Introduction to Phylogenetics
Week 6

Maximum Likelihood

III. ML Methods

- Go into CIPRES gateway
- Use aligned data
- Run your data using the following tasks:
  - FastTree
  - GARLI
  - PAUP*Rat (set nchar values)
  - RAxML (set outgroup)

Rooting your tree…

- Can be used to infer ancestral states
- Can determine direction of change
- Determine common ancestors
- Your *outgroup* becomes the root of your tree
- Important in likelihood calculations

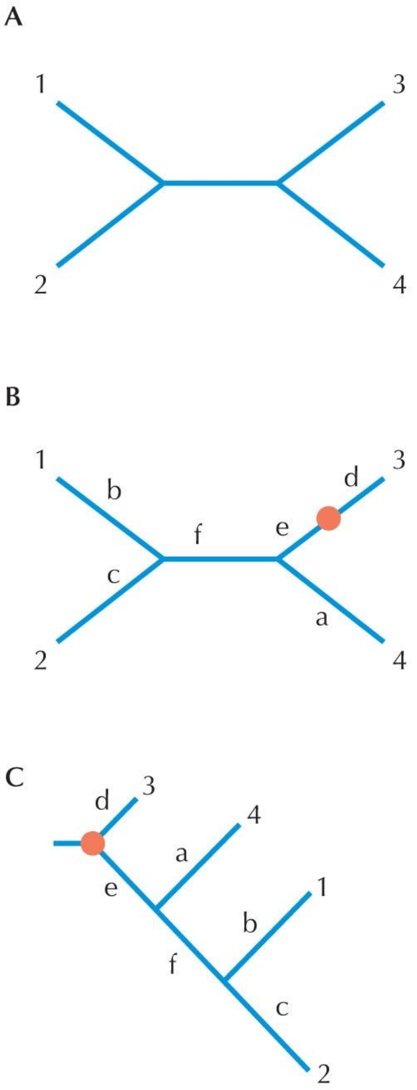


FIGURE 27.24. Rooting a tree with an outgroup. (*A*) Unrooted tree. (*B*) Suppose species 3 is determined to be the outgroup. Then the tree can be rooted between 3 and the other taxa (root shown as *red dot*). Branches are labeled with letters for easier comparison of *B* and *C*. (*C*) Rooted tree.

Evolution © 2008 Cold Spring Harbor Laboratory Press

	Charater-based method	**Non-character based**
Explicit evolutionary model	Maximum likelihood	Pairwise distance
No explicit evolutionary model	Maximum parsimony	

I. Maximum Likelihood

- Likelihood refers to an explanation of the observed data D.
- Determining the probability of data – plausibility of evolutionary process
- If hypotheses vary, some hypotheses fit the data better – have a higher *likelihood* of being correct
- Flip a coin: *n* = 100, *h* = 21, *t* = 79, D = (21,79)

Probability of heads: qÎ $(0,1)$

I. Maximum Likelihood

- Assume that events are independent, thus θ does not change
- Probability of observing *H* number of *h* (heads)

$$\Pr\left[H=h\right]=\binom{n}{h}q^{h}(1-q)^{n-h}$$

- If you know likelihood (θ), then *h* in *n* coin tosses can be calculated – likelihood function

$$L(q)=\Pr\left[H=h\right]=\binom{n}{h}q^{h}(1-q)^{n-h}$$

I. Maximum Likelihood

- Different choices of θ generate better models of observed data than others
- Can calculate log (rather than product) of different θ

$$L(\theta) = \Pr\left[H = h\right] = \binom{n}{h}\theta^{h}(1-\theta)^{n-h}$$

$$\log\left[L(\theta)\right] = \log\binom{n}{h} + h\log\theta + (n-h)\log(1-\theta)$$

I. Maximum Likelihood

- Calculate the differential with respect to θ

$$\log\left[L(q)\right]=\log\begin{matrix}æ\\ ç\\ è\end{matrix}\frac{n}{h}\begin{matrix}ö\\ ÷\\ ø\end{matrix}+h\log q+(n- h)\log(1- q)$$

Becomes

$$L(q)=\frac{¶\log\left[L(q)\right]}{¶q}=\frac{h}{q}- \frac{n- h}{1- q}$$

$L'(\theta)>0$ for $0<\theta<\theta_0$ and $L'(\theta)<0$ for $\theta_0<\theta<0$

Therefore $\log[L(\theta)]$ maximum when $q_0=\frac{h}{n}$

I. Maximum Likelihood

- Therefore the maximum likelihood estimate:

$$\hat{q} = \frac{h}{n}$$

- When the value of θ maximizes log[L(θ)] this is the maximum likelihood.
- L(21,100) – if use θ = 0.5 then likelihood 1.61 x 10^{-9}
- L(21,100) – if use $\hat{q}$ = 0.21 then likelihood 0.0975

I. Maximum Likelihood

CT_1 = HTTTTHTHTH

$\Pr(D|H_1) = \Pr(\text{HTTTTHTHTH}|H_1)$

If $\theta = 0.5$

$L_{CT1} = (0.5)^{10} = 0.0009765$

What if we want to maximize likelihood

$\theta = (T/T+H) = 0.6$

$L_{CT1} = (0.6)^{10} = 0.0060466$

Would use MLE of 0.6

I. Maximum Likelihood

- Mutations are chance events
- Maximum likelihood approach determines how well you observe sequences based on:
  - Tree topology
  - Branch length
  - Evolutionary model used
- Maximum likelihood is the likelihood that data fits hypothesis
  - Your hypothesis is the tree (τ) and evolutionary model (θ) for the data

