Algorithms for constructing hybridization networks from multiple gene trees

Yufeng Wu
Dept. of Computer Science and Engineering
University of Connecticut, USA

Workshop on Phylogenetic Networks
July 30, 2015

Hybridization Networks

Gene trees: phylogenetic trees from gene sequences

- Assume: **Binary** and **rooted**
- **Different** topologies at different genes

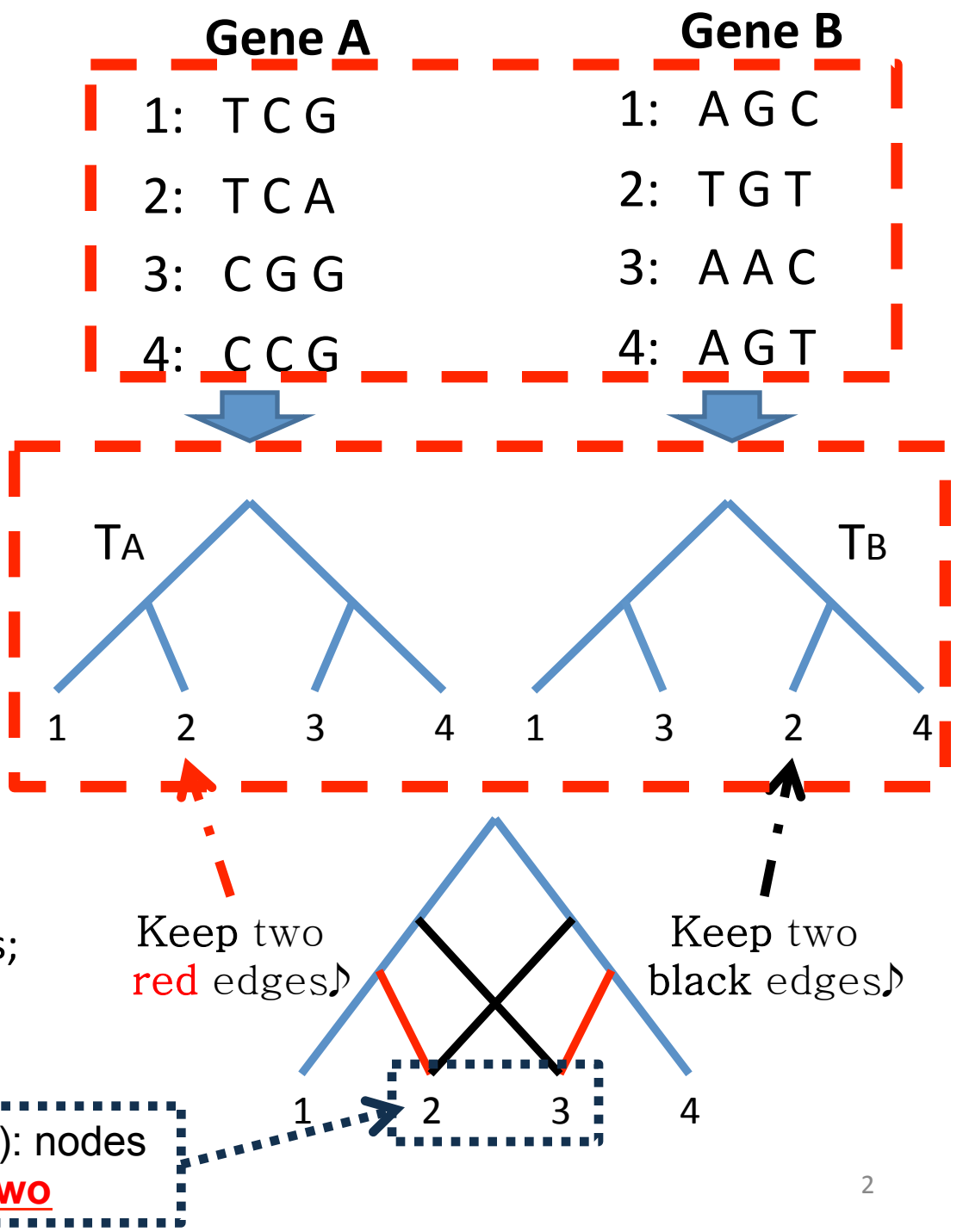
Reticulate evolution:

one explanation

- Hybrid speciation, horizontal gene transfer

Hybridization network:

A directed acyclic graph displaying **each** of the gene trees; each node with one or two incoming edges.



Hybridization Networks

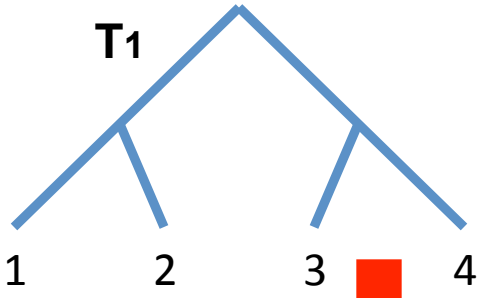
Given: a set of **K** binary gene trees **G**.

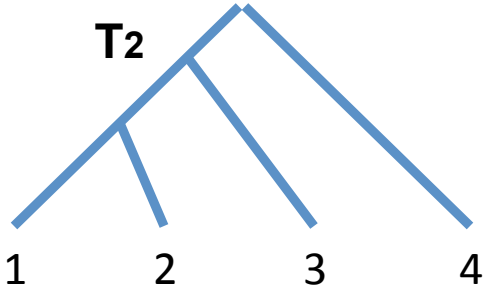
Problem: reconstruct hybridization networks with Rmin(**G**), the **minimum** number, reticulation events ***displaying*** each gene tree

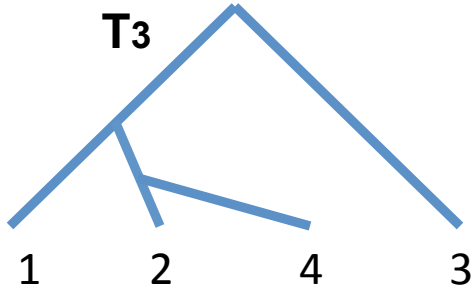
NP complete: even for **K=2**

Current approaches:

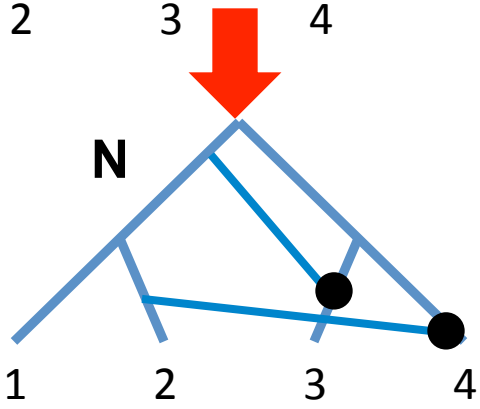
- exact methods for K=2 case (see Semple, Linz, et al) or K ≥2 (Wu, RECOMB 2013)
- impose topological constraints (e.g. galled networks, see Huson, Kelk, Van Iersel, et al.)







