# Computing a tree: III

Genome 559: Introduction to Statistical and Computational Genomics

Prof. James H. Thomas

# Parsimony trees

1) Construct all possible trees

2) For each informative site in alignment count changes on each tree

3) Add them all up for each tree

4) Pick the lowest scoring

# Distance trees

- Compute pairwise corrected distances.

- Build tree by sequential clustering algorithm (UPGMA or Neighbor-Joining).

- Don't consider all tree topologies, so they are very fast, even for large trees.

# Maximum-likelihood trees

- Tree evaluated for likelihood of data given tree.
- Uses a specific model for evolutionary rates (such as Jukes-Cantor).
- Like parsimony, must search tree space.
- Usually most accurate method but slow.

# Searching tree space

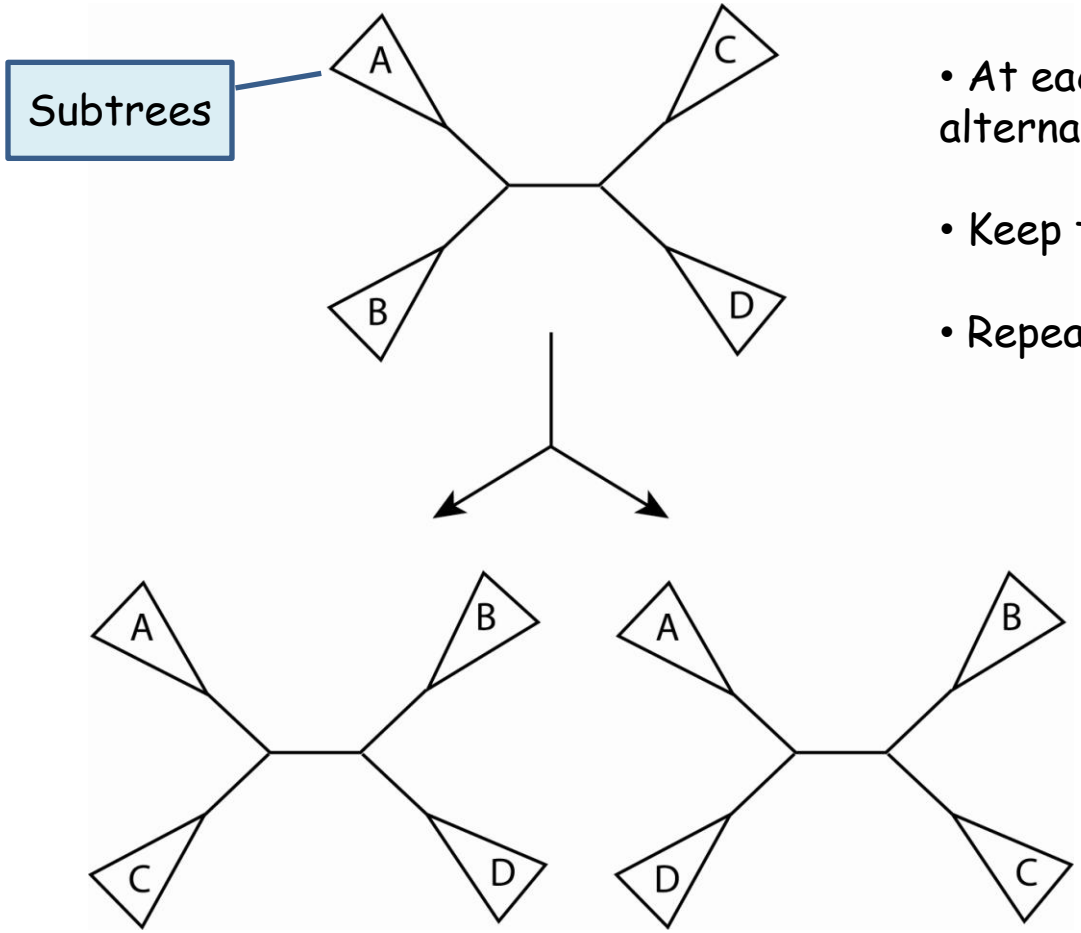
- Exhaustive search: up to 8-10 leaves, guaranteed results.

- Branch-and-bound: up to 10-20 leaves, guaranteed results *.

