Phylogenetic reconstruction based on **multiple genes**

Many studies in the last several years have inferred trees based on 100-500 genes.

Phylogenies of several thousands species have been reconstructed.

Methodologies for building trees from such large combined data fall into two broad categories:

- Supermatrix (superalignement) approaches that concatenate aligned sequences and create one big alignment

- Supertree approaches that algorithmically combine trees constructed from several individual alignments.

Intrinsic tradeoffs that limit the density of the data assembled in large datasets

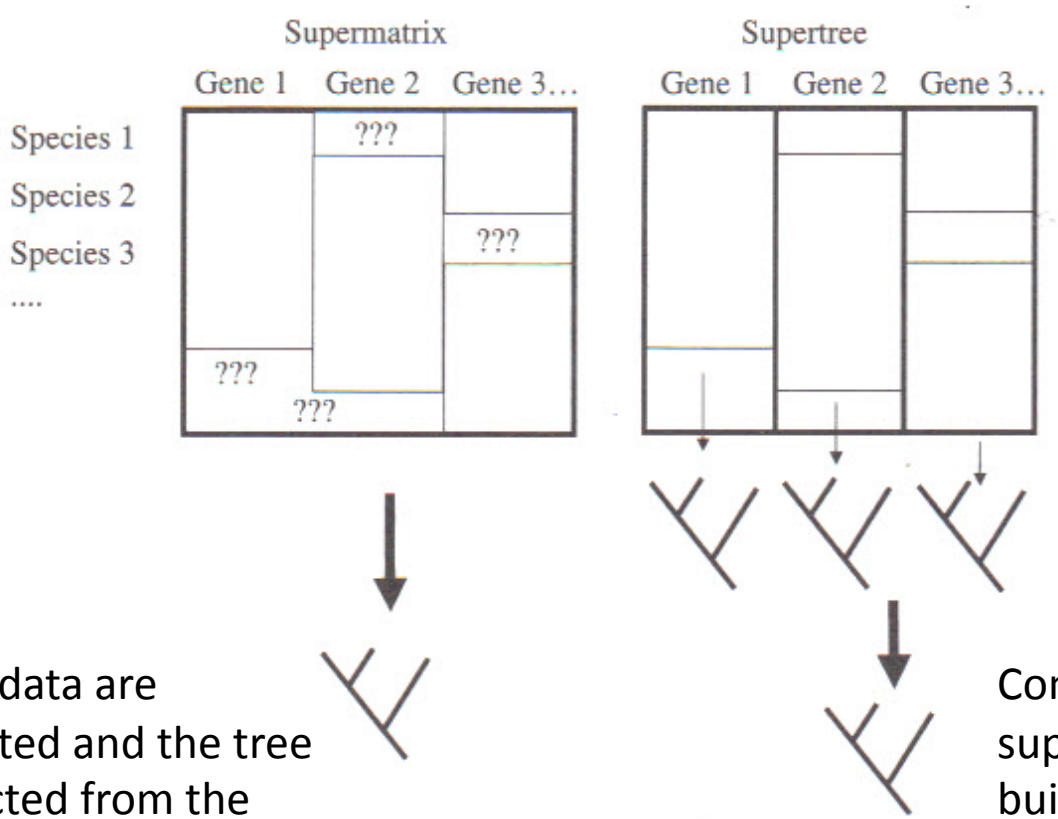
Incompleteness of the information (density) available for each taxon relative to each locus (in a supermatrix analysis) or tree (for a supertree analysis) concerns:

- gene loss
- sequence divergence that obscures homology
- sampling biases
- ….

and it makes construction of large datasets difficult.

How to deal with data fragmentation and copying with it in tree inference?

Two main strategies to reconstruct phylogenomic-scale data sets



Sequence data are concatenated and the tree is constructed from the combined data

Construction of supertrees by first building trees from each locus and then combining the trees.

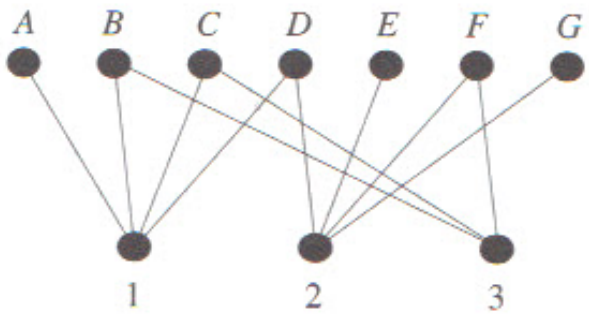
Different ways to **represent** fragmentation in phylogenetic data

Loci/genes

Taxa/species

	1	2	3
A	ACGTT	?????	?????
B	ACGTT	?????	TCTCC
C	ACCGG	?????	TCGTC
D	ACACG	TAATA	?????
E	?????	TT-TT	?????

	123
A	100
B	101
C	101
D	110
E	010



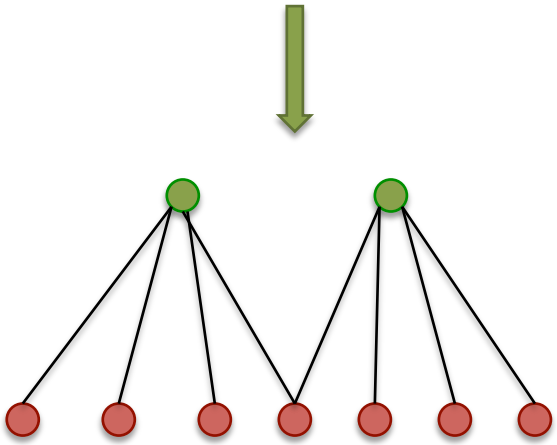
Graphical representation:
the density of the graph represents the density of the matrix

Example: subsets of data share information

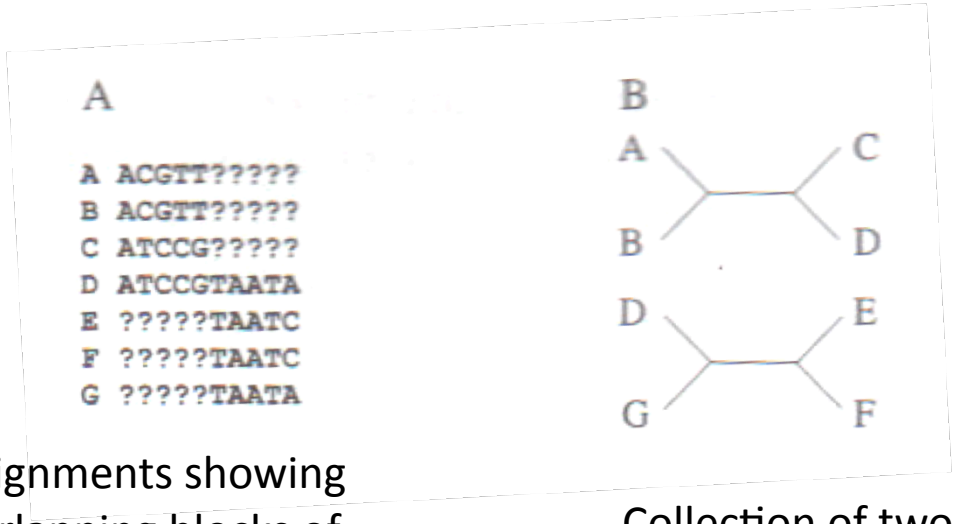
A

```
A ACGTT?????
B ACGTT?????
C ATCCG?????
D ATCCGTAATA
E ?????TAATC
F ?????TAATC
G ?????TAATA
```

Sequence alignments showing partially overlapping blocks of sequences.

