades** in T1, but not in T2



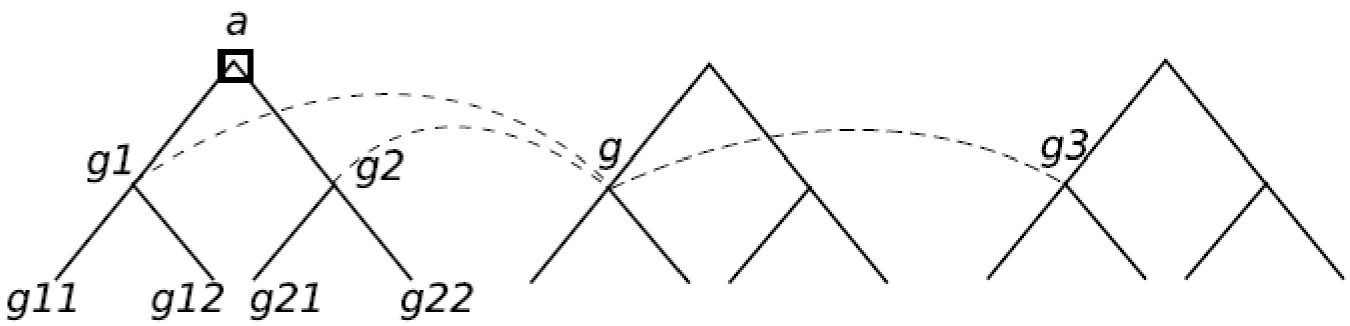
RF distance

- In the case of rooted binary trees T1, T2 with the same leaves :
- RFDist(T1, T2) is simply two times the number of **clades** in T1, but not in T2



Detecting untrustworthy duplications

- Some duplications are labeled “dubious” or given low confidence values by Ensembl
- We can use synteny to infer orthology/paralogy relationships [1]
- Software inferring ancestral adjacencies might pick up erroneous duplications
    - Using DeCo, one can identify bad duplications when more than two adjacencies are inferred on an ancestral gene [2]

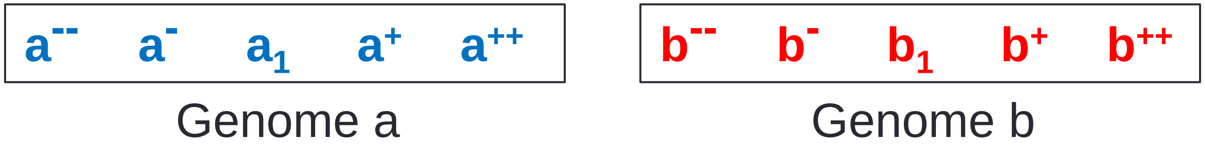


[1] Lafond, Swenson, El-Mabrouk, “Error detection and correction of gene trees”, MASGE (2013)

[2] Chauve, El-Mabrouk, Guéguen, Semeria, Tannier, “Duplication, Rearrangement and Reconciliation: A Follow-Up 13 Years Later“, MAGE (2013)

Detecting untrustworthy duplications

- Suppose genes a1, b1 from genomes a and b are in syntenic blocks (they are in a conserved region of homologous genes)
  - In this example, a conserved region involving 5 genes families



a^{--} a^{-} a_1 a^{+} a^{++}

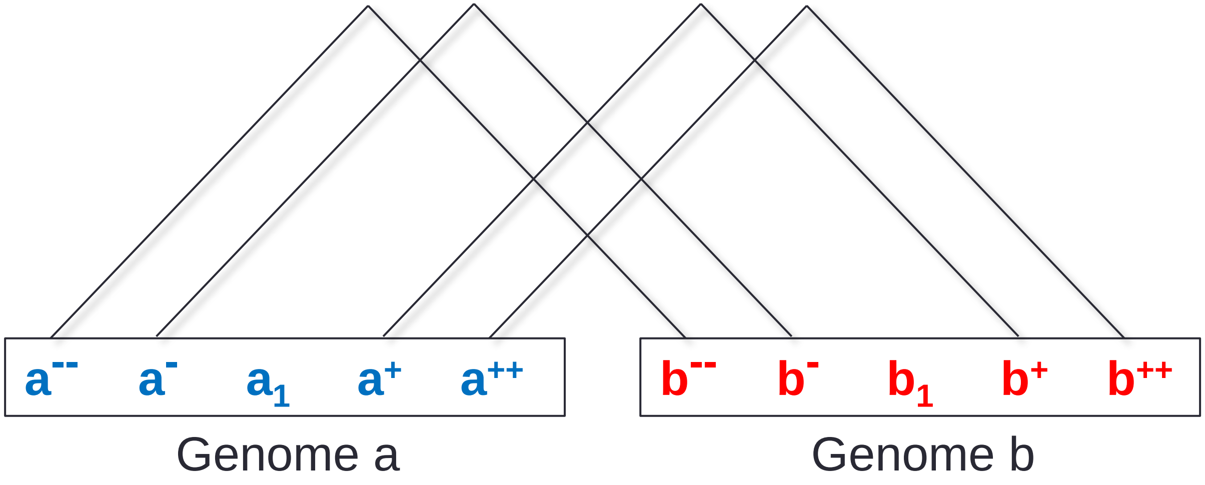
Genome a

b^{--} b^{-} b_1 b^{+} b^{++}

Genome b

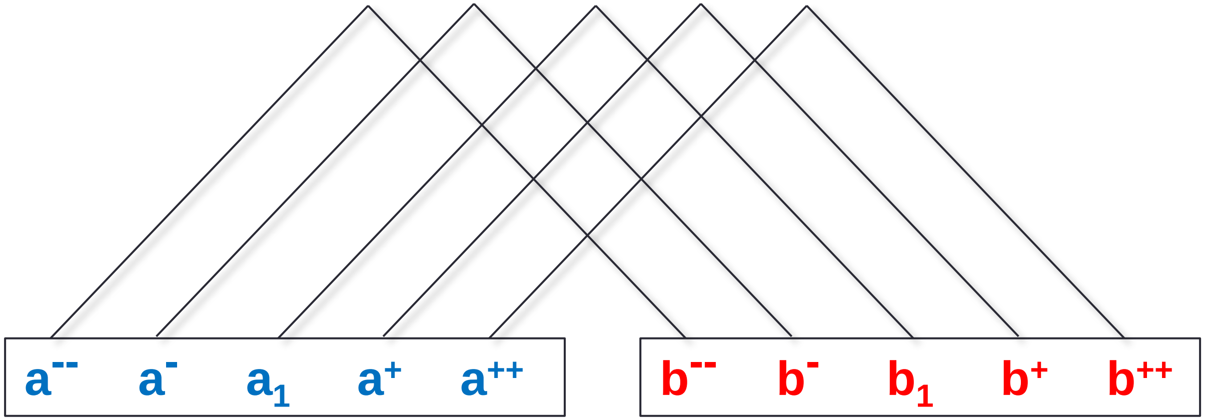
Detecting untrustworthy duplications

- Suppose genes a1, b1 from genomes a and b are in syntenic blocks (they are in a conserved region of homologous genes)
  - In this example, a conserved region involving 5 genes families
- Look at the gene trees of each involved family



Detecting untrustworthy duplications

- If all the homologous genes in the regions are orthologous, we expect a_1 and b_1 to also be orthologous



a^{--} a^{-} a_1 a^{+} a^{++}

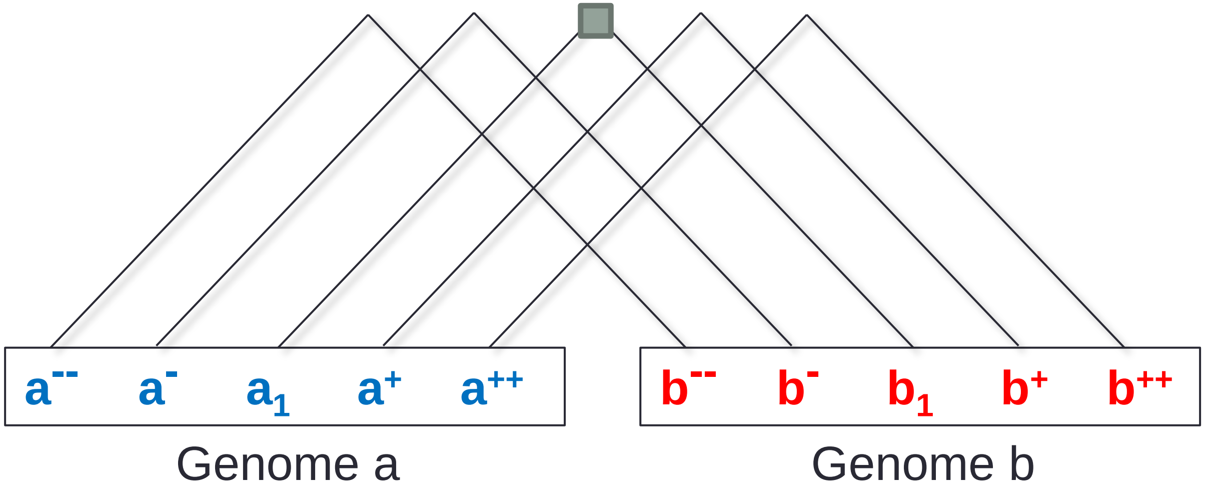
b^{--} b^{-} b_1 b^{+} b^{++}

Genome a

Genome b

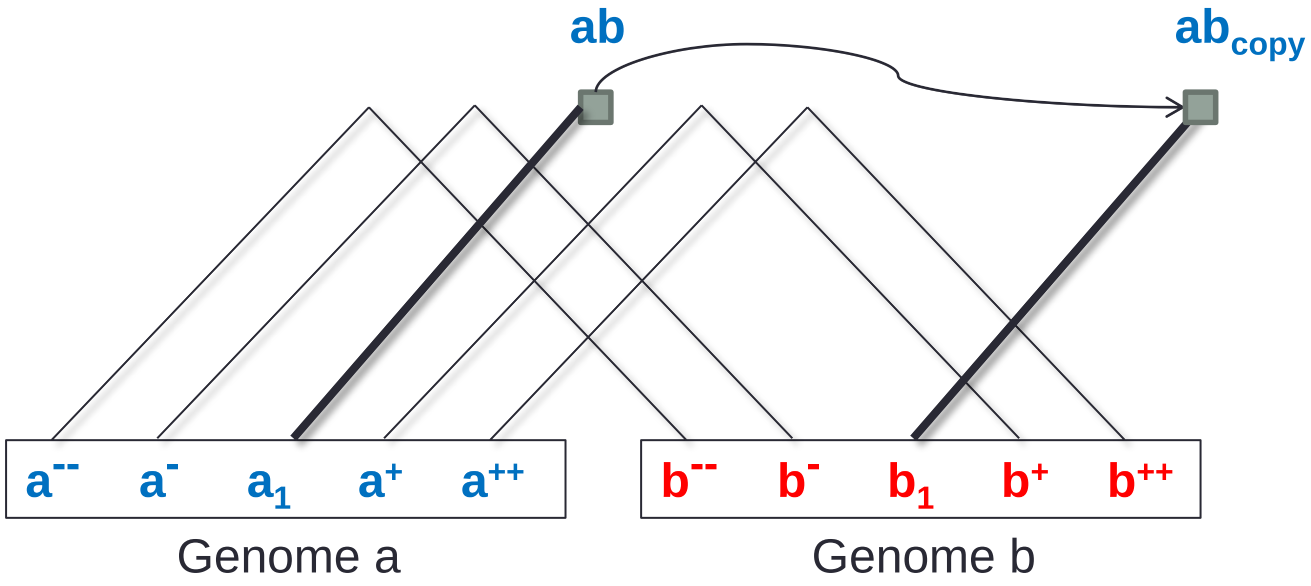
Detecting untrustworthy duplications

- If all the homologous genes in the regions are orthologous, we expect a_1 and b_1 to also be orthologous
  - If not, some unlikely event occurred



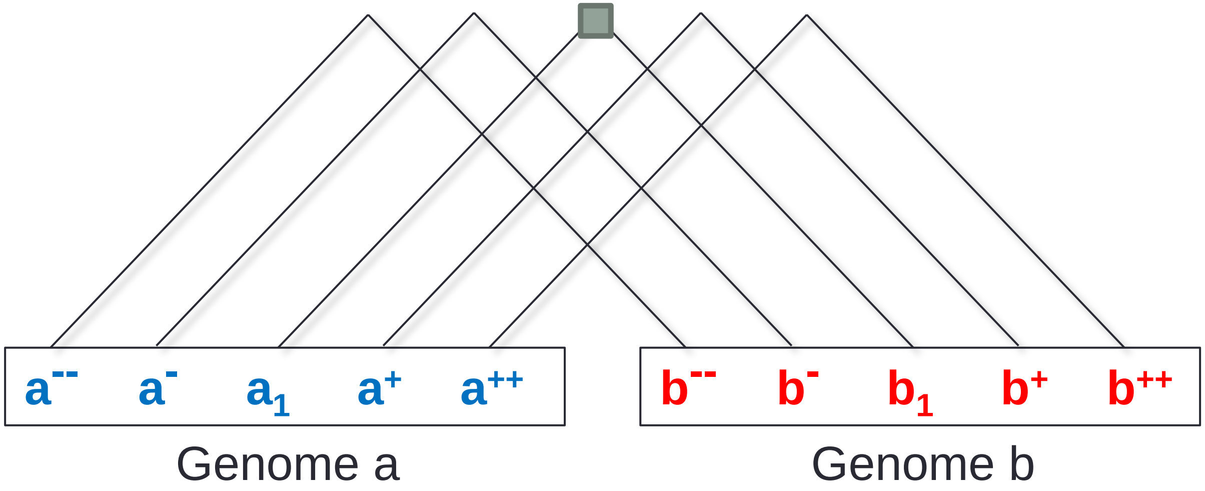
Detecting untrustworthy duplications

- What’s wrong with this ?
  - If only the ancestral gene ab duplicated, the copy typically went somewhere else on the ancestral genome
  - And somehow, it ended up in a region similar to the original gene…mostly by **chance**.