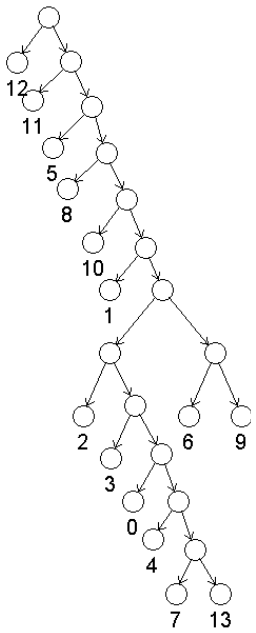
reticulation



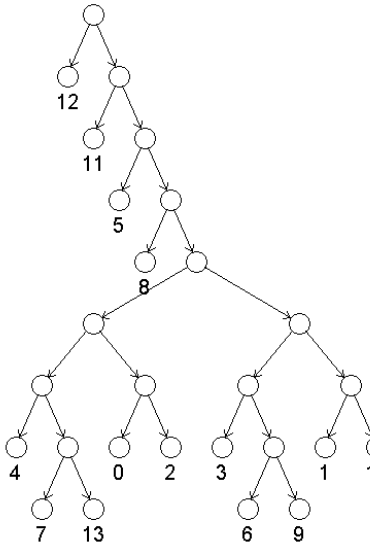
Two reticulation events**. Claim: minimum.**

Optimality: more difficult for **larger trees.**

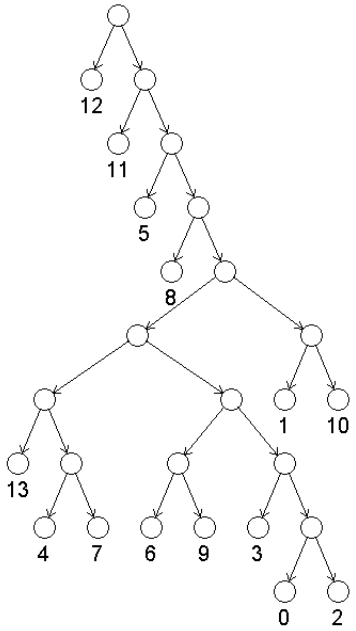
Five Poaceae Trees (13 taxa): how many reticulation events do we need?



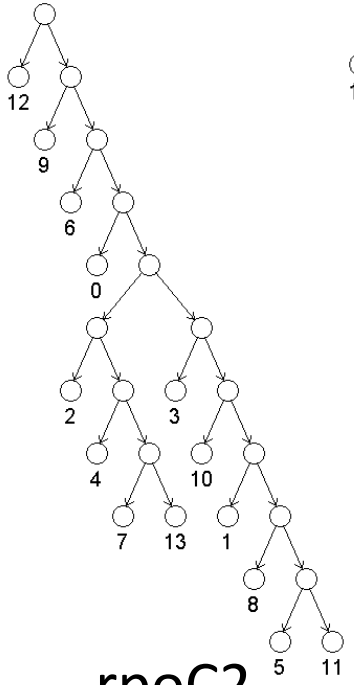
ndhF



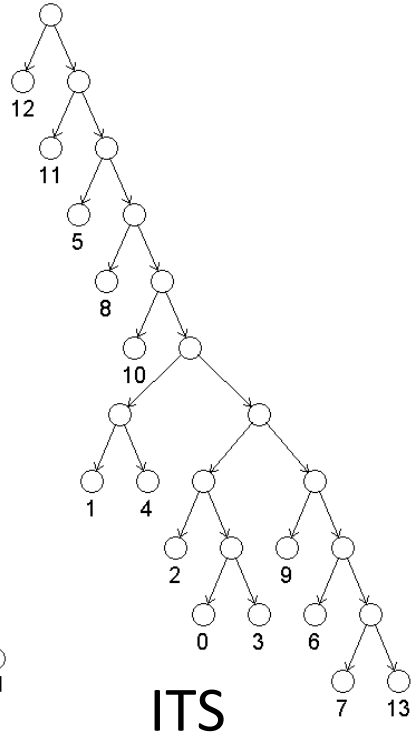
phyB



rbcL



rpoC2



ITS

Disclaimer: obtained from S. Linz

Key question: how to build good networks? Exact (but slow) method or **heuristics**.

Key question: how good is a given network? A **lower bound** for optimality.

Challenge: exact network is **infeasible** by existing methods.

Our idea: estimate **lower and upper bounds** on these trees instead

Close Lower and Upper Bounds for Minimum **Reticulation** of **Multiple** Gene Trees (Wu, ISMB 2010)

Key idea: developing novel lower and upper bounds for Rmin(**G**): G is the set of **K** gene trees.

$$RH(\mathbf{G}) \leq R_{min}(\mathbf{G}) \leq SIT(\mathbf{G})$$

RH(G): Lower bound
First non-trivial bound

Rmin(G): Minimum
Challenging for K ≥ 3

SIT(G): Upper Bound
Works for any K

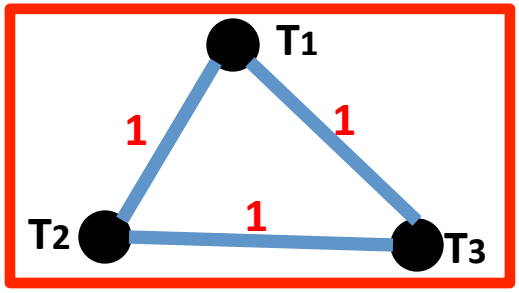
Bounds provides **range** of $R_{min}(G)$

