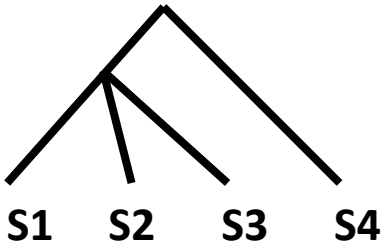
Parsimonious Reconciliation of Non-binary Trees

Louxin Zhang
National University of Singapore
matzlx@nus.edu.sg

Gene Tree vs the (Containing) Species Tree.

A species tree S represents the evolutionary history of species

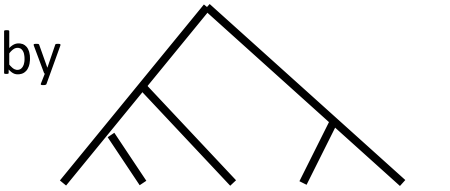
- S can be binary or non-binary .
  -- Hard polytomy
  -- Soft polytomy.
- Each leaf represents a modern species.



Species tree S

A gene tree G is over the members of a gene family

- G can be non-binary.
  --- Non-binary nodes are soft polytomies.
- Each leaf represents a family gene and is labeled by the species where it resides.

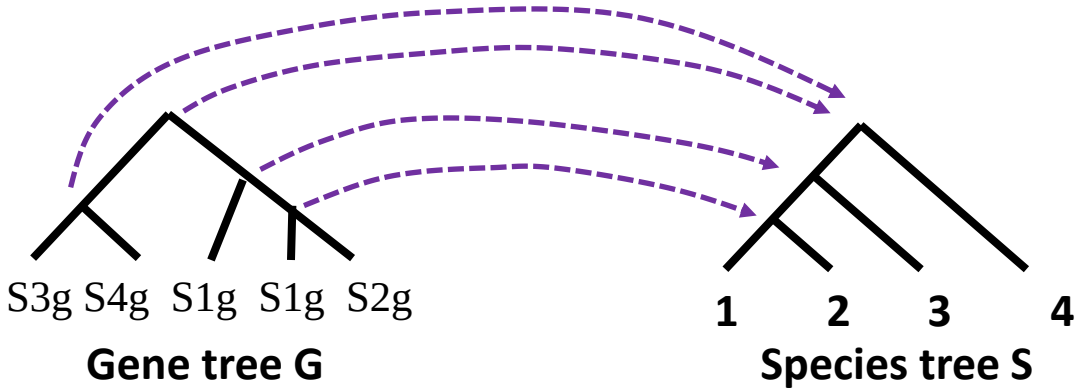


Gene tree G

Part 1: Reconciliation

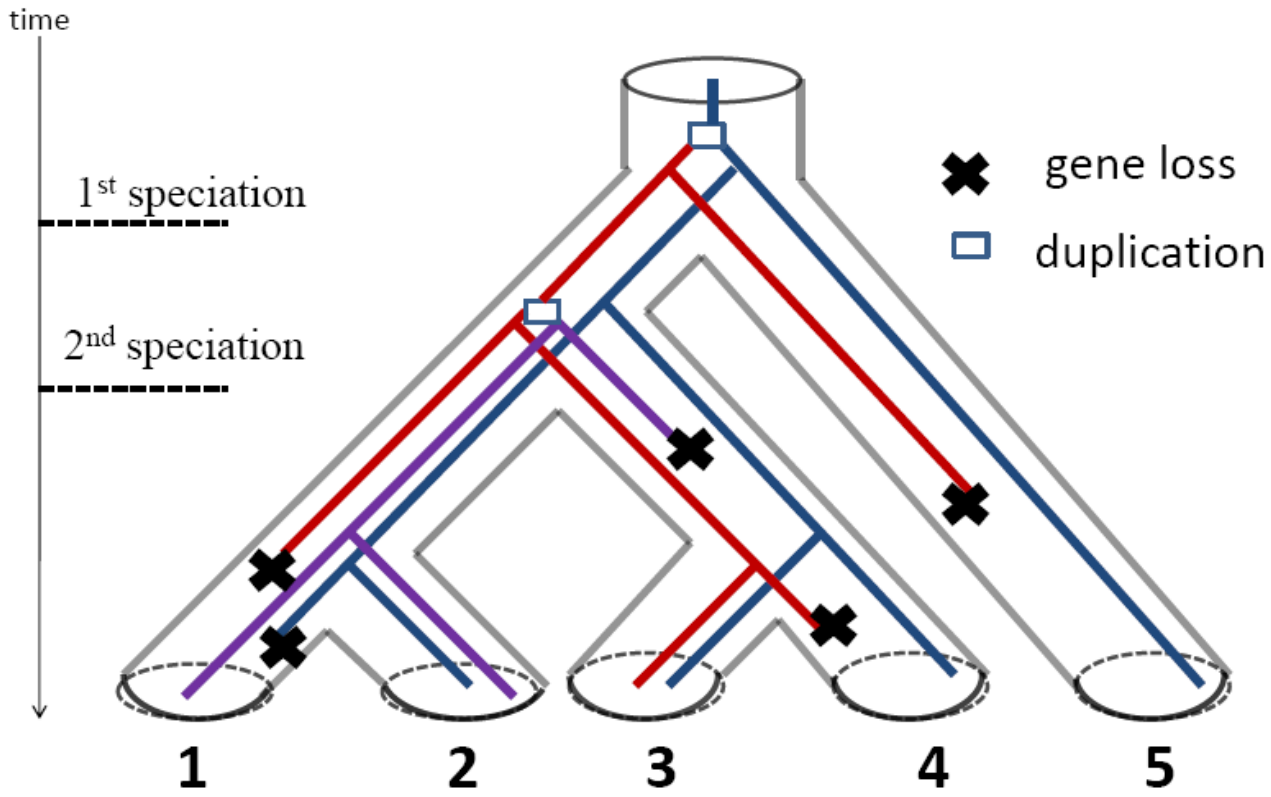
A **reconciliation** between gene tree G and species tree S is a map from $V(G)$ to $V(S)$ with the following properties:

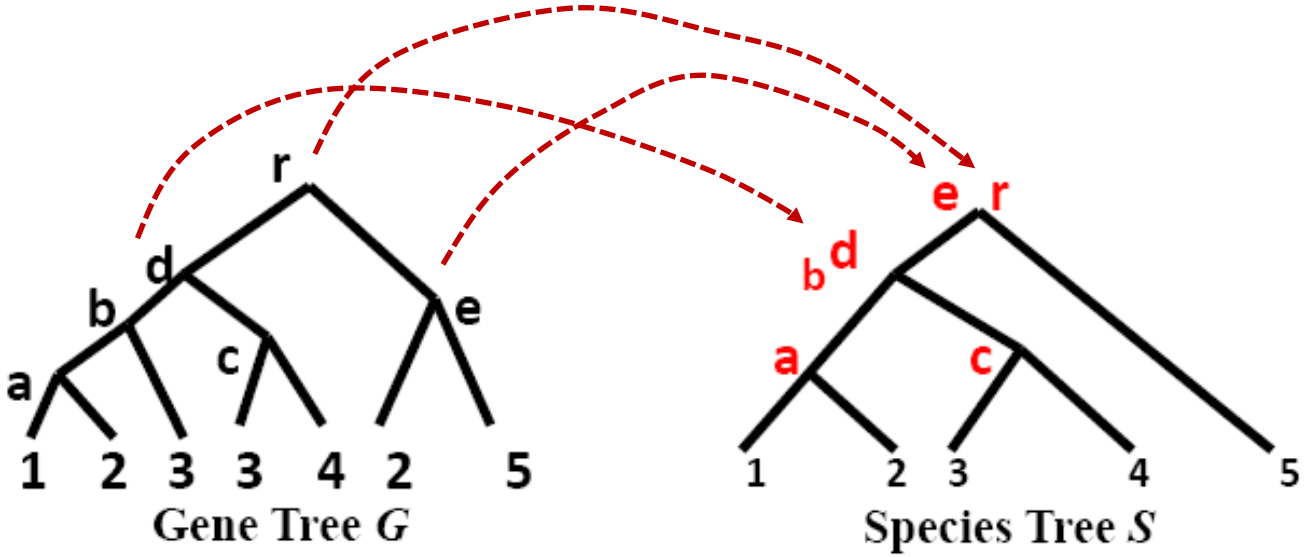
- **Leaf-preserving:** $f(x)$ is a leaf with the same label if $x \in \text{Leaf}(G)$;
- **Order-preserving:** $u \le v \Rightarrow f(u) \le f(v), \forall u, v \in V(G)$.



