

## Phylogenetic trees: estimation by maximum likelihood

There are four frequently-used approaches to phylogenetic tree estimation. These are parsimony (discussed in class), the UPGMA method, the neighbor-joining method and the method of *maximum likelihood*. Details of the UPGMA and the neighbor-joining method will not be discussed here. In these notes we focus on estimation of phylogenetic trees by the method of maximum likelihood. This is the only one of these four methods that is statistical: the other three methods are *algorithmic*.

To understand how the maximum likelihood method of tree construction works, we must first discuss maximum likelihood estimation as a general statistical process. This is done below. The discussion is very brief and overlooks many subtle points, which we just ignore.

### Maximum likelihood estimation: general concepts

Suppose that we have some data  $D$  from some experiment or survey. The probability of these data is assumed to depend on some unknown *parameter*  $\xi$ , and so we write the probability, or likelihood, of the data  $D$  as  $L(D; \xi)$ . We wish to estimate  $\xi$  using the data  $D$  by the method of maximum likelihood.

The value  $\hat{\xi}$  of  $\xi$  for which  $L(D; \xi)$  is maximized (as a function of  $\xi$ ) is called the *maximum likelihood estimator* of  $\xi$ . This value is a function of the data  $D$ . Examples are given below.

This estimator is often found by differentiating  $L(D; \xi)$  with respect to  $\xi$  and solving the equation  $\frac{d}{d\xi}L(D; \xi) = 0$ . Care must be used with this procedure, since the maximum might be reached at a boundary point.

In those cases where the maximum of  $L(D; \xi)$  is found through the differentiation procedure, it is equivalent, and often easier, to solve the equation

$$\frac{d}{d\xi} \log L(D; \xi) = 0. \tag{1}$$

The above is a very brief introduction to the concept of maximum likelihood estimation. We have used the generic symbol  $\xi$  for the

unknown parameter in this procedure: in practice, we normally employ a notation that is typically used in any specific application of the theory. This is done in the examples below.

*Example 1.* Suppose we wish to estimate the recombination fraction between two gene loci. This recombination fraction is an unknown parameter, and the commonly accepted notation for it in the genetics literature is  $\theta$ . We use this notation here instead of  $\xi$ .

Consider  $n$  independent genetic transmissions from parent to offspring. Each of these results in either a recombinant transmission ( $R$ ) (with probability  $\theta$ ) or a non-recombinant transmission ( $N$ ) (with probability  $1 - \theta$ ). Suppose that  $n$  transmissions result in the collection  $NNRNRRN \dots NR$ . This collection is our data  $D$ . The probability, or likelihood, of these data is

$$L(D; \theta) = (1 - \theta)(1 - \theta)\theta(1 - \theta)\theta\theta(1 - \theta) \dots (1 - \theta)\theta. \quad (2)$$

This can be written as  $L(D; \theta) = \theta^x(1 - \theta)^{n-x}$ , where  $x$  is the total number of recombinant transmissions.

The logarithm of  $L(D; \theta)$  is  $x \log \theta + (n - x) \log(1 - \theta)$ . Differentiating this with respect to  $\theta$  and setting the derivative equal to zero, we get

$$\frac{x}{\hat{\theta}} - \frac{n - x}{1 - \hat{\theta}} = 0.$$

The solution of this equation is  $\hat{\theta} = \frac{x}{n}$ . It can be checked that this corresponds to a maximum of  $L(D; \theta)$ .

Note that what we are saying is that our *estimate* of  $\theta$ , namely  $\hat{\theta}$ , is  $\frac{x}{n}$ . We still do not know what the exact true value of  $\theta$  is. But at least we now have an estimate of that value.

This is a common-sense estimate: if we see 23 recombinations in 100 transmissions, it is natural to estimate the recombination fraction  $\theta$  by  $\frac{23}{100}$ . Note however that this is only an *estimate* of the true recombination fraction  $\theta$ ; with another set of data we would almost

certainly get another estimate.

*Example 2.* As a more complicated example, suppose that  $x_1, x_2, \dots, x_n$  are independent observations, each having a Poisson distribution with parameter  $\lambda$ . These values  $x_1, x_2, \dots, x_n$  are our data  $D$ . Then

$$\begin{aligned} L(D; \lambda) &= \frac{e^{-\lambda} \lambda^{x_1}}{x_1!} \times \frac{e^{-\lambda} \lambda^{x_2}}{x_2!} \times \dots \times \frac{e^{-\lambda} \lambda^{x_n}}{x_n!} \\ &= \frac{e^{-n\lambda} \lambda^{\sum x_i}}{\prod (x_i!)}, \end{aligned}$$

all sums and products in this example being over  $i = 1, 2, \dots, n$ . Thus

$$\log L(D; \lambda) = -n\lambda + \left( \sum x_i \right) \log \lambda - \log \left( \prod (x_i!) \right)$$

and

$$\frac{d}{d\lambda} \log L(D; \lambda) = -n + \frac{\sum x_i}{\lambda}. \quad (3)$$