If RH(G)=SIT(G), then **R_{min}(G) = RH(G) = SIT(G)**

RH Lower Bounds from **Pairwise** Distance

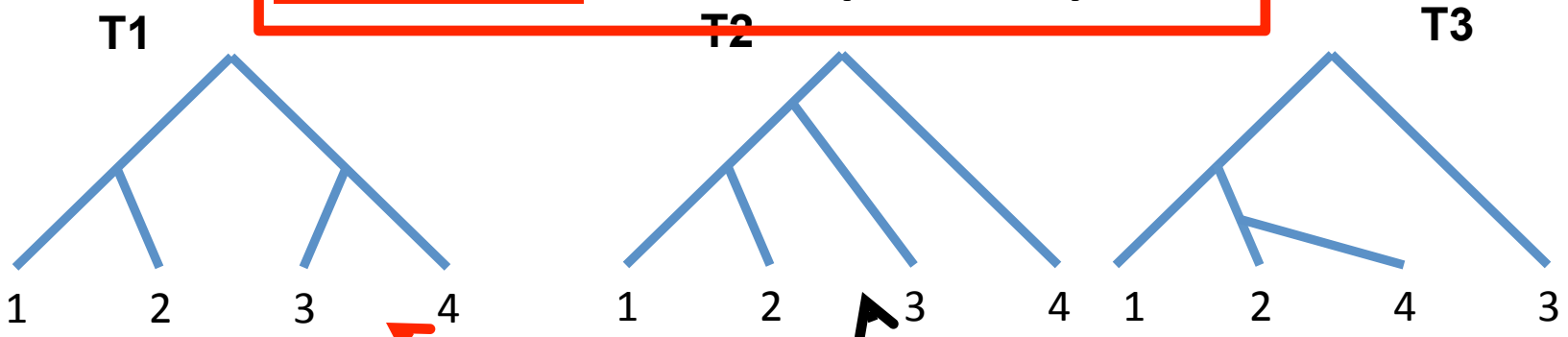
Pairwise reticulation distance of T_1 and T_2: **$d(T_1,T_2)$**, the **minimum** reticulation in any reticulate network for T_1 and T_2. Exists several methods for this.

Pairwise distances for **T_1, T_2 and T_3**

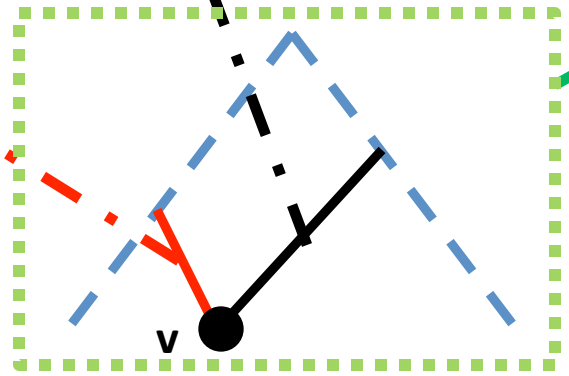


$R_{min}(T_1,T_2,T_3) \geq \max(1,1,1) = 1$

Question: can **$R_{min}(T_1,T_2,T_3) = 1$**?



Choosing same reticulate edge ➔ same gene trees

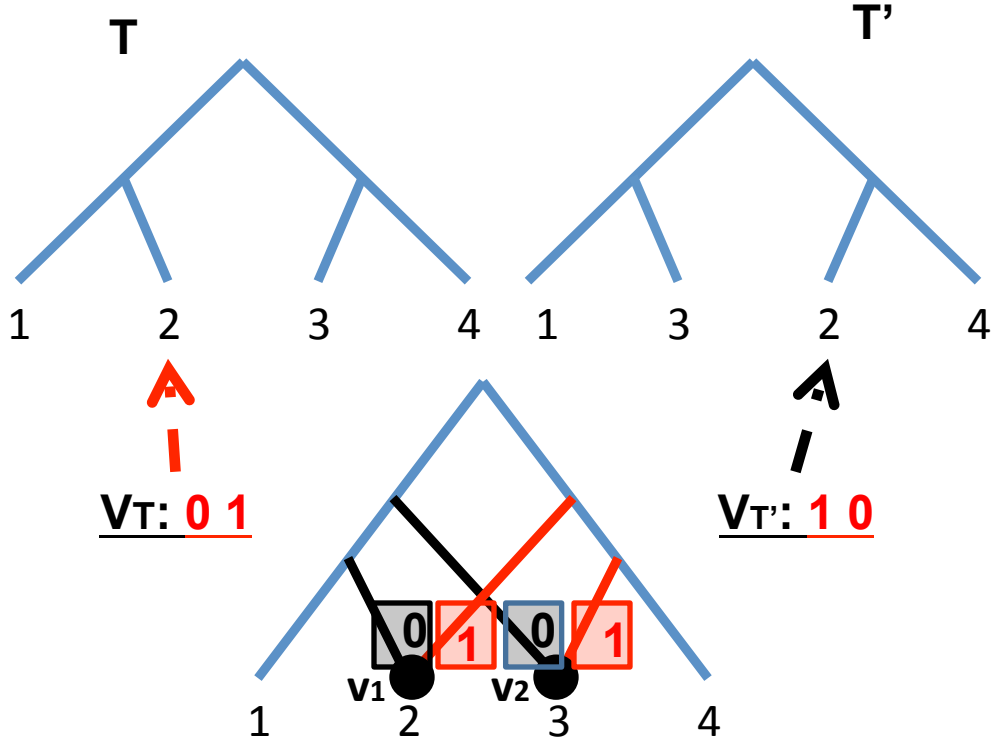


?

Imaginary network with one reticulation node

$R_{min}(T_1, T_2$ and $T_3) \geq 2$!

Display Vector



Tree T is **displayed** in a network

Each tree has a display vector

V_T : **Display vector** of T, how T is displayed in the network

- **one** bit per reticulation node ➔ length of display vector = number of reticulation nodes in the network
- value 0/1: at each reticulation node, which edge (the 0-edge or 1-edge) is kept for T?

Intuition: display vectors can *not* be too similar for two different trees

Lemma: $D(V_{T1},V_{T2}) \geq d(T_1,T_2)$ for *any* network displaying T_1 an T_2.

$D(V_{T1},V_{T2})$: Hamming distance of V_{T1} and V_{T2}.

