

ALGORITHMS & METHODS FOR PHYLOGENETIC TREES

- 1) Clustering algorithms
- 2) Parsimony algorithms
- 3) Maximum likelihood (ML) method
- 4) Bootstrapping procedure

Aim of phylogenetic algorithms:

based on a set of aligned sequences, construct phylogenetic tree with sequences as leaves and branch lengths representing evolutionary distances between sequences and their ancestors.

References:

- Durbin, Eddy, Krogh & Mitchison (1998): Biological Sequence Analysis. Cambridge University Press.
- Lake & Moore (1998): Phylogenetic analysis & comparative genomics. In Trends Guide to Bioinformatics 1998, 22–23.

Programs: we use the Phylip package

- Joe Felsenstein, <http://evolution.genetics.washington.edu/phylip.html>
- other programs: references at Felsenstein's site.

CLUSTERING ALGORITHMS

Data:

- distance or dissimilarity matrix for n sequences:
 $d_{ij}; i, j = 1, \dots, n,$
- variety of methods for computing distances d_{ij} (ML methods are introduced later on).

Methods:

- UPGMA (Unweighted Pair Group Method using Arithmetic averages):
 - result: rooted and additive tree with molecular clock,
 - intuitive and simple method.
- Neighbour joining:
 - result: unrooted and additive tree,
 - mathematical property: reconstructs correctly any additive tree.
- other approaches: least-squares, Fitch, Kitsch ... in Phylip package.

Biological assumptions:

- essentially inherent in distance matrix.

UPGMA EXAMPLE

Figure 7.4 in Durbin et al., 1998.

NEIGHBOUR JOINING ALGORITHM

Initialisation:

- tree $T = \{ \text{leaf nodes} \}$ (\sim sequences),
- leaves $L = T$.

Iteration:

- choose pair (i, j) in L with minimal distance D_{ij} (1),
- define new node k (in T and L), and corresponding
 - distances d_{km} , $m \in L$, (2)
 - branch lengths t_{ik} and t_{jk} , (3)
- remove nodes i and j from L .

Termination:

- when $|L| = 2$: join remaining two leaves and calculate branch length.

Formulas:

$$(1) D_{ij} = d_{ij} - (r_i + r_j), \text{ where } r_i = \frac{1}{|L|-2} \sum_{k \in L} d_{ik},$$

$$(2) d_{km} = (d_{im} + d_{jm} - d_{ij})/2,$$

$$(3) t_{ik} = (d_{ij} + r_i - r_j)/2, \quad t_{jk} = d_{ij} - t_{ik}.$$

NEIGHBOUR JOINING EXAMPLE

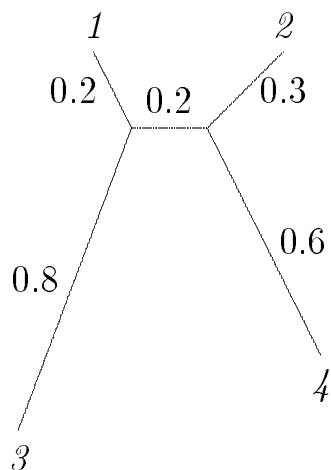


Figure 1. Example tree (difficult to reconstruct).

Initialisation: given distances d_{ij} , compute D_{ij} and r_i .

d_{ij}	D_{ij}				r_i
	1	2	3	4	
1	–	-2.1	-2.3	-2.1	1.35
2	0.7	–	-2.1	-2.3	1.45
3	1.0	1.3	–	-2.1	1.95
4	1.0	0.9	1.6	–	1.75

Step 1: select nodes (1,3), denote junction as node 5, recompute table, and $d_{15} = 0.2$ and $d_{35} = 0.2$.

d_{ij}	D_{ij}			r_i
	2	4	5	
2	–	-2.2	-2.2	1.4
4	0.9	–	-2.2	1.7
5	0.5	0.8	–	1.3

Step 2: select nodes (2,4), denote junction as node 6, compute $d_{26} = 0.3$ and $d_{46} = 0.6$.

Step 3: two remaining nodes joined, and $d_{56} = 0.2$.

The original tree has been successfully reconstructed.

PARSIMONY ALGORITHMS

Data:

- aligned sequences $1, \dots, n$ with values in an alphabet (nucleotides or proteins).

