This is a description of the method behind `weighbor 2.15`. The only changes from 2.14 is a new parameter ρ, the transition/transversion ratio, whic enters into σ_{total}.

Basic formulas:

First, let D be the dissimilarity (equivalent to the Hamming measure) corresponding to a given temporal (additive) distance d. Inverting the Jukes-Cantor formula gives

$$D(d) = \frac{b-1}{b}(1 - e^{-db/(b-1)}) \tag{1}$$

Where b is the number of allowed characters; in the Jukes-Cantor DNA model, $b = 4$. Fluctuations in D are simply binomial; therefore

$$\sigma_D^2 = D(1-D)/L \tag{2}$$

where L is the length of the sequence. By propagation of errors, the total expected variance of observed distance d is

$$\sigma_{\text{t}}{}^2(d) \equiv \sigma_{\text{total}}^2(d) = e^{2db/(b-1)} D(d)(1 - D(d))/L \tag{3}$$

This can be generalized by allowing transitions to be faster than transversions. The new formula is

$$\sigma_{\text{t}}{}^2(d) \equiv \left[L \left(\frac{(p'_P)^2}{p_P} + \frac{(p'_Q)^2}{p_Q} + \frac{(p'_P + p'_Q)^2}{1 - p_P - p_Q} \right) \right]^{-1} \tag{4}$$

where

$$p_P \equiv \left(1 - \frac{1}{c}\right) \left[\frac{c}{b} - e^{-(c - \frac{1}{1+\rho})d/(c-1)} + \left(\frac{b-c}{b}\right) e^{\frac{-bd}{(1+\rho)(b-c)}} \right] \tag{5}$$

$$p'_P \equiv \frac{\partial p_P}{\partial d} = \left(1 - \frac{1}{c}\right) \left(\frac{c - \frac{1}{1+\rho}}{c-1} e^{-(c - \frac{1}{1+\rho})d/(c-1)} - \frac{1}{1+\rho} e^{\frac{-bd}{(1+\rho)(b-c)}} \right) \tag{6}$$

$$p_Q \equiv \left(\frac{b-c}{b}\right) \left(1 - e^{\frac{-bd}{(1+\rho)(b-c)}}\right) \tag{7}$$

$$p'_Q \equiv \frac{\partial p_Q}{\partial d} = \frac{1}{1+\rho} e^{\frac{-bd}{(1+\rho)(b-c)}} \tag{8}$$

where b remains the alphabet size, c (where $c \leq b$) is a new input paramter for the number of states within each transition category ($b = 4$, $c = 2$ for Kimura 2-parameter), and ρ is the rate of transitions relative to transversions ($\rho = (c-1)/(b-c)$ should give our old Jukes-Cantor result, which is independent of c; for K2P this happens at $\rho = .5$). If I recall, our default value of b is 4. I suppose our default for c should be 2. And the default for ρ should be $1/(b-2)$. There should be a warning if someone changes c but tries to use the default ρ; but it's OK to change ρ and leave c alone.

If we want to consider the noise in the distance between taxa i and j, we use the previous formula. If we want to consider only "non-additive" noise, which we do, we must also specify the point where the non-additivity takes place, or, equivalently, the branch lengths in which we ignore the additive noise. We typically will do this by specifying a third taxon k, and compute non-additivity about the point where the path from k joins the path connecting i and j. We may call this point the internal node l. The formula for the "non-additive" noise resulting when branches of length d_{il} and d_{lj} are joined is, in the Jukes-Cantor model, assuming long sequences:

$$\sigma_{ij;k}^2 \equiv \sigma_{\text{n.a.}}^2(d_{il} + c_i, d_{lj} + c_j) \equiv \sigma_{\text{t}}{}^2(d_{ij} + c_i + c_j) - \sigma_{\text{t}}{}^2(d_{il} + c_i) - \sigma_{\text{t}}{}^2(d_{lj} + c_j) \tag{9}$$

the c's are zero and can be ignored when i and j are actual sequences, and we use

$$d_{il} = \begin{cases} 0 & \text{when } d_{jk} > d_{ij} + d_{ik} \\ d_{ik} & \text{when } d_{ij} > d_{ik} + d_{jk} \\ (d_{ij} + d_{ik} - d_{jk})/2 & \text{otherwise} \end{cases} \tag{10}$$

$$d_{lj} = d_{ij} - d_{il} \tag{11}$$

The first two cases in equation **??** only arise when the triangle inequality is violated.