This derivative is zero when  $\hat{\lambda} = \bar{x}$ , the average of the values  $x_1, x_2, \dots, x_n$ . It can be checked that this corresponds to a maximum of  $L(D; \lambda)$ . Thus

$$\text{maximum likelihood estimator of } \lambda = \hat{\lambda} = \frac{\sum x_i}{n} = \bar{x}. \quad (4)$$

Maximum likelihood estimators can be shown to have many desirable optimality properties, and because of this they are used almost invariably in scientific estimation procedures. However these properties apply only when the parameter  $\xi$  is a number. They do not apply if the parameter  $\xi$  is the topology of a phylogenetic tree, since this topology is a shape, not a number. Unfortunately, the properties of maximum likelihood estimation when used in the literature in the context of phylogenetic tree estimation are often applied incorrectly, because claims are made about optimal estimation of tree shape when such claims should not be made.

\*\*\*\*\*

We now return to the idea of using maximum likelihood in the context of phylogenetic tree estimation. Note that we use the word “*estimation*”; we are only estimating the tree from our data, and cannot be at all certain that we are thereby reconstructing the correct tree. With different data we might obtain a different estimate of the tree. (The same comment applies for estimating a tree by parsimony, UPGMA and neighbor-joining. These all provide tree estimates, not tree reconstructions.)

We assume that the data used in the maximum likelihood method are aligned DNA sequences in a collection of species at the “leaves” of the tree to be estimated. An example is given below.

Since maximum likelihood estimation of anything must start from a likelihood of these observed data, we have to find a likelihood of the data. This likelihood comes from assuming some evolutionary model. In practice, these models are highly simplified and cannot really be expected to describe “reality.” Three commonly-used evolutionary models in the phylogenetic context are the Jukes-Cantor model, the Kimura two-parameter model and the Felsenstein model. These are now briefly described.

#### *The Jukes-Cantor model*

Suppose we follow the predominant nucleotide at any specific site in the line of descent leading from an original ancestor to some contemporary species. From time to time this nucleotide type will change, for example from an  $a$  to a  $t$  or from a  $c$  to an  $a$ . The Jukes-Cantor model is a Markov chain model with four states,  $a, g, c$  and  $t$ . and is (for example) in state  $g$  if the predominant nucleotide in the population of interest is  $g$ . The transition matrix for the Jukes-Cantor Markov chain, with states written in the order  $a, g, c$  and  $t$ , is

$$P = \begin{bmatrix} 1 - 3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1 - 3\alpha & \alpha & \alpha \\ \alpha & \alpha & 1 - 3\alpha & \alpha \\ \alpha & \alpha & \alpha & 1 - 3\alpha \end{bmatrix}. \quad (5)$$

Here  $\alpha$  is a parameter depending on the time-scale chosen: If unit time were chosen as 100,000 generations,  $\alpha$  would take a value

smaller than it would be if unit time were chosen as 200,000 generations. Whatever time scale is chosen, it is clearly necessary that  $\alpha$  be less than  $\frac{1}{3}$ .

*The Kimura two-parameter model*

The highly symmetric assumptions implicit in the Jukes–Cantor model are not realistic. A *transition*, that is, the replacement of one purine by the other (for example of *a* by *g*) or of one pyrimidine by the other, is in practice more likely than a *transversion*, that is, the replacement of a purine by a pyrimidine or of a pyrimidine by a purine. Kimura proposed a two-parameter model to allow for this. The transition matrix  $P$  for this model, with the same ordering of states as that used for the Jukes–Cantor model, is

$$\begin{bmatrix} 1 - \alpha - 2\beta & \alpha & \beta & \beta \\ \alpha & 1 - \alpha - 2\beta & \beta & \beta \\ \beta & \beta & 1 - \alpha - 2\beta & \alpha \\ \beta & \beta & \alpha & 1 - \alpha - 2\beta \end{bmatrix}. \quad (6)$$

Here  $\alpha$  is the probability of a transition in one time unit, while  $\beta$  is the probability that a purine is substituted by a nominated pyrimidine in one time unit and is also the probability that a pyrimidine is substituted by a nominated purine in one time unit. It is required that  $\alpha + 2\beta < 1$ , and would normally be assumed that  $\alpha > \beta$ .

*The Felsenstein model*

A different form of generalization was introduced by Felsenstein. In this model the probability of substitution of any nucleotide by another is proportional to the “long-term” frequency of the substituting nucleotide. This implies a transition matrix  $P$  of the form

$$\begin{bmatrix} 1 - \alpha + \alpha\varphi_a & \alpha\varphi_g & \alpha\varphi_c & \alpha\varphi_t \\ \alpha\varphi_a & 1 - \alpha + \alpha\varphi_g & \alpha\varphi_c & \alpha\varphi_t \\ \alpha\varphi_a & \alpha\varphi_g & 1 - \alpha + \alpha\varphi_c & \alpha\varphi_t \\ \alpha\varphi_a & \alpha\varphi_g & \alpha\varphi_c & 1 - \alpha + \alpha\varphi_t \end{bmatrix}, \quad (7)$$

where  $(\varphi_a, \varphi_g, \varphi_c, \varphi_t)$  are the “long-term frequencies” of *a, g, c* and *t* respectively, and  $\alpha$  is a parameter of the model.