Method:

- result: unrooted tree with no branch lengths,
- idea: search for tree with minimal number of substitutions (“cost” in traditional parsimony):

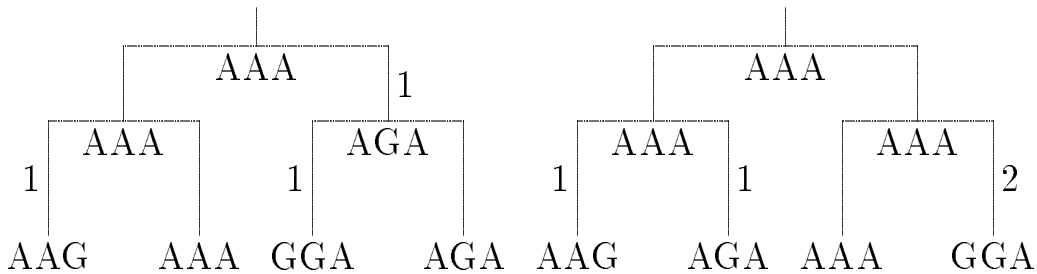


Figure 2. Two trees with parsimony costs.

- 2 steps in tree search:
 - 1) given tree structure and values at leaves, compute minimal cost at each site and sum over sites (recursion algorithm),
 - 2) search among tree structures
 - * simple exhaustive, for n small,
 - * branch and bound algorithm (“clever exhaustive”), still n rather small,
 - * stochastic/heuristic: random tree modifications adopted if cost lowered; note that sequence order determines starting point of search.

PARSIMONY ALGORITHMS - CTD.

Biological assumptions (Felsenstein):

- 1) Each site evolves independently,
- 2) Different lineages evolve independently,
- 3) The probability of a base substitution at a given site is small over the lengths of time involved in a branch of the phylogeny,
- 4) The expected amounts of change in different branches of the phylogeny do not vary by so much that two changes in a high-rate branch are more probable than one change in a low-rate branch.
- 5) The expected amounts of change do not vary enough among sites that two changes in one site are more probable than one change in another.

Other researchers claim that there are no assumptions involved in parsimony (references in Phylip documentation).

MINIMAL COST COMPUTATION IN PARSIMONY

Cost $S(a, b)$ for substitution of a by b (in traditional parsimony $S(a, b) = 1$ if $a \neq b$, and $S(a, a) = 0$).

Initialisation:

- denote by $S_k(a)$ the minimal cost at node k (and below) for the assignment of value a to node k ,
- start at $k = 2n - 1$, the number of the root node.

Iteration (recursion):

compute $S_k(a)$ for all a as follows,

- if k is not a leaf node:
 - compute $S_i(b)$ and $S_j(b)$ for all b at the daughter nodes i and j , and define
$$S_k(a) = \min_b[S_i(b) + S(a, b)] + \min_b[S_j(b) + S(a, b)],$$
- if k is a leaf node:
 - let $S_k(a) = 0$ if a equals sequence value, otherwise $S_k(a) = \infty$.

Termination:

- minimal cost of tree = $\min_a S_{2n-1}(a)$.

Note: it is possible to keep track of the minimal assignments to parent nodes in tree during the recursion, allowing display of a minimal tree.

MAXIMUM LIKELIHOOD (ML) METHOD

ML estimation of *parameters* in a *statistical model*:

- choose parameter which gives the data highest likelihood (“probability”),
- examples: binomial, Poisson, normal distributions, allele frequency for population in H-W equilibrium...

Need statistical model(s) for phylogenetic trees!
(parameters: tree structure T and branch lengths (t_i)).

Data:

- n aligned sequences x^j , $j = 1, \dots, n$, where $x^j = (x_u^j)_{u=1, \dots, N}$, $u = \text{site}$, $x_u^j \in \text{alphabet}$.

The idea:

- define “somehow” (according to evolutionary model)
 $P(x|y, t) = \text{prob. } y \mapsto x \text{ along branch of length } t$,
- $P(x^1, \dots, x^5 | (t_i), T) = P(x^1|x^4, t_1) \cdot P(x^2|x^4, t_2) \cdot P(x^3|x^5, t_3) \cdot P(x^4|x^5, t_4) \cdot P(x^5)$,
- ancestors x^4 and x^5 usually not known:
$$P(x^1, x^2, x^3 | (t_i), T) = \sum_{x^4, x^5} P(x^1, \dots, x^5 | (t_i), T),$$
- distribution of root (x^5) usually taken as frequencies in data.

MAXIMUM LIKELIHOOD (ML) METHOD - CTD.