Gene tree and species tree reconciliation is an important method for

- Inferring duplications, losses, and horizontal transfers
- Inferring orthology and paralogy gene relationship



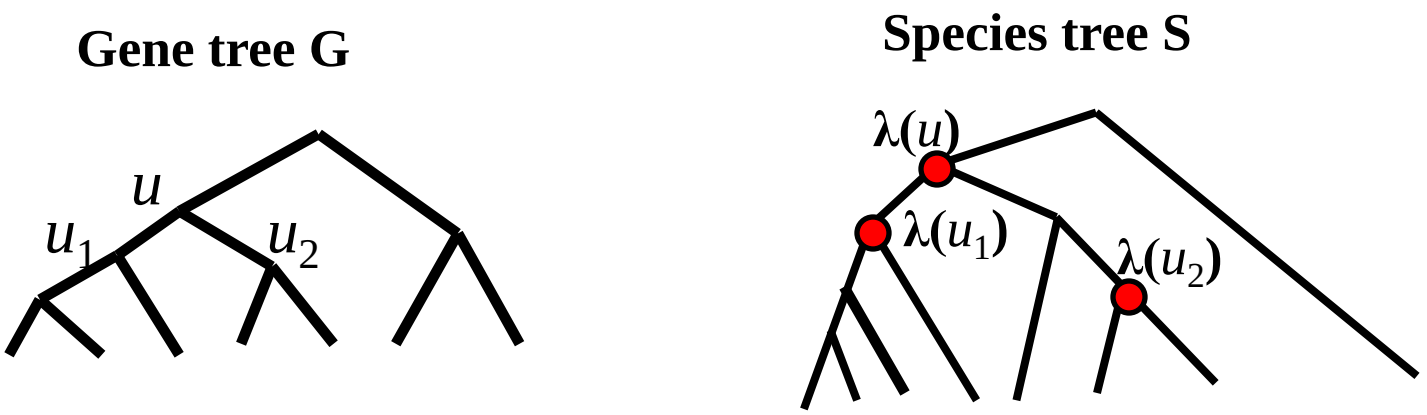


Lowest **C**ommon **A**ncestor **Reconciliation λ**

(Goodman et al, 1979, Page, 1994)

-- It maps leaves to respective leaves with the same label;

-- it maps internal node *g* is the lowest common ancestor of the images of its children.



Let $u \in V(G)$ with children u_1 and u_2.

- A duplication is inferred with u iff

$$\lambda(u) = \lambda(u_1), \text{ or } \lambda(u) = \lambda(u_2).$$

The inferred duplication occurred in the branch entering $\lambda(u)$

- Gene losses are computed using the # of branches on the path from $\lambda(u)$ to $\lambda(u_i), i = 1,2.$

(The duplication cost of λ) = (the # of inferred duplications).
The gene loss cost is defined similarly.

$\mathcal{R}$(G, S): All the reconciliations of G and S.

λ : The lca reconciliation of G and S.

Theorem Let G and S be binary. Then,

i). λ has the smallest duplication cost in $\mathcal{R}$(G, S) **(Gorecki & Tiuryn, 2006)** ;

ii). λ is the unique one with the smallest loss cost in $\mathcal{R}$(G, S) **(Chauve et al, 2009)**;

iii). λ is the unique one with the smallest deep coalescence cost in $\mathcal{R}$(G, S) **(Wu & Zhang, 2011)**;

iv). λ is computable in O(|G|+|S|) **(Zhang, 1997)**.

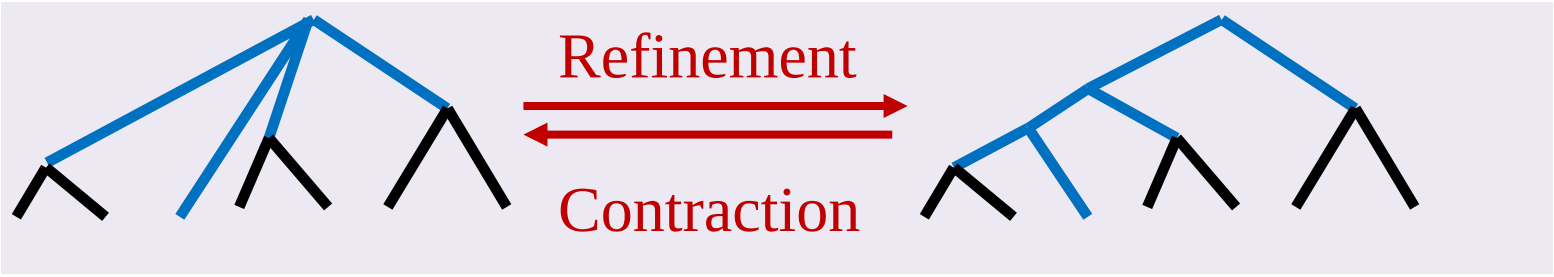
λ is the parsimonious solution for binary trees.

Part II: Reconciliation with Non-binary Trees

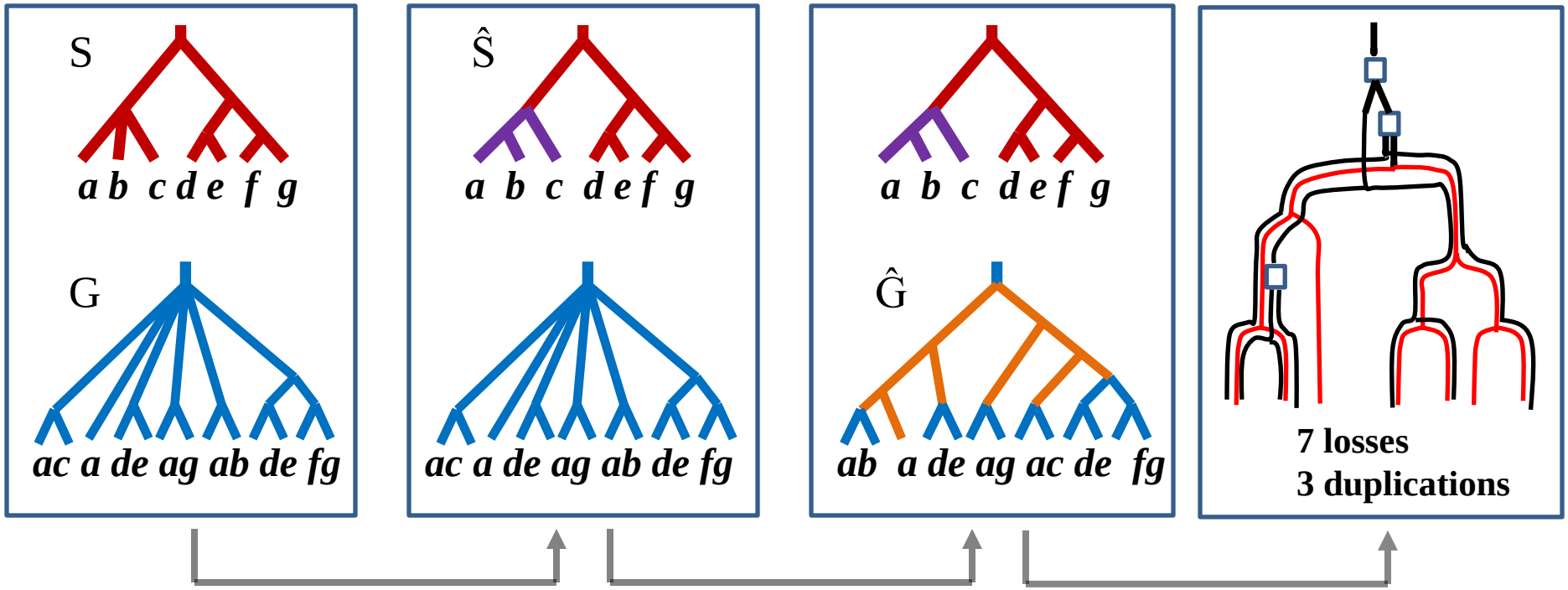
General Reconciliation Problem:

Instance: A gene tree G and a species tree S;

Solution: Binary refinement $\hat{G}$ of G and $\hat{S}$ of S such that the lca reconciliation of $\hat{G}$ and $\hat{S}$ minimizes a reconciliation cost.



Our Heuristic Reconciliation Procedure



Step 1
Refine S based on structural inform. of gene tree G

Step 2
Refine G based on the refinement $\hat{S}$ of species tree S

Step 3
Reconcile $\hat{G}$ and $\hat{S}$ to infer the evolution of the gene family