Algorithm Summarized:

Ideally, all pairs are scored by a single function $R(i,j)$ which is symmetric in i and j. The pair with the smallest value gets joined (note $R(i,j) \geq 0$).

In practice, conflicting pairs are scored according to an estimate of their relative log-likelihood R(i,j,j'). Using a tournament one can define $q[i]$, the taxon that makes the best join to i. Thus the full $R(i,j)$ is evaluated only for pairs that are better than all conflicting pairs, i.e., $R(i,q(i))$, which allows an N^3 implementation.

Algorithm:

Assume i and j are neighbors. The distance from i to the most recent common ancestor of i and j is

$$b_{i;j} \equiv \begin{cases} (\Delta b_{ij} + d_{ij})/2 & \text{for } d_{ij} \geq |\Delta b_{ij}| \\ 0 & \text{for } d_{ij} < -\Delta b_{ij} \\ d_{ij} & \text{otherwise} \end{cases} \tag{12}$$

where Δb_{ij} is $b_{i;j} - b_{j;i}$, and the second and third cases only come into play when the estimate of Δb_{ij} violates the triangle inequality.

An important quantity in the algorithm is Δb_{ij}. It is the amount by which i is farther from the other taxa than j. Actually, this definition is only unambiguous when i and j are neighbors, but the equation below is always defined. If one assumes a star phylogeny, then the following is the generalized least-squares estimate of Δb_{ij}.

$$\Delta b_{ij} = \frac{1}{s_{ij}} \sum_{k \notin \{i,j\}} \frac{1}{\sigma^2_{ik;j} + \sigma^2_{jk;i}} (d_{ik} - d_{jk}) \tag{13}$$

where

$$s_{ij} \equiv \sum_{k \notin \{i,j\}} \frac{1}{\sigma^2_{ik;j} + \sigma^2_{jk;i}} \tag{14}$$

There is an iterative improvement that can be done here, but it makes it harder to implement an N^3 version, and doesn't seem to improve accuracy or bias a great deal. This is done when using the -r option, described at the end of this document.

Which pair to join?:

$R(i,j)$ is an approximation of the loglikelihood of i and j being toplogical neighbors, with a non-negative branch length separating them from the rest of the tree:

$$R(i,j) \equiv s_{ij}(\Delta^2 b_{ij} - \Delta b^2_{ij})x - 2\ln\left[\frac{1}{2}\text{erfc}\left(-z(i,j)/\sqrt{2}\right)\right] \tag{15}$$

where x has to do with the model for correlations. For a star phylogeny, $x = 1$. Using our default parameters,

$$x \equiv 1/(N-3) \tag{16}$$

which corresponds to $N-3$ taxa being highly correlated, and one being different. I would like to try an option (-Xx, maybe?) where

$$x \equiv 2/(N-2) \tag{17}$$

which corresponds to dividing the taxa evenly into two highly correlated groups.

To complete our definitions

$$\Delta^2 b_{ij} \equiv \frac{1}{s_{ij}} \sum_{k \notin \{i,j\}} \frac{1}{\sigma^2_{ik;j} + \sigma^2_{jk;i}} (d_{ik} - d_{jk})^2 \tag{18}$$

and the $z(i,j)$, the number of standard deviations by which the length of the internal branch separating the joined taxa from the rest of the tree, is defined by below.

The first term in this expression is essentially a χ^2 measure of the extent to which all possible estimates of of Δb_{ij} give the same value, which they should do if and only if i and j are neighbors. This is equivalent to twice the log-likelihood if you approximate the distance fluctuations as Gaussian (true in the limit of long sequences) and you assume a star phylogeny.

Actually, if you assume a star phylogeny, then there is no factor of $N-3$ in the denominator. We put that in, which has the effect of assuming there are only 2 independent branches of the tree (not counting i and j), and all other taxa are closely clustered at the tips of those branches. This assumption biases us toward NJ-like behavior, because NJ is like an approximation of the erfc term alone.

Note, I am very open to considering other possible assumptions here. I believe that removing the $N-3$ will make us vulnerable to missing easy trees that NJ gets right. If we knew what kinds of trees to expect we could probably come up with some simple function of N that would do better than $N-3$. Or, we could try to estimate the amount of correlation of branches in the data and try to reflect that. It would be interesting to play around and see how close we can get to DNAML if we chose a value optimized for a given tree.

