Deciphering Reticulate Evolution Using Phylogenetic Reconciliation

Mukul S. Bansal

Department of Computer Science and Engineering,
University of Connecticut,
USA

July 9, 2014

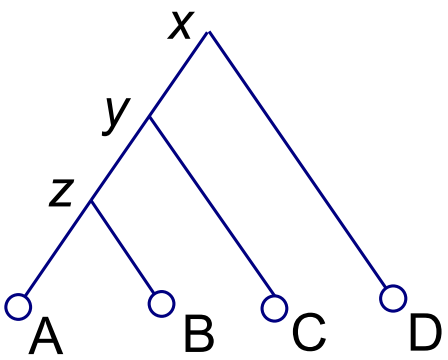
Gene Family Evolution

Problem

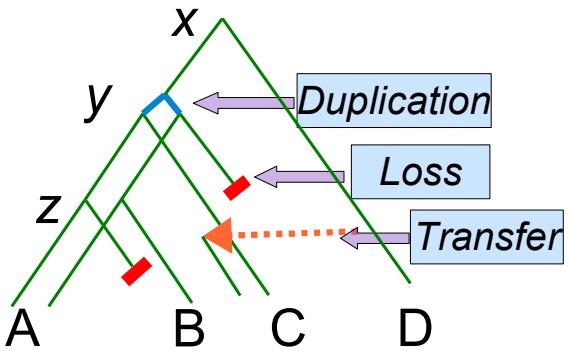
How did any given gene family evolve?

- Gene families evolve inside species trees.
- Affected by evolutionary events such as gene duplication, horizontal gene transfer, and gene loss.

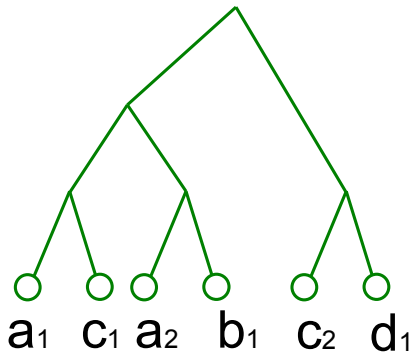
Species Tree *S*



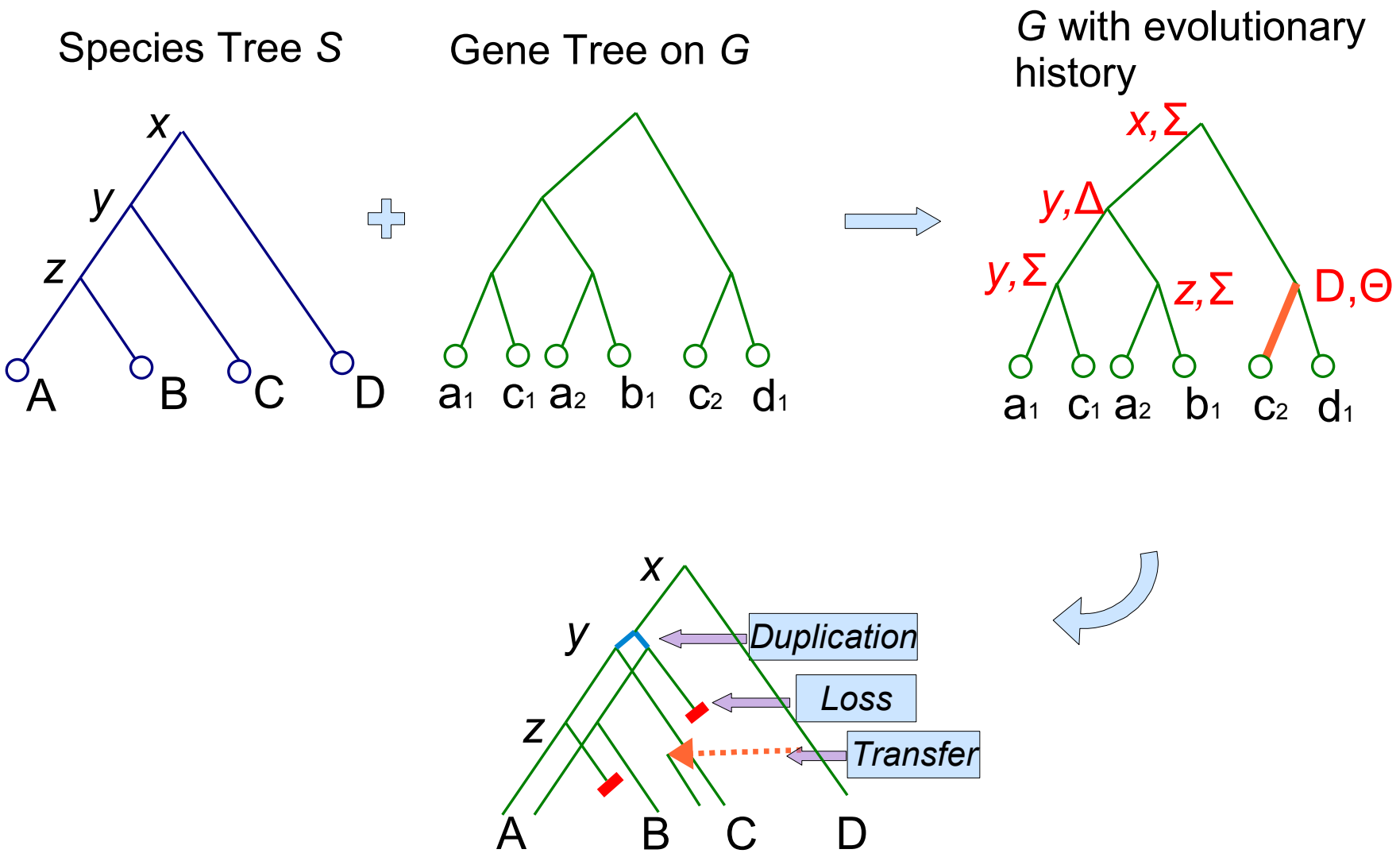
Evolution of Gene Family *G*



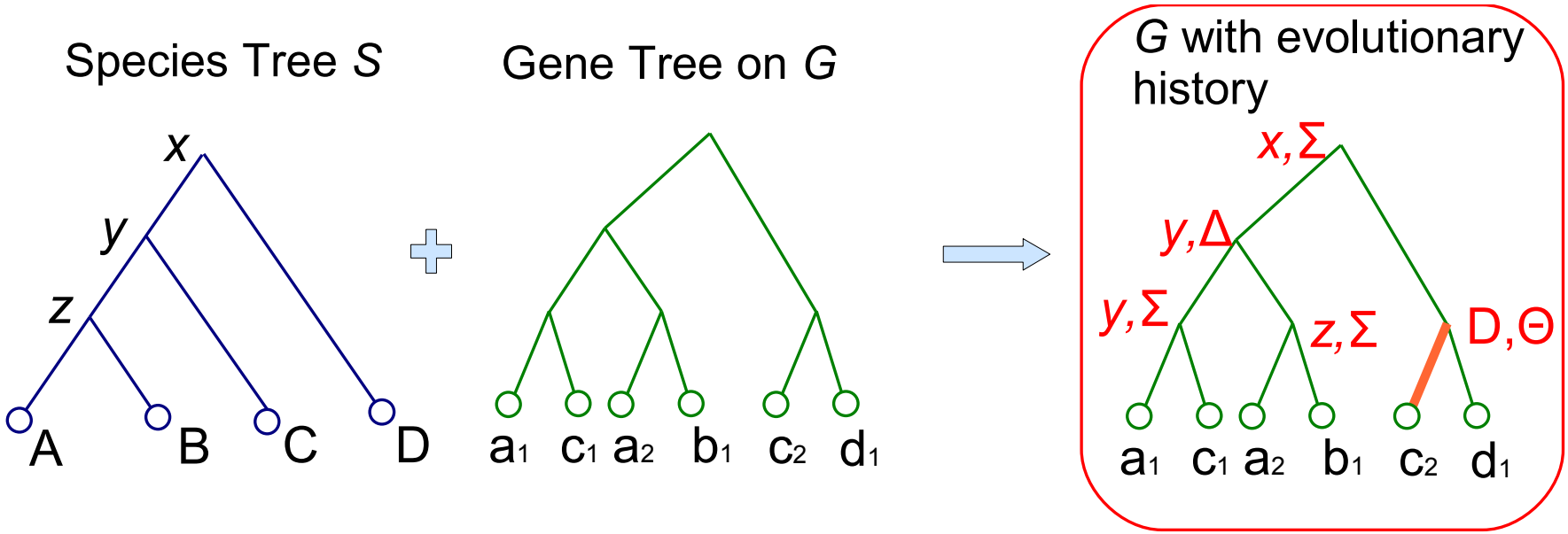
Gene Tree on *G*



Definition: DTL Reconciliation



Definition: DTL Reconciliation



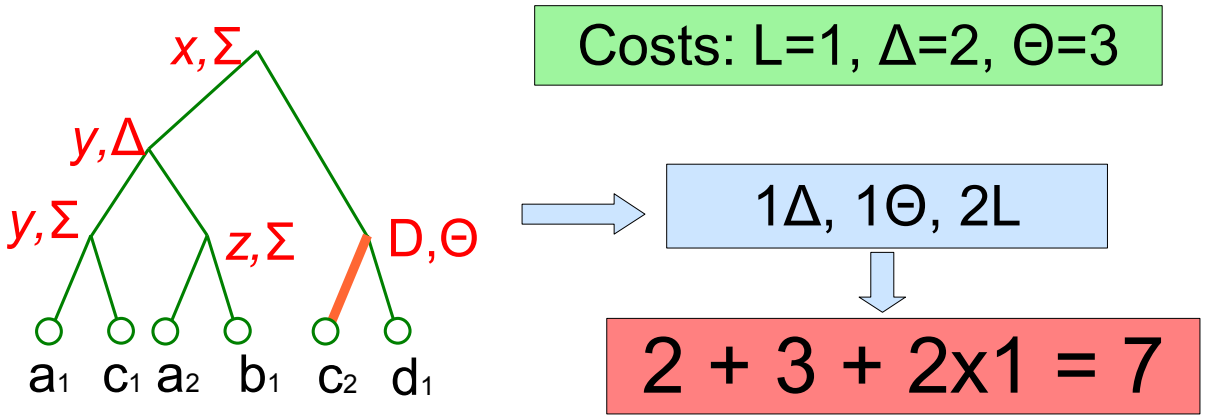
Input: A gene tree for that gene family, and a trusted rooted species tree.