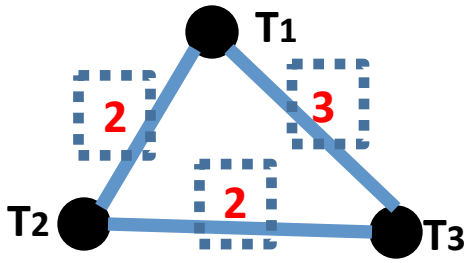
$d(T_1,T_2)$: pairwise reticulation distance of T_1 and T_2

The RH Lower Bound

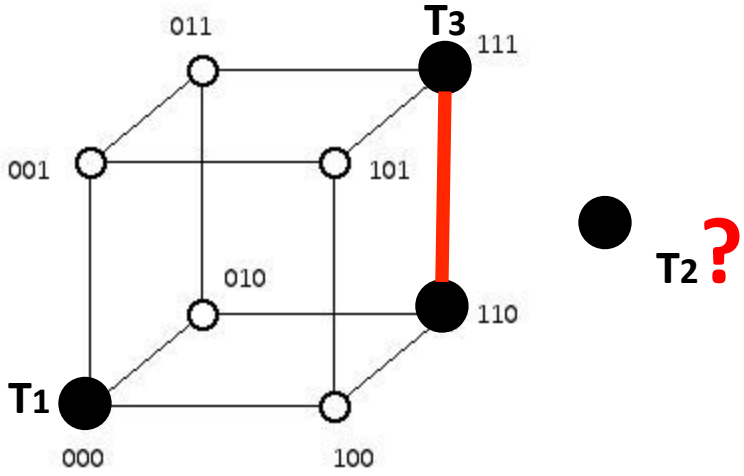
Key: if **R** reticulation events possible, then exist K length **R** display vectors, satisfying the **distance constraints**: Hamming distance D(VT,VT') ≥ d(T,T')

- **Analogy**: Selecting K points on R dimensional binary **hypercube** s.t. the points can *not* be too close
- If such K points do not exist, then we must need at least R+1 reticulation events.

RH lower bound: the **smallest** R s.t. K points can be selected on R-dimensional hypercube satisfying the distance constraints.



Question: can $R_{min}(T_1,T_2,T_3) = 3$?



$R_{min}(T_1, T_2, T_3) \geq 4$!

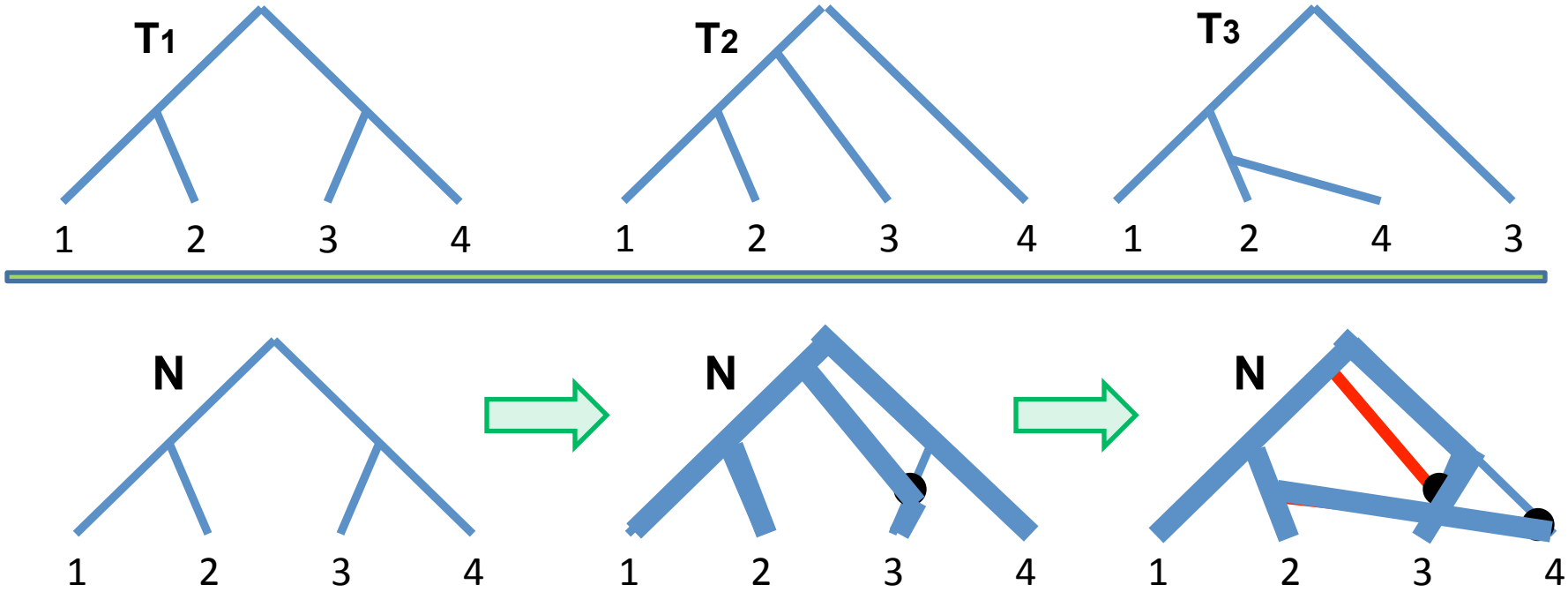
We use **integer linear programming** to solve it.
Closed-form formula of RH bound for K=3.

SIT Bound Heuristic for Constructing Networks (implemented in program PIRN)

