Level-k Phylogenetic Networks are Constructable from a Dense Triplet Set in Polynomial Time

Thu-Hien To   Michel Habib

LIAFA and University Paris Diderot - Paris 7

Combinatorial Pattern Matching – June 22-24 2009

Introduction

- Phylogenetic: infer plausible evolutionary histories from biological data of currently living species

Introduction

- Phylogenetic: infer plausible evolutionary histories from biological data of currently living species
- Biological data: sequences, distance matrix, triplets, quartets, subtrees, characters etc

Introduction

- Phylogenetic: infer plausible evolutionary histories from biological data of currently living species
- Biological data: sequences, distance matrix, triplets, quartets, subtrees, characters etc
- Considered problem: infer phylogenetic networks from a triplet set

Combining triplets

Phylogenetic tree: a binary rooted tree whose each leaf is labeled by a species

Combining triplets

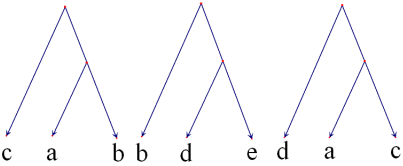
Phylogenetic tree: a binary rooted tree whose each leaf is labeled by a species

Triplet: a phylogenetic tree on 3 species

Combining triplets

Phylogenetic tree: a binary rooted tree whose each leaf is labeled by a species

Triplet: a phylogenetic tree on 3 species

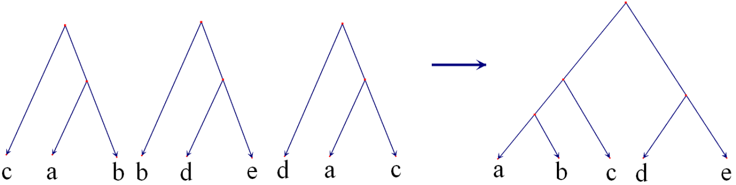


A triplet set $\mathcal{T}$ on the leaf set $\mathcal{L}$

Combining triplets

Phylogenetic tree: a binary rooted tree whose each leaf is labeled by a species

Triplet: a phylogenetic tree on 3 species

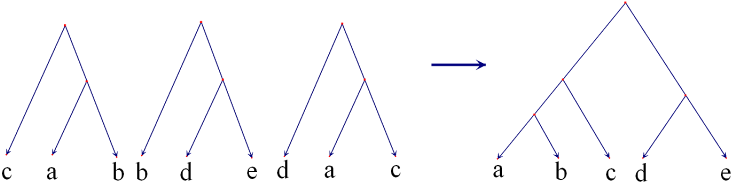


A triplet set $\mathcal{T}$ on the leaf set $\mathcal{L}$

A phylogenetic tree consistent with $\mathcal{T}$

Combining triplets

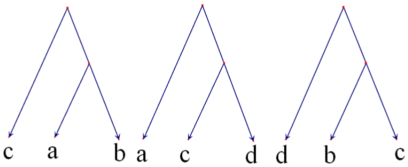
Phylogenetic tree: a binary rooted tree whose each leaf is labeled by a species
Triplet: a phylogenetic tree on 3 species



A triplet set $\mathcal{T}$ on the leaf set $\mathcal{L}$ A phylogenetic tree consistent with $\mathcal{T}$

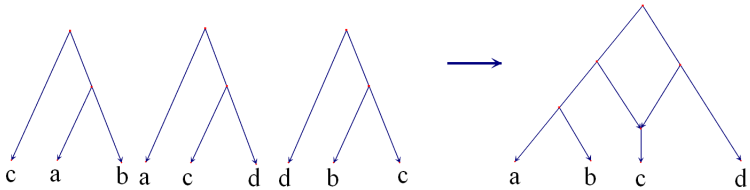
$\Rightarrow O(|\mathcal{T}|.|n|)$ algorithm, with $n = |\mathcal{L}|$, by Aho, Sagiv, Szymanski, and Ullman '81

Conflicting triplets



Conflict !

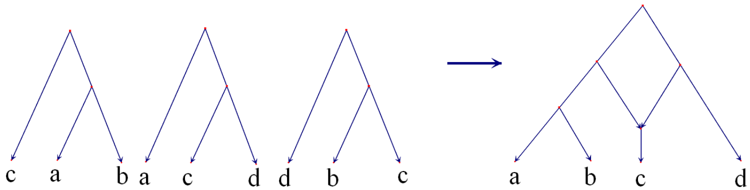
Conflicting triplets



Conflict !