Output: An evolutionary history of that gene family showing horizontal gene transfers, gene duplications, losses, and speciation events.

DTL Reconciliation Problem Formulation

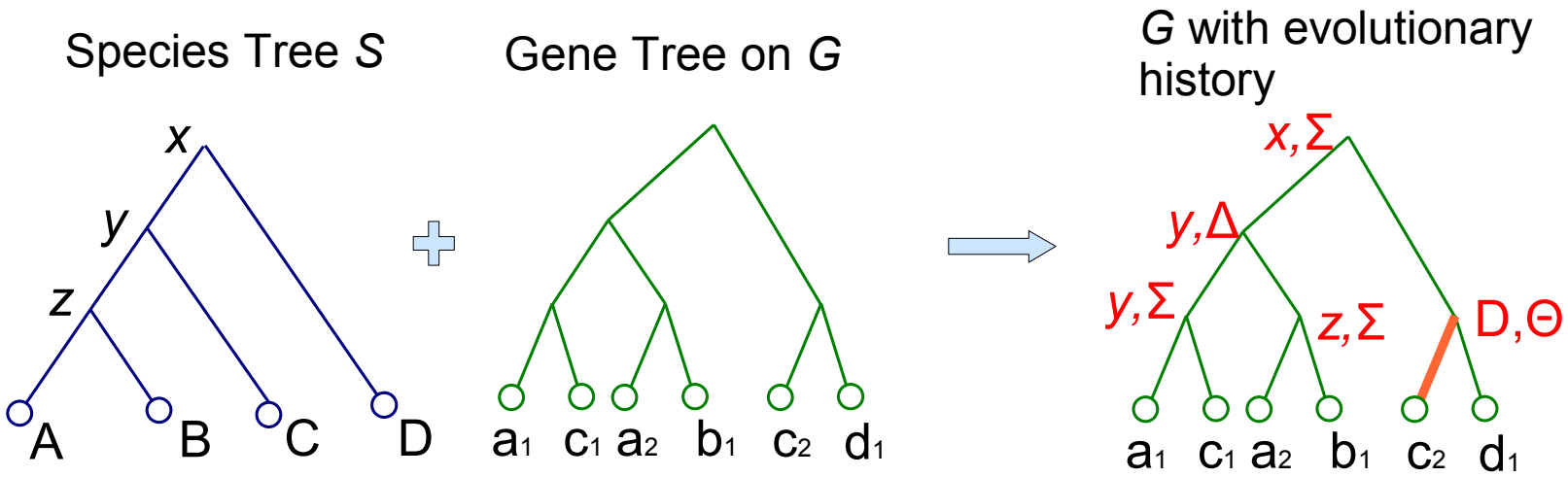
Typical formulation:

- Costs are assigned to duplications, transfers, and losses.
- Goal: Find the reconciliation that minimizes the total cost.
- Easy to compute cost for a given reconciliation.



Different reconciliation could have different cost.

Applications of DTL Reconciliation



- Understanding how gene families evolve.
- Dating gene birth.
- Inferring orthologs/paralogs/xenologs.
- Accurate gene tree reconstruction.
- Whole genome species tree construction.
- Constructing species phylogenetic networks.

Traditional Bottlenecks of DTL Reconciliation

1. Slow algorithms: Fastest algorithms had cubic time complexity.
   - Could not study large gene families.
   - No gene tree reconstruction or species tree reconstruction.
2. Inaccurate gene trees: Accuracy of reconciliation deeply impacted by accuracy of gene trees.
3. Multiple optima: There are often multiple optimal solutions.
   - Unclear how to interpret single reconciliation.
   - Algorithms to output all optimal reconciliations have exponential time complexity.
4. Event costs: Event costs impact reconciliations.
   - What is the "correct" event cost assignment.

Recent Advances

Addressing the bottlenecks.

(a) Asymptotically faster algorithms.

(b) Method for accurate prokaryotic gene tree reconstruction.

(c) Handling multiple optima by uniform sampling.

(d) Techniques and tools for studying impact of event costs.

(a) Faster Algorithms

Our Results

For undated species tree:

- $O(mn)$-time algorithm.
- Factor of n speedup.

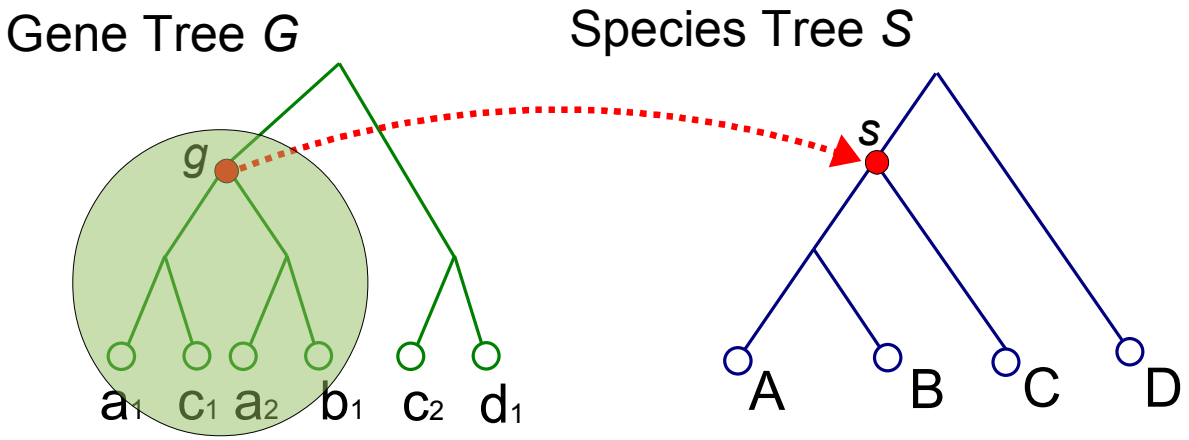
For dated species trees:

- $O(mn \log n)$-time algorithm for the fully dated version. (Transfers restricted to co-existing species).
- Factor of $n/\log n$ speedup.

For improved accuracy:

- General $O(mn^2)$-time algorithm that can handle distance-dependent transfer costs.
- Factor of n speedup.

Dynamic Programming Algorithm



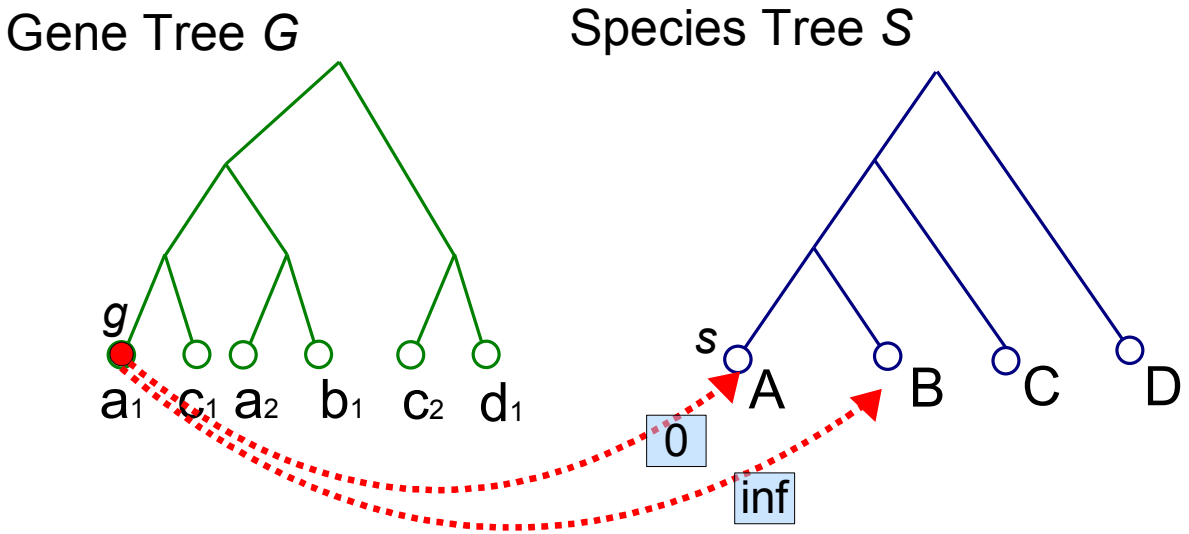
Given any $g \in V(G)$ and $s \in V(S)$, let

- $c(g, s)$ = cost of an optimal reconciliation of $G(g)$ with S such that g is mapped to s.

Similarly, define:

- $c_{\Sigma}(g, s)$: with restriction that g is speciation.
- $c_{\Delta}(g, s)$: with restriction that g is duplication.
- $c_{\Theta}(g, s)$: with restriction that g is transfer.