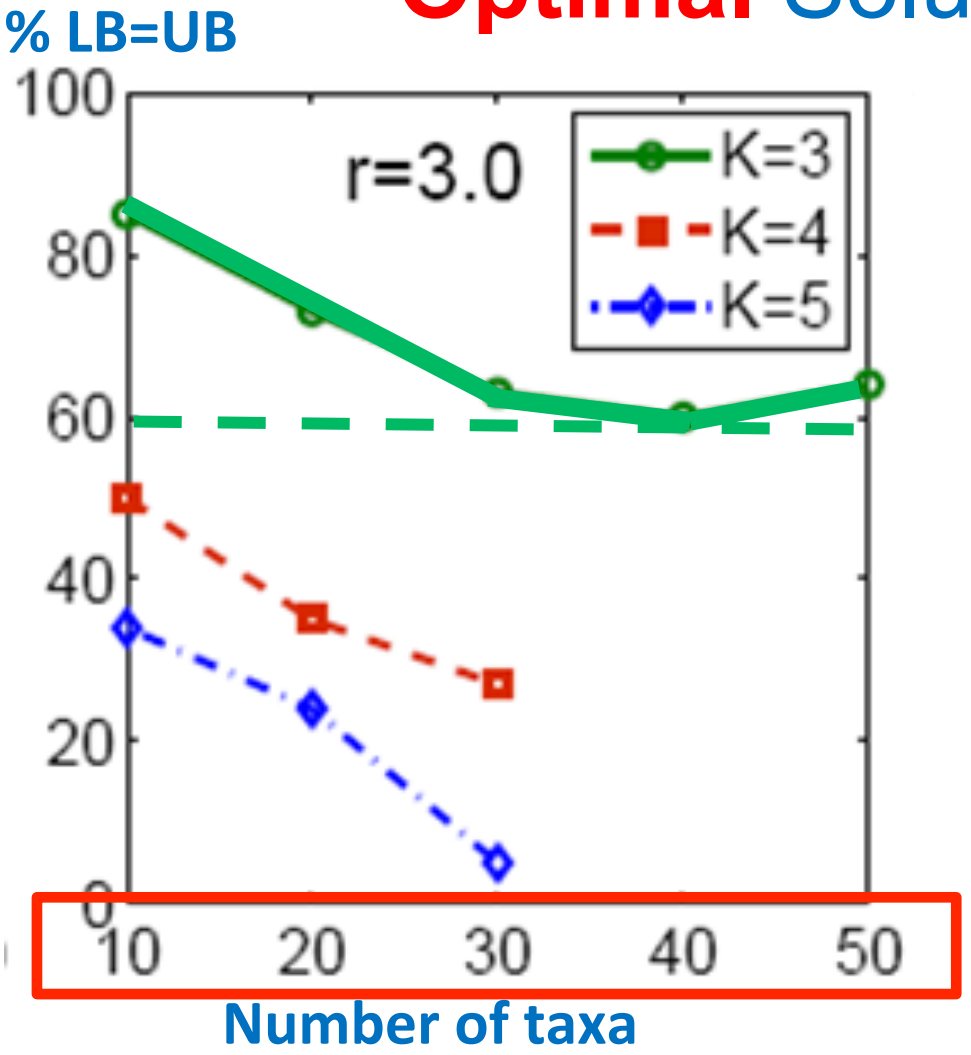
Heuristic: how to reconstruct a network for $T_1, T_2, \ldots, T_K$ using small (may *not* be minimum) number, U, of reticulation events?

Key idea (1) try all ordering of T_i to find the **best**; (2) for each order of $T_1, T_2, \ldots$, sequentially **insert** gene trees T_i into a **growing** network **N.**

- Each step inserts a tree into N by **new** reticulation to *display* T_i in N.
- **Minimize** the new reticulation events at each step (i.e. making smallest changes to accommodate the new tree Ti).



Performance of PIRN: % of Datasets **Optimal** Solution Found



Horizontal axis: *number* of taxa

Vertical axis: % of datasets lower = upper bounds

K: *number* of gene trees

r: *level* of reticulation

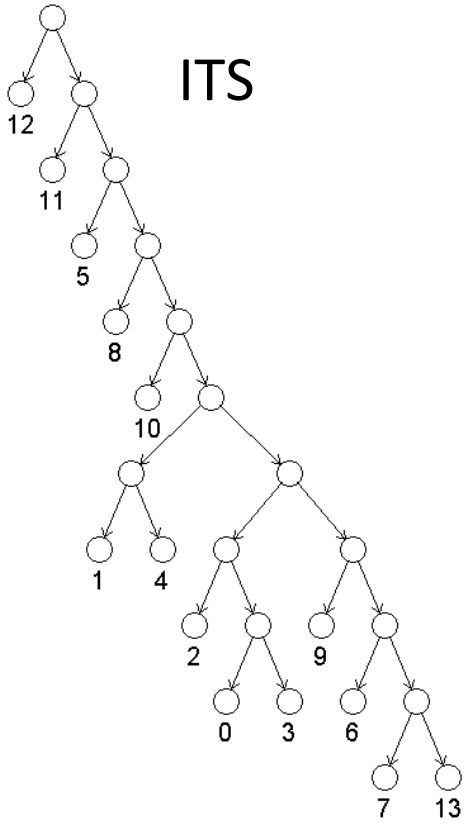
Average over 100 datasets

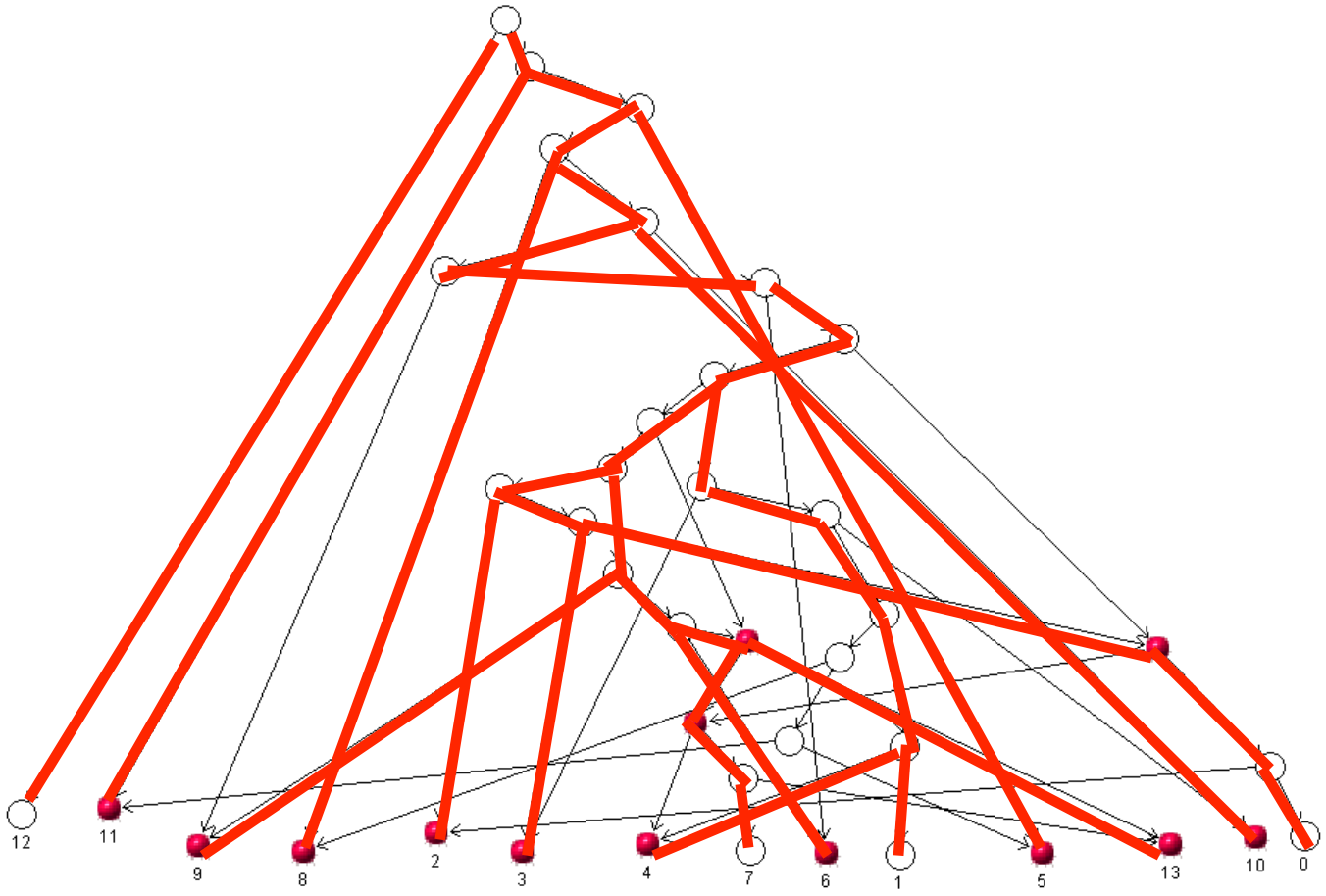
Lower and upper bounds often **match** for many data