This model generalizes the Jukes–Cantor model, to which it reduces if  $\varphi_a = \varphi_g = \varphi_c = \varphi_t = \frac{1}{4}$ .

We now assume that one or other evolutionary model has been chosen, and that the topology of the tree is given. Our aim is to find the maximum likelihood estimate of the various arm lengths in the tree, given the assumed evolutionary model.

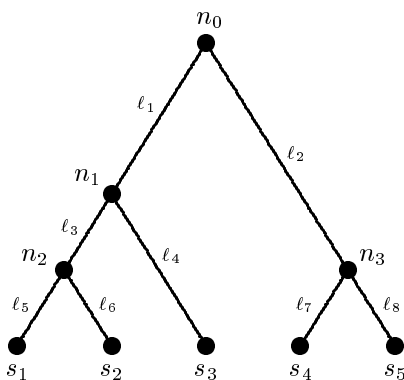
This is done by writing down the likelihood of the data in terms of these lengths as parameters, and then maximizing this likelihood with respect to these lengths. This maximization process is, in practice, usually extremely difficult.

When, as does happen in practice, data from many sites are used in the estimation process, many simplifying assumptions are, in practice, made. One such simplifying assumption often made is that the substitution processes at different sites within any species are described by the same stochastic model (for example, all are described by the Jukes–Cantor model and the same parameter  $\alpha$  of that model applies at every site.) It is also essentially always assumed that the evolutionary processes at different sites are independent of each another. This last assumption allows an analysis of the evolution at the various sites separately, with an overall likelihood obtained by multiplying individual site likelihoods.

It is also frequently assumed that time homogeneity of the common stochastic process applies, and that different species evolve independently.

All of the above assumptions can hardly be expected to approximate reality closely, and much recent research attempts to remove them or at least to assess the biases involved when they are incorrectly made.

We now discuss an analysis in which the all the above assumptions are made. Initially, no specific assumption is made below about the evolutionary model chosen.



Suppose that we have data from five species  $s_1, \dots, s_5$  and that the topology of the phylogenetic tree connecting them is as given in the figure shown. At any particular site the five respective nucleotides in these five species comprise all the data available for this site.

Consider first some specific nucleotide site. Denote these nucleotides at this site in the five species at the leaves of the tree by  $A, B, C, D, E$ . We do not know the nucleotides at the internal nodes  $n_0, n_1, n_2$ , and  $n_3$ , nor do we know the lengths  $\ell_1, \dots, \ell_8$  of the various arms in the tree. The aim of the process is to estimate  $\ell_1, \dots, \ell_8$ , which are lengths of time.

Suppose that the nucleotide at the root  $n_0$  is  $W$  and is fixed, and denote the nucleotides at the nodes  $n_1, n_2$ , and  $n_3$  by  $X, Y$ , and  $Z$ , respectively.

There is a certain likelihood for the arm lengths  $\ell_1, \dots, \ell_8$ , deriving from whichever nucleotide evolutionary model is chosen.

Denote the probability of the nucleotide  $W$  by  $\varphi_W$  and the probability, under the model assumed, of a substitution of nucleotide  $A$  by nucleotide  $B$  after time  $\ell$  by  $P_{AB}(\ell)$ .

Then the joint probability that  $W$  is indeed the nucleotide at the root of the tree, that the nucleotides  $X, Y$  and  $Z$  occur at the nodes as indicated, and that the arm lengths take the values  $\ell_1, \dots, \ell_8$  is

$$\varphi_W P_{WX}(\ell_1) P_{WZ}(\ell_2) P_{XY}(\ell_3) P_{XC}(\ell_4) \times$$

$$P_{YA}(\ell_5) P_{YB}(\ell_6) P_{ZD}(\ell_7) P_{ZE}(\ell_8).$$

Expressions of this form are now computed for all 64 possible combinations of nucleotides at the internal nodes. The sums of the resulting expressions give the likelihood of  $\ell_1, \dots, \ell_8$  conditional on the assumption that the nucleotide at the root of the tree is  $W$ . This calculation is then made for all four possible nucleotides values for  $W$ , and the sum of these four expressions is, for the site in question, the likelihood of  $\ell_1, \dots, \ell_8$ .

This procedure is now repeated over all the nucleotide sites in the data, and the overall likelihood is computed from the product of the various site likelihoods. This overall likelihood can now, at least in principle, be maximized.

This procedure can now be done, again in principle, for all possible topologies, and then the maximum likelihood tree is that which, taken over all possible topologies, has the maximum likelihood.

This description glosses over extremely difficult computational problems, which can be simplified to some extent by various algorithmic devices.

It is clear that many assumptions and simplifications are involved in the procedure, some highly dubious. Much current research is involved with relaxing these assumptions.