The erfc term reflects the requirement that the distance between the point where the taxa join and the next closest internal node must be positive.

Note that I had said to put an additional factor of $N-3$ in front of the ln. This is wrong. What one could consider doing is

$$\frac{1}{2}\mathrm{erfc}(-z(i,j)/\sqrt{2}) \to 1 - [\frac{1}{2}\mathrm{erfc}(z(i,j)/\sqrt{2})]^{N-3} \tag{19}$$

which recognizes that we are choosing an extreme value for z (see below). This assumes no additional correlation among the remaining branches. If you assume maximal correlation, then the $N-3$ is replaced by 1, and you recover the previous form (perhaps up to a constant). Using the 1 is also equivalent to ignoring the fact that z actually comes from an extreme value distribution.

I would also like to experiment with replacing the $\frac{1}{N-3}$ factor in the first term by $\frac{1}{N-3}\frac{N-2}{2}$.

We choose the following, extreme value for z (because I don't think we can find a good average value on order N steps):

$$z(i,j) \equiv \min_{kl} \frac{PQ(ij,kl)}{\sqrt{\sigma^2_{PQ}(ij,kl) + (\sigma^2_{ij;k} + \sigma^2_{ij;l})/8}} \tag{20}$$

where the $\min_{kl}$ is approximated below, and the following also get used in the full, slow z-score:

$$\sigma^2_{PQ}(ij,kl) \approx [\frac{1}{\frac{1}{\min(\sigma^2_{ik;j},\sigma^2_{ik;l})+\min(\sigma^2_{jl;i},\sigma^2_{jl;k})} + \frac{1}{\min(\sigma^2_{il;j},\sigma^2_{il;k})+\min(\sigma^2_{jk;i},\sigma^2_{jk;l})}} + (\sigma^2_{kl;i} + \sigma^2_{kl;j})/2]/4 \tag{21}$$

$$PQ(ij,kl) \approx [\frac{\frac{(d_{ik}+d_{jl})}{\min(\sigma^2_{ik;j},\sigma^2_{ik;l})+\min(\sigma^2_{jl;i},\sigma^2_{jl;k})} + \frac{(d_{il}+d_{jk})}{\min(\sigma^2_{il;j},\sigma^2_{il;k})+\min(\sigma^2_{jk;i},\sigma^2_{jk;l})}}{\frac{1}{\min(\sigma^2_{ik;j},\sigma^2_{ik;l})+\min(\sigma^2_{jl;i},\sigma^2_{jl;k})} + \frac{1}{\min(\sigma^2_{il;j},\sigma^2_{il;k})+\min(\sigma^2_{jk;i},\sigma^2_{jk;l})}} - d_{ij} - d_{kl}]/2 \tag{22}$$

The $\min_{kl}$ would take N^2 to compute in full, so instead we employ a heuristic to find k and l that should nearly minimize z. We start with a trial value $k = q_2[i]$, which is the taxon that placed second in the tournament (see below) that determines $q[i]$. Then, holding k fixed, try all values $l \notin \{i,j,k\}$ to find a minimum. Taking this value of l as fixed, we now try all values of $k \notin \{i,j,l\}$. The minimum value of z is returned.

Also, the min functions inside PQ are approximations to the correlations you get when you consider 4 taxa exactly (not in a star). The result is a somewhat biased estimate, but it behaves properly in all the relevant limits.

Rather than evaluate $R(i,j)$ for all pairs, we try to estimate log-ratios $R(i,j) - R(i,j')$ for all triplets; but we don't try all triplets. We do a single-elimination tournament to define $q(i)$, the best j for i. In a contest between j and j', the winner is determined by the quantity

$$R(i,j,j') \equiv [s_{ij}(\Delta^2 b_{ij} - \Delta b^2_{ij}) - s_{ij'}(\Delta^2 b_{ij'} - \Delta b^2_{ij'})]y - 2\ln\left[\frac{\mathrm{erfc}\left(\frac{d_{ij} - \overline{\Delta b''_{jj'}} - \overline{d_{ij'}}}{\overline{\sqrt{2(\overline{\sigma^2_{ij;j'}} + \overline{1/s''_{jj'}} + \overline{\sigma^2_{ij';j}})}}}\right)}{\mathrm{erfc}\left(\frac{d_{ij'} - \overline{\Delta b''_{j'j}} - \overline{d_{ij}}}{\overline{\sqrt{2(\overline{\sigma^2_{ij';j}} + \overline{1/s''_{j'j}} + \overline{\sigma^2_{ij;j'}}))}}}\right)}\right] \tag{23}$$