When the number of trees **K** increases, performance *declines*.

PIRN: cannot handle large number of gene trees

Hybridization Network for Five Poaceae Trees





RH bound for five Poaceae trees: **11**

SIT bound: **13** reticulation events used in the constructed network.

Question: is this network optimal?

Lower bound: no stronger bound is known

Upper bound: a new method by Mirzaei and Wu

PIRNc: Exact algorithm for hybridization networks with unbounded K (Wu, RECOMB 2013)

PIRNc: allow arbitrary num of gene trees and taxa; no assumption on network topology, etc.

Constraint: Practical for networks with **4** (*maybe* 5) hybridization events.

	n=10			n=20			n=30		
	K=3	K=4	K=5	K=3	K=4	K=5	K=3	K=4	K=5
#Data ≤ 4	98	98	93	88	77	65	84	76	65
$PIRN_C$ = RH	96	93	90	88	74	63	84	75	61
$PIRN_C$ > RH	2	5	3	0	3	2	0	1	4
$PIRN_C$ < SIT	0	1	0	0	1	0	0	1	0
$PIRN_C$ = SIT	98	97	93	88	76	65	84	75	65
$PIRN_C$ > SIT	0	0	0	0	0	0	0	0	0
#Data not optimal by SIT	2	*6*	3	0	4	2	0	1	4
Time	13.4	49.9	92.6	276.8	705.8	1686.6	606.7	2227.1	2811.5

n: number of taxa. K: number of trees

Time: seconds

#Data <= 4: # of data w/ 4 or less reticulations among 100 simulations

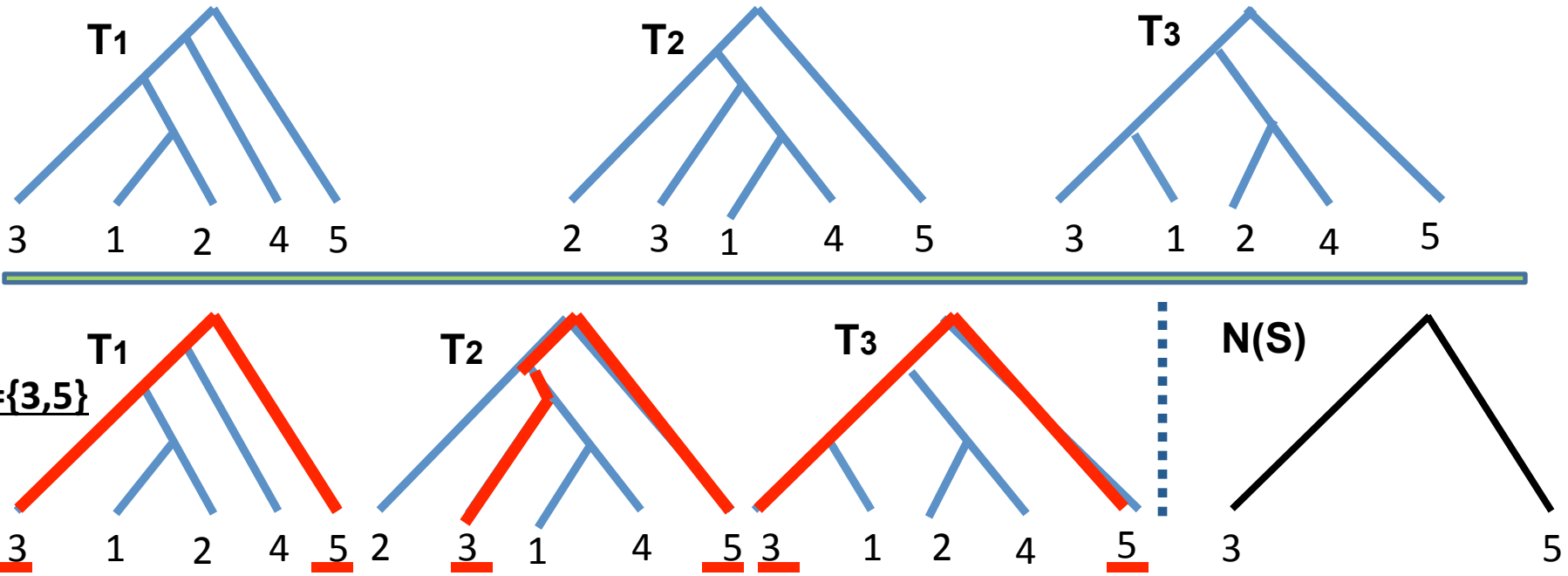
PIRNs: a new heuristic for large number (K) of trees (Mirzaei and Wu, TCBB, 2015 to appear)