Refine S based on the Structural Information of G

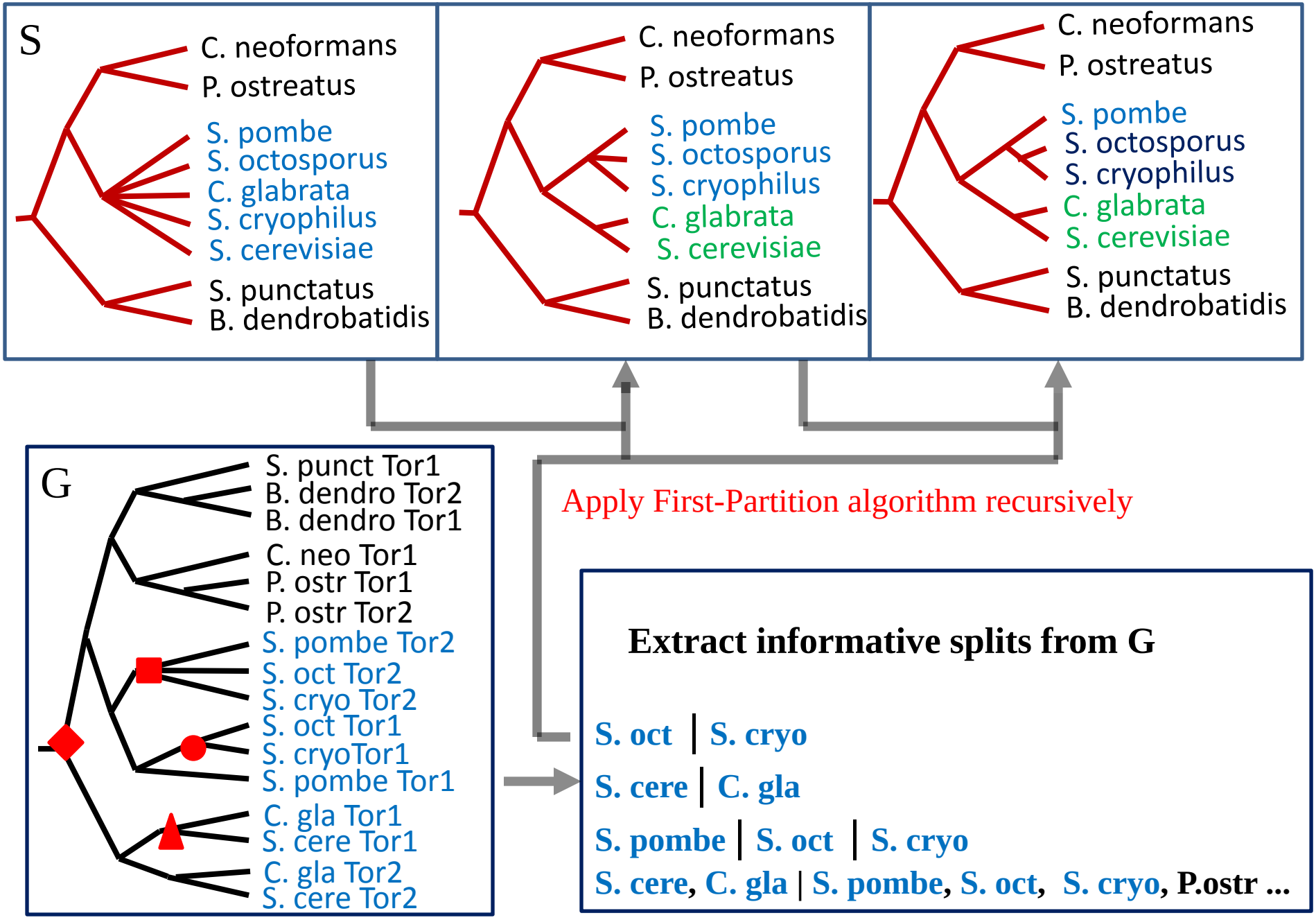
Instance: A (binary or non-binary) G and a non-binary S.
Solution: A binary refinement $\hat{S}$ of S that minimizes a reconciliation cost of reconciling G and $\hat{S}$.

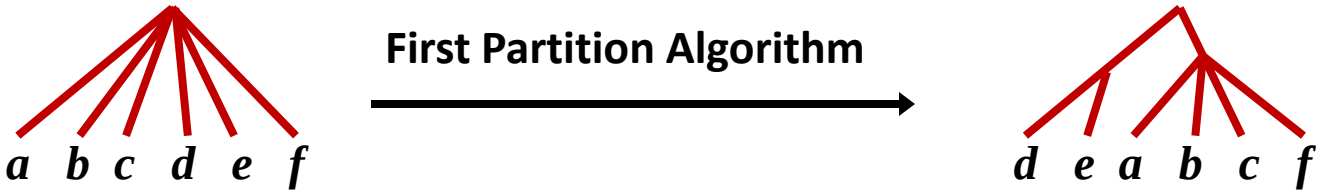
The above refinement problem is NP-hard even for binary gene trees, which is proved for the duplication cost via a reduction from

Species tree problem

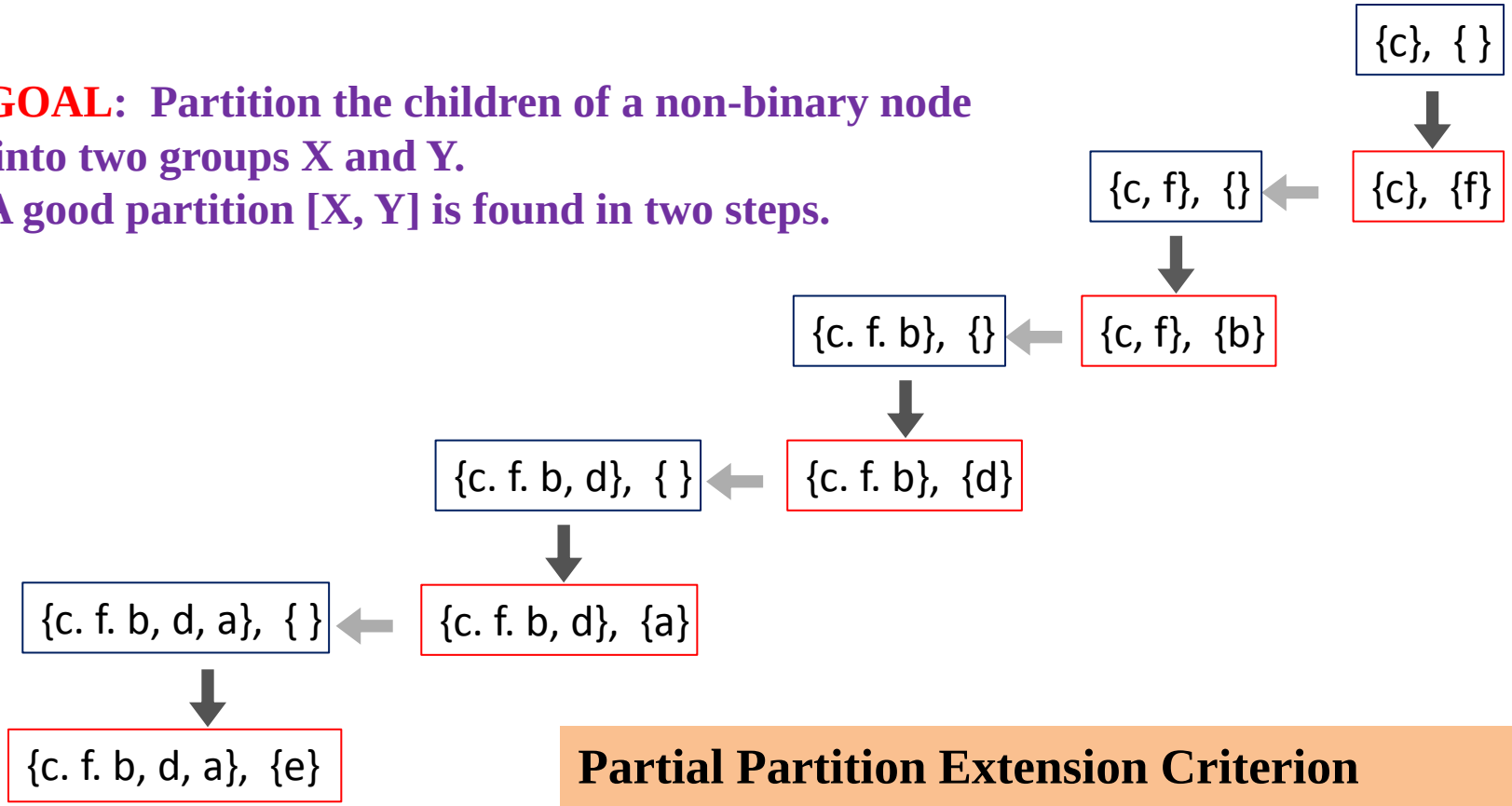
Instance: A set of gene trees G_i ($0<i<n$).
Solution: A species tree S' that minimizes $\sum_{0<i<n} c(S', G_i)$

Ma, Li, & Zhang, SIAM J. Computing, 2000
Zhang, IEEE TCBB, 2011
Bansal & Shamir, IEEE TCBB, 2011
Than & Nakhleh, PLoS Comput. Biol. 2009


S
C. neoformans
P. ostreatus
S. pombe
S. octosporus
C. glabrata
S. cryophilus
S. cerevisiae
S. punctatus
B. dendrobatidis
C. neoformans
P. ostreatus
S. pombe
S. octosporus
S. cryophilus
C. glabrata
S. cerevisiae
S. punctatus
B. dendrobatidis
C. neoformans
P. ostreatus
S. pombe
S. octosporus
S. cryophilus
C. glabrata
S. cerevisiae
S. punctatus
B. dendrobatidis
G
S. punct Tor1
B. dendro Tor2
B. dendro Tor1
C. neo Tor1
P. ostr Tor1
P. ostr Tor2
S. pombe Tor2
S. oct Tor2
S. cryo Tor2
S. oct Tor1
S. cryoTor1
S. pombe Tor1
C. gla Tor1
S. cere Tor1
C. gla Tor2
S. cere Tor2
Apply First-Partition algorithm recursively
Extract informative splits from G
S. oct | S. cryo
S. cere | C. gla
S. pombe | S. oct | S. cryo
S. cere, C. gla | S. pombe, S. oct, S. cryo, P.ostr ...



GOAL: Partition the children of a non-binary node into two groups X and Y.
A good partition [X, Y] is found in two steps.

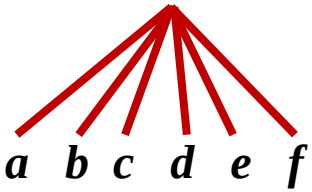


Step 1: Find k partial partitions (in red)

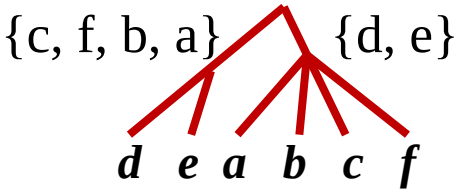
Partial Partition Extension Criterion

[P, Q] maximizes the number of 'non-cut' splits $A_1|A_2$, which have the following property:

$$A_i \cap P = \varphi \text{ or } A_i \cap Q = \varphi \text{ for every i} = 1,2$$



First Partition Algorithm



Step 2: Extend every partial partition found in Step 1 into a partition.