Phylogenetic networks: allow a species to have more than one parent

Conflicting triplets



Conflict !

Phylogenetic networks: allow a species to have more than one parent

Problem

Input: a triplet set $\mathcal{T}$ on a leaf set $\mathcal{L}$

Output: a phylogenetic network consistent with $\mathcal{T}$

Related works - Main result

- If $\mathcal{T}$ is arbitrary: NP-complete for all levels > 0
  Jansson, Nguyen, Sung '06
  Iersel, Keijsper, Kelk, Stougie '08
  Iersel, Kelk, Mnich '09

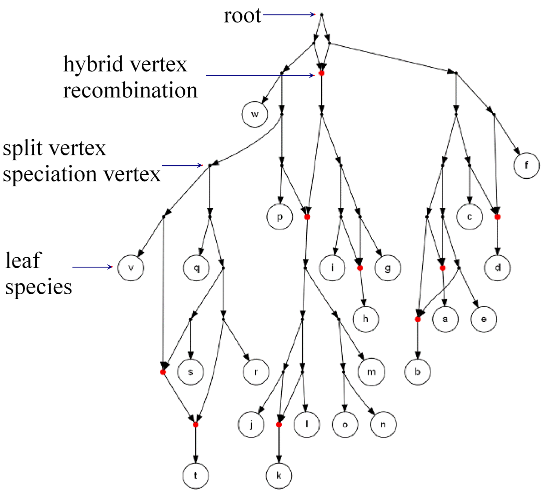
Related works - Main result

- If $\mathcal{T}$ is arbitrary: NP-complete for all levels > 0
  Jansson, Nguyen, Sung '06
  Iersel, Keijsper, Kelk, Stougie '08
  Iersel, Kelk, Mnich '09
- If $\mathcal{T}$ is **dense**, i.e there is at least one triplet in $\mathcal{T}$ on each three leaves, then:
  - $O(n^3)$ algorithm for level-1 networks - Jansson, Nguyen, Sung '04,'06
  - and $O(n^8)$ algorithm for level-2 network - Iersel, Keijsper, Kelk, Stougie '08

Related works - Main result

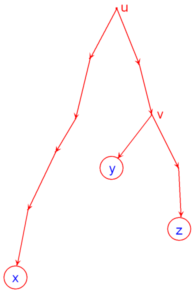
- If $\mathcal{T}$ is arbitrary: NP-complete for all levels > 0
  Jansson, Nguyen, Sung '06
  Iersel, Keijsper, Kelk, Stougie '08
  Iersel, Kelk, Mnich '09
- If $\mathcal{T}$ is **dense**, i.e there is at least one triplet in $\mathcal{T}$ on each three leaves, then:
  - $O(n^3)$ algorithm for level-1 networks - Jansson, Nguyen, Sung '04,'06
  - and $O(n^8)$ algorithm for level-2 network - Iersel, Keijsper, Kelk, Stougie '08
- Main result presented here: If $\mathcal{T}$ is **dense**, for any fixed integer k, it is possible to construct a level-k network in **polynomial time**

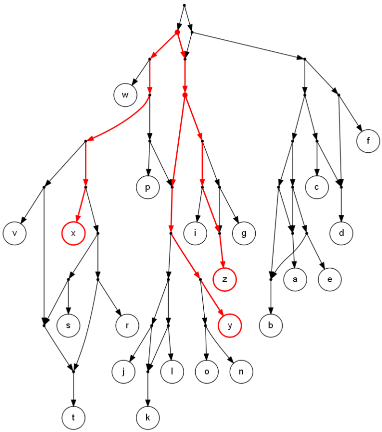
Phylogenetic network



Hybrid vertices model:
- Recombination
- Hybridization
- Horizontal gene transfer
- Ambiguity

Consistency of a triplet with a network

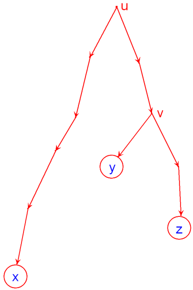




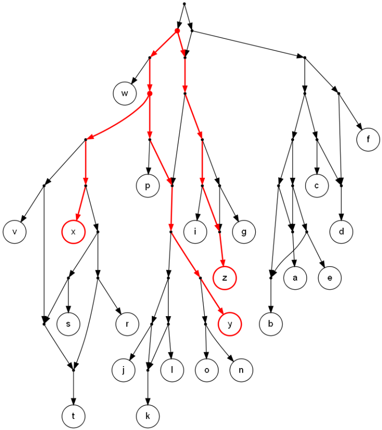
$\exists$ 2 vertices $u \neq v$ and pairwise internally vertex-disjoint paths: $u \rightsquigarrow x$, $u \rightsquigarrow v$, $v \rightsquigarrow y$, $v \rightsquigarrow z$

