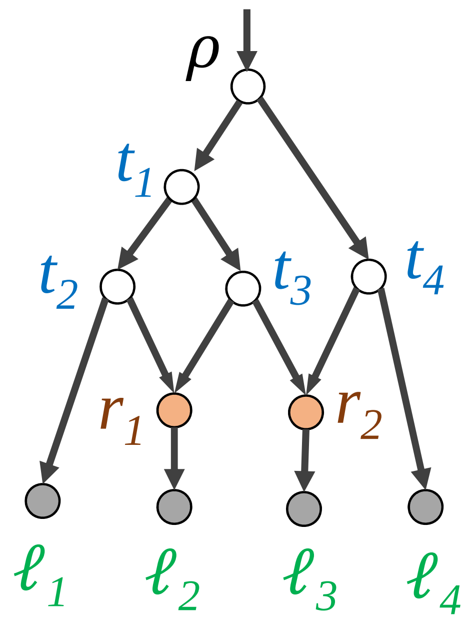
Stable Networks

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

Phylogenetic Networks

- A (**binary**) **phylogenetic network** is a rooted directed acyclic graph
  - -- The root: ρ
  - -- Leaves: $\ell_1, \ell_2, \ell_3, \ell_4$
  - -- Tree nodes: t_1, t_2, t_3, t_4
  - -- Reticulation nodes: r_1, r_2
- A **binary tree** is a binary phylogenetic network without reticulations.



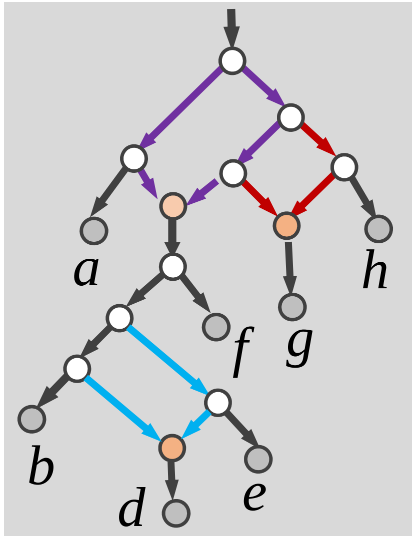
Galled Trees, Galled Networks & Stable Networks

- A network is
  a **galled tree** if
  the smallest cycles
  (ignoring direction)
  containing different
  reticulation nodes are
  node-disjoint.

In a galled tree, reticulations occur in non-overlapping regions.

Wang, Zhang, Zhang, JCB, 2001
Gusfield, Eddu, Langley, Proc. of CSB, 2003

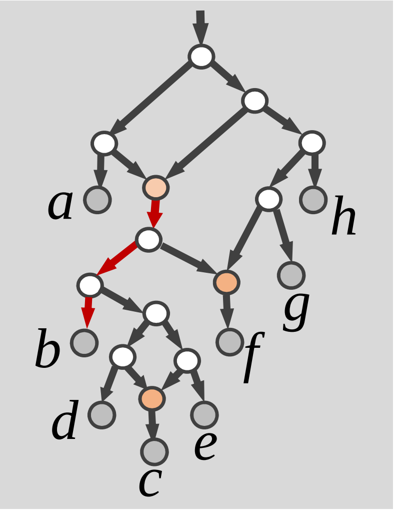
- A network is a galled network if there exists a cycle containing only tree nodes and *r* for each ret. node *r*.



In a galled network, reticulation nodes in the same region have no ancestor-descendent relation.

Huson, Klopper, RECOMB, 2007

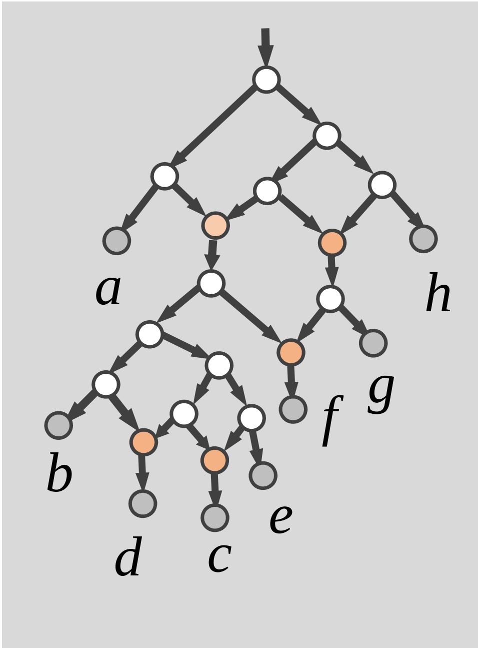
- A network is a tree-child network, iff each node has a tree node as its child, and iff there exists a tree-node path from each node to a leaf.