\*\*\*\*\*

### Example

In this example we give a tree estimation using the four popular approaches, namely parsimony, UPGMA, neighbor-joining and maximum likelihood. The same data are used for each approach. The four tree methods in effect use different optimality criteria, and thus they will often produce different trees even though using the same



tggacctccgacctgacctggcggccacat ttcggcatccctatgtcatctccta cctgcacccagggaacac  
tagacctccggcactgcacaggggccacgt ttcggcatccctatgtcatctccta cctgcacccagggaacac  
tggacctccggcactgcactggggggccacgt ttcggcatccctatgtcatctccta cctgcacccagggaacac  
tagacctccacactgcactggggggccacgt ttcgggatctcctatgtcatctccta cctgcacccagggaacac  
tggacctccgacactgcactggggggccatgt ctctgggatccctacgtcatctccta cctgtaccagggaacac  
tggatctccggcactgcacaggaggccaggt ctctggcatccctacatcatctccta cctgcacccagggaacac  
tggacctccggcactgtatgggtggccatgt ctccagcatccctacatcatctccta cctacacccggaggagc  
tggacctaaaggcactgcacggggggccatgt ctccagatccctacctcatctccta cctgcactcaggggagc

cctgctcctagttagacacagttta tga ccggccatcaaacactaacacagagatctggaccagcctcaggtgctg  
tctgctgcatgttagacacagttta tga ccggccatcaaacacacacacagagatctggaccctgcccaggtgctg  
aatcatgcatgtgaacacacatctatgatcggccctctaataccaacacagagatctggaccctggccaaggtcctg  
ggtcctgca cgttgacacattta caa cgtccctctaaca caa cactgagctctggactttgctcaggtgctt  
cgtgctgca cgtggacacattta caa cggccctccaatacgaactacggagctctggaccctgcccaggtgctg  
agtccctgca cgtggataccatctatgatcggccctctaatacga cactgagatctggaccctgcccgaagctcta  
agtccctgcatgtggataccatctatgatcggccctctaatacga cactgagatctggaccctgcccgaagctcta  
ggtcctgca cgtggacacacatctatgatcggccctccaatacga cactgagatctggaccctgcccgaagctcctg  
ggtcctgca cgtggacacacatctatgatcggccctccaatacga cactgagatctggaccctgcccgaagctcctg  
cgtcctgaa tgggaccca ctctatgacccccctccaacacga ccaacagagatctggaccctgcccaggtcctg  
ggtcctgcatatggacacacatctatgacccccctccaatacga cactgagctctggaccctgcccagctccag  
catcctgca cgtggacacacatctatcaaacggccctccaacacga ccaacagagatctggaccctgcccaggtcctg  
agtgctgca cgtggacacacattta caa cggccctccaatacga actactggggtctggaccctgcccaggtgctg  
tgtgctgca cgtggacacacattta caa cggccctccaatacga actactgagctctggaccctgcccaggtgctg

ggtgagaggtatggaggggagaaggacatgggtggttctcaccagccatcatactgtaggggtagctgaggatctg  
ggtgagaggtacgggtggggagaaggacgtgggtggtcctcaccagccatcacacggctcggggtagcagaggatattg  
ggggagaggtacagtgctgacaaggatgtgggtggtcctcaccagtgccacactggaggagtggtgtaggacattg  
ggggagagatacagtgctgagaaggatgtgggtggtcctcaccagtggtcaaacgggggtggtgctgaggacattg  
ggggagaggtatagcggcaaggatgtgggtggtcctcaccagtggtcaccacagggggctggcggaggacatcg  
ggagagaactacgggtccgataaggatgtgggtggtcctcaccagtggtcgcaacgggggtggtgctgaggacatcg  
ggagacaactacgggtccgataaggatgtgggtggtcctcaccagtggtcgcaacgggggtggtgctgaggacatct  
ggagacaggtacagtgctgacaaggatgtgggtggtcctcaccagtggtcaccacagggggctggtgtaggacatcg  
ggagagaggtacagtgctgacaaggatgtgggtggtcctcaccagtggtcaccacgggggctggtgtaggacatctg  
ggagagaggtacagtgctgacaaggatgtgggtggtcctcaccagtggtcaccacactggaggagtggtgtaggacattg  
ggagagcggta cggtgtagacaaggatgtgggtggtcctcatcagcgaccacactgggggtggtgtaggacattg  
ggagaaaggtacgggtccgacaaggatgtgggtggtcctcaccagtggtcaccacagggggctggtgtaggacatcg  
ggagaaaggtacagtgctgacaaggatgtgggtggtcctcaccagtggtcaccacagggggctggtgtaggacatcg  
ggggagagatacagtgctgacaaggatgtgggtggtcctcaccagtggtcaccacagggggtggtgtaggacatcg