I. Maximum Likelihood

$$L(\tau, \theta) = \Pr(\text{Data} | \tau, \theta)$$

or

$$L(\tau, \theta) = \Pr(\text{aligned sequences} | \text{tree, model})$$

- Thus the MLE are the values for:

$$\hat{t}, \hat{q} = \arg\max L(t, q)$$

I. Maximum Likelihood

- Using JC69 model for 2 sequences

$$L(d) = \prod_{j=1}^{l} \pi_{s_1^j} P_{s_1^j} P_{s_2^j}(-\frac{4d}{3})$$

d = substitutions per site

$P_{xy}(t)$ = probability of seeing nucleotide *y* if nucleotide *x* was originally found

$\pi_{s_1^j}$ = is the probability of character $s^j{}_1$

Number of identical pairs (l_0) and different pairs (l_1) = 1

I. Maximum Likelihood

- Data summarized (for 2 sequences using the JC69 model) as D=(l_0,l_1)

$$d = -\frac{3}{4}\log\left[1 - \frac{4}{3}.\frac{l_1}{l_1 + l_0}\right]$$

- When *n*>2, rather that calculating based on observed sequences – calculate probability of finding specific nucleotide in a position
- Depends on model of sequence evolution *M*
- Each sequence on each alignment must be calculated – math intensive!

I. Maximum Likelihood

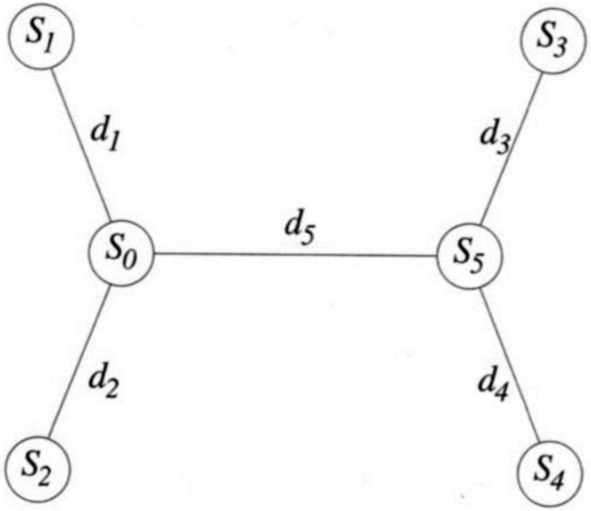
- Need to simply math:
    - *D_j* = nucleotide pattern at specific site
    - Each site (*s*) evolves according to *M*
    - Assume all sites evolve at rate μ
    - To weigh evolution – use rate specific factor ρ_j>0
    - Probability at each site simplified to:

$$\Pr\left[D_j \mid \tau, M, \rho_j\right],\ j = 1, \ldots, l$$

$$L = (\tau, M, \rho \mid D) \equiv \Pr\left[D_j \mid \tau, M, \rho_j\right] = \prod_{j}^{l} \Pr\left[D_j \mid \tau, M, \rho_j\right]$$

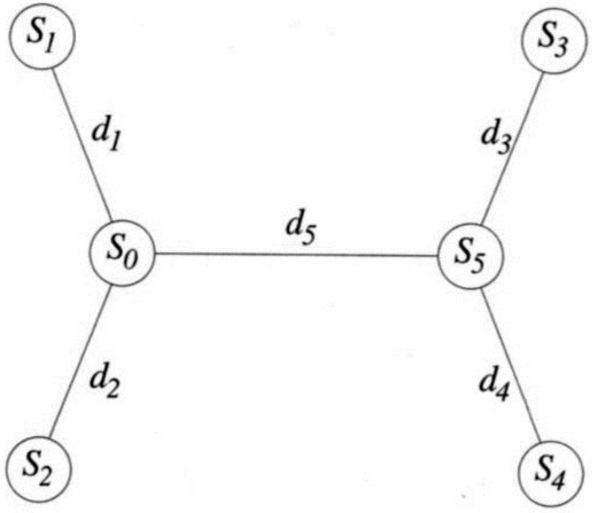
I. Maximum Likelihood

Tree for 4 sequences – assume ρ_j=1 for each *j*
Assume evolution began at S_0
Need to compute all 4 possible trees



At specific site $D_j = (s_1^j, s_2^j, s_3^j, s_1^j)$ and ancestral sites s_0^j, s_5^j