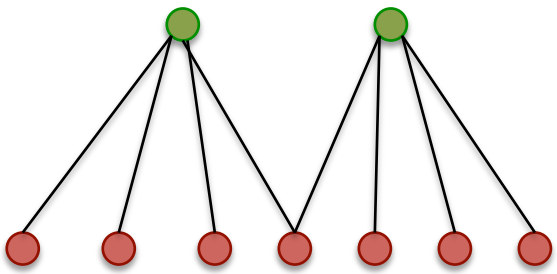


Example: subsets of data share information



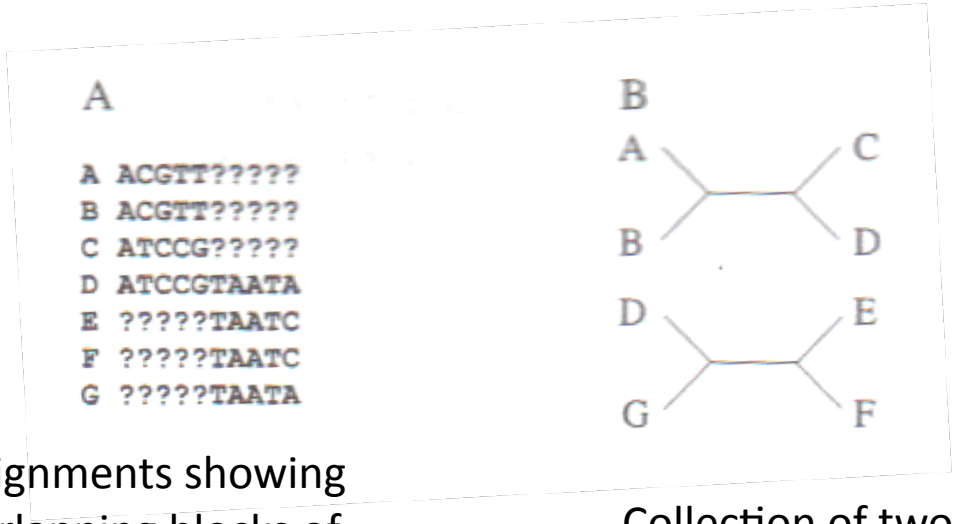
Sequence alignments showing partially overlapping blocks of sequences.

Collection of two unrooted input trees illustrating the same pattern of taxon overlap that occurs in the alignment.



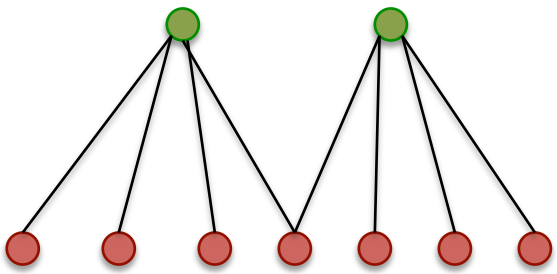
Splitting the information into ``**redundant**'' subsets

Example: subsets of data share information



Sequence alignments showing partially overlapping blocks of sequences.

Collection of two unrooted input trees illustrating the same pattern of taxon overlap that occurs in the alignment.



Splitting the information into ``**redundant**'' subsets

Both representations are based on parsimony

The main question:

Is it better **to break the data** into subsets to be analyzed separately or to handle the effect of fragmentation after the analysis, by some methods of sorting, through the output trees?

Notice that here the question arises simply by virtue of the occurrence of missing data. Nothing more profound than this.

In certain phylogenetic reconstructions we might arrive to have 90% of missing data!

There is no elegant treatment of the problem.

There are three classes of strategies for handling data fragmentation that seems to arise commonly in large-scale phylogenetic analysis.

1: post-processing strategies:

- ignore the fragmentation until after phylogenetic analysis is performed
- process the resulting trees to tease apart the underlying signals

2 and 3: pre-processing strategies:

- break up data into pieces
- do separate phylogenetic analysis.

An approach constructs the 'ideal' separation set and the other applies clustering methods or optimality criteria (heuristics) to obtain 'good' data sets.

A **data availability matrix** A is a matrix of N rows (labelled by taxon names) and M columns (labelled variously: by character names, names corresponding to sets of characters, or by tree names).

Each cell of the matrix is scored 1 if data are available for the entry, 0 otherwise.

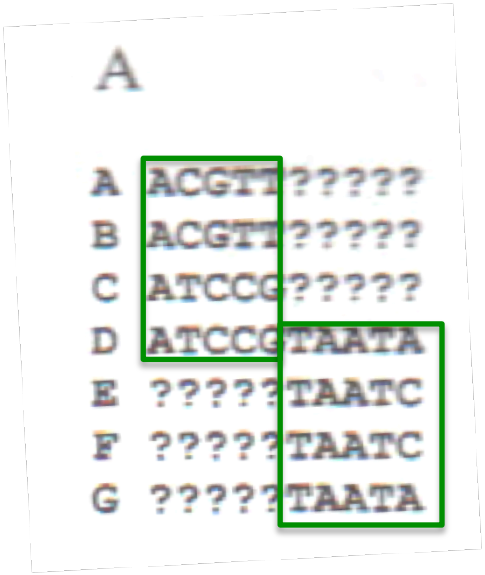
m(A) the number of entries in A containing 1
m(A)/NM the density of A

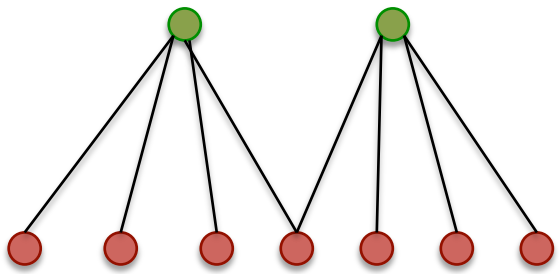
A block in A is a submatrix defined by a subset of A's rows and columns entirely filled by 1

Two columns are non-overlapping if no row is present that has a 1 in both columns.

Two blocks in a phylogenetic data matrix are non-overlapping if every pair of columns, one from each block, is non overlapping.

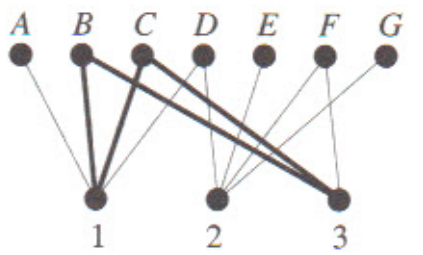
Overlapping blocks



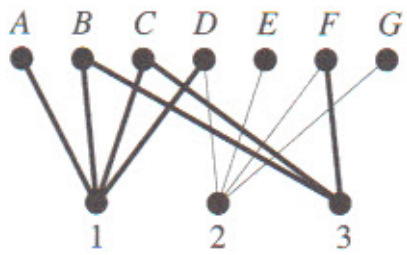


A dataset is fragmented if there are non-overlapping blocks in A or equivalently if A is disconnected. It might be of interest to consider a more relaxed notion of fragmentation, occurring when A can become disconnected by the removal of only a ‘few’ edges.

For instance:



Maximal bi-clique on nodes B C and 1 3



Quasi-bi-clique extension of the **maximal** bi-clique by **adding** all taxa that are connected to **at least 50%** of the loci. This corresponds to a data-availability matrix that has no more than 50% of missing data.

The idea is to split the problem in subproblems from which to reconstruct separated trees.

Strategy 1: postprocessing collection of trees

Several methods to reach consensus were developed. Many have been developped to specifically treat problematic taxon subsets of the data that appear to be unstable and that can arise from many reasons including:

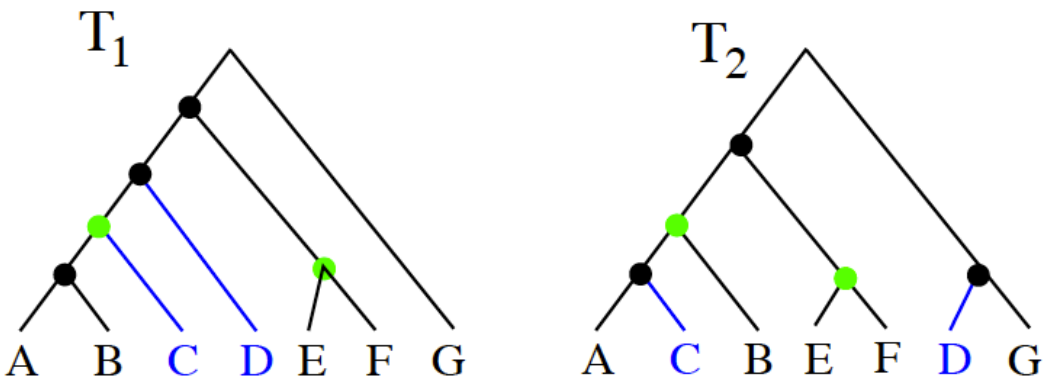
long-branch attraction
missing data
heterogeneous histories
hybridization
…

A useful technique is identifying maximum agreement subtrees (MAST), which are the largest subtrees **common** to an entire collection of input trees.