cctatattctcaagaagatgcgccgggccaatgtggtgggagagcagactctgggaggggccctagatctccggaa  
cctacatcctcaagaagatgcgccgggccaatgtggtgggagagcagactctgggaggggccctagatctccggaa  
cctatattctcaaa cagatgcgcaaggccaatgtggtgggtgagcagactgaaagtggtgctcctggacctccagaa  
tctacatcctcaagcagatgggcaaggccaatgtggtgggtgaa cgtactgggggggtctcctggacctccagaa  
tctacatcctcaaa caaatgcgcaaggccaatgtggtgggagcagcagactgaggggggccttggacctccagaa  
cttatattctcaaa cagatggacagggccaatgtggtggagcagaa cggactgaggggggccttggacctccagaa  
cctacatcctcaaa cagatgcgcaaggccaatgtggtgggagcagaa cgtgagggggtgctcctggacctccagaa  
cttatattctcaaa cagatgcgcaaggccaatgtggtgggtgagcagaa cgtgagggggtgctcctggacctccagaa  
cttatattctcaaa cagatgcgcaaggccaatgtggtgggtgagcagactgaggggggcctcctggacctccagaa  
cttatattctcaaa cagatgcgccgggctatgtggtgggagcagcagactgagggggctgctcctggacctccagaa  
cgcaatccttaagcagatgcgcaaggccaatgtggtgggagcagcagactgaggggaggggccttggacctccggaa  
ttcaatccttaagcagatgggcaaggccaatgtggtgggagaaagcaggaggcaggtgctcctggacctccagaa  
tctacatcctcaagcagatgggcaaggccaatgtggtgtaggagcagcagagggtggtgctcctggacctccagaa

gctgcatcggctcagtgacttttcatcactgtgccgtgtcacgctccctgagccccttgggggggagtg  
gctcgtatggctcagtgacttttcatcactgtgccgtgtccgttctctgagcccctcagtggggggagc  
actgaggataggccagtcacacttttctcacaactgtcctcctcagactgctcctggggccgatgggtggaggtggc

gctaaggatagccaactctgacttcttcctcactctacctgtgtccaggtccttggggcctctgggtggaggcacc  
gataggccactctgacttcttcctcactctgcctgtgtctaggtccttagggcccctgggcgggggaagccagaca  
gataggccagtctgacttctttctcacctgtcccgtgtccaggtccttggggcccctgggcaagggcagtcagact  
gataggccagtctgacttctttctcacctgtcccgtgtccaggtccttggggcccctgggcaagggcagccagact  
gataggccagtccgacttctttctcacctgtcctgtgtccaggtccttggggcccctgggtgagggcagccagaca  
gataggccagtccgacttcttcctcacctgtcccgtgtccaggtccttgggtctgcgcgaggtcctcatgcataac  
gataggccagtctgacttcttcctcactctgcctgtgtctaggtccttgggggctctgggtgggggcaggcagaca  
gataggccagtctgacttcttcctcactctgcctgtgtccaggtccttgggggactctgggcgggggcagccagaca  
gataggcgagtctgacttcttcttcacgtgtcccgtgtccaggtccttggggccccttgggtggaggcagccagacg  
gatagggtcactctgatttctttctcactctgcctgtgtccaggtccttggggccttgggcagggggaagccagaca  
aatagggtcactcagacttcttttctcactctgcctgtgtccaggtcactggggccccttaggcagggggaagccagaca

The tree estimation using the parsimony, maximum likelihood, UP-GMA, and neighbor-joining methods was carried out using PHYLIP (Phylogeny Inference Package, Felsenstein (1980–2000)). This package uses the Kimurs two-parameter model as the stochastic process in the maximum likelihood approach. The trees found are as follows. It is interesting to note their differences and their similarities.