I. Maximum Likelihood



Probability of data given ancestral states s_0^j, s_5^j

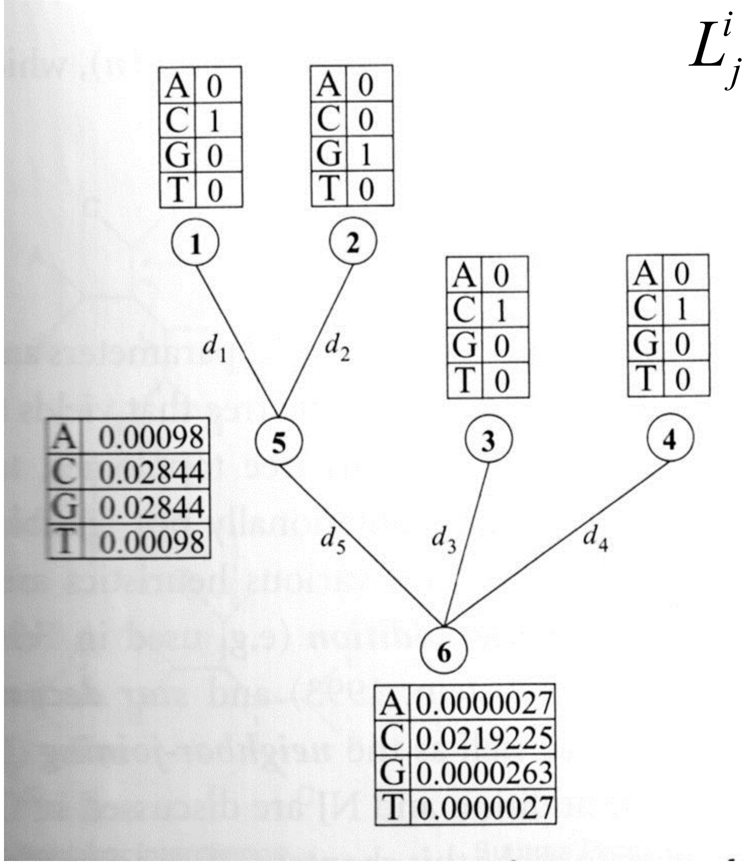
$$\text{Pr}\,\text{é}\!\!\!\text{ë}\, D_j, t, M, 1 | s_0^j, s_5^j\, \text{ù}\!\!\!\text{û} = P_{s_0^j, s_1^j}(d_1).P_{s_0^j, s_2^j}(d_2).P_{s_0^j, s_5^j}(d_5).P_{s_5^j, s_3^j}(d_3).P_{s_5^j, s_4^j}(d_4)$$

[Don’t really know ancestral state, so use substitution model to estimate from data (assuming π_G=π_A=π_T=π_C)]

I. Maximum Likelihood

- Alignments need to be calculated for inner nodes of tree

$$L_j^i(s) = \left[\sum_{x \in \{G,A,T,C\}} P_{sx}(d_{o_1}) L_j^{o_1}(x) \right]$$

