Seeing the wood for the trees: Analysing multiple alternative phylogenies

Tom M. W. Nye, Newcastle University
tom.nye@ncl.ac.uk

Isaac Newton Institute, 17 December 2007

Multiple alternative phylogenies

Phylogenetic analysis often produces many possible trees

- Variability in data / uncertainty in inferred trees:
  - ML bootstrap trees
  - Bayesian posterior samples
- Different trees for different genes

How can we summarize / represent this information?

How can we compare different alternative tree topologies?

Multiple alternative phylogenies

Phylogenetic analysis often produces many possible trees

- Variability in data / uncertainty in inferred trees:
  - ML bootstrap trees
  - Bayesian posterior samples
- Different trees for different genes

How can we summarize / represent this information?

How can we compare different alternative tree topologies?

Representing collections of trees

Existing approaches include...

- Consensus trees
- Consensus networks

But also...

- Multi-dimensional scaling (Hillis *et al*, Systematic Biology 2005)
- Clustering (Stockham *et al*, Bioinformatics 2002)

Representing collections of trees

Existing approaches include...

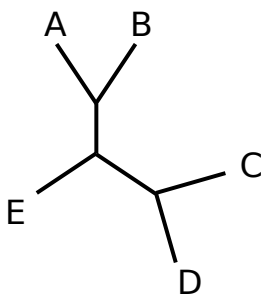
- Consensus trees
- Consensus networks

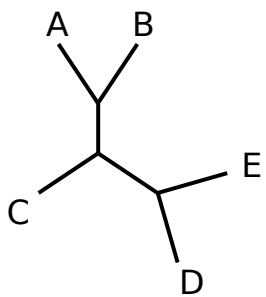
But also...

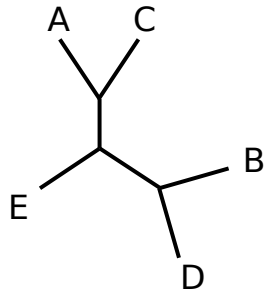
- Multi-dimensional scaling (Hillis *et al*, Systematic Biology 2005)
- Clustering (Stockham *et al*, Bioinformatics 2002)

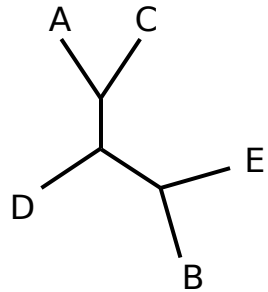
Toy example

How are these trees related?



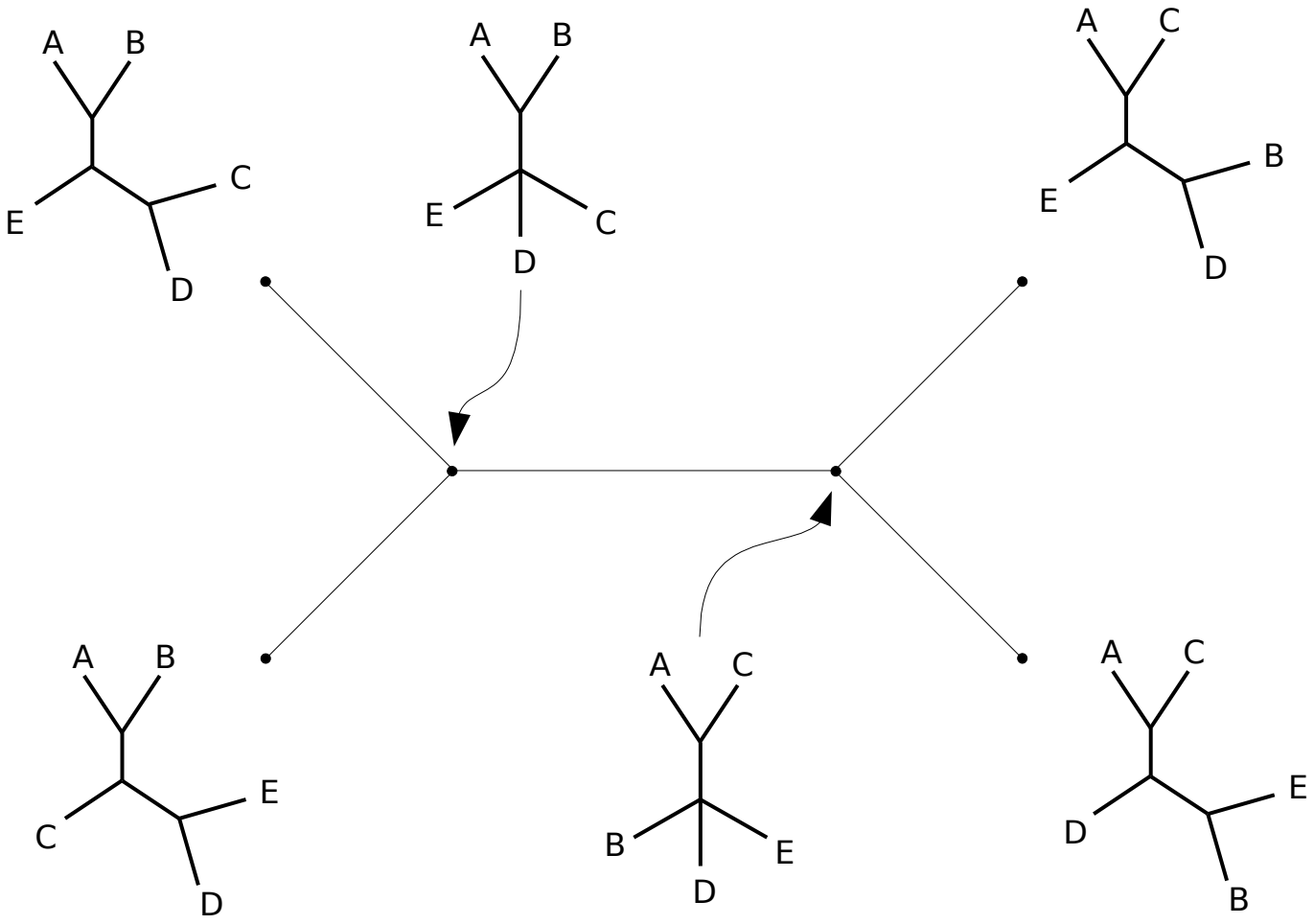






Toy example

Relate trees by a ‘tree of trees’:



Why use a tree of trees?

Tree-space does not have a tree-like structure

Advantages:

- Cluster similar trees together – edges represent gain/loss of topological features
- Conflicting histories show up as separate clades on the meta-tree:
  - different modes in a distribution
  - outliers
- Convenient form of visualisation

Meta-trees

Definition

Given a fixed set of trees $T_1, T_2, \ldots, T_n$ all having leaf-set L, a meta-tree $\hat{T}$ is an unrooted tree with n leaves such that

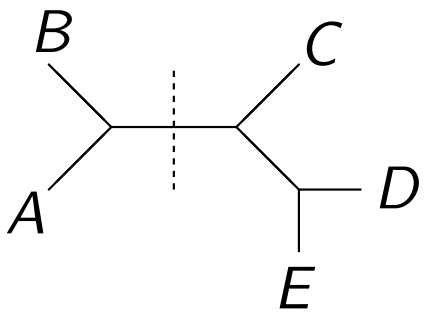
- every vertex $\hat{v}$ in $\hat{T}$ has associated to it a species tree $T_{\hat{v}}$ with leaf set L, and
- the leaf vertices of $\hat{T}$ are associated to the trees $T_1, \ldots, T_n$

Aim to find meta-trees with minimum score

- Tree score = sum of edge scores
- score $(\hat{e}) = d(T_{\hat{v}_1}, T_{\hat{v}_2})$ for an edge $\hat{e}$ between vertices $\hat{v}_1, \hat{v}_2$
- Different metrics $d(\cdot, \cdot)$ are available

Splits

A split is a bi-partition of the leaves L induced by cutting a branch:



$AB|CDE$

Trees consist of sets of compatible splits

e.g. $AB|CDE$ and $AC|BDE$ cannot both be in a tree

The majority consensus of $T_1, \ldots, T_n$ is the tree consisting of all splits in strictly greater than $n/2$ trees

The Robinson-Foulds metric is defined by:

$$d(T_a, T_b) = (\text{number of splits in } T_a \setminus T_b) + (\text{number of splits in } T_b \setminus T_a)$$

Analogy with parsimony for DNA trees

- Suppose we use the Robinson-Foulds metric
- Represent tree topologies by strings of 0's and 1's for presence / absence of splits
- Could we apply DNA parsimony algorithms to these strings to build an optimal meta-tree?