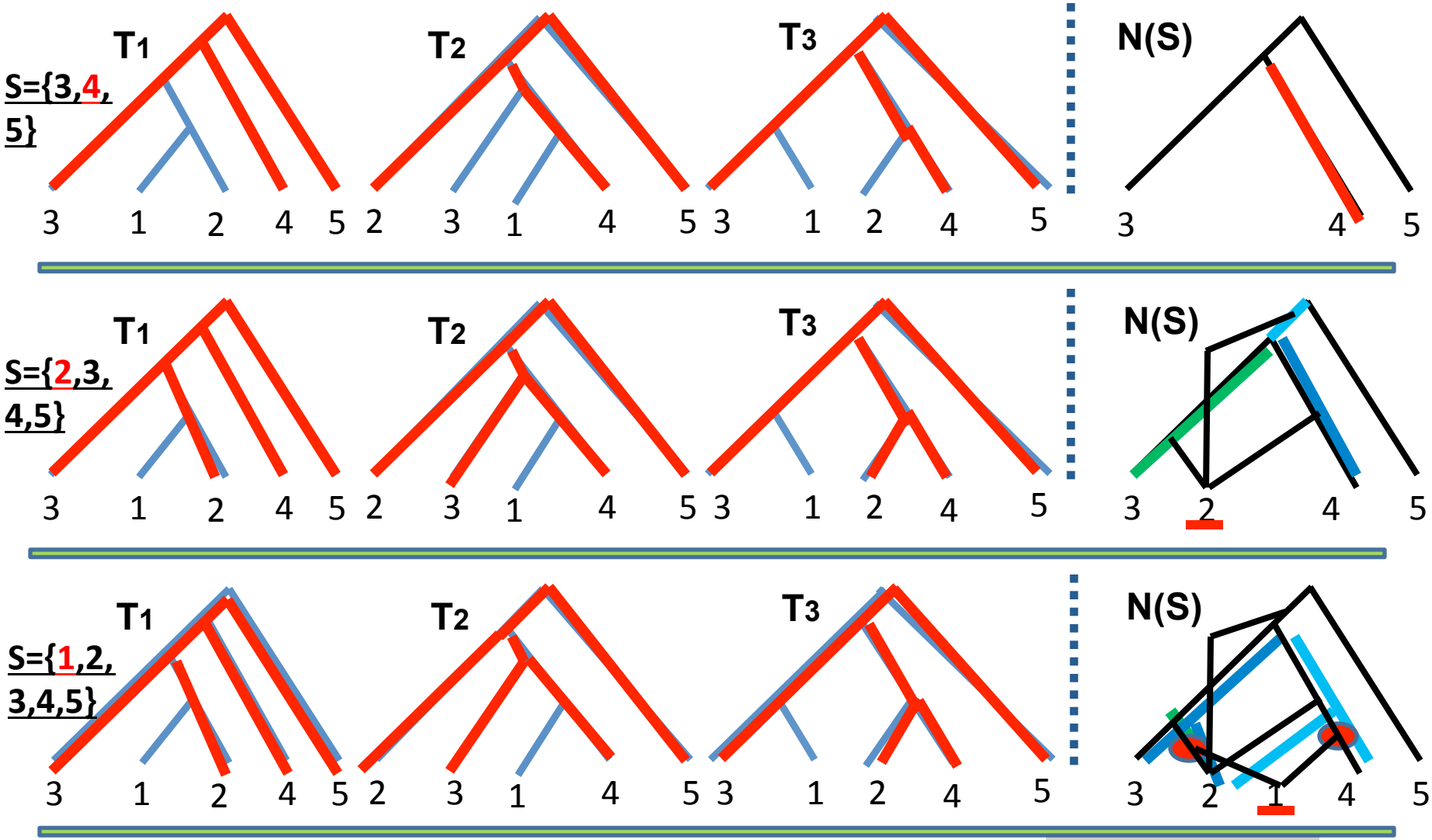
Key idea: consider **subsets** of taxa ***S***

Networks for *S*: display *subtrees* of Ti with taxa only from ***S***

Grow network for larger S by modifying networks for small subsets by adding as few reticulation events as possible

More efficient when **K** (num of trees) is large and num of taxa is not too large





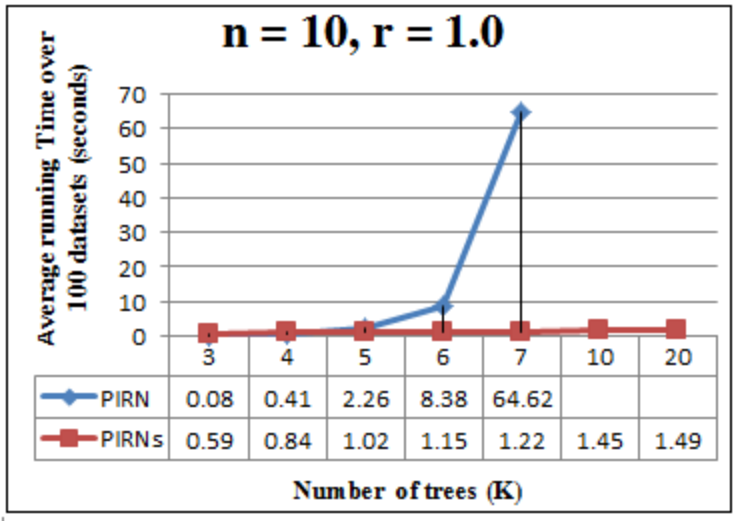
Share reticulation: choose the smallest number of edges to attach reticulation edges (a hitting set problem)