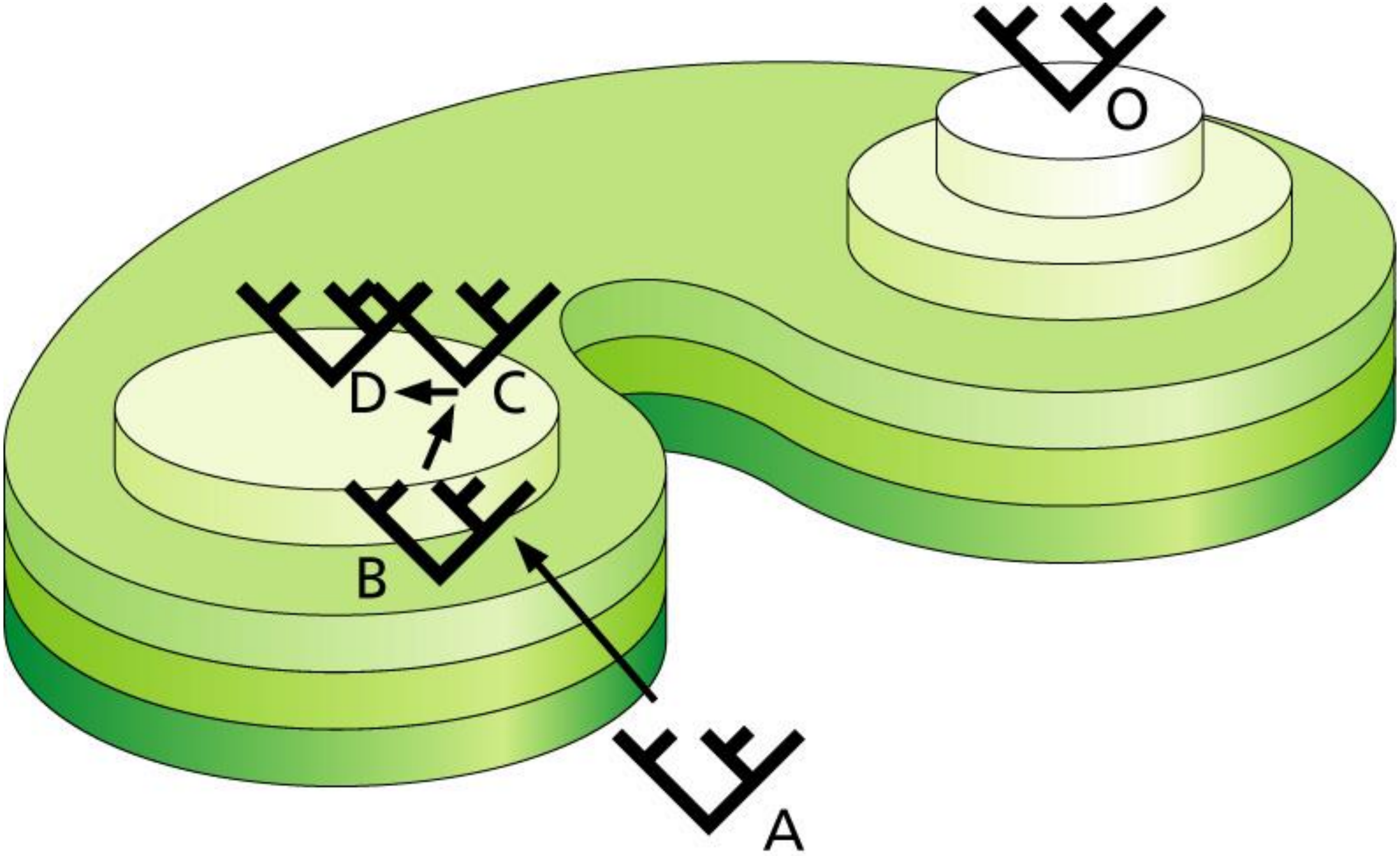


- D_j = *C,G,C,C*
- $d_1,\ldots d_5$ = 0.1
- Pr[*C*, *C*] = 0.9058
- Pr[*C*, *G*] = 0.0314
- $L^5_j(s)$ = $P_{CC}(d_1).P_{CG}(d_2)$
  = 0.9058 . 0.0314
  = 0.0054886

Ln*L* = -5.2051

II. ML Calculating Tree

- Don’t know branch lengths – need to calculate branch lengths for tree (τ) that maximizes Ln*L*
- Math complex for multiple sequence alignments
- Math massively complex to analyze all possible trees to actually find Ln*L*
- Very difficult to identify correct tree in tree space (huge number tree possibilities)

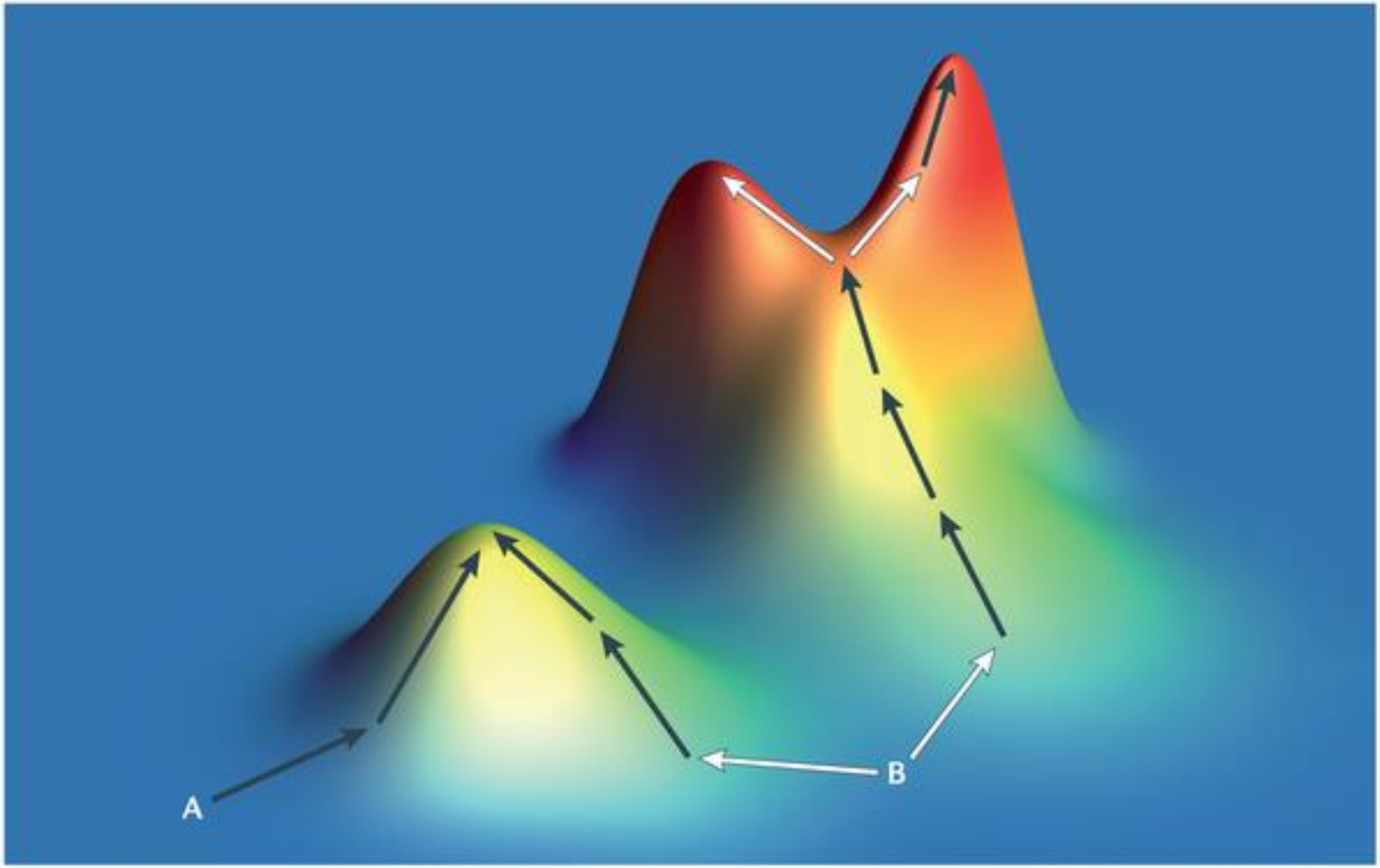

O
D
C
B
A

II. ML Calculating Tree

- Potential number of trees that can be analyzed is huge

$$t_n = \frac{(2n - 5)!}{2^{n-3}(n-3)!} = \prod_{i=1}^{n} (2i - 5)$$

- Use a heuristic approach to identifying trees
- Begin randomly changing things up
- Fast way of estimating ML
- Can easily become trapped in local optima

