Maximum agreement and compatible supertrees

VINCENT BERRY & FRANÇOIS NICOLAS

Méthodes et Algorithmes pour la Bioinformatique

L.I.R.M.M. (Université Montpellier II - C.N.R.S.)

`http://www.lirmm.fr/~w3ifa/MAAS`

Supported by the *Action Incitative Informatique-Mathématique-Physique en Biologie Moléculaire* [ACI IMP-Bio].

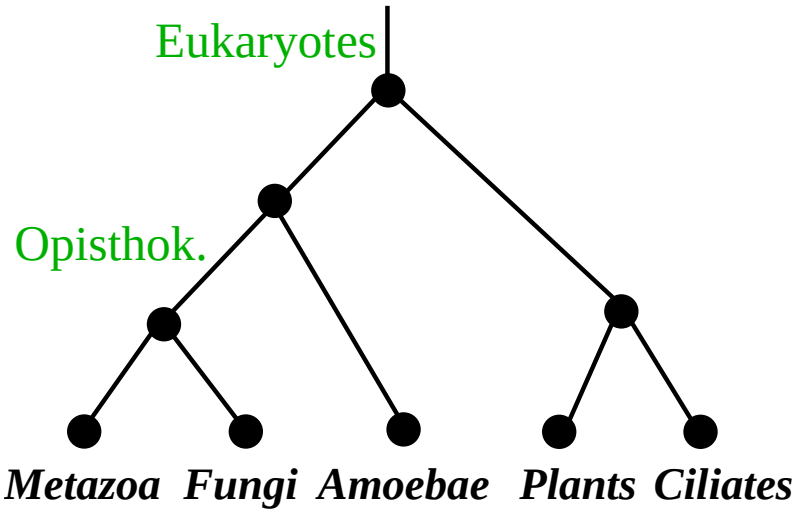
Outline of the talk

I - **Trees with identical leaf sets**

- Consensus of a collection of trees
- Maximum agreement subtree
- Maximum compatible tree
- Results on MAST and MCT

II - **Trees with overlapping leaf sets**

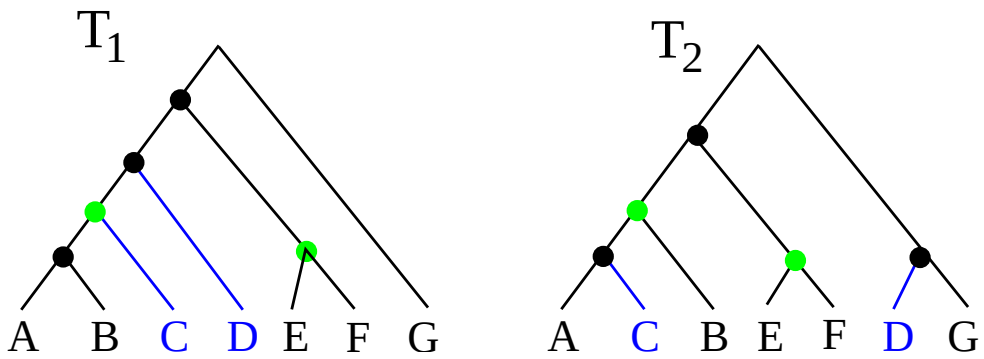
- Supertree context
- Defining SMAST and SMCT
- Solving the case of two trees
- Other results (intractability, etc)



Evolutionary tree

Consensus of evolutionary trees

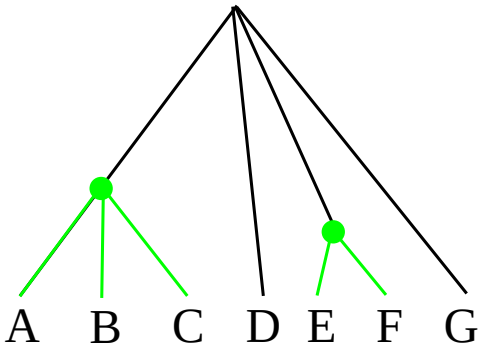
The source trees usually *conflict* on the position of some leaves.



Approaches not contradicting any input information:

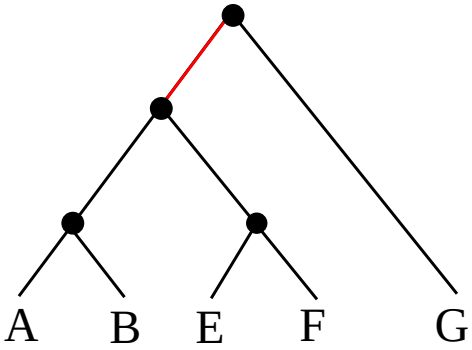
Strict consensus

removing input clades



Maximum Agreement SubTree

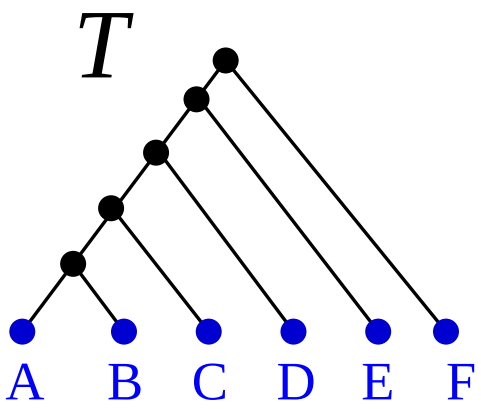
removing input leaves



Definitions

The following definitions apply on rooted and unrooted trees.

- $L(T)$ denotes the leaf set of a tree T.
- The size of T, denoted $\#T$, is the number of its leaves.



$L(T)=\{A,B,C,D,E,F\}$

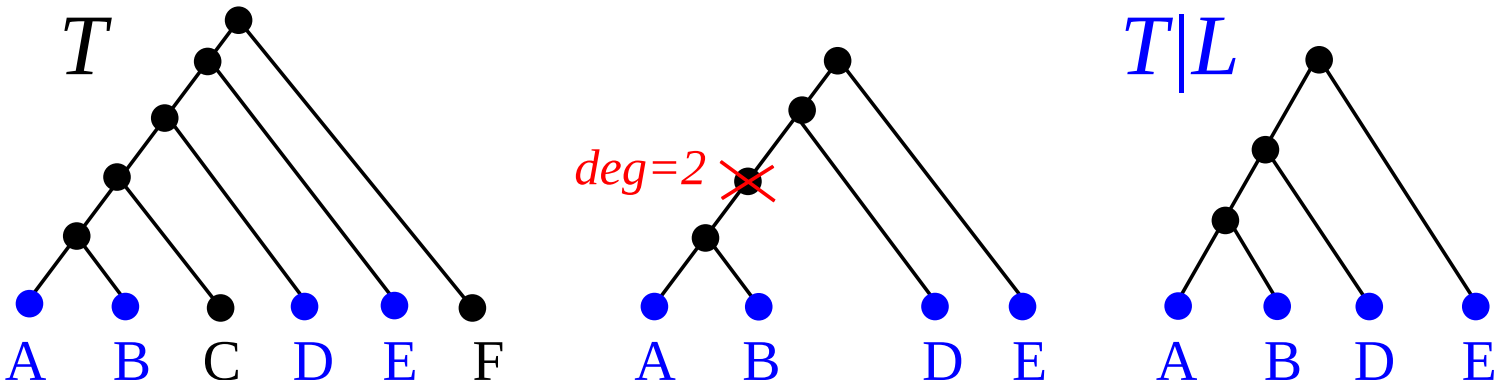
$\#T = 6$

Definitions

Let T be a tree and L be a set of labels

- $T|L$ is the topological restriction of T to leaves with label in L.

E.g., if $L = \{A, B, D, E, G\}$:

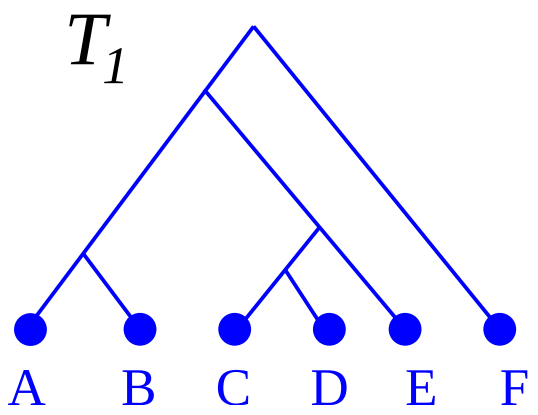


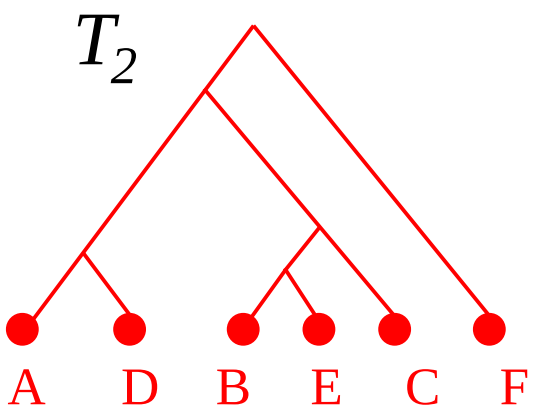
Maximum Agreement Subtree - MAST

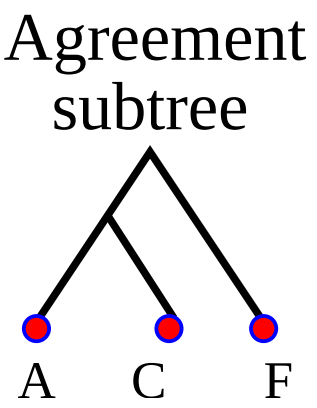
Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a collection of trees on a same set of leaves L.