{c}, { }

{c}, {f}

{c, f}, {}

{c}, {f, a}

{c, f}, {b}

{c. f. b}, {}

{c, e}, {f, a}

{c. f. b}, {d}

{c. f. b, d}, { }

{c, e, b}, {f, a}

{c. f. b, d}, {a}

{c. f. b, d, a}, { }

{c, f, b}, {a, d, e}

{c, e, d, b}, {f, a}

{c. f. b, d, a}, {e}

{c, f, b, e, d}, {a}

{c, f, b, a}, {d, e}

Performance of First-Partition Algorithm

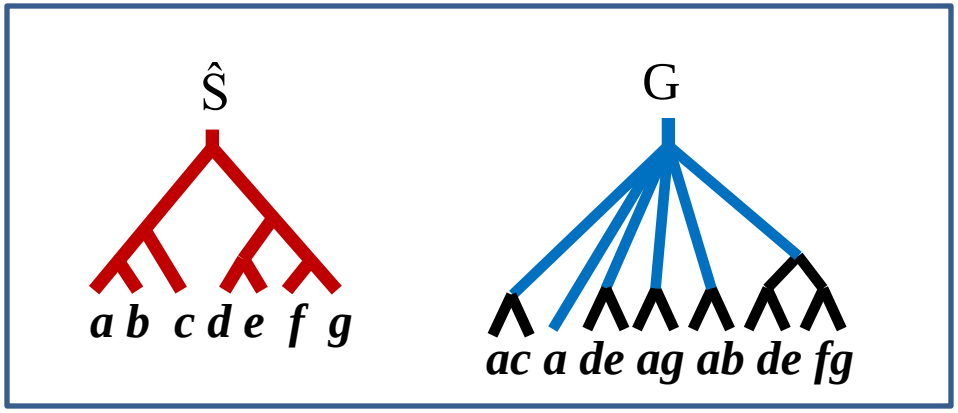
We repeat the following test 1000 times for each of 8 combinations of k (# of splits over a ground set) and t (the size of the ground set)

# of elements (t)	# of splits (k)	# of errors by FP†	# of errors by HC ‡
5	5	7	15
	10	0	18
10	5	0	4
	10	1	2
	20	0	0
15	7	0	3
	15	0	1
	30	0	1

† An algorithm made an error if it output a non-optimal partition on the input.

‡ HC is an algorithm designed through a reduction to the Min Hypergraph Cut problem (**Ouangraoua, Swenson, Chauve, 2009**).

Resolving non-binary nodes in G based on Ŝ



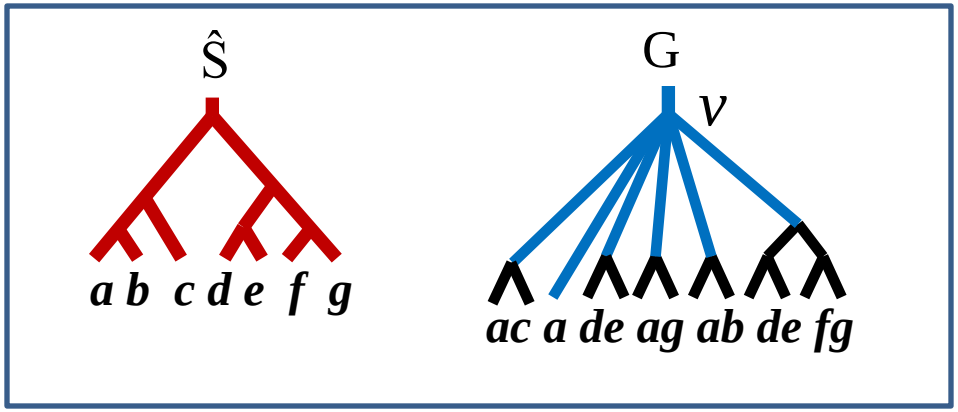
- The following duplication inference rule does not work for non-binary nodes:

A duplication is associated with u having children u_1 and u_2 iff

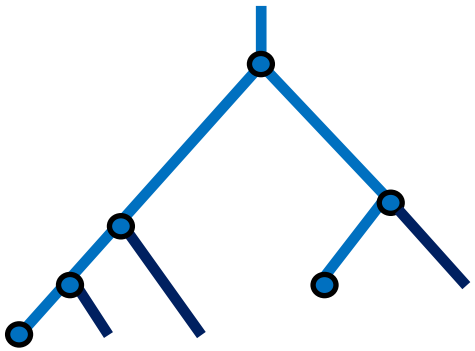
$$\lambda(u) = \lambda(u_1), \text{ or } \lambda(u) = \lambda(u_2).$$

- We present an extension of above rule to non-binary nodes. The whole process takes O(|G|+|Ŝ|) time.

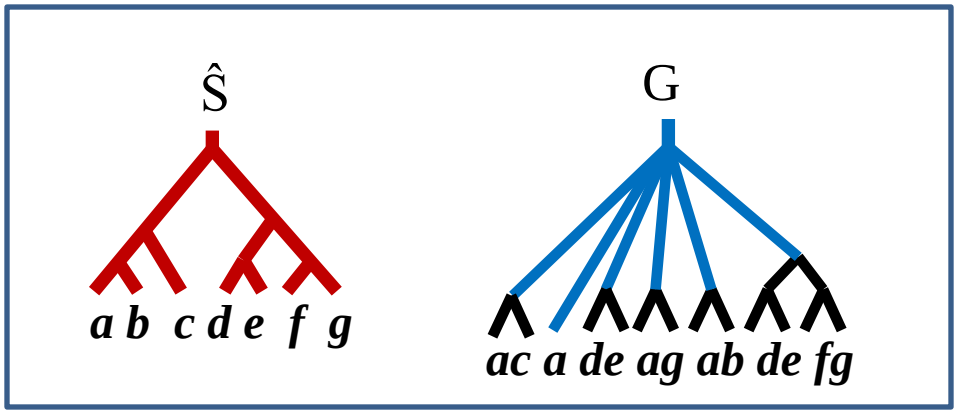
λ: The lca reconciliation of G and Ŝ

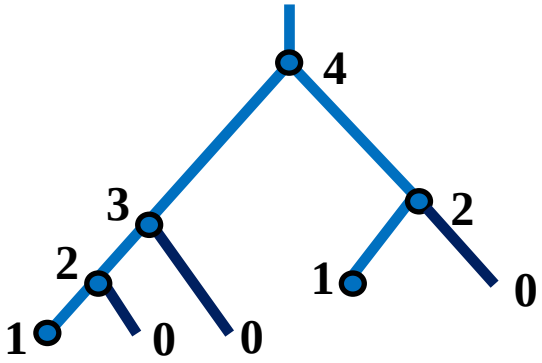


- The node *v* and its children are mapped to a subtree (in blue) under λ, which is expanded into a binary subtree (by adding dark blue edges).



The image subtree in Ŝ



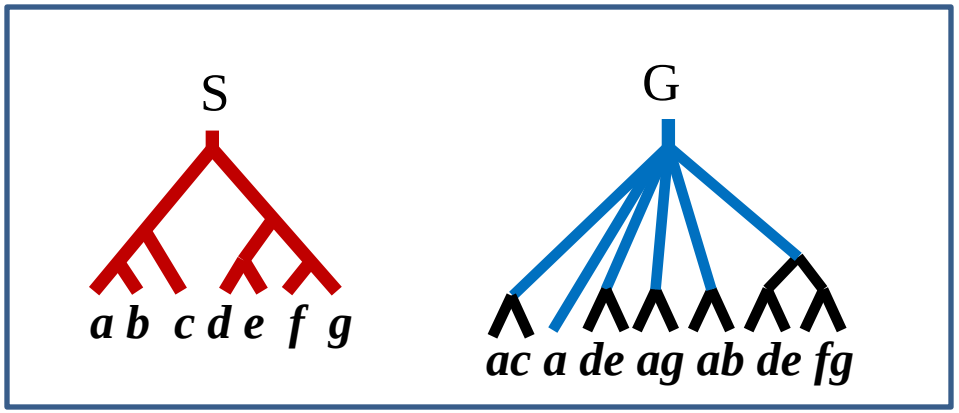


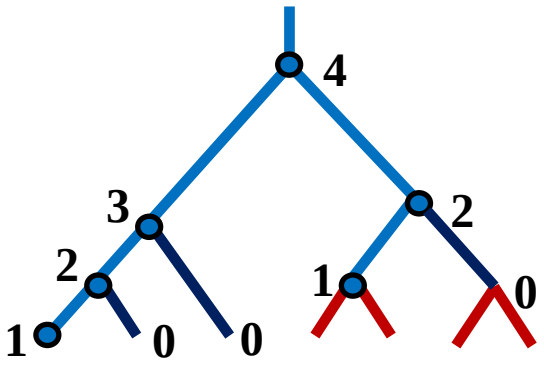
Step1: Compute ***m*(*u*)**, the maximum number of child images on a path from an internal node ***u*** to a leaf descendant.