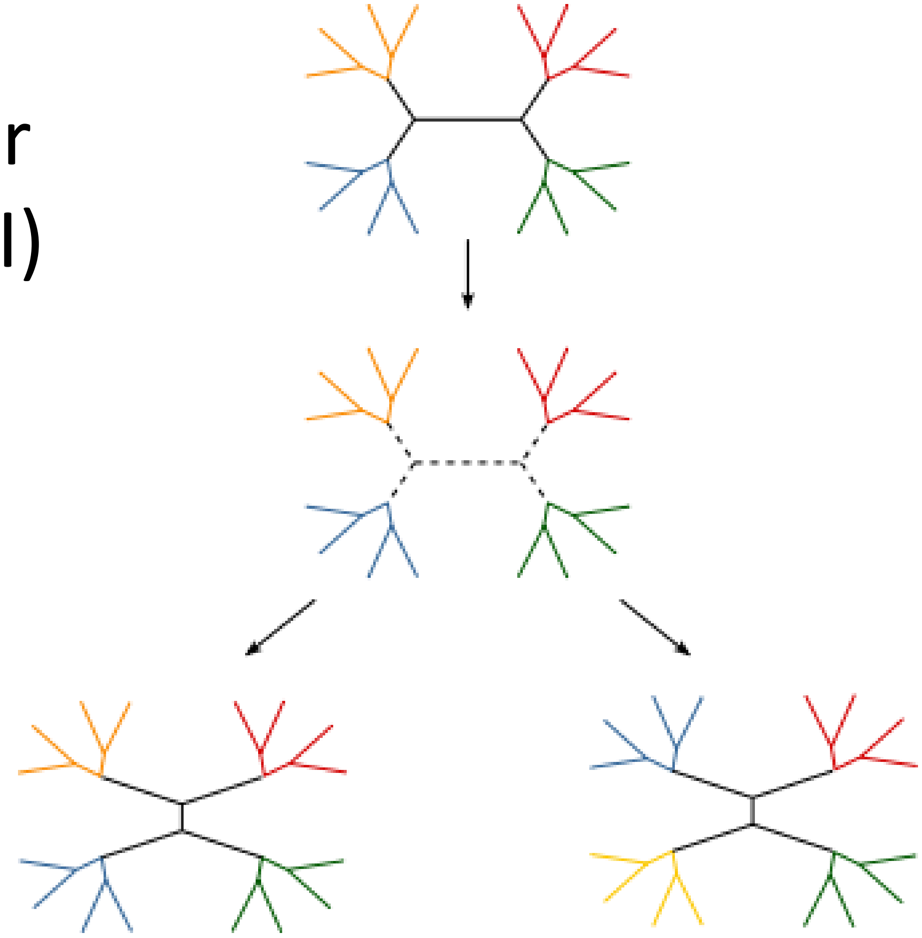


Nature Reviews | Genetics

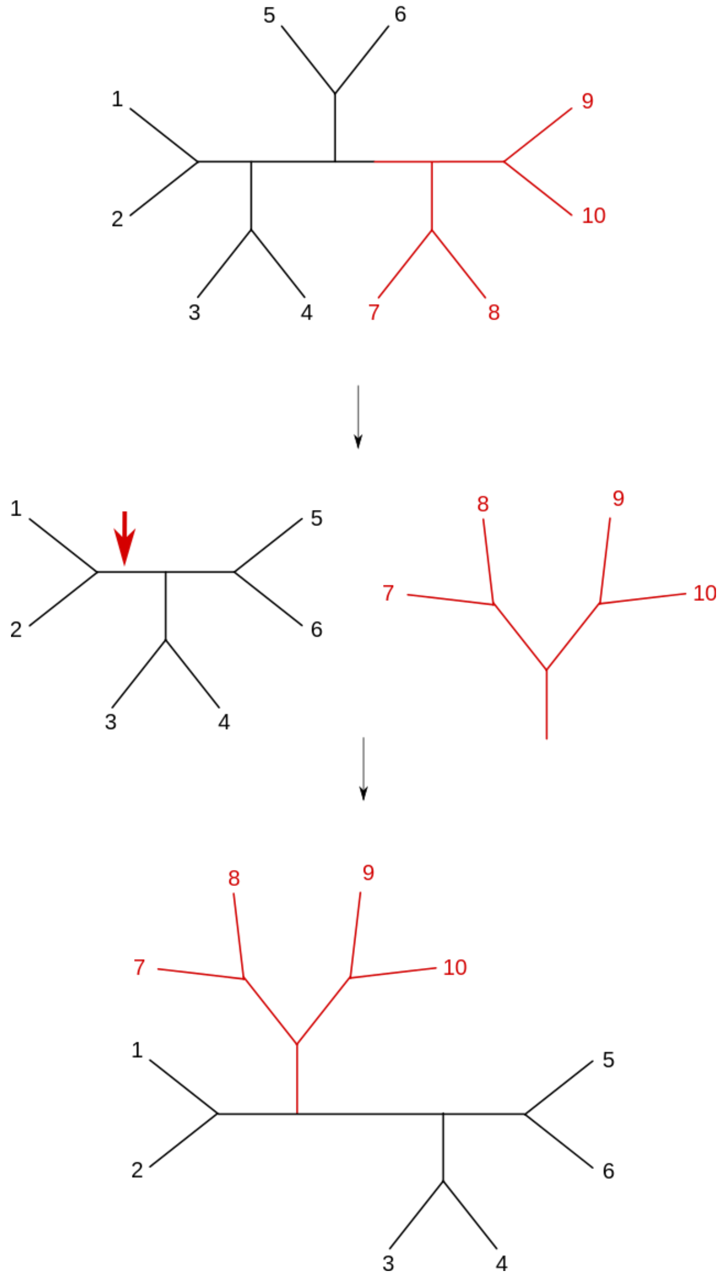
II. ML Calculating Tree

- Tree re-arrangements can limit ‘trapping’
- Change structure for tree with *n* nodes
  - Nearest neighbor interchange (NNI)