where y corrects for some model of the correlations; $y = 1$ is a star phylogeny; in what we have been doing

$$y \equiv \frac{1}{N-3} \tag{24}$$

This assumes that there is a group of $N-3$ taxa that are effectively identical to eachother, and one different taxon, and that they all have equal variances. We should be able to do better, since we are most interested in the case where the one different taxon is j'. We thus redefine the $-Xn$ option to set

$$y \equiv \frac{\sigma^2_{ij';j} + \sigma^2_{jj';i} + 1/s''_{ij}}{\sigma^2_{ij';j} + \sigma^2_{jj';i} + (N-3)/s''_{ij}} \tag{25}$$

where as always

$$s''_{ij} \equiv s_{ij} - 1/(\sigma^2_{ij';j} + \sigma^2_{jj';i}) \tag{26}$$

The stuff inside the erfc is similar to the z score. The difference is that here we are using a sophisticated measure of the distance from the internal node joining i and j, to the internal node joining j' to the rest of the tree. With the z score, we are using a simpler measure, but taking the min over j. That simpler measure involves a min over k as well. It is not clear to me how robust the complex measure is; it should work well when i, j and k branch near eachother and far from the rest of the tree. The simple measure is robust but sloppy and slow.

One could do a simpler version of this complex measure by replacing the overlined quantities by the quantities themselves. Instead, we attempt to make better estimates using other information that is available. I don't know what kind of improvement this actually gives us, because we implemented this back when the method wasn't working very well. There was a bug in one of the overline quantities, which was revealed by the occasional easy tree we got wrong. I suppose it could be interesting to see how we do without the overline quantities, or with a simplified version. But

I don't thing the speedup would be dramatic, and the code seems to work as implemented. The quantities are:

$$\overline{d_{ij'}} \equiv \frac{d_{ij'}/(\overline{\sigma^2_{ij';j}} + 1/s''_{jj'}) + d_{jj'}/(\overline{\sigma^2_{jj';i}} + 1/s''_{ij'})}{1/(\overline{\sigma^2_{ij';j}} + 1/s''_{jj'}) + 1/(\overline{\sigma^2_{jj';i}} + 1/s''_{ij'})} \tag{27}$$

and similarly

$$\overline{\Delta b''_{jj'}} \equiv \frac{\Delta b''_{jj'}/(\overline{\sigma^2_{ij';j}} + 1/s''_{jj'}) + \Delta b''_{ij'}/(\overline{\sigma^2_{jj';i}} + 1/s''_{ij'})}{1/(\overline{\sigma^2_{ij';j}} + 1/s''_{jj'}) + 1/(\overline{\sigma^2_{jj';i}} + 1/s''_{ij'})} \tag{28}$$

(for documentation purposes, note that both the above quantities are derived with a Δb_{ij} in the second term, but they have opposite signs and cancel out). Note that $s''_{ij'}$ means the j term gets removed; i.e., any reference to the triplet i, j, j' is subracted off. Thus

$$\Delta b''_{jj'} \equiv [s_{jj'}\Delta b_{jj'} - (d_{ji} - d_{j'i})/(\sigma^2_{ji;j'} + \sigma^2_{j'i;j})]/s''_{jj'} \tag{29}$$

and

$$s''_{jj'} \equiv s_{jj'} - 1/(\sigma^2_{ji;j'} + \sigma^2_{j'i;j}) \tag{30}$$

and we should add EPSILON to the denominators as always. We should choose EPSILON such that $(x + 1/\text{EPSILON}) - 1/\text{EPSILON} = x$ when x is of order 1. I've had success with EPSILONs of 10^{-8} and 10^{-9}.