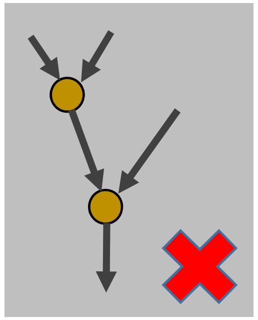


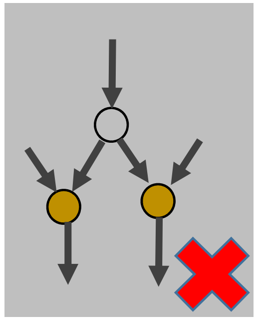
Gardona, Rossello, Galiente, TCBB, 2007

- A reticulation node r is a stable stable (or visible) if there is a leaf ℓ such that every path $P(\rho, \ell)$ must go through r.

- A network is stable (or reticulation-visible) if each reticulation node is stable.

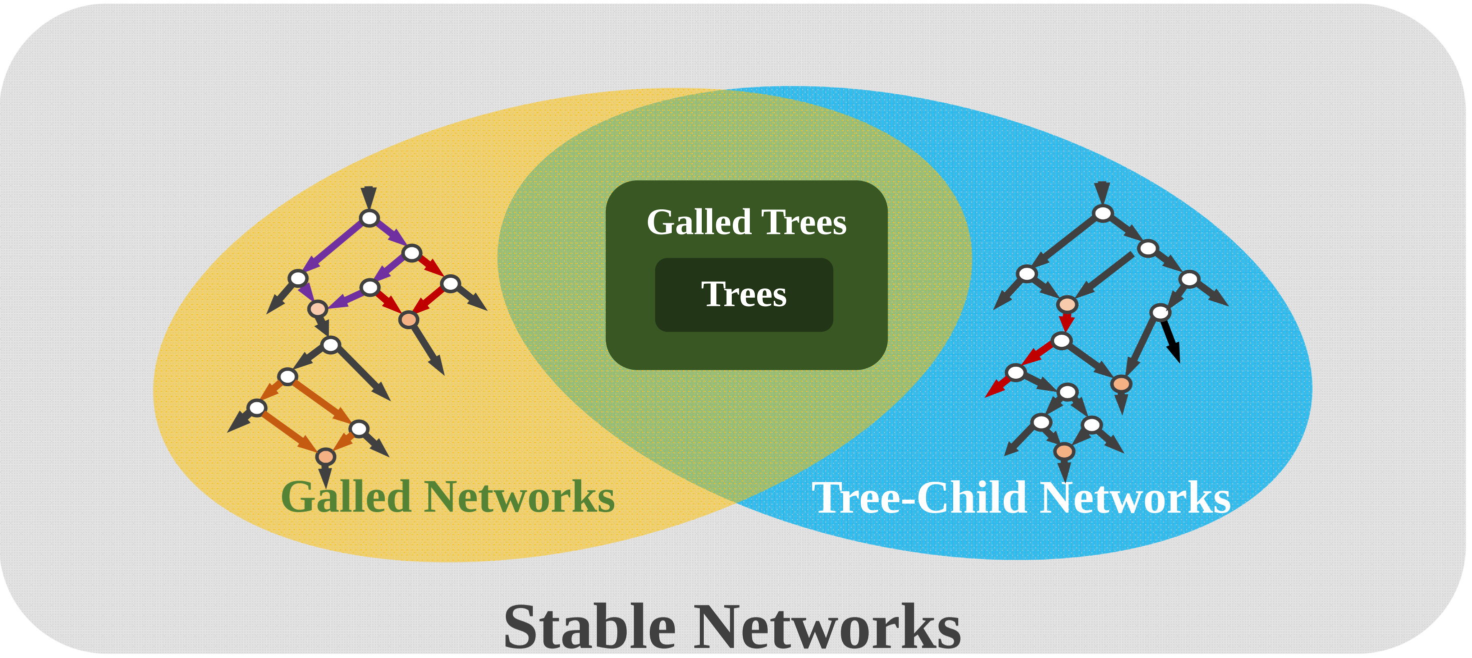






In a stable network, each ret. node is surrounded by three tree nodes.