T is an agreement subtree of $\mathcal{T}$ iff

- $L(T) \subseteq L$
- $\forall T_i \in \mathcal{T},\ T = T_i|L(T)$





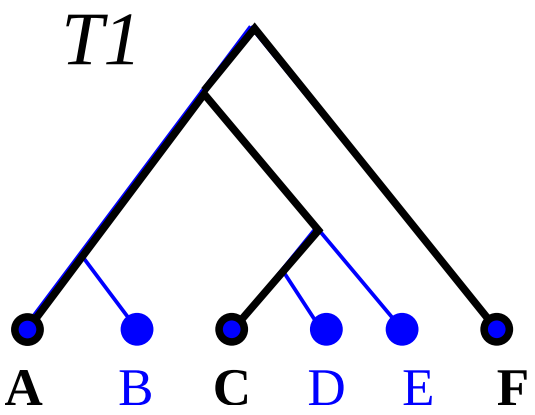


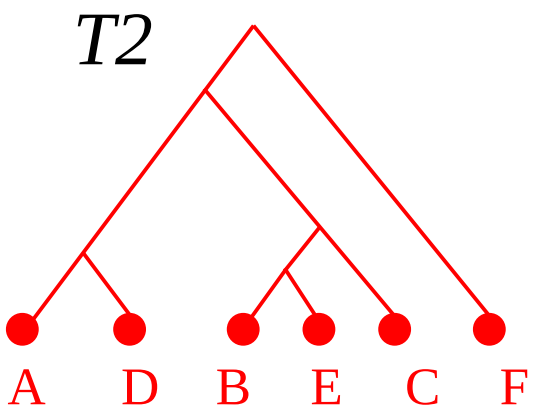
Maximum Agreement Subtree - MAST

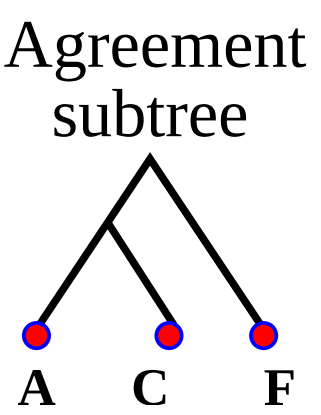
Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a collection of trees on a same set of leaves L.

T is an agreement subtree of $\mathcal{T}$ iff

- $L(T) \subseteq L$
- $\forall T_i \in \mathcal{T},\ T = T_i|L(T)$





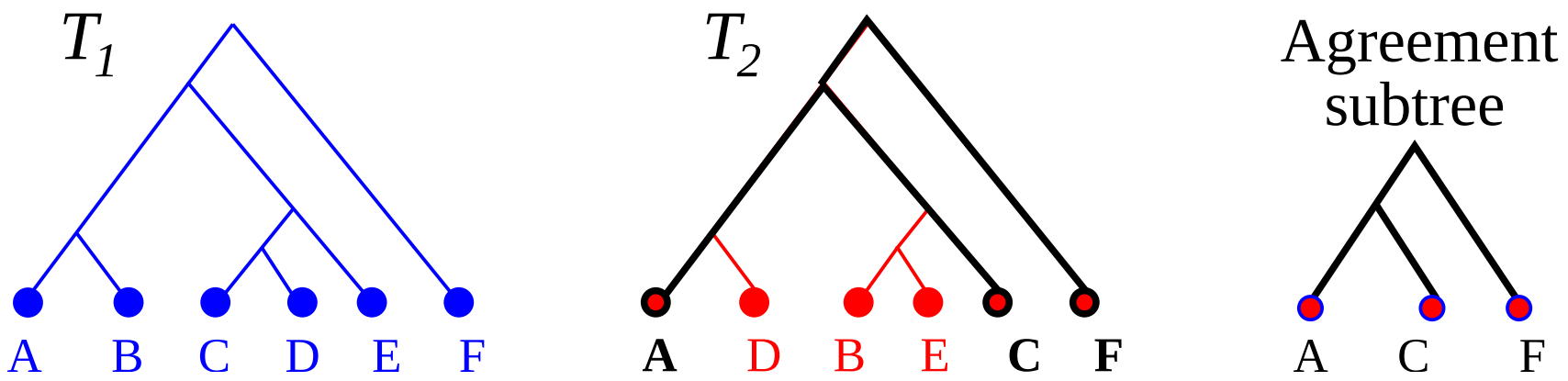


Maximum Agreement Subtree - MAST

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a collection of trees on a same set of leaves L.