A more relaxed technique identifies maximum compatible trees (MCT), which are smaller trees that are **compatible** with all input trees **once taxa are removed**.

Consensus of evolutionary trees

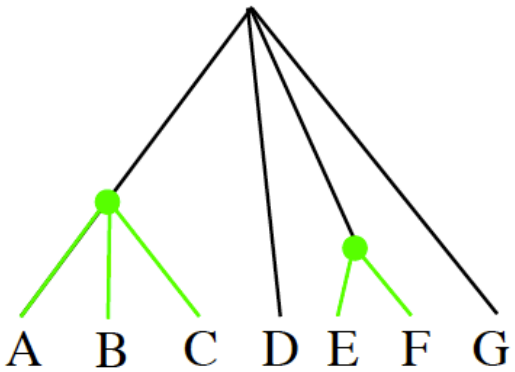
The source trees usually *conflict* on the position of some leaves.



Approaches not contradicting any input information:

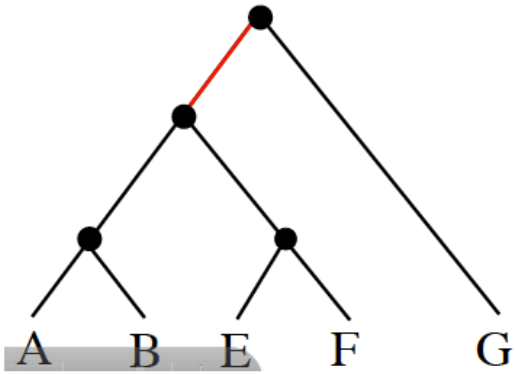
Strict consensus

removing input clades

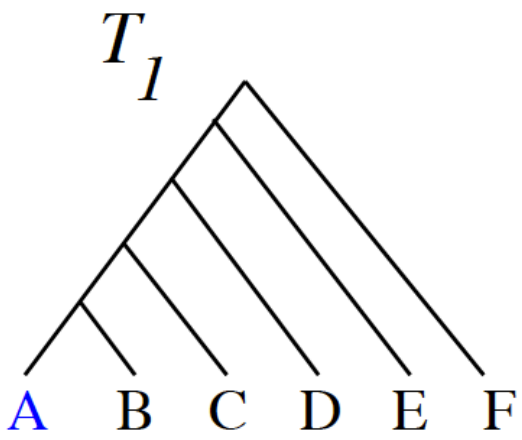


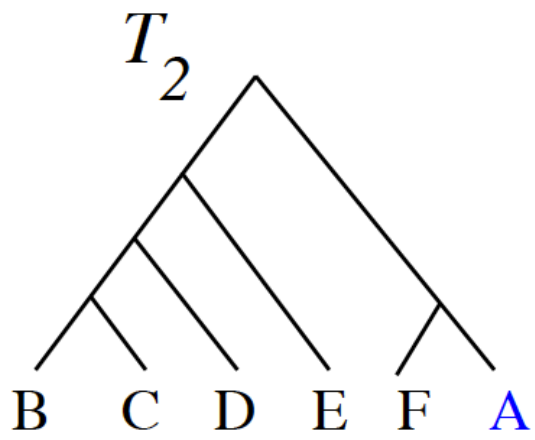
Maximum Agreement SubTree

removing input leaves

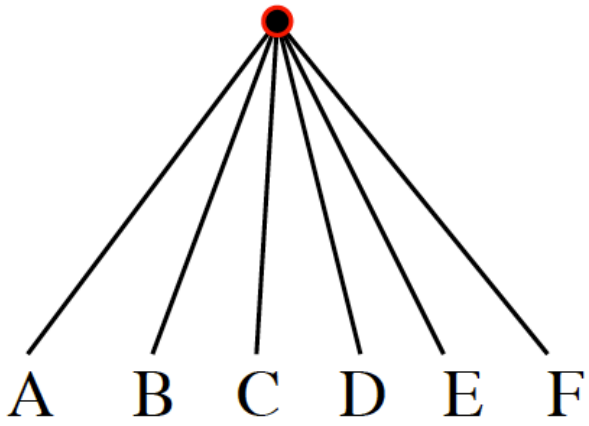


The maximum agreement subtree is often more informative than the strict consensus:

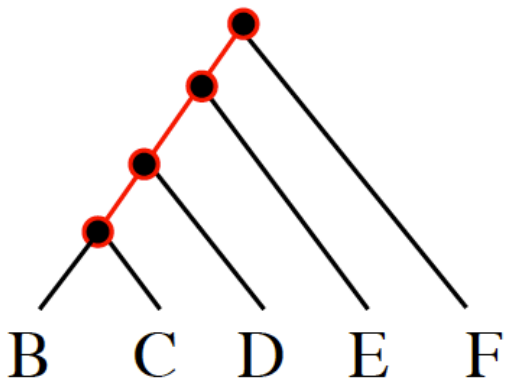




Strict consensus



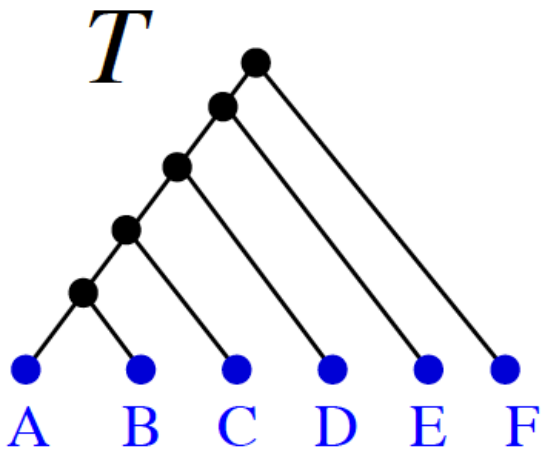
Maximum Agreement SubTree



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.