Huson, Rupp, Scornavacca, Phylogenetic Networks, 2010


Galled Trees
Trees
Galled Networks
Tree-Child Networks
Stable Networks

Today's Talk

- **The sizes of binary stable networks**

  Theorem 1 There are at most 4(n-1) reticulation nodes and at most 5(n-1) tree nodes in a binary stable network with n leaves.

- **The tree containment problem (TCP)**

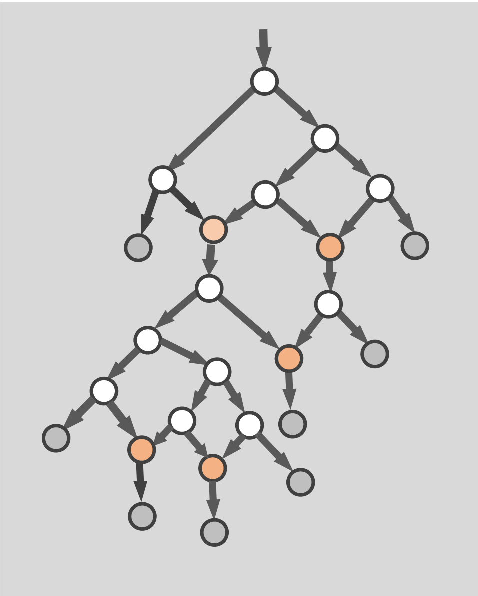
  Theorem 2 The TCP is solvable in cubic-time for (binary) stable networks.





Gambette, Gunawan, Labarre, Vialette, Zhang, RECOMB'15
Gunawan, DasGupta, Zhang, http://arxiv.org/abs/1507.02119

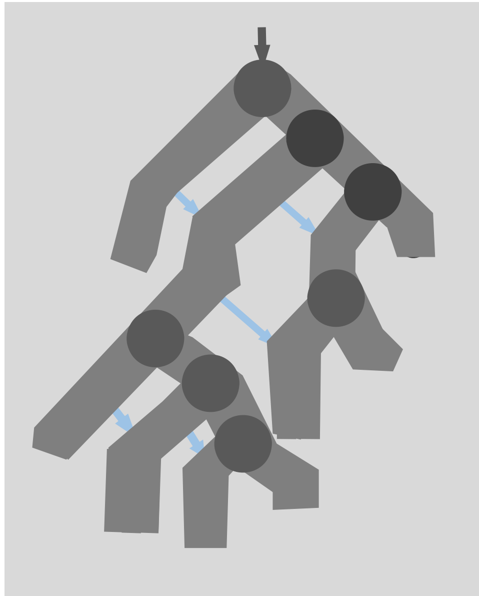
Part 1 #(Ret. Nodes) $\leq 4(n-1)$



N : a binary stable with n leaves

Consider a subtree

obtained by removing an incoming edge for each ret. node



T : a subtree of N with *the same* n leaves.

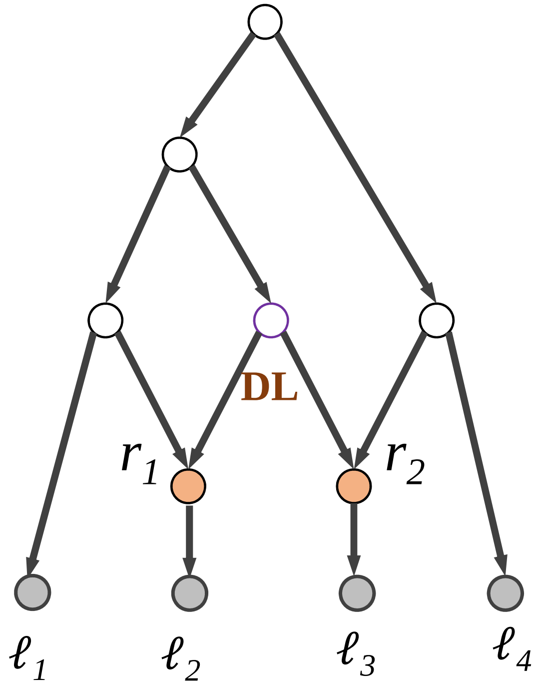
#(deg-3 nodes) = n-1;

#(paths) = 2(n-1).

Lemma At most two edges were removed from each of 2(n-1) paths.

