



# Supertrees

(an introduction)

David Bryant  
McGill Centre for Bioinformatics  
Montréal, Québec

McGill Centre for Bioinformatics

McGill University · Montreal · Quebec · Canada

*Edward N. Adams III (1972):*

A new problem in the science of classification is presented, along with its solution.

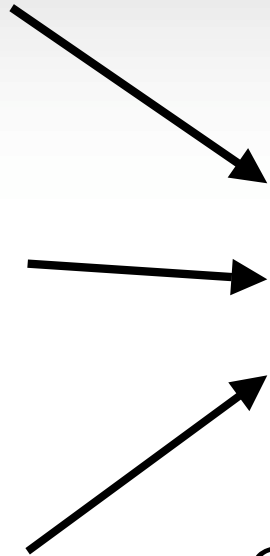
The problem is to combine the information in several taxonomic trees into a single tree. The solution is a computational method for computing a tree which represents only that information shared by the rival trees.

Such a method is called the *consensus* method.

# Consensus Trees



Input trees



Consensus method



Consensus tree

*Allan Gordon (1986)*

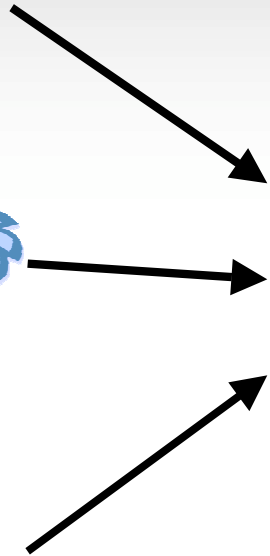
Consensus Supertrees: the synthesis of rooted trees containing overlapping sets of labelled leaves.

A *supertree* is a dendrogram [rooted tree] from which each of the original trees can be regarded as samples.

# Supertrees



Input trees



Supertree method



Supertree

# An introduction to Supertrees

1. What is a supertree method?
2. Why supertrees?
3. A taste of supertree mathematics
4. A tour of supertree methods
5. Reservations

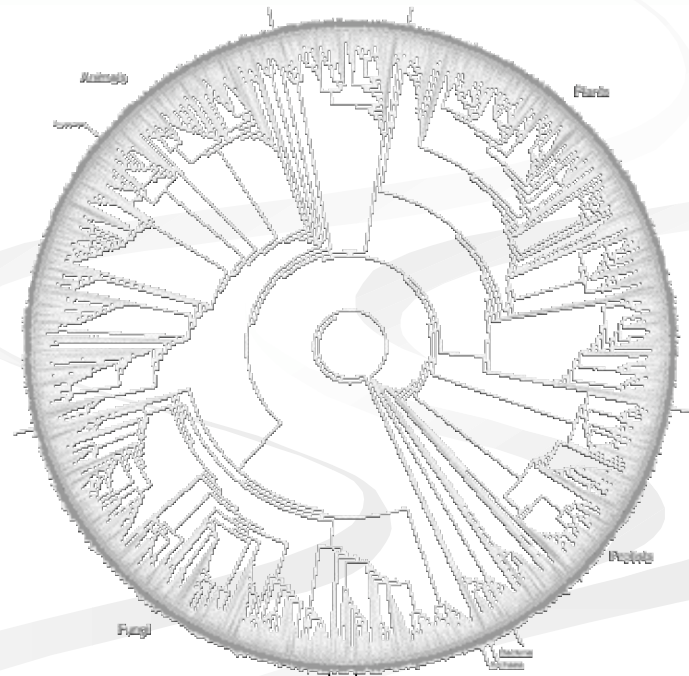
# An introduction to Supertrees

1. What is a supertree method?
2. Why supertrees?
3. A taste of supertree mathematics
4. A tour of supertree methods
5. Reservations

# Why supertrees? Goals

The quest for the best supertree is complicated by the fact that there are several quite different goals in supertree construction.

- a) Assist optimisation
- b) Compare, contrast, collate
- c) Uncover hidden phylogenetic information
- d) Analyse heterogeneous data

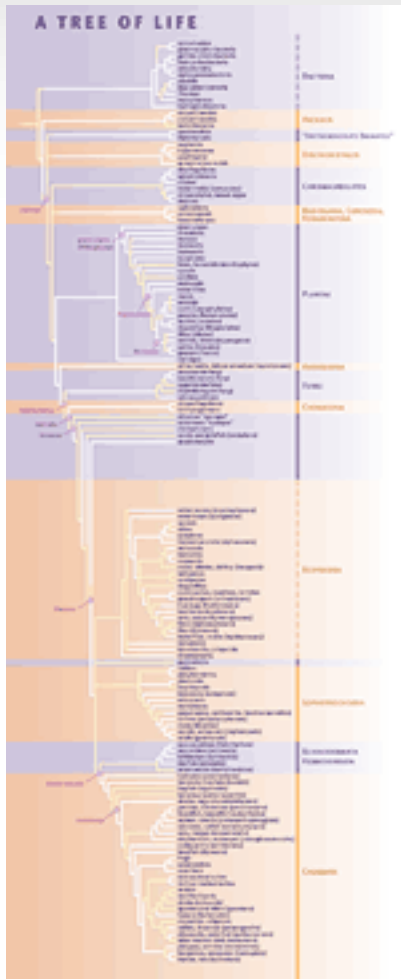




# (a) Optimisation

- Supertrees can help find the tree(s) optimizing some objective criterion.
- Divide and conquer is one of the oldest (and most effective) strategies in computer science.
- Three steps:
  - ❖ Divide sequences in smaller groups
  - ❖ Analyse each group separately
  - ❖ Combine analyses into a supertree
- Can give dramatic reductions in computing time

# (b) Compare - contrast - collate

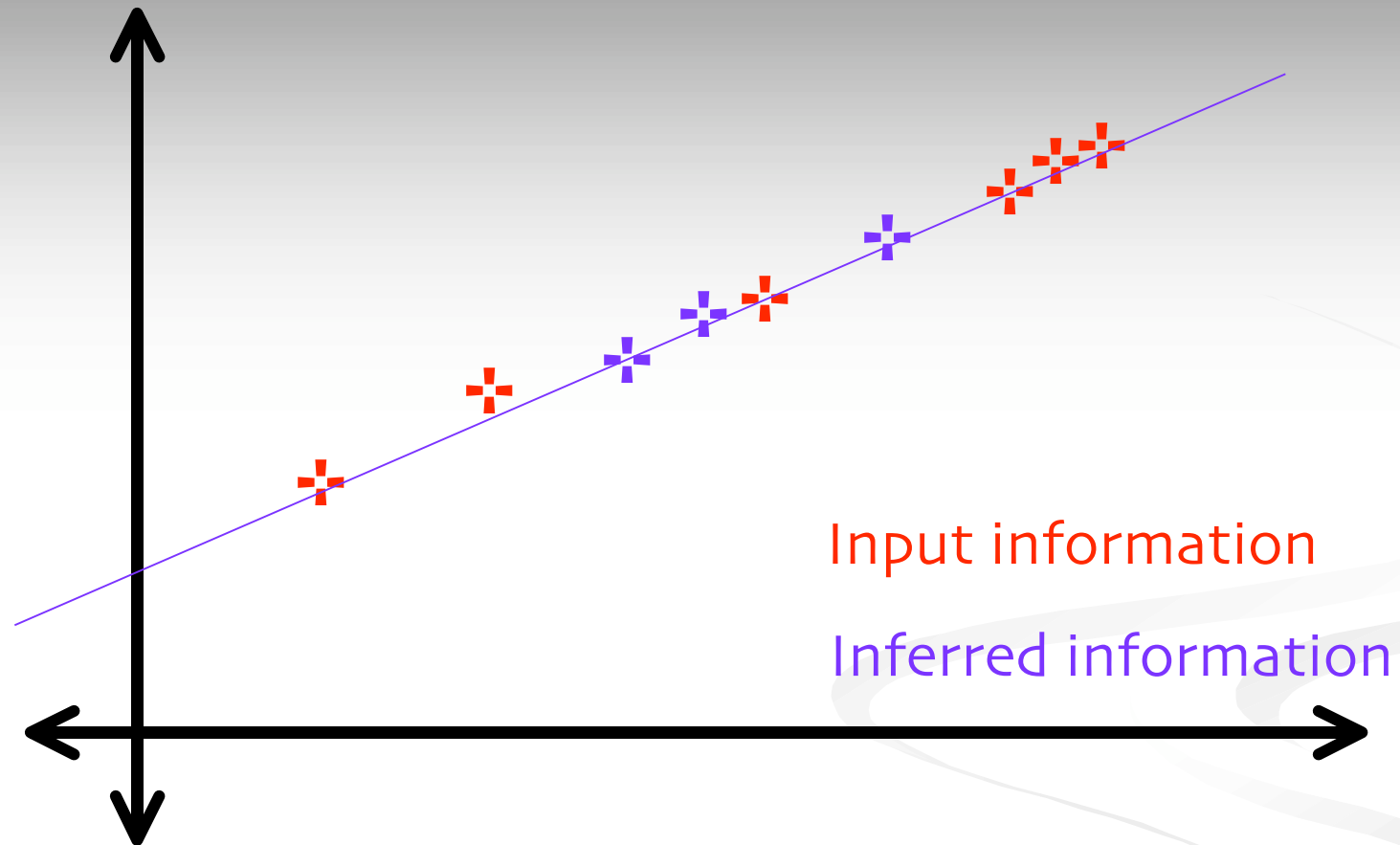


- The oldest `supertree' approach
- Exponentially growing number of phylogenetic studies
- Need to understand
  1. Wider relationships
  2. Interesting conflicts
  3. Poorly sampled groups
- e.g. mammal study of Liu et al. (315 articles)

# (c) Uncover hidden phylogenetic information

“Supertrees can make novel statements about relationships of taxa that do not co-occur on any single input tree while still retaining hierarchical information from the input trees.”

Mathematically: the assumption that all the input trees are sampled from a single tree allows us to make phylogenetic inferences not present in any *single* input tree.



**Supertree = interpolation**

# (d) Combine heterogeneous data

- Construct trees from different data sets then combine these trees
- The great *total evidence* versus *consensus* debate extended to trees with different leaf sets...
- Should combine more than one tree from each data source!
- Consider specialist tools (e.g. for gene tree - species tree problem)

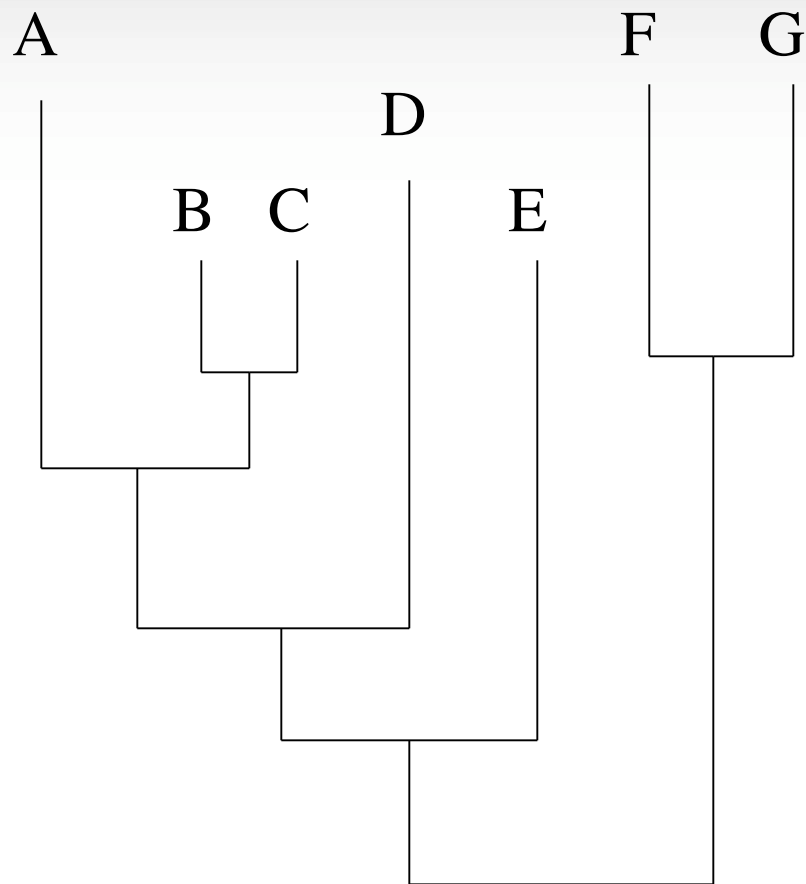


**USE WITH  
CARE**

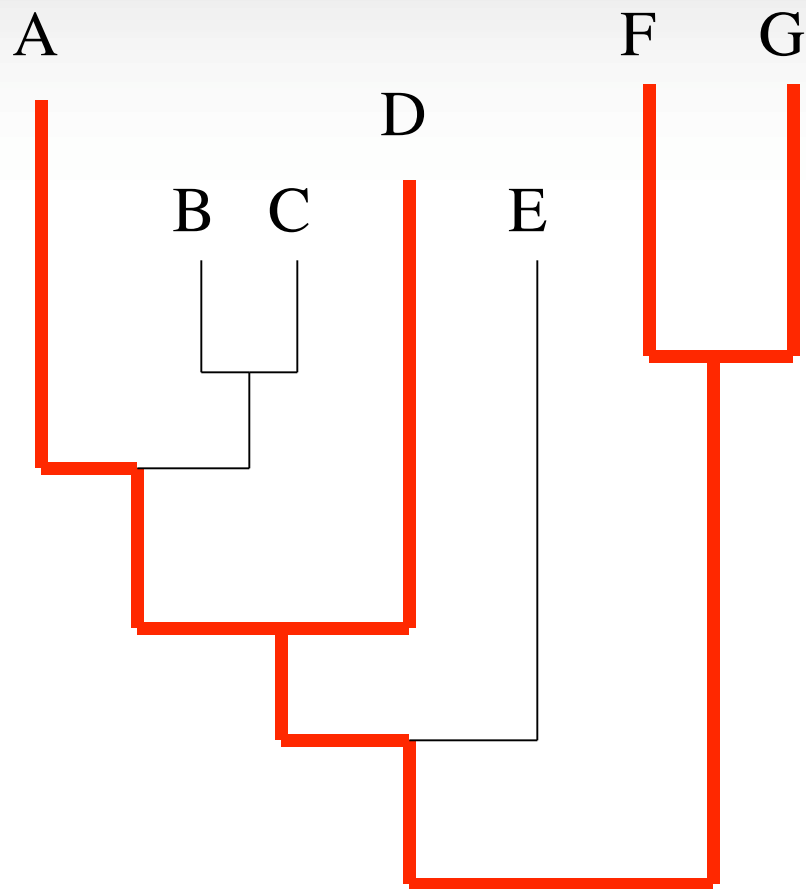
# An introduction to Supertrees

1. What is a supertree method?
2. Why supertrees?
3. A taste of supertree mathematics
4. A tour of supertree methods
5. Problems