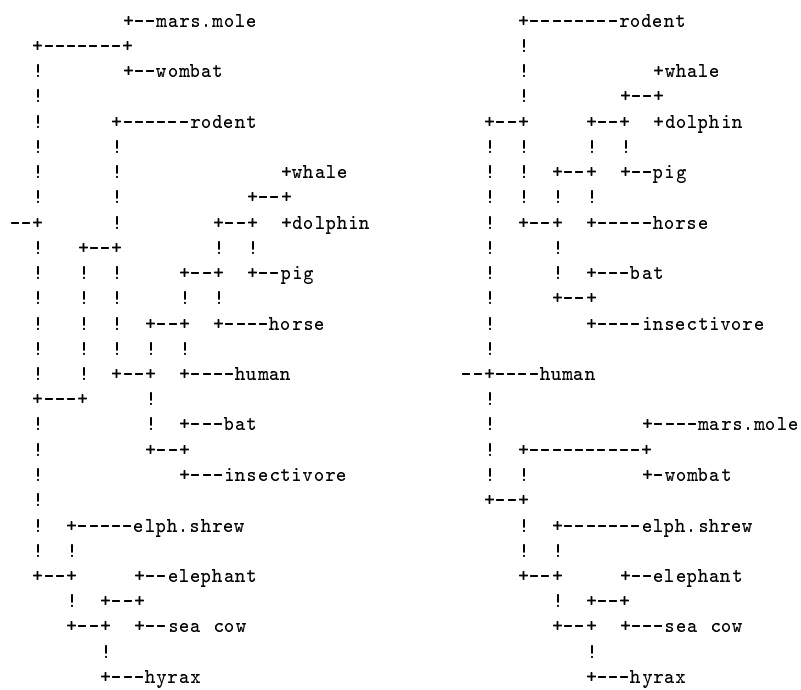


Figure 1: UPGMA and neighbor-joining trees

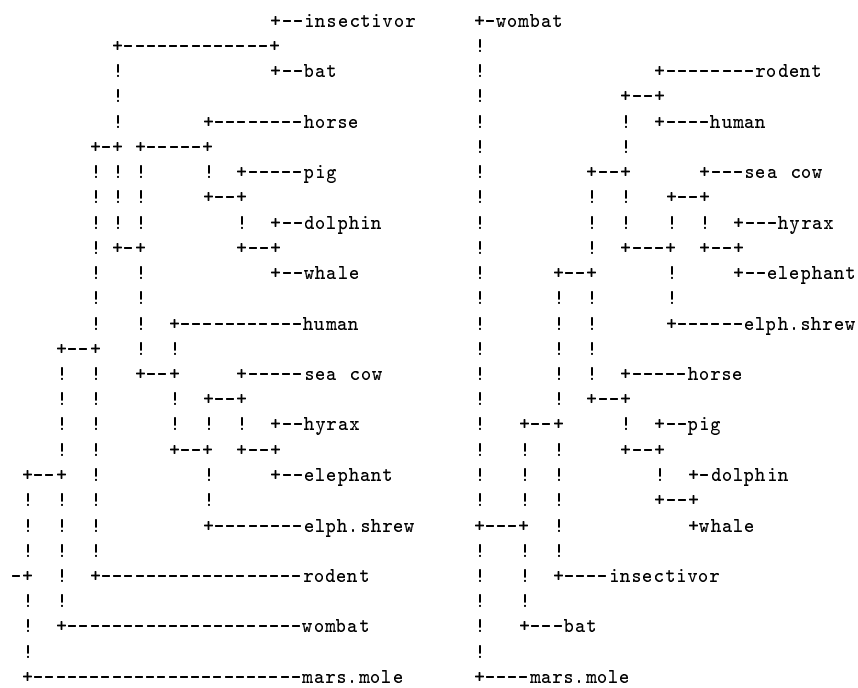


Figure 2: Parsimony and Maximum Likelihood trees



there is a very good alignment. How do we allow for giving ourselves the chance to insert this gap in our statistical assessment of the significance of the alignment?

As another problem, it was in effect assumed that the two sequences are already aligned so that a “straightforward” test can be done. But in practice, what often happens is that the first sequence is only a sub-sequence of a much longer sequence  $A$ , and the second sequence is only a sub-sequence of a much longer sequence  $B$ , and these two sub-sequences represent the best match between any part of  $A$  and any part of  $B$ . In this case the statistical test has to recognize this and make allowance for it in the significance level that it attributes to the match. Thus if sequences  $A$  and  $B$  are each several thousand nucleotides long, the match found above, as the best possible between these two sequences, would no longer be significant. How do we allow for this effect?

Statistics also concerns the *optimal* methods of analyzing data. A significant component of this optimality requirement is the appropriate choice of what is to be computed from the data in order to carry out the statistical analysis. Should we focus on the total number of matches (in this case, 11) between the two sequences? Should we focus on the size of the longest observed run of matches (here 3, occurring in positions 9–11 and also positions 23–25)? BLAST, the most frequently used statistical method for assessing the similarity of two sequences, uses neither of these quantities. Why?

We address these questions in the material below.

We first distinguish between *global* and *local* alignments. Our concept of these will be rather different from that given in the “algorithmic” lectures. In our case, we think of a global alignment as the unique alignment of two sequences of the same length. If the alignment in (8) is global, then statistical theory together with the binomial distribution leads directly to the optimal test of similarity between the two sequences. This is in effect the test given above: we count the number of matches of nucleotides in corresponding positions between the two alignments, and if this is significantly large we say we have evidence that the alignment is non-random.

The above leaves several matters unresolved. First, we have to make assumptions about *independence* of the nucleotides at the various sites, and second we have to make assumptions about the frequencies, or probabilities, of the various nucleotides. The calculation 0.04 made in connection with (8) assumed that the nucleotides at the different sites are independent of each other, and that the probability that any nominated nucleotide occurs in any specific position is  $1/4$ . In practice these might not be reasonable assumptions.

Next, how do we do the calculations when we allow ourselves the luxury of inserting as many gaps as we like, in one or other sequence, so as to maximize the number of matches? For example, consider the match of two sequences of tosses of two fair coins. We say that a match occurs at toss  $j$  if the outcome of toss  $j$  on coin 1 is the same as the outcome of toss  $j$  on coin 2 (that is, both heads or both tails.) We might get, with 20 tosses of each coin:-

coin 1 : *H H T T H H H T H T T H T T T H H T H T*

coin 2 : *H T H T T T H T T T H H H T T H T H T T*

(10 matches)

It is easy to use the binomial distribution to assess whether then number of matches is significantly large compared to what is expected if the two sets of tosses are random with respect to each other. With the above data it is not: the number of matches is, as it happens, equal to the mean value of 10 expected under randomness.