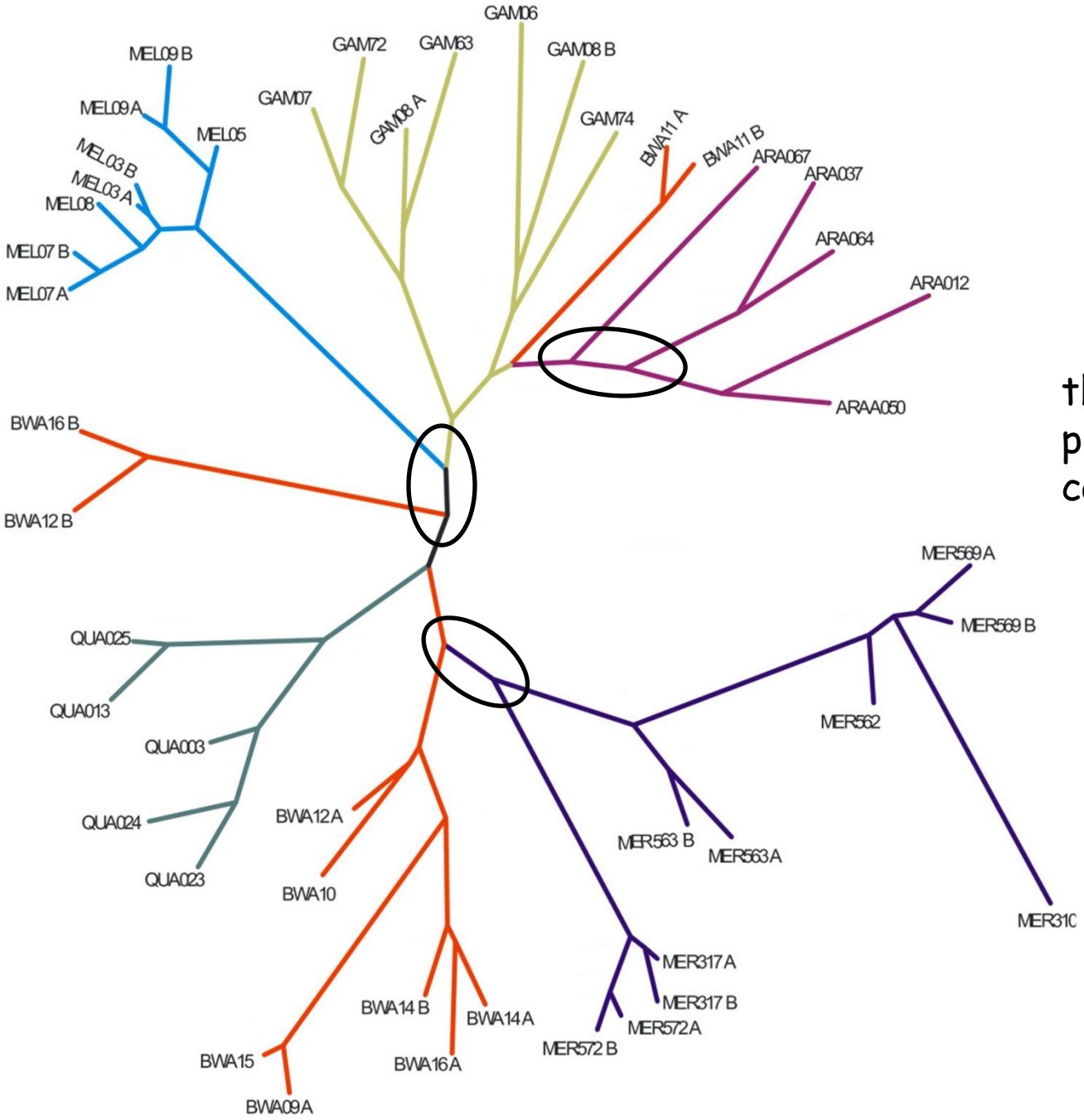
- Heuristic search: 20+ leaves, but may not find correct solution (e.g. NNI hill-climb).

* Branch-and-bound is a clever way of ruling out most trees as they are built, so you can evaluate more trees by exhaustive search.