$x|yz$ is consistent with the network

Consistency of a triplet with a network

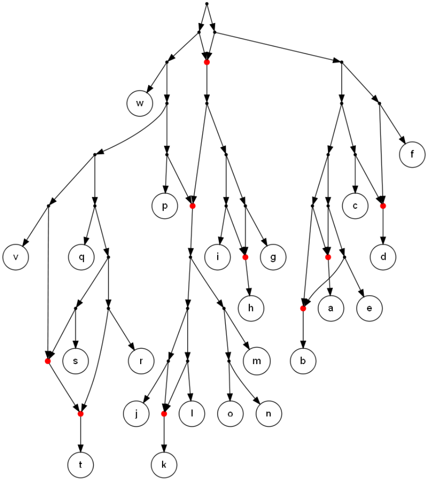


$\exists$ 2 vertices $u \neq v$ and pairwise internally vertex-disjoint paths: $u \rightsquigarrow x$, $u \rightsquigarrow v$, $v \rightsquigarrow y$, $v \rightsquigarrow z$



$z|xy$ is also consistent with the network

Level of a Phylogenetic network



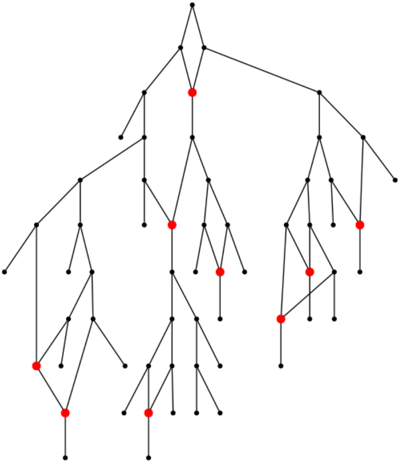
First defined by Choy, Jansson, Sadakane, Sung ’05

- Take the underlying undirected graph $\mathcal{U}(N)$

- Take the biconnected components of $\mathcal{U}(N)$

- Level $k \Leftrightarrow$ each biconnected component has at most k hybrid vertices

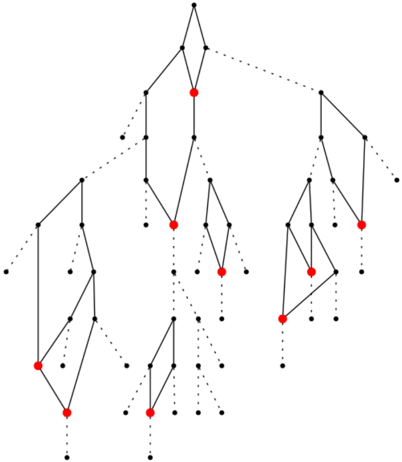
Level of a Phylogenetic network



First defined by Choy, Jansson, Sadakane, Sung '05

- **Take the underlying undirected graph** $\mathcal{U}(N)$
- Take the biconnected components of $\mathcal{U}(N)$
- Level $k \Leftrightarrow$ each biconnected component has at most k hybrid vertices

Level of a Phylogenetic network



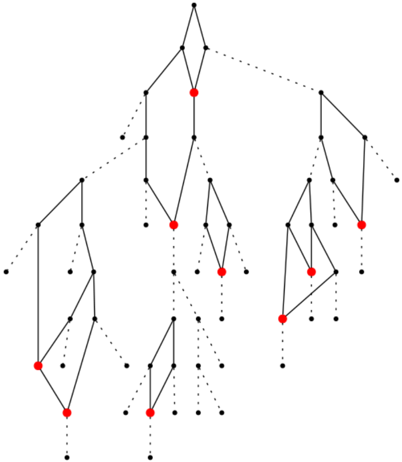
First defined by Choy, Jansson, Sadakane, Sung '05

- Take the underlying undirected graph $\mathcal{U}(N)$

- **Take the biconnected components of $\mathcal{U}(N)$**

- Level k $\Leftrightarrow$ each biconnected component has at most k hybrid vertices