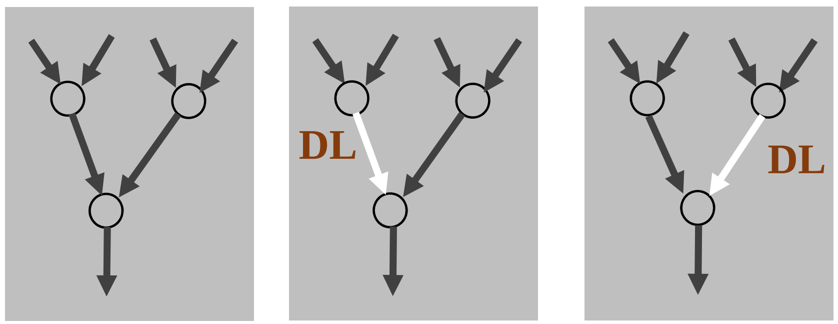
Proof of Theorem 1:
#(Ret. nodes)
= #(Removed edges)
= 2 ×2(n-1).

Dummy Leaf (DL)



Question Does there exist an edge set E containing an incoming edge for each reticulation node such that N - E is a subtree without dummy leaf for a binary network?

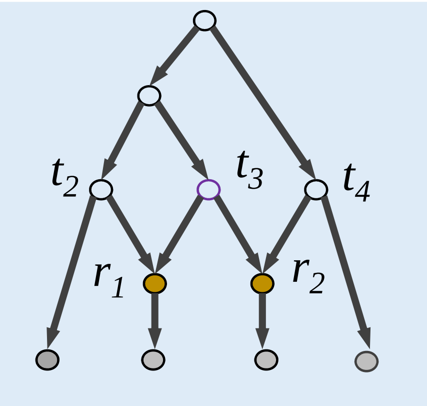
No!



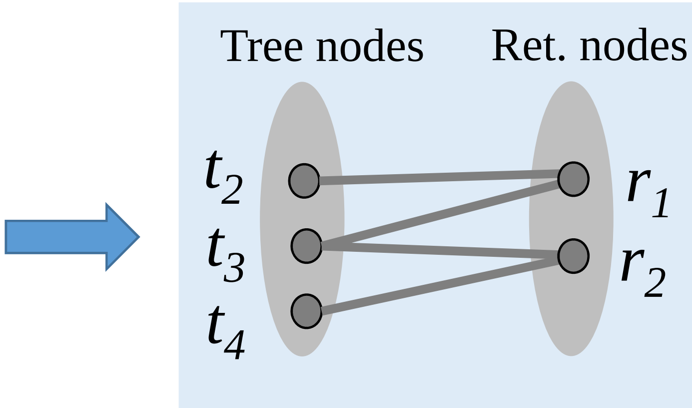
Question Does there exists an edge set E such that N - E is a subtree without dummy leaf ?

Yes for a binary stable network

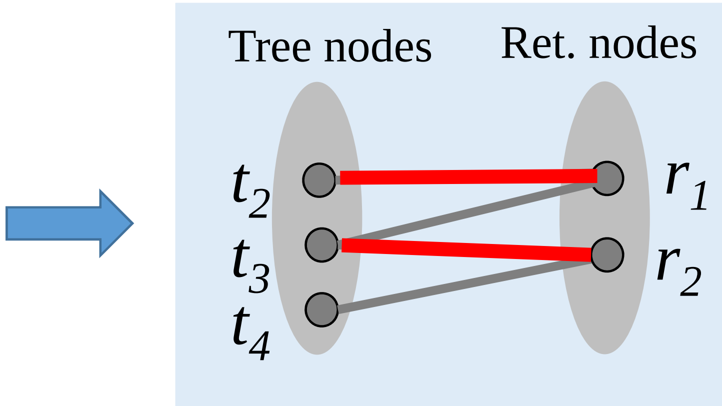
Fact N - E is a subtree without dummy leaf iff |E|=#(Ret. nodes) and E is a matching consisting of reticulation edges in N.



The parents are both tree nodes for each reticulation node.



Degree is at most 2 for t's degree is exactly 2 for r's. (Alon's condition)



Hall Theorem on matching in bipartite graphs

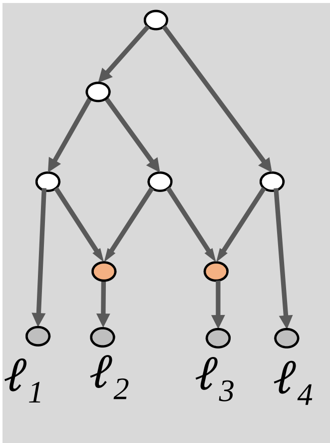
Size Bounds for Binary Networks

Network Type	#(Reticulation Nodes)	#(Int. Tree Nodes)	References
Galled Tree	$\leq n-1$	$\leq 2(n-1)$	well-known
Tree-Child Network	$\leq n-1$	$\leq 2(n-1)$	Well-known
Galled Network	$\leq 2(n-1)$	$\leq 3(n-1)$	Ours
Stable Network	$\leq 4(n-1)$	$\leq 5(n-1)$	Ours

