Pairwise distance methods

Peter Beerli

October 26, 2005

When we infer a tree from genetic data using parsimony we minimize the amount of change along the branches of the tree. Similarly when we use the likelihood principle we minimize change conditional on a specific mutation model. The mutation model is crucial. The model can take into account that we do not observe all substitution events, because recent events might hide ancient events. Parsimony is therefore undercounting the number of changes and so might have a shorter tree than the true tree. Likelihood does not escape this problem either, we have either a tree that is shorter or the same length as the true tree.

An alternative to likelihood or parsimony is an approach based on evolutionary distances between a pair of sequences, where the distance is accounting for all unseen events, for example using similar mutation models as likelihood.

Pairwise distance methods are not so popular anymore because the are outperformed by likelihood methods. Pairwise methods evaluate all pairs of sequences and transform the differences into a distance. This essentially is a data reduction from a possibly many state difference to a single number. Combining these distances to estimate a tree must be less powerful than the full likelihood approach. In addition, an identical distance can be generated from different sequence pairs and once we only analyze the distance matrix that difference is lost. Using the number of different sites as a distance measure makes quickly clear that we can arrive at the same measure from different sequences.

Distance methods have still their merit because once the distance matrix is calculated the tree building can be very fast and under many circumstances are the trees generated with such methods not all that terrible and often are identical to the likelihood tree.

Table 1: Example of a problematic data sets for distance methods, the used distance is simply counting sites that are different between pairs.

Individual	Sequence
one	ATTAGC
two	ATTGGC
three	ATGGGC
four	GTGGGC

→

Individual	one	two	three	four
one	-	1	2	3
two		-	1	2
three			-	1
four				-

1 Additive distances

If we could estimate branch length on a tree with absolute certainty all distances on a tree would be additive, for example the distances between all the tips of the tree in Figure 1 are

$$d_{AB} = v_1 + v_2$$
$$d_{AC} = v_1 + v_3 + v_4$$
$$d_{AD} = v_1 + v_3 + v_5$$
$$d_{BC} = v_2 + v_3 + v_4$$
$$d_{BD} = v_2 + v_3 + v_5$$
$$d_{CD} = v_4 + v_5$$

Additive trees satisfy the *four-point metric condition*, for any four taxa A, B, C, and D,

$$d_{AB} + d_{CD} \leq \max(d_{AC} + d_{BD}, d_{AD} + d_{BC})$$

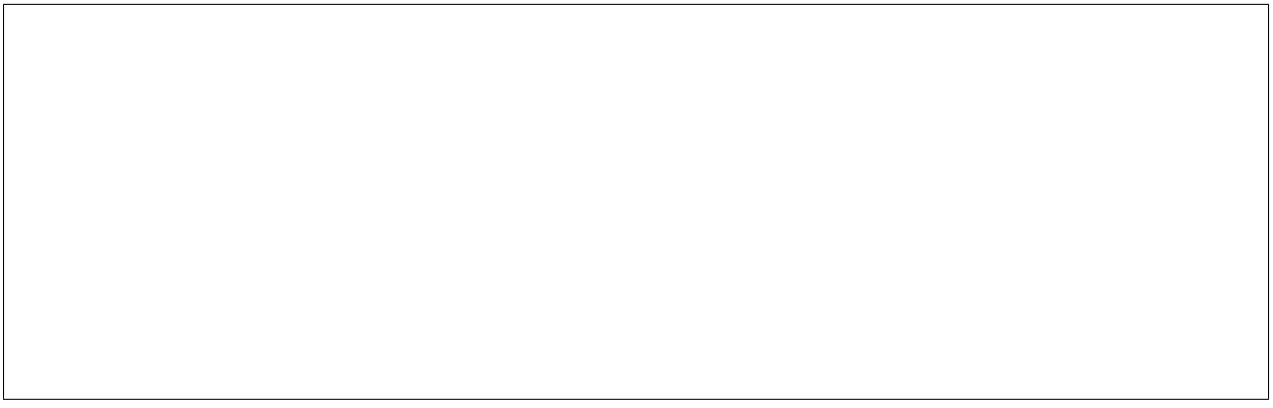

Figure 1: Addititive tree.