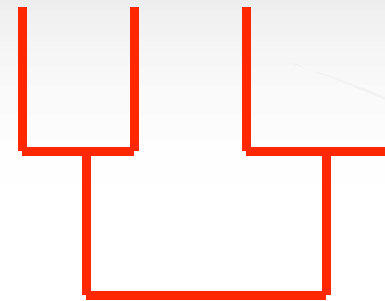
# Sampling from trees



# Sampling from trees

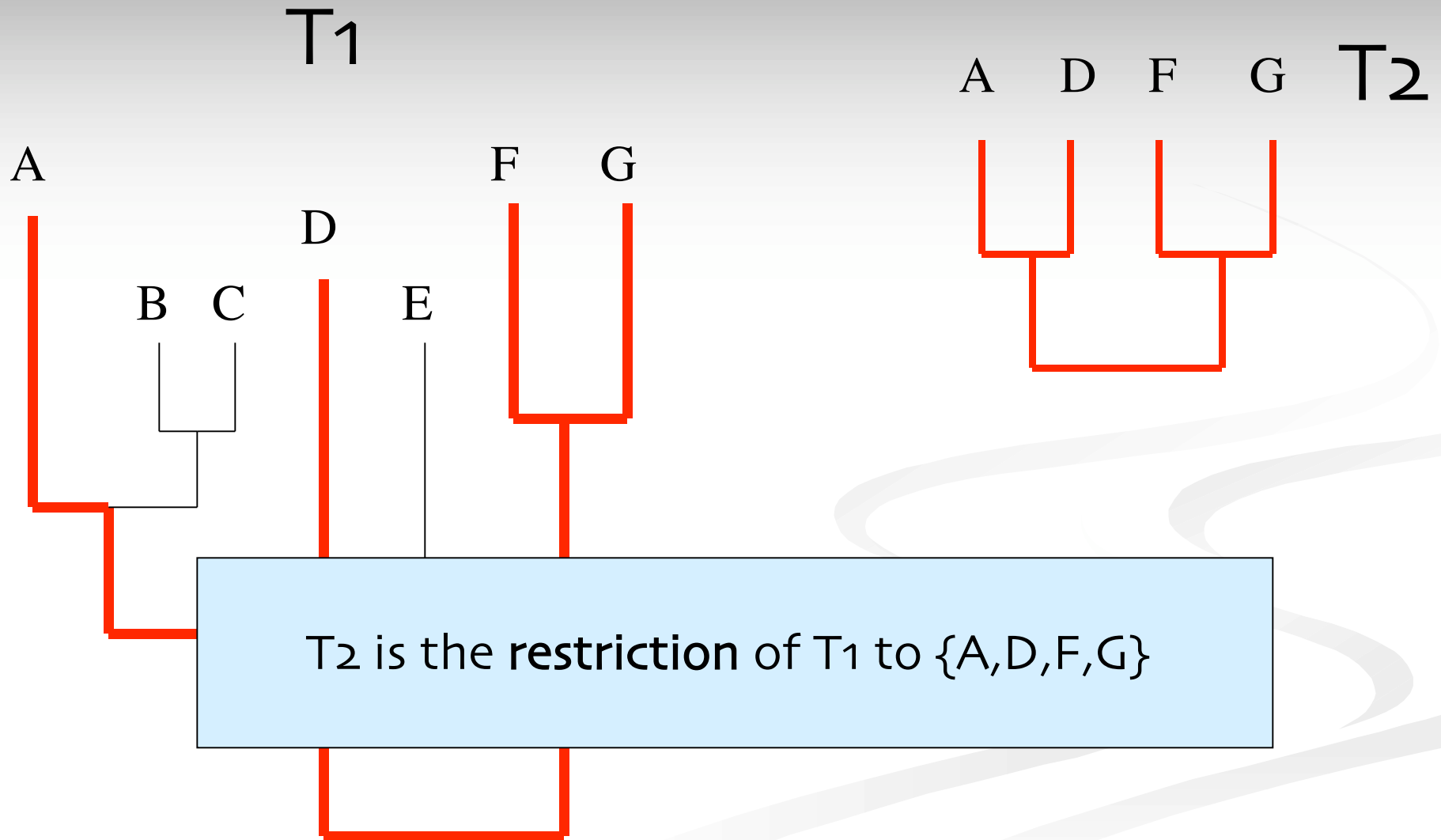


A D F G



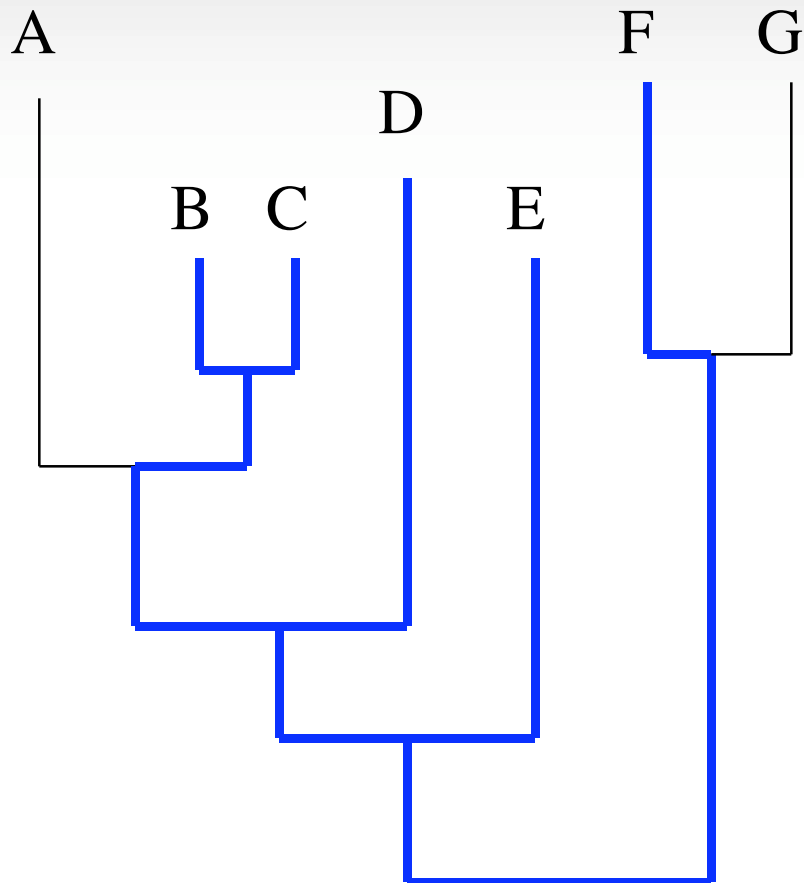


# Sampling from trees

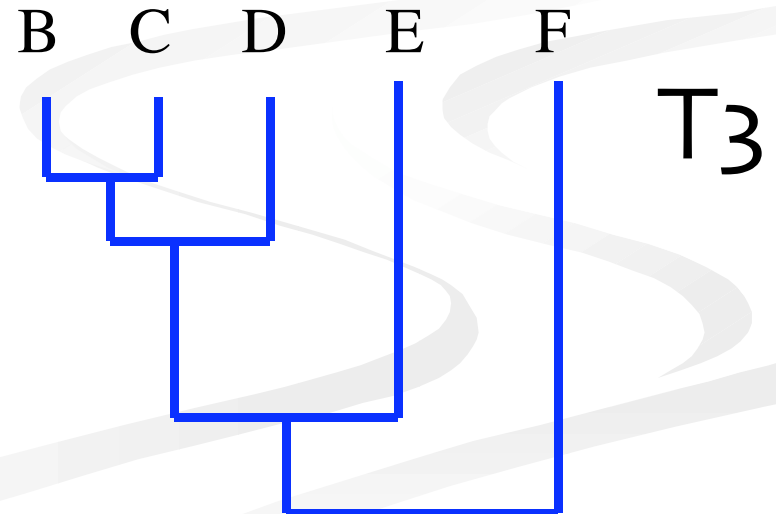
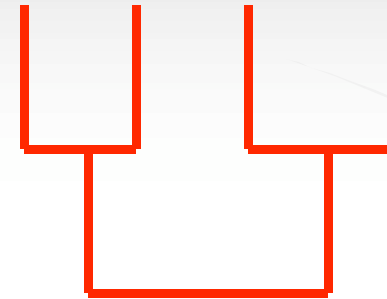


# Sampling from trees

T<sub>1</sub>

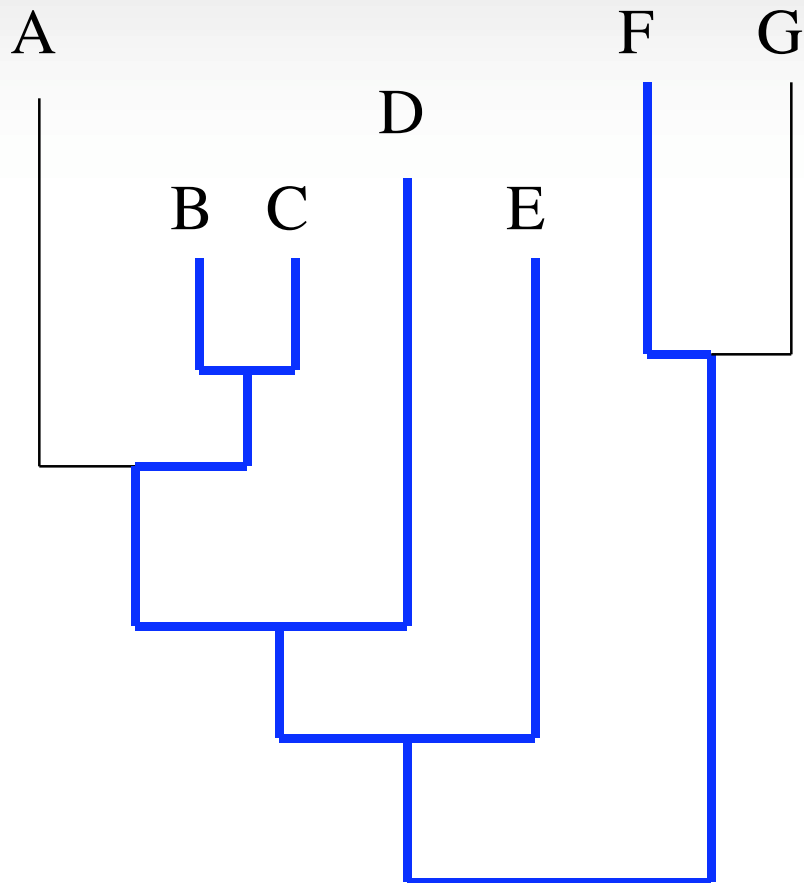


A D F G T<sub>2</sub>

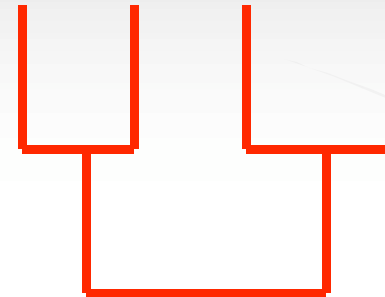


# Sampling from trees

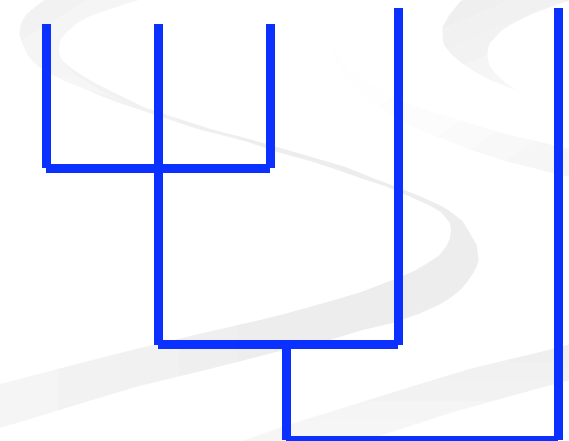
T<sub>1</sub>



A D F G T<sub>2</sub>

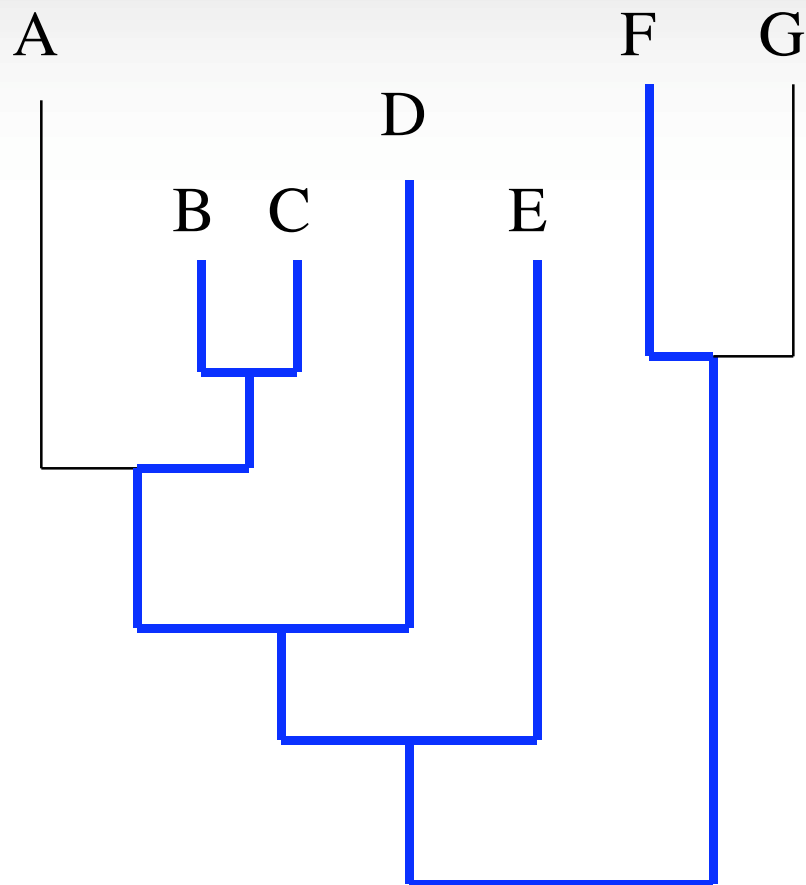


B C D E F T<sub>4</sub>

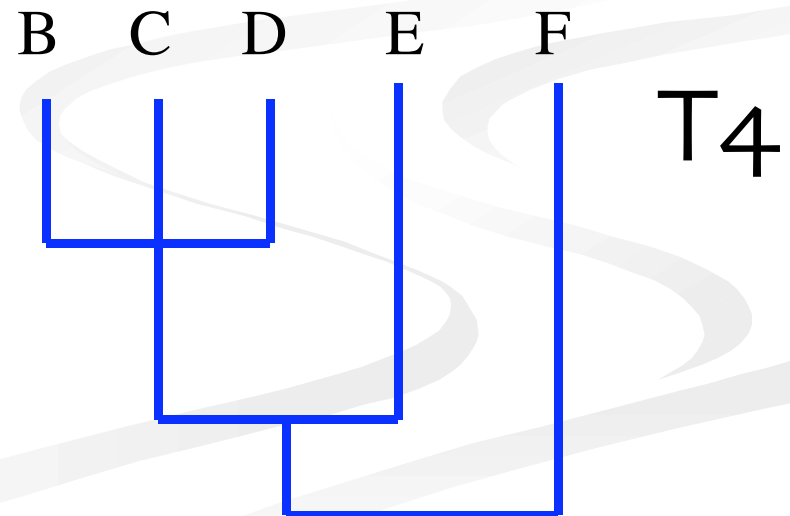


# Sampling from trees

T<sub>1</sub>

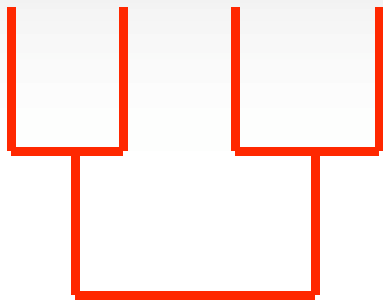


We say the T<sub>1</sub> displays T<sub>4</sub> because each clade in T<sub>4</sub> is a clade in T<sub>1</sub> restricted to the taxon set of T<sub>4</sub>