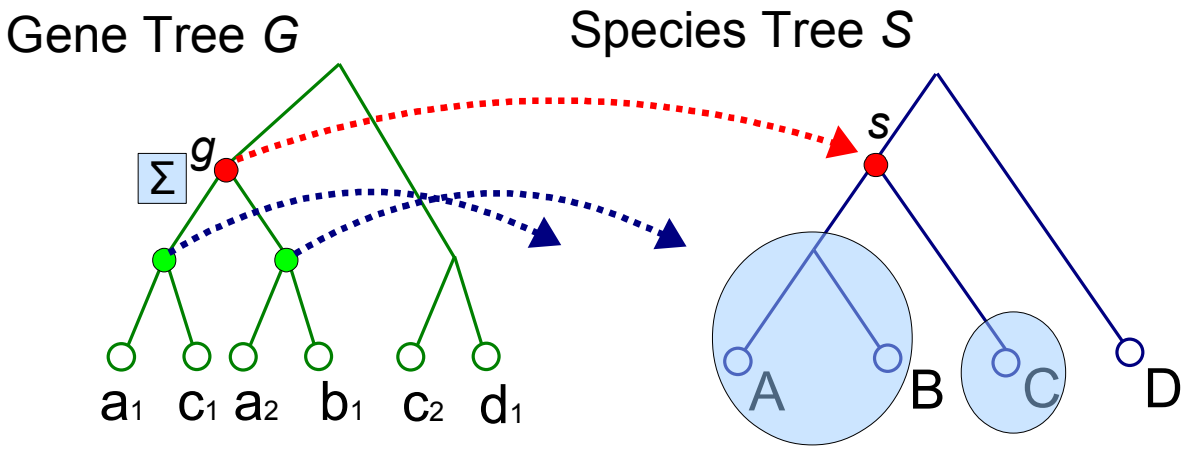
Dynamic Programming Algorithm



$$c(g,s) = \begin{cases} 0 & \text{if } g \in Le(G) \text{ and } s = \mathcal{M}(g) \\ \infty & \text{if } g \in Le(G) \text{ and } s \neq \mathcal{M}(g) \\ \min\{c_{\Sigma}(g,s), c_{\Delta}(g,s), c_{\Theta}(g,s)\} & \text{otherwise.} \end{cases}$$

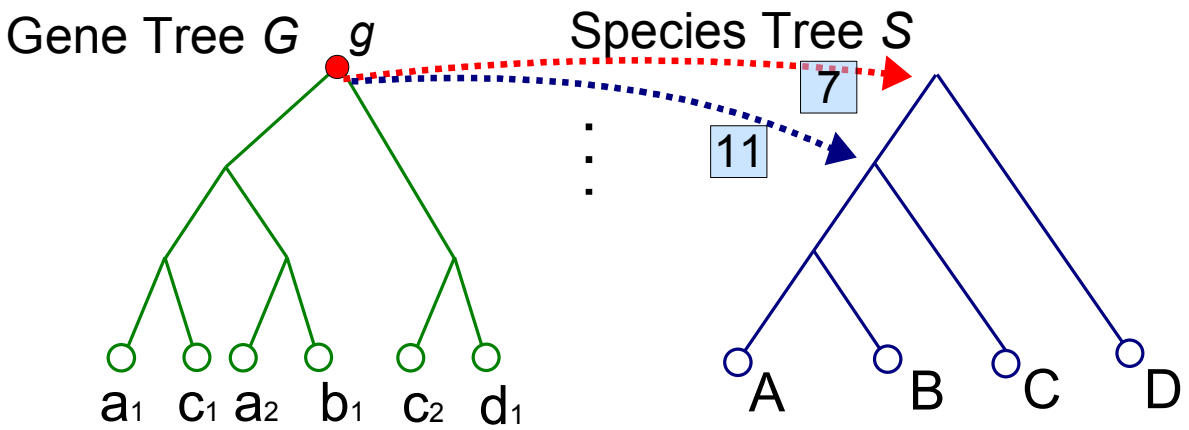
DP Algorithm: Nested Post-Order Traversal

- Nested post-order traversal of the gene tree and species tree to compute all values ($c_\Sigma(g,s)$,$c_\Delta(g,s)$,$c_\Theta(g,s)$,$c(g,s)$).



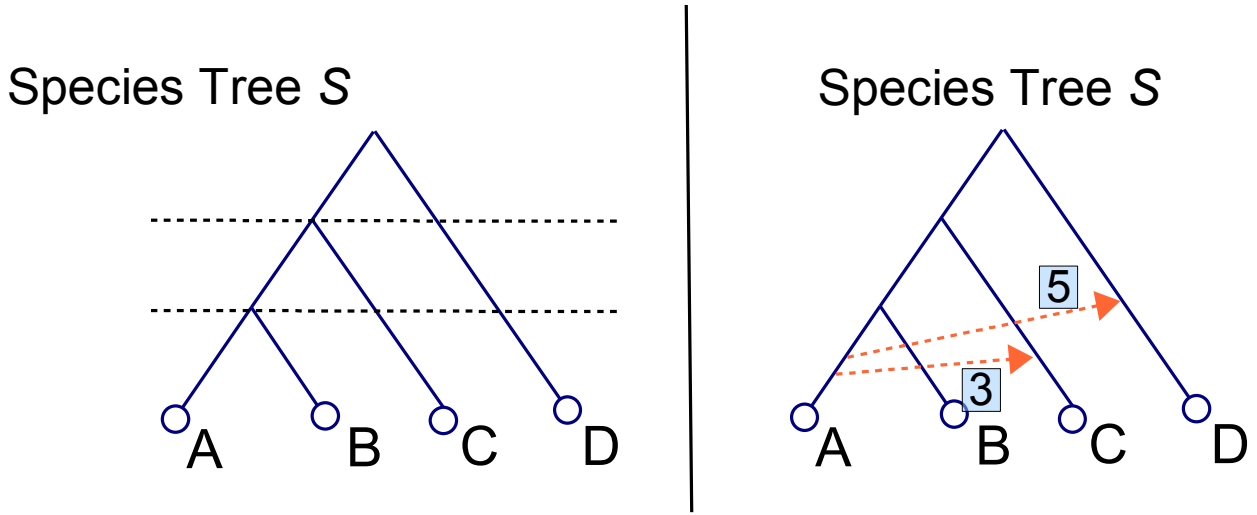
- Source of speedup is to compute each value in constant time.

DP Algorithm: Termination



- The optimal reconciliation cost of G and S is simply: $\min_{s \in V(S)} c(rt(G), s)$

Dynamic Programming Algorithm



Dated species trees:

- Places constraints on transfers. Breaks clean structure.
- Constant time slows to $\log n$ time.

Distance dependant transfer costs:

- Each potential transfer event must be evaluated separately.
- Constant time slows to $O(n)$ time.

Drastic Improvement in Runtime and Scalability

Dataset Type	Dataset Size	RANGER-DTL-U	AnGST	Mowgli
Simulated	50 taxa (100 datasets)	2s	3m:26s	28m:30s
	100 taxa (100 datasets)	3s	15m:4s	3h:52m
	200 taxa (100 datasets)	9s	1h:2m	29h:43m
	500 taxa (100 datasets)	35s	>800h	>400h
	1,000 taxa (100 datasets)	2m:57s	–	>6,000h
	10,000 taxa (1 dataset)	4h:7m	–	–
Biological	4,733 gene trees, 100 taxa	1m:03s	3h:45m	41h:36m

- 50 taxa: 3m/28m $\rightarrow$ 2s
- 100 taxa: 15m/3h $\rightarrow$ 3s
- 200 taxa: 1h/29h $\rightarrow$ 9s
- 500 taxa: >800h/>400h $\rightarrow$ 35s
- 1000 taxa: –/>6000h $\rightarrow$ 3m
- 10000 taxa: –/– $\rightarrow$ 4h.

Faster Algorithms Enable Many New Applications

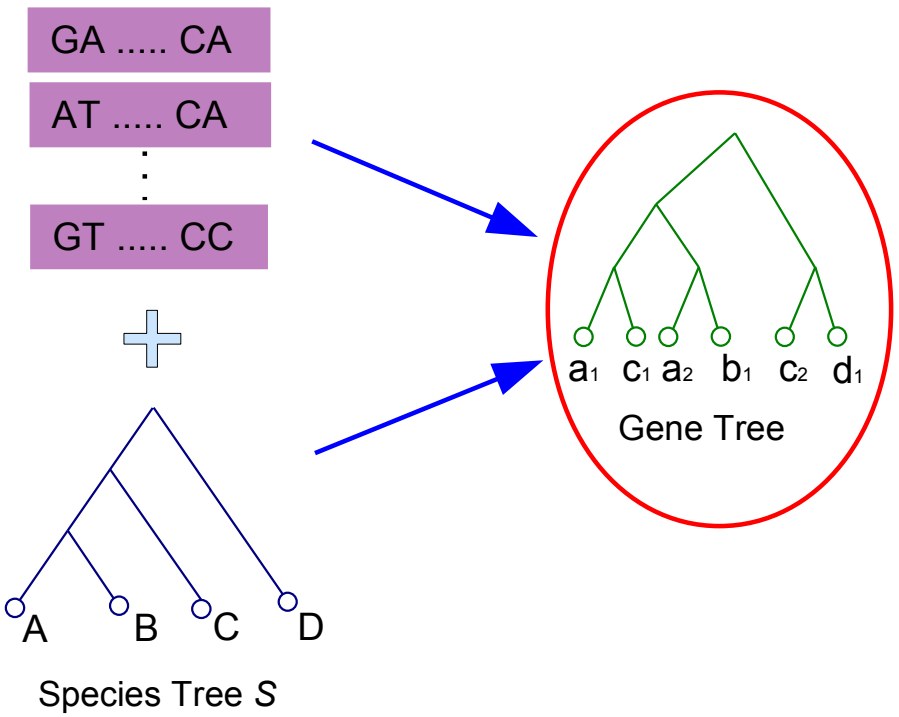
- Efficient genome-scale analysis.
- Much larger species tree.
- Species tree reconstruction.
- Accurate gene tree reconstruction.
- Phylogenetic network inference.

→ M. S. Bansal, E. J. Alm, and M. Kellis, "Efficient Algorithms for the Reconciliation Problem with Gene Duplication, Horizontal Transfer, and Loss". *Twentieth Annual International Conference on Intelligent Systems for Molecular Biology (***ISMB 2012***); Bioinformatics* 2012, 28: i283–i291.