Level of a Phylogenetic network



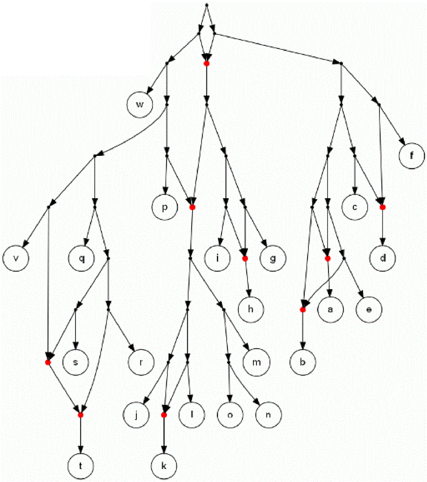
First defined by Choy, Jansson, Sadakane, Sung '05

- Take the underlying undirected graph $\mathcal{U}(N)$

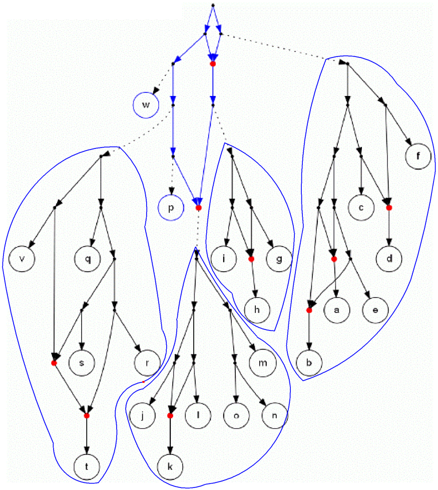
- Take the biconnected components of $\mathcal{U}(N)$

- **Level** k $\Leftrightarrow$ **each biconnected component has at most** k **hybrid vertices**