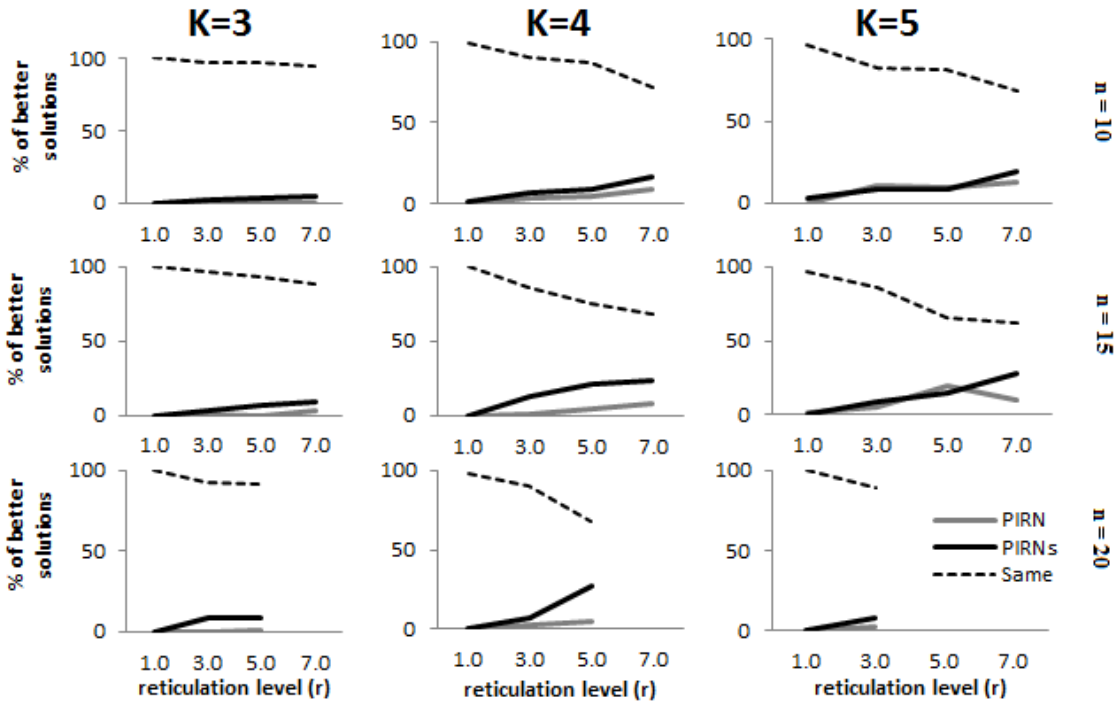
Share a common edge with T1

- For each tree Ti, find the set of edges to attach edges from the new taxa to display T
- This is based on how Ti is displayed in the network

Performance of PIRNs

Running time: PIRNs is much faster for larger K values.

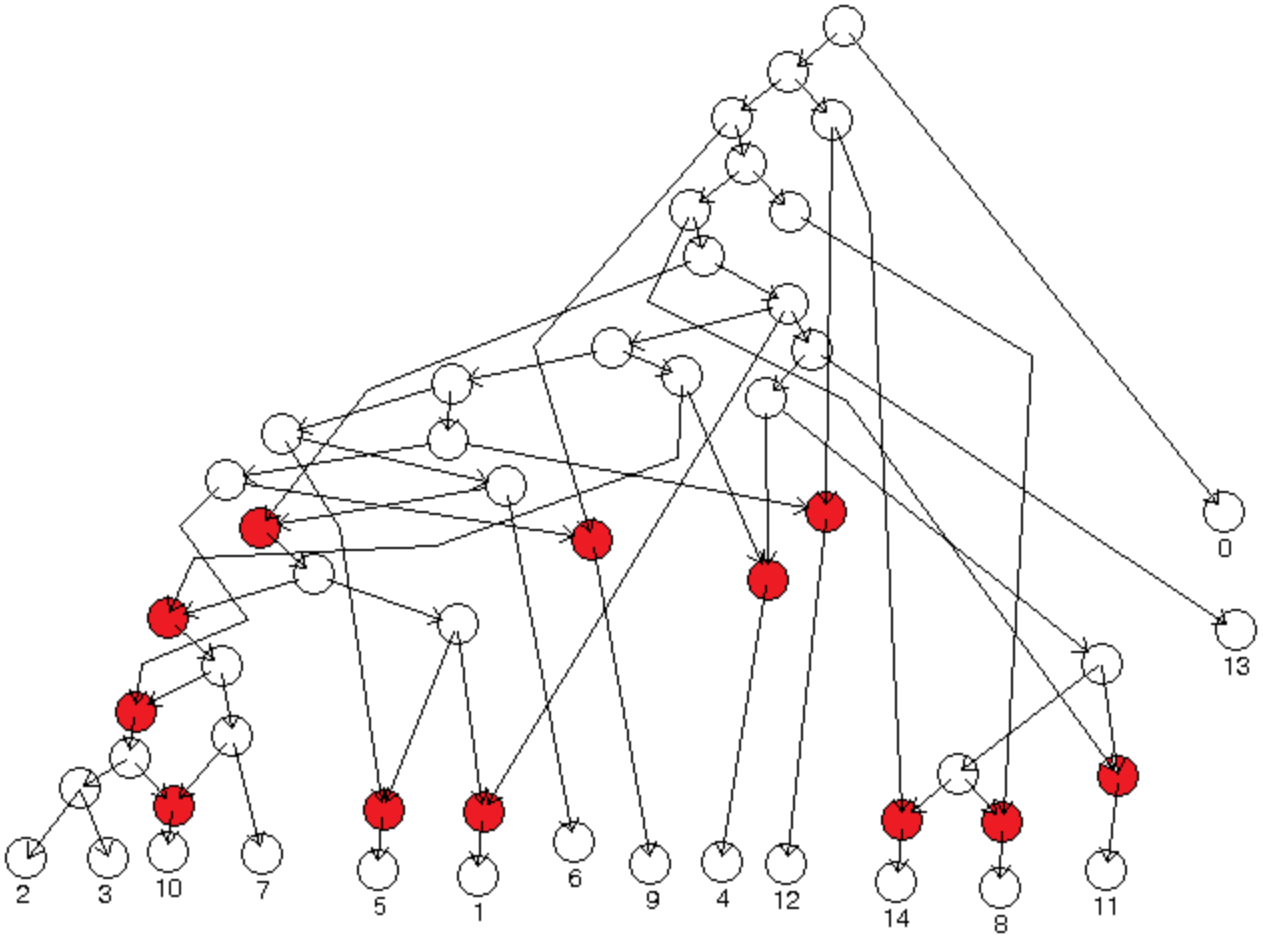




Parsimony: PIRNs produces more parsimonious networks when K increases than PIRN (the SIT bound).

K: number of trees. Dashed lines: PIRN=PIRNs. **Dark solid line**: PIRNs better. Light solid line: PIRN better.

A Better Network for Five Poaceae Trees



PIRNs: 12 reticulation events used in the constructed network (lower bound is 11). Optimality: still not known

- Papers from me and my student
  - Yufeng Wu: **Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees**. Bioinformatics [ISMB] (2010)
  - Yufeng Wu: An Algorithm for Constructing Parsimonious Hybridization Networks with Multiple Phylogenetic Trees. Journal of Computational Biology (RECOMB, 2013)
  - Sajad Mirzaei and Yufeng Wu: **Fast Construction of Near Parsimonious Hybridization Networks for Multiple Phylogenetic Trees**, IEEE/ACM TCBB, accepted, 2015
- Related work from other groups: Park and Nakhleh (2012), Chen and Wang (2012), Albrecht (2015)
- More information available at: **http:// www.engr.uconn.edu/~ywu**
- Research supported by National Science Foundation (CCF-1116175 and IIS-0953563)