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

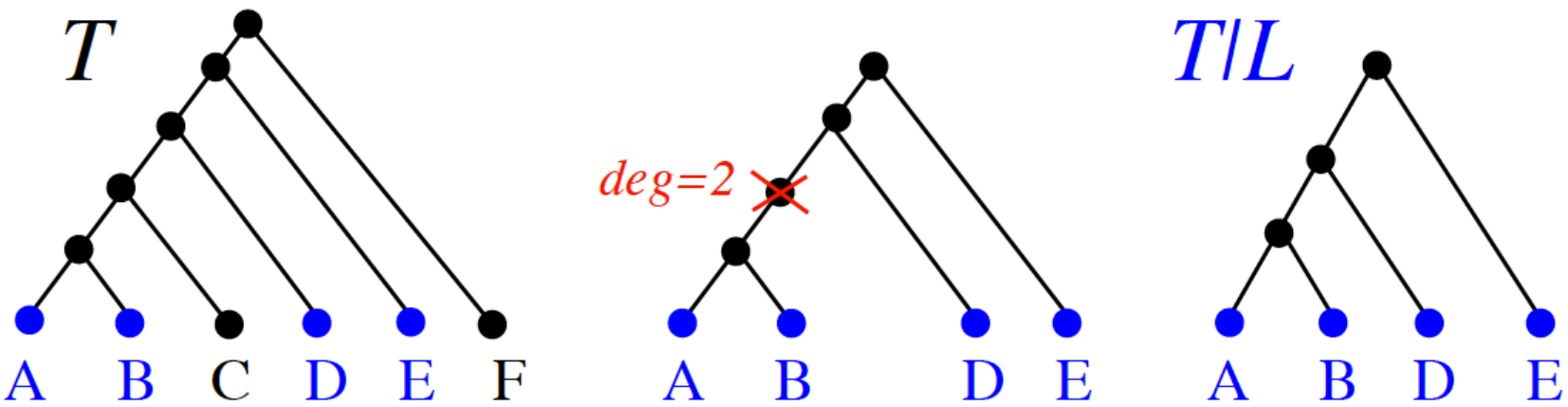
$\#T = 6$

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

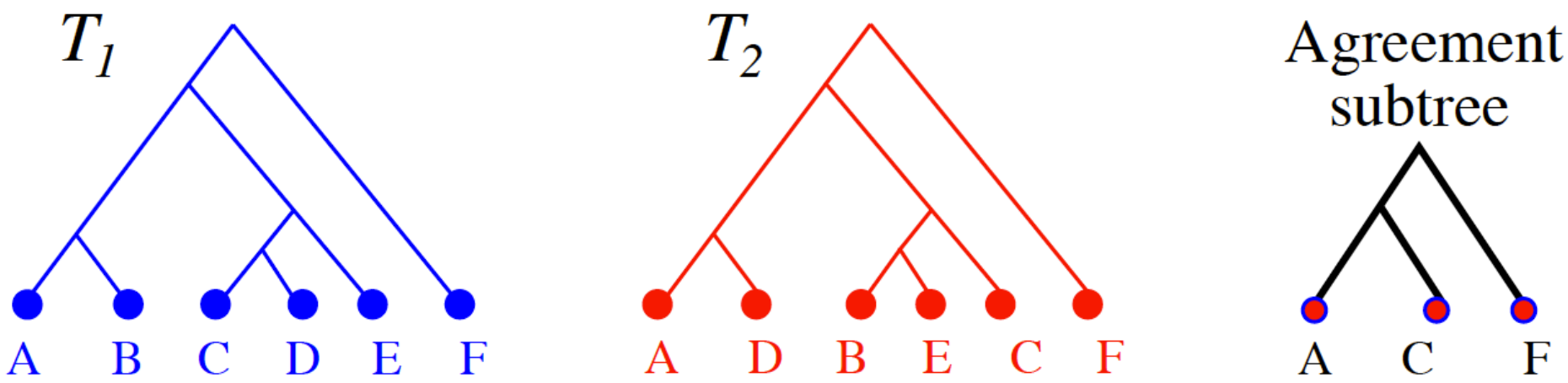


Maximum Agreement Subtree - MAST

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

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

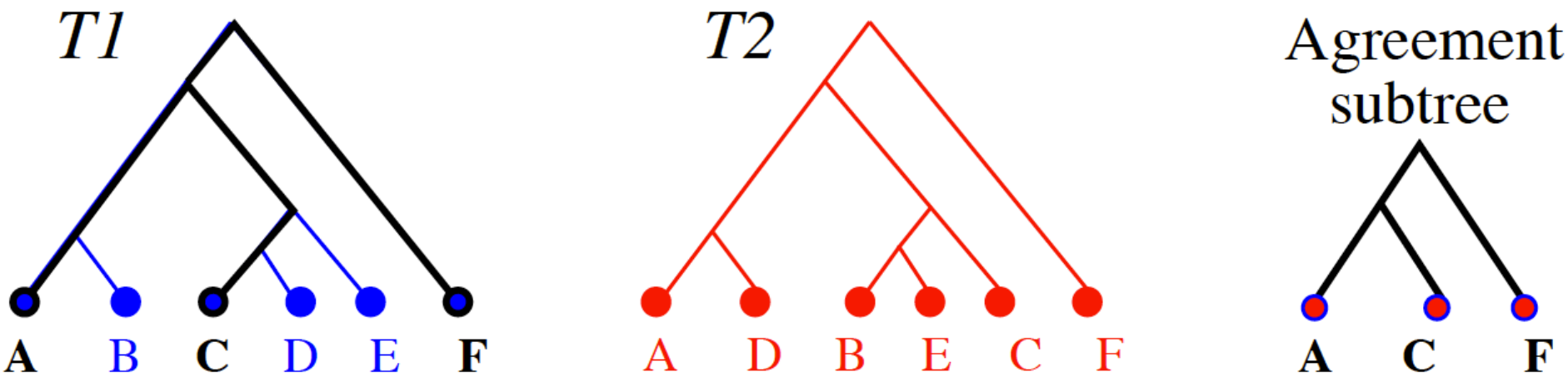


Maximum Agreement Subtree - MAST

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

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

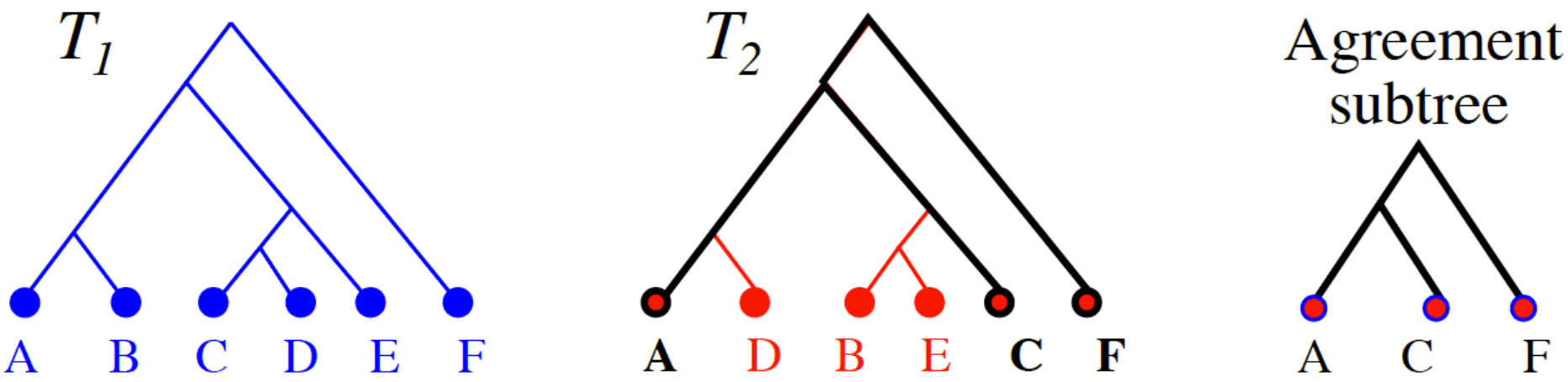


Maximum Agreement Subtree - MAST

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

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