  - Subtree pruning and regrafting (SPR)
  - Tree-bisection/reconnection (TBR)

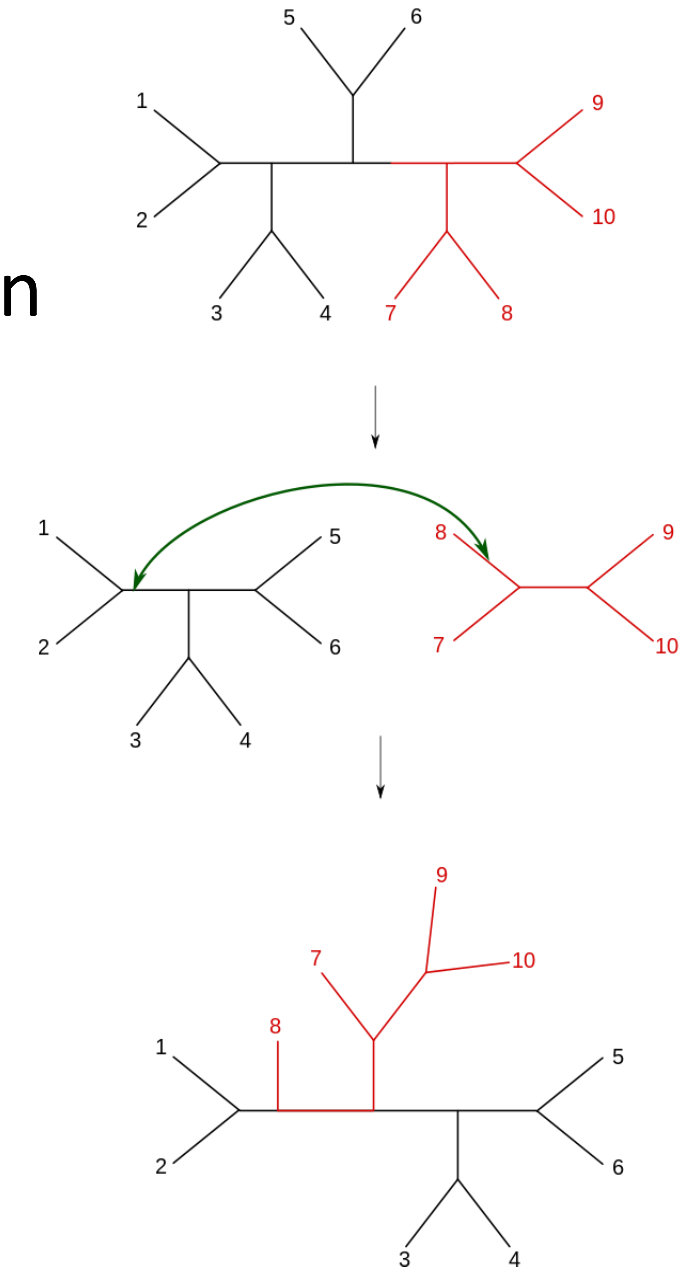
Nearest neighbor interchange (NNI)



Subtree pruning and regrafting (SPR)



Tree-bisection/reconnection (TBR)