(b) Accurate Gene Tree Reconstruction

Gene Tree Reconstruction

- Gene trees are hard to reconstruct accurately.
- Eukaryotes: TreeBest, SPIDIR, PrIME-GSR, SPIMAP, NOTUNG, tt, TreeFix.
- Prokaryotes: Recent efforts: AnGST and MowgliNNI. (Also ALE in probabilistic framework.)



Developing a Principled Approach

Goal:

- Use fast algorithms to develop an efficient and principled approach for prokaryotic gene tree reconstruction.

Outcome:

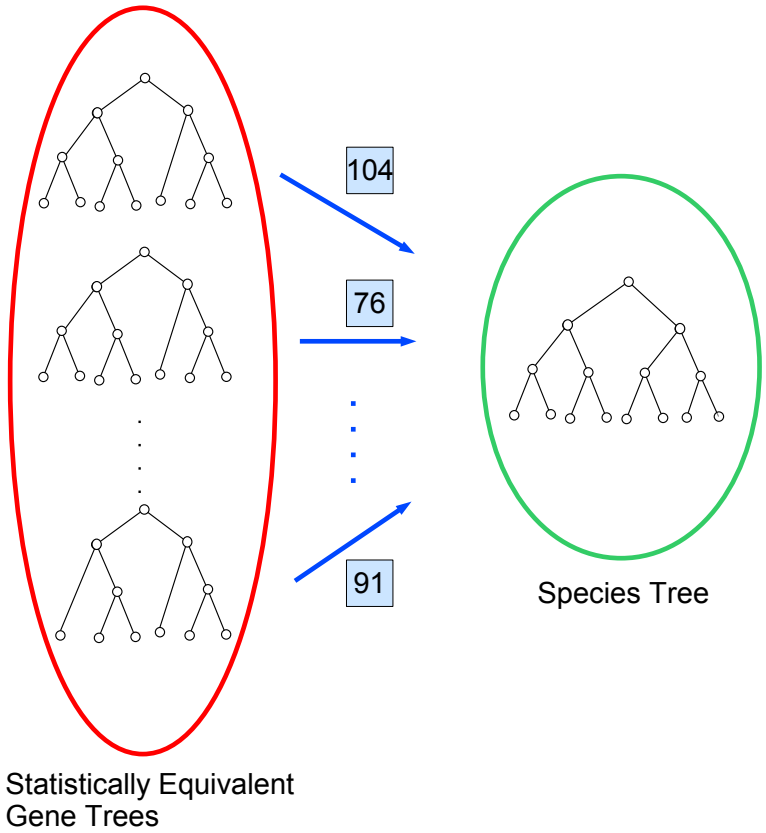
- TreeFix-DTL: A highly accurate method for prokaryotic gene tree reconstruction.

TreeFix-DTL: Algorithm Overview

Input: ML gene tree, multiple sequence alignment, and rooted species tree.

Output: Reconstructed (error-corrected) gene tree.

- Statistically informed, fast and scalable, easy to use.



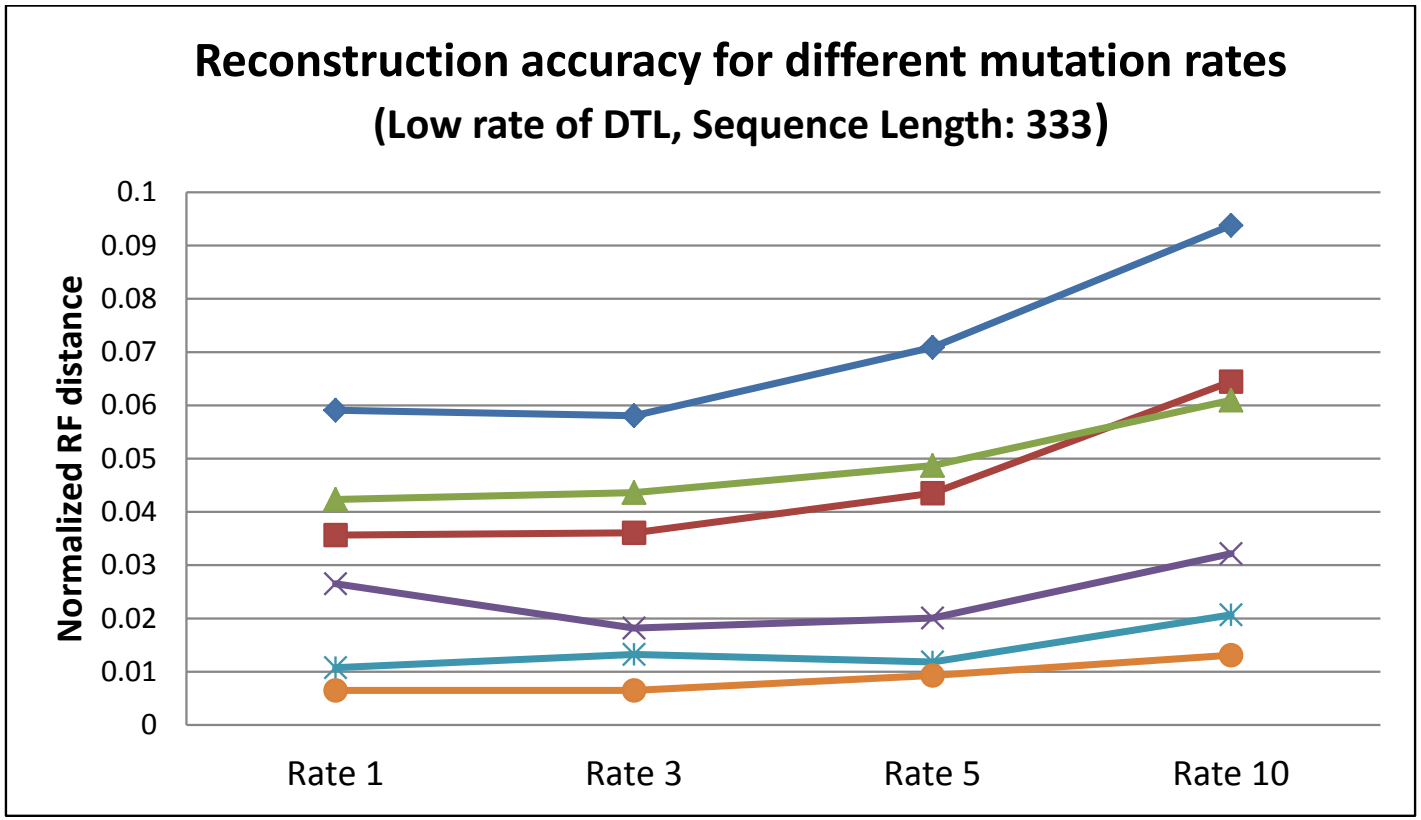
Dataset Generation

Basic simulation setup:

- 50 taxa Yule species trees.
- Gene trees with low, medium, and high rates of D, T, L.
- Mutation rates (substitutions per site) 1, 3, 5, and 10.
- Simulated amino acid sequences of length 173 and 333.
- Reconstructed using RAxML.

Total of 24 different datasets, each with 100 gene trees.

Performance: TreeFix-DTL is Highly Accurate



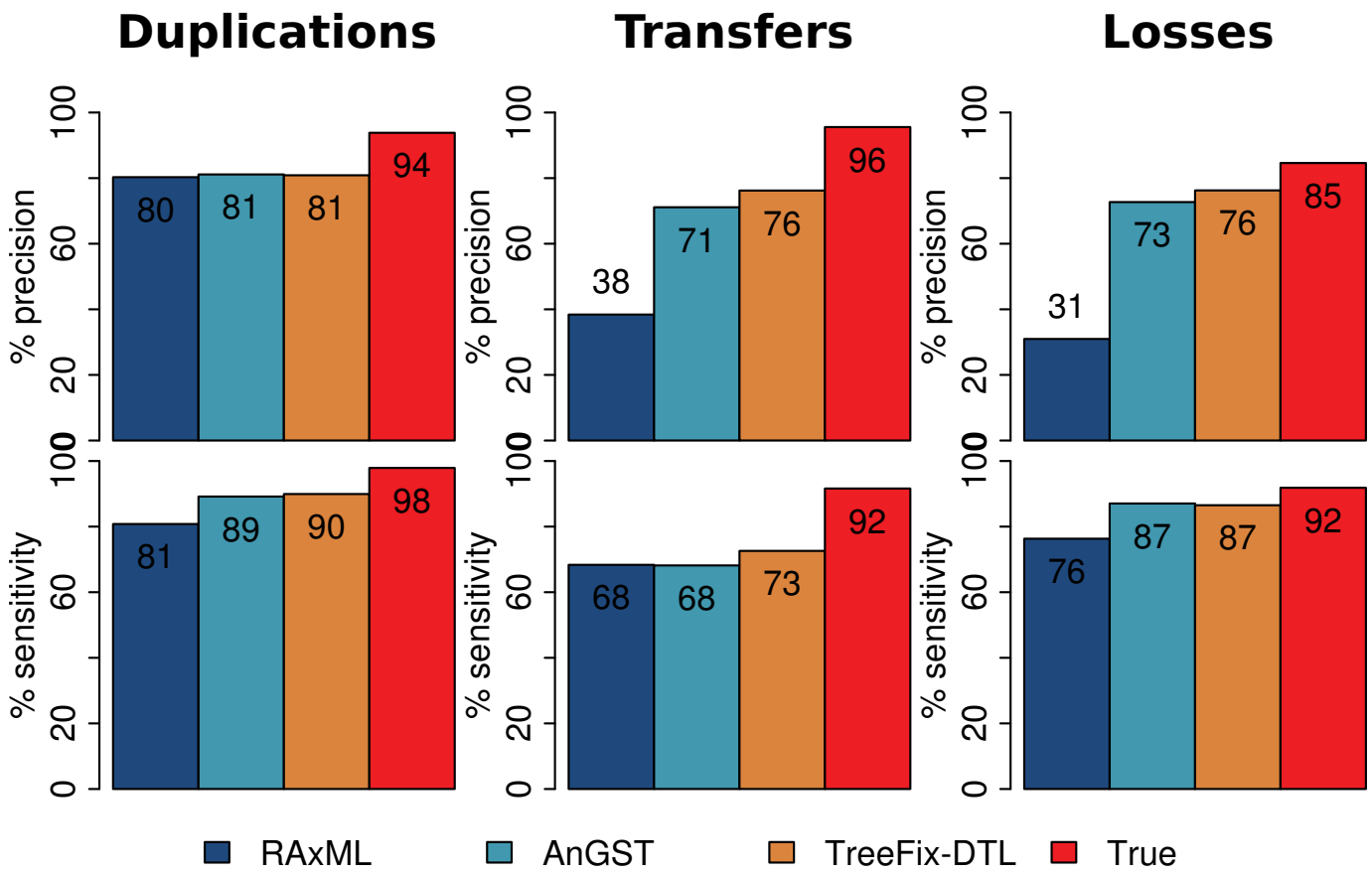
Performance: Tree Reconstruction Accuracy