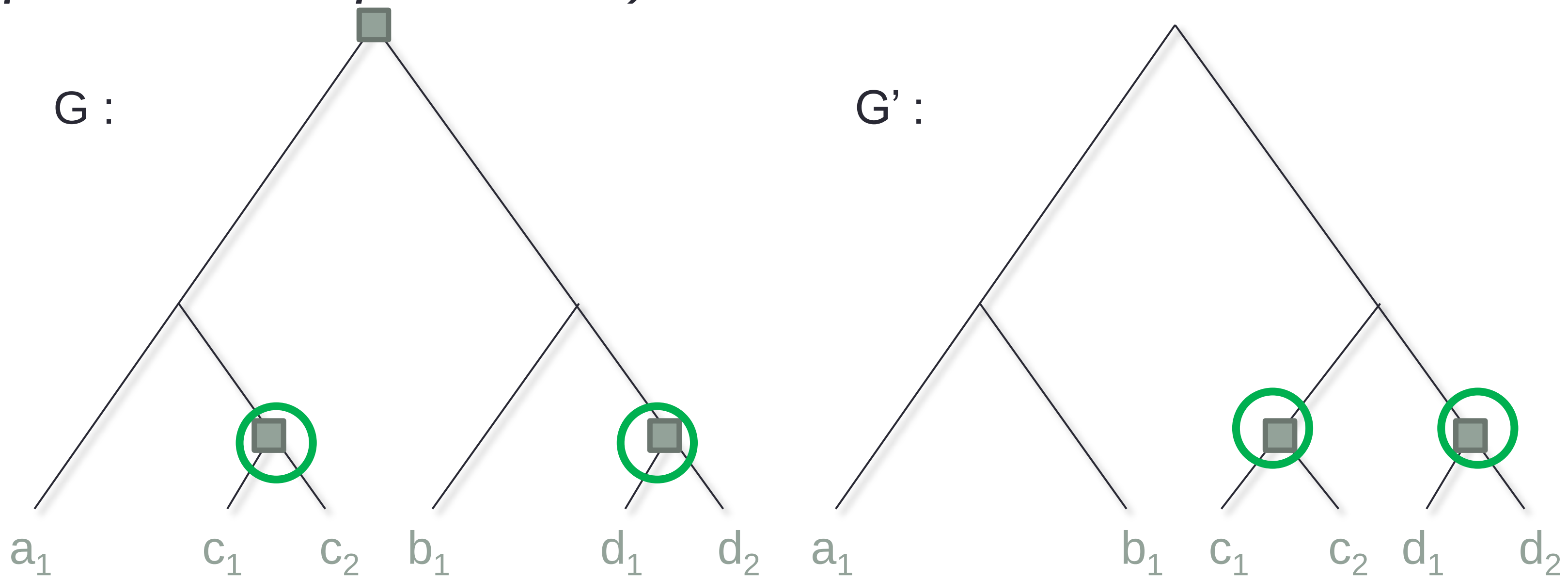
Detecting untrustworthy duplications

- We looked at ~6000 Ensembl gene trees
  - The trees for the Zebrafish, Medaka, Tetraodon and Stickleback species
- 22% (~1200) of these trees contained this type of bad duplication



Problem 1

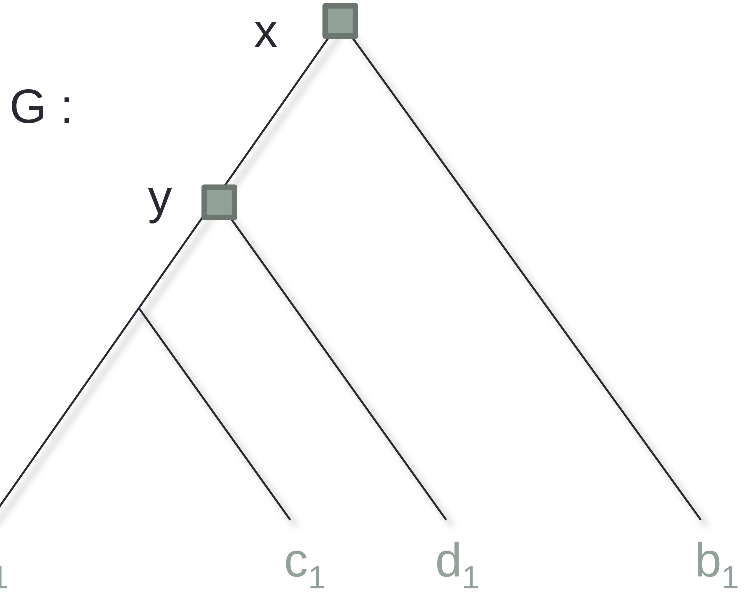
- **Given**: given a gene tree G, a species tree S, and a set C of clades that are required to be speciations
- **Find** : A corrected gene tree G' in which all clades in C are preserved, are speciations, and such that RFDist(G, G') is minimized *(as many clades as possible are preserved)*



In green : preserved clades

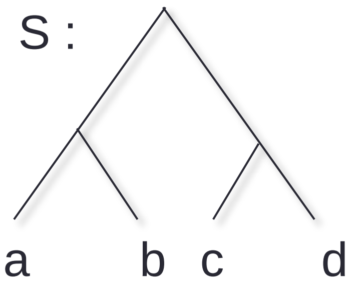
Problem 1

- A solution doesn’t always exist
  - In this example, if C = {x,y}, we cannot correct both x and y into speciations
    - A solution exists iff for any two x, y in C, we don’t have that x is an ancestor of y and s(x) = x(y)
  - We will assume there exists a solution



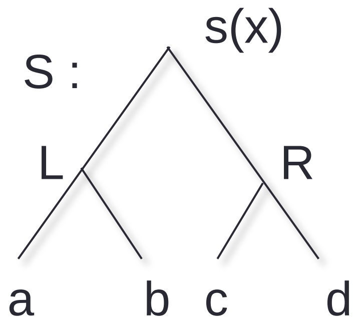
C = {x, y}

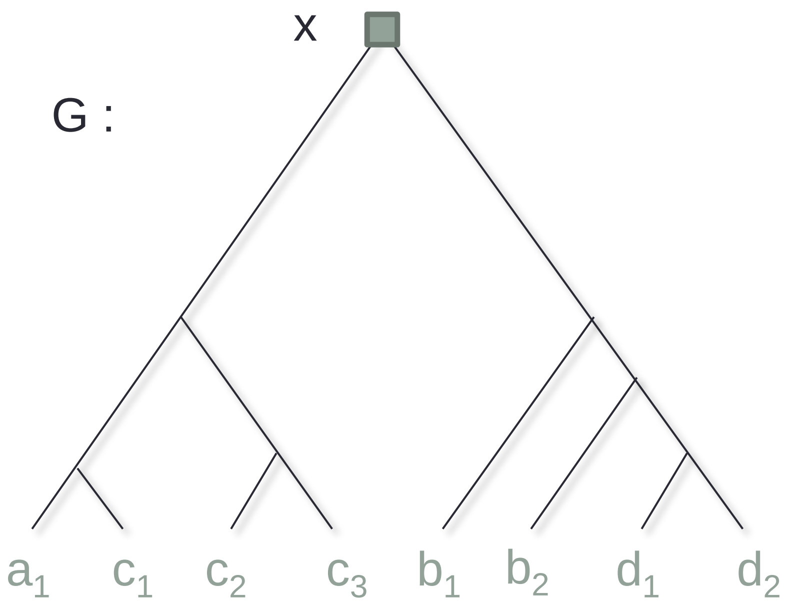
s(x) = s(y)



Problem 1

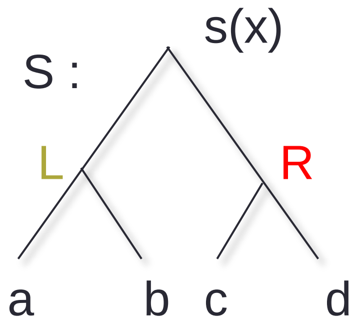
- To transform x into a speciation
    - Let L and R be the two children of s(x)

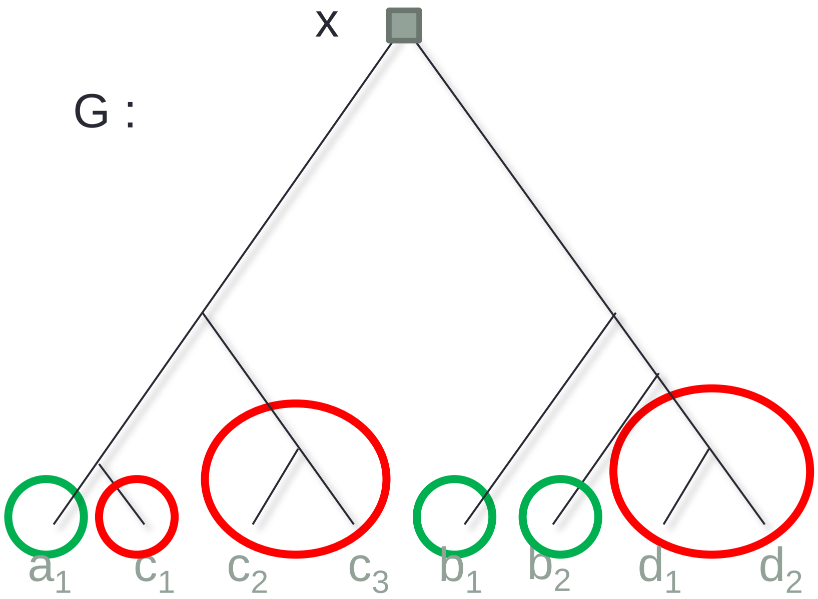




Problem 1

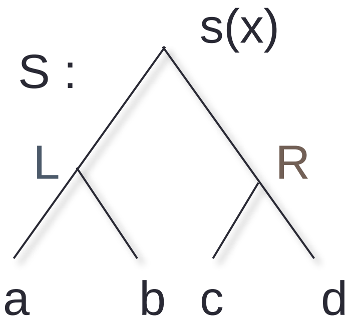
- Find G_L (resp. G_R), the set of maximal subtrees of G that contains only genes mapped to species in L (resp. R)

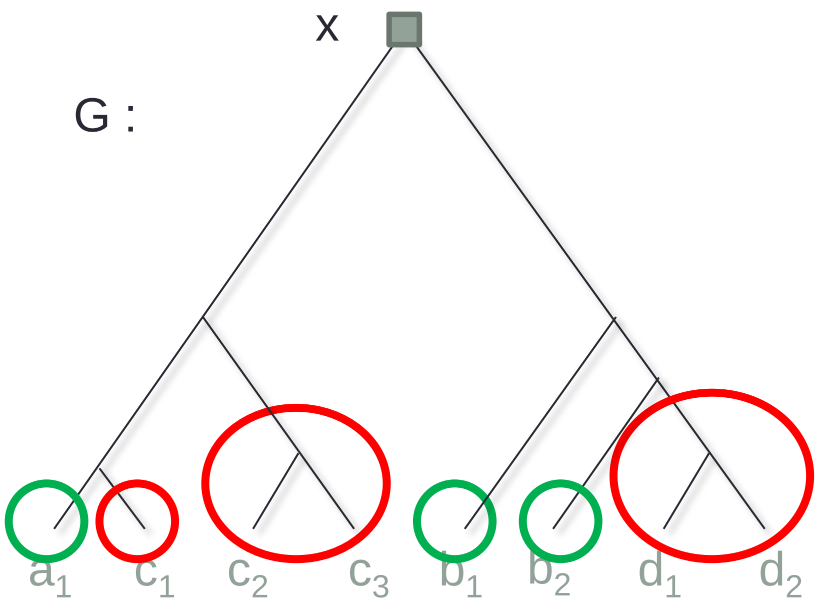


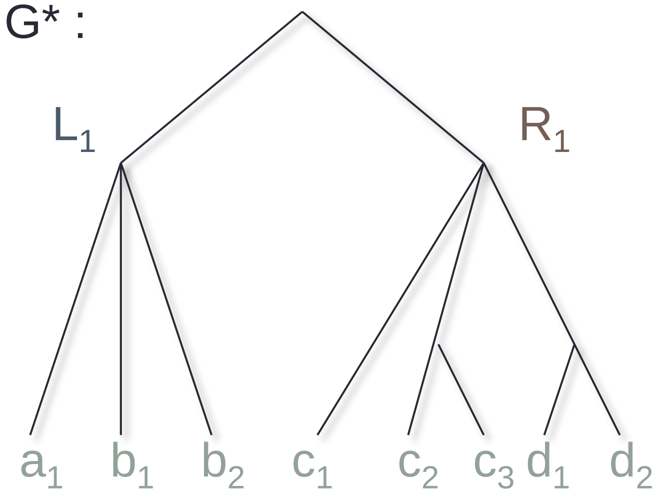


Problem 1

- Form G* by making two polytomies (non-binary subtrees) with G_L and G_R, joined under a common parent