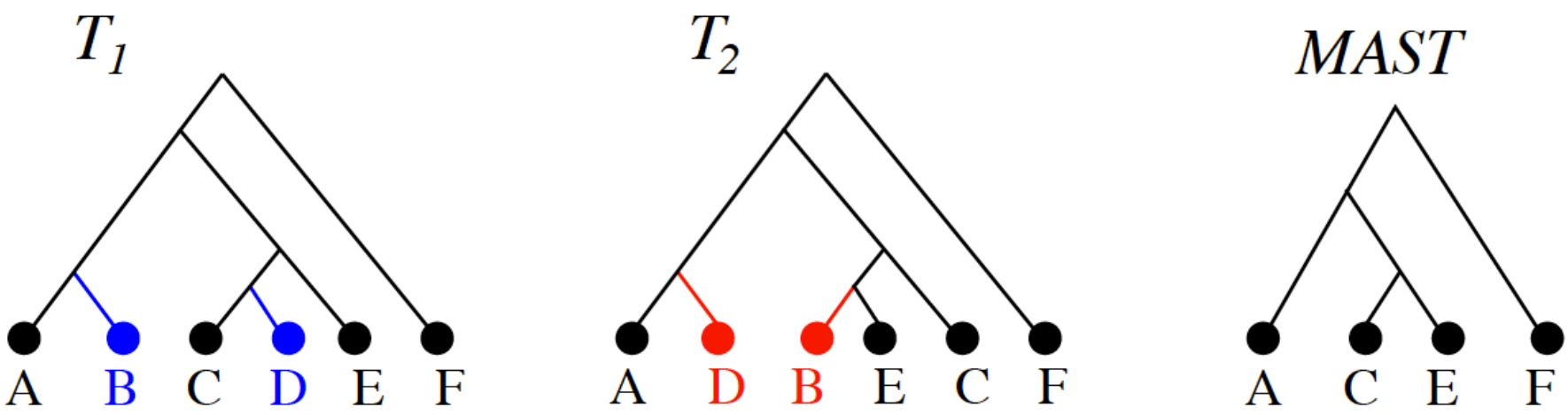


Maximum Agreement Subtree - MAST

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

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

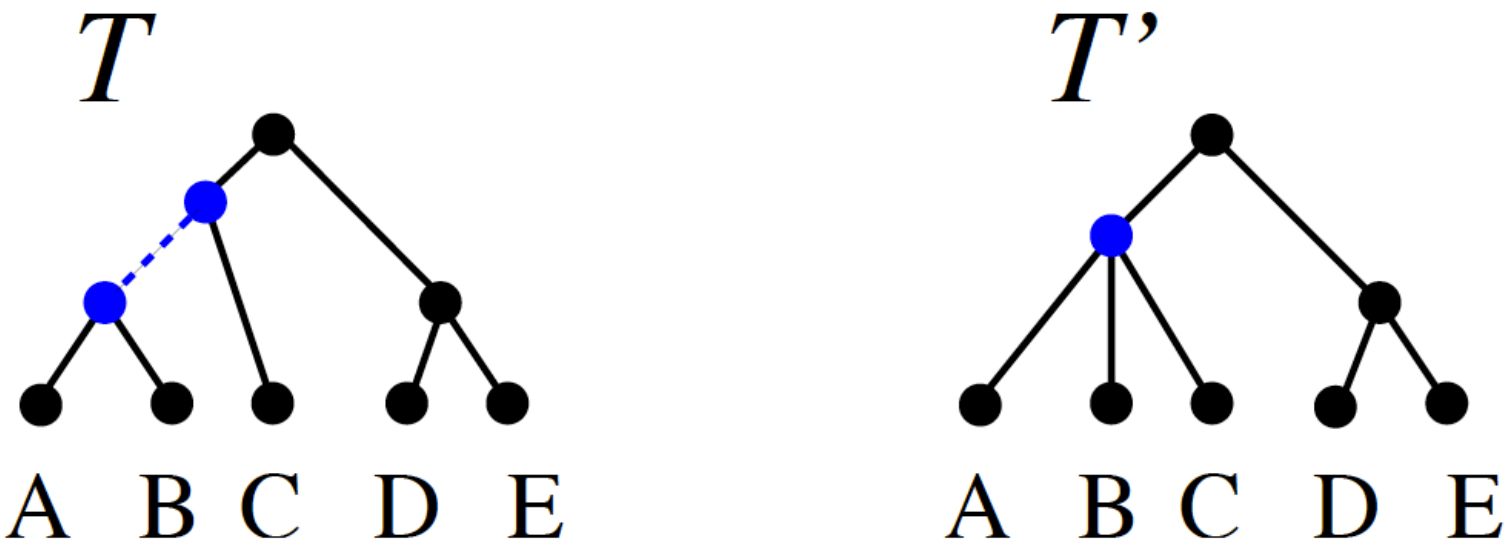
T is a Maximum Agreement SubTree (MAST) of $\mathcal{T}$ iff it is of maximum size among all agreement subtrees of $\mathcal{T}$.



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:

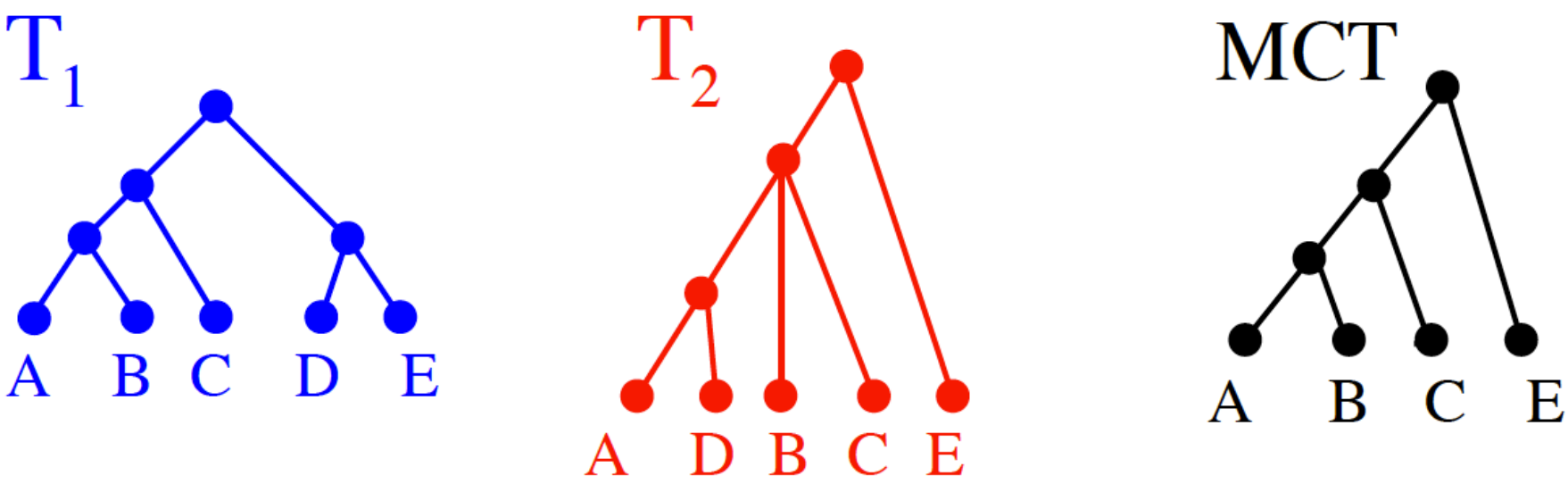


Maximum Compatible Tree - MCT

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 $\mathcal{T}$.

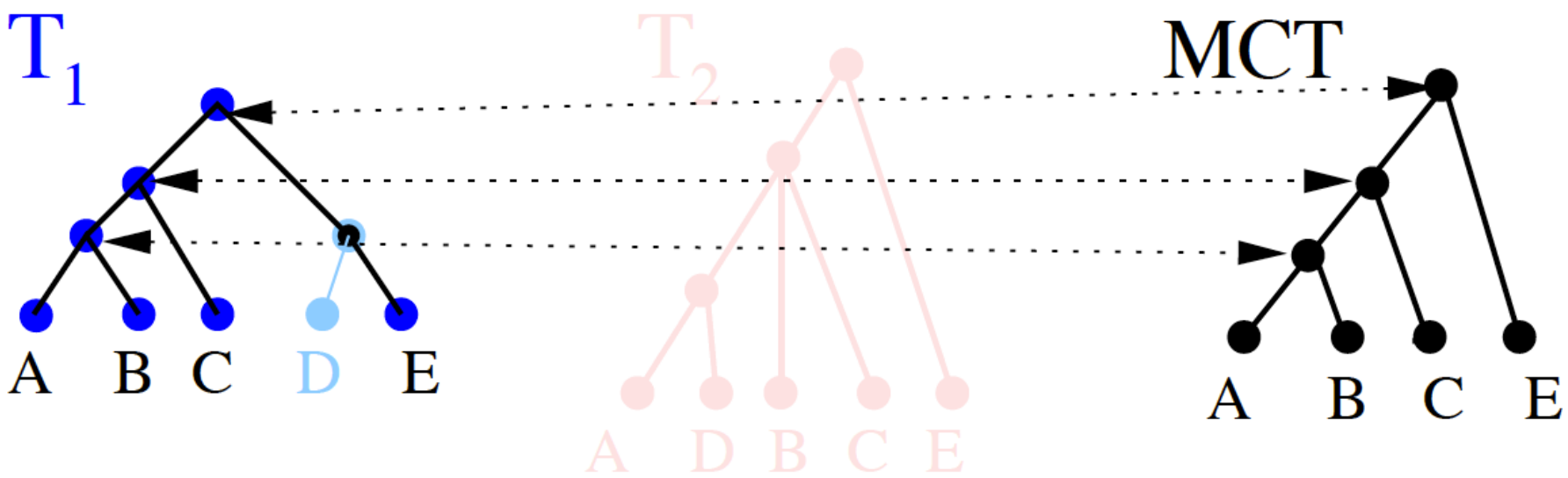


Maximum Compatible Tree - MCT

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 $\mathcal{T}$.