Method	Normalized RF distance	Perfect reconstruction
RAxML	0.097	3.04%
NOTUNG	0.088	13.08%
TreeFix	0.079	10.29%
MowgliNNI	0.039	22.17%
AnGST	0.032	29.08%
TreeFix-DTL	0.028	38.21%

1. RAxML and eukaryotic methods error prone and ineffective.
2. TreeFix-DTL is highly accurate.

Performance: Greatly Improved Reconciliation Accuracy

Accuracy of inferred duplications and transfers: Averaged across all simulated datasets.



Impact of TreeFix-DTL

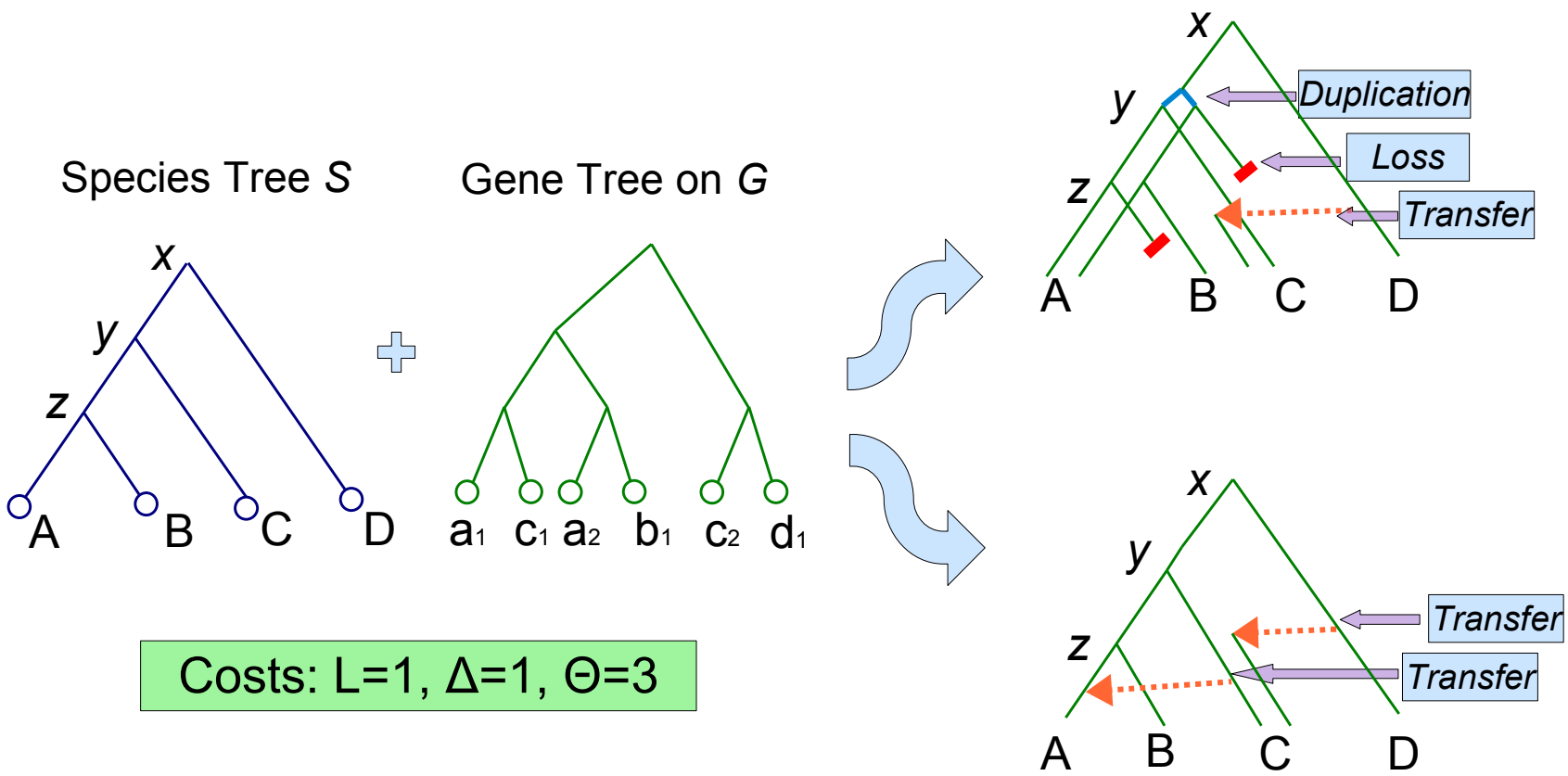
- Now possible to reconstruct gene trees very accurately.
- Large impact on accuracy of reconciliation and on any downstream analyses.

→ M. S. Bansal, Y. Wu, E. J. Alm, and M. Kellis, "Improved Gene Tree Error-Correction for Deciphering Microbial Evolution". Under review.

→ Y. Wu, M. D. Rasmussen, M. S. Bansal, M. Kellis. TreeFix: statistically informed gene tree error correction using species trees. *Systematic Biology 62(1): 110-120, 2013.*