T is an agreement subtree of $\mathcal{T}$ iff

- $L(T) \subseteq L$
- $\forall T_i \in \mathcal{T},\ T = T_i|L(T)$

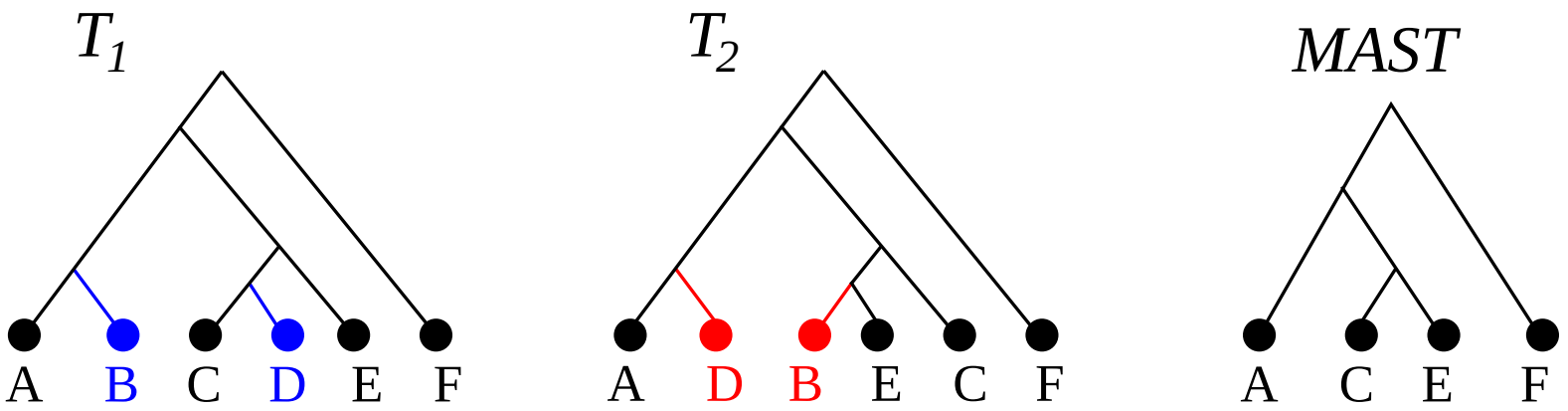


Maximum Agreement Subtree - MAST

T is an agreement subtree of $\mathcal{T}$ iff

- $L(T) \subseteq L$
- $\forall T_i \in \mathcal{T},\ T = T_i|L(T)$

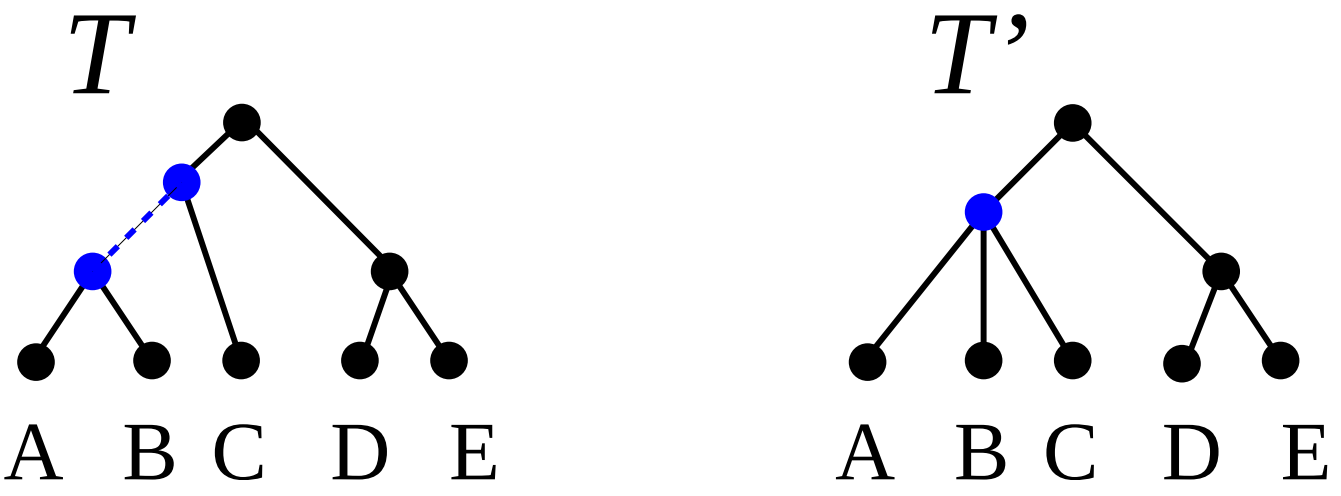
T is a Maximum Agreement SubTree (MAST) of $\mathcal{T}$ iff
it is of maximum size among all agreement subtrees of $\mathcal{T}$.



Maximum Compatible Tree - MCT

A tree T *refines* a tree T', denoted $T \trianglerighteq T'$,

if T' can be obtained from T by contracting some edges of T:

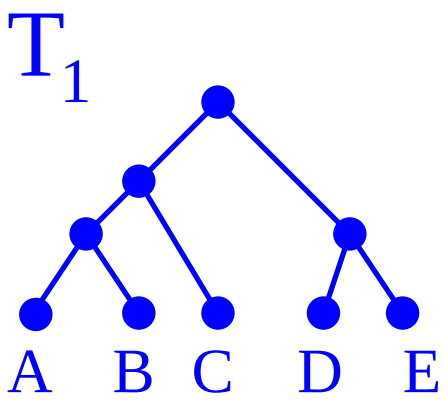


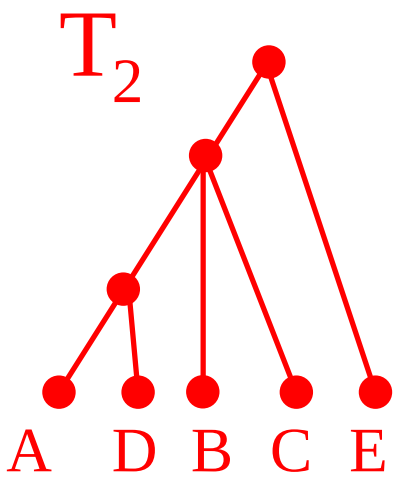
Maximum Compatible Tree - MCT

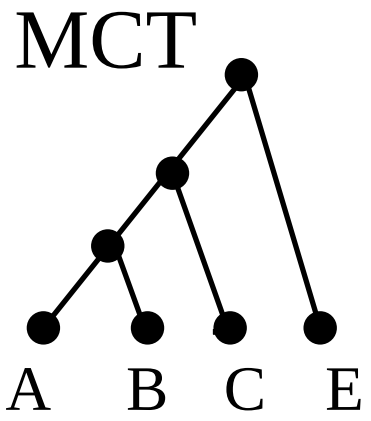
A tree T is compatible with a collection $\mathcal{T} = \{T_1, \ldots, T_k\}$ iff

- $L(T) \subseteq L$
- $\forall T_i \in \mathcal{T},\ T \trianglerighteq T_i | L(T)$

T is a Maximum Compatible Tree (MCT) of $\mathcal{T}$ iff it is of maximum size among all trees compatible with $\mathcal{T}$.





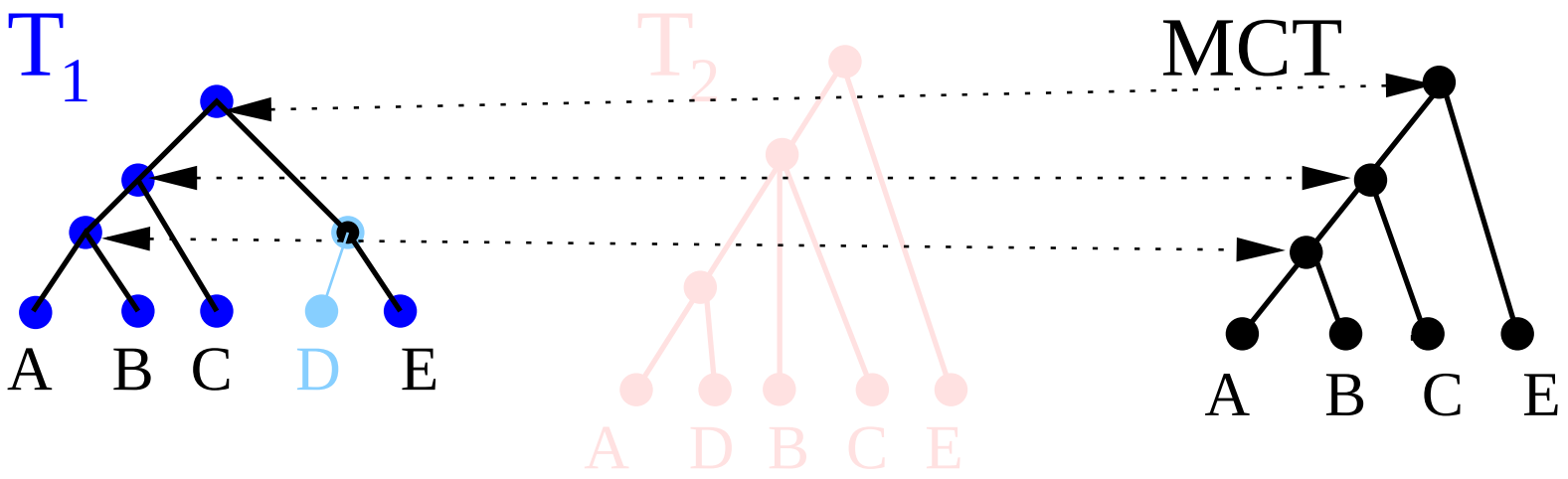


Maximum Compatible Tree - MCT

A tree T is compatible with a collection $\mathcal{T} = \{T_1, \ldots, T_k\}$ iff

- $L(T) \subseteq L$
- $\forall T_i \in \mathcal{T},\ T \trianglerighteq T_i | L(T)$

T is a Maximum Compatible Tree (MCT) of $\mathcal{T}$ iff it is of maximum size among all trees compatible with $\mathcal{T}$.

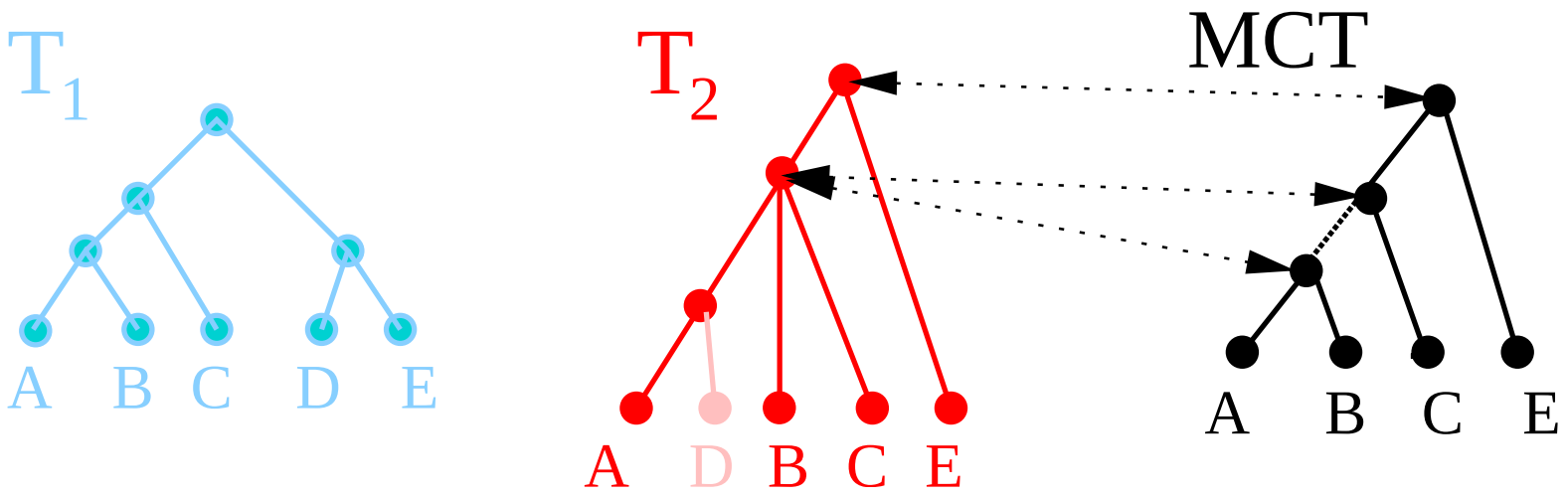


Maximum Compatible Tree - MCT

A tree T is compatible with a collection $\mathcal{T} = \{T_1, \ldots, T_k\}$ iff

- $L(T) \subseteq L$
- $\forall T_i \in \mathcal{T},\ T \trianglerighteq T_i | L(T)$

T is a Maximum Compatible Tree (MCT) of $\mathcal{T}$ iff it is of maximum size among all trees compatible with $\mathcal{T}$.