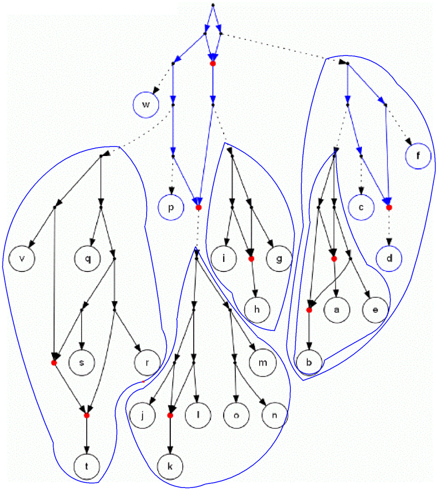
Decompose a Phylogenetic network



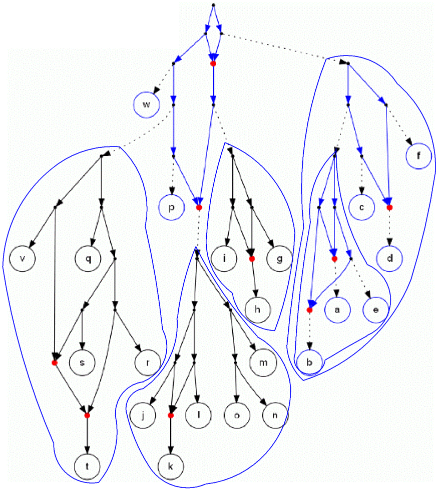
Decompose a Phylogenetic network



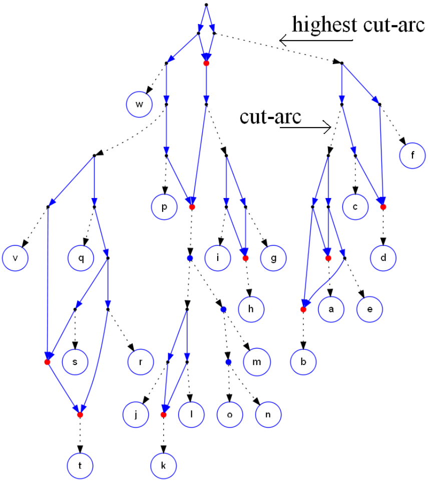
Decompose a Phylogenetic network



Decompose a Phylogenetic network

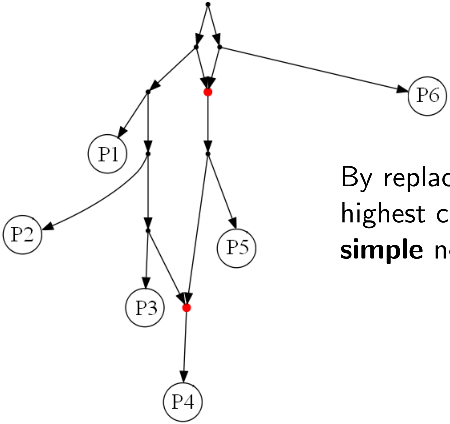


Cut-arc



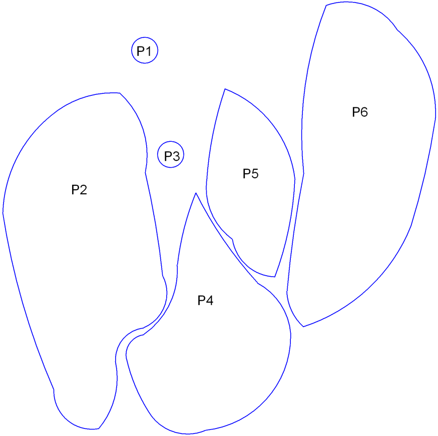
Cut-arc: arc whose removal disconnects the network
Highest cut-arc: cut-arc connects with the highest biconnected component

Simple network



By replacing each sub-network below a highest cut-arc by a leaf, we obtain a **simple** network

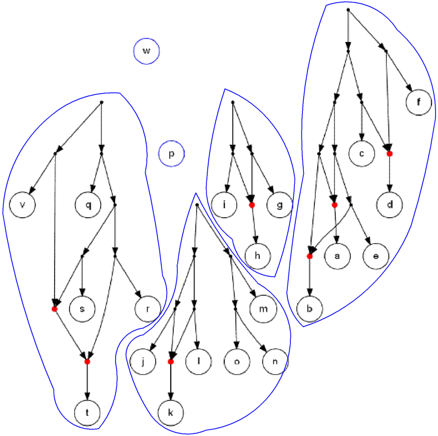
Reconstruct Phylogenetic networks from a Triplet Set



first proposed by Jansson, Sung '04

- **Partition the leaf set** $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$
- Construct recursively a network consistent with $\mathcal{T}|P_i$ on each part P_i
- Construct a simple network N_s consistent with $\mathcal{T}\nabla\mathcal{P}$
- Replace each leaf of N_s by the known corresponding sub-network

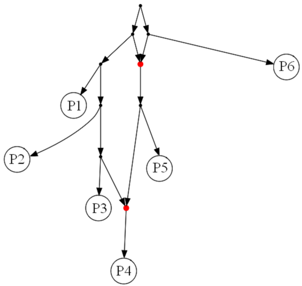
Reconstruct Phylogenetic networks from a Triplet Set



first proposed by Jansson, Sung - 2004

- Partition the leaf set $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$
- **Construct recursively a network consistent with $\mathcal{T}|P_i$ on each part P_i**
- Construct a simple network N_s consistent with $\mathcal{T}\nabla\mathcal{P}$
- Replace each leaf of N_s by the known corresponding sub-network

Reconstruct Phylogenetic networks from a Triplet Set

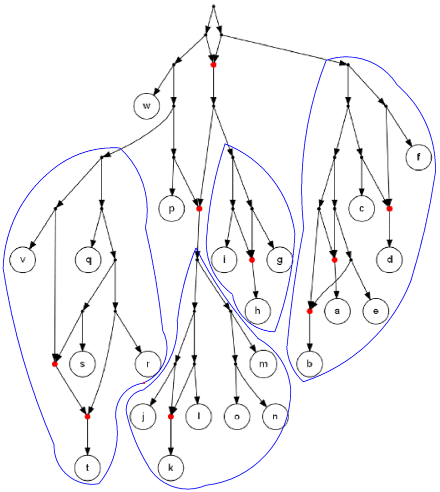


$\mathcal{T}\nabla\mathcal{P} = \{P_i|P_jP_k$ such that i, j, k are pairwise distinct and $\exists x \in P_i, y \in P_j, z \in P_k$ for which $x|yz \in \mathcal{T}\}$

first proposed by Jansson, Sung - 2004

- Partition the leaf set $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$
- Construct recursively a network consistent with $\mathcal{T}|P_i$ on each part P_i
- **Construct a simple network N_s consistent with $\mathcal{T}\nabla\mathcal{P}$**
- Replace each leaf of N_s by the known corresponding sub-network

Reconstruct Phylogenetic networks from a Triplet Set



first proposed by Jansson, Sung - 2004

- Partition the leaf set $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$
- Construct recursively a network consistent with $\mathcal{T}|P_i$ on each part P_i
- Construct a simple network N_s consistent with $\mathcal{T}\nabla\mathcal{P}$
- **Replace each leaf of N_s by the known corresponding sub-network**