  No because strings must always represent trees
- General problem: replaced set of characters $\{A, C, G, T\}$ with the set of trees with leaf-set L
- Meta-tree construction equivalent to Steiner tree problem

Analogy with parsimony for DNA trees

- Suppose we use the Robinson-Foulds metric
- Represent tree topologies by strings of 0's and 1's for presence / absence of splits
- Could we apply DNA parsimony algorithms to these strings to build an optimal meta-tree?

  No because strings must always represent trees
- General problem: replaced set of characters $\{A, C, G, T\}$ with the set of trees with leaf-set L
- Meta-tree construction equivalent to Steiner tree problem

Analogy with parsimony for DNA trees

- Suppose we use the Robinson-Foulds metric
- Represent tree topologies by strings of 0's and 1's for presence / absence of splits
- Could we apply DNA parsimony algorithms to these strings to build an optimal meta-tree?

  No because strings must always represent trees
- General problem: replaced set of characters $\{A, C, G, T\}$ with the set of trees with leaf-set L
- Meta-tree construction equivalent to Steiner tree problem

Analogy with parsimony for DNA trees

- Suppose we use the Robinson-Foulds metric
- Represent tree topologies by strings of 0's and 1's for presence / absence of splits
- Could we apply DNA parsimony algorithms to these strings to build an optimal meta-tree?

  No because strings must always represent trees
- General problem: replaced set of characters $\{A, C, G, T\}$ with the set of trees with leaf-set L
- Meta-tree construction equivalent to Steiner tree problem

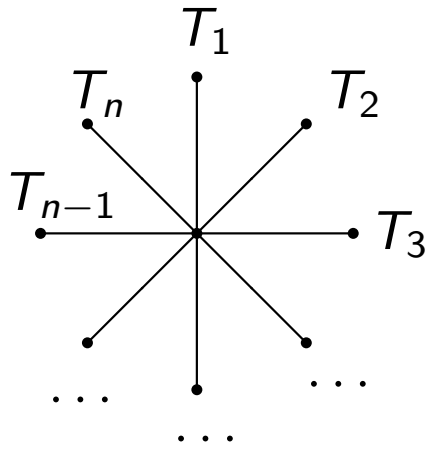
Analogy with parsimony for DNA trees

- Suppose we use the Robinson-Foulds metric
- Represent tree topologies by strings of 0's and 1's for presence / absence of splits
- Could we apply DNA parsimony algorithms to these strings to build an optimal meta-tree?

  No because strings must always represent trees
- General problem: replaced set of characters $\{A, C, G, T\}$ with the set of trees with leaf-set L
- Meta-tree construction equivalent to Steiner tree problem

Majority consensus and optimality

Consider a meta-tree with the star topology, central node T_0:



$$\text{Meta-tree score} = \sum_{\text{splits } p} \big(\text{number of edges with } p \text{ at one end but not at the other end}\big)$$

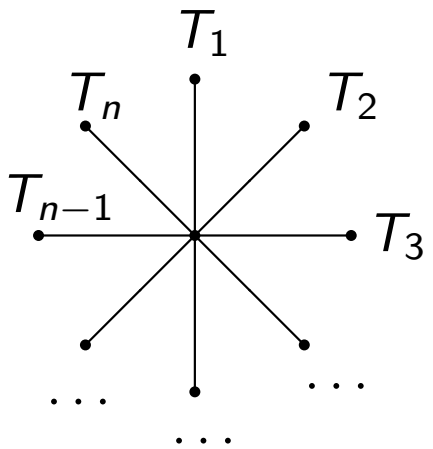
Score minimised by majority consensus:

"majority consensus is a median tree"

NB: optimisation performed for each split independently

Majority consensus and optimality

Consider a meta-tree with the star topology, central node T_0:



$$\text{Meta-tree score} = \sum_{\text{splits } p} \big(\text{number of edges with } p \text{ at one end but not at the other end}\big)$$

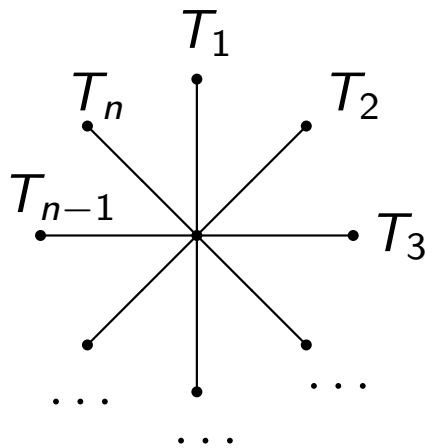
Score minimised by majority consensus:

"majority consensus is a median tree"

NB: optimisation performed for each split independently

Majority consensus and optimality

Consider a meta-tree with the star topology, central node T_0:



$$\text{Meta-tree score} = \sum_{\text{splits } p} \big(\text{number of edges with } p \text{ at one end but not at the other end}\big)$$

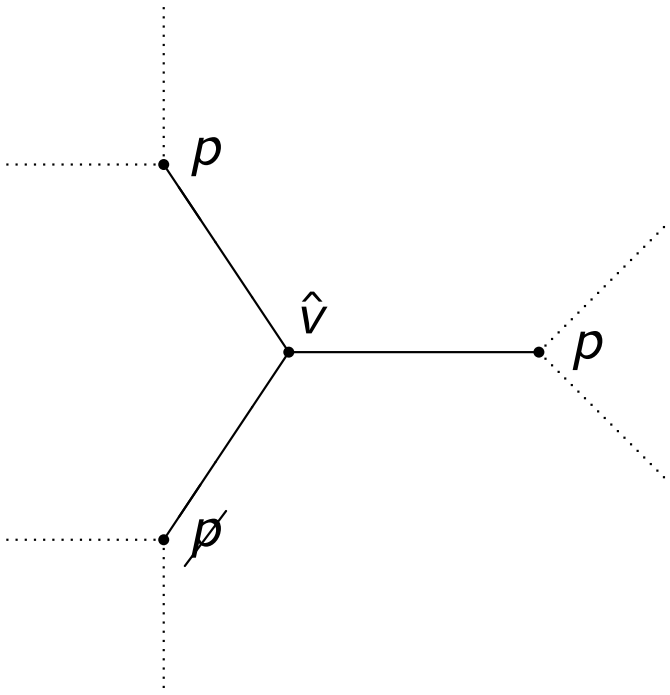
Score minimised by majority consensus:

"majority consensus is a median tree"

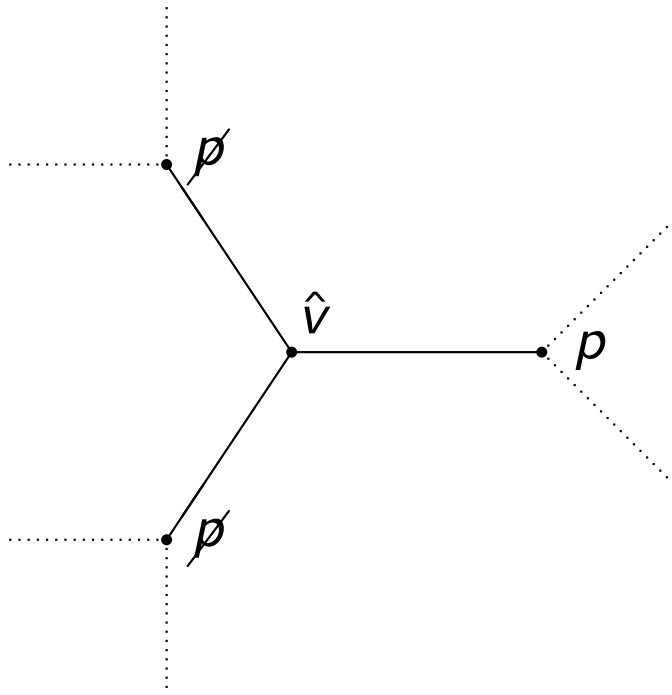
NB: optimisation performed for each split independently

A local optimality condition

Consider internal vertex $\hat{v}$ on an optimal meta-tree: when does $T_{\hat{v}}$ contain a split p?