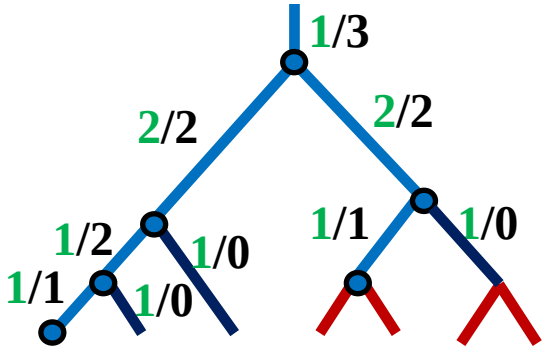
Algorithm

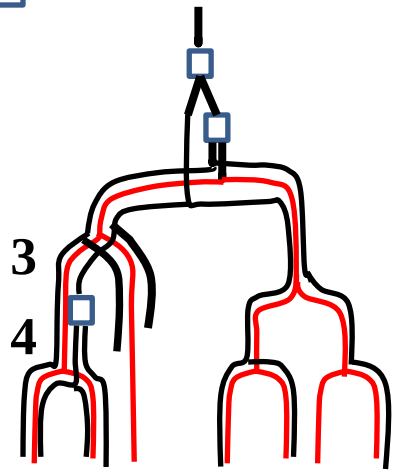
ω(u) is the # of children mapped to *u*.

$$m(u) = \max\{\ m(u_1), m(u_2)\} + \omega(u)$$









Step2: Compute ***α(u)*** / ***β(u)*** using *m*(*u*).

Algorithm

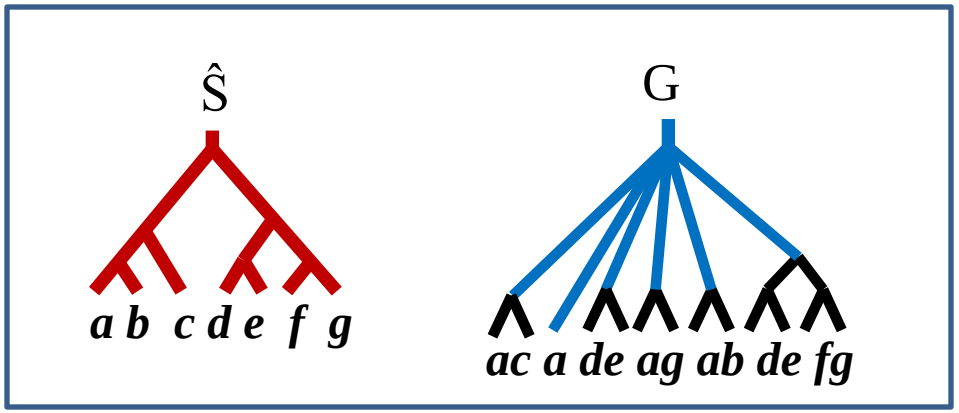
$$\alpha(r) = 1$$

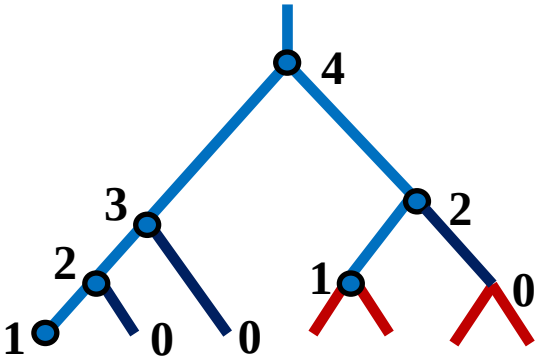
$$\alpha(u) = \beta(p(u)) - \omega(p(u))$$

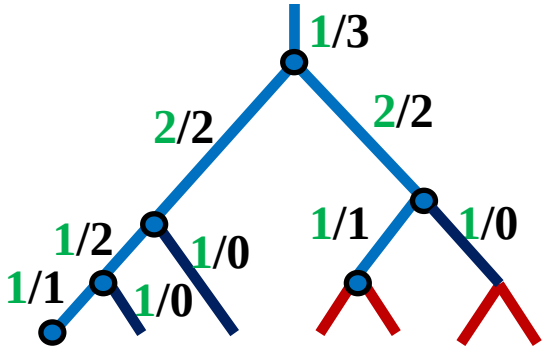
$$\beta(u) = \begin{cases} m(u) \text{ if } \alpha(u) \geq m(u) \text{ or } u \text{ is a leaf;} \\ \max\{\alpha(u), \min\{m(u_1), m(u_2)\} + \omega(u), 1 + \omega(u)\}. \end{cases}$$

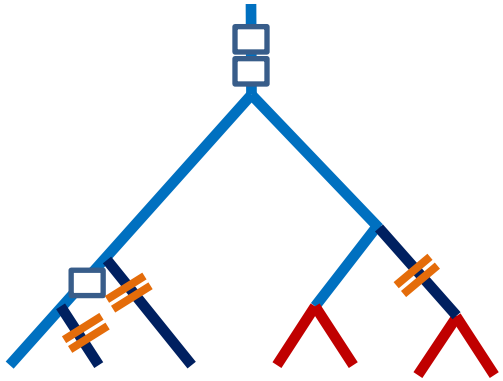
α(u): the # of genes entering a branch.

***β(u)*:** the # of genes leaving a branch.









Step1: Compute ***m*(*u*)**, :

Step2: Compute ***α*(*u*)** / ***β*(*u*)** using *m*(*u*).

Step 3: Infer duplications and losses.

If $\alpha(u) < \beta(u)$, duplications (▭) are postulated.

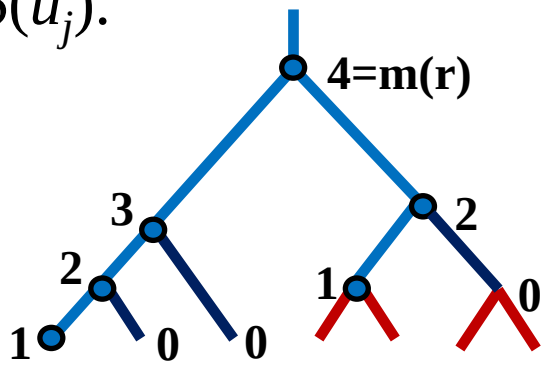
If $\alpha(u) > \beta(u)$, losses (〓) are postulated.

Theorem **The above algorithm resolves a non-binary node *v* with *m*(r)-1 duplications.**

Sketch of Proof. Assume the following path from the root to a leaf contains the largest number, *m*(r), of child images,

$$P : r = u_0, u_1, \cdots u_k$$

(1) There are no gene losses on P, i.e., $\alpha(u_j) <= \beta(u_j)$.
(2) All the duplications are postulated on P.



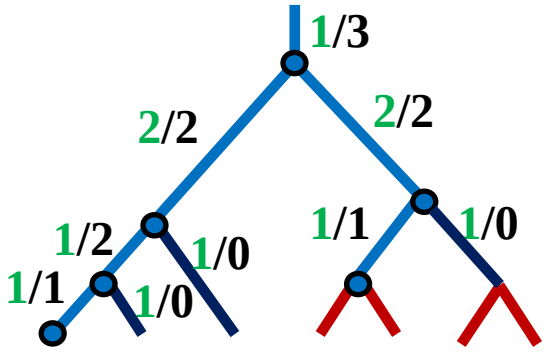
$$\#\text{ of duplications}$$

$$= \sum_{i=0}^{k} [\beta(u_i) - \alpha(u_i)]$$

$$= \beta(u_0) - 1 + \sum_{i=1}^{k} [\beta(u_i) - \beta(u_{i-1}) + \omega(u_{i-1})]$$

$$= \sum_{i=0}^{k} \omega(u_i) - 1$$

$$= m(\lambda(r)) - 1.$$



Thm (i) The obtained reconciliation of a non-binary node v has the optimal dup. cost.

(ii) It also has the smallest loss cost over all the reconciliations with the same duplication cost.

Idea of Proof. Let v have children $v_1, v_2, \ldots v_k$. We consider partially ordered set:

$$\mathcal{P} = (\{L(\lambda(v_i)) : 1 \leq i \leq k\}, \subseteq).$$

(1) L: The size of the longest chain in $\mathcal{P}$

P: The min. # of antichains into which $\mathcal{P}$ may be partitioned.

Dual of Dilworth Theorem (Mirsky, 1971): L=P.

(2) At least p-1 duplications are needed to produce all the children of v (Berglund-Sonnhammer et al, '06, Chang & Eulenstein'06)

Part III: Software and Experiments

Our algorithms have been implemented in Python.

- Our program reconciles one or more gene trees and a species tree in the duplication cost, or the duplication and then loss cost.
- It is executed from command line to allow for automated analysis of large data sets.
- No limitation on the number of species.
- Automatically rerooting gene trees.

We validated our program on simulated and real data; we also compared it with NOTUNG (Durand et al'08), which requests either gene tree or species is binary.