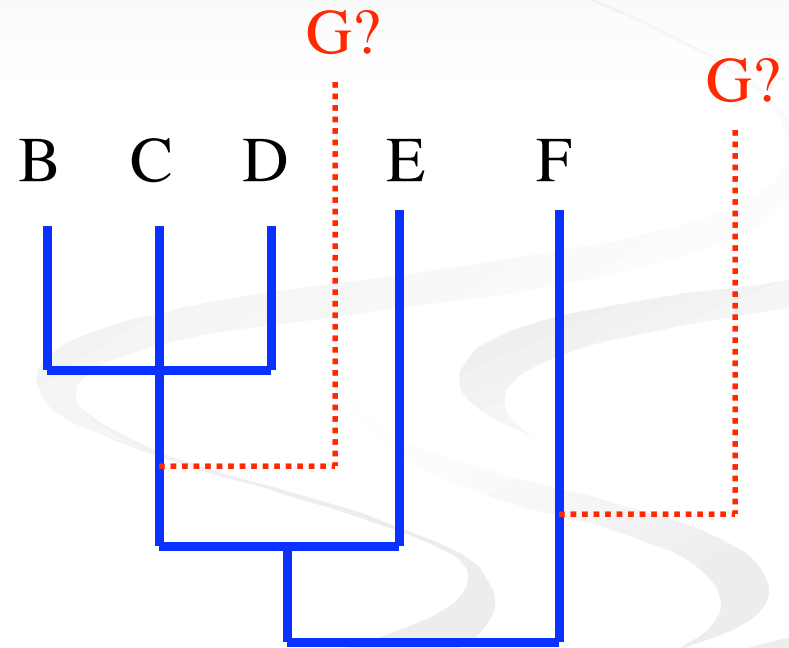
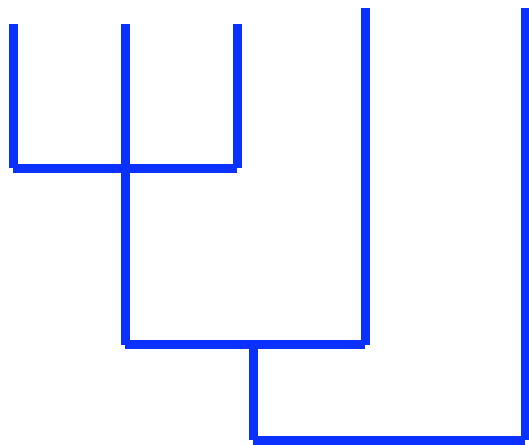


# Reconstructing trees

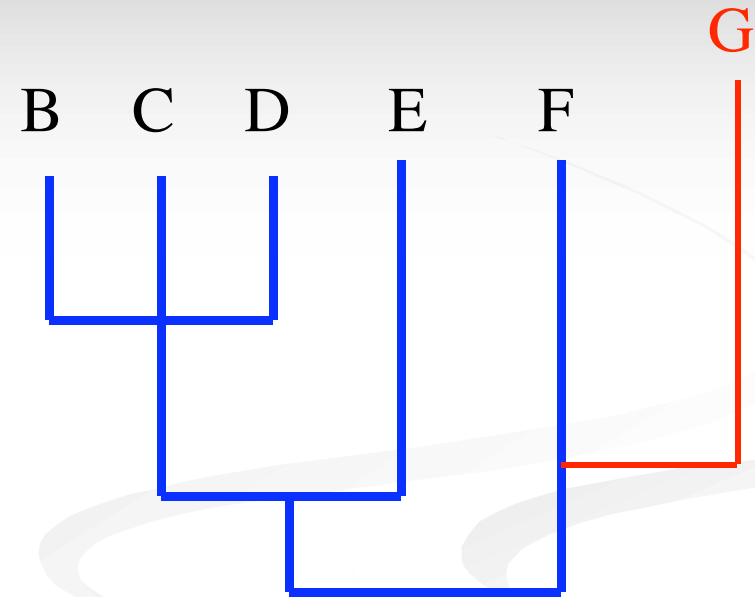
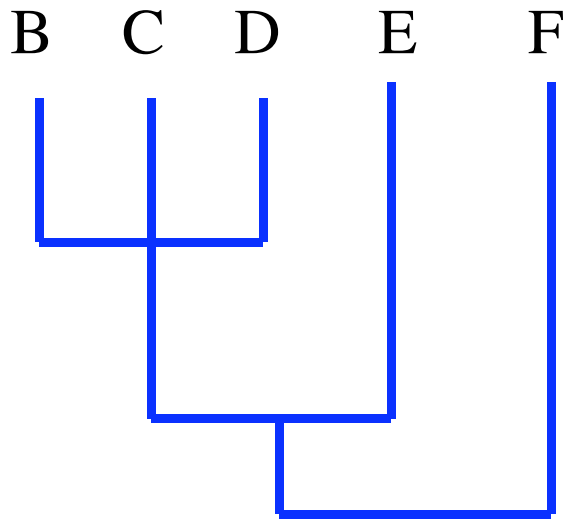
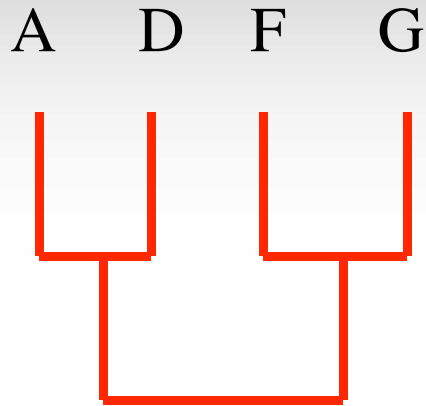
A D F G



B C D E F



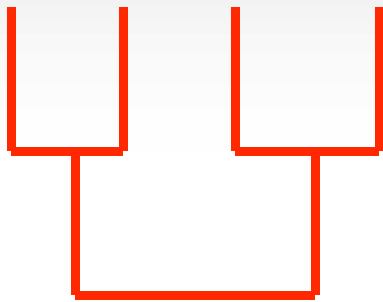
# Reconstructing trees



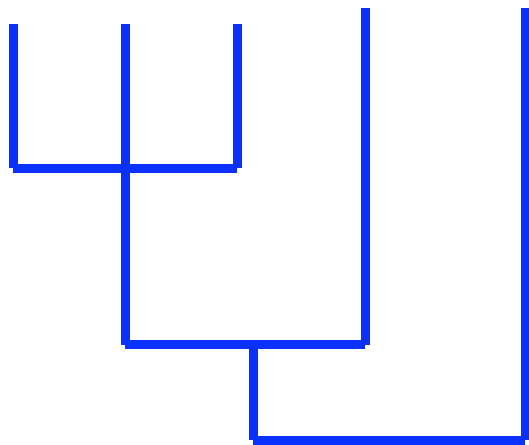
We can infer that B and C are more closely related than either is to G, even though there is no tree containing B, C, and G.

# Reconstructing trees

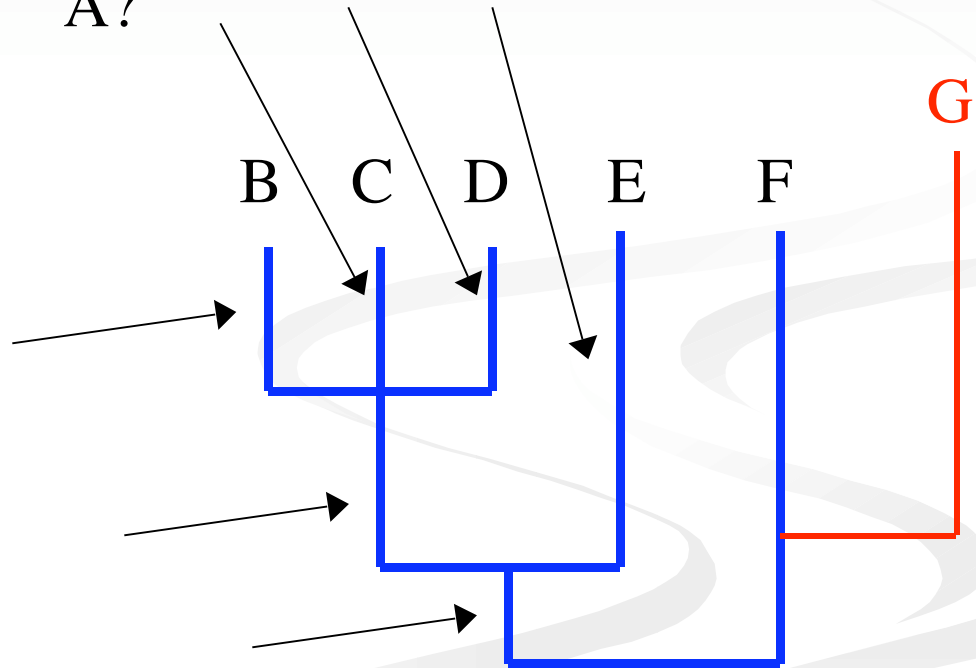
A D F G



B C D E F



A?

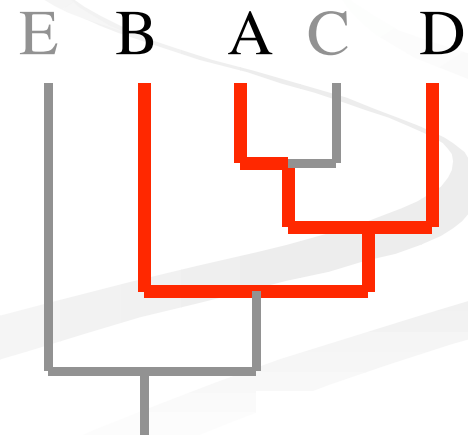
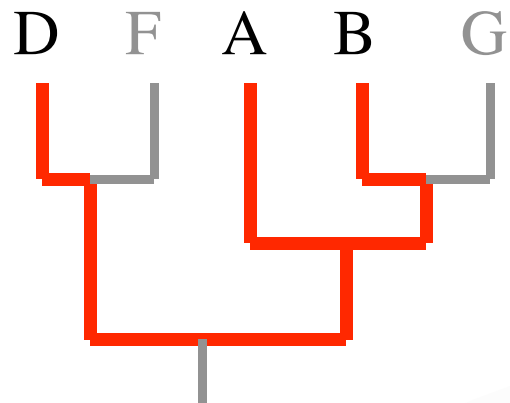
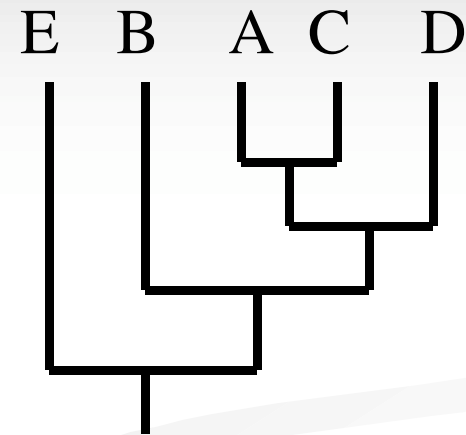
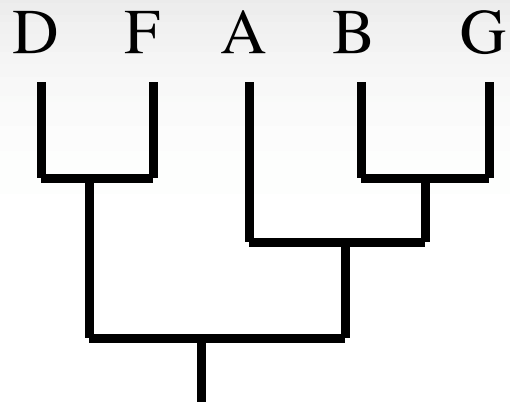


# Ambiguity

- A collection of sampled trees might not define a unique “parent” supertree.
- In fact there can be exponentially many different parent trees, and they may not be at all similar
- Every supertree method must resolve this problem in some way



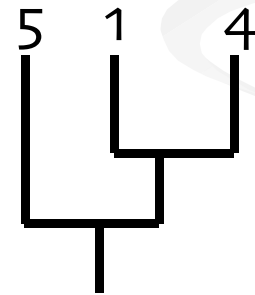
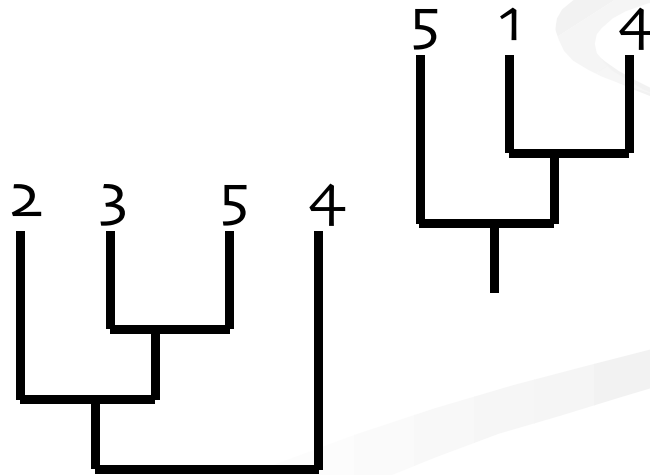
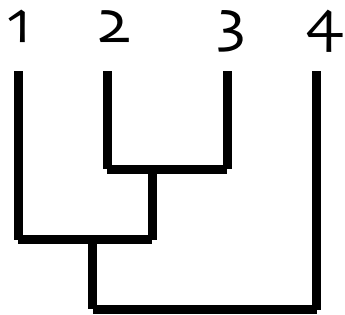
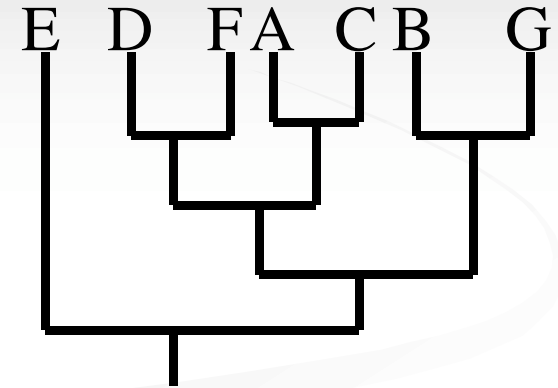
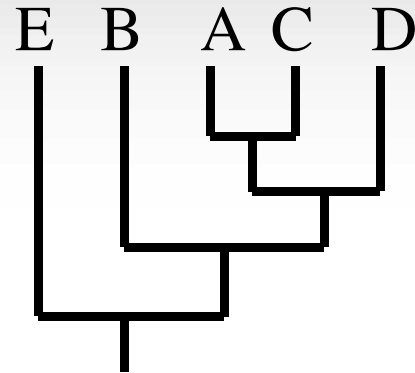
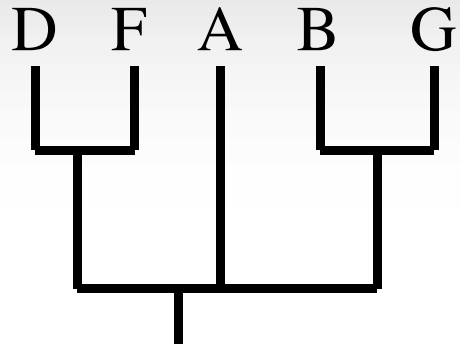
# Incompatibility



# Compatibility

- A set of trees is **compatible** if there is a tree that displays all of them.
- If it is not compatible, it is **incompatible**.
- Two trees are compatible if and only if they do not conflict on their overlapping taxa.