Results on MAST and MCT

- An $O(\min(3^p nk, 2.27^p + kn^3))$ FPT algorithm for MAST and MCT.

  n is the number of input leaves

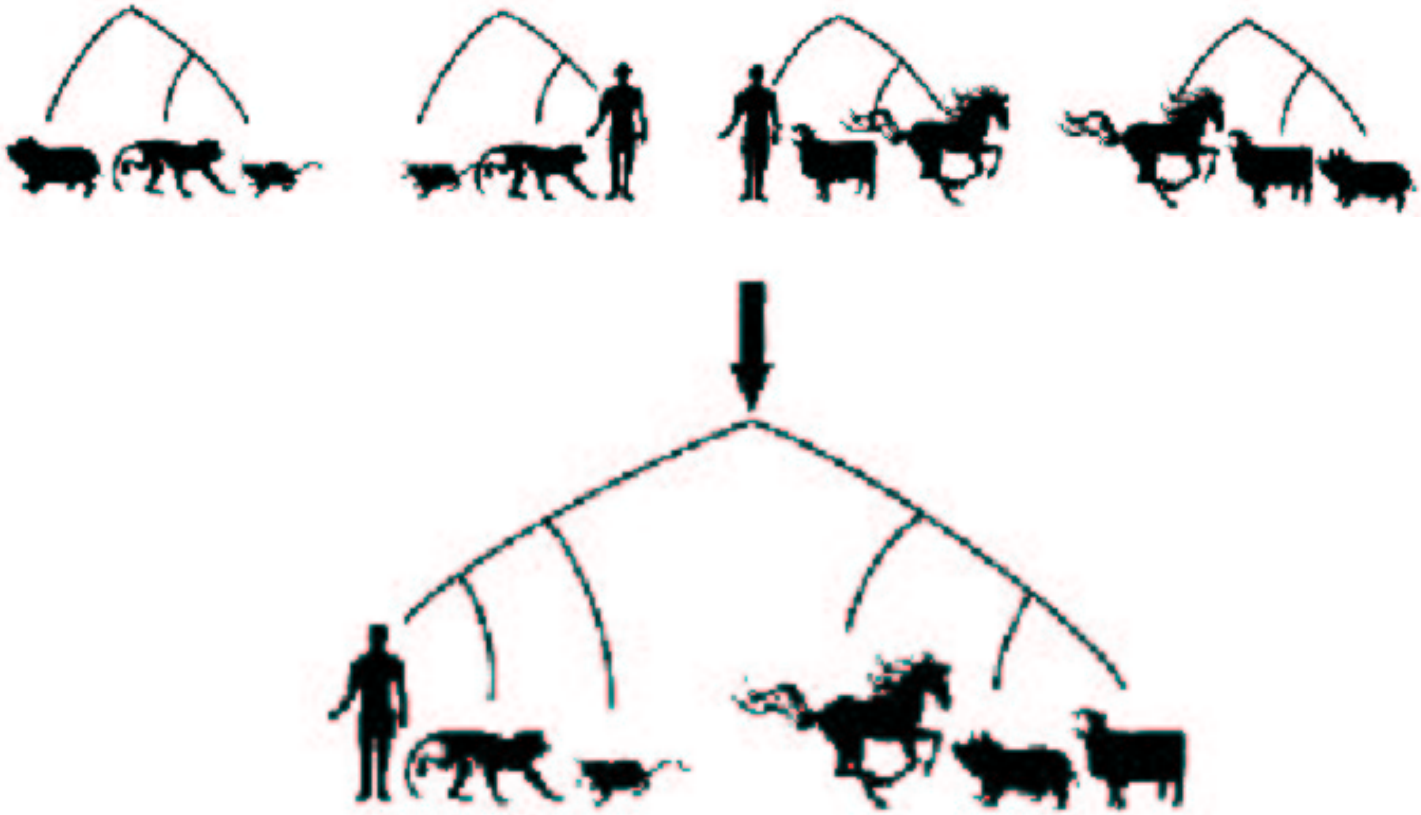
  p is the number of input leaves to remove to obtain agreement, resp. compatibility.

$\Rightarrow$ Fixing $p = 0$ gives a simple linear time algorithm for solving the *perfect phylogeny* (2-state) problem and the *character compatibility* problem.

- A 3-approximation algorithm running in $O(kn^3)$ for the complement of the MAST and MCT problems.

Results apply for rooted or unrooted collections of trees.

Extending MAST and MCT to the supertree context



Maximum agreement supertree - SMAST

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a collection of trees with overlapping sets of leaves and let $L(\mathcal{T}) := \cup L(T_i)$.

T is an agreement supertree of $\mathcal{T}$ iff

- $L(T) \subseteq L(\mathcal{T})$
- $\forall T_i \in \mathcal{T}, \quad T|L(T_i) = T_i|L(T)$

Maximum agreement supertree - SMAST

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a collection of trees with overlapping sets of leaves and let $L(\mathcal{T}) := \cup L(T_i)$.

T is an agreement supertree of $\mathcal{T}$ iff

- $L(T) \subseteq L(\mathcal{T})$
- $\forall T_i \in \mathcal{T}, \quad T|L(T_i) = T_i|L(T)$

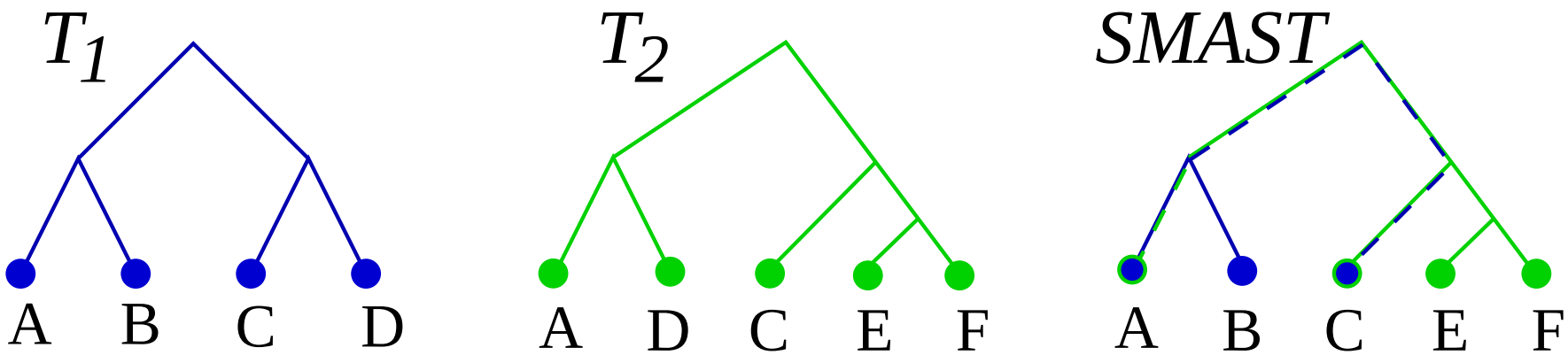
T is a maximum agreement supertree ($SMAST$) iff it is maximum in size among agreement supertrees of $\mathcal{T}$.

Maximum agreement supertree - SMAST

T is an agreement supertree of $\mathcal{T}$ iff

- $L(T) \subseteq L(\mathcal{T})$
- $\forall T_i \in \mathcal{T}, \quad T|L(T_i) = T_i|L(T)$

T is a maximum agreement supertree ($SMAST$) iff it is maximum in size among all agreement supertrees of $\mathcal{T}$.