Repeated the following experiment 1000 times for n=20, 40, 60, 80,100:

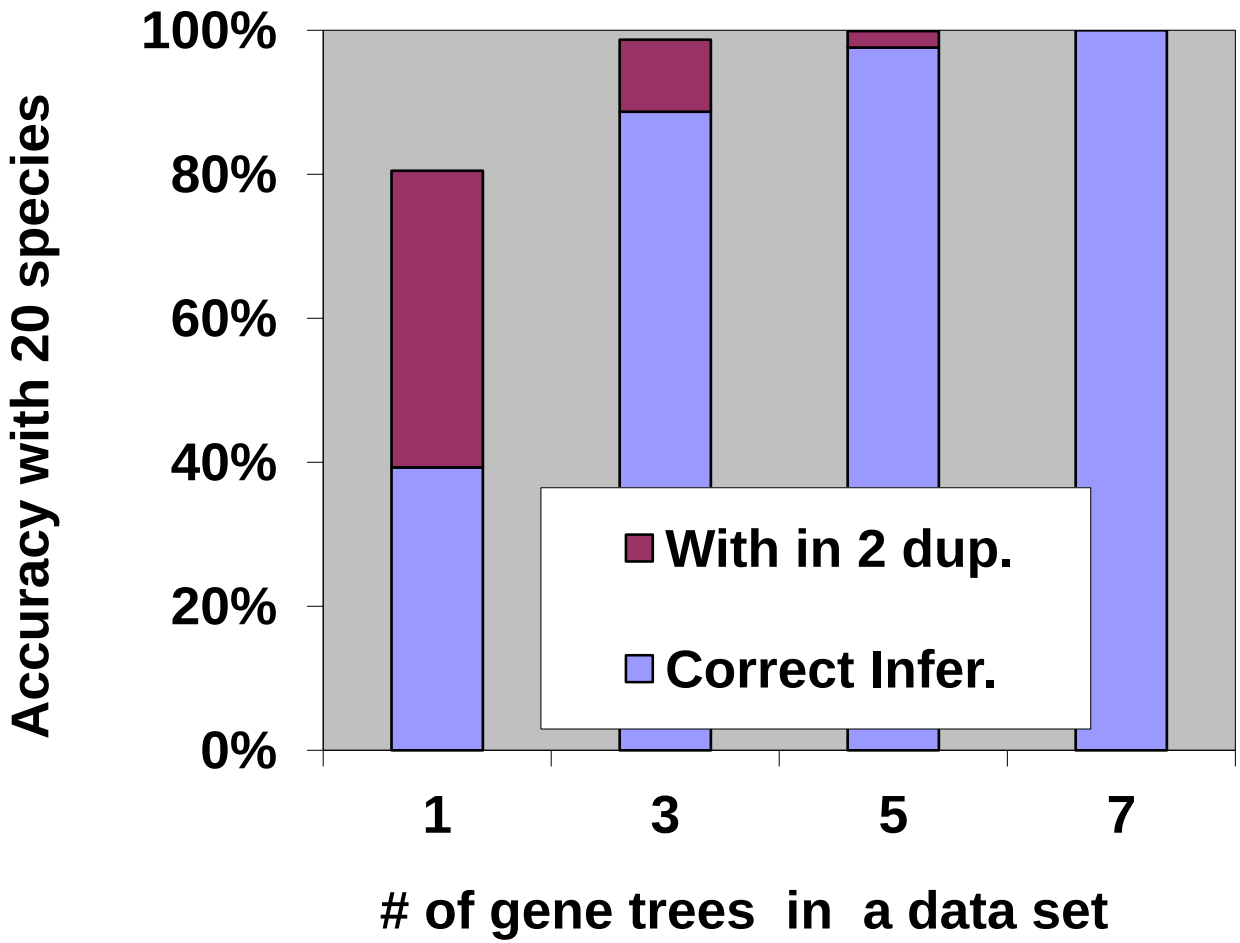
-- Generate a binary species tree S over n species, and a non-binary species tree S' from S by randomly contracting edges.

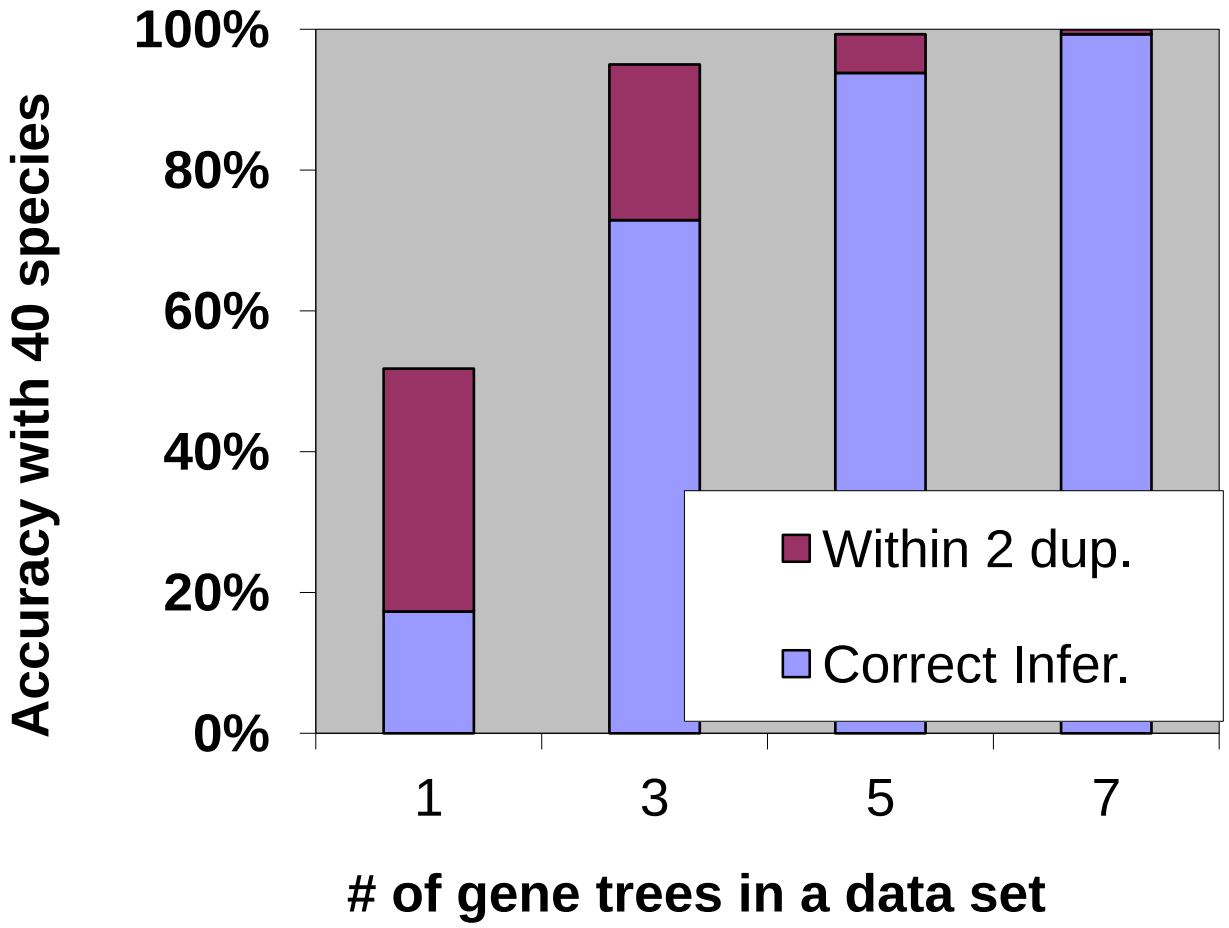
-- Generate 16 binary gene trees over totally about 1.2n genes in n species using proper duplication and gene loss rates in S; and divided them into 4 groups containing 1, 3, 5, 7 gene trees respectively.

-- Find the reconciliation of each group of gene trees and S'.

-- Check whether S and the refinement of S' are identical or not.

Performance Analysis




100%
80%
60%
40%
20%
0%
Accuracy with 40 species
Within 2 dup.
Correct Infer.
1
3
5
7
# of gene trees in a data set