# Exercises



DOES  
NOT

COMP  
UTE

# Compatibility

- A set of trees is **compatible** if there is a tree that displays all of them.
- If it is not compatible, it is **incompatible**.
- Two trees are compatible if and only if they do not conflict on their overlapping taxa.
- Three (or more) trees might be incompatible even though each subset is compatible.
- Any supertree method must resolve the incompatibility problem in some way...

# Unrooted trees

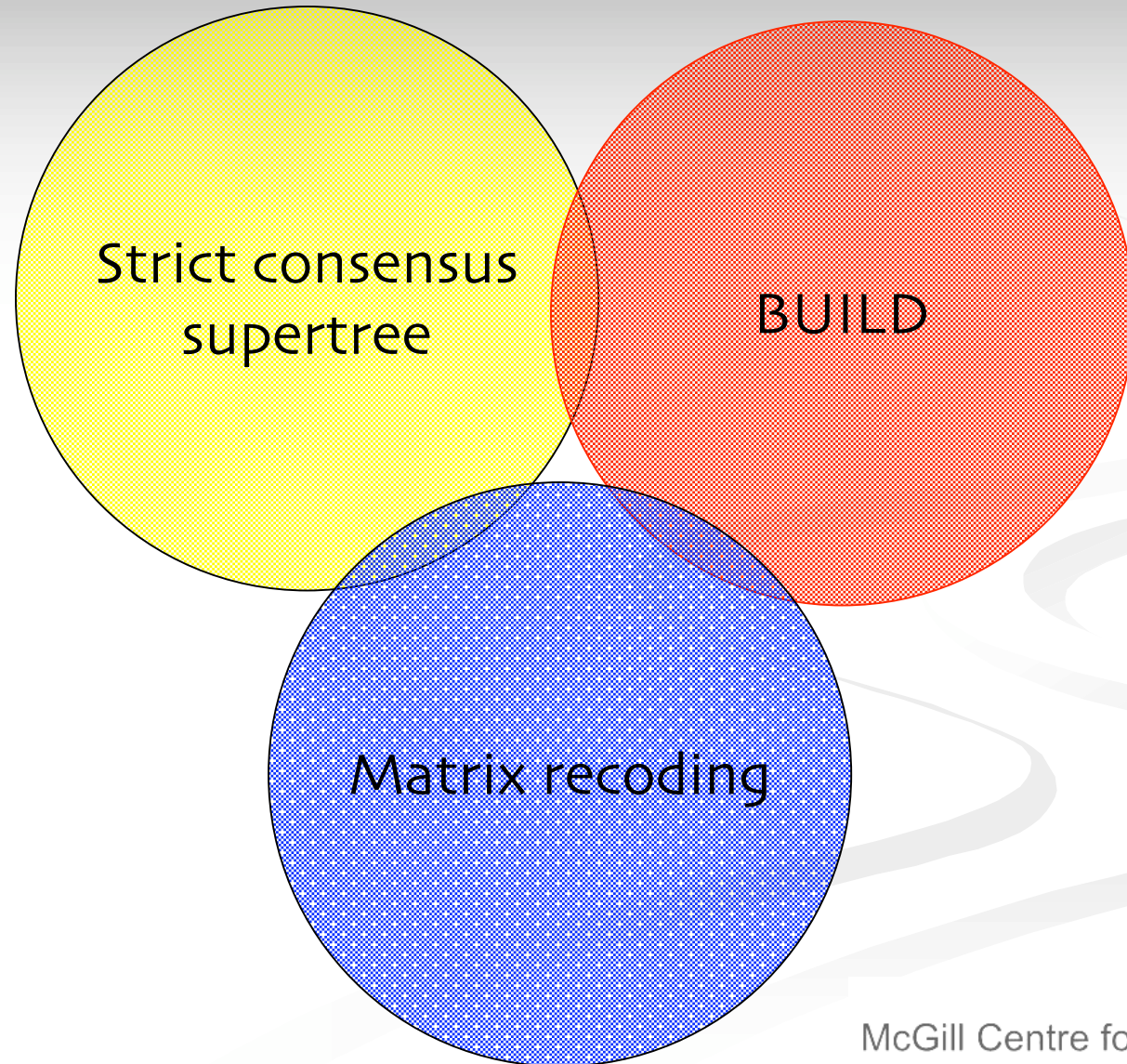
- The definitions of restriction, display, compatibility, apply even when we don't have information about the root
- Unrooted case is harder - there is no fast algorithm even to determine compatibility!
- Problems with ambiguity are also more severe
- On-going mathematical investigation...



# An introduction to Supertrees

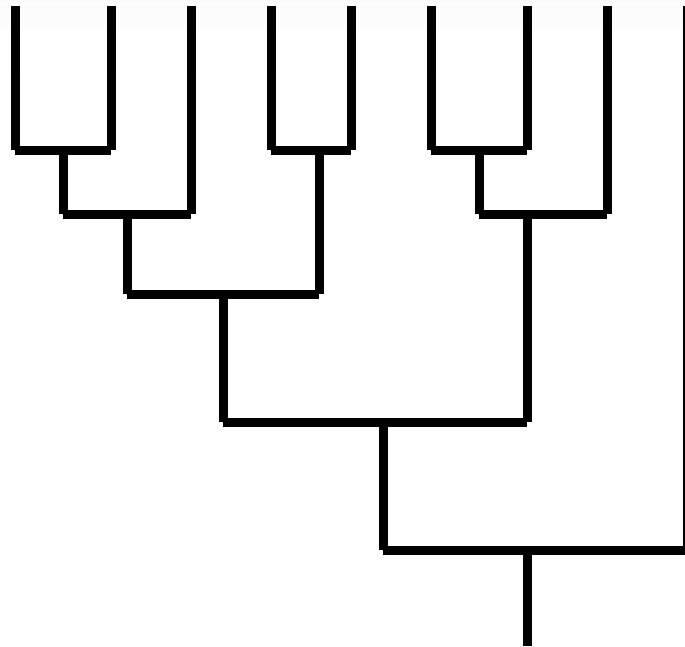
1. What is a supertree method?
2. Why supertrees?
3. A taste of supertree mathematics
4. A tour of supertree methods
5. Reservations

# The Three Supertree Methods

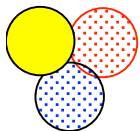
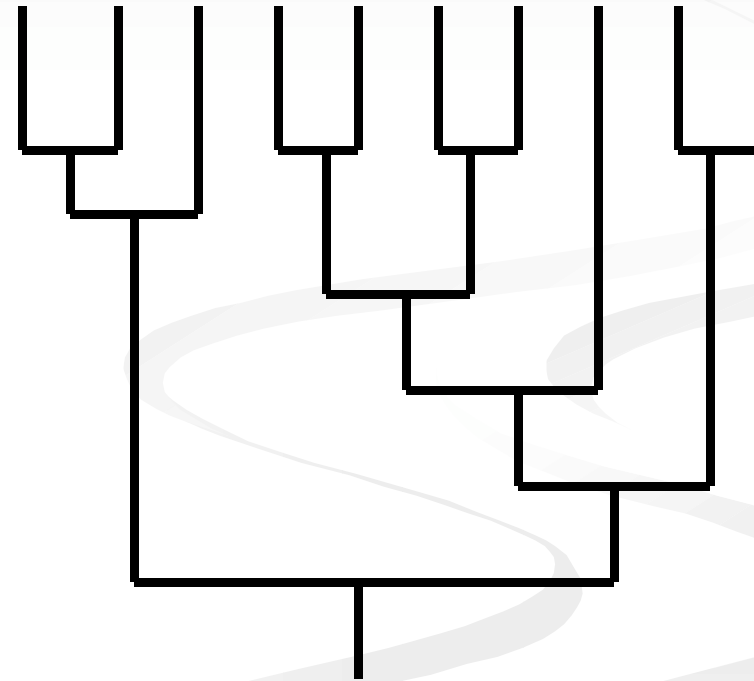


# Strict consensus supertree

*1 2 6 7 8 3 4 5 9*



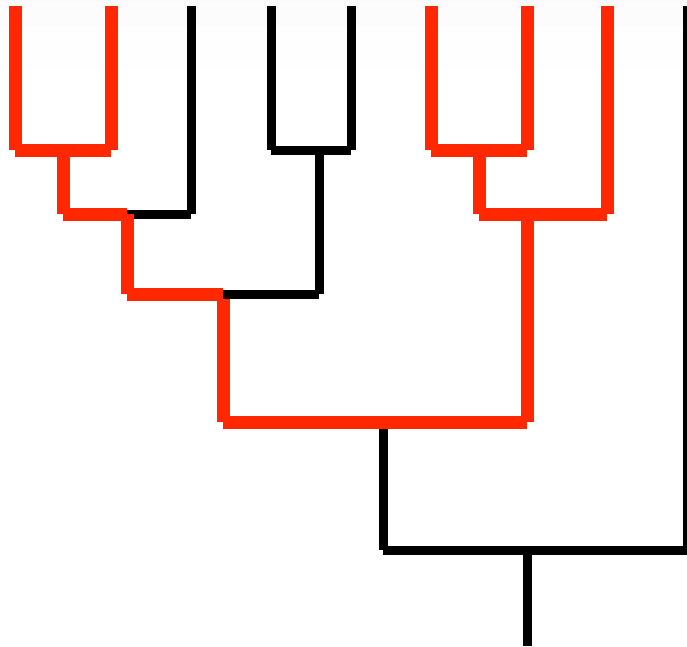
*1 2 10 3 5 11 12 13 4 14*



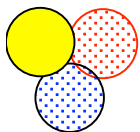
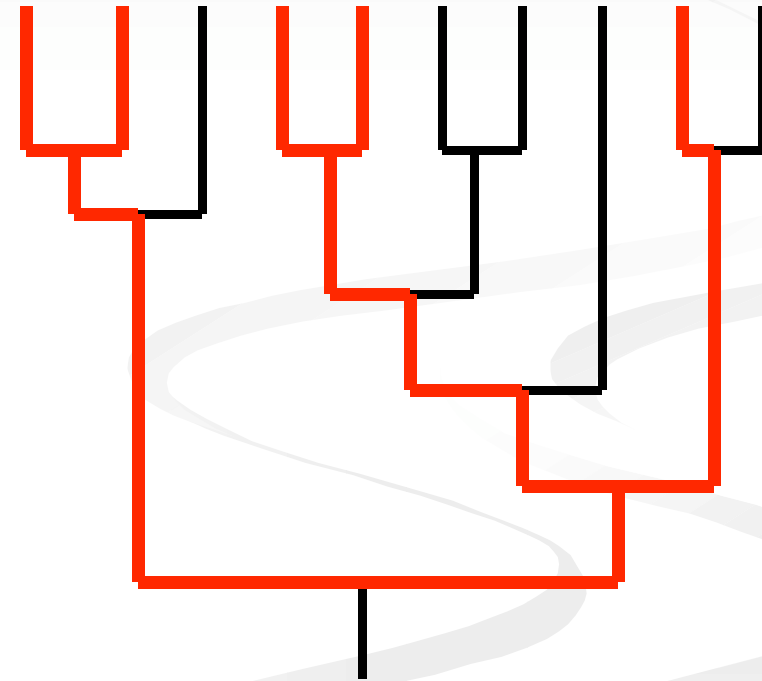


# Strict consensus supertree

*1 2 6 7 8 3 4 5 9*

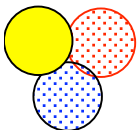


*1 2 10 3 5 11 12 13 4 14*



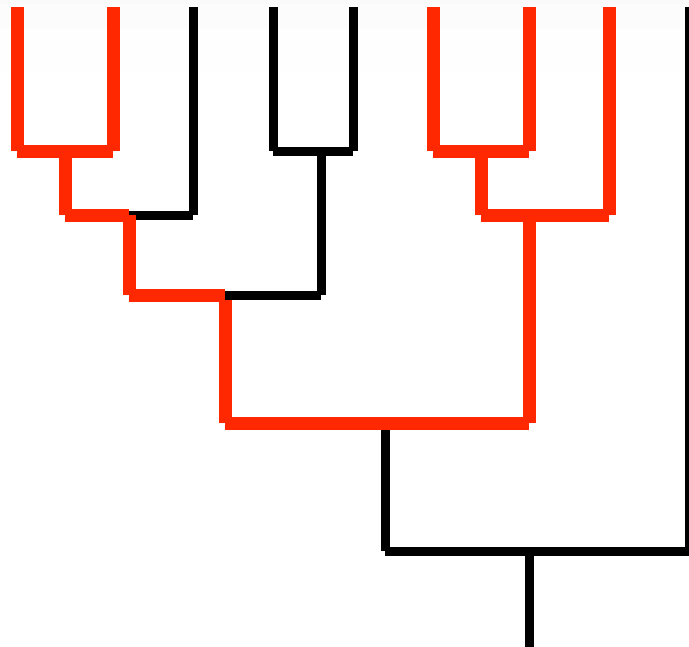
# Strict consensus supertree

- Step one: contract edges until both trees agree on their overlapping edges

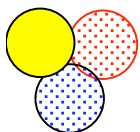
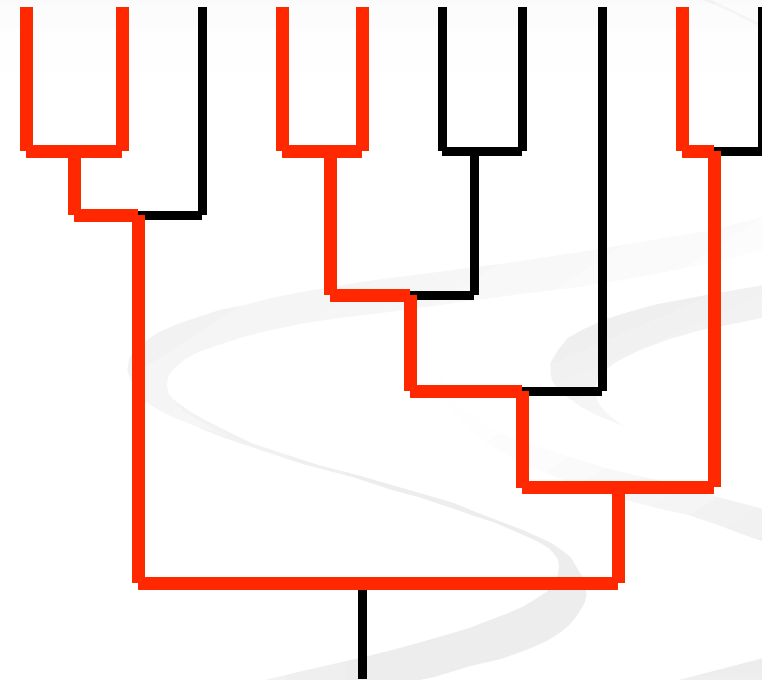