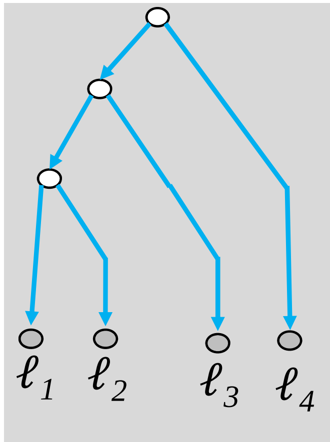
Part 2 A Cubic-Time Alg. for the Tree Containment Problem

Tree Containment Problem (TCP)

Input: A network $N=(V, E)$ and a binary tree T with the same leaves.
Question: Does N display T ?



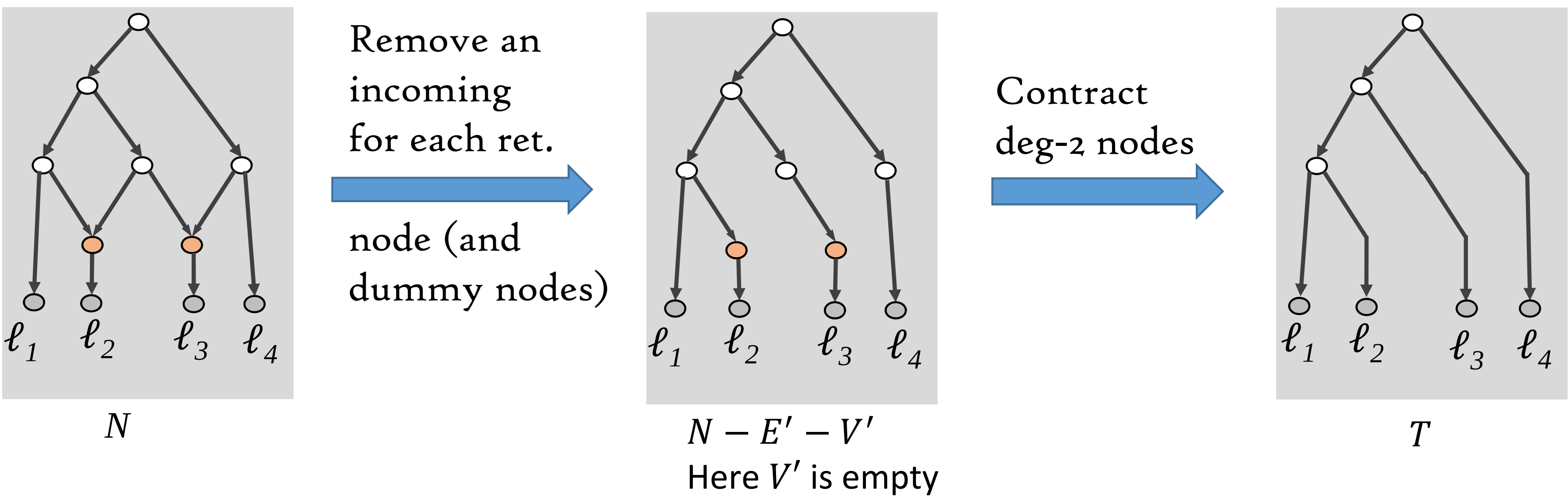
N



T

Tree Containment Problem (TCP)

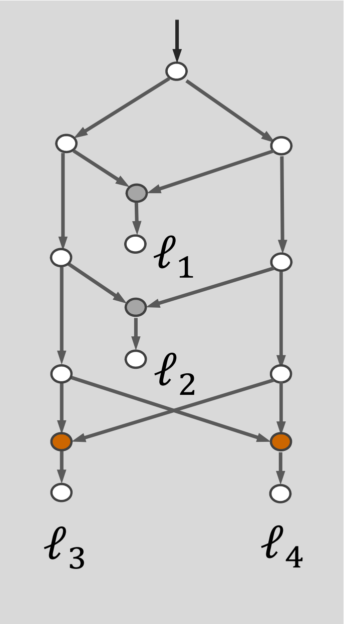
Input: A network N=(V, E) and a binary tree T with the same leaves.
Question: Find $E' \subseteq E$ and $V' \subseteq V$ such that $N - E' - V'$ is a subdivision of T.