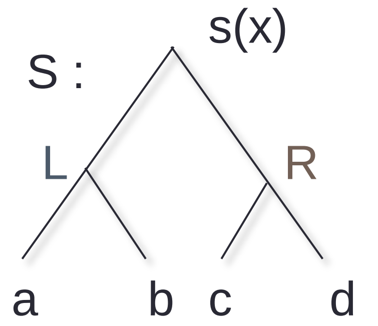


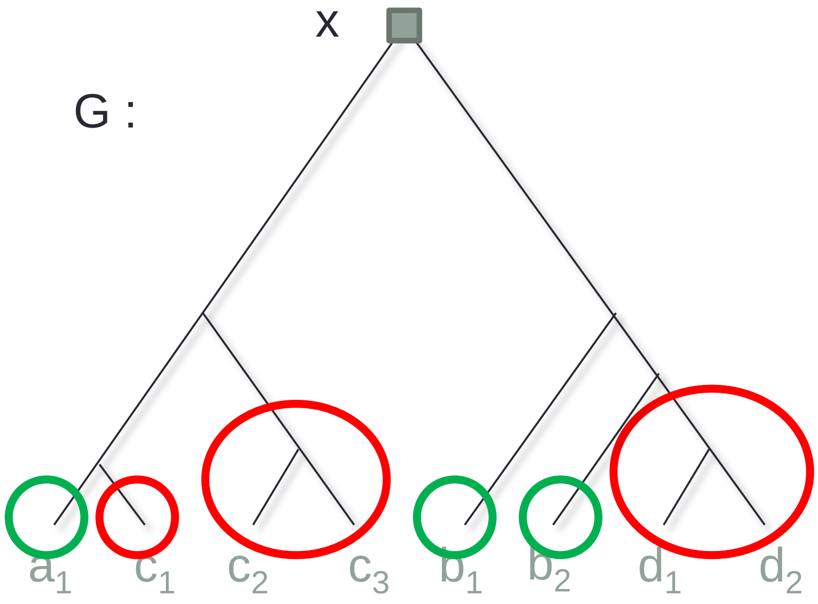


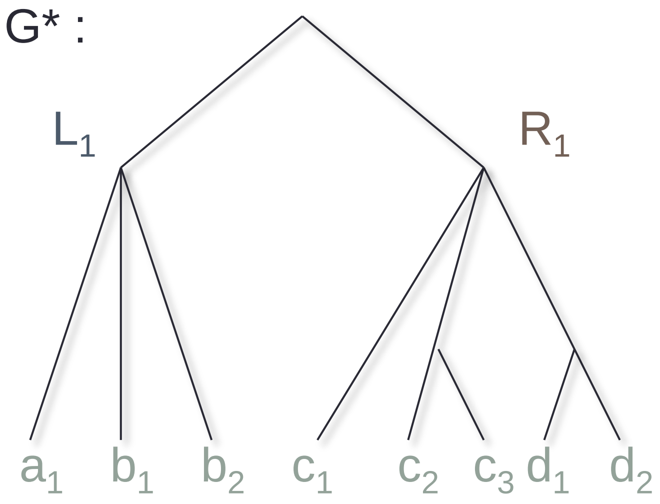


Problem 1

- Theorem : **any** binary resolution of G* is a solution to Problem 1.

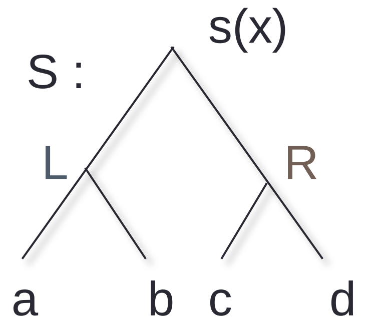


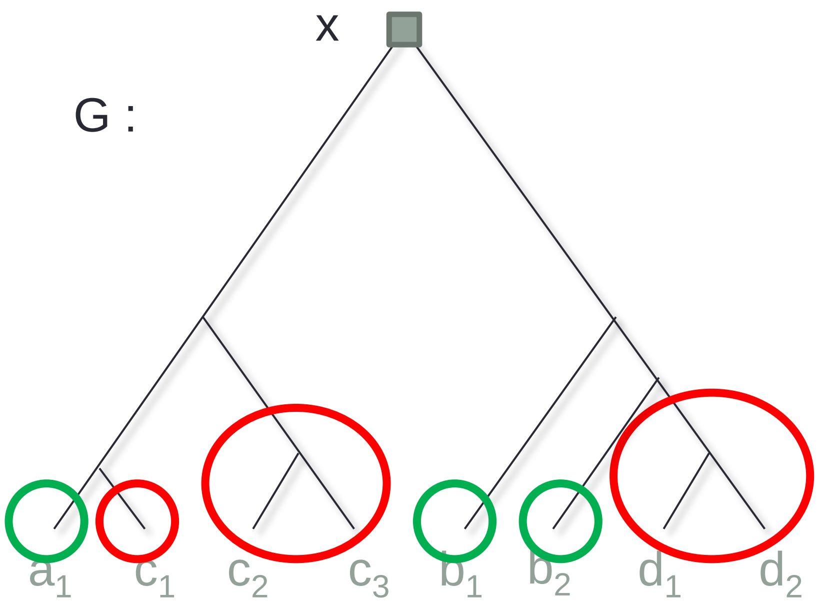


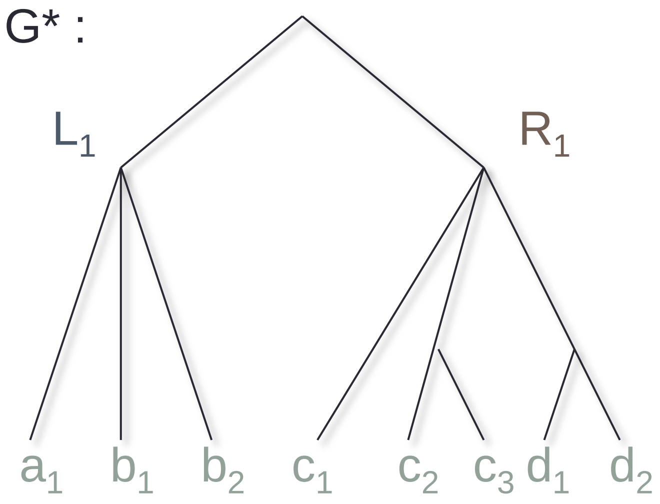


Problem 1

- Theorem : **any** binary resolution of G* is a solution to Problem 1.
- In fact, **every** solution is the result of a binary resolution of G*.

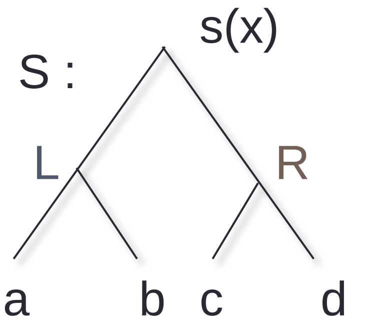


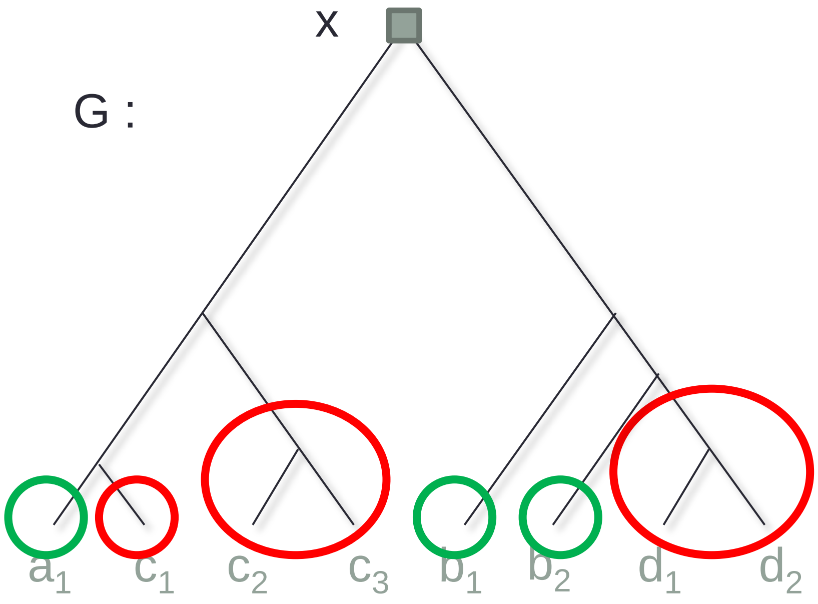


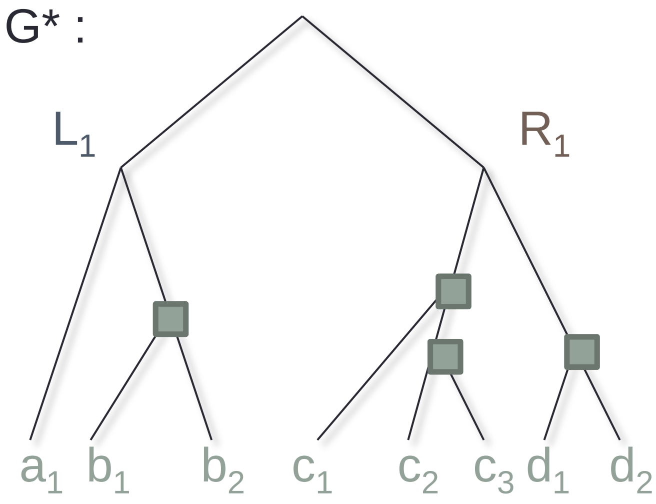


Problem 1

- Theorem : **any** binary resolution of G* is a solution to Problem 1
- In fact, **every** solution is the result of a binary resolution of G*.

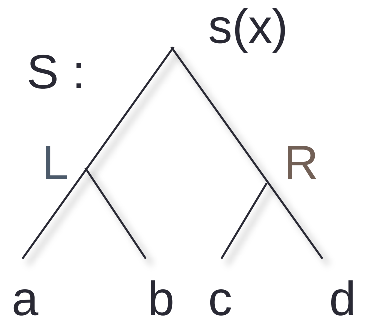


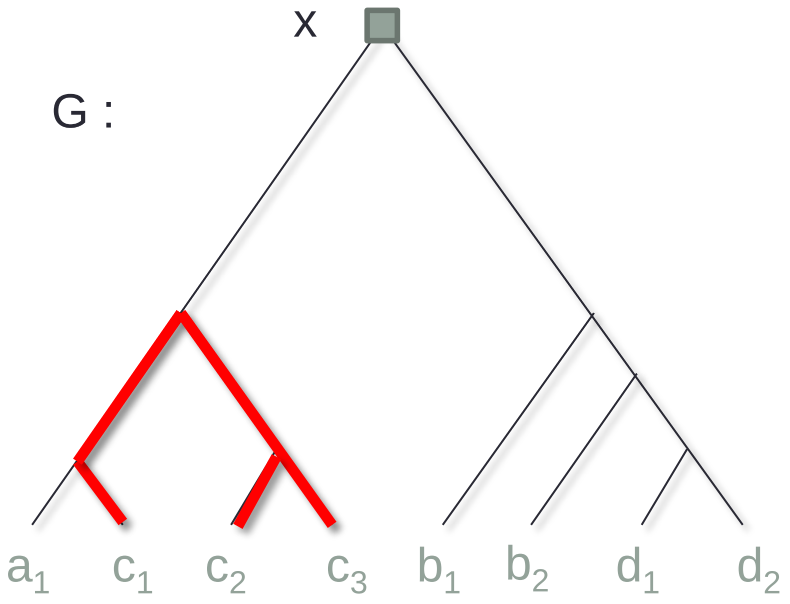




Problem 1

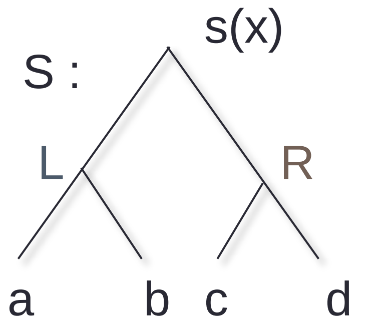
- Triplet maximizing solution :
  - For leaves x,y,z, a triplet ((x, y), z) is in G if LCA(x,y,z) is above LCA(x, y).
    - e.g. $((c_2, c_3), c_1)$ is a triplet

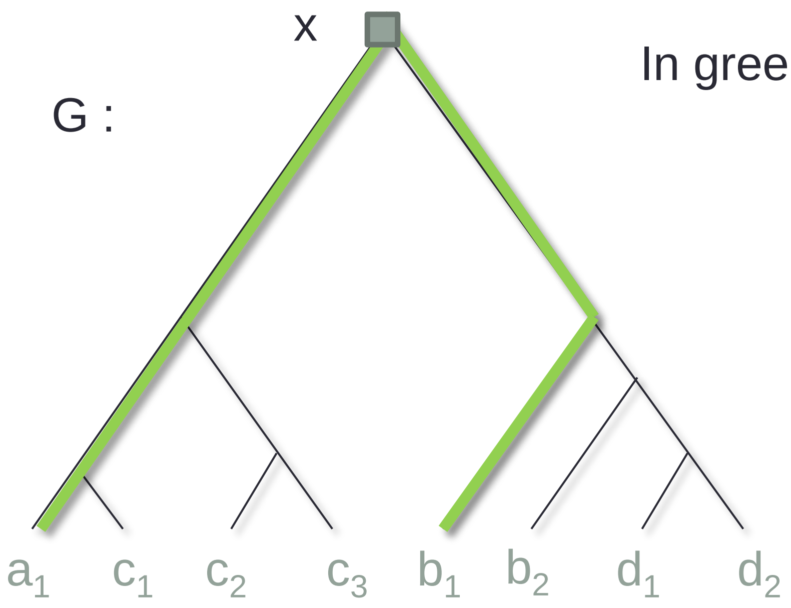




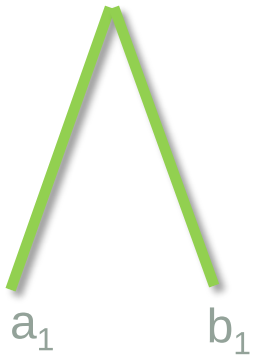
Problem 1

- Triplet maximizing solution :
  - Make G_L (resp. G_R) by taking the maximum induced tree of G containing only leaves in L (resp. R).