# Strict consensus supertree

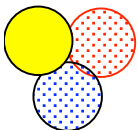
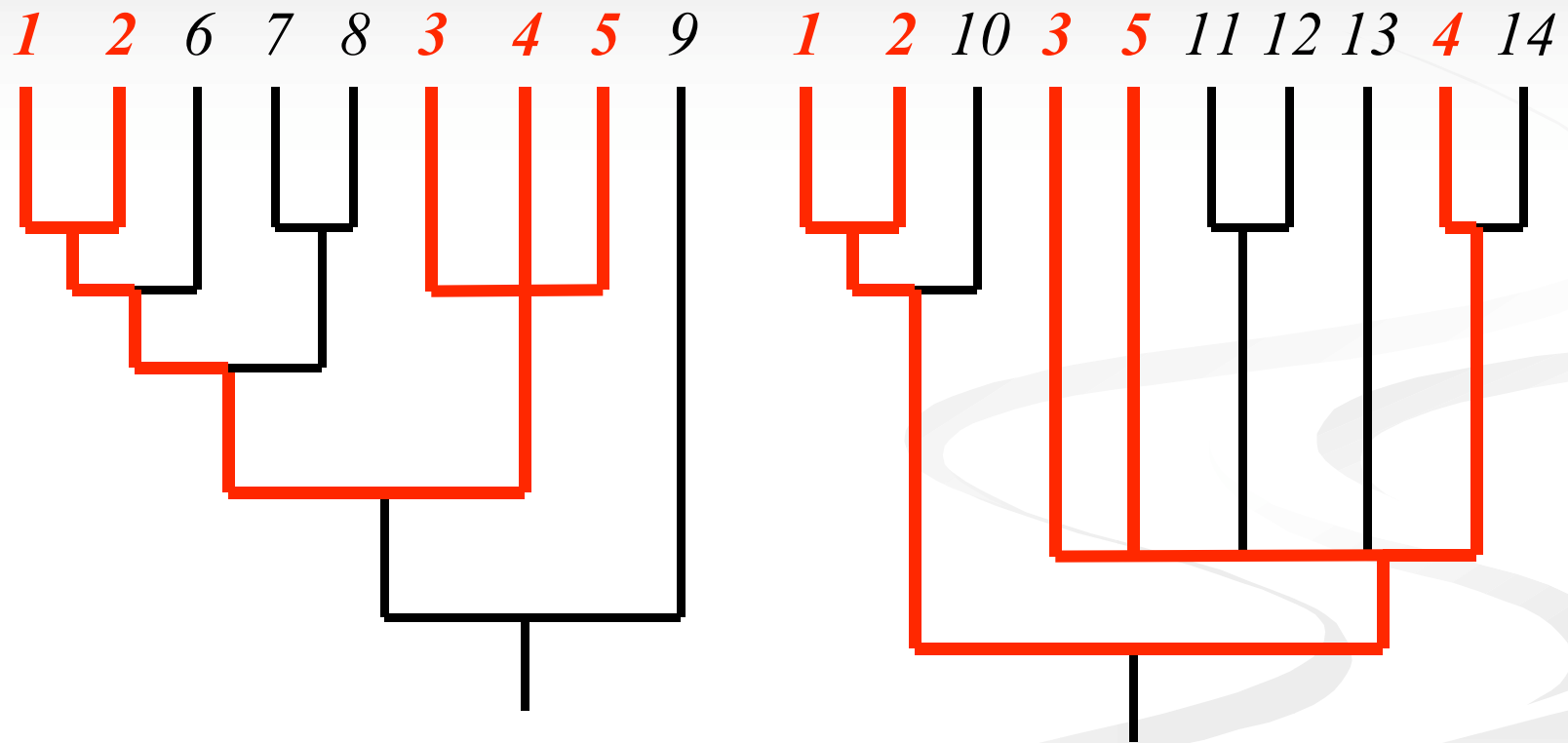
*1 2 6 7 8 3 4 5 9*



*1 2 10 3 5 11 12 13 4 14*

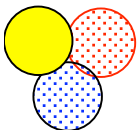


# Strict consensus supertree

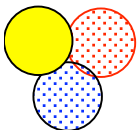
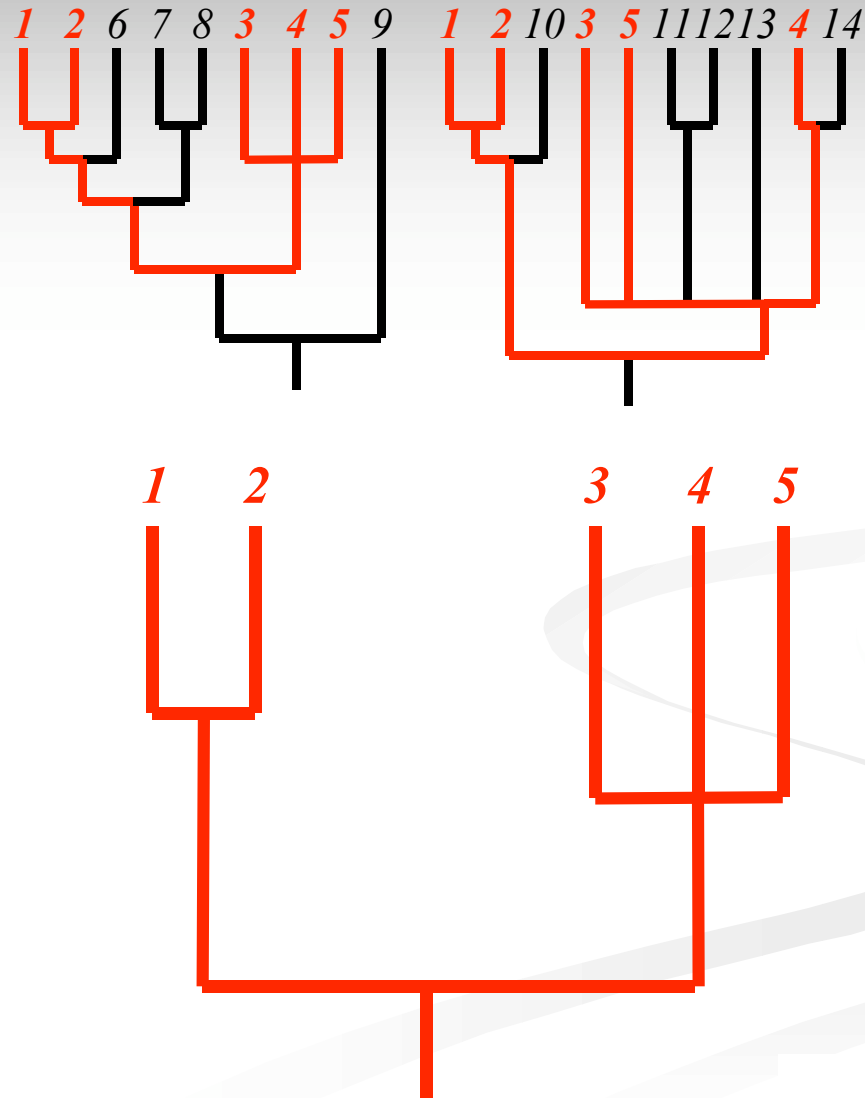


# Strict consensus supertree

- Step one: contract edges until both trees agree on their overlapping edges
- Step two: extract the part of the trees that overlap

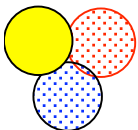


# Strict consensus supertree

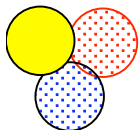
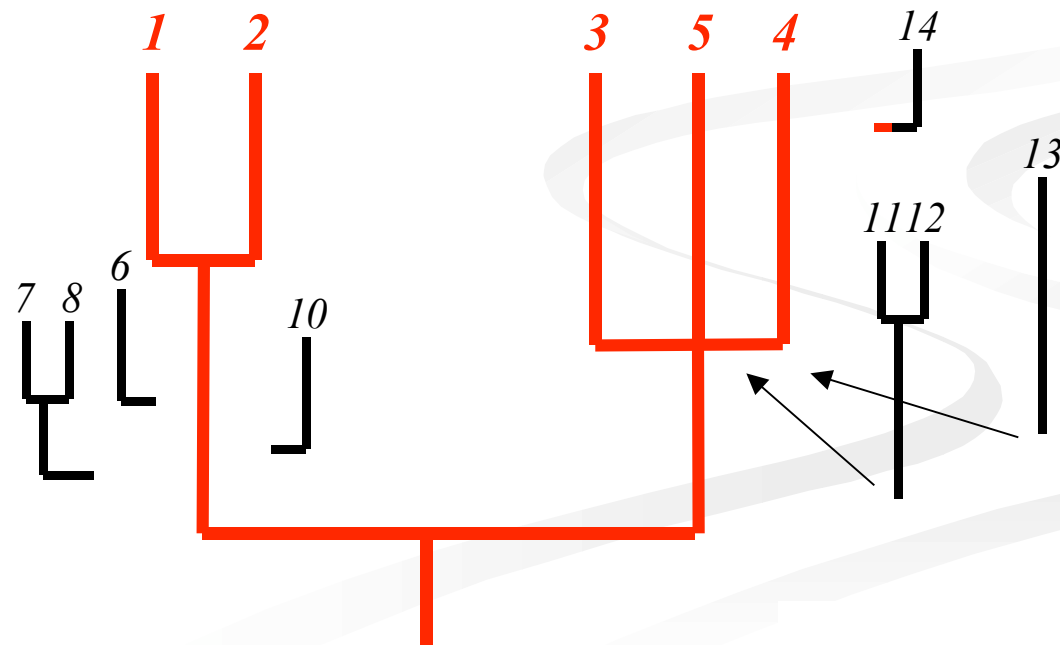
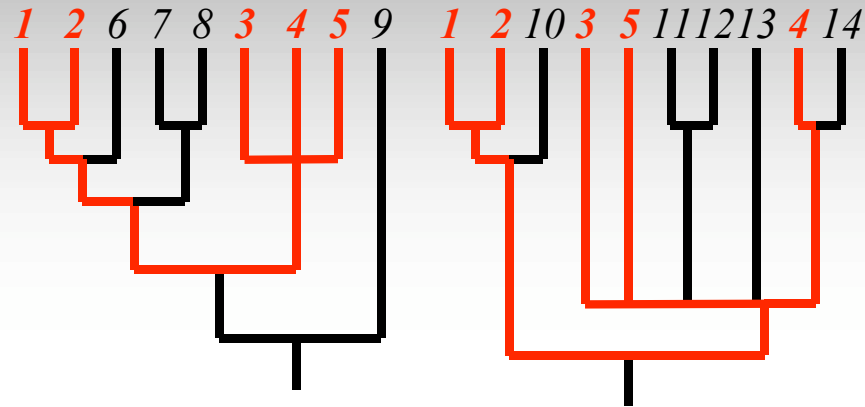


# Strict consensus supertree

- Step one: contract edges until both trees agree on their overlapping edges
- Step two: extract the part of the trees that overlap
- Step three: locate “grafting” points for remaining portions of the input trees



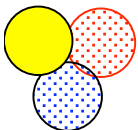
# Strict consensus supertree



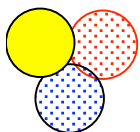
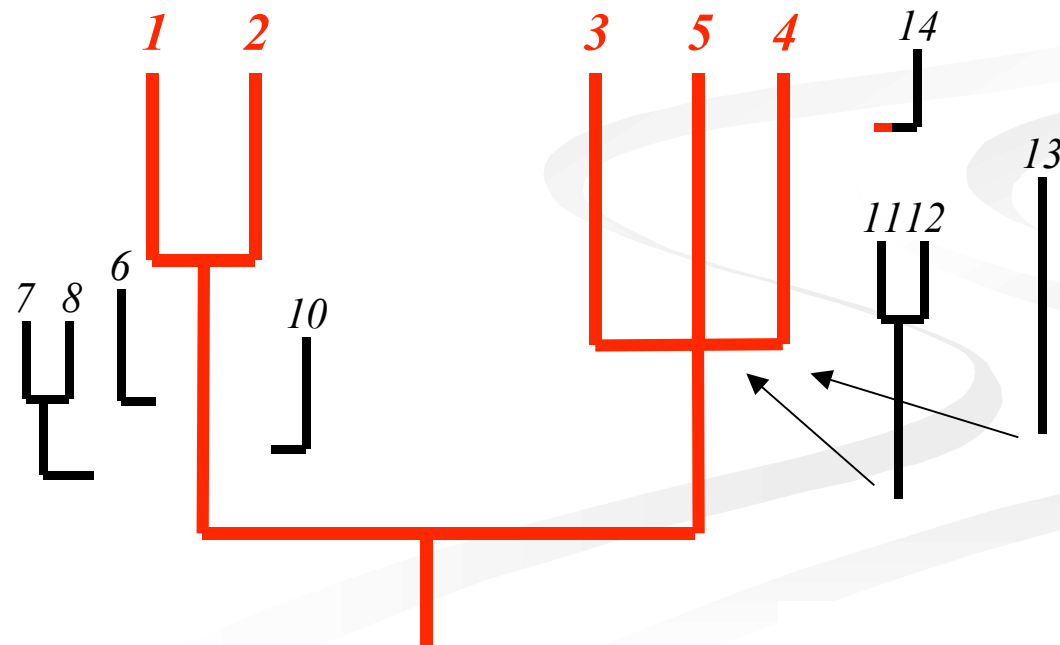
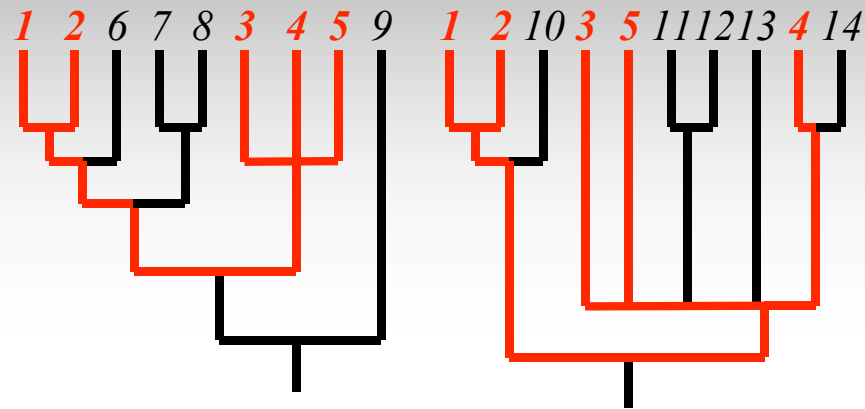


# Strict consensus supertree

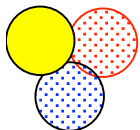
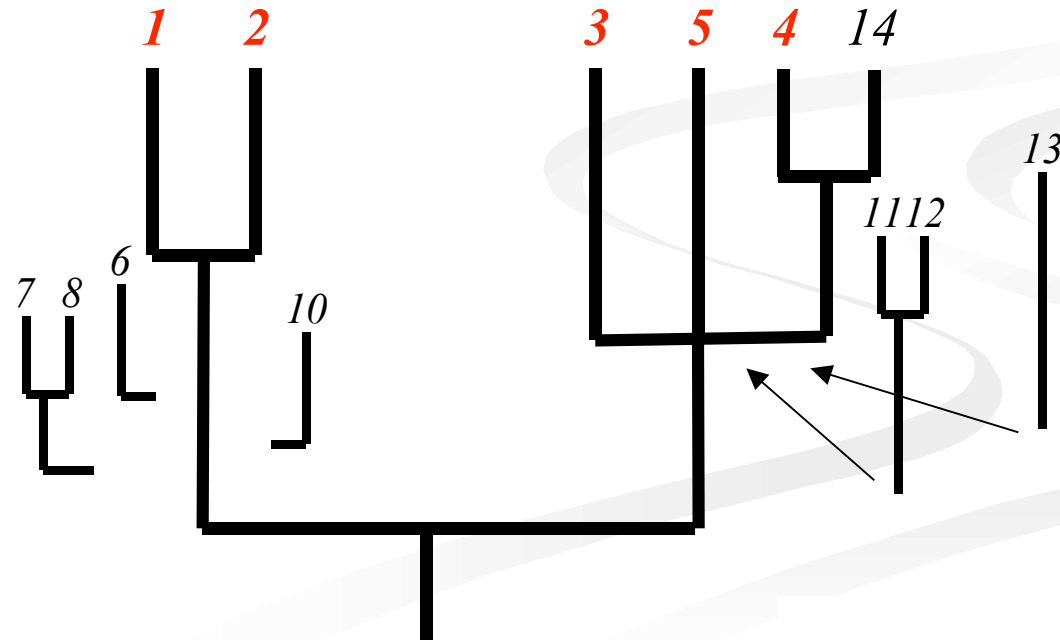
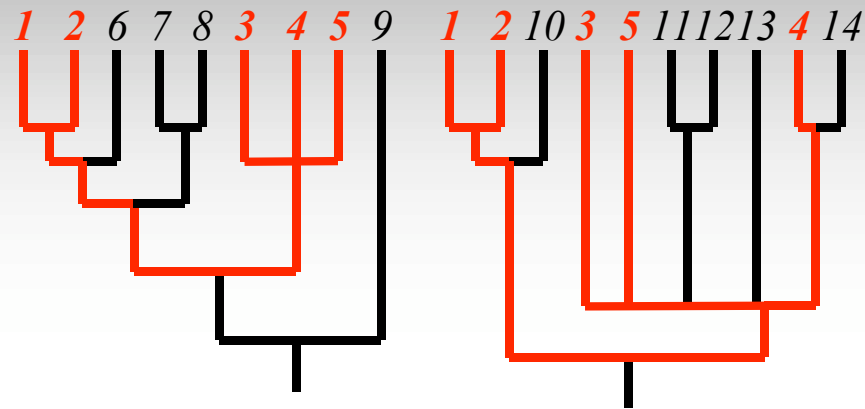
- ❑ Step one: contract edges until both trees agree on their overlapping edges
- ❑ Step two: extract the part of the trees that overlap
- ❑ Step three: locate “grafting” points for remaining portions of the input trees
- ❑ Step four: graft on extra portions. If parts of both trees need to graft onto the same place, collapse all clusters in both parts