Going on,

$$\overline{1/s''_{jj'} + \overline{\sigma^2_{ij';j}}} \equiv \frac{1}{1/(\overline{\sigma^2_{ij';j}} + 1/s''_{jj'}) + 1/(\overline{\sigma^2_{jj';i}} + 1/s''_{ij'})} \tag{31}$$

and these $\overline{\sigma^2_{ij';j}}$ are defined below. They are probably not crucial after all, but should be better than the regular $\sigma^2_{ij';j}$, especially for short sequences. The $\overline{\sigma^2_{ij';j}}$ are

$$\overline{\sigma^2_{ij';j}} \equiv \sigma^2_{\text{n.a.}}(b_i + c_i, b_{j'} + c_{j'}) \tag{32}$$

and so on, and where the c's are the usual renormalization c's, but the b's are new, special b's defined by

$$b_j = \max[(d_{jj'} + \Delta b''_{jj'})/2, b_{\min}] \tag{33}$$

$$b_{j'} = \max[(d_{jj'} - \Delta b''_{jj'})/2, b_{\min}] \tag{34}$$

$$b_i = \frac{\max[(d_{ij} - b_j), b_{\min}]/\sigma^2_{ij;j'} + \max[(d_{ij'} - b_{j'}), b_{\min}]/\sigma^2_{ij';j}}{1/\sigma^2_{ij;j'} + 1/\sigma^2_{ij';j}} \tag{35}$$

where $b_{\min}$ is some minimum resolvable amount of distance, such as $1/(2L)$.

If $R(i, j, j')$ is negative, j wins; positive, j' wins. The point of finding $q(i)$ is that we will only evaluate the full $R(i, j)$ for $j = q(i)$.

The tournament thus defines $q[i]$ for all i. We then join the pair $(i, q[i])$ that minimizes $R(i, q[i])$. Depending on the $-q$ option, we can restrict to pairs for which $q[q[i]] == i$; when this equation is not satisfied, there may be cause for suspicion. I'm not sure what we should do here; for now let's just print something to weighbor.out (at the -vv level?). Maybe we should print a warning to stderr for now, so if it happens a lot we can look into it; but we probably would want to disable that in a release. See the options -ec and -ez below; when these are used we will probably not want to require $q[q[i]] == i$.

```
  i                 i'
    \              /
     \            /
      \          /
       \        /
\bar{i} |
        |     j
        |    /
        |   /
        |  /
        |
        |
        |
        |
```

Figure 1: This figure reminds us that i and i' join to become $\bar{\imath}$.

Renormalization:
The $\sigma^2_{ij;k}$ are computed by equation **??**. The interpretation of c_i is the amount longer the tip of the branch ending in taxon (or renormalized pair of taxa) i would be just judging by its noise. So, in the first iteration, all c_i are zero. Below we see how they renormalize.

Let's say we've decided to join taxon i with taxon i'. Let's call the newly created taxon (or, as Nei would say, 'OTU') $\bar{\imath}$. It will be located at the nearest common ancestor of taxa i and i'. The question is, what are the new distances and variances.

Actually, back up a bit. What about the distance from i (and/or i') to $\bar{\imath}$?

To a first approximation it's $b_{i;i'}$ (and/or $b_{i';i}$). We'll use these whenever the z-score is positive. Note that if the fast "double-min" z-score is positive, then the more complicated, slow z-score must also be positive.

If the fast z-score is negative, we should then evaluate the slow, N^2 z-score. These formulas are recapitulated in the options section under -z (replace j by i'). Note there was a factor of 4 missing in equation **??**, which is now fixed. Then define

$$\phi = \frac{-PQ(ii')s_{PQ}(ii')/2}{1/\sigma^2_{ii'} + s_{PQ}(ii')/4} \tag{36}$$

where

$$\sigma^2_{ii'} \equiv \frac{1}{N-2} \sum_{k \notin \{i,i'\}} \sigma^2_{ii';k} \tag{37}$$

and ϕ should never be negative, and should never be greater than $-2PQ(ii')$. Once ϕ is determined, it should be subtracted from $d_{ii'}$ in all of the renormalization calculations. Thus $b_{i;i'}$ (equation **??**) is reduced, and this is reflected in the distances from i and i' to $\bar{\imath}$. The following distances are increased, because of the decrease in $b_{i;i'}$ and $b_{i';i}$.