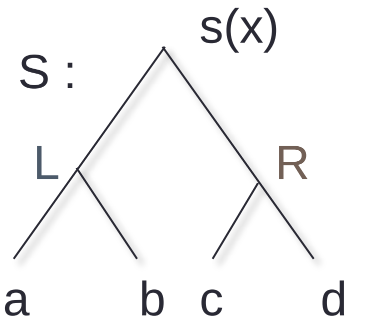


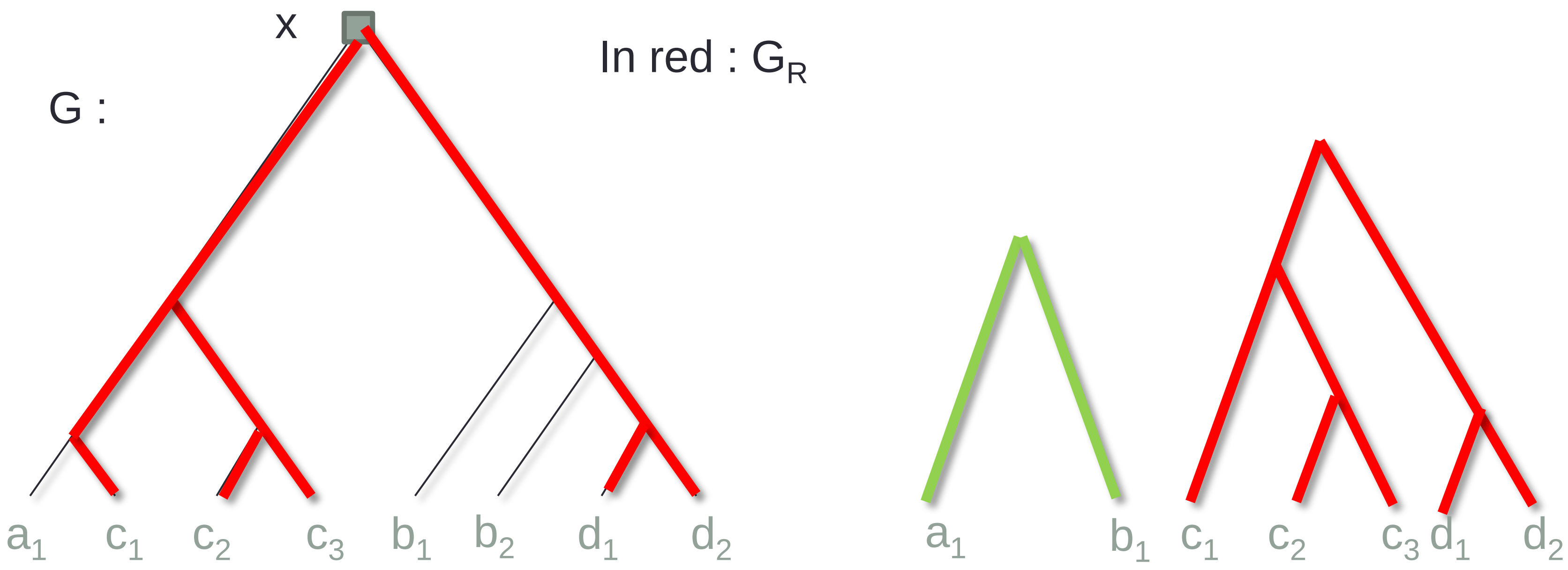
In green : G_L



Problem 1

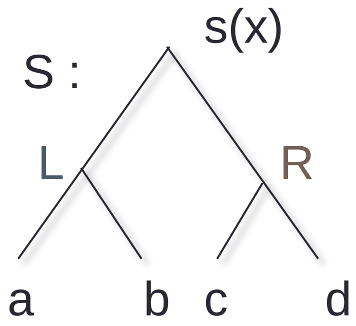
- Triplet maximizing solution :
  - Make G_L (resp. G_R) by taking the maximum induced tree of G containing only leaves in L (resp. R).

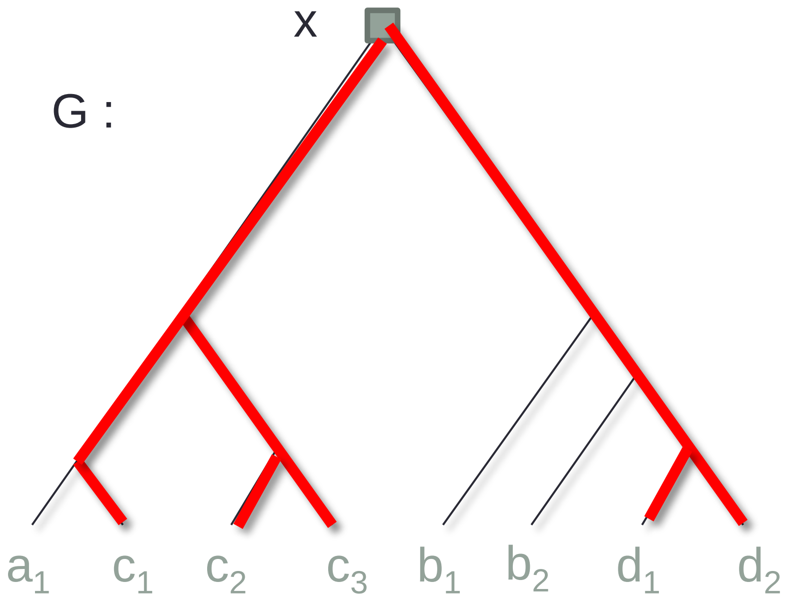


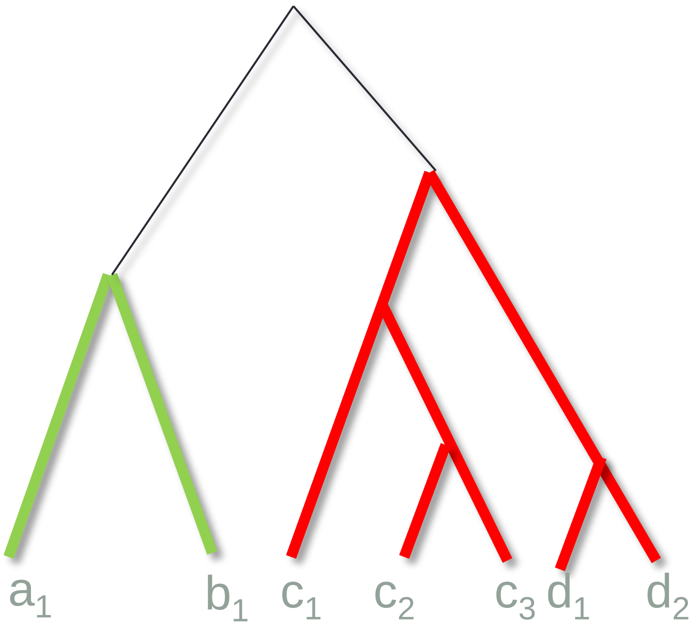

In red : G_R

Problem 1

- Triplet maximizing solution :
  - Join G_L and G_R
  - This minimizes the RF-Distance and the triplets distance !

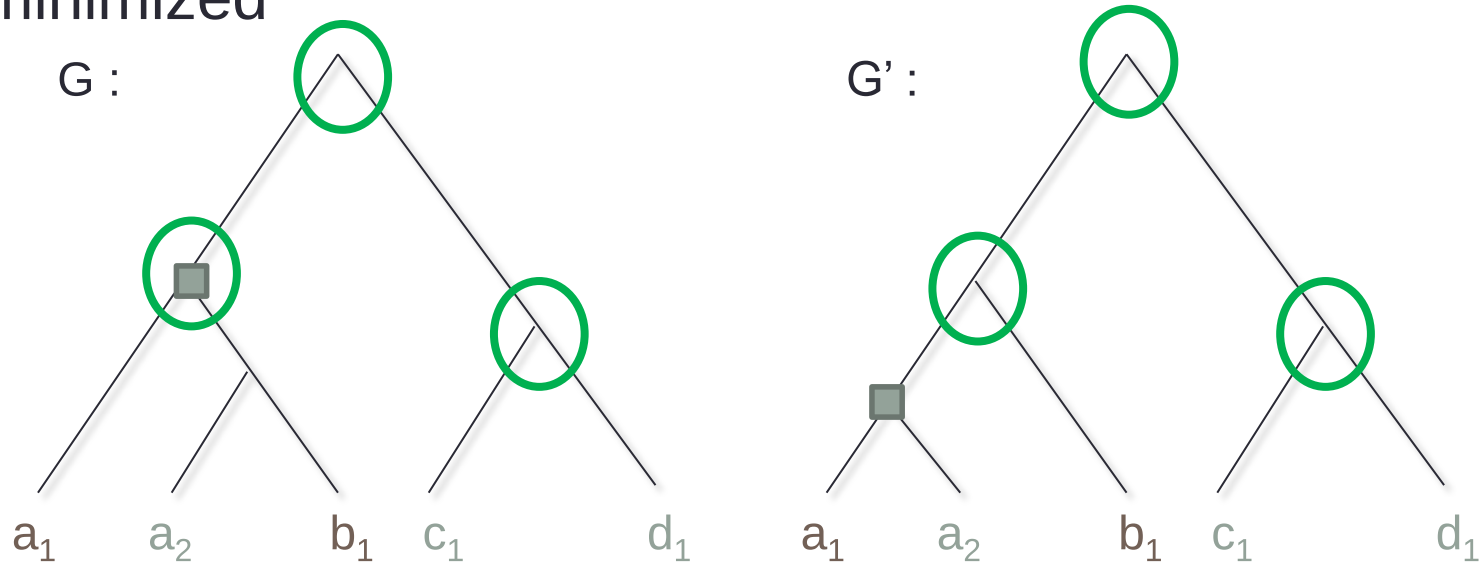






Problem 2

- **Problem 2**: given a reconciled gene tree G, a species tree S, and a set P of pairs of genes that are required to be orthologous
- **Find** : A corrected gene tree G' in which all gene pairs in P are orthologous, such that RFDist(G, G') is minimized

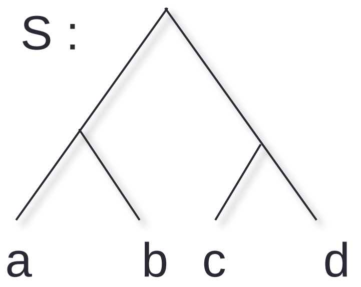


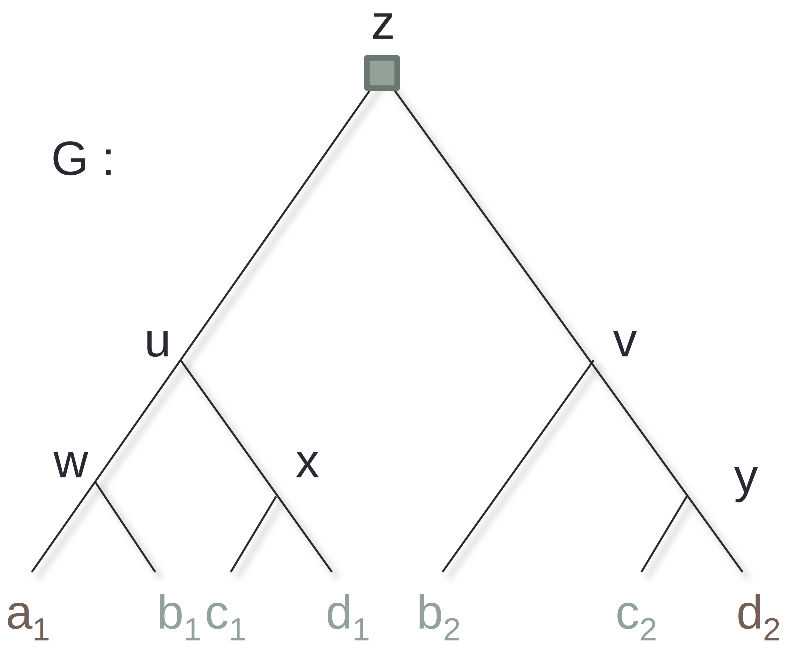
P = {(a_1, b_1)}

In green : preserved clades

Problem 1 : simple example

- a_1, d_2 should not be paralogs
- Which clades in {u,v,w,x,y,z} can we preserve ?