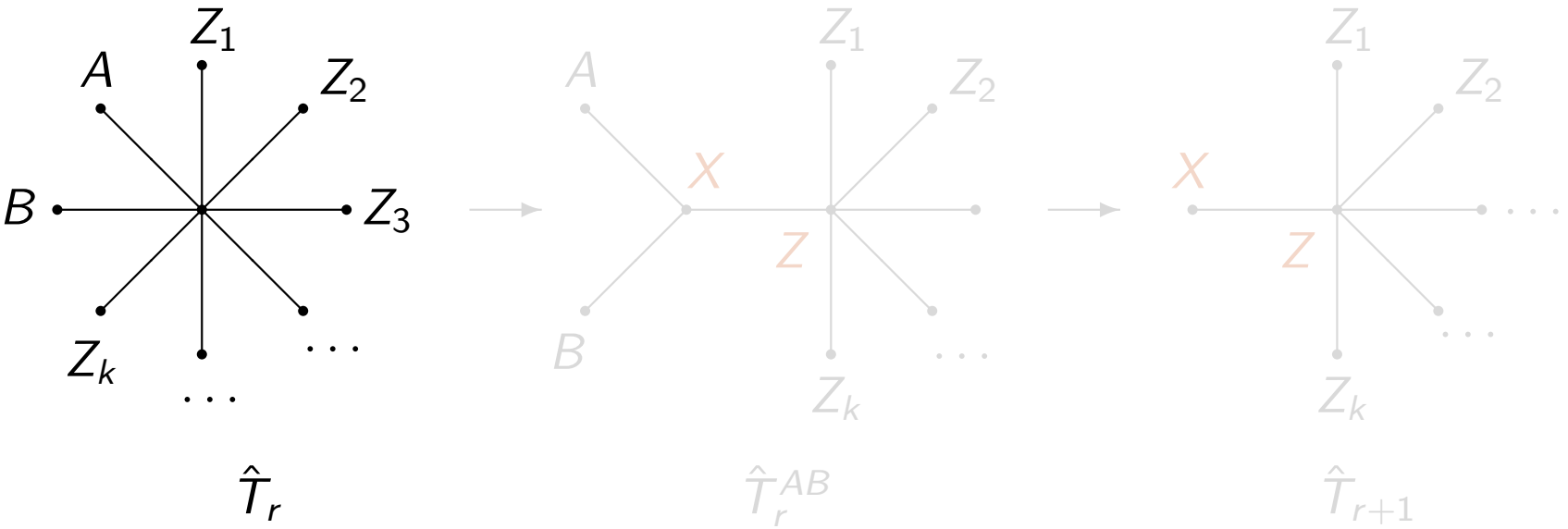


(a) Score=1 if $p \in T_{\hat{v}}$
Score=2 if $p \notin T_{\hat{v}}$



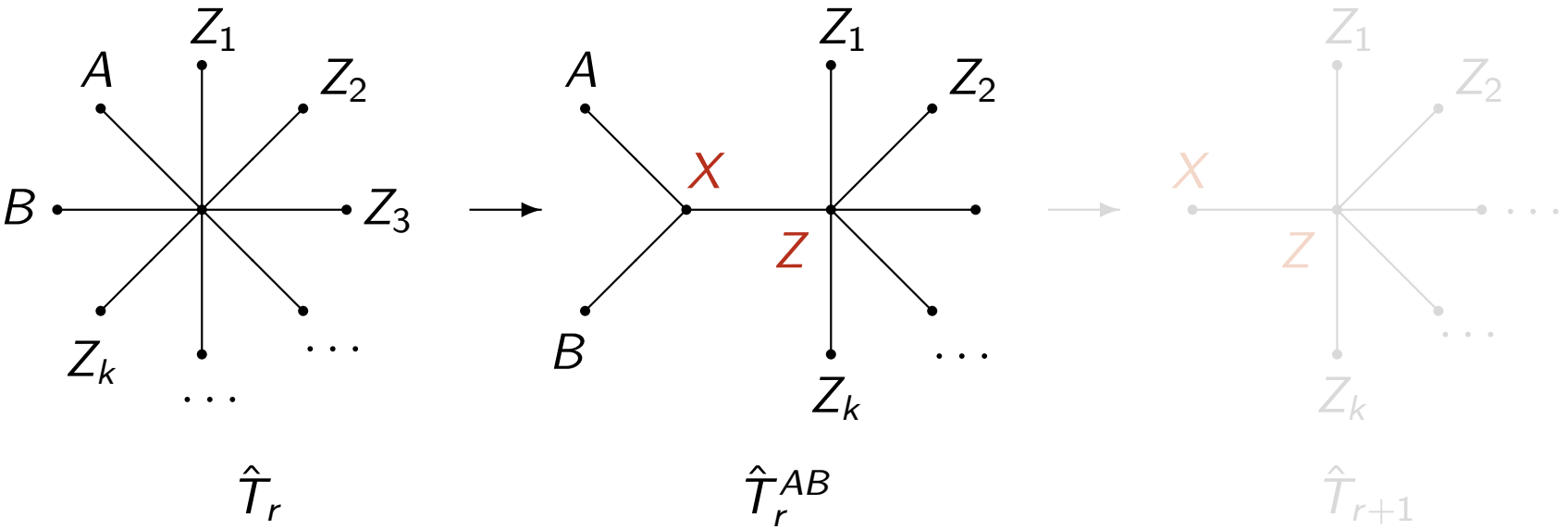
(b) Score=2 if $p \in T_{\hat{v}}$
Score=1 if $p \notin T_{\hat{v}}$

The Meta-NJ algorithm



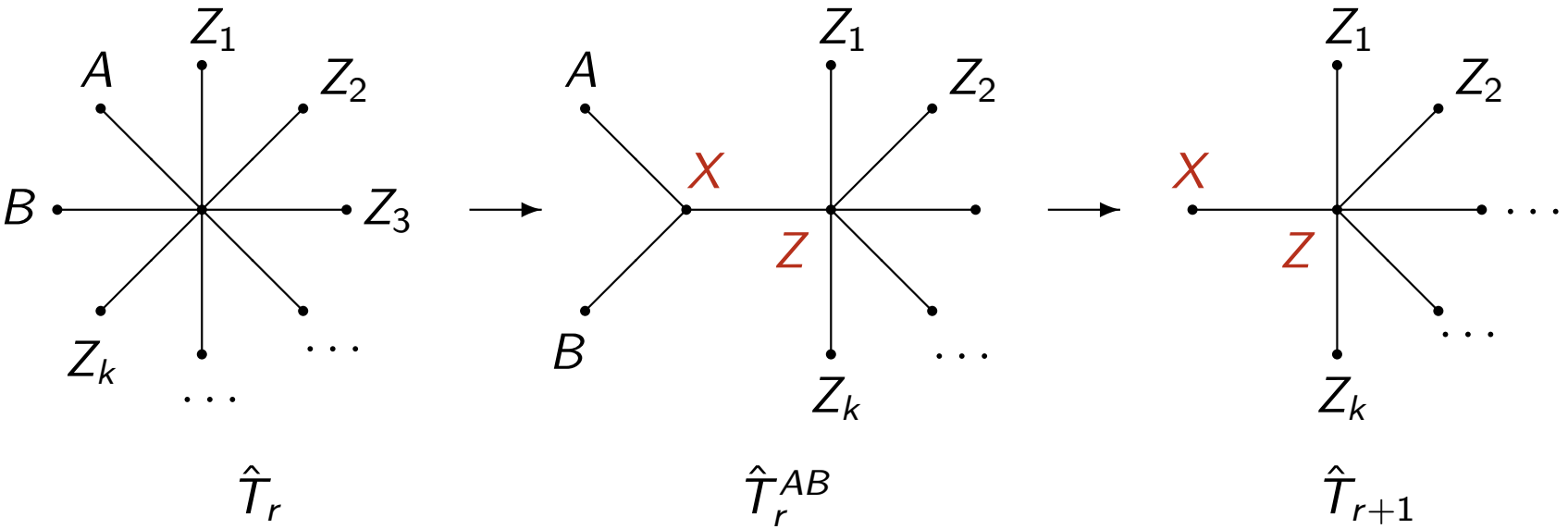
- Start with star phylogeny
- At r-th step pick two nodes A, B to agglomerate
- Form new nodes X and Z that are the majority consensus of their neighbours: $X = \text{maj}\{A, B, Z\}$ and $Z = \text{maj}\{X, Z_1, \ldots, Z_k\}$
- Calculate score for the resulting configuration $\hat{T}_r^{AB}$
- Try every pair A, B and pick the pair with min score