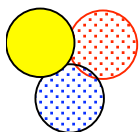
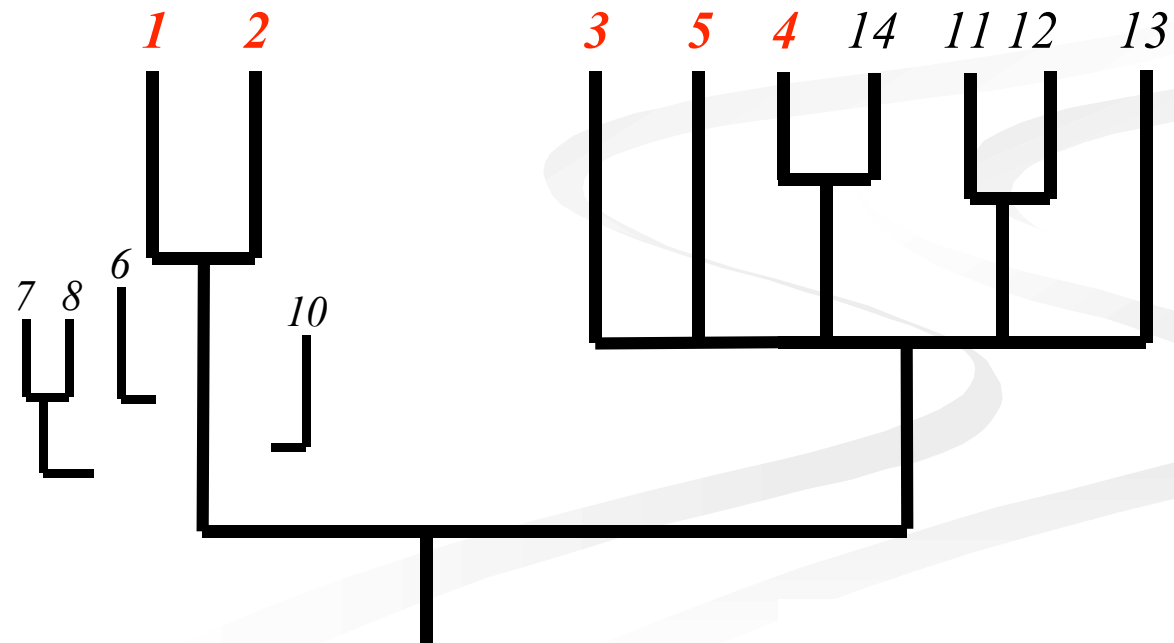
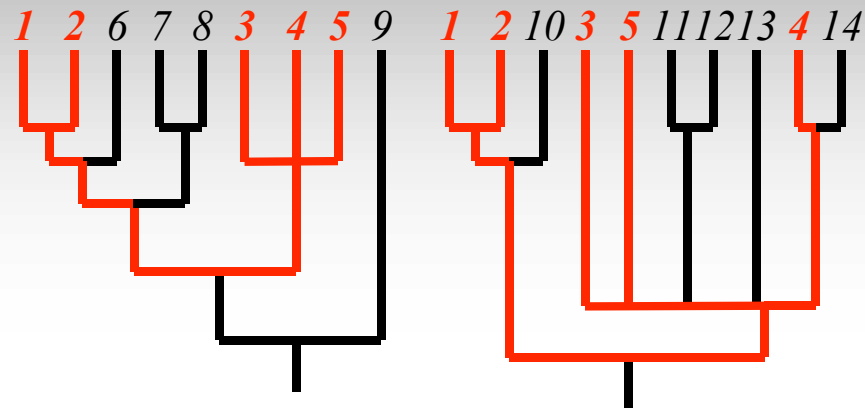
# Strict consensus supertree



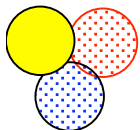
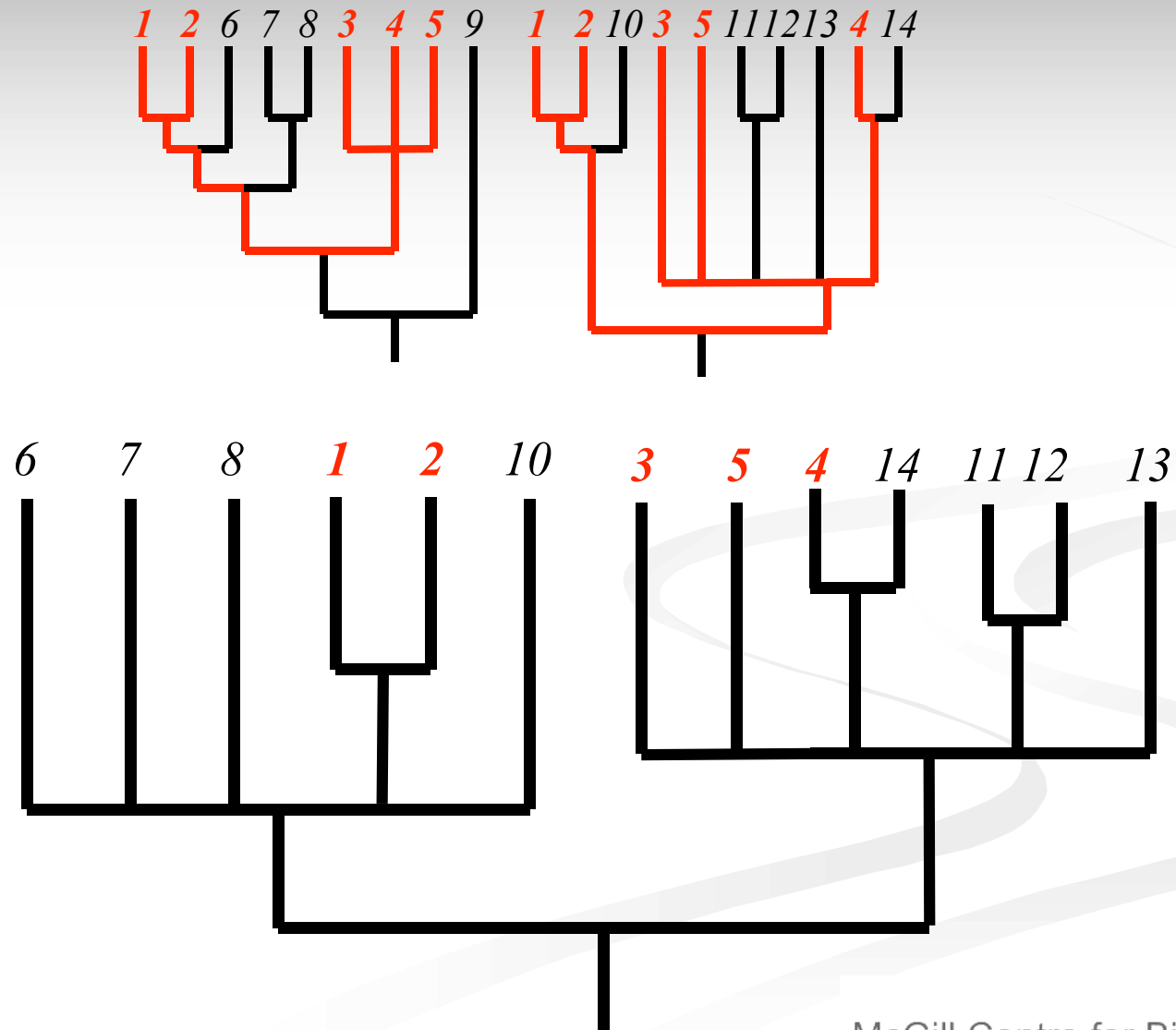
# Strict consensus supertree



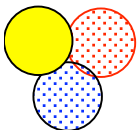
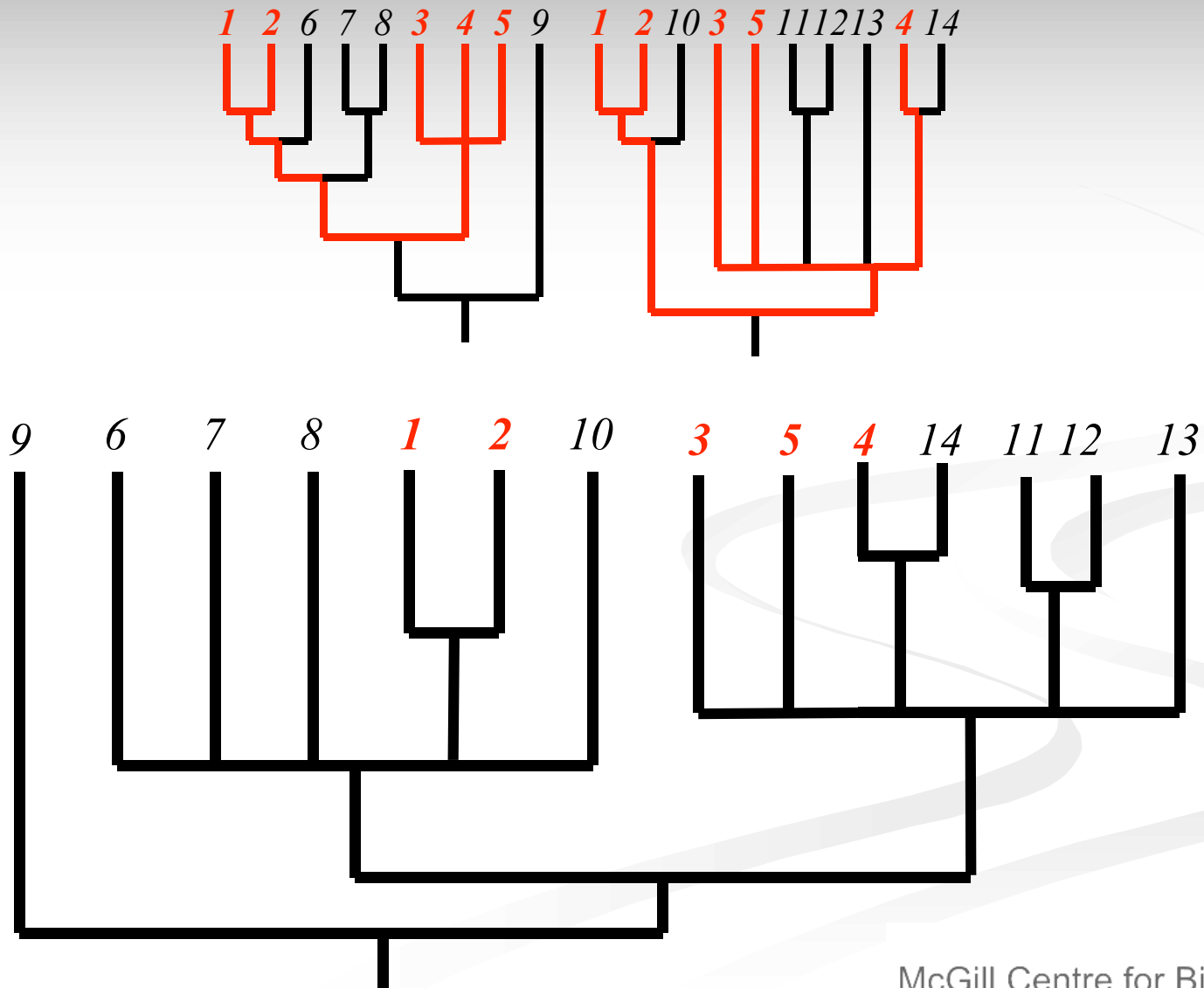
# Strict consensus supertree



# Strict consensus supertree

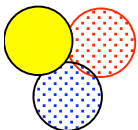
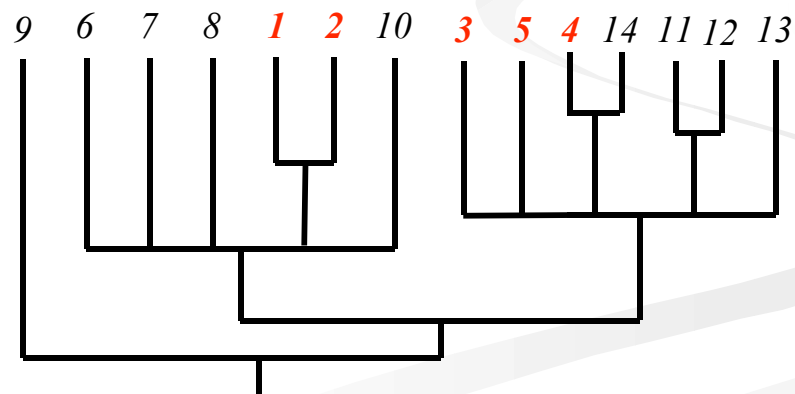


# Strict consensus supertree

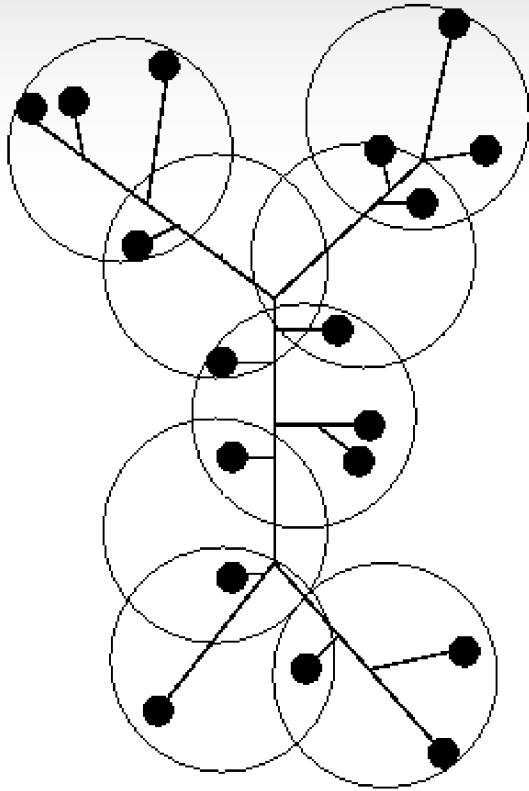


# Properties

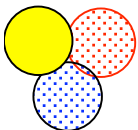
- Fast (linear time?)
- If the two trees agree on their overlapping parts, the trees returned will be the *strict consensus* of all the trees that display the input trees
- Very conservative (lots of multifurcations)
- The algorithm does not extend directly to more than two trees as three trees might be incompatible even if they agree on all overlaps.