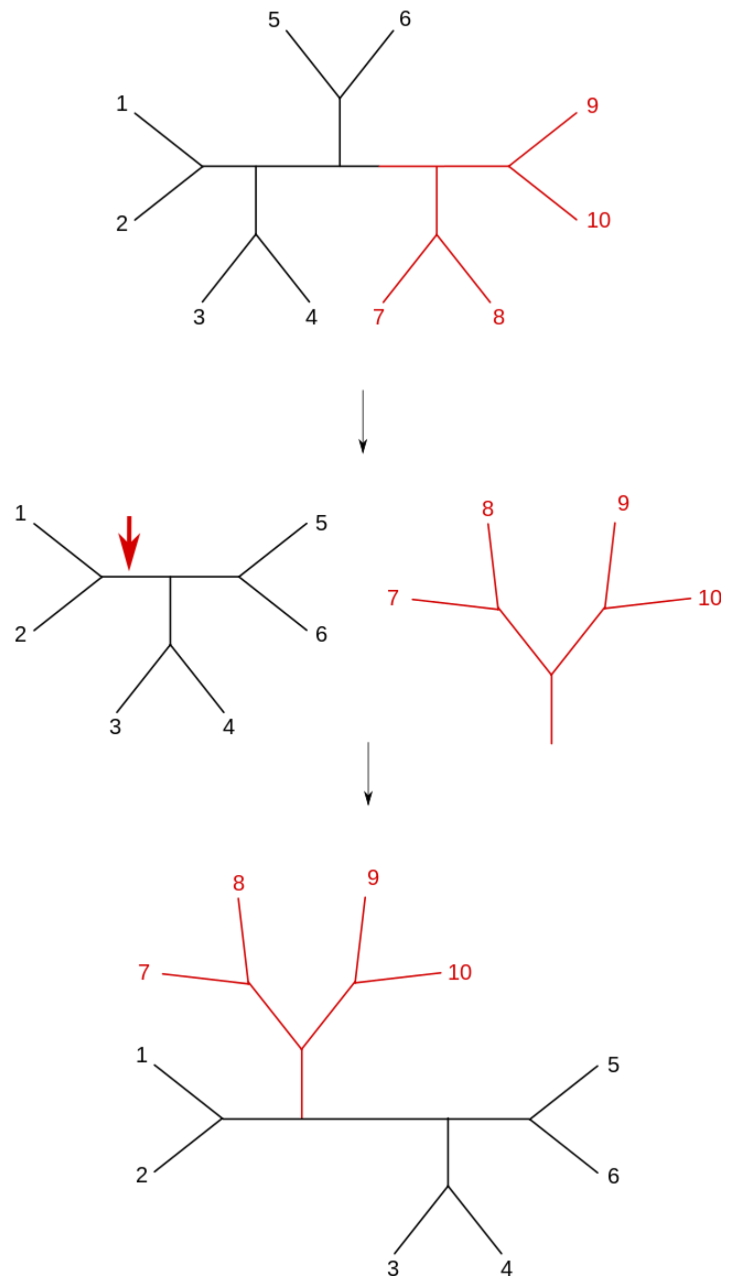
II. ML Calculating Tree

- Tree re-arrangements can limit ‘trapping’
- Change structure for tree with *n* nodes
  - Nearest neighbor interchange (NNI)
  - Subtree pruning and regrafting (SPR)
  - Tree-bisection/reconnection (TBR)
- Increases size of neighborhood
- Recalculates ML values
- Repeat!
- Whether you find local optima – depends on your data!

II. Calculating Tree using RAxML

- Randomized Axcelerated Maximum Likelihood
- Builds a tree using maximum parsimony
- Uses SPR variant lazy subtree rearrangement (LSR)
  - Assigns maximum distance between pruning and insertion (<25 branches)
  - Optimizes branch that originates at pruning site
- Best 20 trees optimized
- Process repeats until ML no longer changes

Subtree pruning and regrafting (SPR)



II. Calculating Trees: Tree Puzzle

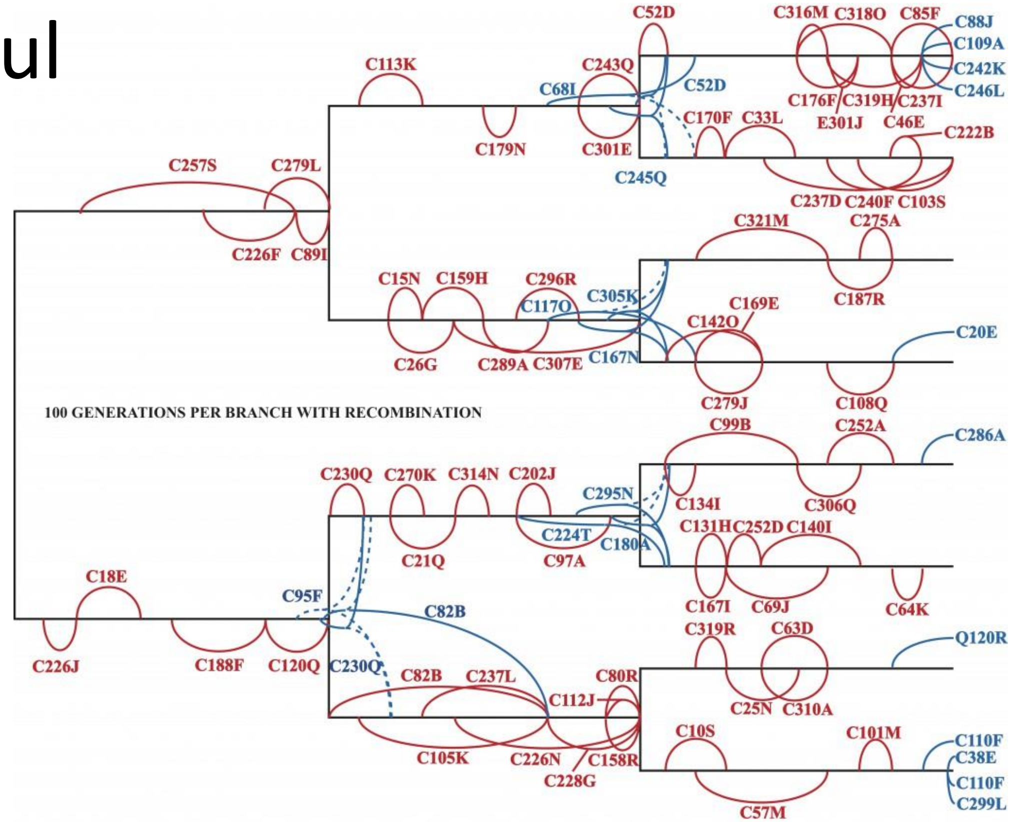
- Tree Puzzle 2002 (http://www.tree-puzzle.de/)
- Quartet puzzling method
  - Uses trees with four sequences
  - Generate NJ tree of each quartet
  - Optimize to obtain best ML score
  - Highest ML topology stored
  - Insert branches based on best quartet structure
  - Generate intermediate trees – score for consensus