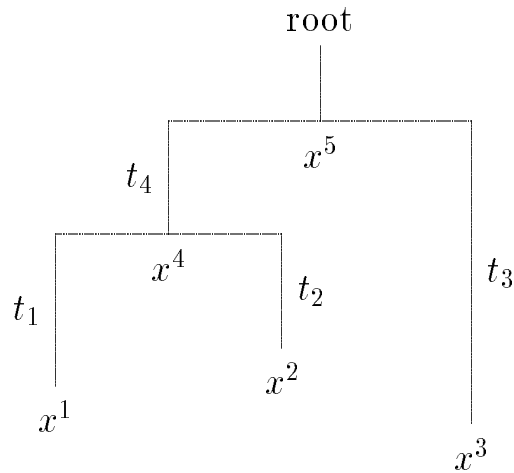


Figure 3. An example of a tree with three sequences.

Method:

- use $P(x|y, t)$ from model to calculate $P(x^1, \dots, x^n | (t_i), T)$ for given tree structure T and branch lengths (t_i) (Felsenstein's algorithm),
- define ML estimates by

$$(\hat{T}, (\hat{t}_i)) = \text{maximiser}_{(T, (t_i))} P(x^1, \dots, x^n | (t_i), T)$$

$$= \text{maximiser}_T \text{maximiser}_{(t_i)} P(x^1, \dots, x^n | (t_i), T).$$
- maximisation over (t_i) : EM-algorithm or standard optimisation algorithm,
- maximisation over T : “search” (exhaustive for very small trees only, or stochastic/heuristic),
- MCMC (Markov chain Monte Carlo) methods have been tried also.

MAXIMUM LIKELIHOOD (ML) METHOD - CTD. II

Biological assumptions (`dnaml`, `fastDNaml` programs):

- 1) Each site evolves independently,
- 2) Different lineages evolve independently,
- 3) Each site undergoes substitution at an expected rate which may be constant or vary across sites, and is chosen from a series of rates (each with a probability of occurrence) specified as constants,
- 4) Further model-specific assumptions.

EVOLUTIONARY MODEL BEHIND ML METHOD

Simple model for $P(x|y, t)$ (prob. $y \mapsto x$ over branch t):

- no insertions or deletions,
- independence between sites $\rightarrow P(x|y, t) = \prod_u P(x_u|y_u, t)$,
- single site residue substitutions occur according to stationary Markov process $\{x_u(t)\}$ in “time” $t =$ branch length (interpretation: $t \propto$ mutation rate \times time),
- model characterised by matrix R of substitution rates (mathematically: intensity matrix);
very simple example (Jukes-Cantor model)

$$R = \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix},$$

interpretation:

- from state **A** the substitution process jumps to other states with probability $\alpha/(\alpha+\alpha+\alpha) = 1/3$,
- waiting time until next jump \sim exponential distribution: $P(\text{jump before } t) = 1 - e^{-3\alpha t}$,
- substitution/transition matrix for short time ε :
 $P(\varepsilon) \approx I + \varepsilon R$.
- general form of substitution probabilities:
 $P(t) = \exp(tR)$.

EVOLUTIONARY MODEL BEHIND ML METHOD II

Another example: Kimura model

$$R = \begin{pmatrix} -2\beta - \alpha & \beta & \alpha & \beta \\ \beta & -2\beta - \alpha & \beta & \alpha \\ \alpha & \beta & -2\beta - \alpha & \beta \\ \beta & \alpha & \beta & -2\beta - \alpha \end{pmatrix}, \alpha > \beta,$$

- α = rate of transitions (**A**↔**G** or **C**↔**T**),
- β = rate of transversions (**A, G**↔**C, T**).

Felsenstein's model (in **dnaml** program):

$$r_{ij} = \begin{cases} (k/\Pi_j + 1)u\pi_j & \text{transition,} \\ u\pi_j & \text{transversion,} \end{cases}, i \neq j,$$

- u = overall rate parameter,
- k = transition/transversion ratio parameter,
- π_j = frequency of base j ,
- $\Pi_j = \pi_A + \pi_G$ or $\Pi_j = \pi_C + \pi_T$, depending on whether j is purine or pyrimidine.