Problems

Two problems to solve:

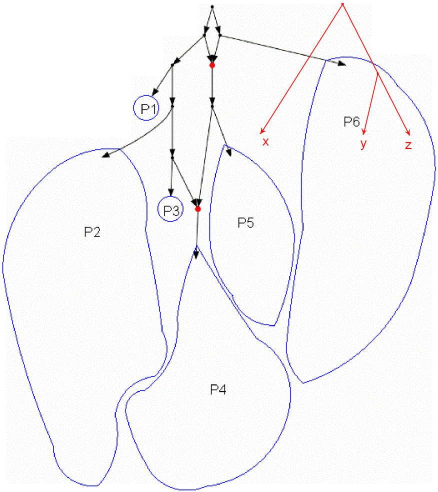
- Construct a simple network consistent with $\mathcal{T}\nabla\mathcal{P}$
  Exist an $O(n^{k+1})$ algorithm to find all simple networks consistent with a dense triplet set - Iersel,Kelk '08

Problems

Two problems to solve:

- Construct a simple network consistent with $\mathcal{T}\nabla\mathcal{P}$
  Exist an $O(n^{k+1})$ algorithm to find all simple networks consistent with a dense triplet set - Iersel,Kelk '08
- Find all possible partitions $\mathcal{P}$ of the leaf set

Properties of a Partition of the Leaf Set



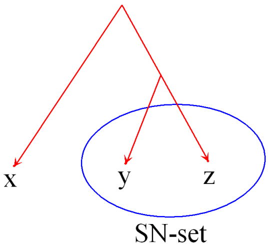
$\forall x \notin P_i$ and $y, z \in P_i$, the only triplet on three leaves x, y, z, if there is any, is $x|yz$

SN-set

SN-set

first defined by Jansson, Sung '04
Let $A \subseteq \mathcal{L}$. A is a SN-set, or Simple Network set, if either:
- it is a singleton
- or the whole $\mathcal{L}$
- or $\forall x \in \mathcal{L} \backslash A$, $y, z \in A$, the only triplet on $\{x, y, z\}$ in $\mathcal{T}$, if there is any, is $x|yz$.

Remark: Each part of the partition is a SN-set

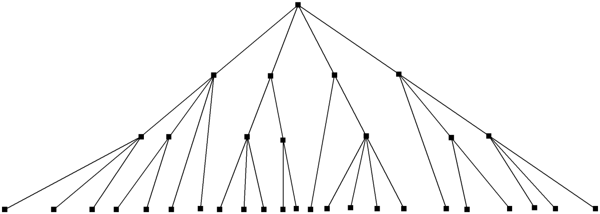


SN-tree

If $\mathcal{T}$ is **dense**, the collection of all SN-sets is laminar - Jansson, Nguyen, Sung '06

SN-tree

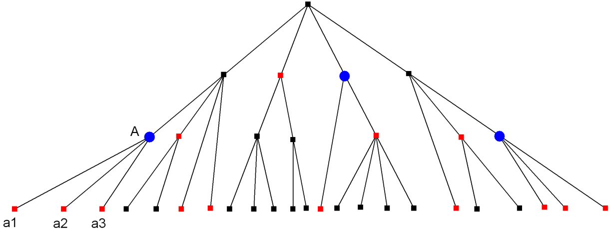
If $\mathcal{T}$ is **dense**, the collection of all SN-sets is laminar - Jansson, Nguyen, Sung '06
$\Rightarrow$ is tree-representable



- each node of the SN-tree represents a SN-set
- the number of non-singleton SN-sets is $O(n)$ with $n = |\mathcal{L}|$
- SN-tree can be computed from $\mathcal{T}$ in $O(n^3)$ - Jansson,Nguyen,Sung '06

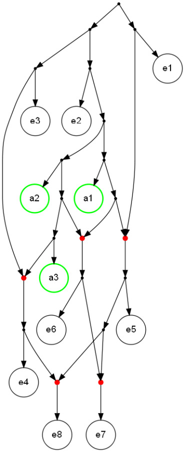
Characterize a partition by split SN-sets

Remark: **A partition** $\Leftrightarrow$ **a set of split SN-sets**



split SN-set: each child of a split SN-set is a part of the partition.