We tested our program on

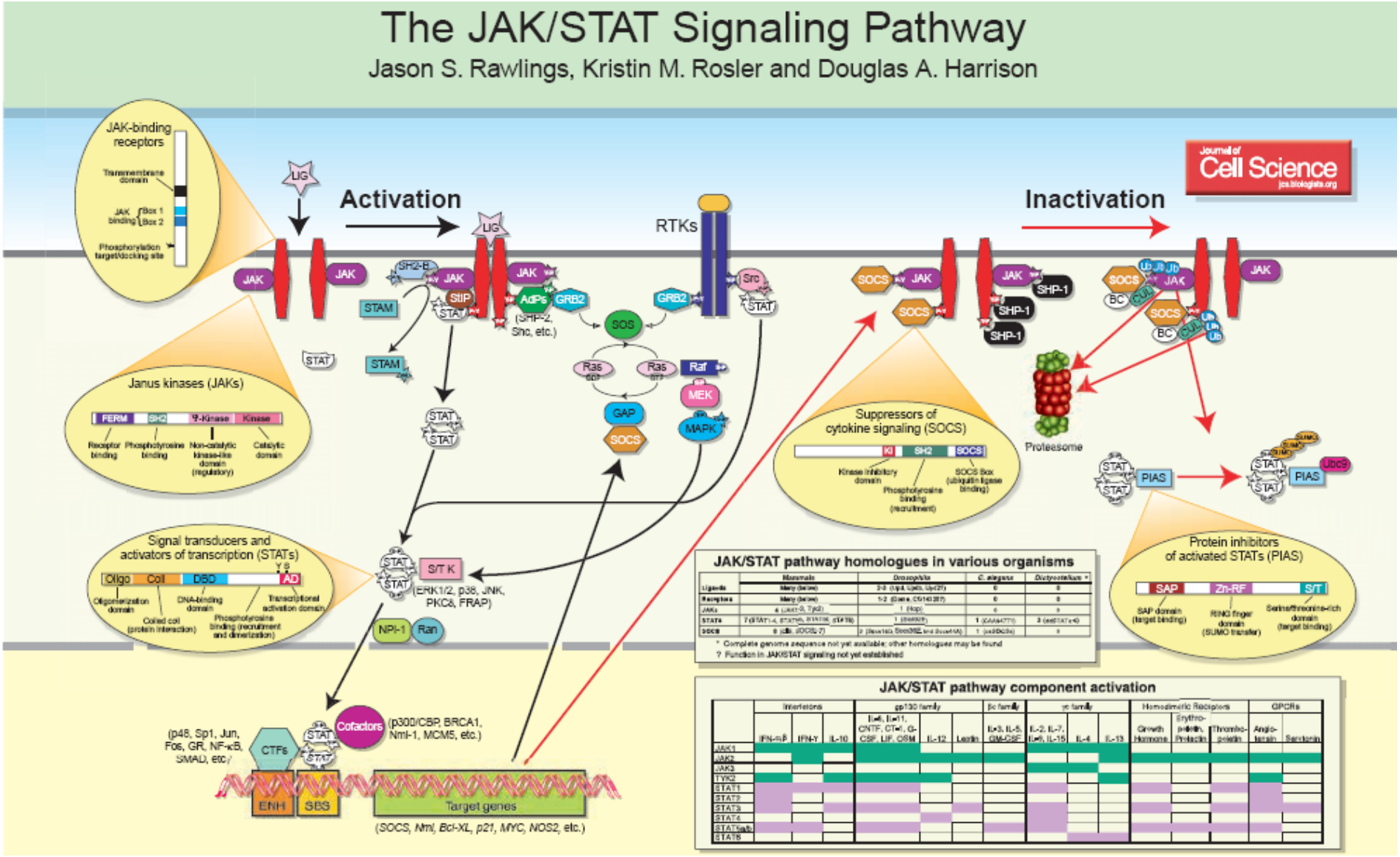
1. A gene tree of Tor in 13 fungal species (Shertz et al. 2011) and a non-binary species tree from NCBI taxonomy database.

   --- Inferred duplication events are identical to those reported in the paper;

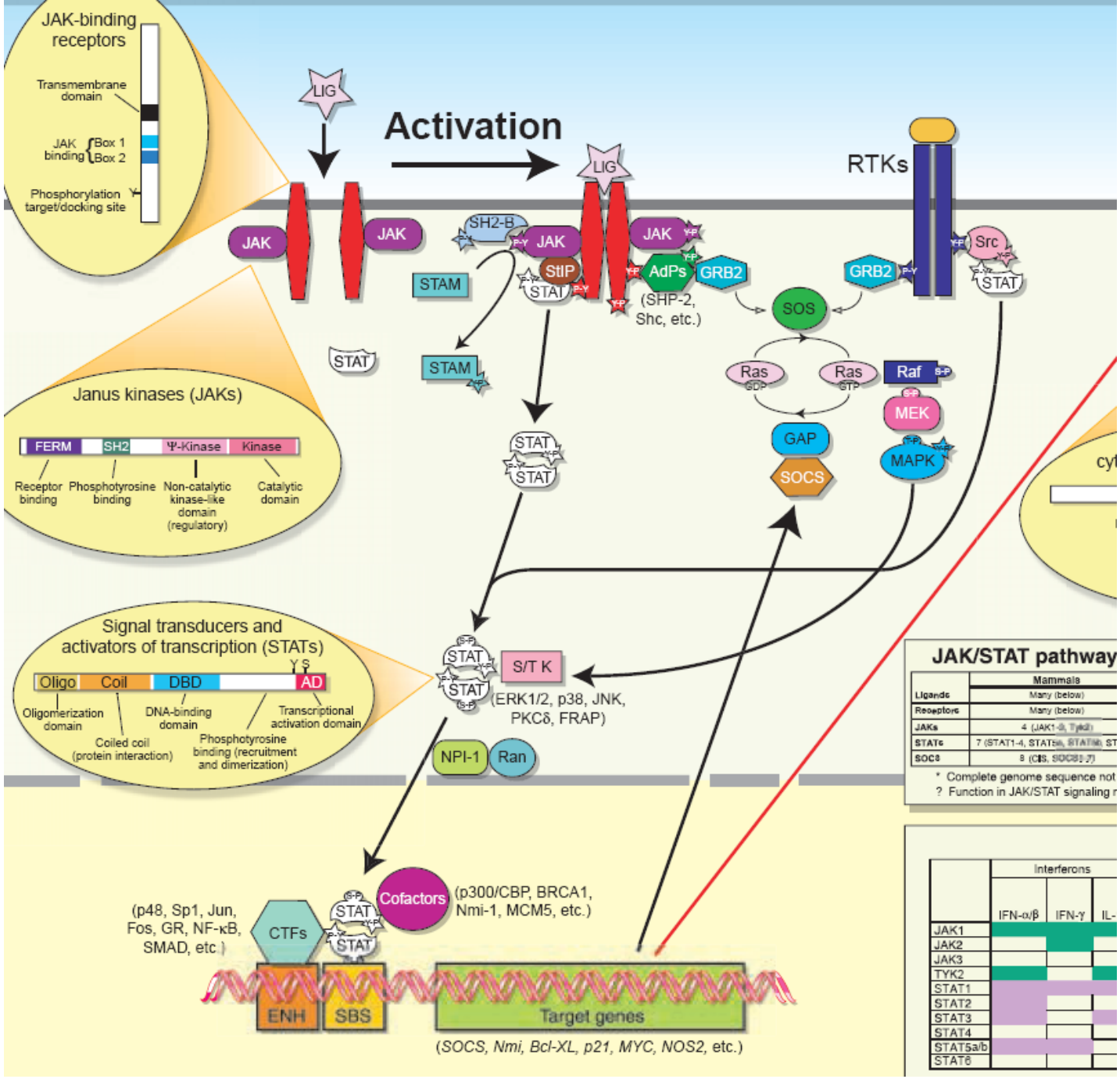
   --- Output refinement of the species tree is consistent with a large binary fungal species tree appearing in literature (www.broadinstitute.org)

2. A non-binary STAT gene tree and a binary species tree.

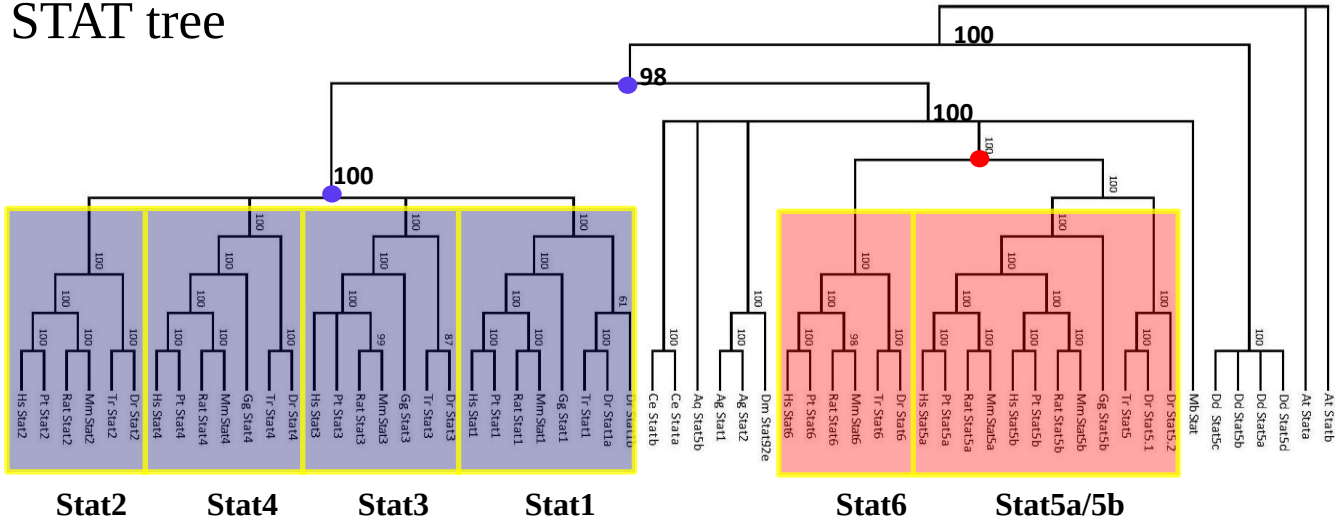
   --- Co-evolution of STAT and other proteins in its signaling pathway.