By inserting gaps we can increase the number of matches from 10 to 15:-

coin 1 : *H - H T T H H H T H T T H - - T T T H - H T H T*

coin 2 : *H T H T T T H - T - T T H H H T T - H T H T - T*

(15 matches)

How can we test whether the observed number of 15 is significantly large? No-one knows the answer to this question, since the probability distribution of the number of matches when optimal insertion of gaps is allowed is not known.

All the above refers to global alignments. It might be claimed that we should focus instead on local alignments. The reason for this view is that evolution causes gaps, deletions and various other changes, and we should thus look only at local alignments. For example, we could compare two DNA sequences, and look at the length of the longest subsequence in which there are no mismatches. In the case of the comparison in (8), this length is 3, as noted above. More generally we could look at the length of the longest subsequence in which there are at most (say) 2 mismatches. We do not consider

these approaches, since in practice the two sequences that we wish to compare are rarely aligned, and a more general approach, which automatically has to use local alignments, is needed.

So finally we consider cases when there is no natural alignment between the two sequences being compared. One obvious reason why this might happen is that the two sequences might be of quite different lengths. This in fact happens in the comparison of a query sequence, of length maybe 1000 nucleotides, with a data base sequence of length maybe  $10^9$  nucleotides. This is the problem addressed by BLAST, and we now outline the BLAST approach to this problem. In doing so we will consider the case of comparing a query amino acid sequence with an amino acid database sequence, which is the most frequently arising case in practice.

We start with a (partial) BLAST output.

BLASTP 1.4.10MP-WashU [29-Apr-96] [Build 22:25:52 May 19 1996]

Query= gi|557844|sp|P40582|YIV8\_YEAST HYPOTHETICAL 26.8 KD PROTEIN IN HYR1  
3'REGION.

(234 letters)

Database: SWISS-PROT Release 34.0  
59,021 sequences; 21,210,388 total letters.

-----				
			Smallest	
			Sum	
			High	
			Probability	
			P(N)	
			N	
Sequences producing High-scoring Segment Pairs:				
sp P46429 GTS2_MANSE	GLUTATHIONE S-TRANSFERASE 2 (EC 2....	53	0.010	3
sp P46420 GTH4_MAIZE	GLUTATHIONE S-TRANSFERASE IV (EC 2...	70	0.14	1
sp P41043 GTS2_DROME	GLUTATHIONE S-TRANSFERASE 2 (EC 2....	54	0.19	2
sp P34345 YK67_CAEEL	HYPOTHETICAL 28.5 KD PROTEIN C29E4...	50	0.42	2
sp Q04522 GTH_SILCU	GLUTATHIONE S-TRANSFERASE (EC 2.5....	62	0.87	1

-----  
>sp|P46429|GTS2\_MANSE GLUTATHIONE S-TRANSFERASE 2 (EC 2.5.1.18) (CLASS-SIG).  
Length = 203

Score = 53 (24.4 bits), Expect = 0.010, Sum P(3) = 0.010  
Identities = 10/19 (52%), Positives = 15/19 (78%)

Query: 167 ISKNNGYLVDGKLSGADIL 185  
I+KNNG+L G+L+ AD +  
Sbjct: 136 ITKNNGFLALGRLTWADFV 154

Score = 46 (21.2 bits), Expect = 0.010, Sum P(3) = 0.010  
Identities = 8/21 (38%), Positives = 13/21 (61%)

Query: 45 PELKKIHPLGRSPLEVDRE 65  
PE K P G+ P+LE+ ++  
Sbjct: 39 PEFKPNTPFQMPVLEIDGKK 59

Score = 36 (16.6 bits), Expect = 0.010, Sum P(3) = 0.010  
Identities = 8/26 (30%), Positives = 12/26 (46%)

Query: 202 EDYPAISKWLKTITSEESYAASKEKA 227  
E YP K ++T+ S A + A  
Sbjct: 173 EQYPIFKPIETVLSNPCLKAYLDSA 198

>sp|P46420|GTH4\_MAIZE GLUTATHIONE S-TRANSFERASE IV (EC 2.5.1.18) (GST-IV)  
(GST-27) (CLASS PHI).  
Length = 222

Score = 70 (32.3 bits), Expect = 0.15, P = 0.14  
Identities = 17/56 (30%), Positives = 27/56 (48%)

Query: 18 RLLWLLDHLNLEYEIVPYKRDANFRAPPELKKIHPLGRSPLEVDRETGKKKILA 73  
R L L+ ++YE+VP R PE +P G+ P+LE D + + +A  
Sbjct: 18 RALLALEEAGVDYELVPMRQDGDHRRPEHLARNPFQKVPVLEDGDLTLFESRAIA 73

```
>sp|Q04522|GTH_SILCU GLUTATHIONE S-TRANSFERASE (EC 2.5.1.18) (CLASS-PHI).
      Length = 216
```

```
Score = 62 (28.6 bits), Expect = 2.1, P = 0.87
Identities = 15/43 (34%), Positives = 21/43 (48%)
```

```
Query:    18 RLLWLLDHLNLEYEIVPYKRDANFRAPPELKKIHPLGRSPLLE 60
          R+L L +LE+E VP A P ++P G+ P LE
Sbjct:    15 RVLVALYEKHFVFPIDMGAGGHKQPSYLALNPFQVPALE 57
```

---

Matrix name	Lambda	K	H
BLOSUM62	0.320	0.137	0.401

What does all this mean? The BLAST output above relates to amino acid sequences, but it is easiest to explain what is happening by considering DNA (i.e. nucleotide) sequences.