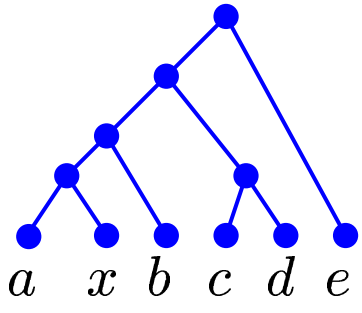
Maximum compatible supertree - SMCT

T is a supertree compatible with $\mathcal{T}$ iff

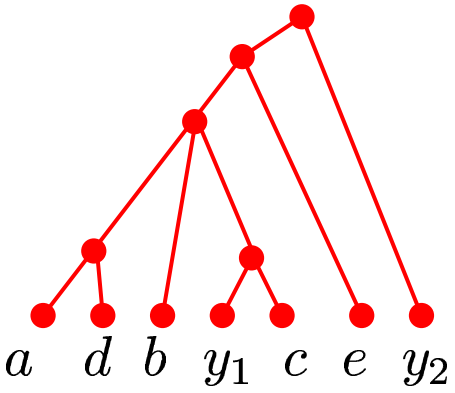
- $L(T) \subseteq L(\mathcal{T})$
- $\forall T_i \in \mathcal{T}, \quad T|L(T_i) \trianglerighteq T_i|L(T)$

T is a maximum compatible supertree ($SMCT$) iff it is maximum in size among all supertrees compatible with $\mathcal{T}$.

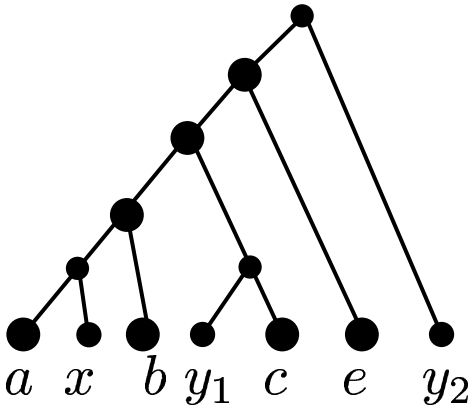
T_1



T_2



$SMCT(\mathcal{T})$

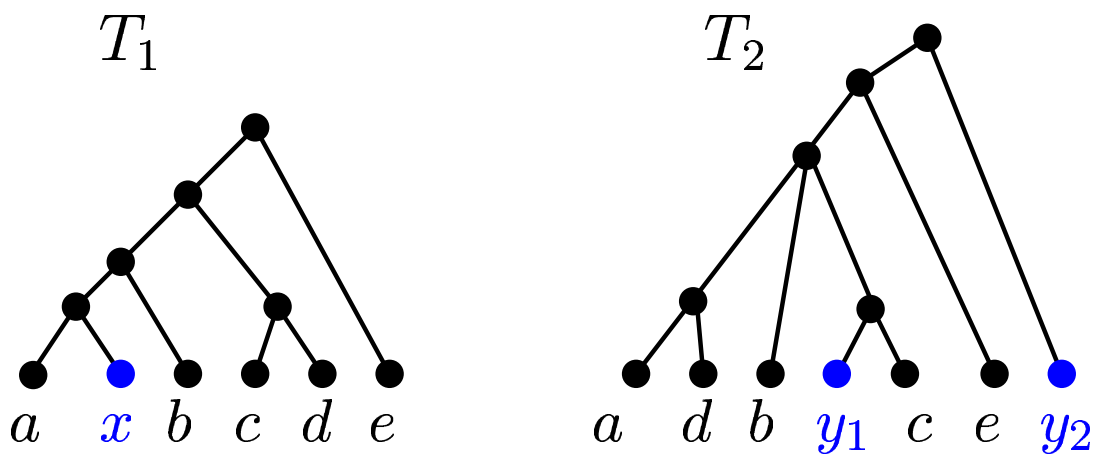


Definition

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a collection of trees with overlapping sets of leaves

- A leaf is specific if it appears in only one source tree.

E.g., $\{x, y_1, y_2\}$:

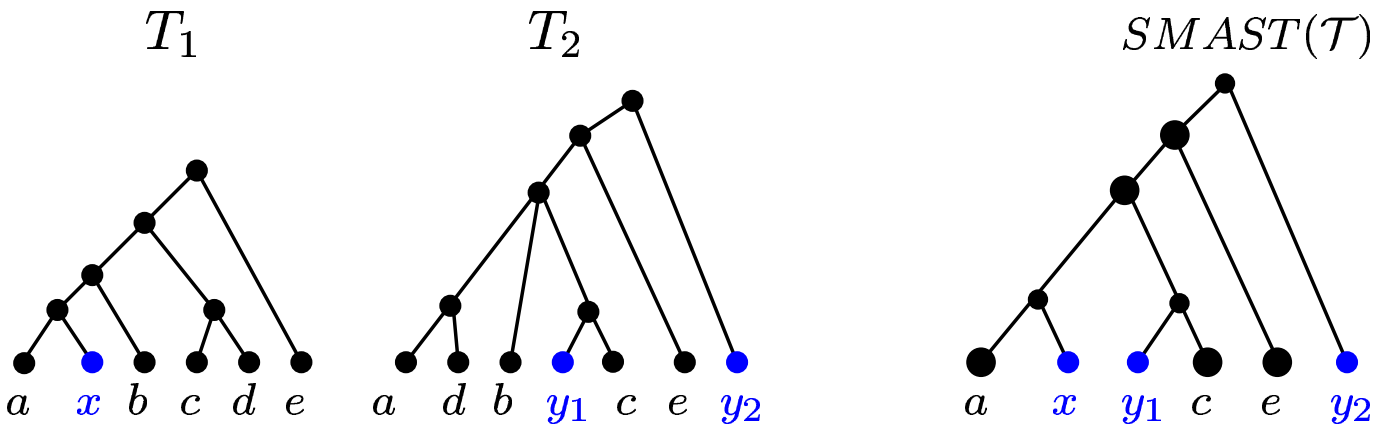


Inclusion of *specific* leaves

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a collection of tree with overlapping sets of leaves

- A leaf is specific if it appears in only one source tree. E.g., $\{x, y_1, y_2\}$:

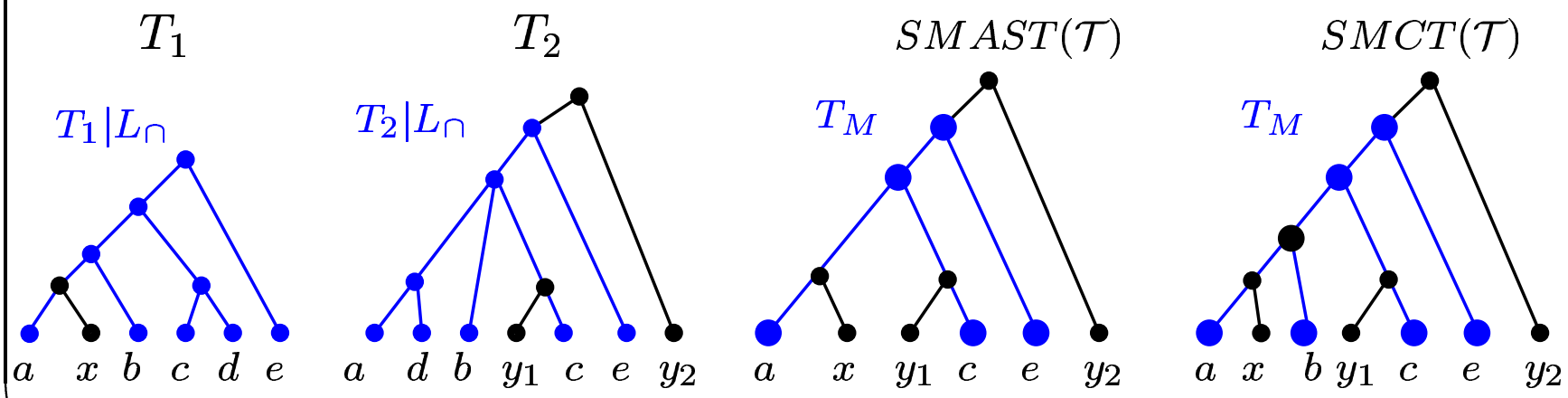
Lemma: Any tree $SMAST(\mathcal{T})$ and $SMCT(\mathcal{T})$ includes all specific leaves.



Solving the case of two trees

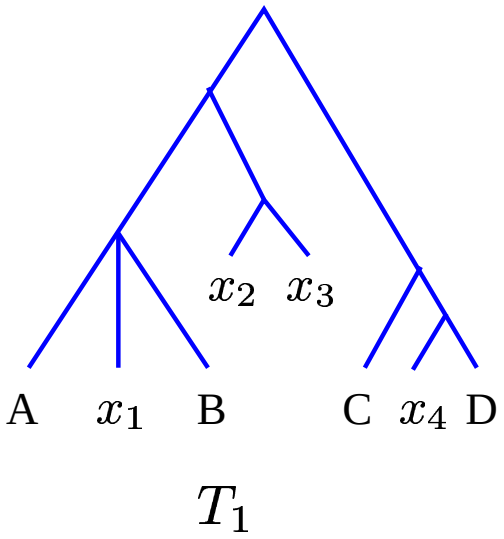
Let T_1, T_2 two source trees, define $L_\cap := L(T_1) \cap L(T_2)$.