# Disk-cover

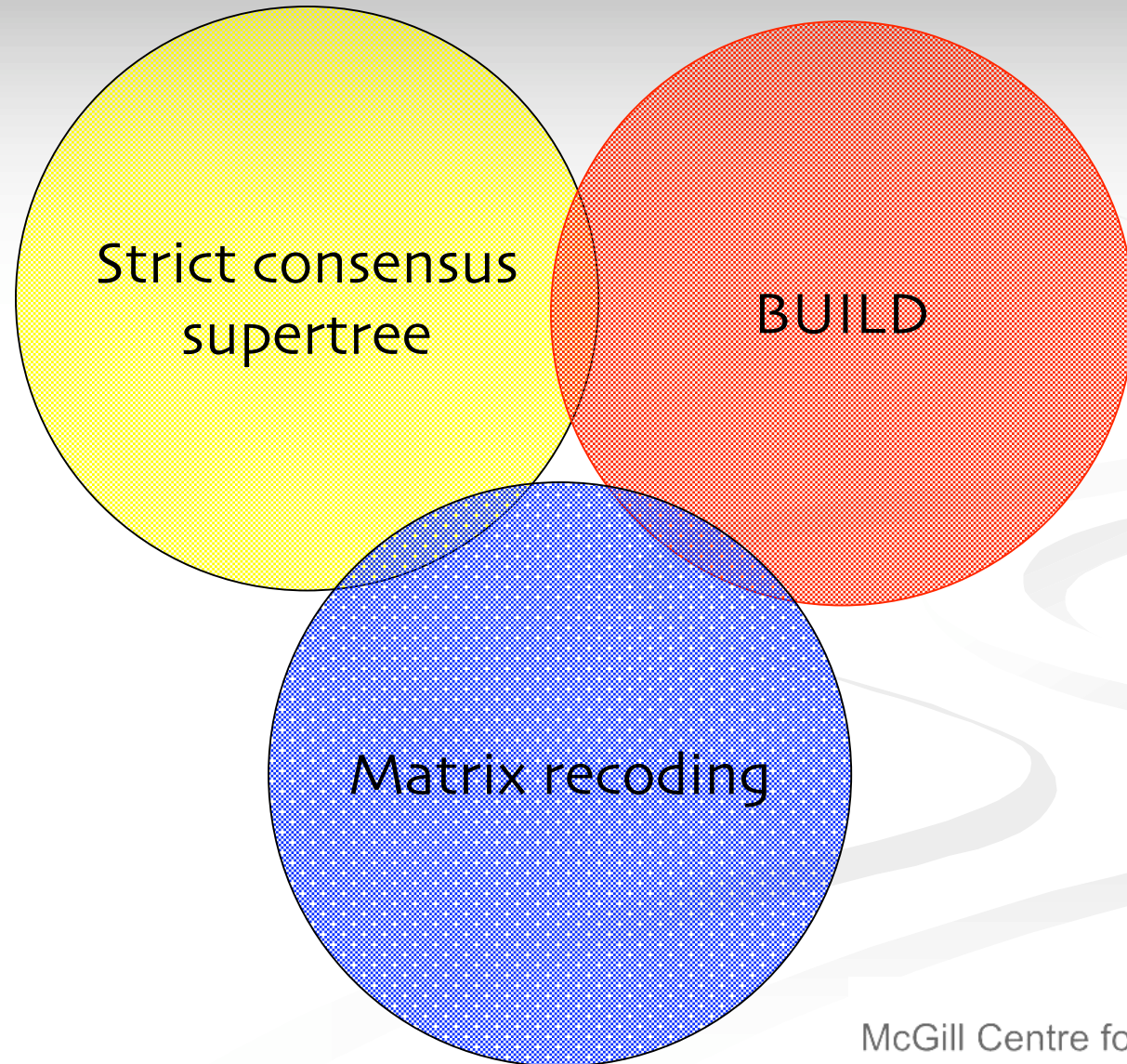


- Strict consensus supertrees extended to unrooted trees by Warnow et al. as part of the disk-cover method
- First step of disk-cover is to divide data into sets of small diameter
- Separate analyses are combined in an agglomerative fashion using the strict consensus supertree algorithm
- Improved performance on trees that are not too “deep”
- Remarkable speed-ups with gene order analysis (wait till Wed.)



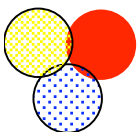


# The Three Supertree Methods



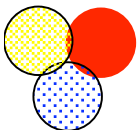
# BUILD

- Introduced by Aho, Sagiv, Szymanski & Ullman in 1981 - but for a problem in database design.
- Extended and refined by Constantinescu & Sankoff (1995), McMorris & Ng (1996), Henzinger & Warnow (1996), Semple (2002).
- Can be used to (quickly) check whether a collection of rooted trees is compatible or not
- Works by subdividing and subdividing and subdividing...

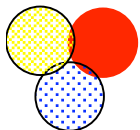
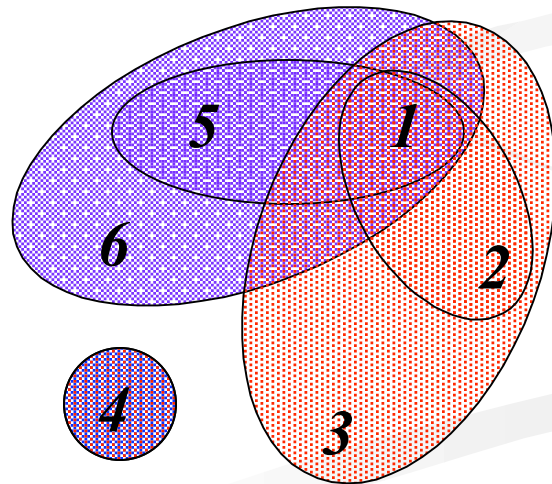
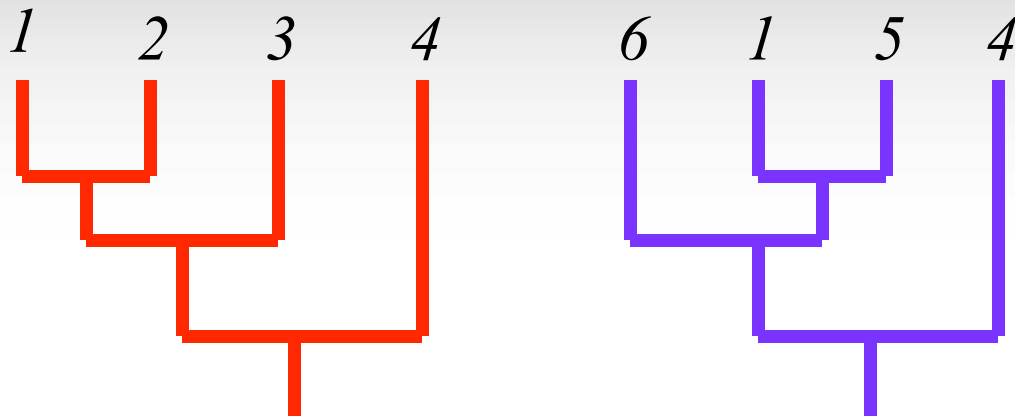


# BUILD

- Step one: Divide the taxa into two or more 'super-clades'. Put two taxa in the same clade if they are in the same clade of one of the input trees.

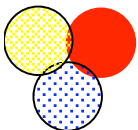


# BUILD

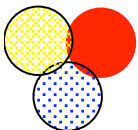
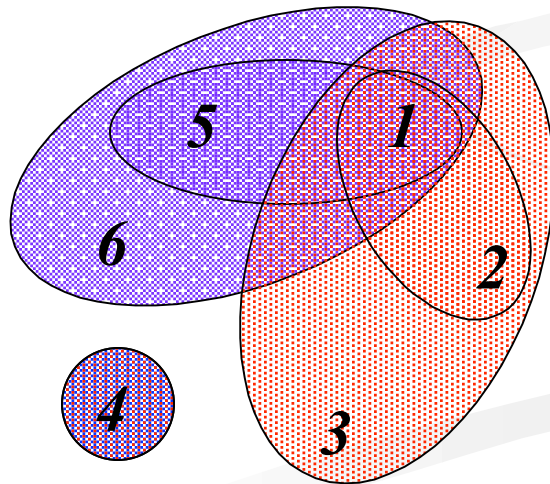
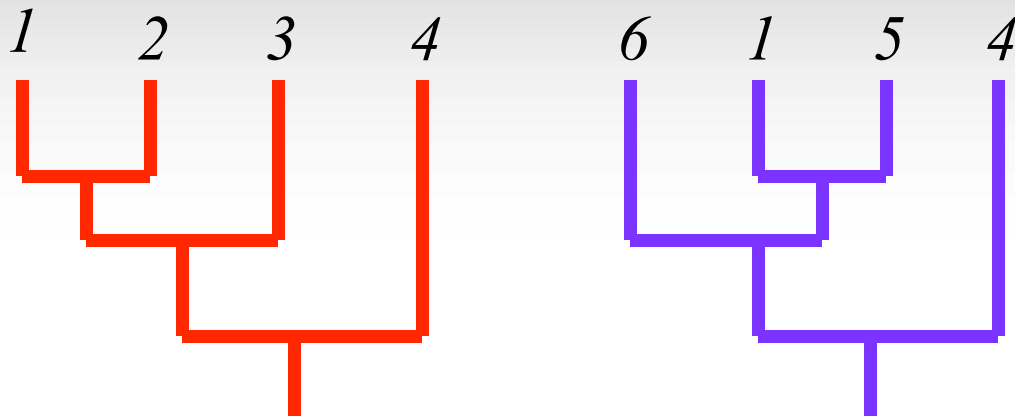


# BUILD

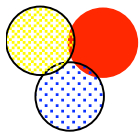
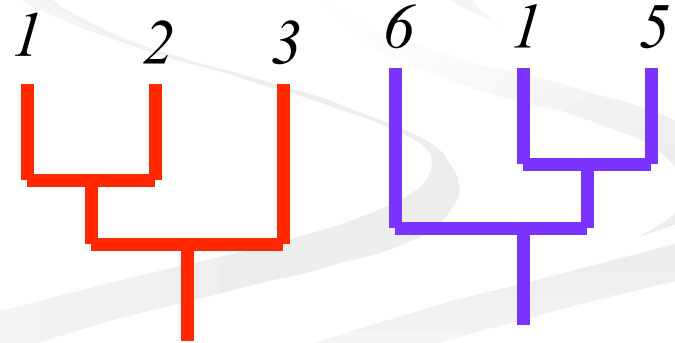
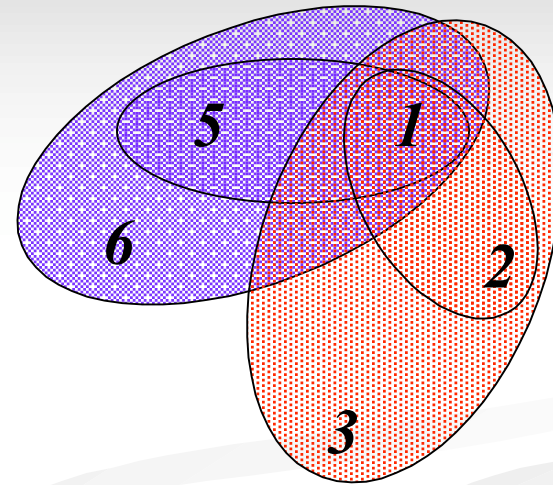
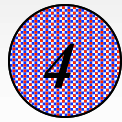
- ❑ Step one: Divide the taxa into two or more 'super-clades'. Put two taxa in the same clade if they are in the same clade of one of the input trees.
- ❑ Step two: Compute the restriction of the input trees to each super-clade



# BUILD

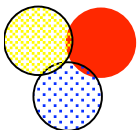


# BUILD



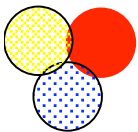
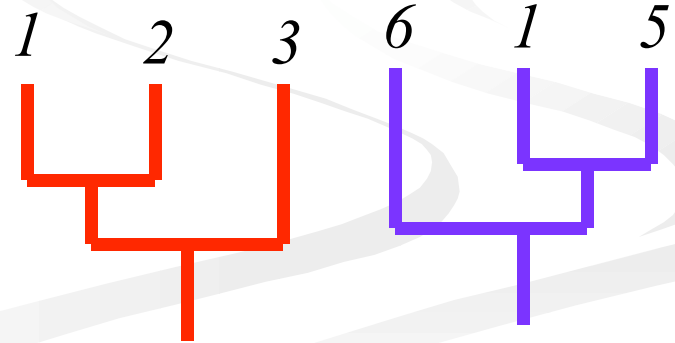
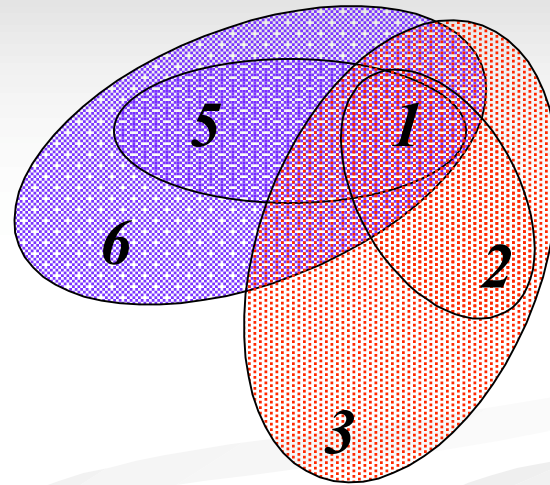
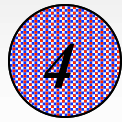
# BUILD

- ❑ Step one: Divide the taxa into two or more 'super-clades'. Put two taxa in the same clade if they are in the same clade of one of the input trees.
- ❑ Step two: Compute the restriction of the input trees to each super-clade
- ❑ Step three: Apply the algorithm to each sub-problem (if there is only one taxa, don't bother)

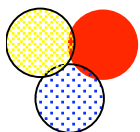
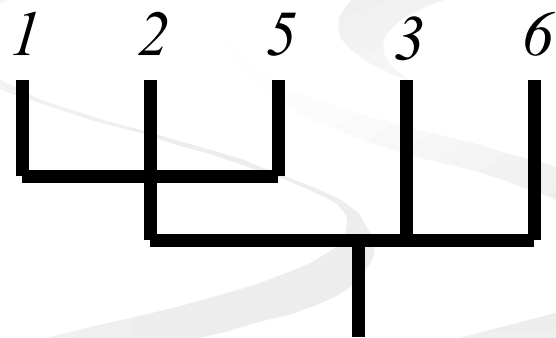
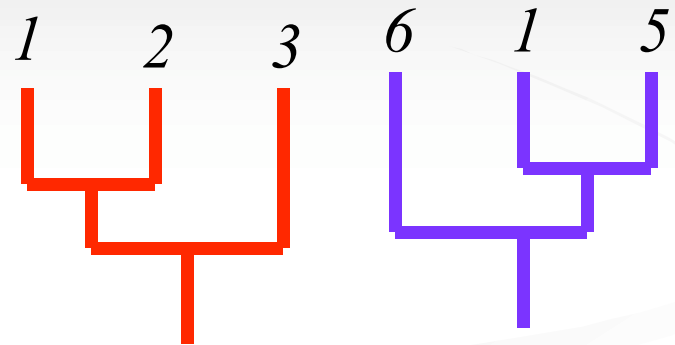