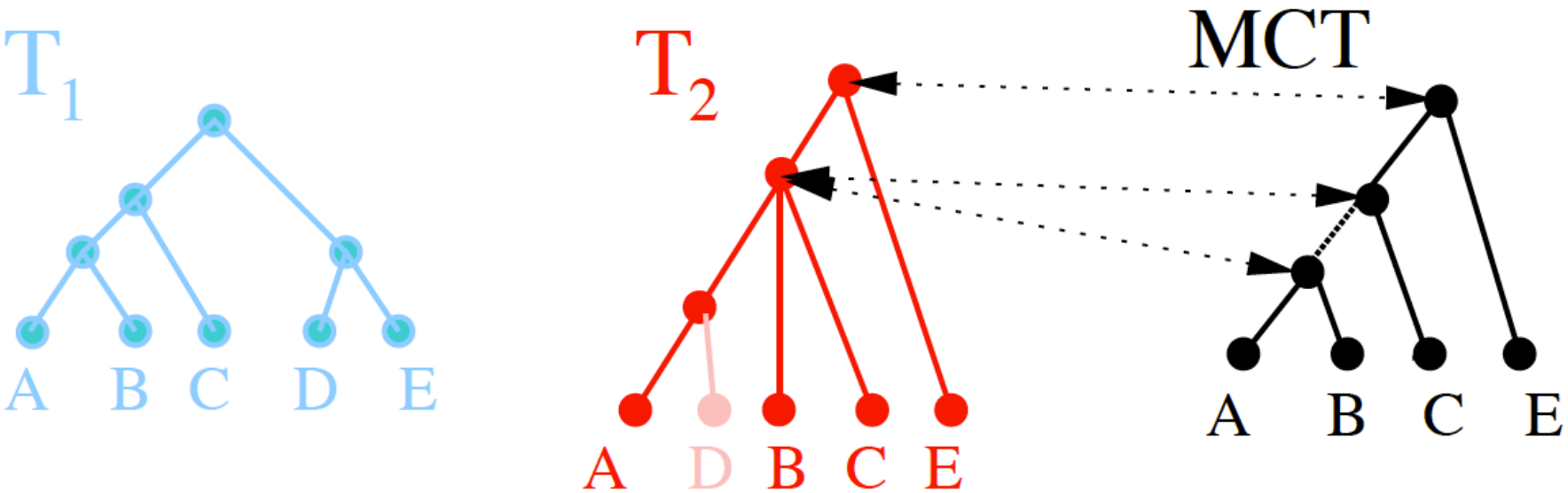


Maximum Compatible Tree - MCT

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

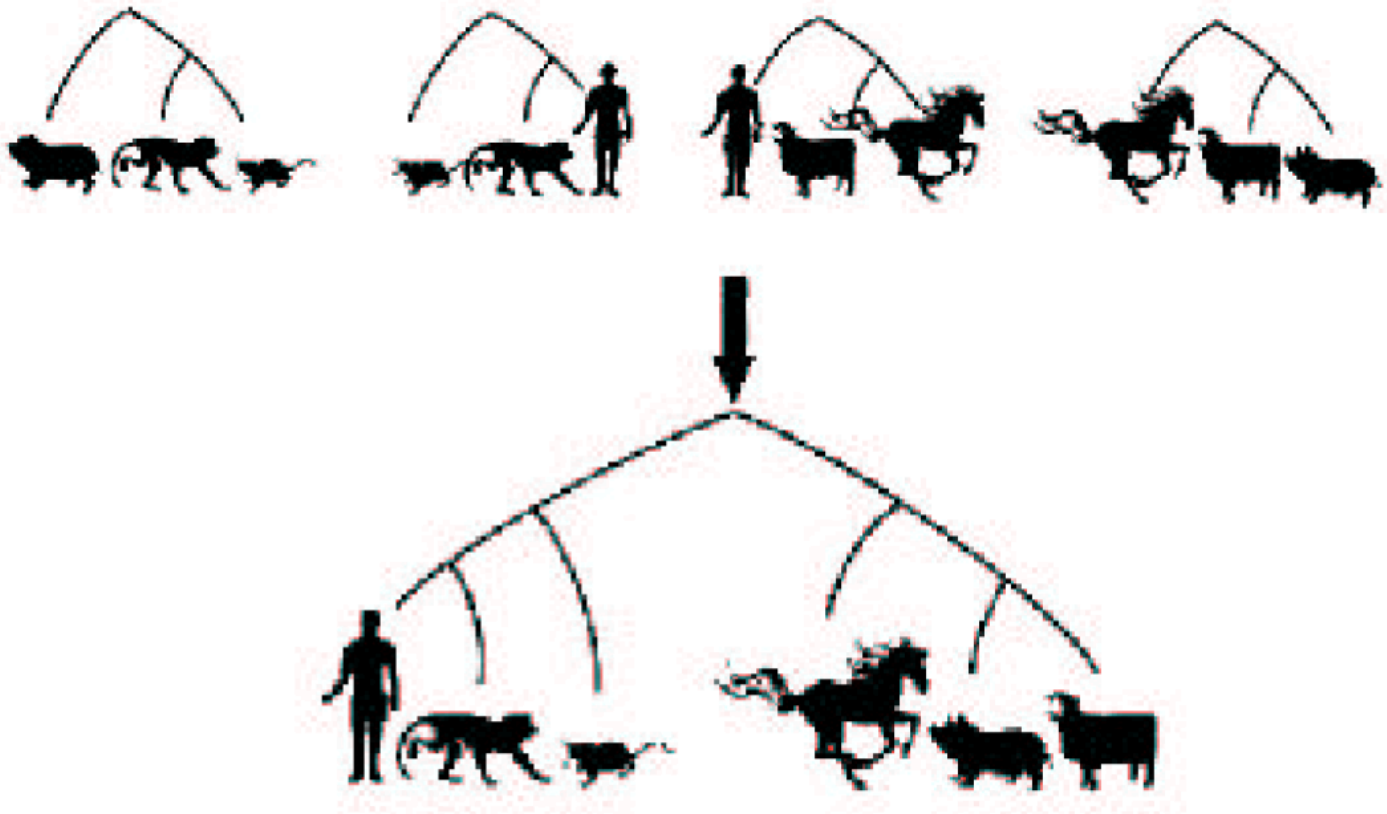
  p is the number of input leaves to remove to obtain agreement, resp. compatibility.

(FPT: fixed parameter tractable problems)

<u>In conclusion</u>: Finding the MAST/MCT is a NP-hard problem when the number of input trees is >2 and the degree of the nodes is not bounded.

It is solvable in polynomial time if one of the input trees has a degree bound, but the time is exponential in the bound.