MAXIMUM LIKELIHOOD DISTANCE

Given an evolutionary model $P(x|y, t)$, define the ML distance between two sequences x^i and x^j of length N by

$$d_{ij}^{ML} = \text{maximiser}_t \prod_{u=1}^N P(x_u^i | x_u^j, t).$$

Can be shown (mathematically):

- d_{ij}^{ML} equals the ML estimate of the distance between leaves i and j in any given tree,
- d_{ij}^{ML} is an approximately (asymptotically when N large) additive distance in any given tree.

Consequences:

- if the evolutionary model is “correct”, then d_{ij}^{ML} has some optimal properties as estimate of distance,
- if the sequences are long and d_{ij}^{ML} is approximately additive, the neighbour joining method for tree reconstruction should perform well;
note that this is interesting because the computational effort is much less than in ML estimation of trees.

In the Phylip package (program **dnadist**), all distance matrices are calculated by the maximum likelihood method, but one can choose between different models.

BOOTSTRAP METHODS IN GENERAL

- collection of ideas involving simulation of “new” observations and aiming at statistical inference without relying on exact or asymptotic distributions,
- no unifying theory and in non-standard situations no guarantee that it works; in practice, sensible methods perform well. . .
- methods are always based on assumptions (!)

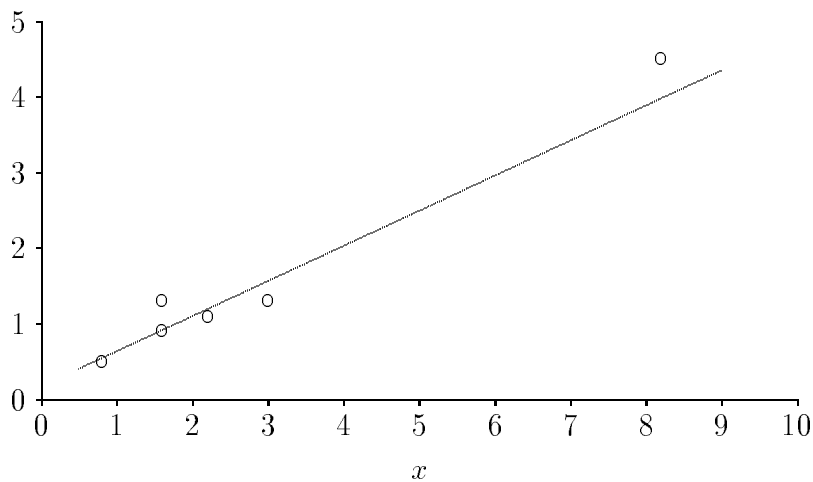
2 major distinctions:

- parametric bootstrap: simulate from a parametric model, e.g. with estimated parameters,
- non-parametric bootstrap: simulate with replacement from “observed data” (empirical distributions), demonstration example:
 - observed values: 1.1, 1.3, 1.3, 0.9, 0.5, 4.5,
 - choose between them with probability $1/6$, repeat 6 times to obtain a “pseudo” dataset (resampling)
 - generate “large” number of pseudo datasets, and use them to represent variability on estimates.
 - note: makes sense only if
 - * values are independent and from same distribution (i.i.d.),
 - * empirical distribution not too far off true distribution.

NON-PARAMETRIC BOOTSTRAP IN REGRESSION

Demonstration example:

x	2.2	1.6	3.0	1.6	0.8	8.2
y	1.1	1.3	1.3	0.9	0.5	4.5



2 different approaches:

- resample residuals (assumes residuals to be i.i.d.),
- resample pairs (x, y) (assumes pairs to be i.i.d.).

to obtain standard errors or confidence intervals for parameters (intercept, slope).

Note that in this example results will be quite different.

BOOTSTRAPPING PHYLOGENETIC TREES

Data:

- n aligned sequences of length N ($n \times N$ matrix).

Aim:

- obtain confidence in particular feature of tree obtained by phylogenetic algorithm, e.g. segregation of some species on separate branch,
- note: idea applicable for any phylogenetic algorithm.

Idea (Felsenstein, 1985):

- do a large number R of times (say, 100 or 1000):
 - create “pseudo” dataset by resampling columns in data matrix with replacement,
 - construct phylogenetic tree by method of interest,
- calculate bootstrap support of feature = proportion where feature present (among R resamples)
- interpretation of bootstrap support:
 - no strict statistical interpretation as a P -value,
 - “rule of thumb”: $>70\%$ bootstrap support \Rightarrow “likely to be correct at 95% level” (Lake & Moore),
- \sim case (pair) sampling in multiple linear regression,
- assumptions: i.i.d. alignments at different sites.