Lemma: Any tree $T_M := MAST(T_1|L_\cap, T_2|L_\cap)$ is the restriction to $L_\cap$ of some tree $SMAST(T_1, T_2)$.
The same result holds between MCT and $SMCT$.

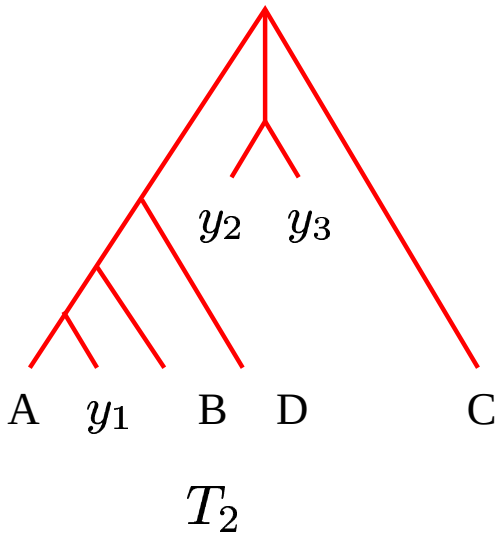


Computing $SMAST(T_1, T_2)$

Given T_1, T_2 two trees on overlapping sets of leaves.



T_1

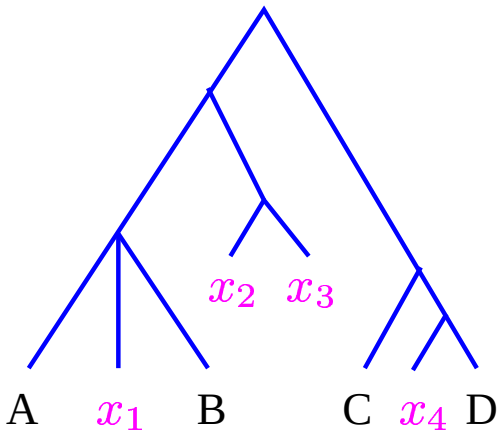


T_2

Computing $SMAST(T_1, T_2)$

Given T_1, T_2 two trees on overlapping sets of leaves.

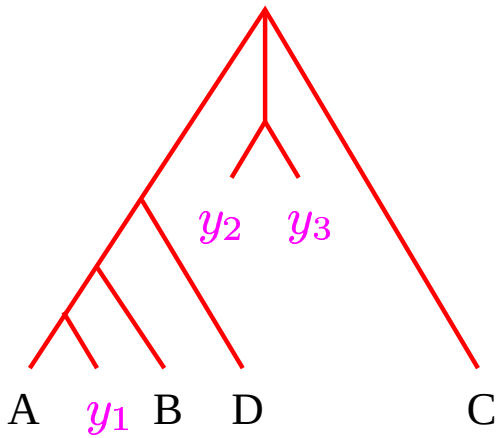
1 Determine specific leaves and *subtrees* of T_1, T_2 $O(n)$



T_1

A	x_1	B	x_2	x_3	C	x_4	D	y_1	y_2	y_3
2	1	2	1	1	2	1	2	1	1	1

occurences

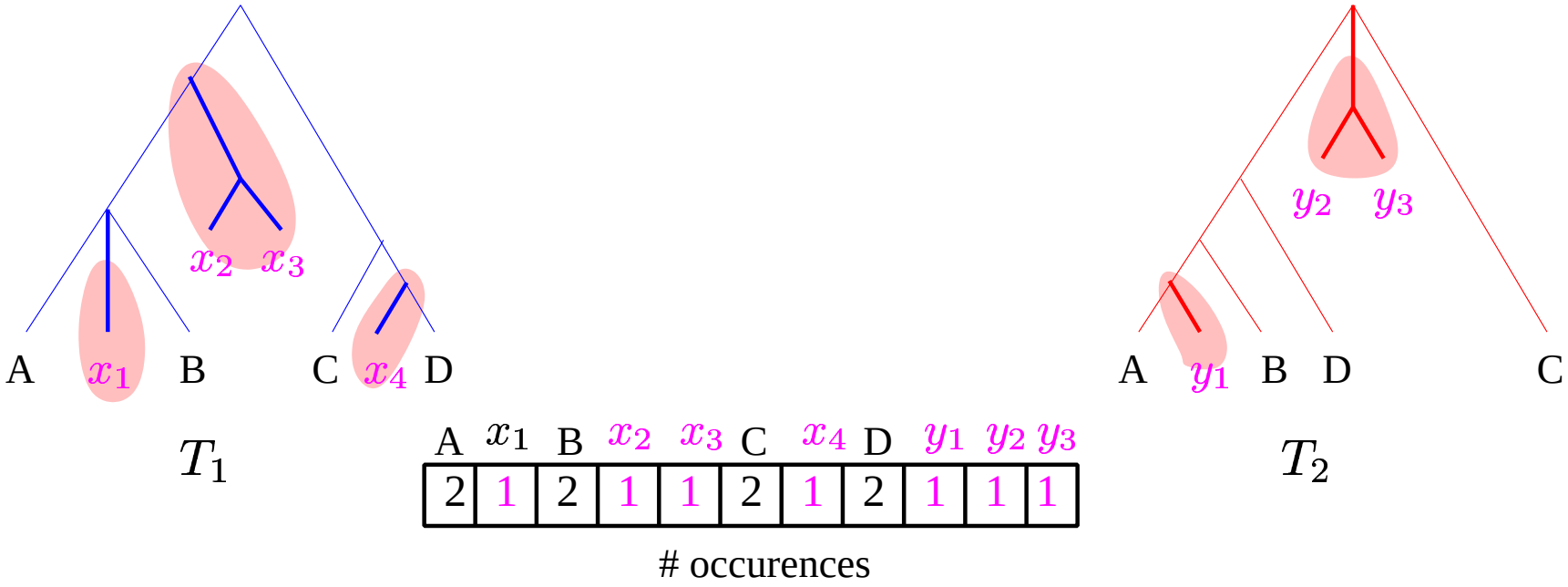


T_2

Computing $SMAST(T_1, T_2)$

Given T_1, T_2 two trees on overlapping sets of leaves.

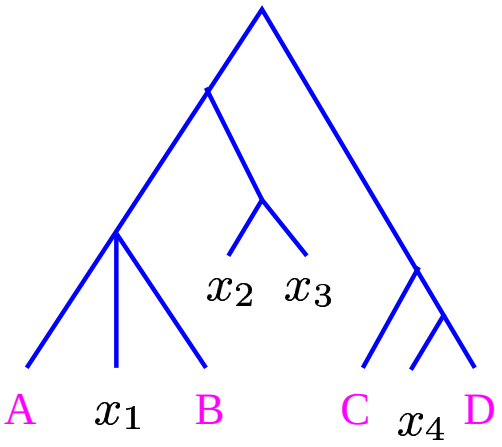
1 Determine specific leaves and *subtrees* of T_1, T_2 $O(n)$



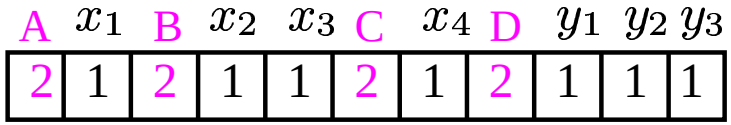
Computing $SMAST(T_1, T_2)$

Given T_1, T_2 two trees on overlapping sets of leaves.