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

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

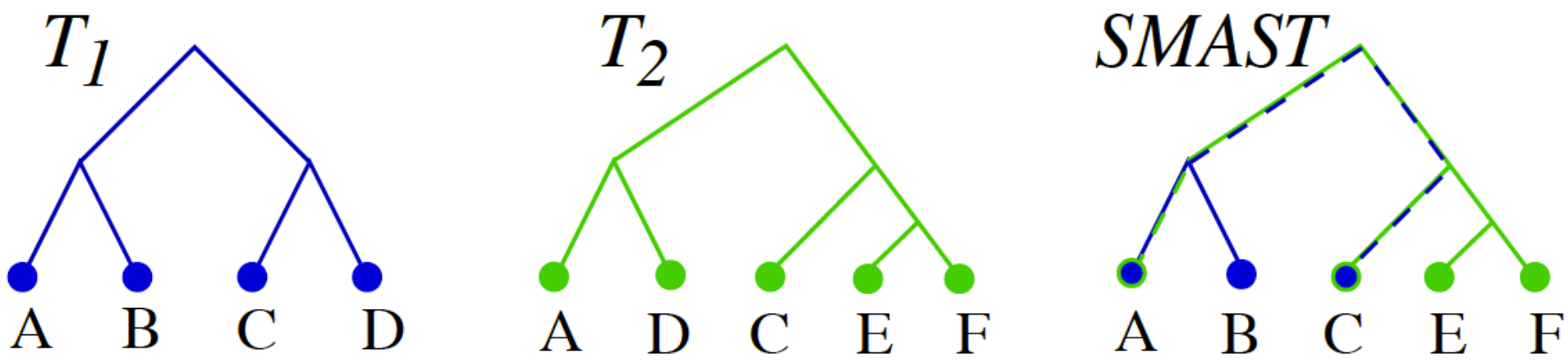
T is a maximum agreement supertree ($SMAST$) iff it is maximum in size among agreement supertrees of $\mathcal{T}$.

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 $\mathcal{T}$.

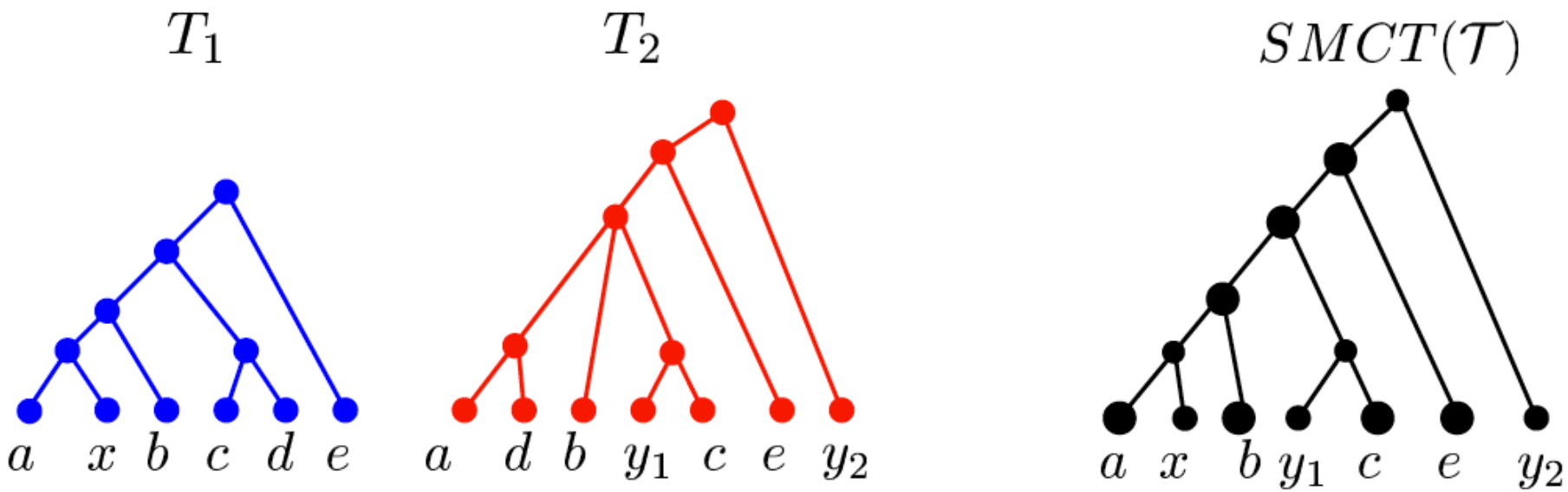


Maximum compatible supertree - SMCT

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 $\mathcal{T}$.



Results of the paper

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

- 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).

Nicolas F., Guillemot S., Berry V., Paul C.
On the Approximability Results for Maximum Agreement Subtree and Maximum Compatible Tree Problems
Discrete Applied Mathematics 157, 7 (2009) 1555-1570

Strategy 2: preprocessing by grove identification

The idea is to break the data into separate pieces prior to undertaking phylogenetic analysis. The partitioning strategy is appealing when nothing is lost or, better, something is gained.

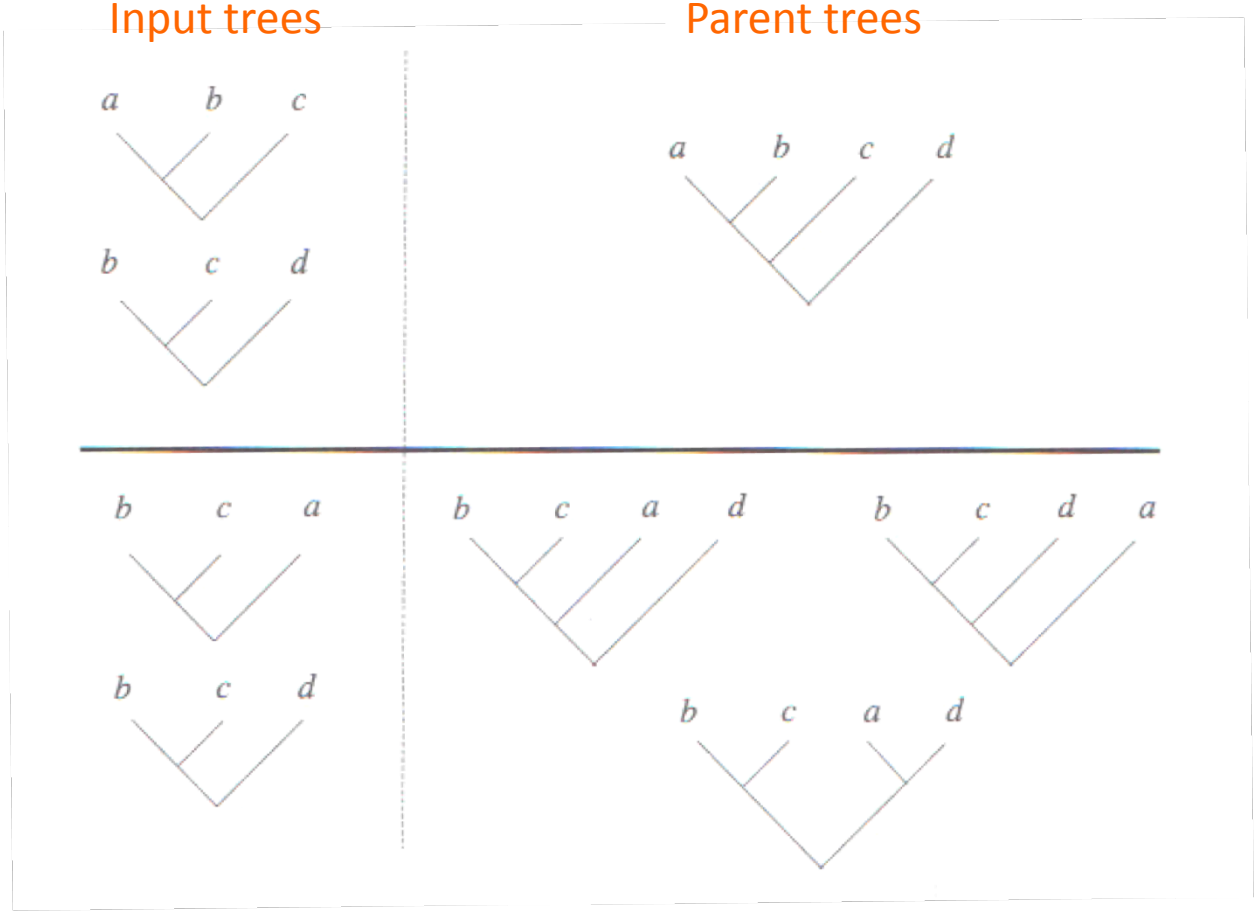
What does it mean **to gain 'new information'**?