II. Calculating Trees: PAUP*Rat

- PAUP*Rat
  - Plug-in for PAUP* (runs on CIPRES)
  - Works well for large datasets
  - Artificially weight data
  - Allow tree rearrangements
  - Set to original values – determine if best tree found
  - More accurate/efficient than NNI, SPR and TBR methods

II. Calculating Tree: Genetic Algorthims

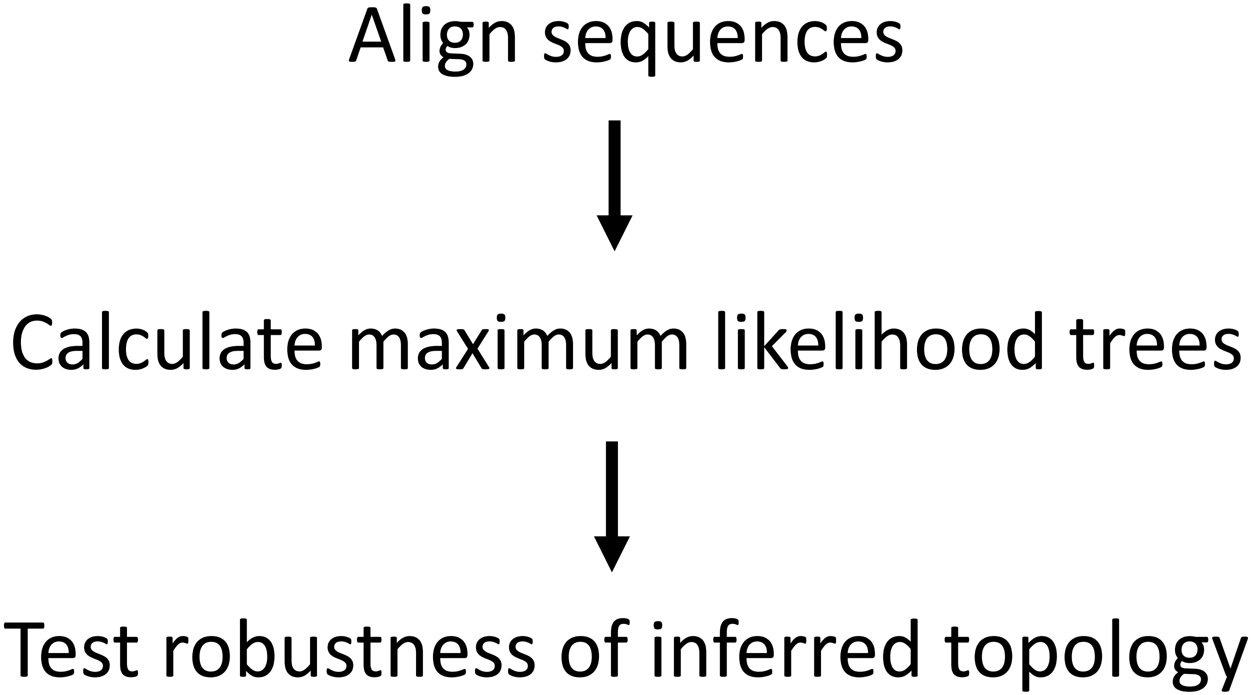
- Introduced in the 1990s
- Difficult to implement – GARLI (Genetic Algorithm for Rapid Likelihood Inference)
- Incredibly powerful


100 GENERATIONS PER BRANCH WITH RECOMBINATION

II. ML Robustness

- Generate best ML tree – no information on support of structure
- Carry out bootstrapping and compare with final ML tree
  - Standard bootstrapping approach
  - Non-parametric bootstrap
  - Shimodaira-Hasegawa-like (SH-like) method – create subtrees and score probability. Accumulate best probabilities

III. ML Methods



Align sequences

Calculate maximum likelihood trees

Test robustness of inferred topology

III. ML Methods

- Go into CIPRES gateway
- Use aligned data
- Run your data using the following tasks:
  - FastTree
  - GARLI
  - PAUP*Rat (set nchar values)
  - RAxML (set outgroup)
- Look at your trees
- In FigTree labeled ‘nodes/branches’ should be ‘bootstrap’