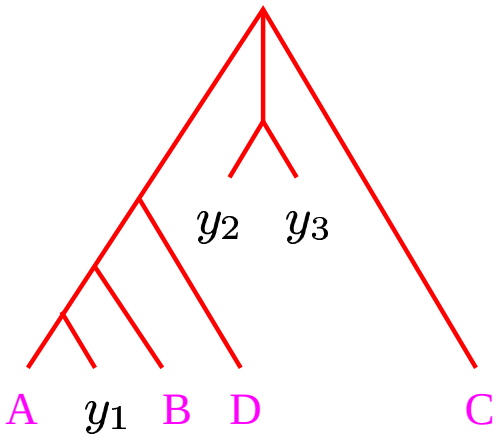
1. Determine specific leaves and *subtrees* of T_1, T_2 — $O(n)$
2. Compute $L_\cap = L(T_1) \cap L(T_2)$ and $T_1|L_\cap, T_2|L_\cap$ — $O(n)$



T_1



occurences



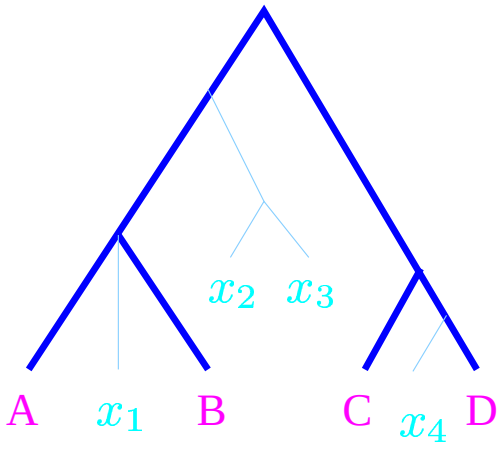
T_2

Computing $SMAST(T_1, T_2)$

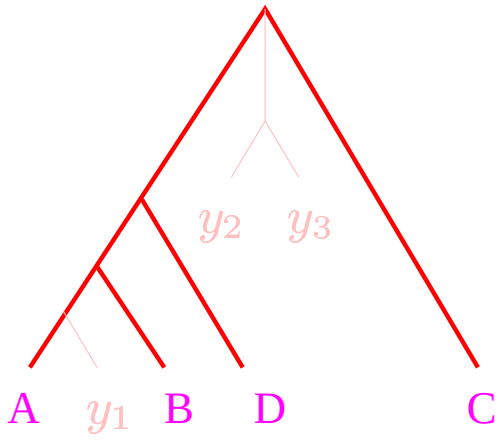
Given T_1, T_2 two trees on overlapping sets of leaves.

1 Determine specific leaves and *subtrees* of T_1, T_2 $O(n)$

2 Compute $L_\cap = L(T_1) \cap L(T_2)$ and $T_1|L_\cap, T_2|L_\cap$ $O(n)$



$T_1|L_\cap$



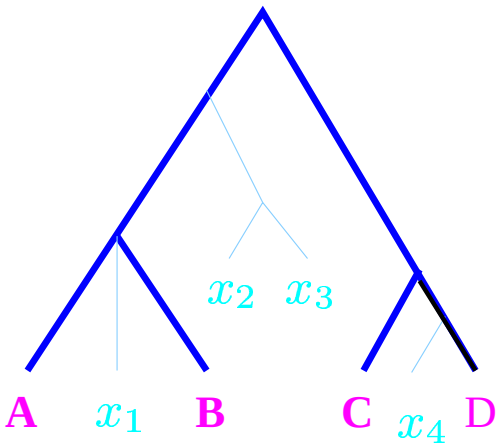
$T_2|L_\cap$

A	x_1	B	x_2	x_3	C	x_4	D	y_1	y_2	y_3
2	1	2	1	1	2	1	2	1	1	1

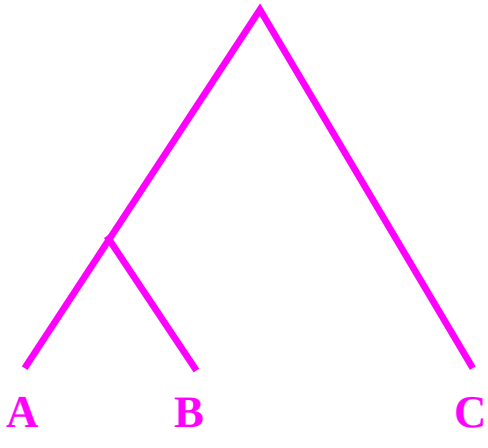
occurences

Computing $SMAST(T_1, T_2)$

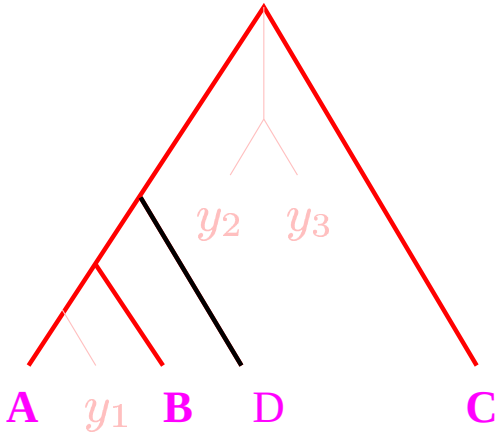
1	Determine specific leaves and *subtrees* of T_1, T_2	$O(n)$
2	Compute $L_\cap = L(T_1) \cap L(T_2)$ and $T_1\|L_\cap, T_2\|L_\cap$	$O(n)$
3	Compute $T_M := MAST(T_1\|L_\cap, T_2\|L_\cap)$	$O(N)$