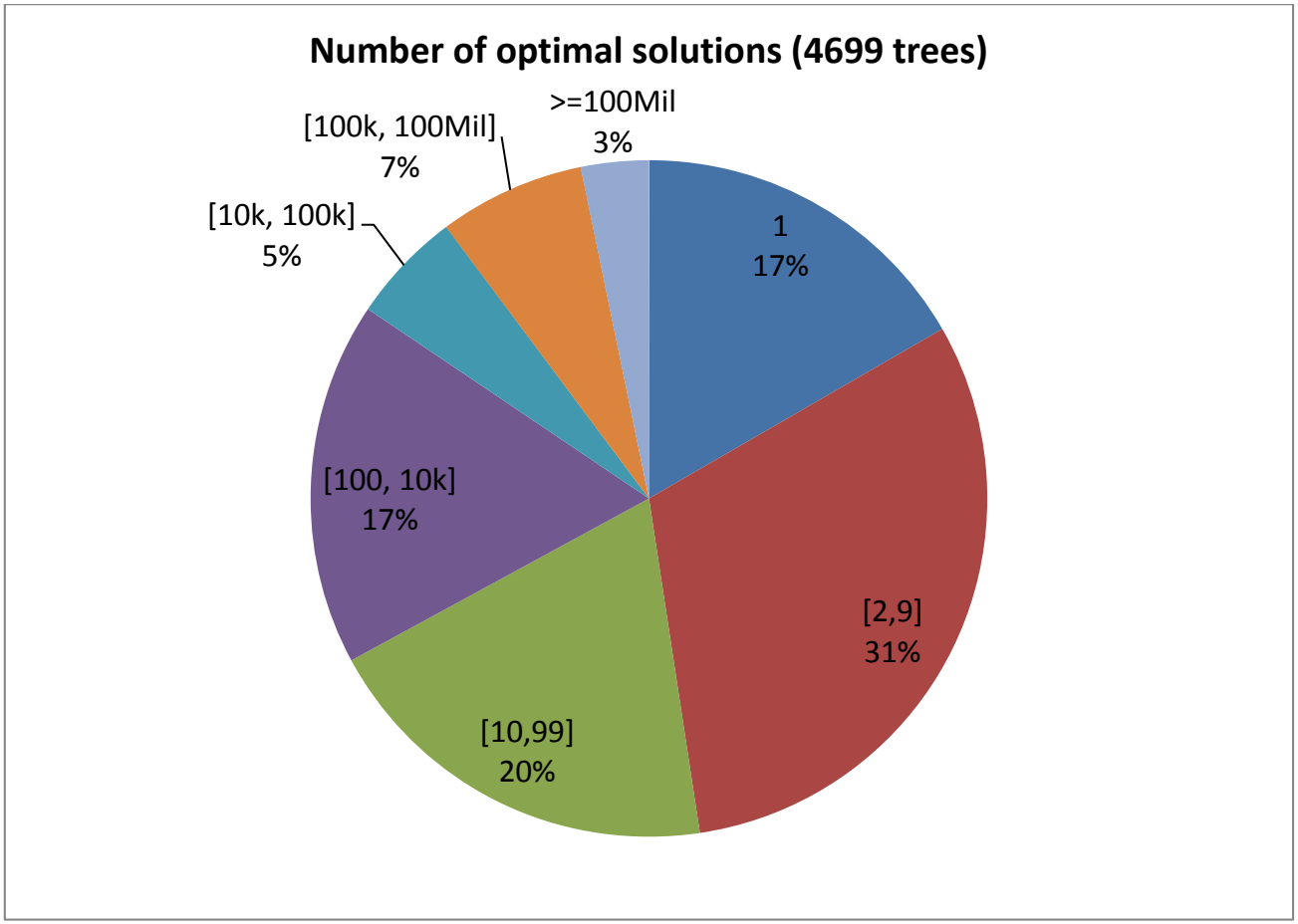
(c) Handling Multiple Optima

Optimal Reconciliations Need Not Be Unique

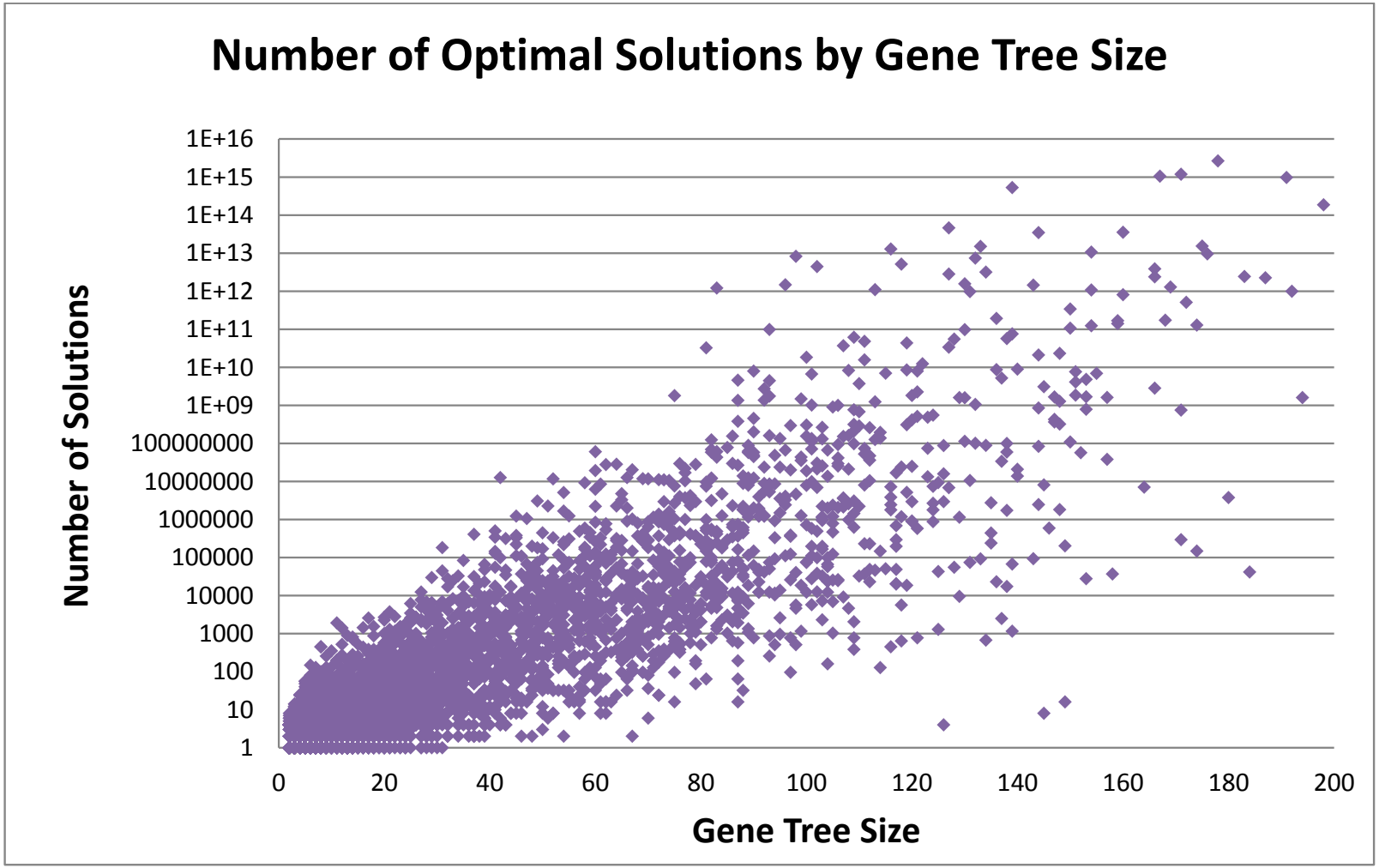


- Which one is the "correct" reconciliation?

Number of Optimal Solutions



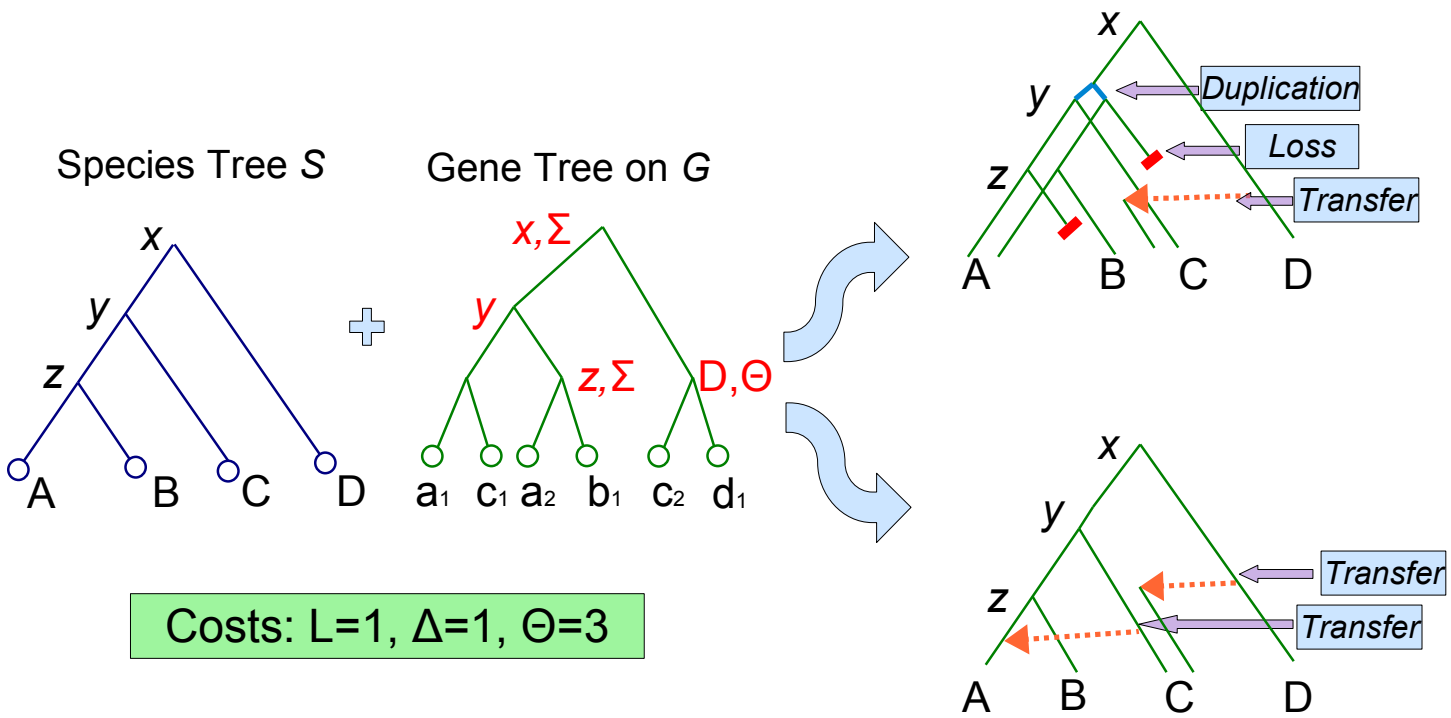
Exponential Increase with Gene Tree Size



Dealing With Non-Uniqueness

Fundamental question:

- How different are the different optimal reconciliations?



In this case, most node mappings and event assignments are consistent!

Goals

Goals:

- Understand the space of optimal reconciliations.
- Specifically, understand how different the mapping and event assignments can be in the different optima.

Solution: Sample the space of optimal reconciliations uniformly at random.

- Enables systematic study of space of optima for any input instance.
- Enables quick estimates of "support" for any event or mapping assignment.

Sample Optimal Reconciliations Uniformly

- We show that this is possible to do using a modification of the DP algorithm.
- Idea is to keep track of number of optimal solutions for each subproblem, and then weigh different equally optimal choices during backtracking step.
- Complexity increases from $O(mn)$ to $O(mn^2)$.