The Meta-NJ algorithm



- Start with star phylogeny
- At r-th step pick two nodes A, B to agglomerate
- Form new nodes X and Z that are the majority consensus of their neighbours: $X = \text{maj}\{A, B, Z\}$ and $Z = \text{maj}\{X, Z_1, \ldots, Z_k\}$
- Calculate score for the resulting configuration $\hat{T}_r^{AB}$
- Try every pair A, B and pick the pair with min score

The Meta-NJ algorithm



- Start with star phylogeny
- At r-th step pick two nodes A, B to agglomerate
- Form new nodes X and Z that are the majority consensus of their neighbours: $X = \text{maj}\{A, B, Z\}$ and $Z = \text{maj}\{X, Z_1, \ldots, Z_k\}$
- Calculate score for the resulting configuration $\hat{T}_r^{AB}$
- Try every pair A, B and pick the pair with min score

Features of the algorithm

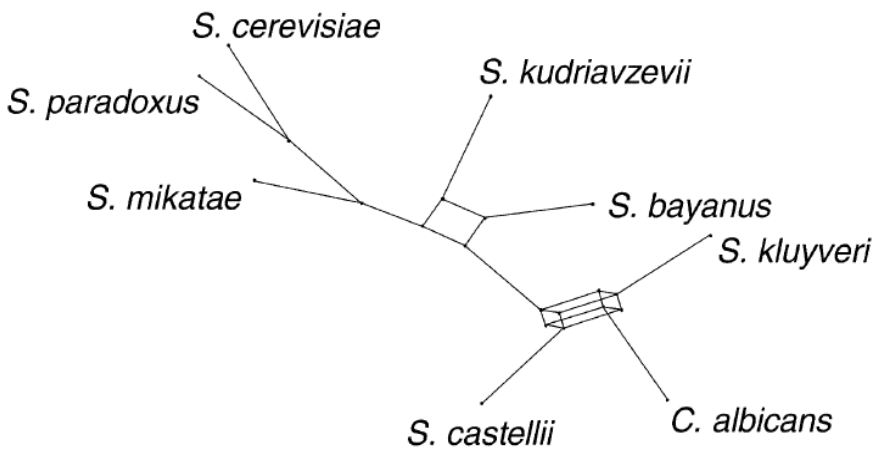
- Simultaneous equations for X and Z can be solved (almost) uniquely: splits are considered independently
- Each vertex on resulting meta-tree is majority consensus of its neighbours
- Algorithm greedily constructs meta-trees with the local optimality condition
- Zero length branches are sometimes produced – leads to multifurcations
- Ties in score: pick one agglomeration at random

Yeast data set

Rokas *et al*, 'Genome-scale approaches to resolving incongruence', Nature 2003:

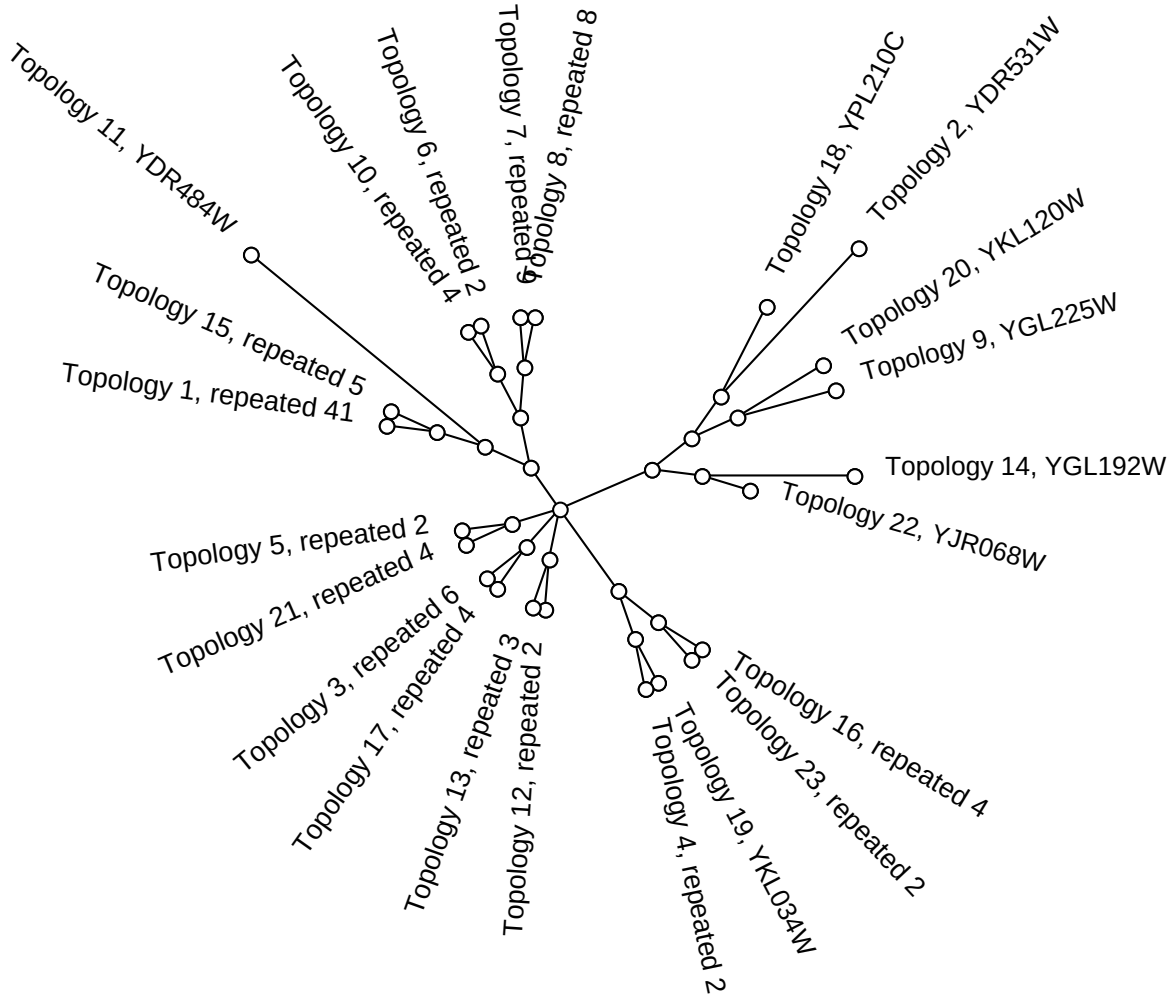
- Genomes from 8 species of yeast
- ML trees constructed for 106 orthologs
- 23 different topologies obtained

Consensus network (Holland *et al*, Mol. Biol. and Evolution 2004):