# Nearest-Neighbor Interchange (NNI)

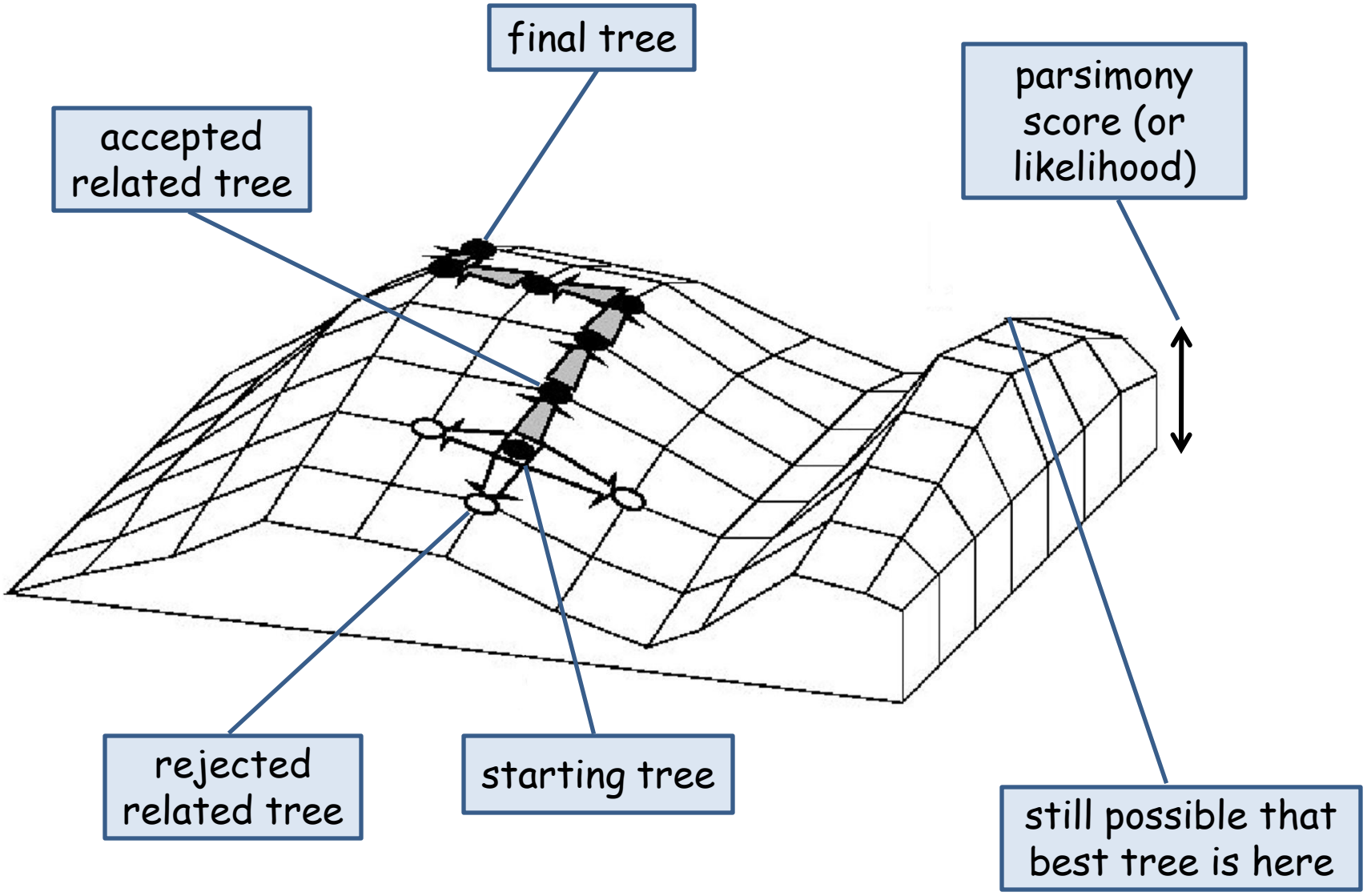



- Find a tree with some score (or likelihood).
- At each internal node consider the two alternative arrangements of the 4 subtrees.
- Keep the tree that has the best score.
- Repeat.




three (of many) places where NNI can be considered

# Hill-climbing with NNI



I got behind a bit in lectures, but for your information, I have posted additional slides on tree branch support measures.