Application to Biological Dataset

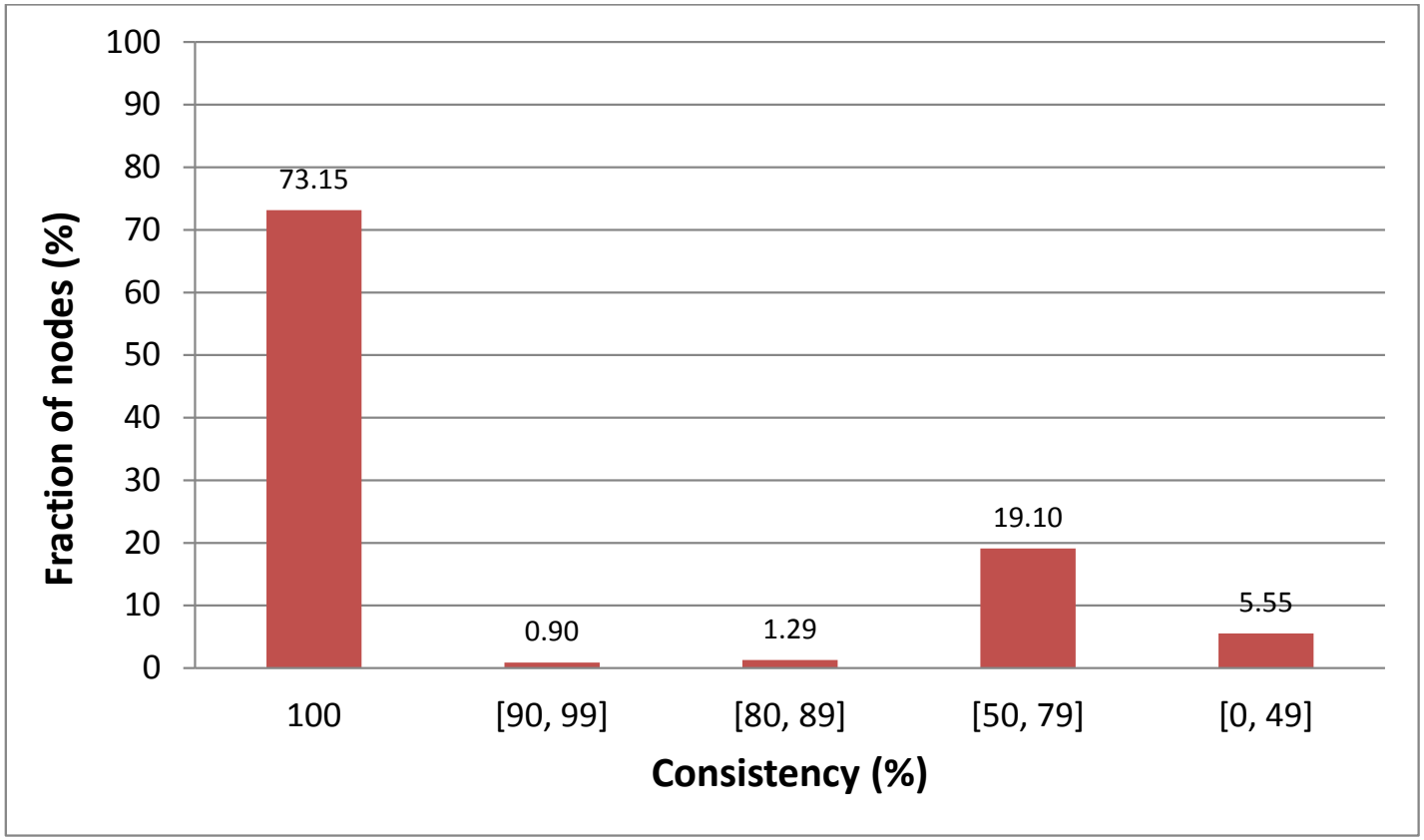
Dataset:
>4700 gene trees from a set of 100 species broadly sampled across the tree of life.

- Ran 500 times on each gene tree and aggregated the 500 reconciliations.

Aggregation: For each internal node in the gene tree:

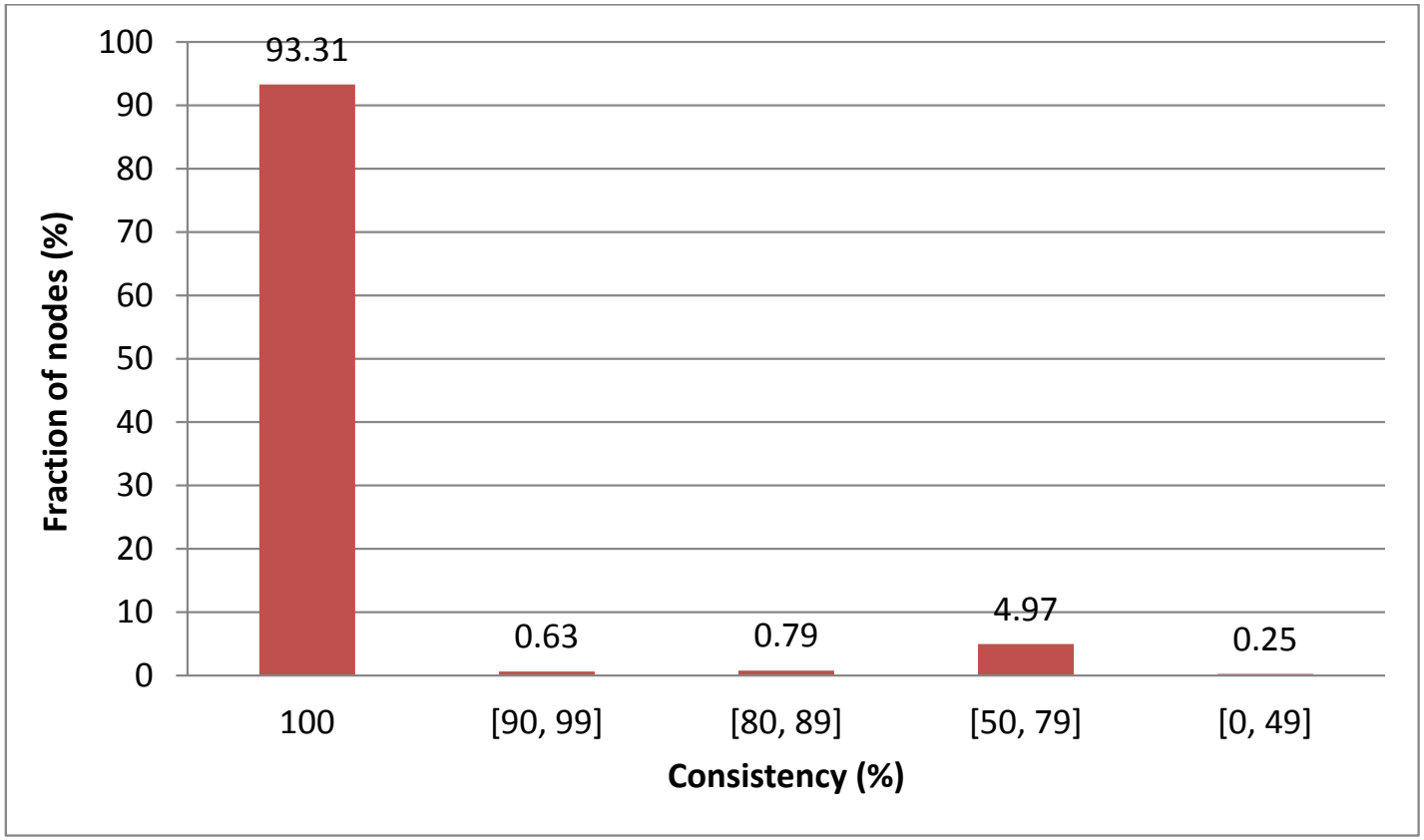
- How consistent are the 500 mapping assignments? (Frequency of most frequent mapping).
- How consistent are the 500 event assignments? (Frequency of most frequent event assignment).

Mapping Assignment Consistency



- For many nodes, the mapping assignments are 100% consistent.
- But significant fraction of nodes have inconsistent mapping assignments.

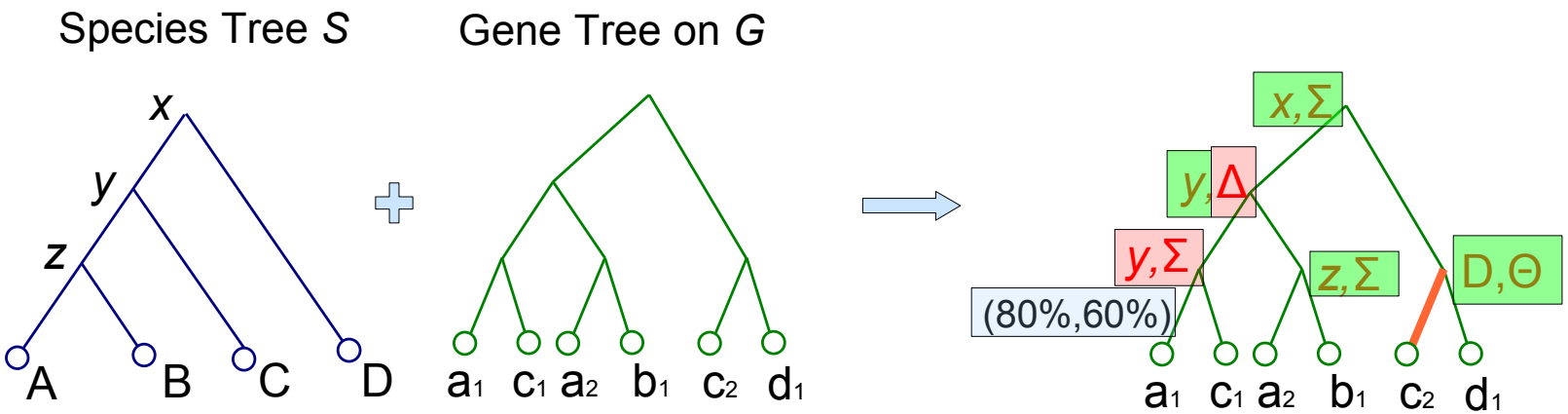
Event Assignment Consistency



- Good news: Event assignments are highly consistent!