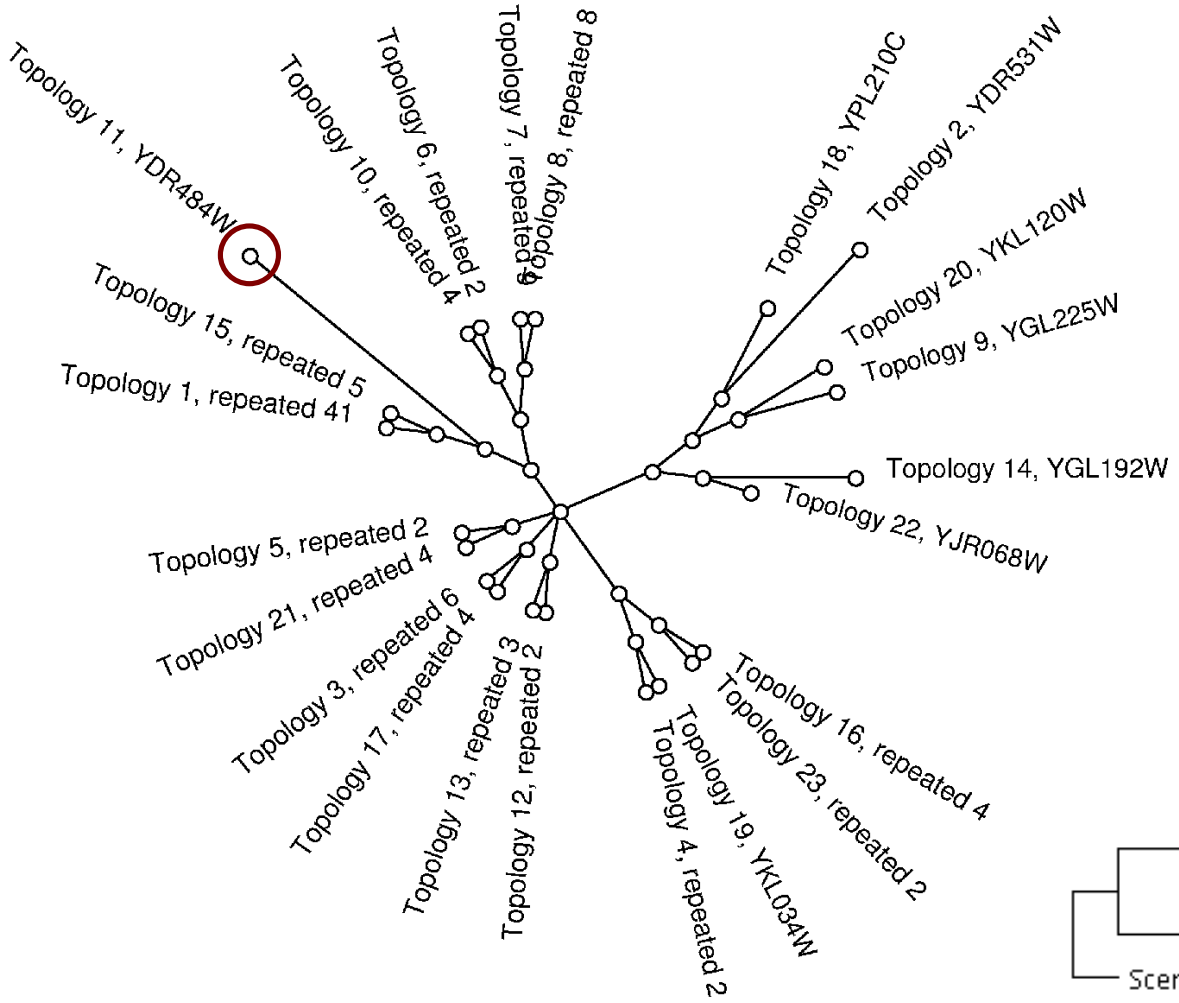


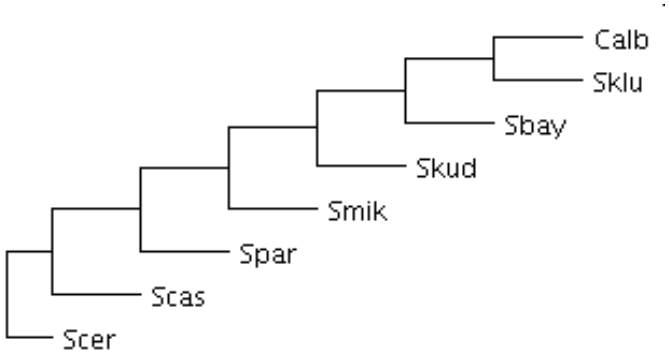
Yeast data set results

Link to web

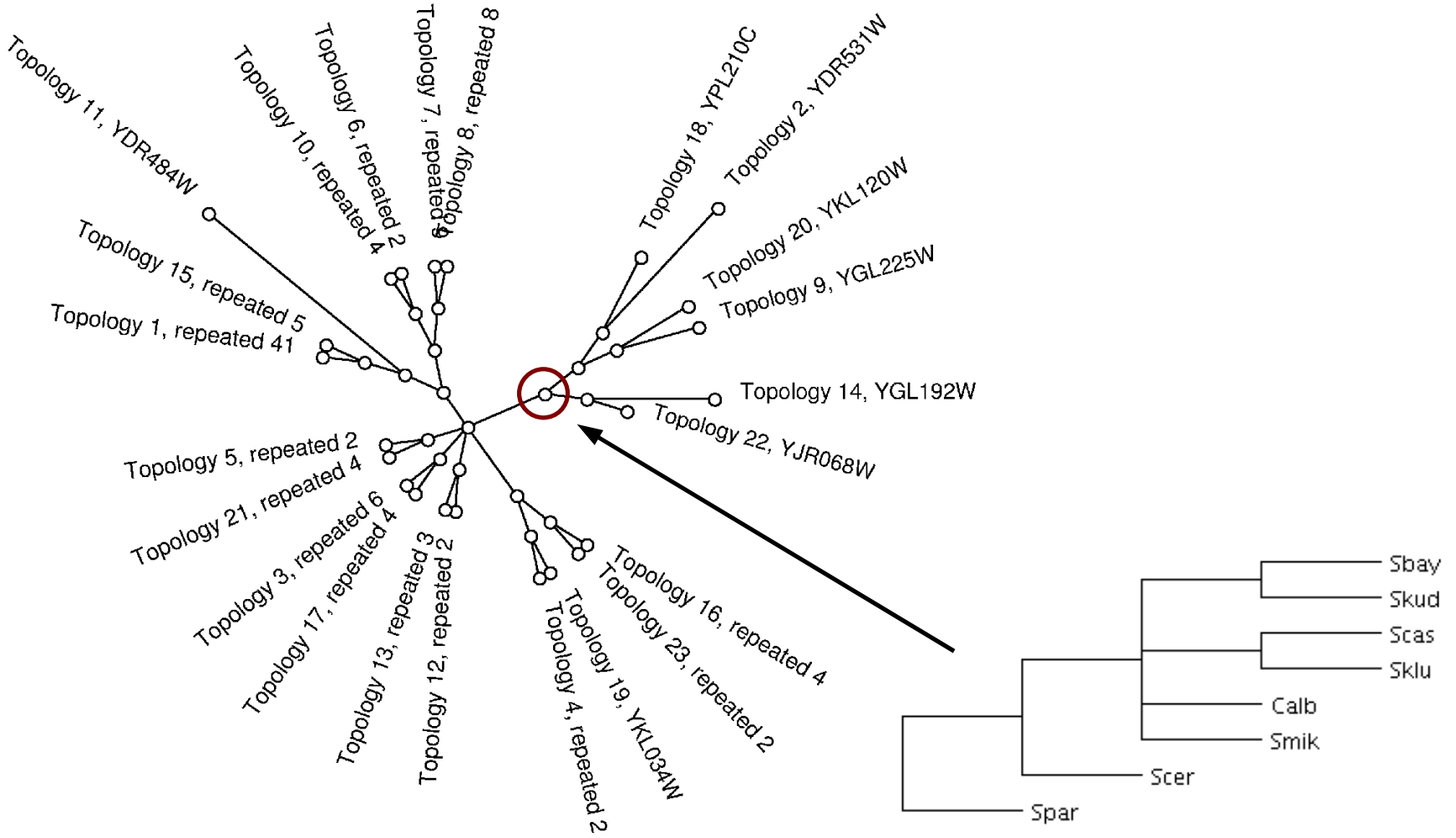


Yeast data set results

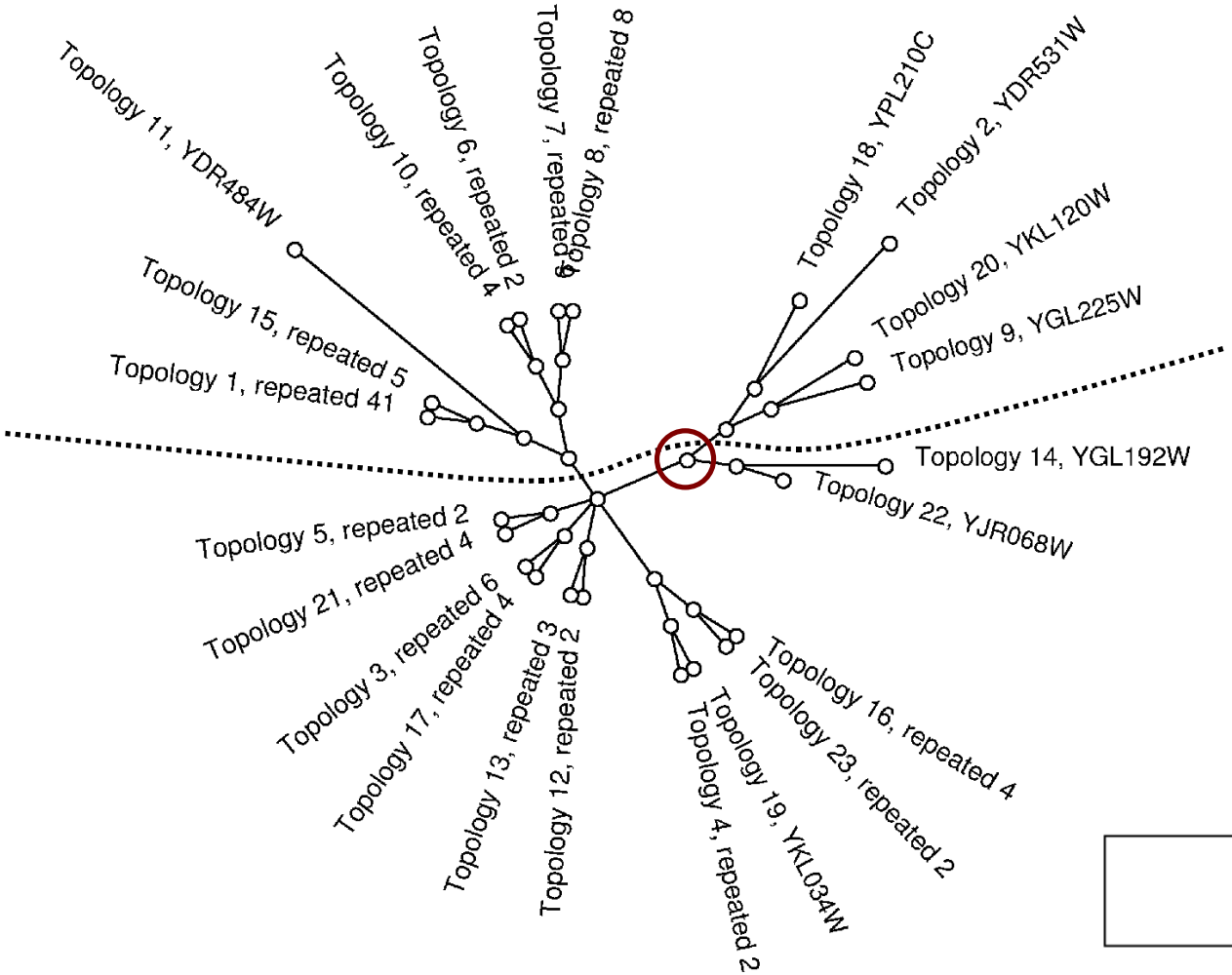




Yeast data set results