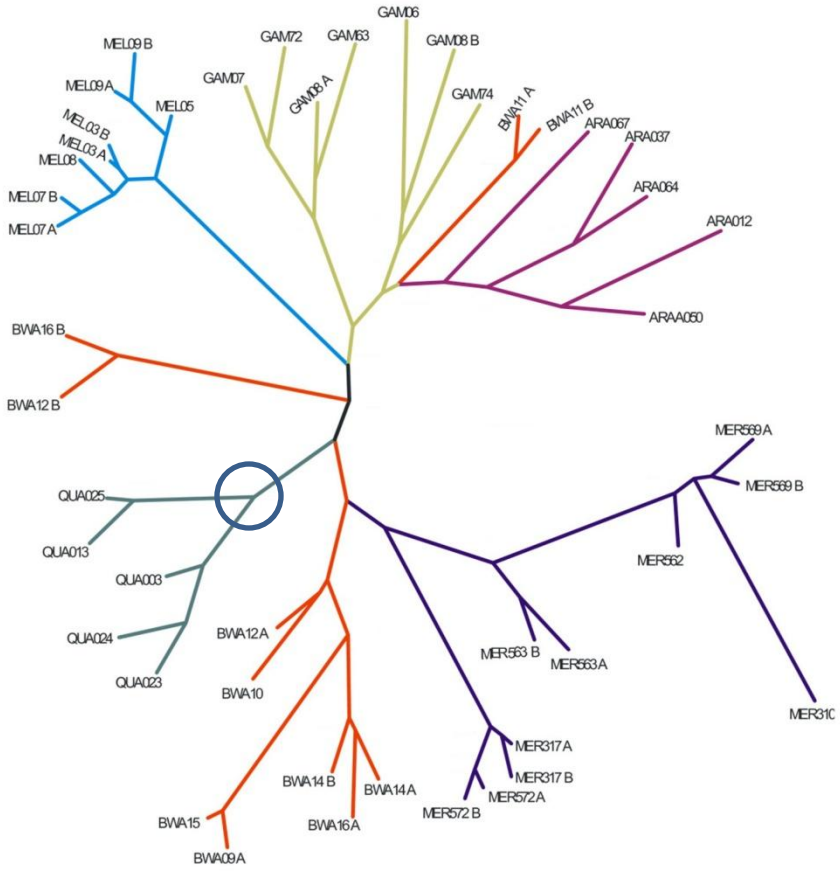
# Branch confidence

How certain are we that this is the correct tree?

Can be reduced to many simpler questions - how certain are we that each branch point is correct?

For example at the circled branch point, how certain are we that the three subtrees have the correct content:

subtree1 - QUA025, QUA013
subtree2 - QUA003, QUA024, QUA023
subtree3 - everything else




These values are often drawn on the tree as % support values.