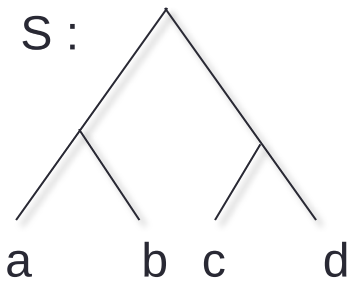


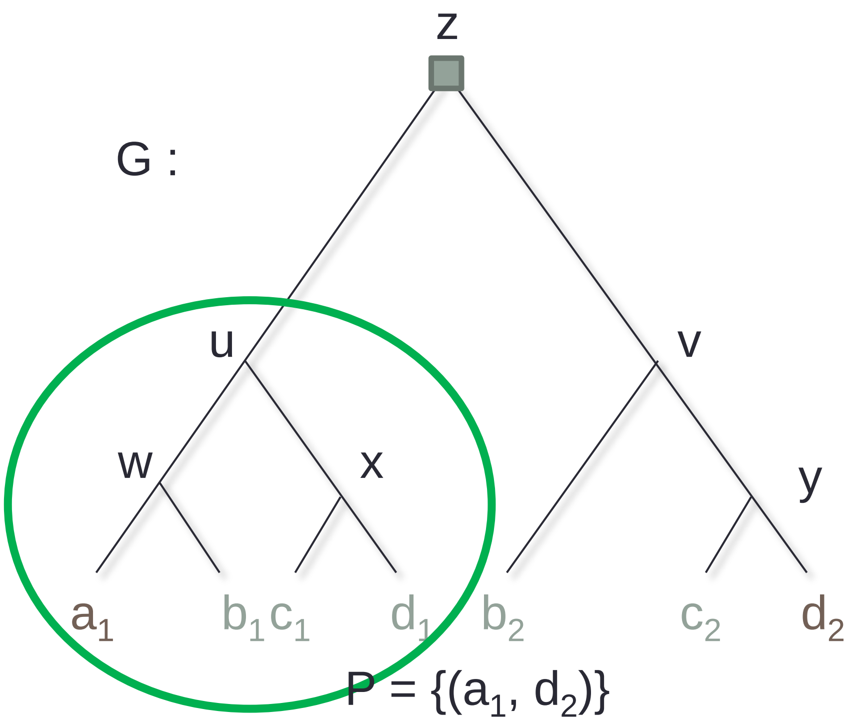


$P = \{(a_1, d_2)\}$

Problem 1 : simple example

- Can we preserve the u clade ?

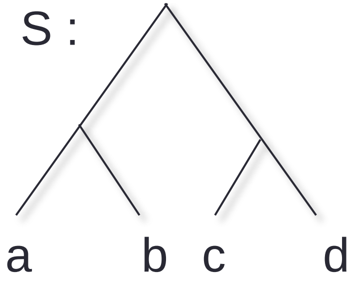


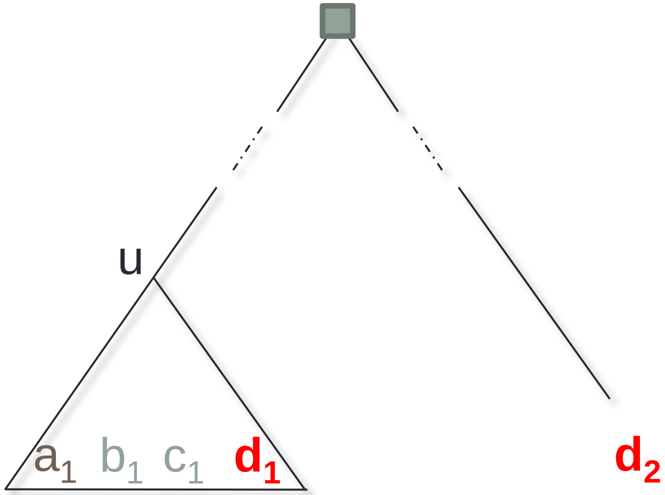


P = {(a_1, d_2)}

Problem 1 : simple example

- Can we preserve the u clade ?
- No ! Wherever d_2 ends up, by reconciliation LCA(a_1, d_2) will be a duplication (because of d_1, d_2)

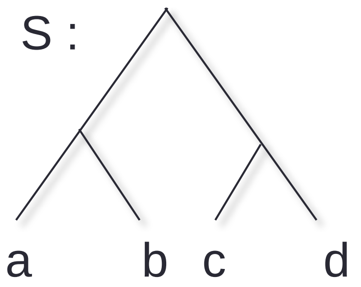


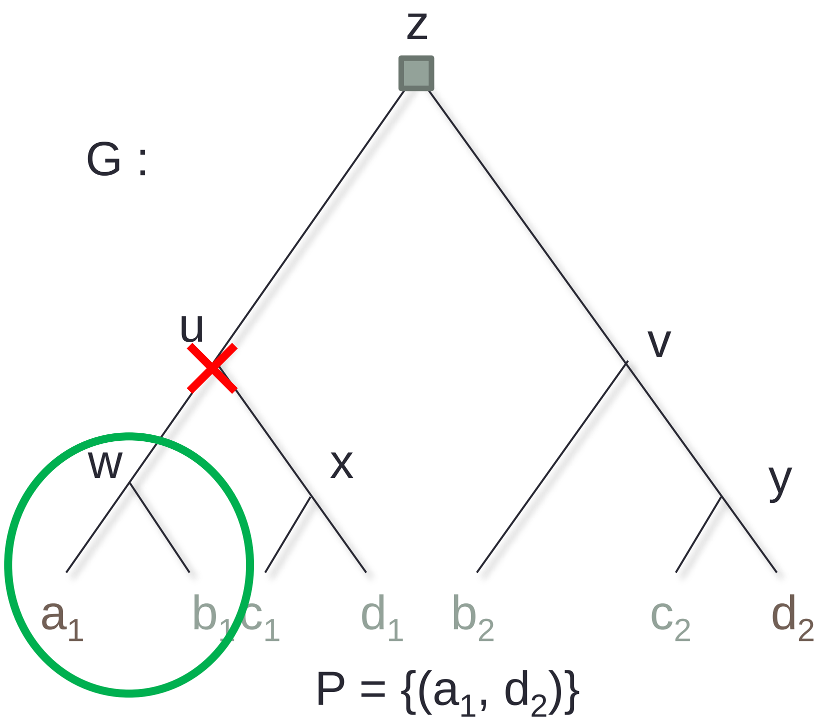


P = {(a_1, d_2)}

Problem 1 : simple example

- Can we preserve the w clade ?

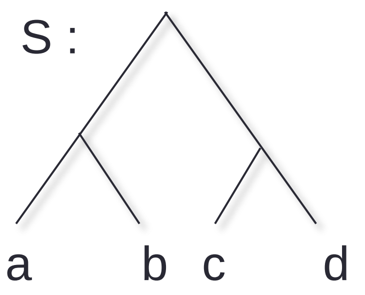


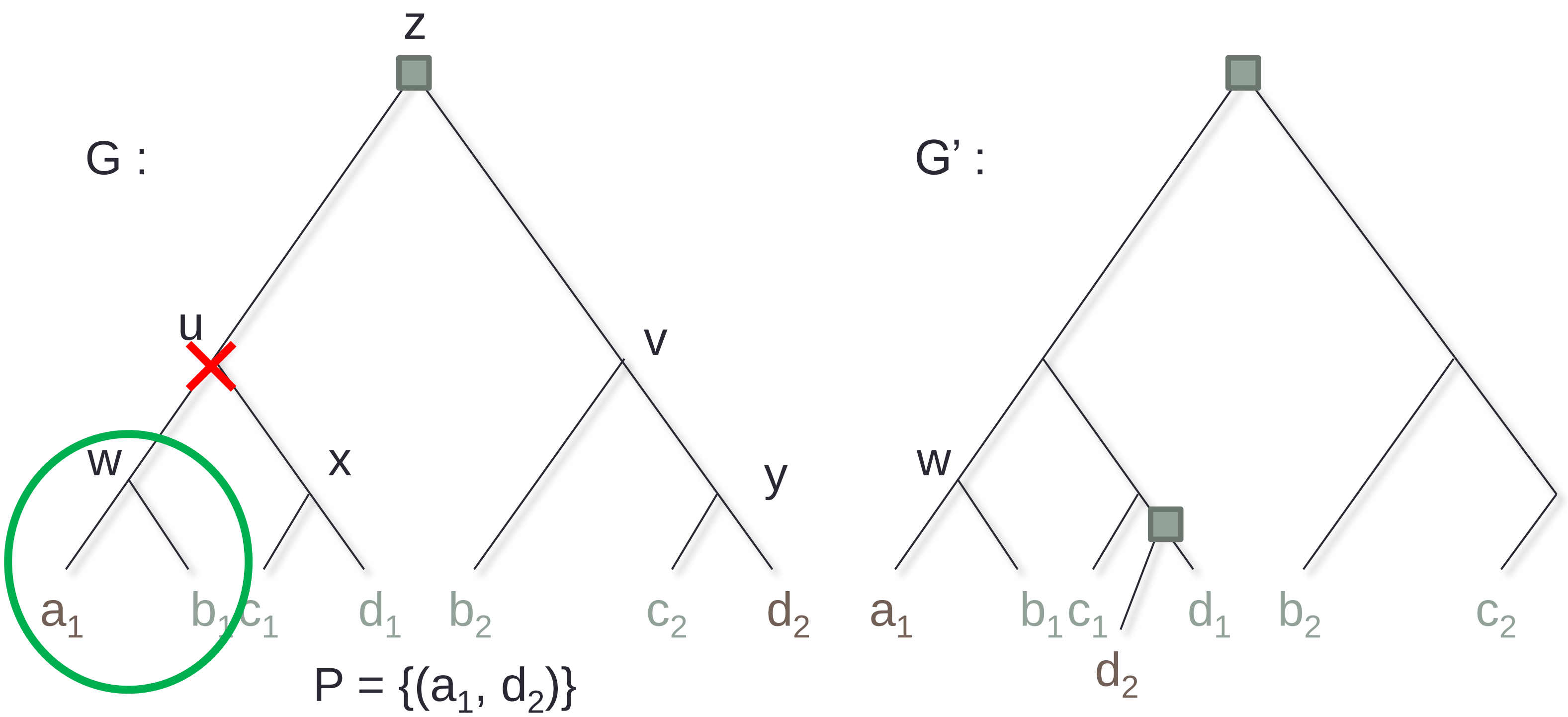


$P = \{(a_1, d_2)\}$

Problem 1 : simple example

- Can we preserve the w clade ?
- Sure ! Here’s how !

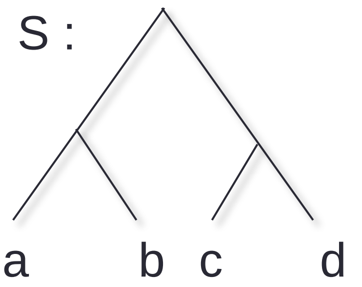


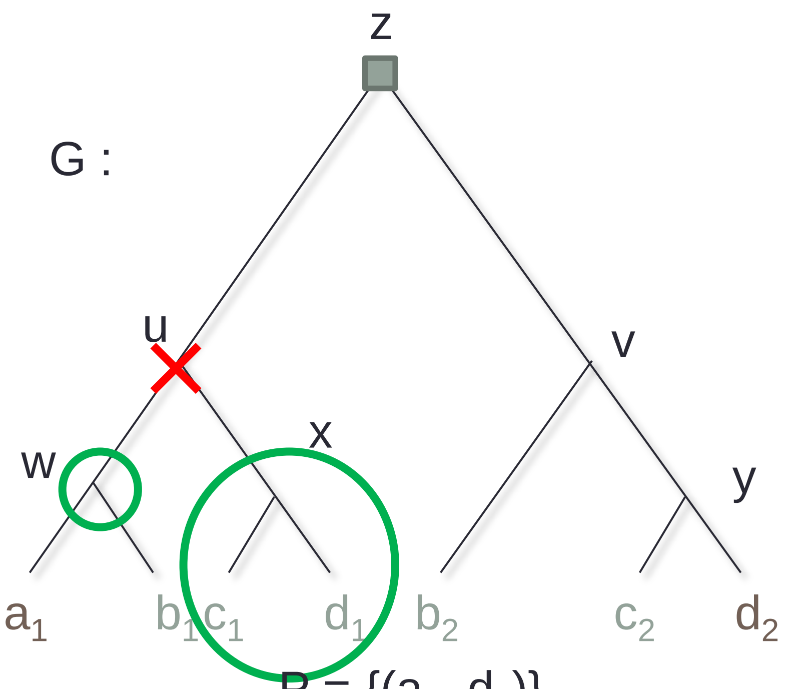


P = {(a_1, d_2)}

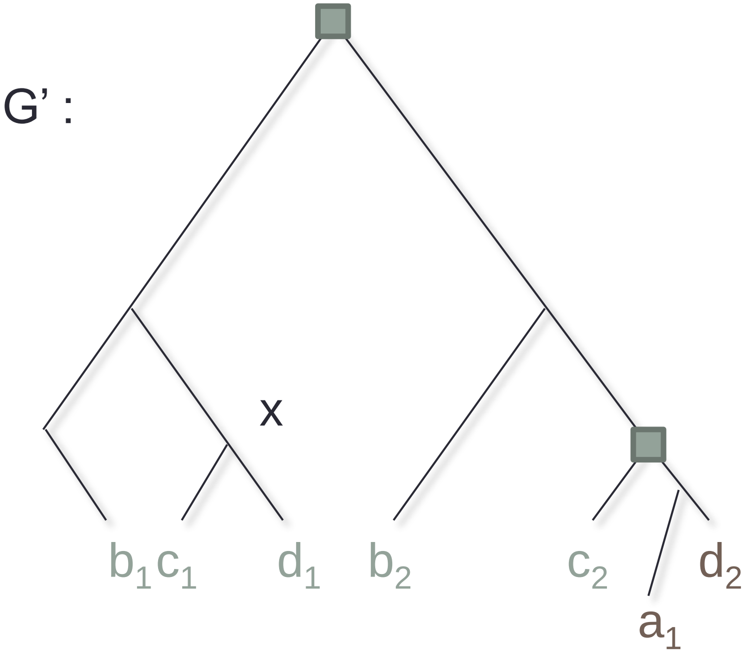
Problem 1 : simple example

- What about the x clade ?
- Just send a_1 near d_2 !