Impact of Uniform Sampling on Handling Multiple Optima

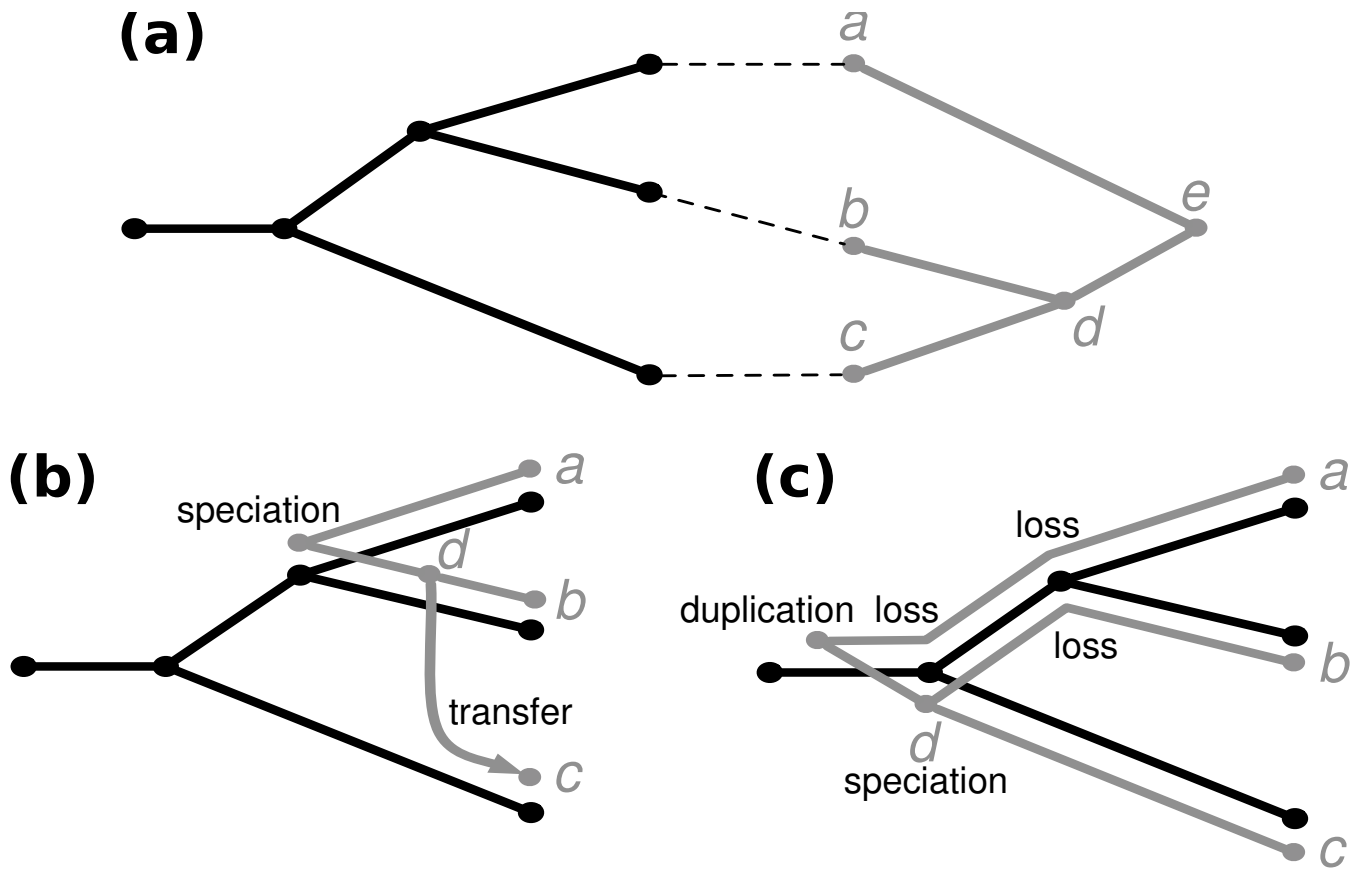
- Can efficiently study the space of optimal reconciliations.
- Can be used to determine consistency of the mapping and event assignment for any gene tree node.
- Event assignments are highly consistent: Great for functional genomic studies.



→ M. S. Bansal, E. J. Alm, and M. Kellis, "Reconciliation Revisited: Handling Multiple Optima when Reconciling with Duplication, Transfer, and Loss". *Journal of Computational Biology (JCB), 20(10): 738-754, 2013.*

(d) Impact of Event Costs

Event Costs Define Optimal Reconciliations



- Which one is the "correct" reconciliation?

Uncertainty in Event Cost Assignments

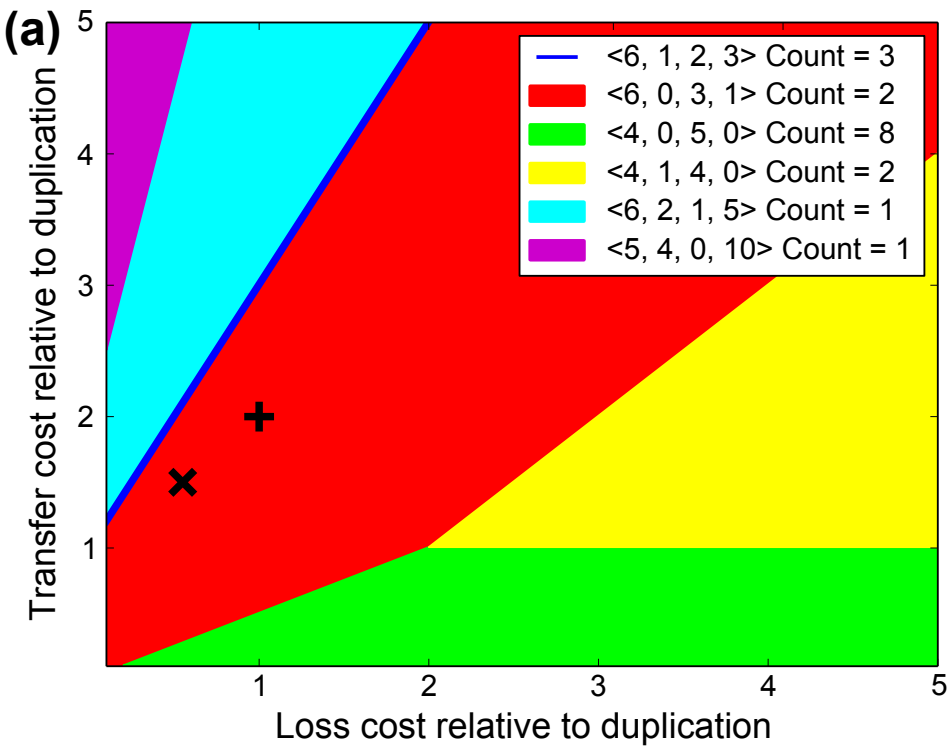
Fundamental question:

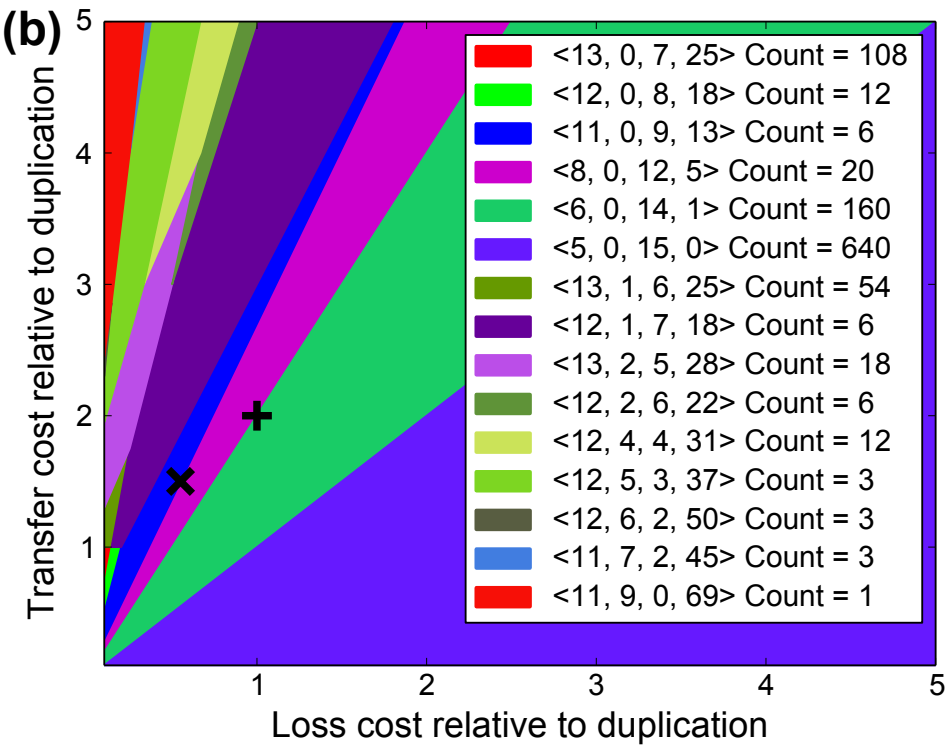
- What are the "correct" event costs?
  - Difficult to estimate.
  - "Correct" event costs may not even exist.
- How do reconciliations vary as we change event costs?

Pareto-optimal Reconciliations

- Keep track of all Pareto-optimal Reconciliations.
  - No other reconciliation is better in all three event counts.
  - Represents reconciliations that could be optimal for some assignment of event costs.
- $O(m^5 n \log m)$ time.
- Use to partition event cost space into equivalence regions.

Equivalence Regions





Impact in Practice

- Can visualize equivalence regions.
- Can choose suitable event cost assignments.
- Can be used to determine "consensus" mappings and event assignments.

→ R. Libeskind-Hadas, Y. Wu, M. S. Bansal, and M. Kellis, "Pareto-Optimal Phylogenetic Tree Reconciliation". *ISMB 2014; Bioinformatics 30: i87-i95, 2014*.

In Summary

- There now exist effective algorithms and methods to handle the major bottlenecks:
  1. Slow algorithms $\rightarrow$ Asymptotically faster algorithms
  2. Inaccurate gene trees $\rightarrow$ TreeFix-DTL
  3. Multiple optima $\rightarrow$ Uniformly random sampling
  4. Impact of event costs $\rightarrow$ Pareto-optimality, equivalence regions
- Easy to use, Fast, Accurate, Scalable.

Constructing Phylogenetic Networks

Using existing software:

- Build accurate gene trees.
- Obtain a species tree (can be subset of network).
- Use DTL reconciliation to reconcile individual gene trees.
- Aggregate transfers inferred for gene trees onto species tree.
- Advantages: Scalable, can handle all gene families, not fooled by duplication/loss and ILS.

Future Work:

- Program to aggregate reticulations onto species tree to draw networks and infer highways. Use global view to refine individual reconciliations.
- Reconciliation against species networks.

Thank You!

Questions!