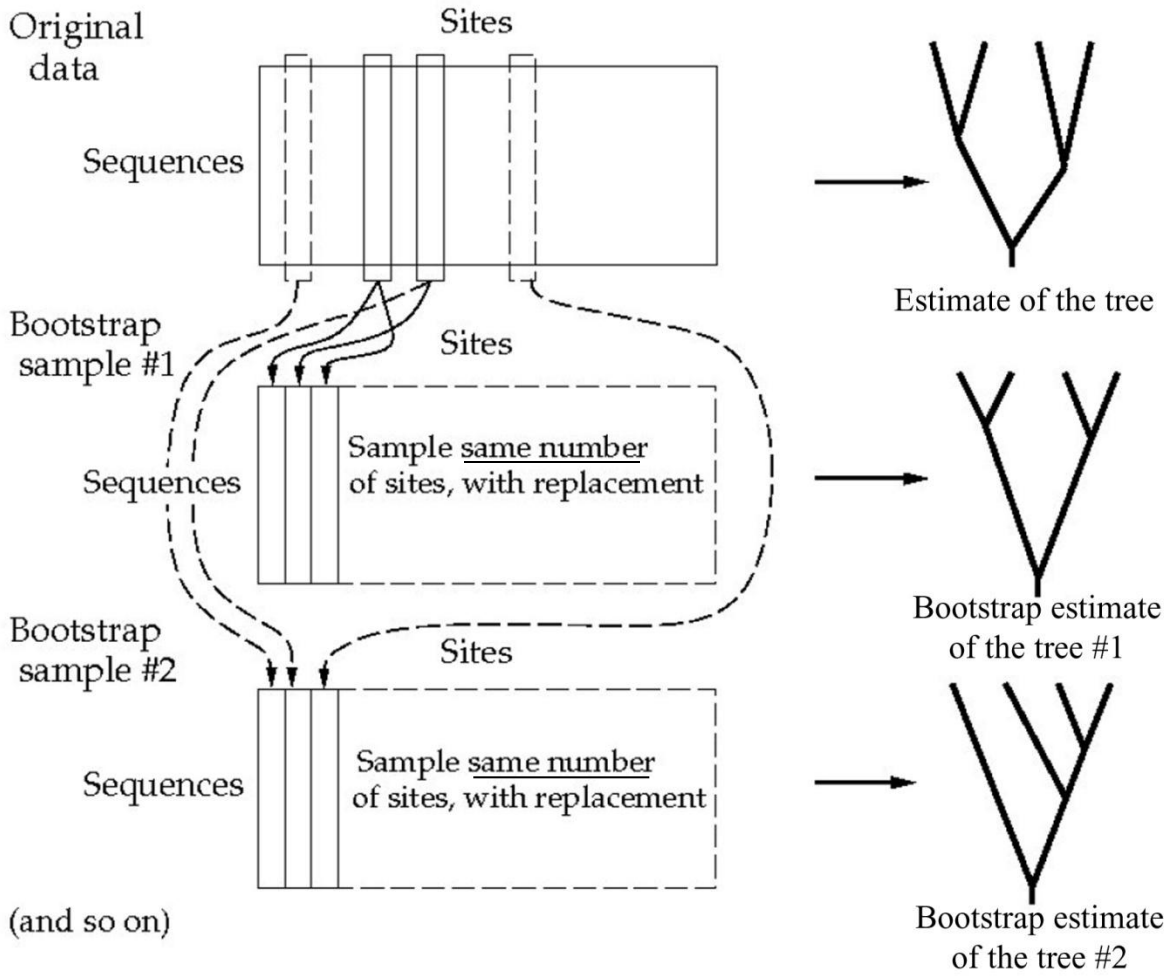
# Bootstrap support

Most commonly used branch support test.

Randomly sample alignment sites, new estimate of tree.

Repeat many times.

Tests the robustness of the tree to data perturbation.




(sample with replacement means that a sampled site remains in the source data after each sampling, so that some sites will be sampled more than one time)

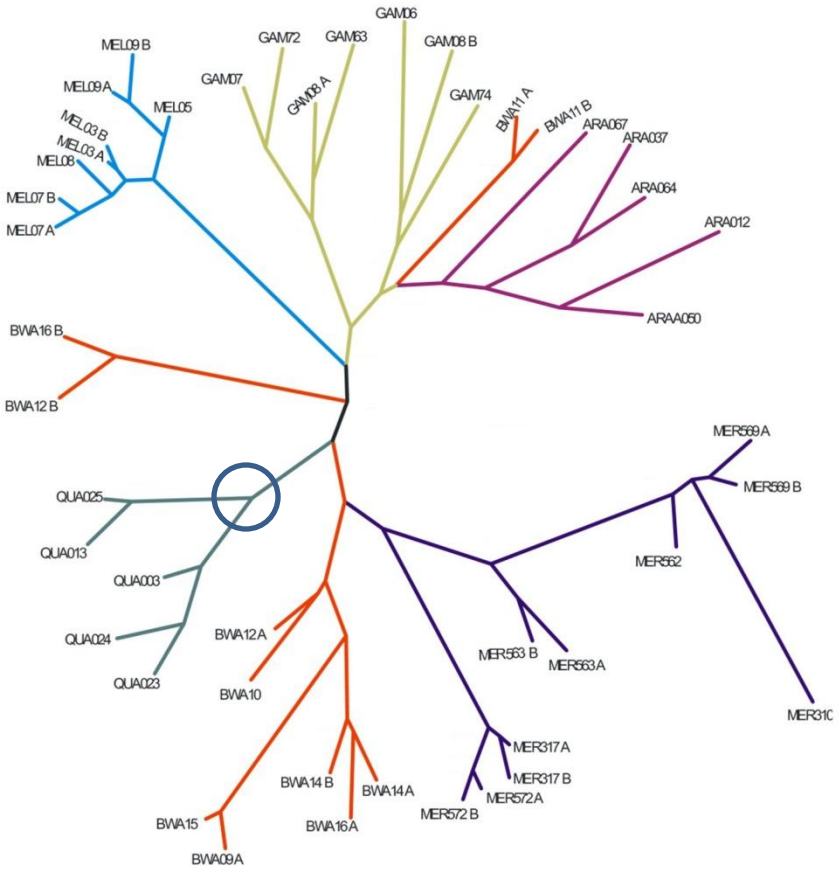
# Bootstrap support

For each branch point on the computed tree, count what fraction of the bootstrap trees have the same subtree partitions (regardless of topology within the subtree).