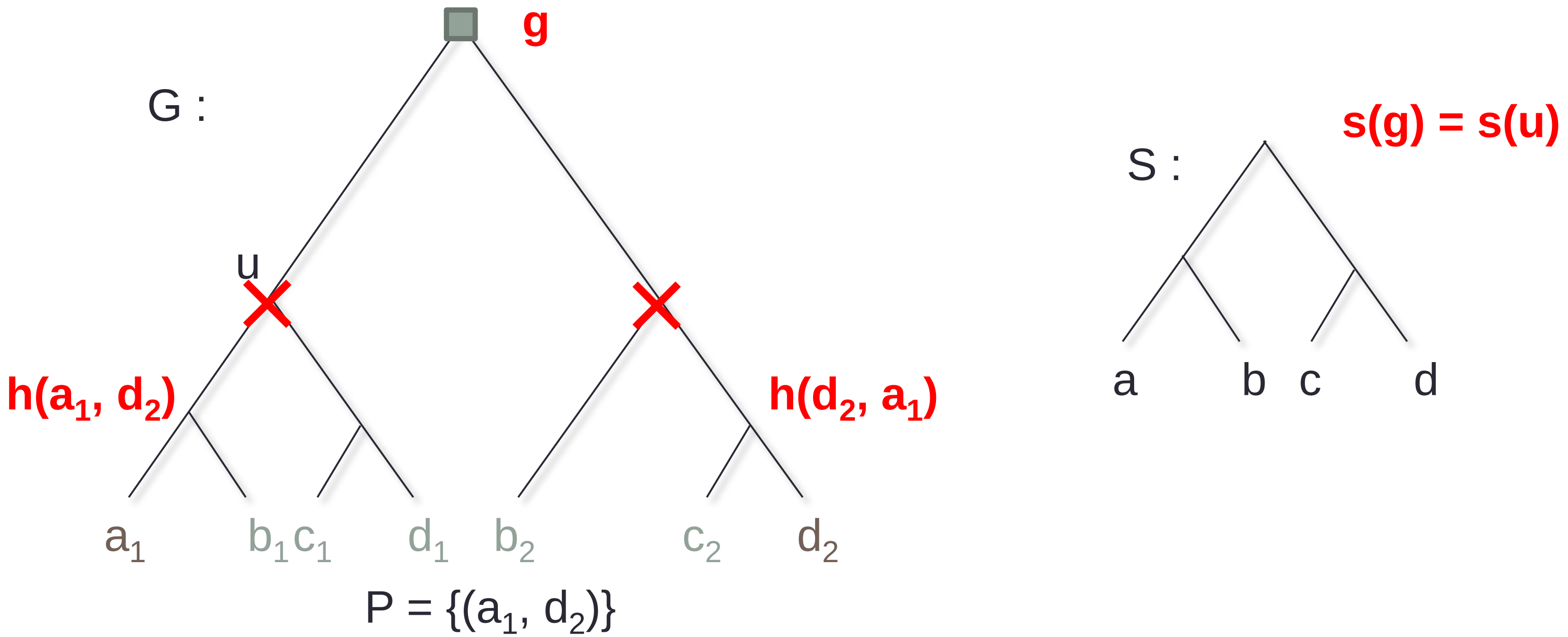


P = {(a_1, d_2)}



Problem 1 : simple example

- For some constraint (a, b) in P :
  - Let g = LCA(a,b)
  - Let $h_{a,b}$ be the highest node on the path from a to g such that $s(h_{a,b})$ is a descendant of s(g).
  - Every node on the path from $h_{a,b}$ to g (excluding h and g) corresponds to an unpreservable clade.
  - Define $h_{b,a}$ analogously



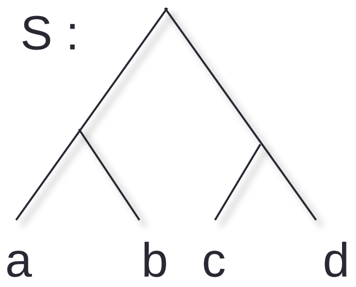
Problem 1 : simple example

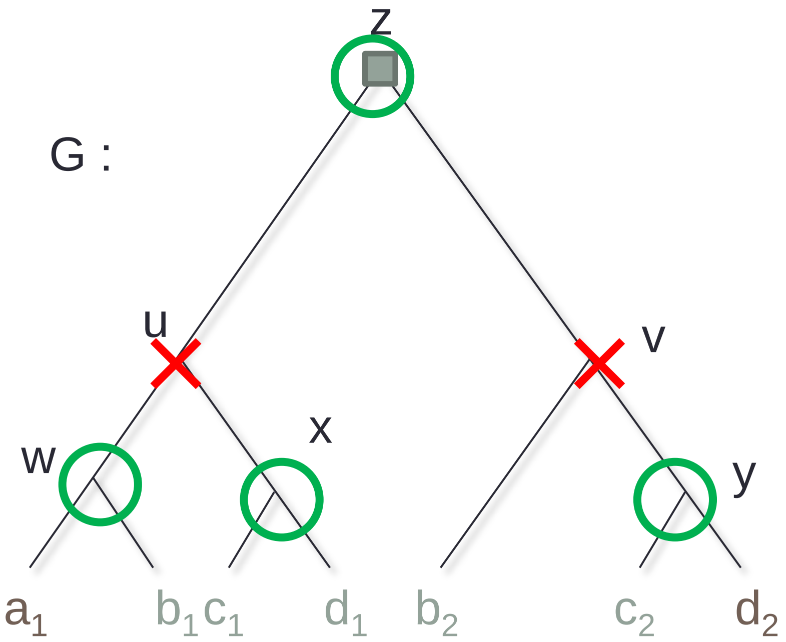
- For some constraint (a, b) in P :
  - Let g = LCA(a,b)
  - Let $h_{a,b}$ be the highest node on the path from a to g such that s($h_{a,b}$) is a descendant of s(g).
  - Every node on the path from $h_{a,b}$ to g (excluding h and g) corresponds to an unpreservable clade.
  - Define $h_{b,a}$ analogously

- For every constraint (a, b) in P
  - Compute $h_{a,b}$ and $h_{b,a}$
  - Find the unpreservable clades they imply
- Identifies all unpreservable clades.
- Can be done in time O(|P| |V(G)|)

Problem 1 : simple example

- We can identify preservable nodes rather easily (in O(|P||V(G)|) time).
- But, can we preserve them all at once ?

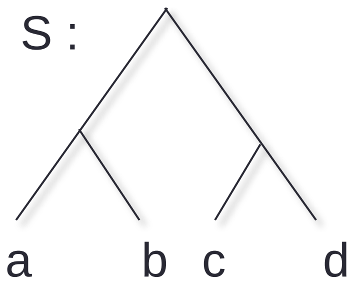


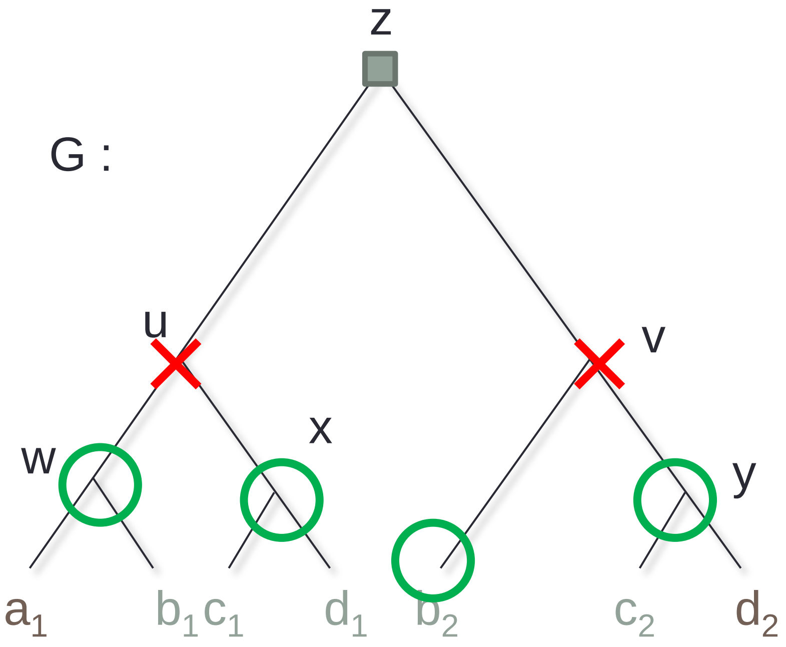


$P = \{(a_1, d_2)\}$

Problem 1 : simple example

- It turns out we can !
- ***Highest preservable descendant*** : a preservable node whose only preservable ancestor is the root





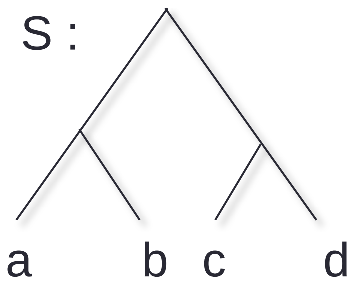
$P = \{(a_1, d_2)\}$

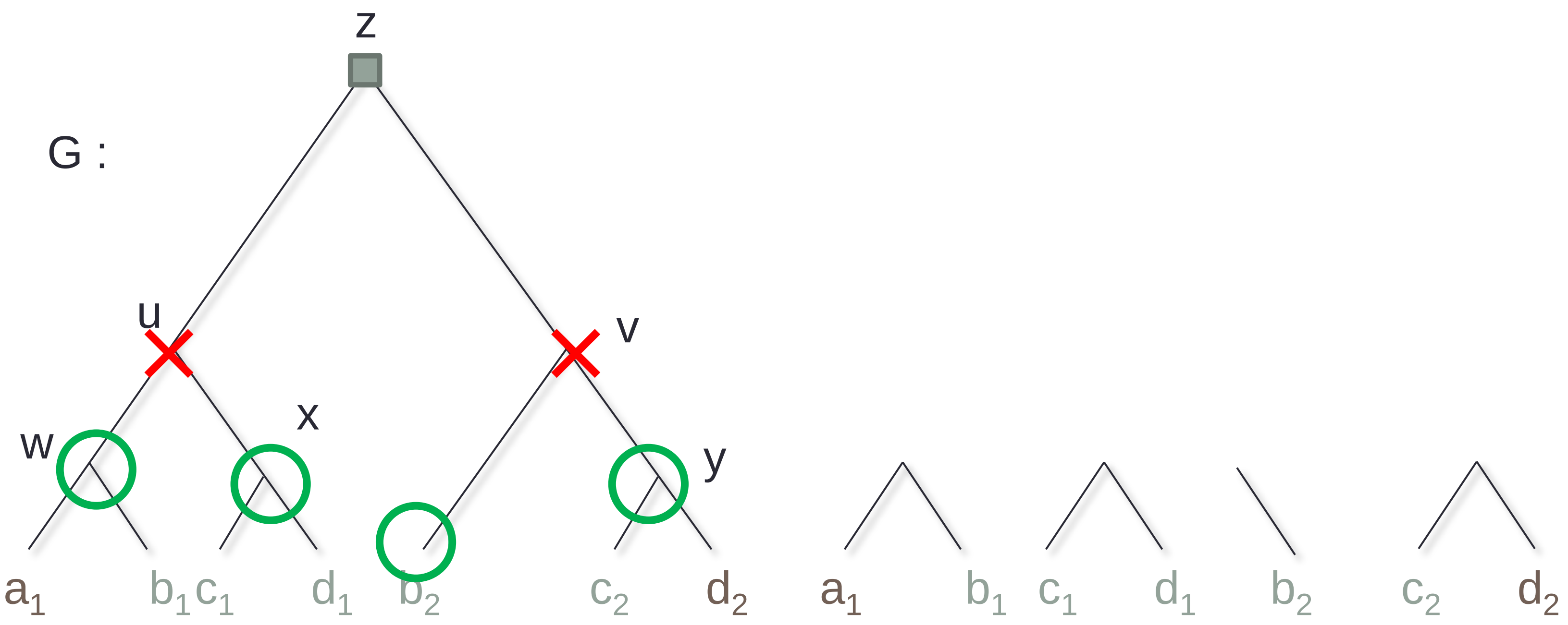
- The set of highest preservable descendants in G is $\{w, x, b_2, y\}$