Consider the comparison of two DNA sequences given in (8) above. Suppose we give a score +1 if the two nucleotides in corresponding positions are the same and a score of -1 if they are different. As we compare the two sequences, starting from the left, we can calculate the accumulated value of these scores. This accumulated value performs a simple random walk, with steps of  $\pm 1$ . The walk in the above example is depicted graphically in Figure 3 (see next page). The filled circles in this figure relate to (downwards) ladder points in the walk, that is to points in the walk lower than any previously reached point.

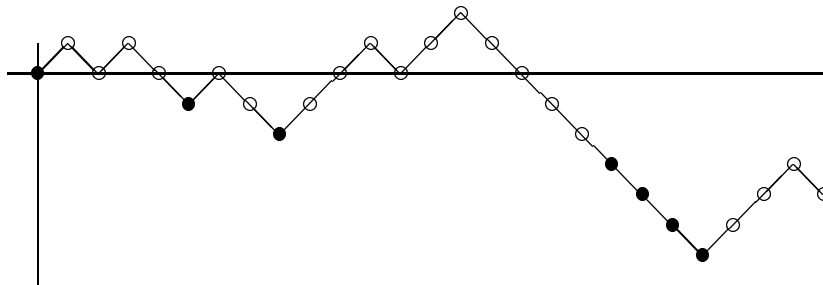


Figure 3:

We use the term “upwards excursion” to describe that part of the walk between two consecutive ladder points, relative to the first such point. We will be interested in the maximum height (relative to the first of the two ladder points) of this upwards excursion. (If the walk proceeds immediately from one ladder point to the next, as happens often, this height is taken as 0.) Simple BLAST theory uses as test statistic the maximum height achieved by the various excursions relative to the various ladder points from which they started. In Figure 3 the observed values of the heights of the various upward excursions shown are, respectively, 1, 1, 4, 0, 0, 0, 3, with a maximum of 4.

**In the above case, this value of 4 would be “SCORE”, or “HIGH SCORE”, in the BLAST output.**

This is a quite different statistic than a total number of matches or the length of the longest sequence of matches, both discussed above.

In practice BLAST theory relates to cases that are much more complicated than this simple example. It is most often applied, for example, to the comparison of two protein sequences. In this case it uses scores (that is, step sizes in the random walk) given by the entries in a  $20 \times 20$  *substitution matrix* such as those given in the BLOSUM62 substitution matrix shown in Table 1 on the following page. (The twenty letters in this table refer to the twenty amino acids, in a generally accepted notation.) The derivation of these scores derived from statistical theory and will not be discussed here - we will just take them as given.

For example, if the score for any amino acid comparison is found

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-3	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Table 1: The BLOSUM62 substitution matrix

from the BLOSUM62 substitution matrix in Table 1, then for the alignment

$$\begin{array}{l} T Q L A A W C R \dots \\ R H L D S W R R \dots \end{array} \quad (11)$$

the respective scores are  $-1, 0, +4, -2, +1, +11, -3, +5, \dots$ , and therefore the graph of the random walk described by the accumulated score goes through the points

$$(1, -1), (2, -1), (3, 3), (4, 1), (5, 2), (6, 13), (7, 10), (8, 15), \dots \quad (12)$$

As for the simple  $\pm 1$  walk, the quantity “SCORE” (or “HIGH SCORE”) used in BLAST as applied to amino acid data is the height of the largest upwards excursion following a ladder point relative to that ladder point, before the walk reaches the next ladder point. Since the possible steps in the walk depend on the substitution matrix chosen, it is necessary for the statistical theory of BLAST to consider arbitrary scoring schemes and thus aspects of

the theory of random walks with an arbitrary array of possible step sizes. This theory is very complicated.

The only other quantity that we consider in the BLAST output is “ $P$ ”. This is a  $P$ -value. This is the probability, assuming that the two sequences being compared are completely random with respect to each other, of getting a value of “SCORE” equal to or larger than that observed. A sufficiently low  $P$ -value leads us to claim that there *is* a significant similarity between the two sequences.

### Complications

The above calculations ignore many complications which arise in practice in BLAST. These complications arise in part because focusing on the largest score loses information provided by other high scores. Here BLAST uses the “SUM” of one, two, three, or some other small number of such scores. The quantity “ $N$ ” in the BLAST output indicates how many such scores are used in the sum. Another complication arises because the database with which the query sequence is compared consists in practice of a large number of comparatively short sequences, of varying lengths, and with no automatic alignment of the query with any part of these. These complications are all allowed for in the BLAST calculations.