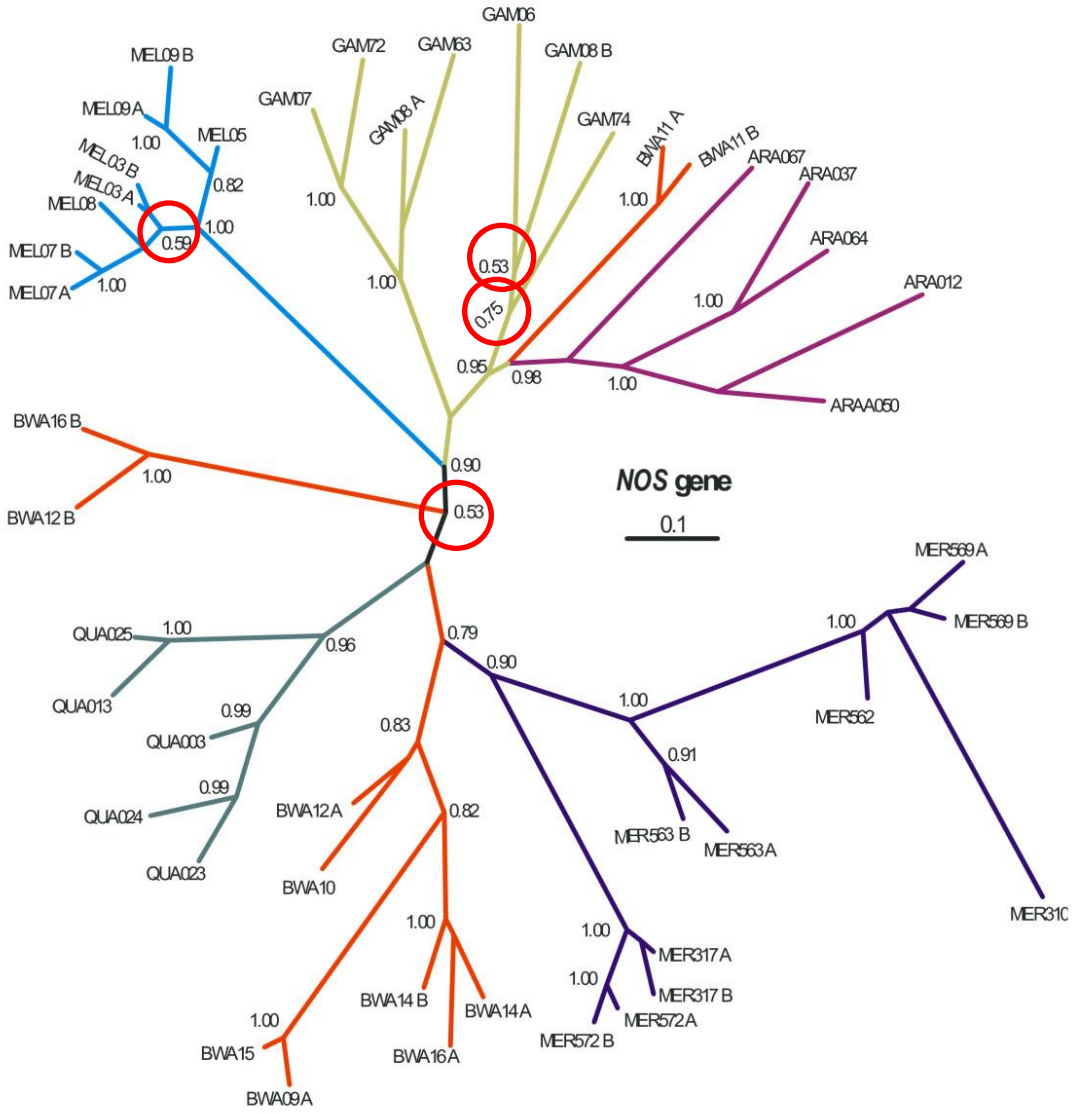
For example at the circled branch point, what fraction of the bootstrap trees have a branch point where the three subtrees include:

subtree1 - QUA025, QUA013
subtree2 - QUA003, QUA024, QUA023
subtree3 - everything else

This fraction is the bootstrap support for that branch.



# Original tree figure with branch supports (here as fractions, also common to give % support).




low-confidence branches circled by me

(for some unknown reason not all the branches are labeled with support values)

# Bootstrap support

Advantages:

- Can be applied to any tree-building method.
- Widely used and understood.

Disadvantages:

- Very slow (typically run 100 or more bootstrap trees).
- Bootstrap support is not the same as P-value (underestimates P-value at high support and overestimates at low support).
- Does not detect flaws in the tree inference method (e.g. your evolutionary model).