  (the leaves are always preservable)
- This set partitions the leaves of G

Problem 1 : simple example

- Extract all the highest preservable subtrees

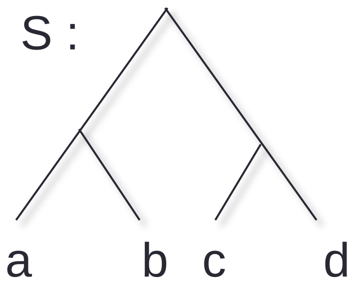


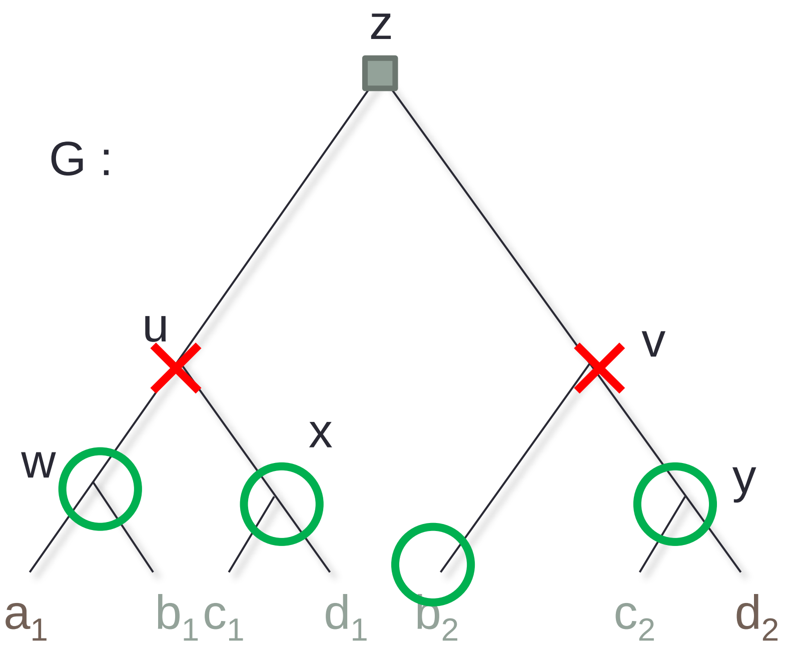


P = $\{(a_1, d_2)\}$

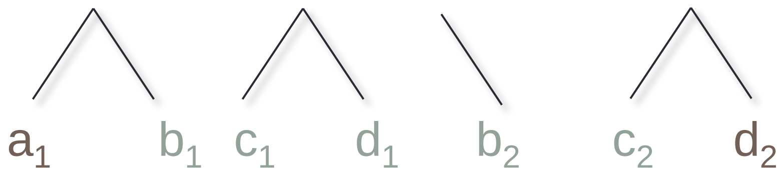
Problem 1 : simple example

- Extract all the highest preservable subtrees
- Join the subtrees in the order given by a bottom-up traversal of S
  - i.e. priorize creating a new root r such that s(r) is the lowest in S



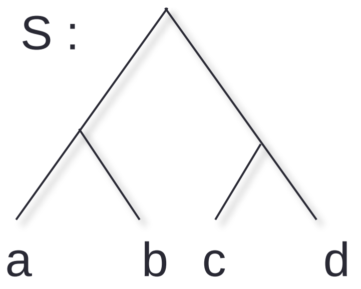


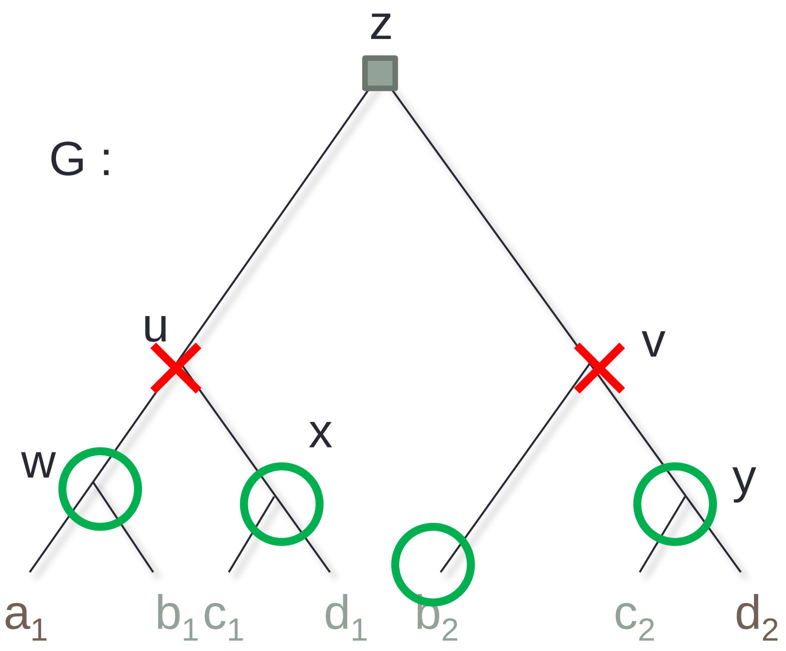
$P = \{(a_1, d_2)\}$



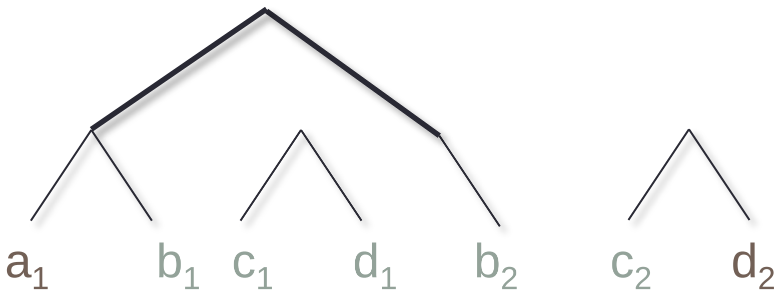
Problem 1 : simple example

- Extract all the highest preservable subtrees
- Join the subtrees in the order given by a bottom-up traversal of S
    - i.e. priorize creating a new root r such that s(r) is the lowest in S



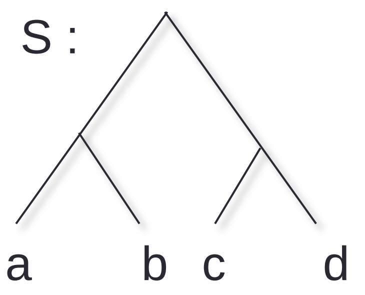


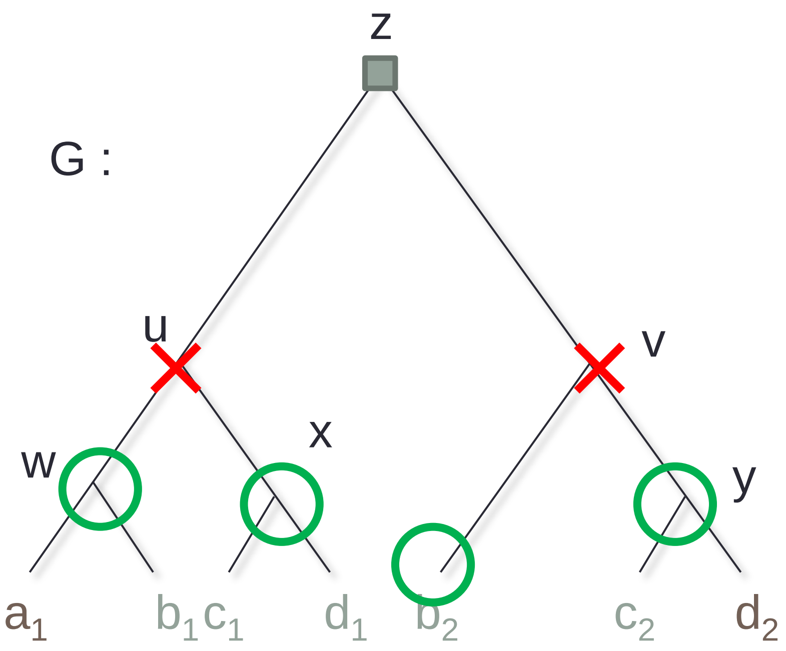
$P = \{(a_1, d_2)\}$

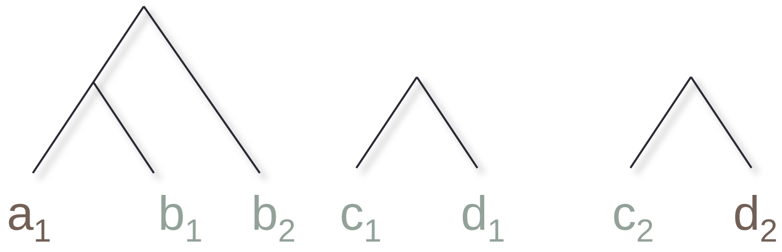


Problem 1 : simple example

- Extract all the highest preservable subtrees
- Join the subtrees in the order given by a bottom-up traversal of S
  - i.e. priorize creating a new root r such that s(r) is the lowest in S



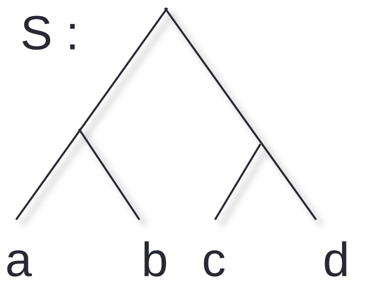


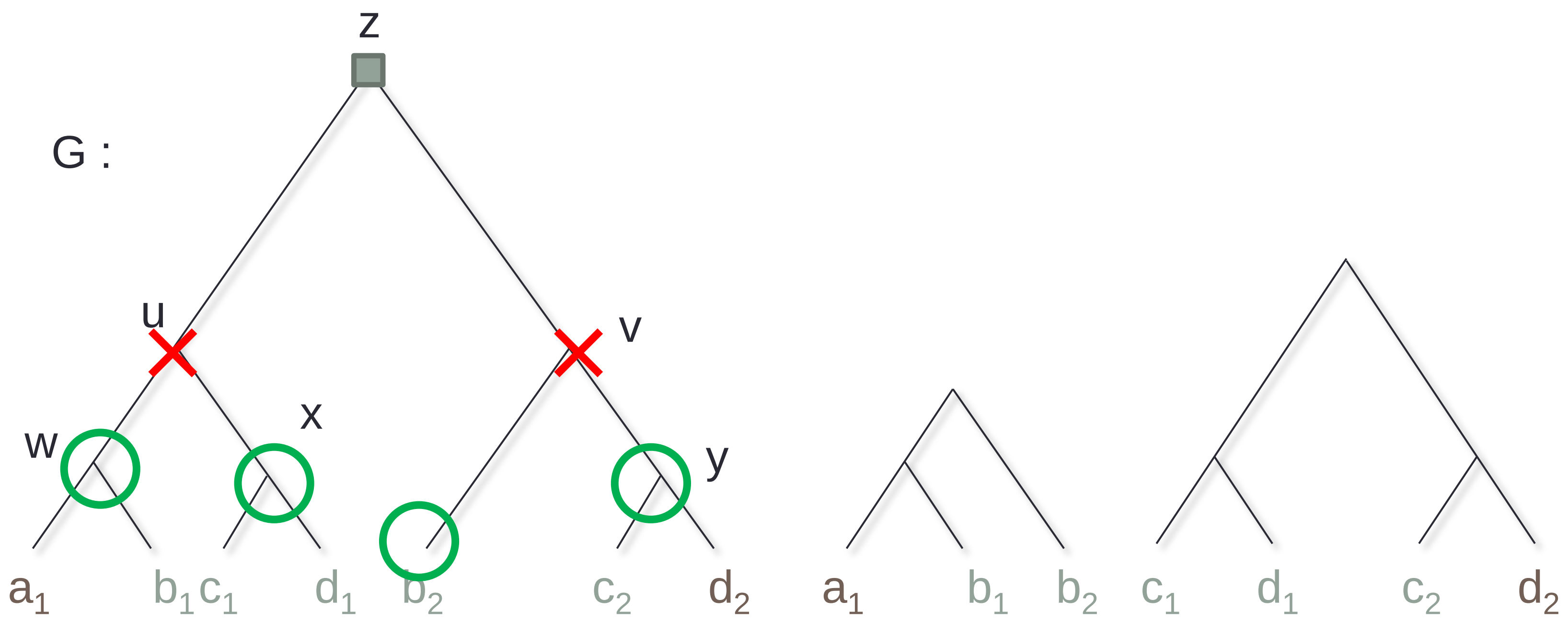


$P = \{(a_1, d_2)\}$

Problem 1 : simple example

- Extract all the highest preservable subtrees
- Join the subtrees in the order given by a bottom-up traversal of S
    - i.e. priorize creating a new root r such that s(r) is the lowest in S

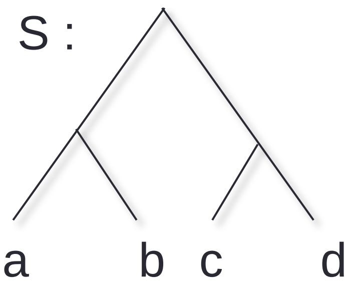


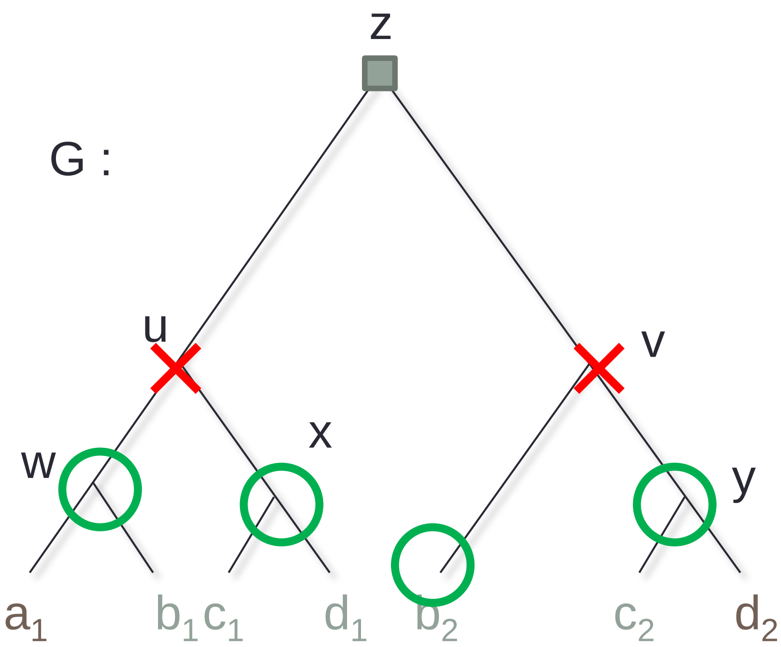


P = $\{(a_1, d_2)\}$

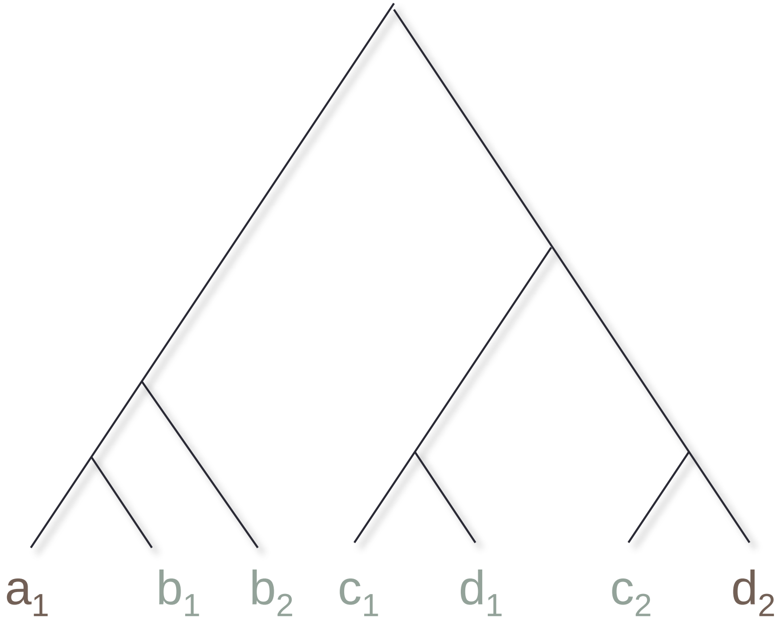
Problem 1 : simple example

- Extract all the highest preservable subtrees
- Join the subtrees in the order given by a bottom-up traversal of S
  - i.e. priorize creating a new root r such that s(r) is the lowest in S