2 Ultrameric trees

Ultrameric trees have additive properties and also obey ultrameric properties:

$$\begin{aligned} d_{AB} &= v_1 + v_2 + v_3 \\ d_{AC} &= v_1 + v_2 + v_4 \\ d_{BC} &= v_3 + v_4 \\ v_3 &= v_3 \\ v_1 &= v_2 + v_3 = v_2 + v_4 \end{aligned}$$

Ultrametric trees are often expressed as molecular clock tree, also such trees do not necessarily assume that there is linear change of the mutation rate through time.

2.1 Additive tree method

The discussion about additive trees all real great but unfortunately due the finiteness of the data there will be random fluctuations that will result in deviations from the perfect additivity. Methods were derived that used this deviation (or distortion) from the perfect additivity as an optimality criterion. Fitch and Margoliash and others derived a method that minimizes the following objective function

$$E = \sum_{i=0}^{T-1} \sum_{j=i+1}^{T} w_{ij} |d_{ij} - p_{ij}|^{\alpha} \tag{1}$$

where E is the error of fitting the distance estimates and the tree, T is the number of taxa, d_{ij} is a distance measure between the taxa i and j; p_{ij} is the length of the path connecting i and j; w_{ij} is a

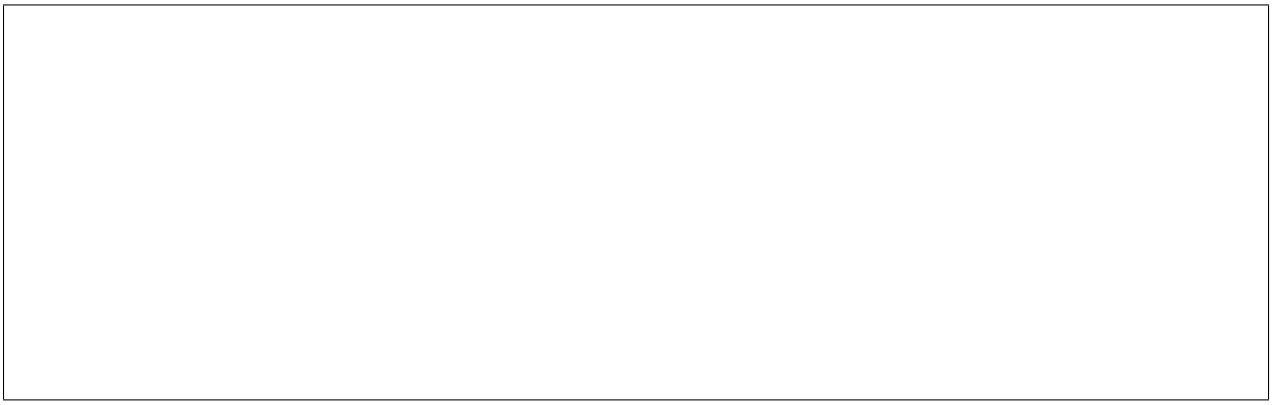

Figure 2: Ultrameric tree.

weighting to separate the taxa i and j, and α can take values such as 1 or 2. With a value of $\alpha = 2$ one uses a least-squares minimization. If $\alpha = 1$ then the absolute differences will be minimized. The weightings w_{ij} can accommodate previous knowledge about the data, but often it is unclear what we know. The most common weightings are

$w_{ij} = 1$	All errors of the distances are the same
$w_{ij} = 1/d_{ij}^2$	percentage error is the same
$w_{ij} = 1/d_{ij}$	square root of the error is the same
$w_{ij} = 1/\sigma^2$	error is inversely correlated to the expected variance of the measurements of d_{ij}

Missing data can be easily accommodated by setting the w_{ij} involving missing data to zero. If the variation of σ^2 is known this weighting would be preferred. Problems might arise when identical sequences are in the dataset.

For an unrooted tree there are $2T - 3$ independent branches defining the p_{ij} values, and there are $T(T-1)/2$ distinct distances. We can represent the tree as an indicator matrix A of $T(T-1)/2$ rows and $2T-3$ columns. An element of this matrix is 1 if the branch k is part of connecting taxon i to taxon j otherwise it is zero. The p values can now be expresses as the

$$Av = p$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} = \begin{pmatrix} p_{AB} \\ p_{AC} \\ p_{AD} \\ p_{BC} \\ p_{BD} \\ p_{CD} \end{pmatrix} \tag{2}$$

If the distances were additive then $p = d$ for all pairs and we could solve the equation directly. but due to the imperfection of the data we use formula 2 to eliminate p in formula 1. Assuming $\alpha = 2$ and $w_{ij} = 1$ we can find the branch length

$$v = (A^T A)^{-1}(A^T d)$$

Still we need an appropriate search strategy to search for the best tree, but we can use any heuristic strategy can be used to do that.