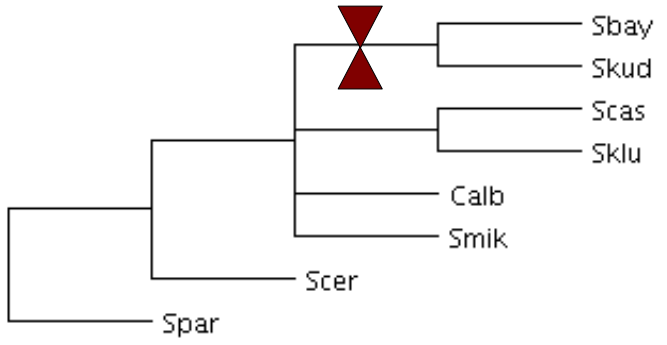


Yeast data set results

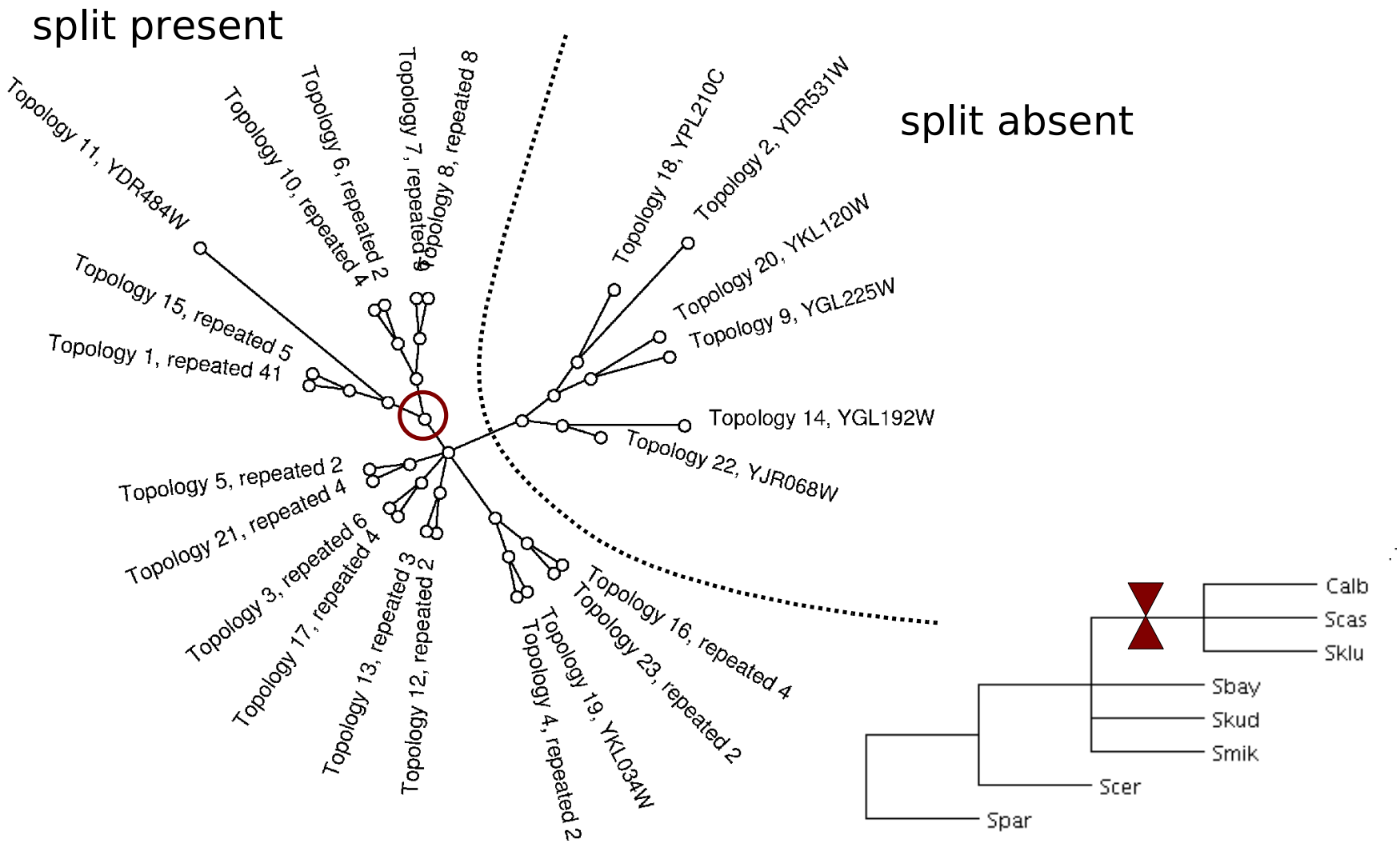


split absent

split present

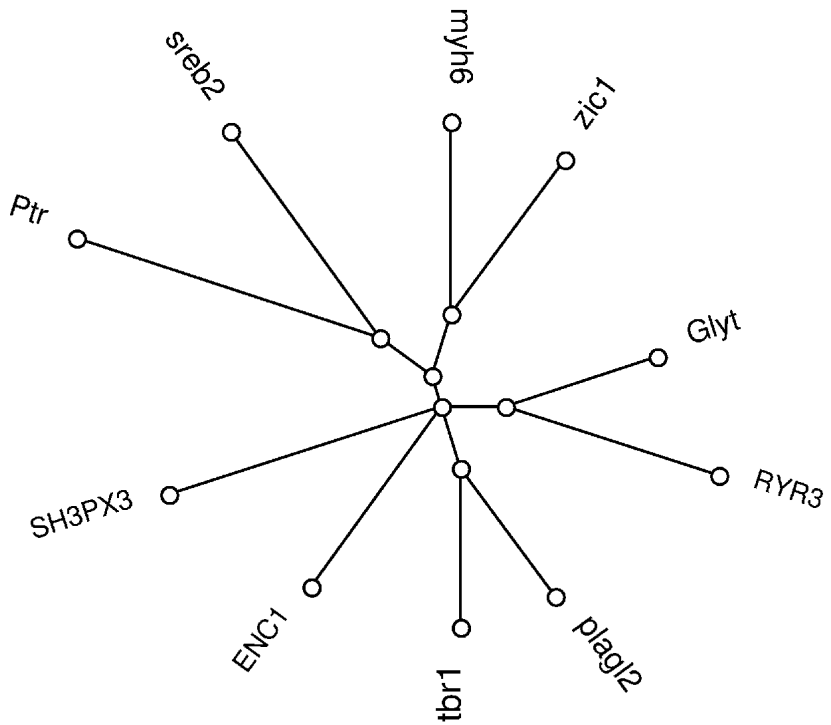


Yeast data set results



Fish data set

10 orthologous genes in 14 species of ray-finned fish