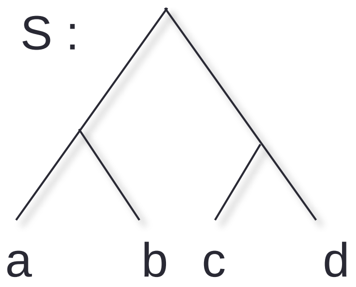


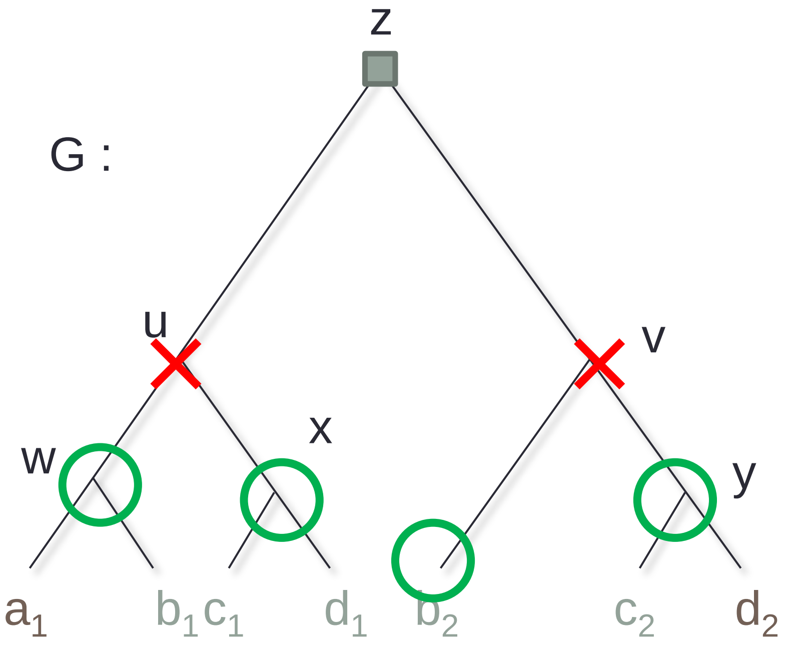
P = {(a_1, d_2)}



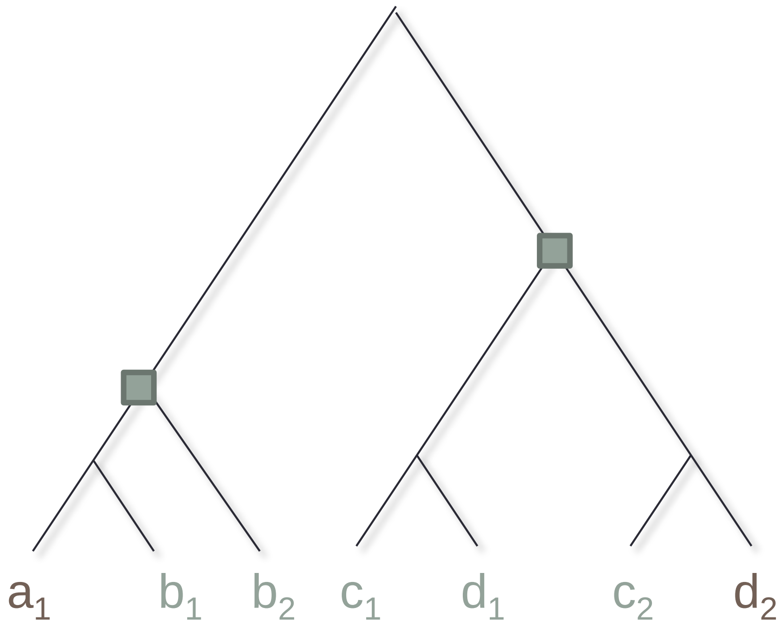
Problem 1 : simple example

- Theorem : We have formed **every** possible orthology relationship with the given subtrees.
- Corollary : our required orthologs are orthologs



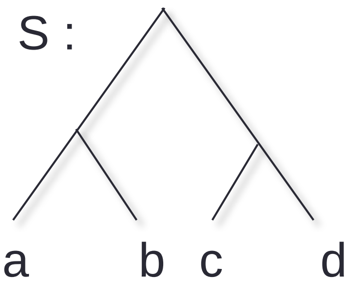


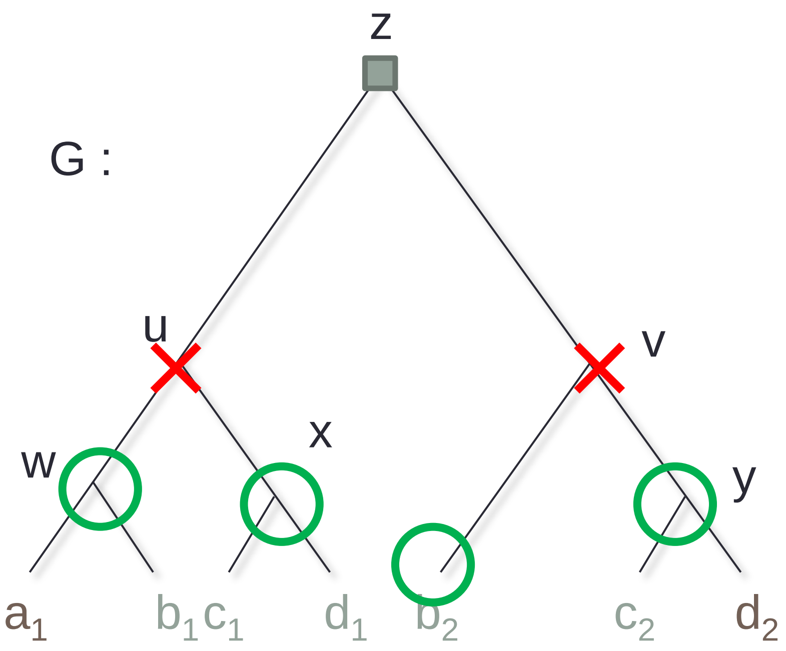
$P = \{(a_1, d_2)\}$



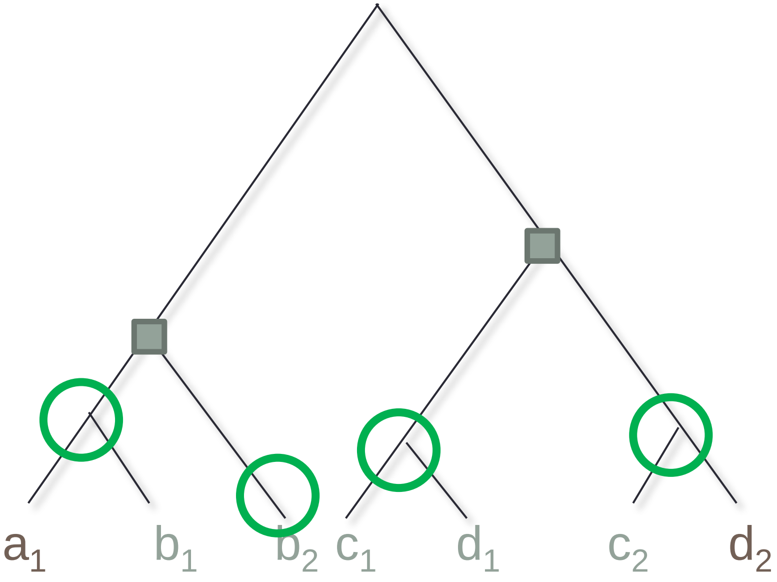
Problem 1 : simple example

- We’re done. Now a_1, d_2 are orthologs, and we saved every clade we could.





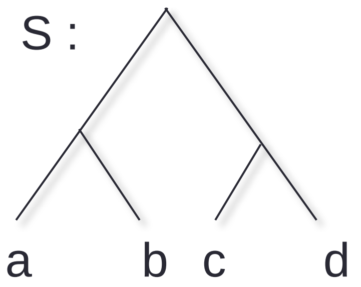
P = {(a_1, d_2)}

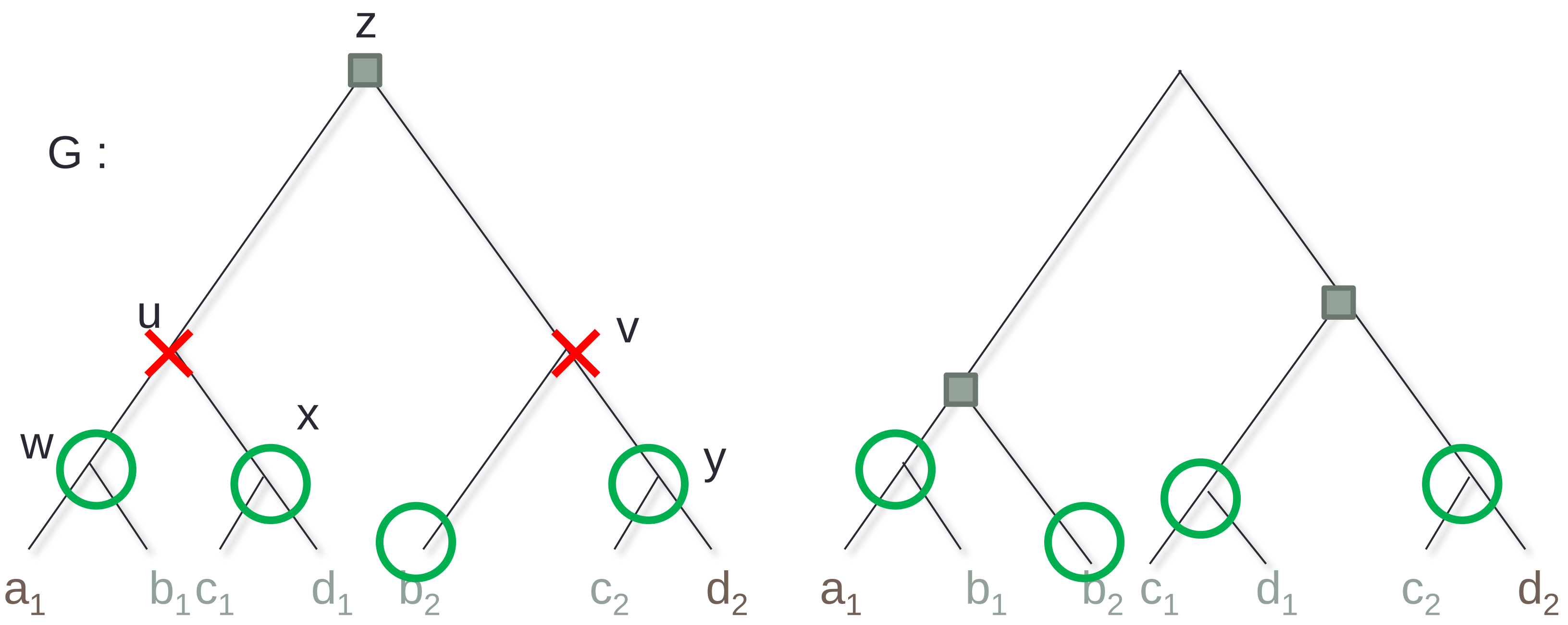


Problem 1 : simple example

- If there are still bad duplications, then they are in the highest preservable subtrees.
  - Recursively repeat the procedure for every one of them, until we get to the leaves.

S :





P = {(a_1, d_2)}

Some results

- Using synteny to find required orthologs, we corrected 1000 Ensembl gene trees with the problem 2 algorithms (with our four favorite fish species)
- Then used the **AU Test** to verify the plausibility of our corrected gene trees
  - 82.3% of our trees were statistically viable

  - 17.7% of our trees were rejected

  - 14.8% of the original Ensembl trees were rejected

Open avenues

- Can we find required ortholog/paralog gene relationships without the gene tree ?
  - The more we have, the more precise the gene tree will be.
- Problem : given a set of required orthologs AND required paralogs, are they compatible ?
  - Does there exist a gene tree that satisfies the given constraints ?

Open avenues

- Is the RF Distance the best ?
  - Other distances : NNI, SPR, …
- Can we incorporate orthology/paralogy constraints into the gene tree building procedure, instead of correcting it a posteriori ?