Sometimes the solution of the above equation results in negative branch length, several strategies are described but most simply the negative branch length are set to zero without adjust the other branch lengths. If the data does not propose negative branch lengths the error estimate E will be accurate but when negative branch length are encountered the estimates of E will be too low.

Errors using this procedure come from two two sources: (1) we assuem each pairwise distance is independent of each other, this is certainly untrue. (2) any error in the data will be amplified by the pairwise use and underestimates similarity by state compared to similarity by descent (homoplasy).

2.2 Minimal evolution

Minimal evolution sets the weights to 1 and $\alpha = 2$ and simply assumes that the sum of all branches are minimized (instead of all individual branches by itself)

$$L = \sum_{i1}^{2T-3} |v_{ij}| \tag{3}$$

This was described by Kidd and Sgaramella-Zonta (1971). This was (re)described in 1992 by Rzhetsky and Nei as *minimal evolution* in a very similar form

$$L = \sum_{i1}^{2T-3} v_{ij} \tag{4}$$

the newer version takes the absolute value which seems a big drawback but under realistic condition is often of no big concern. Some proponent claim that ME is superior to other techniques, although others have shown that the simple Fitch-Margoliash method works as well as ME with enough data.

3 Distance measures

To get a distance we need to have some model of change between the two sequences and a possible way to about this is to look at the frequency of changes between the two taxa:

$$F_{XY} = \begin{pmatrix} n_{AA}/N & n_{AC}/N & n_{AG}/N & n_{AT}/N \\ n_{CA}/N & n_{CC}/N & n_{CG}/N & n_{CT}/N \\ n_{GA}/N & n_{GC}/N & n_{GG}/N & n_{GT}/N \\ n_{TA}/N & n_{TC}/N & n_{TG}/N & n_{TT}/N \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$$

Ambiguities are not coded correctly usign the above scheme, the uncertainties might be counted more than once, for example a Purine would be an A or a G and will overestimate the similarity.

The most simply distance is the *p-distance* or dissimilarity D. It is

$$\begin{aligned} d_{XY} =& b + c + d + e + g + h + i + j + l + m + n + o \\ =& 1 - (a + f + k + p) \end{aligned}$$

The mutation models specified earlier can be used to generate distances and for example the Jukes-Cantor distance is

$$D = 1 - (a + f + k + p)d_{XY} = -\frac{3}{4}ln(1 - \frac{4}{3}D)$$

4 Actual strategies to find optimal trees with distance methods

4.1 Neighbor-Joining

The neighbor-joining technique is kin to clustering technology. It was developed by Saitou and Nei (1987) and uses a distance matrix to construct a tree. It assumes that the data are close to an additive tree, but it does not assume a molecular clock. NJ is a special case of the star decomposition algorithm described earlier. I start wit a star phylogeny and then uses the smallest distance in the distance matrix to find the next two pairs move out of the multifurcation. The none need s to recalculate the distance matrix that now contains a tip less.

Algorithm 1 Neighbor joining

1. Give a matrix of pairwise distances (d_{ij}), for each terminal node I calculate its net divergence r_i from all other taxa using the formula

$$r_i = \sum_{k=1}^{N} d_{ji}$$

where N is the number of terminal nodes in the current matrix. Note that the assumption that $d_{ii} = 0$, otherwise the summation would need to skip over $k = i$.

2. Create a rate corrected distance matrix M in which the elemens are defined as

$$M_{ij} = d_{ij} - (r_i - r_j)/(N-2)$$

only states $i \neq j$ are interesting, even only the minimum needs to be known.

3. define a new node u whose three branches join nodes i, j and the rest of the tree. Define the length of the tree branches from u to i to j as

$$v_{iu} = \frac{\frac{d_{ij}}{2} + (r_i - r_j)}{2(N-2)} v_{ju} = d_{ij} - v_{iu}$$

4. Define the distance from u to each other terminal node

$$d_{ku} = (d_{ik} + d_{jk} + d_{ij})/2$$

5. Remove distance to nodes i and j from the data matrix and decrease N by 1.

6. If more than two nodes remaining, go back to step 1. Otherwise the tree is full defined except for the last branch length which is

$$v_{ij} = d_{ij}$$