Li et al, BMC Evolutionary Biology, 2007



Fish bootstrap analysis

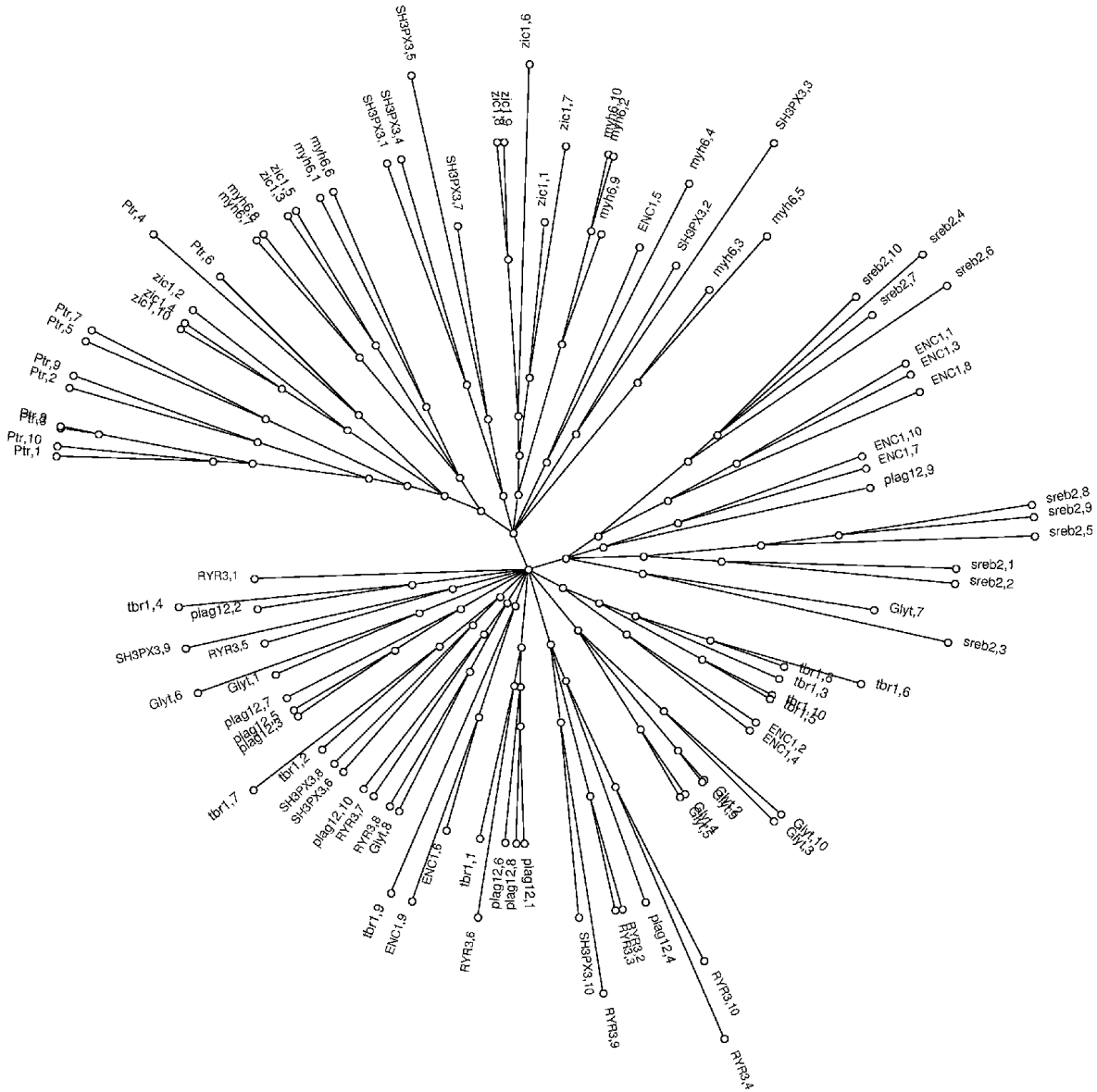
To what extent is incongruence caused by:

(a) lack of phylogenetic signal in each gene sequence, or
(b) genuine evolutionary differences?