# BUILD

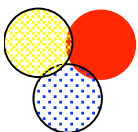


# BUILD



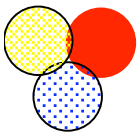
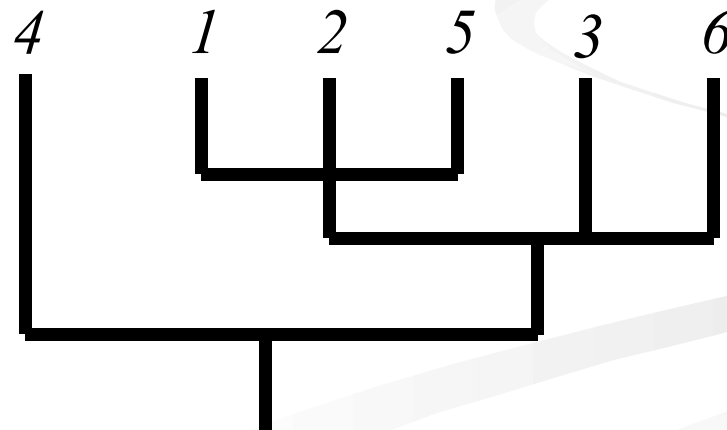
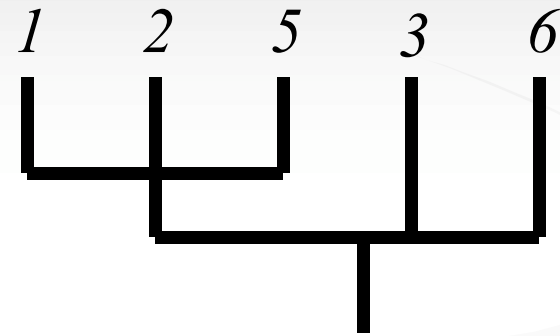

# BUILD

- ❑ Step one: Divide the taxa into two or more 'super-clades'. Put two taxa in the same clade if they are in the same clade of one of the input trees.
- ❑ Step two: Compute the restriction of the input trees to each super-clade
- ❑ Step three: Apply the algorithm to each sub-problem (if there is only one taxa, don't bother)
- ❑ Step four: Combine the trees returned from the sub-problems and return the tree you obtain

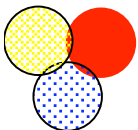
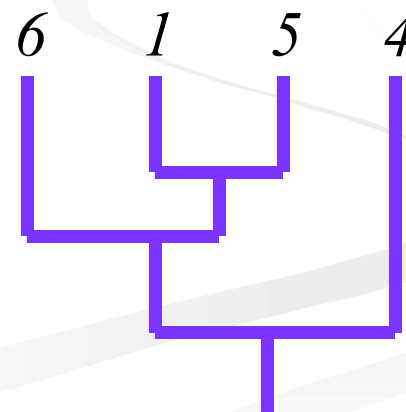
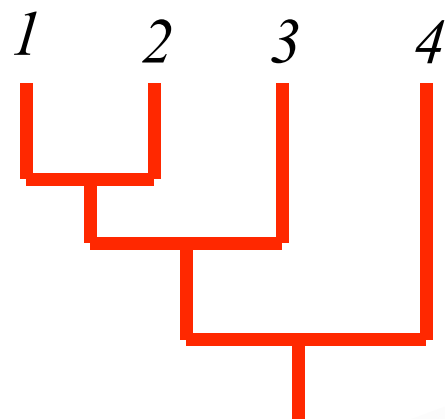
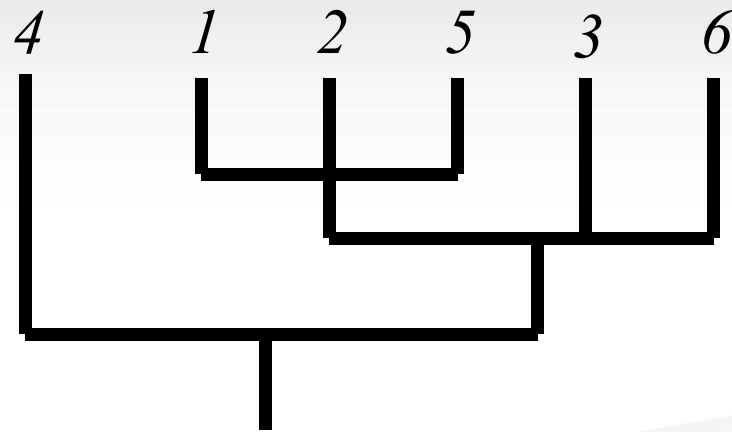


# BUILD

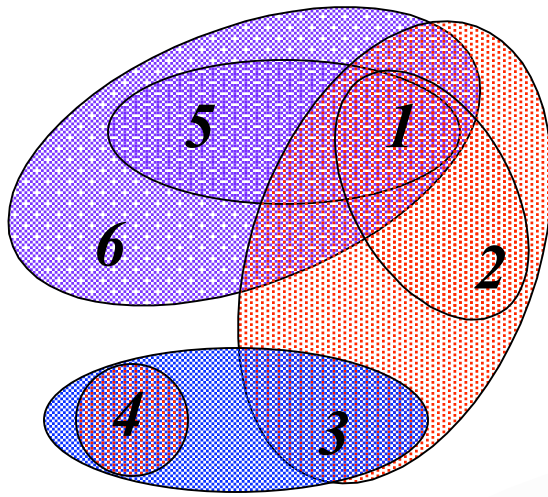
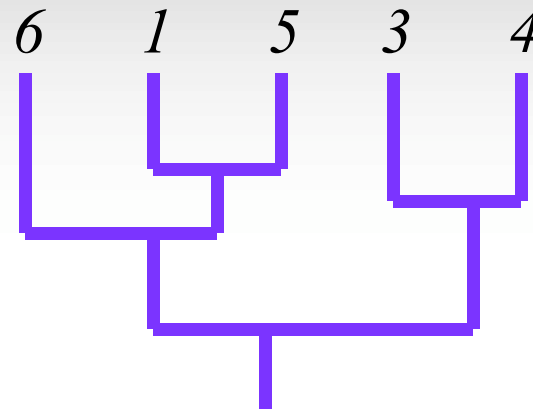
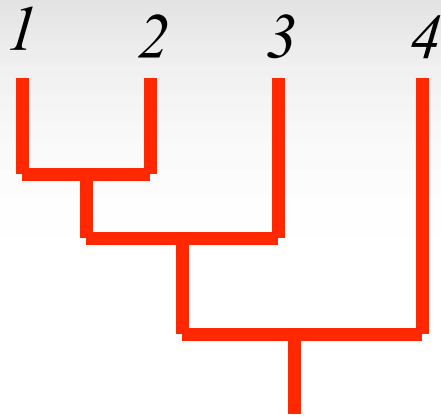
4



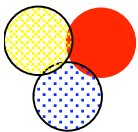
# BUILD



# BUILD

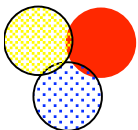


If you can't split the taxa, the trees are incompatible.

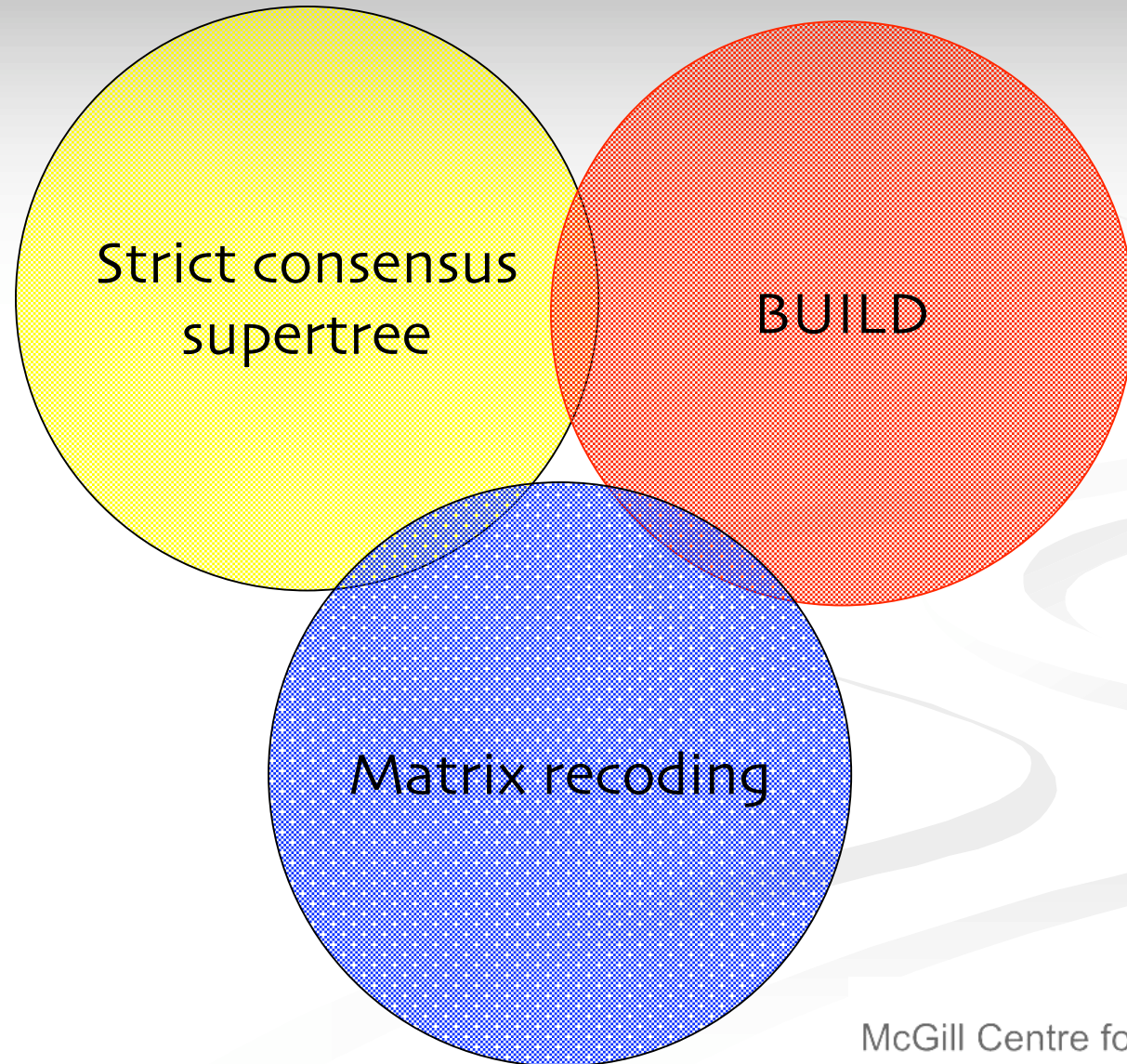


# Incompatible build...

- Thorley and Wilkinson (1998) suggest removing taxa until they are compatible
- Semple and Steel (2000) use a minimum cut algorithm to divide the taxa even when they are all linked up
- Page (2002) proposed a modification of Semple and Steel's algorithm
- Bryant, Semple, and Steel (2003) show how to incorporate ancestral divergence dates



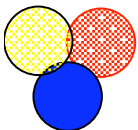
# The Three Supertree Methods



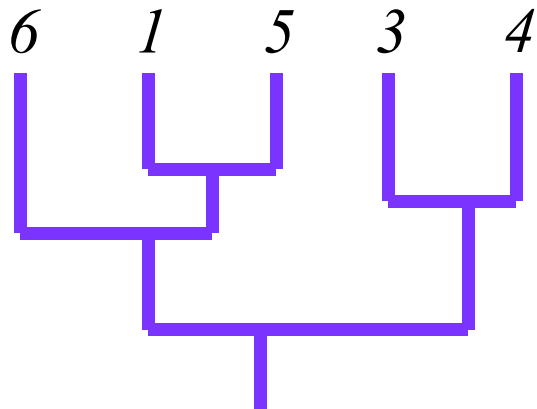
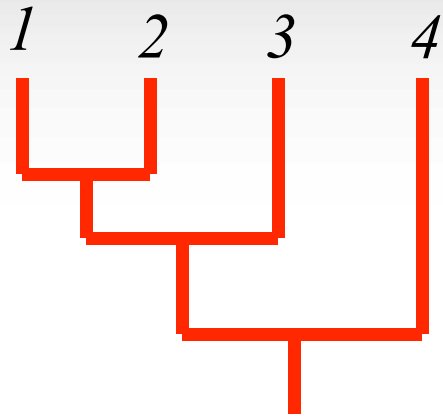


# MRP (Matrix representation with Parsimony)

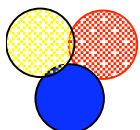
- Step one: Code each tree as a collection of binary characters, with one character for each clade and missing entries for leaves not in the tree.



# MRP (Matrix representation with Parsimony)

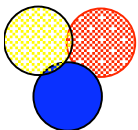


1	1	1	1	1	0
2	1	1	?	?	?
3	0	1	0	0	1
4	0	0	0	0	1
5	?	?	1	1	0
6	?	?	0	1	0
X	0	0	0	0	0



# MRP (Matrix representation with Parsimony)

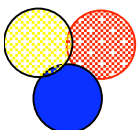
- ❑ Step one: Code each tree as a collection of binary characters, with one character for each clade and missing entries for leaves not in the tree.
- ❑ Step two: Use software to search for maximum parsimonious trees. If there are multiple optima, take the consensus.





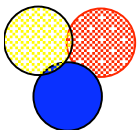
# MRP (Matrix representation with Parsimony)

- ❑ Step one: Code each tree as a collection of binary characters, with one character for each clade and missing entries for leaves not in the tree.
- ❑ Step two: Use software to search for maximum parsimonious trees. If there are multiple optima, take the consensus.
- ❑ There are many variants, depending on
  1. The exact criterion used
  2. How multiple optima are handled



# Theme and variations...

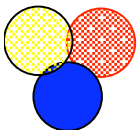
- Many different weighting schemes etc.
- Rodrigo: use max clique
- Lapointe and Cucumel: convert trees to distances then use least squares
- Chen et al. (2003): convert distances to binary matrices then define a minimum “flip” supertree: how many entries must be changed to give a perfect phylogeny.
- The possibilities are endless...



# Make your own Supertree Method



1. Choose your favourite tree construction method
2. Convert your input trees into the appropriate data format
3. Apply your method and return the tree(s)



# An introduction to Supertrees

1. What is a supertree method?
2. Why supertrees?
3. A taste of supertree mathematics
4. A tour of supertree methods
5. **Reservations**



# But...

- Biological inference, or combinatorial trickery?
- Do big (but composite) trees give us a false sense of security?
- We may need bigger trees for NSF funding, but do we need them for phylogenetic applications?
- And should we be combining trees in the first place?

## THE END

# Supertree reviews

- Bininda-Emonds, O.R.P., J.L. Gittleman, and M.A. Steel. 2002. The (super)tree of life: procedures, problems, and prospects. *Annual Reviews of Ecology and Systematics* 33: 265-289.
- Felsenstein, J. 2003 *Inferring phylogenies*. Sinaeur Press
- Janowitz, M.F., F.-J. Lapointe, F.R. McMorris, B. Mirkin, and F.S. Roberts, eds *Bioconsensus*. DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, volume 61. American Mathematical Society-DIMACS, Providence, Rhode Island.
- Semple, C. and Steel, M. *Phylogenetics* Oxford University Press