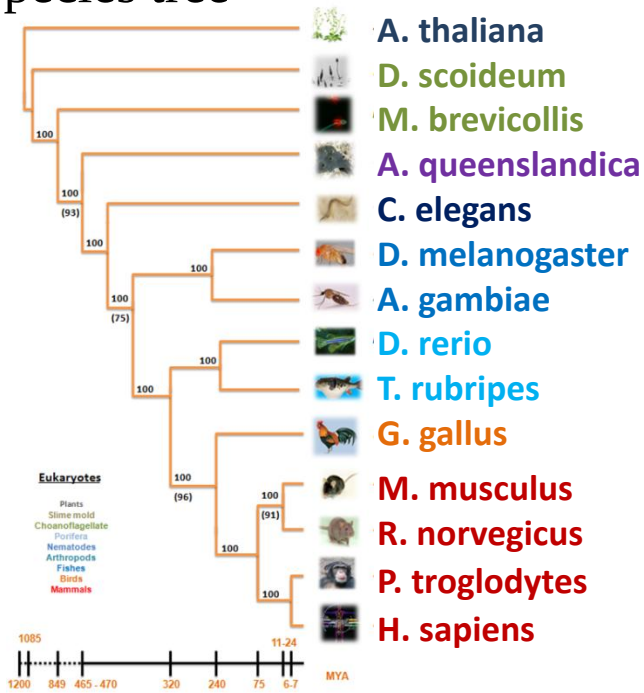
STAT: Signal Transducer and Activator of Transcription


JAK-binding receptors
Transmembrane domain
JAK binding
Box 1
Box 2
Phosphorylation target/docking site
LIG
Activation
LIG
RTKs
JAK
JAK
SH2-B
JAK
JAK
STAM
StIP
STAT
AdPs
GRB2
(SHP-2, Shc, etc.)
GRB2
Src
STAT
SOS
STAT
STAM
Ras GDP
Ras GTP
Raf
MEK
MAPK
GAP
SOCS
Janus kinases (JAKs)
FERM
SH2
Ψ-Kinase
Kinase
Receptor binding
Phosphotyrosine binding
Non-catalytic kinase-like domain (regulatory)
Catalytic domain
STAT
STAT
Signal transducers and activators of transcription (STATs)
Oligo
Coil
DBD
AD
Y S
Oligomerization domain
Coiled coil (protein interaction)
DNA-binding domain
Phosphotyrosine binding (recruitment and dimerization)
Transcriptional activation domain
STAT
STAT
S/T K
(ERK1/2, p38, JNK, PKCδ, FRAP)
NPI-1
Ran
JAK/STAT pathway
Mammals
Ligands
Many (below)
Receptors
Many (below)
JAKs
4 (JAK1-3, Tyk2)
STATs
7 (STAT1-4, STAT5a, STAT5b, ST
SOCS
8 (CIS, SOCS1-7)
* Complete genome sequence not
? Function in JAK/STAT signaling
Interferons
IFN-α/β
IFN-γ
IL-
JAK1
JAK2
JAK3
TYK2
STAT1
STAT2
STAT3
STAT4
STAT5a/b
STAT6
(p48, Sp1, Jun, Fos, GR, NF-κB, SMAD, etc.)
CTFs
STAT
STAT
Cofactors
(p300/CBP, BRCA1, Nmi-1, MCM5, etc.)
ENH
SBS
Target genes
(SOCS, Nmi, Bcl-XL, p21, MYC, NOS2, etc.)

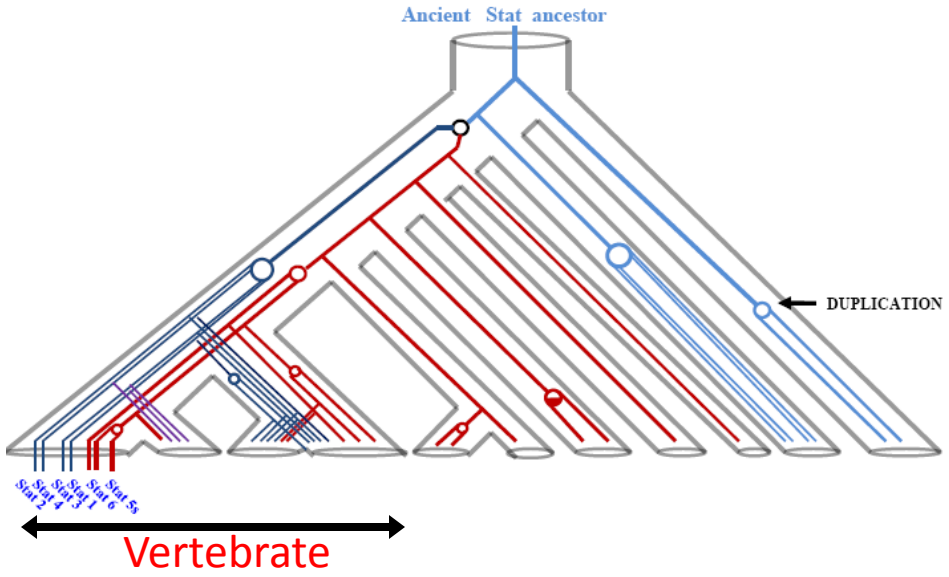
A: STAT tree

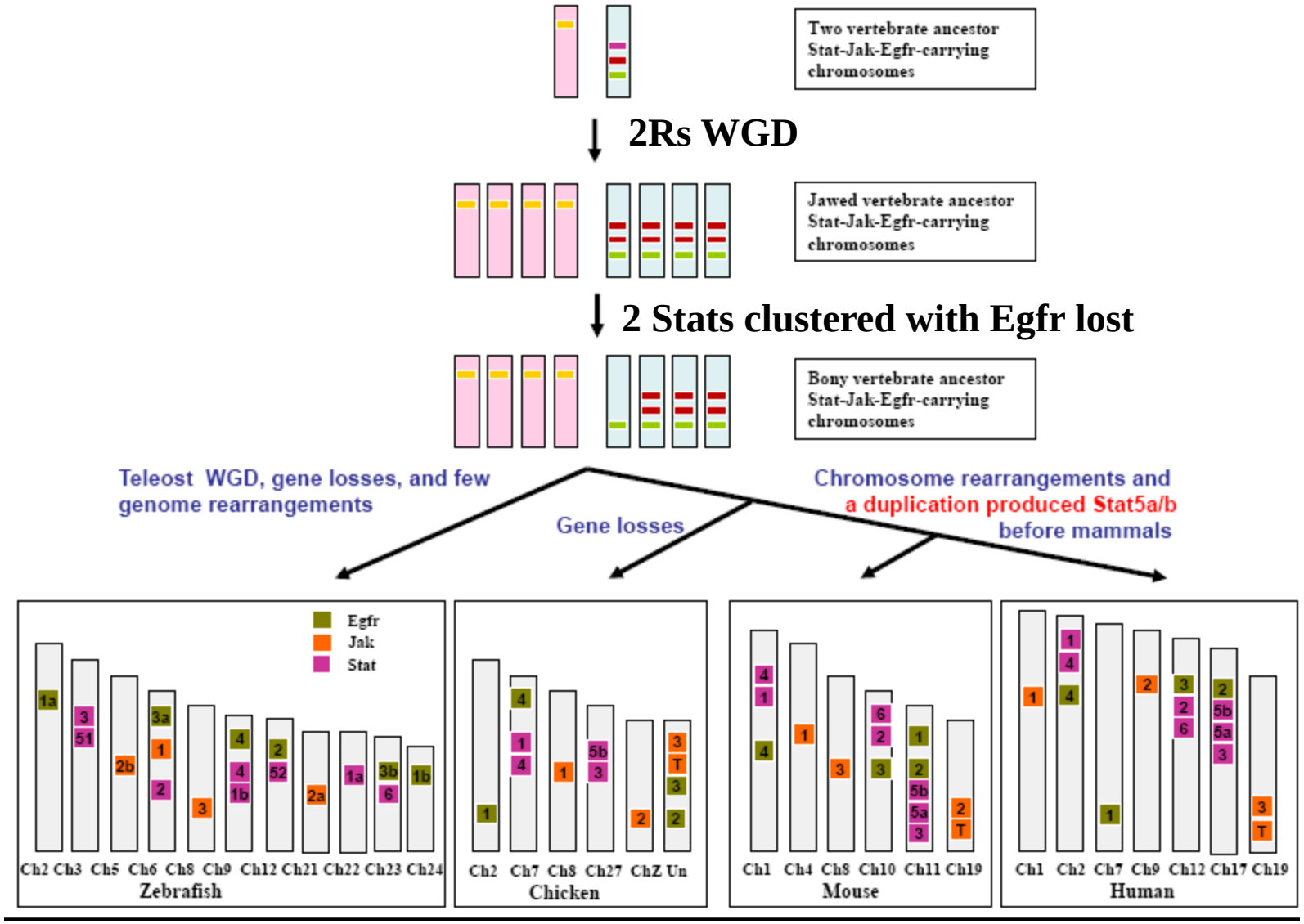


B: Species tree



C: Inferred gene evolution





Co-evolution of Egfr, Jak and STAT genes.

Nakatani, Takeda, Kohara, & Morishita, 2007

Conclusion

- We developed a software for reconciliation with non-binary trees

  -- For binary gene tree and non-binary species tree, our program output a reconciliation with much less duplications than NOTUNG.

  -- **Durand et al** (2005, 2008), **Chang & Eulenstein** (2006), **Berglund-Sonnhammer et al**.(2006)

- Parsimony approach vs Bayesian approach

  -- **Akerborg et al** (2009); **Arvestad et al**, (2009)

- Study how to reconcile non-binary gene tree and HGT (horizontal gene transfer) networks in future.

Acknowledgment

Reconciliation work:

-- Taoyang Wu
-- Yu Zheng

STAT evolution work:

-- Xin-Yuan Fu (Dept of Biochemistry)
-- C. Pawan Kumar Patro
-- Choong Yong Ung

---------- Thank You! ----------