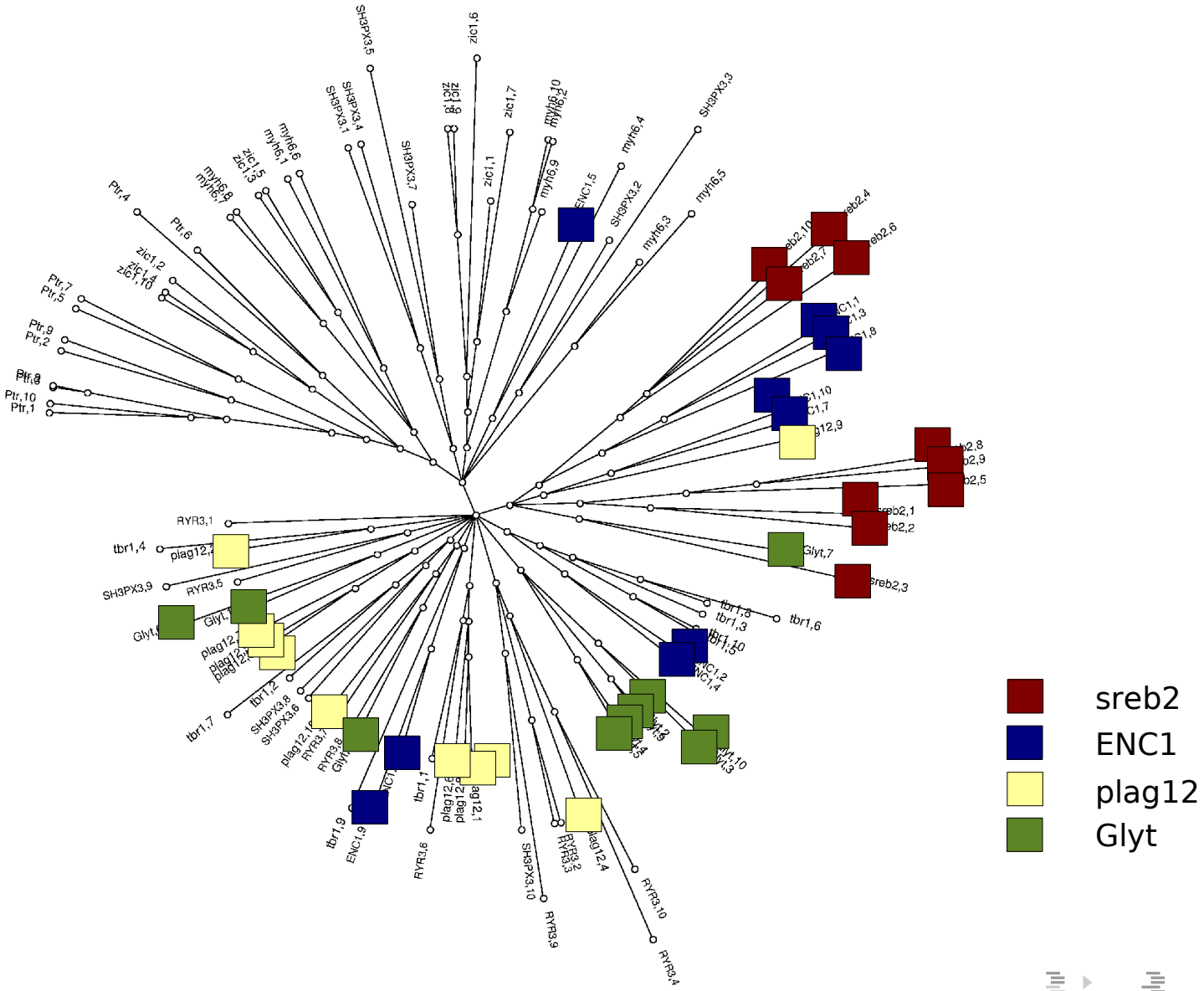
Generate 10 boostrap replicates for each gene

- Replicates for each gene form clusters $\Rightarrow$ distinct evolutionary histories
- Replicates scattered $\Rightarrow$ lack of signal in each gene

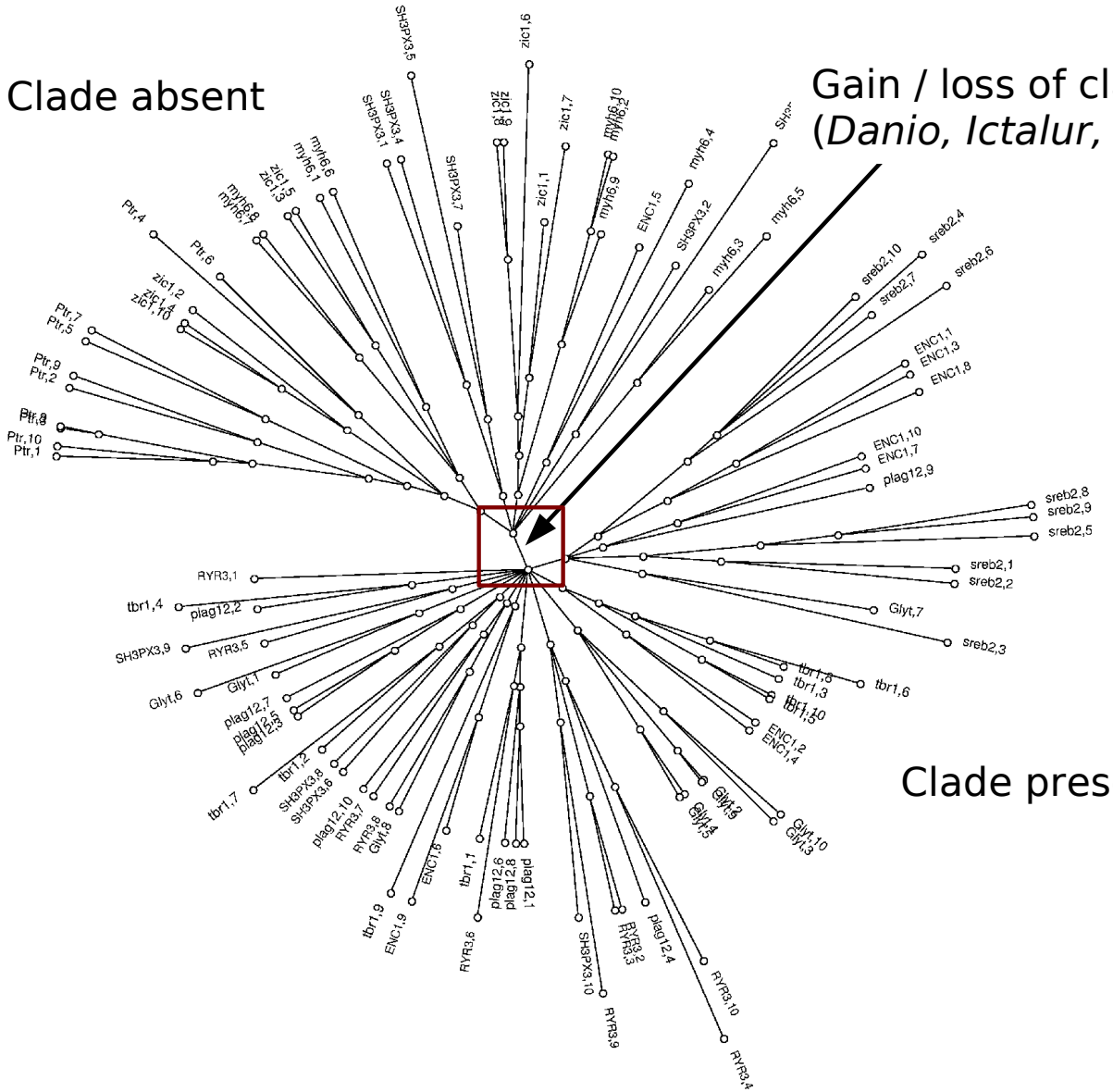
Fish bootstrap results



Fish bootstrap results



Fish bootstrap results



Summary

- Attempt to represent a collection of trees by a tree-of-trees or meta-tree
- Finding an optimal meta-tree is computationally hard
- Meta-NJ algorithm: heuristic approach that builds meta-trees by maintaining a local optimality condition:

  each vertex is the majority consensus of its neighbours
- Examples show typical insights meta-trees can provide

Summary

- Attempt to represent a collection of trees by a tree-of-trees or meta-tree
- Finding an optimal meta-tree is computationally hard
- Meta-NJ algorithm: heuristic approach that builds meta-trees by maintaining a local optimality condition:

  each vertex is the majority consensus of its neighbours
- Examples show typical insights meta-trees can provide

Summary

- Attempt to represent a collection of trees by a tree-of-trees or meta-tree
- Finding an optimal meta-tree is computationally hard
- Meta-NJ algorithm: heuristic approach that builds meta-trees by maintaining a local optimality condition:

  each vertex is the majority consensus of its neighbours

- Examples show typical insights meta-trees can provide

Summary

- Attempt to represent a collection of trees by a tree-of-trees or meta-tree
- Finding an optimal meta-tree is computationally hard
- Meta-NJ algorithm: heuristic approach that builds meta-trees by maintaining a local optimality condition:

  each vertex is the majority consensus of its neighbours

- Examples show typical insights meta-trees can provide

Acknowledgements

Thanks to

- Wally Gilks
- Antonis Rokas (yeast data set)
- Chenhong Li (fish data set)

Web site

Software is available on line at:

www.mas.ncl.ac.uk/∼ntmwn/phylo_comparison/multiple.html