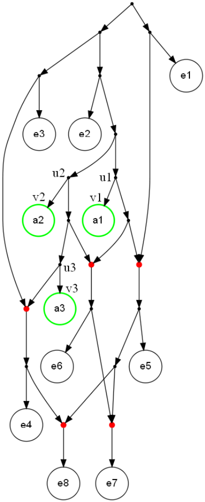
A SN-set split in a network



$A = a_1 \cup a_2 \cup a_3$ is split in this network

Fact: a SN-set is split in a network iff each of its children is hung below a different highest cut-arc of this network

A function from a split SN-set to a set of hybrid vertices

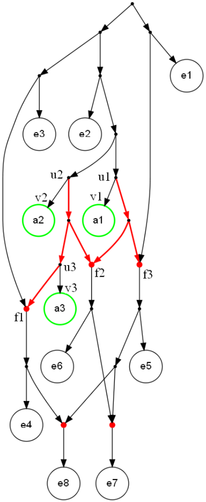


Define a function f from a split SN-set to a set of hybrid vertices

- Let $A = a_1 \cup \cdots \cup a_m$ be a split SN-set. Each a_i is hung below a highest cut-arc (u_i, v_i)

- H is the set of hybrid vertices of the highest biconnected component $\Rightarrow |H| \leq k$

A function from a split SN-set to a set of hybrid vertices



Define a function f from a split SN-set to a set of hybrid vertices

- Let $A = a_1 \cup \cdots \cup a_m$ be a split SN-set. Each a_i is hung below a highest cut-arc (u_i, v_i)

- H is the set of hybrid vertices of the highest biconnected component $\Rightarrow |H| \leq k$

$f(A) = \{h \in H | \exists i$ so that $u_i \rightsquigarrow h$ and the path from u_i to h does not contain any internal hybrid vertex$\}$

Bound the number of split SN-sets in a level-k network

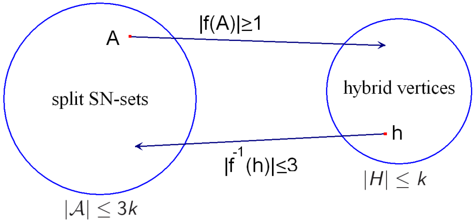
Lemma 1

Let N be any network consistent with $\mathcal{T}$, then
(i) $f(A) \neq \emptyset$ for any SN-set A split in N
(ii) $\forall h \in H$, there are at most **three** SN-sets split in N, whose image by f contains h

Bound the number of split SN-sets in a level-k network

Lemma 1

Let N be any network consistent with $\mathcal{T}$, then
(i) $f(A) \neq \emptyset$ for any SN-set A split in N
(ii) $\forall h \in H$, there are at most **three** SN-sets split in N, whose image by f contains h



Bound the number of split SN-sets in a level-k network

Lemma 1

Let N be any network consistent with $\mathcal{T}$, then
(i) $f(A) \neq \emptyset$ for any SN-set A split in N
(ii) $\forall h \in H$, there are at most **three** SN-sets split in N, whose image by f contains h

Lemma 2

Let $\mathcal{T}$ be a dense triplet set. For any level-k network N consistent with $\mathcal{T}$, there are at most $3k$ SN-sets of $\mathcal{T}$ which are split in N

Theorem

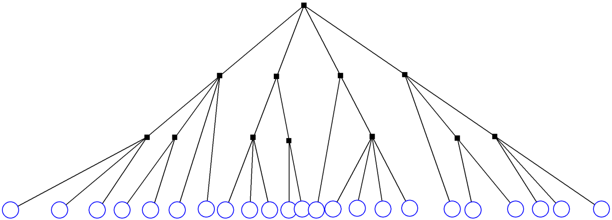
Given a dense triplet set $\mathcal{T}$, and a fixed positive integer k, it is possible to construct a level-k network consistent with $\mathcal{T}$, if one exists, in $O(|\mathcal{T}|^{k+1}n^{3k+1})$ time.

Proof of complexity

Similar to those of level-1 and level-2 networks, we construct on each SN-set A, in small-big order, a sub-network consistent with $\mathcal{T}|A$

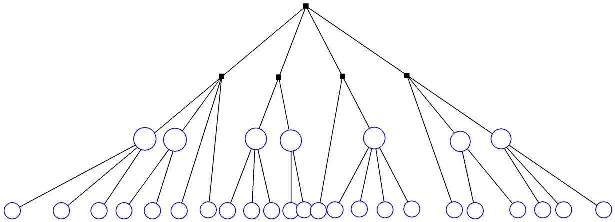
Proof of complexity

Similar to those of level-1 and level-2 networks, we construct on each SN-set A, in small-big order, a sub-network consistent with $\mathcal{T}|A$