- The TCP arises from network model verification
- It is related to the subgraph homomorphism problem
- It is NP-complete. The naïve algorithm takes $O(2^{\#(\text{ret. nodes})}n)$
- Polynomial-time algorithms are known only for
  - -- Tree-child networks
  - -- Genetically-stable networks
  - -- Nearly-stable networks

Kanj, Nakhleh, Than, Xia, TCS, 2008
Van Iersel, Semple, Steel, IPL, 2010
Gambette, Gunawan, Labarre, Vialette, Zhang, RECOMB, 2015a
Gambette, Gunawan, Labarre, Vialette, Zhang, Manuscript, 2015b

- Difficulty
  - -- Examining all the reticulation nodes simultaneously takes o(2^n) time
  - -- Examining reticulation nodes one-by-one does not lead to a correct algorithm
  - -- How to identify the set of reticulation nodes that can be dissolved simultaneously in poly-time?

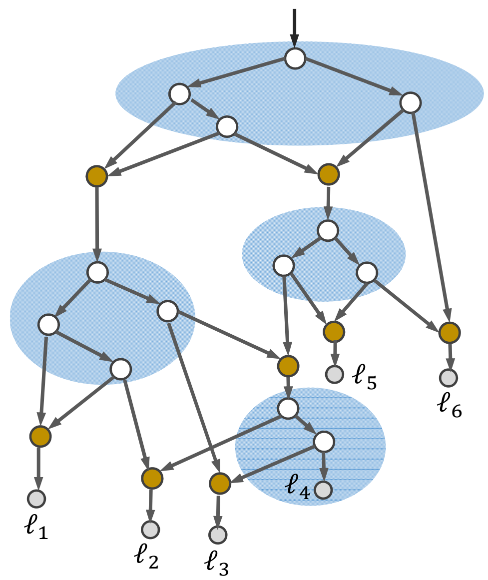


Decomposition Lemma

N: A binary stable network with n leaves

$\mathcal{R}_N$: The set of reticulation nodes

$\mathcal{T}_N$: The set of tree nodes



Consider

$N - \mathcal{R}_N$: The graph restricted on $\mathcal{T}_N$

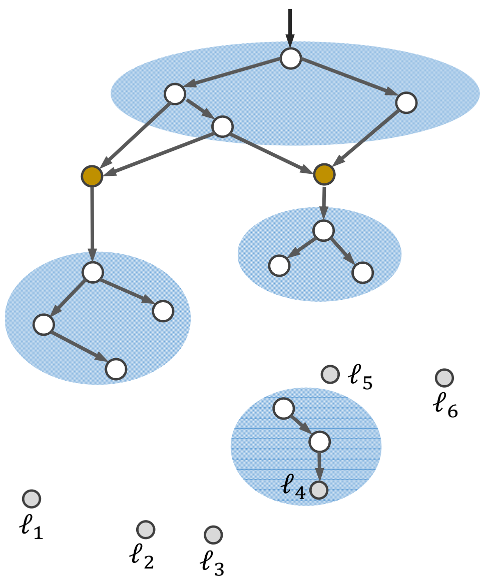
N: A binary stable network with n leaves

$\mathcal{R}_N$: The set of reticulation nodes

$\mathcal{T}_N$: The set of tree nodes

$N - \mathcal{R}_N$: The graph restricted on $\mathcal{T}_N$

- A reticulation node is intra-reticulation if its parents are in the same component of $N - \mathcal{R}_N$;
- A reticulation node is inter-reticulation if its parents are in different components of $N - \mathcal{R}_N$;

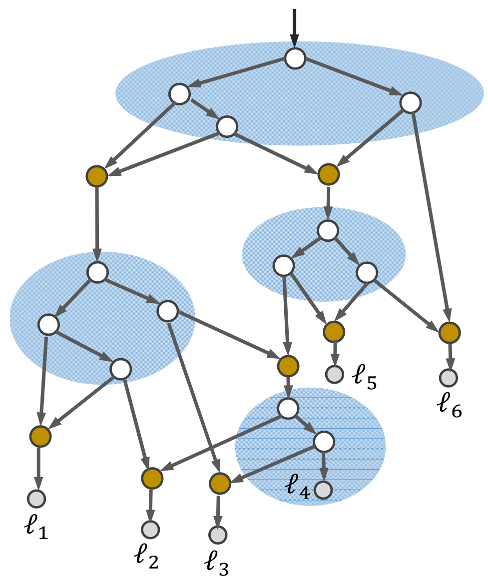


Decomposition Lemma Let N be a binary stable network such that $N - \mathcal{R}_N = C_1 \uplus C_2 \uplus \cdots \uplus C_k$.