$T_1|L_\cap$



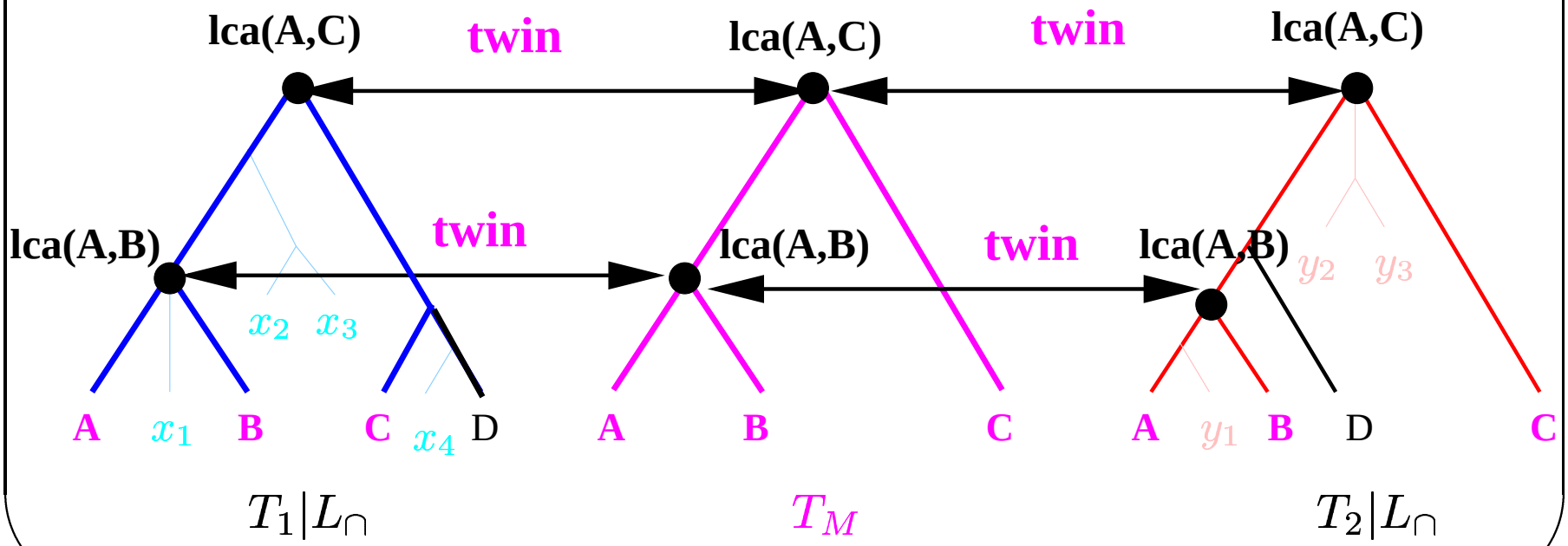
T_M



$T_2|L_\cap$

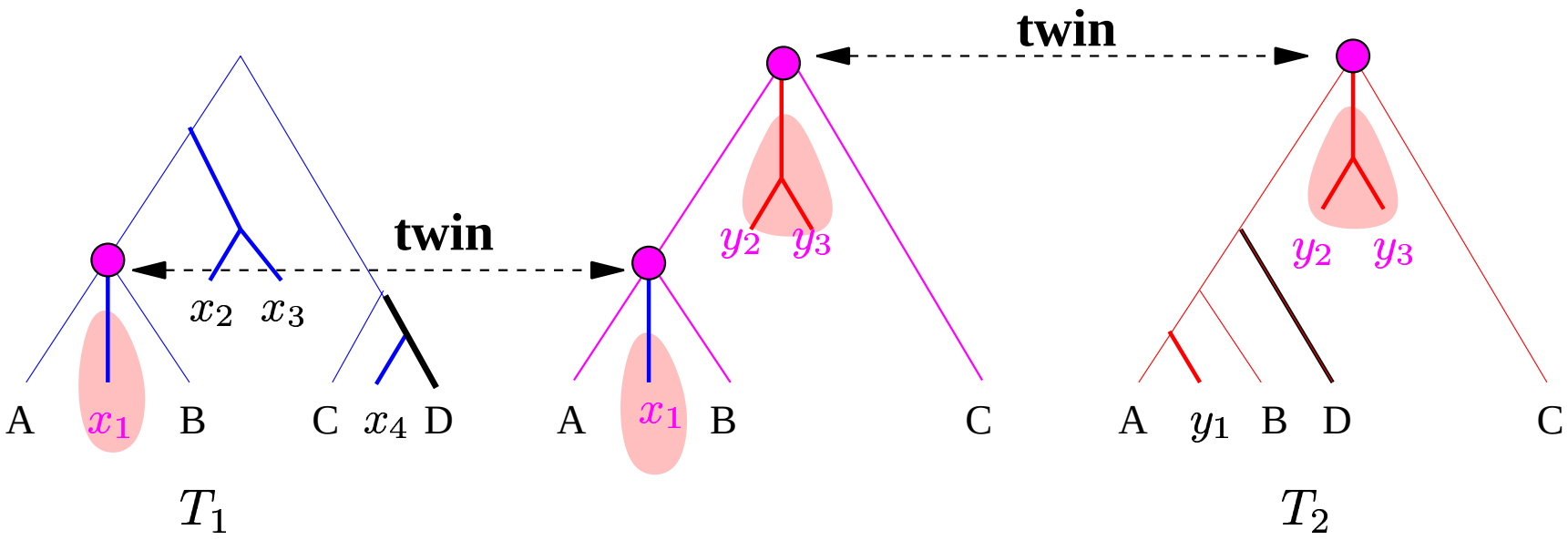
Computing $SMAST(T_1, T_2)$

2	Compute $L_\cap = L(T_1) \cap L(T_2)$ and $T_1\|L_\cap, T_2\|L_\cap$	$O(n)$
3	Compute $T_M := MAST(T_1\|L_\cap, T_2\|L_\cap)$	$O(N)$
4	Determine twin nodes between T_M and T_1, T_2	$O(n)$



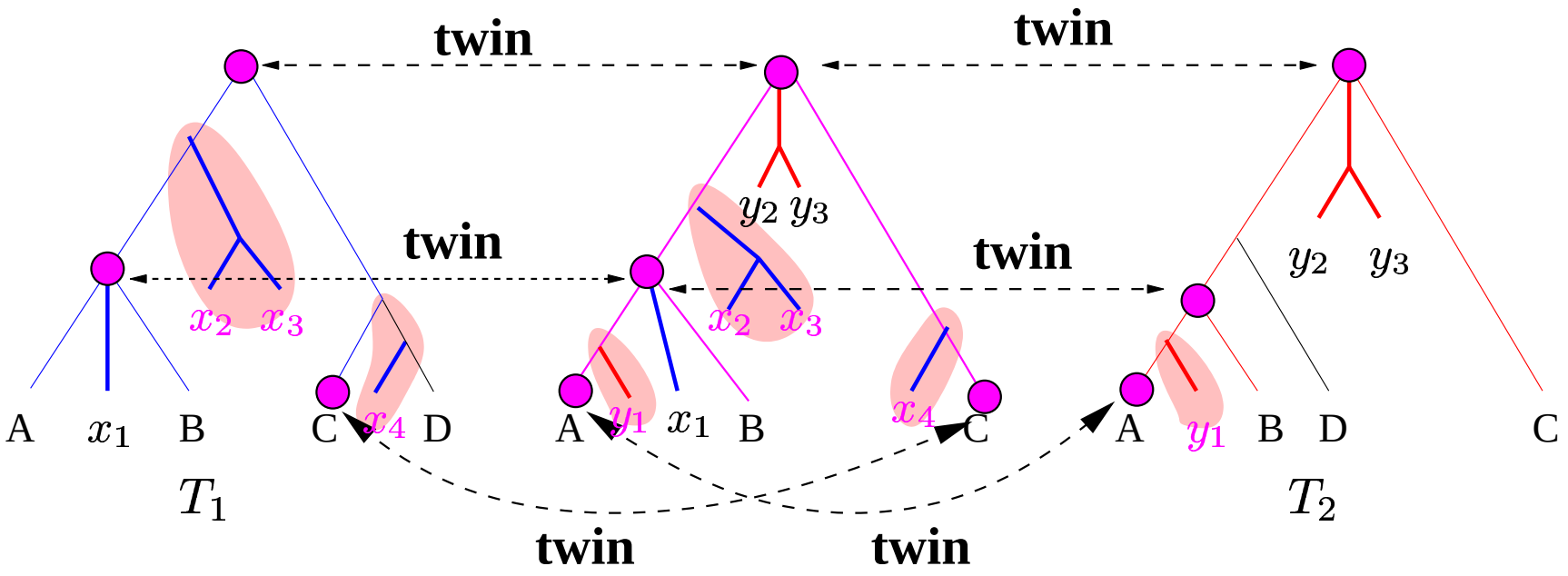
Computing $SMAST(T_1, T_2)$

3	Compute $T_M := MAST(T_1\|L_\cap, T_2\|L_\cap)$	$O(N)$
4	Determine twin nodes between T_M and T_1, T_2	$O(n)$
5	Graft *specific subtrees* at twin nodes of T_M	$O(n)$



Computing $SMAST(T_1, T_2)$

4	Determine twin nodes between T_M and T_1, T_2	$O(n)$
5	Graft *specific subtrees* at nodes of T_M	$O(n)$
6	Graft specific subtrees on edges of T_M	$O(n)$



Computing $SMAST(T_1, T_2)$

Given T_1, T_2 two trees on overlapping sets of leaves.

1	Determine specific leaves and *subtrees* of T_1, T_2	$O(n)$
2	Compute $L_\cap = L(T_1) \cap L(T_2)$ and $T_1\|L_\cap, T_2\|L_\cap$	$O(n)$
3	Compute $T_M := MAST(T_1\|L_\cap, T_2\|L_\cap)$	$O(N)$
4	Determine twin nodes between T_M and T_1, T_2	$O(n)$
5	Graft *specific subtrees* at nodes of T_M	$O(n)$
6	Graft specific subtrees on edges (u, v) of T_M	$O(n)$

Complexity in $O(N) = \Omega(n)$.

Results of the paper

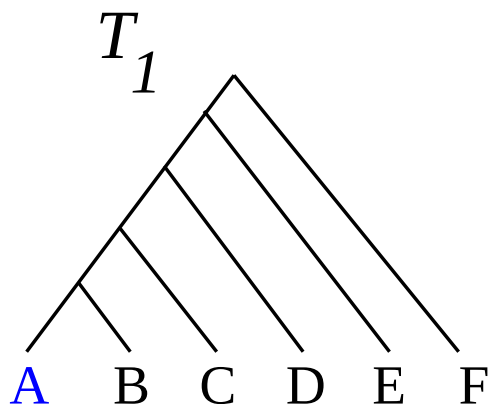
- An $O(\min(3^p nk, 2.27^p + kn^3))$ FPT algorithm for MAST and MCT.

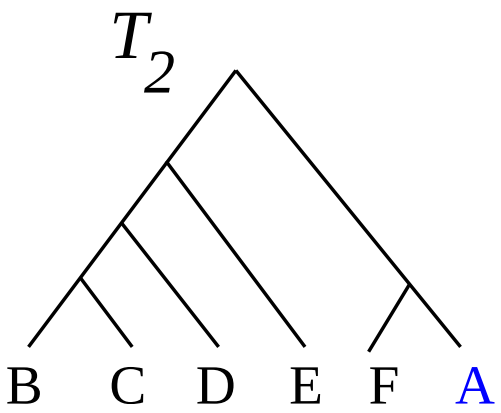
$\Rightarrow$ Fixing $p = 0$ gives a new simple linear time algorithm for solving the *perfect phylogeny* problem and the *character compatibility* problem.

- A 3-approximation algorithm running in $O(kn^3)$ for the complement of the MAST and MCT problems.
- An $O(N)$ algorithm for SMAST and SMCT on 2 trees, where N is the time required to solve MAST, MCT on 2 trees ($N = \Omega(n)$).
- Proofs that SMAST and SMCT are NP-hard, $W[2]$-hard for parameter p, and not approximable within a constant ratio (unless P=NP).

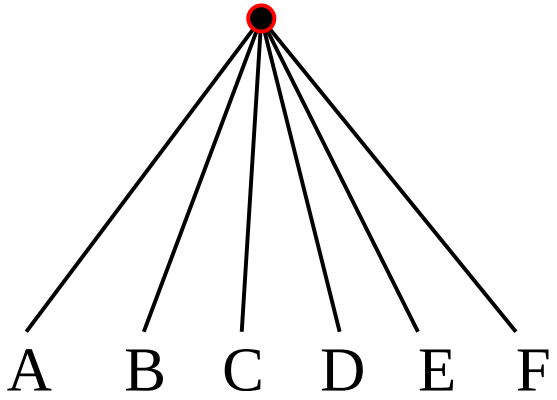
Thanks to anonymous referees and to J.Jansson for sending a paper appearing $\approx$ the same time at *Latin'04* with two results in common with us.

The maximum agreement subtree is often more informative than the strict consensus:





Strict consensus



Maximum Agreement SubTree