This idea has been difficult to formalize. Assume, for a moment, to work with trees.

Definitions: Any set of three taxa {x,y,z} is a triple. A rooted binary tree on these taxa is a triplet, denoted xy|z if xy share a more recent common ancestor than either does with z.

Take trees that are compatible with each other such that there does exist one or more parent trees. We consider three-taxon trees.

Input trees | Parent trees | New information



ab|d and ac|d

{a b d}:
ab|d ad|b bd|a

{a c d}:
ac|d ad|c cd|a

When one cross-triplet is displayed by all parent trees we call it resolved cross-triplet.

Input trees precision is usually unknown : they are “estimates” containing errors and, in practice, rarely compatible.

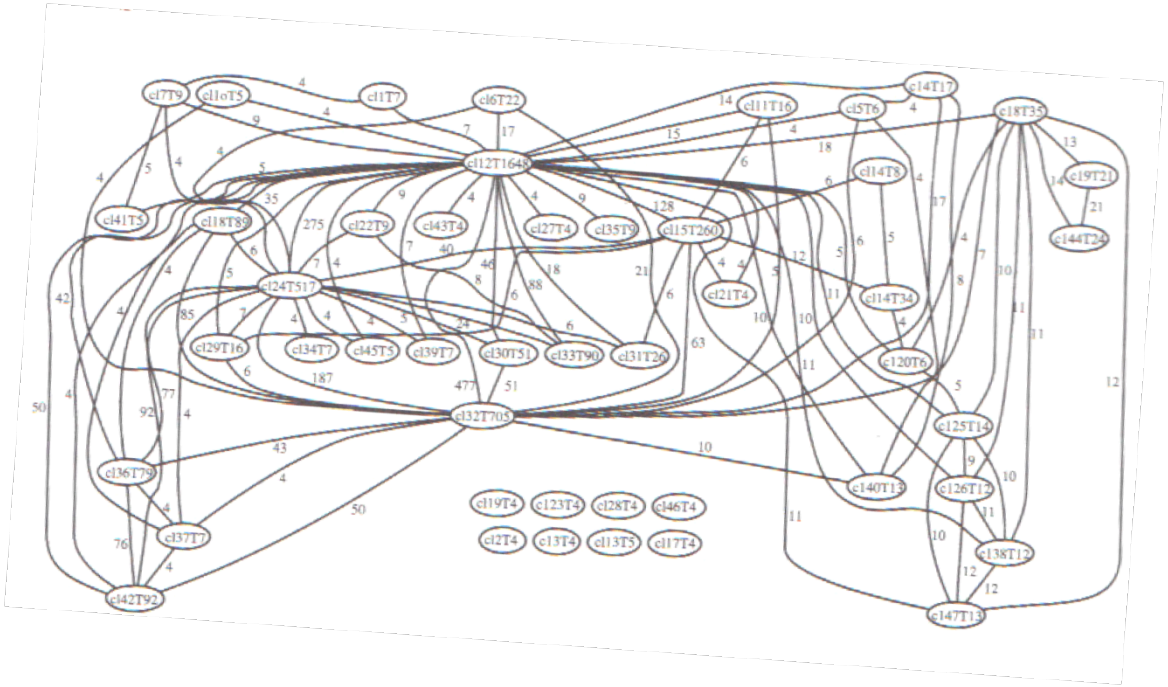
We can work with the **matrix A** without considering the topology of the trees.
Extract new information if possible and start to infer the topology afterwards.
The topologies should preserve new information.

A induces to consider a much larger set of possible input trees with taxonomic structure defined by A. The goal is to find a set of triplets such that their parent trees agree with each other and ask whether any of the triplets of the parent trees are resolved cross-triplets.

Several propositions on the statistical relations between columns of a matrix (approximating trees) have been made.

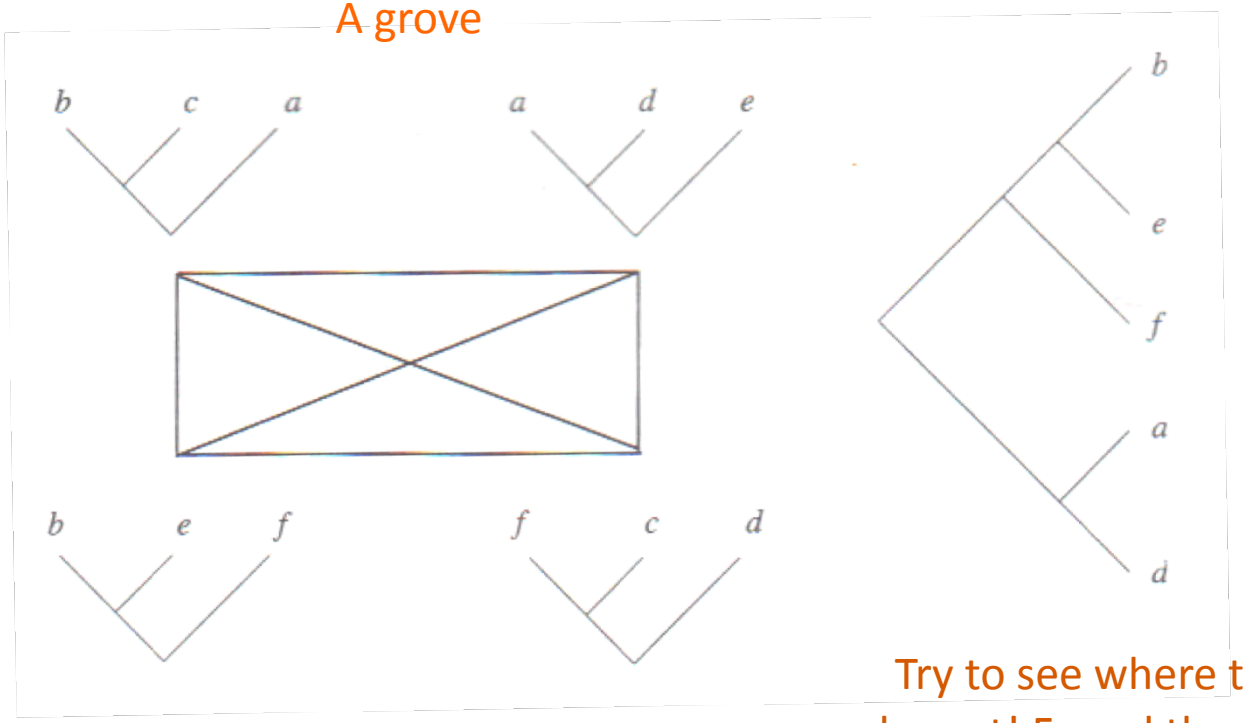
We shall not go too far with that but an important concept that has been introduced is the one of a **grove**, that is **a collection of trees (columns in A) that are mutually informative.**

The basic idea is that a collection of columns in A is a **grove** if every partition of this collection entails some new information when combined together.



Example of a grove of 47 loci.

A grove



Try to see where the **c** could be put! Found the ambiguity?

The ensemble of four trees, **only overlapping in one taxon**, is a grove.
Left: there are four input trees along with their overlapping graph.
Right: the tree is the maximum agreement subtree of five binary parent trees that each display all input trees. The five parent trees can be obtained by attaching taxon c to any of the five branches that are more closely related to b than to a.

Strategy 3: preprocessing by clustering or optimization strategies

More relaxed conditions for data subdivision were explored.
Until today, all has been done based on *ad hoc* strategies working on specific examples.

The belief is that missing data ultimately degrades the quality of phylogenetic analysis
But several empirical studies suggest that **substantial missing data can be tolerated**.

Fragmentation of large data sets in phylogenetic analysis
Sanderson, Ané et al.
In *Reconstructing Evolution – New mathematical and computational advances*
Edited by Olivier Gascuel and Mike Steel
Oxford University Press, 2007