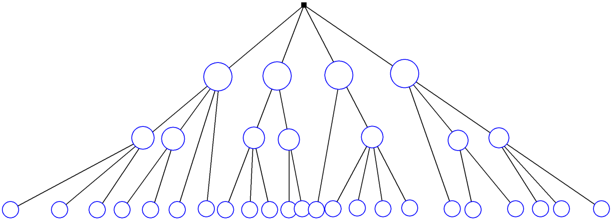
Proof of complexity

Similar to those of level-1 and level-2 networks, we construct on each SN-set A, in small-big order, a sub-network consistent with $\mathcal{T}|A$



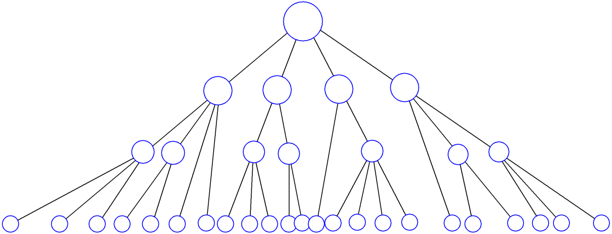
Proof of complexity

Similar to those of level-1 and level-2 networks, we construct on each SN-set A, in small-big order, a sub-network consistent with $\mathcal{T}|A$



Proof of complexity

Similar to those of level-1 and level-2 networks, we construct on each SN-set A, in small-big order, a sub-network consistent with $\mathcal{T}|A$



There are $O(n)$ non-singleton SN-sets
$\Rightarrow O(n)$ constructions of a network on a SN-set by knowing a sub-network on each smaller SN-set

Proof of complexity

For each of the $O(n)$ constructions:
- Constructing a simple level-k network: $O(|\mathcal{T}|^{k+1})$ - Iersel,Kelk '08
- The number of partitions = the number of sets of split SN-sets : $O(n^{3k})$

Proof of complexity

For each of the $O(n)$ constructions:
- Constructing a simple level-k network: $O(|\mathcal{T}|^{k+1})$ - Iersel,Kelk '08
- The number of partitions = the number of sets of split SN-sets : $O(n^{3k})$

$\Rightarrow$ Total: $O(|\mathcal{T}|^{k+1} n^{3k+1})$

The network with the minimum number of recombinations

Theorem

Given a dense triplet set $\mathcal{T}$, and a fixed positive integer k, it is possible to construct a level-k network consistent with $\mathcal{T}$ with the minimum number of recombinations, if one exists, in $O(|\mathcal{T}|^{k+1}n^{3k+1})$ time.

The network with the minimum number of recombinations

Theorem

Given a dense triplet set $\mathcal{T}$, and a fixed positive integer k, it is possible to construct a level-k network consistent with $\mathcal{T}$ with the minimum number of recombinations, if one exists, in $O(|\mathcal{T}|^{k+1}n^{3k+1})$ time.

Idea for the proof: If N is a level-k network with the minimum number of recombinations, then each sub-network below a highest cut-arc of N is also the one with the minimum number of recombinations

Conclusion

- Constructing a level-k network consistent with a dense triplet set with the minimum number of recombinations is polynomial with any fixed k

Conclusion

- Constructing a level-k network consistent with a dense triplet set with the minimum number of recombinations is polynomial with any fixed k
- Better bound? better algorithm?

Conclusion

- Constructing a level-k network consistent with a dense triplet set with the minimum number of recombinations is polynomial with any fixed k
- Better bound? better algorithm?
- Relax the condition of density on the triplet set?

Conclusion

- Constructing a level-k network consistent with a dense triplet set with the minimum number of recombinations is polynomial with any fixed k
- Better bound? better algorithm?
- Relax the condition of density on the triplet set?
- Open problem: the network with the minimum level consistent with a dense triplet set? NP-complete?

Conclusion

- Constructing a level-k network consistent with a dense triplet set with the minimum number of recombinations is polynomial with any fixed k
- Better bound? better algorithm?
- Relax the condition of density on the triplet set?
- Open problem: the network with the minimum level consistent with a dense triplet set? NP-complete?

Conclusion

- Constructing a level-k network consistent with a dense triplet set with the minimum number of recombinations is polynomial with any fixed k
- Better bound? better algorithm?
- Relax the condition of density on the triplet set?
- Open problem: the network with the minimum level consistent with a dense triplet set? NP-complete?

Thank you