(i) Each component is a subtree.

(ii) For each j, $|C_j| = 1$ if and only if it consists of only a network leaf.

(iii) If $|C_j| > 1$, it contains a network leaf or the two parents of a intra-reticulation.



Definition A component is big if its size >1.

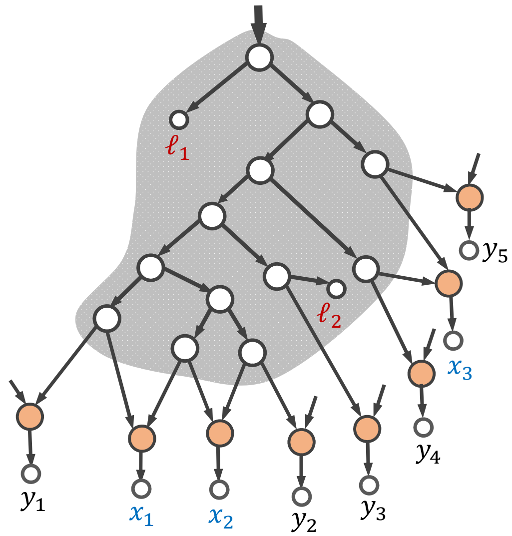
Highlight of Our Algorithm

N: A binary stable network;
T: A binary tree;

- Compute the big components of $N - \mathcal{R}_N$
- Consider a lowest big component C and all nodes below C
- Leaves in C: $\ell_1, \ell_2, \cdots, \ell_j$
- Leaves below intra-reticulations:

  $x_1, x_2, \cdots, x_k$
- Leaves below inter-reticulations:

  $y_1, y_2, \cdots, y_m$



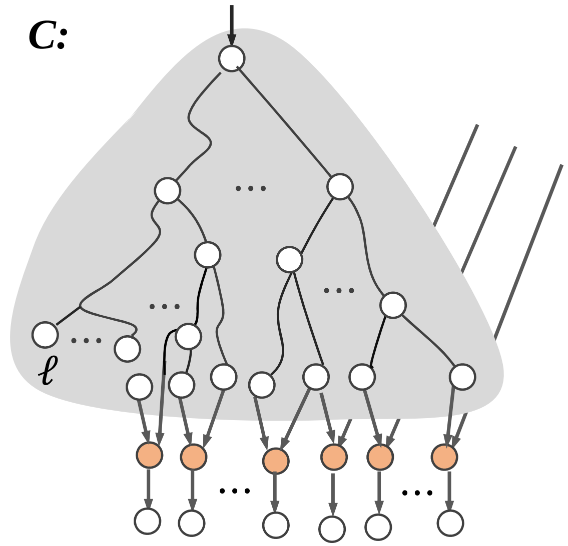
Lemma. (a) $j + k \geq 1$.
(b) The root $\rho(C)$ is stable on each in $\{\ell_i, x_s \mid 1 \leq i \leq j,\ 1 \leq s \leq k\}$.

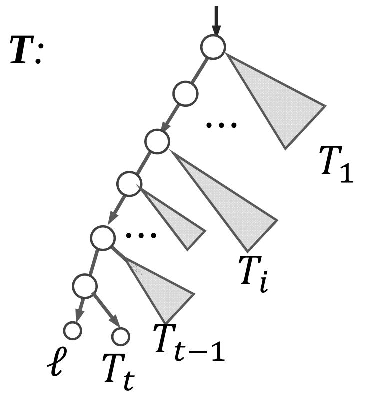
N: A binary stable network
T: A binary tree

- Compute the components of $N - \mathcal{R}_N$
- Consider a lowest big component C and all nodes below C
- Leaves in C: $\ell_1, \ell_2, \cdots, \ell_j$
- Leaves below intra-reticulations:

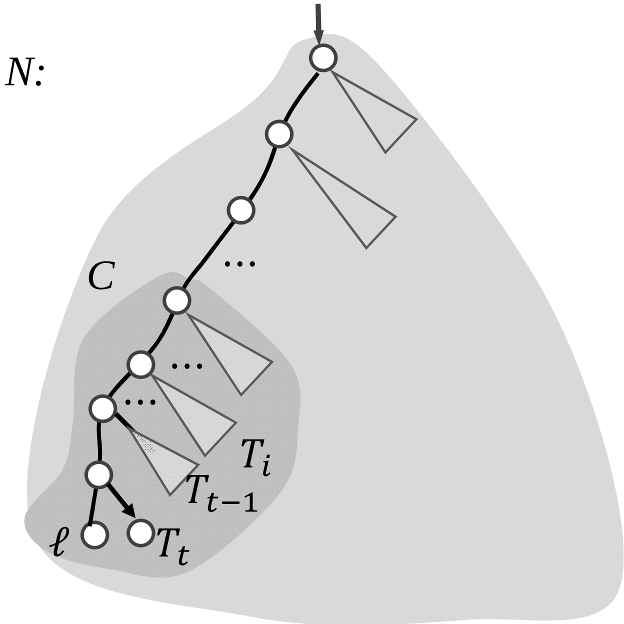
  $x_1, x_2, \cdots, x_k$
- Select $\ell \in L = \{\ell_1, \ell_2, \cdots, \ell_j, x_1, x_2, \cdots, x_k\}$
- Use the path P from ρ(T) to ℓ to partition T as:

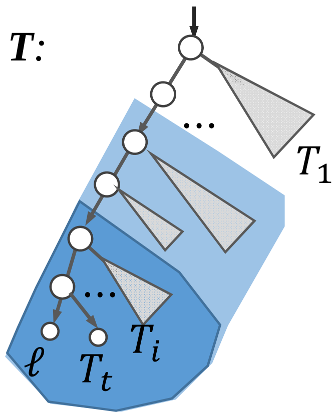
$$T = P + T_1 + T_2 + \cdots + T_t$$





- Consider a lowest big component C and all nodes below C
- Select $\ell \in L = \{\ell_1, \ell_2, \cdots, \ell_j, x_1, x_2, \cdots, x_k\}$
- Use the path P from ρ(T) to ℓ to partition T as $T = P + T_1 + T_2 + \cdots + T_t$



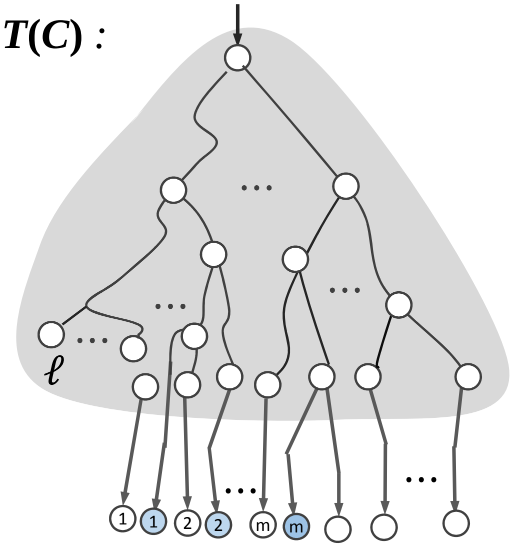


Lemma. Assume N displays T.

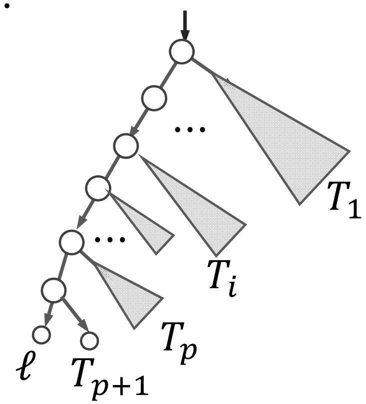
If T_i contains a leaf in L, then, T_i is in C as well as $\cup_{j=i}^{t} T_j$ is also in C.

Compute the largest common subtree of the following two trees:

***T*(*C*) :**



***T*:**



to determine

-- which incoming edge is used for each intra-reticulation (below *C*)

-- whether the incoming edge incident to *C* should be kept or not for each inter-reticulation.

Zhang and Cui, WABI'2010, pp. 300-311

Part 3 Concluding Remarks

- A tight upper bound on the sizes of binary stable networks
- The TCP is solvable in cubic-time for binary stable networks
- The algorithm can be easily modified to solve:

  -- the TCP for non-binary stable networks in $O(E(N)^3)$.

  -- the cluster containment problem (CCP) for stable networks in $O(E(N))$.

- Define new classes of networks?

  -- stable-child networks (SCN).

  -- the TCP and CCP are solvable in poly-time for such networks

  -- #(reticulations) is linear in the number of leaves for such networks.

- How to reconstruct a stable network from a set of clusters or gene trees?

Thank You

Original One

https://www.youtube.com/watch?v=ViKkjSzdwL4

https://www.youtube.com/watch?v=klM9WxH3ijQ

https://www.youtube.com/watch?v=RWGIxK3iPv4