The distance from $\bar{\imath}$ to some other taxon j (not i or i') is given by

$$d_{\bar{\imath}j} = \frac{(d_{ij} - \bar{b}_{i;i'})/\sigma^2_\infty(i\bar{\imath}) + (d_{i'j} - \bar{b}_{i';i})/\sigma^2_\infty(i'\bar{\imath})}{1/\sigma^2_\infty(i\bar{\imath}) + 1/\sigma^2_\infty(i'\bar{\imath})} \tag{38}$$

where the $\bar{b}_{i;i'}$ is whatever we use as the distance from $\bar{\imath}$ to i, which is currently just $b_{i;i'}$, and $\sigma^2_\infty(i\bar{\imath})$ is defined below. The use of $_\infty$ is a historical artifact. We now could use the actual finite average value of the variances. The formula is

$$\sigma^2_\infty(i\bar{i}) = \frac{1}{N-2} \sum_{j \notin \{i,i'\}} \sigma^2_{\text{n.a.}}(\max(b_{i;i'}, b_{\min}) + c_i, \max(d_{ij} - b_{i;i'}, b_{\min}) + c_j) \tag{39}$$

Likewise,

$$\sigma^2_\infty(i'\bar{i}) = \frac{1}{N-2} \sum_{j \notin \{i,i'\}} \sigma^2_{\text{n.a.}}(\max(b_{i';i}, b_{\min}) + c_{i'}, \max(d_{i'j} - b_{i';i}) + c_j) \tag{40}$$

These are now normalized by $N-2$, the number of terms in the sum, but this is not necessary.

Note that I don't expect $d_{\bar{i}j} < 0$ to come up, but if it does, let's set it to zero and print a warning.

What about $c_{\bar{i}}$? First we introduce the inverse of the σ^2_t function. Note that if we generalize σ^2_t we will also need to generalize the inverse.

$$\sigma^2_{\text{inv}}(x) = \frac{b-1}{b} \ln \left(\frac{2[xb^2L + (b-1)^2]}{b\sqrt{4x(b-1)L + (b-1)^2} + (b-1)(b-2)} \right) \tag{41}$$

We should make sure that $\sigma^2_t(\sigma^2_{\text{inv}}(x)) = x$. Our old renormalization formula was

$$c_{\bar{i}} = \sigma^2_{\text{inv}} \left(\frac{1}{1/\sigma^2_t(c_i + \max(b_{i;i'}, b_{\min})) + 1/\sigma^2_t(c_{i'} + \max(b_{i';i}, b_{\min}))} \right) \tag{42}$$

but it makes far more sense to have

$$c_{\bar{i}} = \sigma^2_{\text{inv}} \left(\frac{\frac{\sigma^2_t(c_i + \max(b_{i;i'}, b_{\min}))}{(\sigma^2_\infty(i\bar{i}))^2} + \frac{\sigma^2_t(c_{i'} + \max(b_{i';i}, b_{\min}))}{(\sigma^2_\infty(i'\bar{i}))^2}}{\left(\frac{1}{\sigma^2_\infty(i\bar{i})} + \frac{1}{\sigma^2_\infty(i'\bar{i})}\right)^2} \right) \tag{43}$$

be aware of the $(\sigma^2)^2$! (Note in the paper we changed notation from σ^2_∞ to σ^2_{avg}.) This should have the same following properties as before: $c_{\bar{i}} \leq \min(\bar{b}_{i;i'} + c_i, \bar{b}_{i';i} + c'_i)$ and $c_{\bar{i}} \geq 0$. Furthermore, if, say, $c_{i'}$ is infinite, then $c_{\bar{i}} = c_i + \max(b_{i;i'}, b_{\min})$.

Options

New Option **-Z**. Not to be confused with **-z**.

Here we revive some old formulas for $PQ(ij,kl)$. But first we compute $PQ(ij,kl)$ using the current formula, and use this to get a better value. Naturally we could iterate this, but I think just doing it once will give most of the improvement (if there is any to be had).

So, we have computed $PQ(ij,kl)$, and $\sigma^2_{PQ}(ij,kl)$, but now we want to compute from them $PQ^{\text{new}}(ij,kl)$ and $\sigma^2_{PQ\text{new}}(ij,kl)$. These will then replace the old values in computing $PQ(ij)$ and $\sigma^2_{PQ}(ij)$, which get used in the z's.

$$PQ^{\text{new}}(ij,kl) \equiv [\frac{\frac{(d_{ik}+d_{jl})}{\sigma^2(\text{ord}[i,j;k,l])} + \frac{(d_{il}+d_{jk})}{\sigma^2(\text{ord}[i,j;l,k])}}{\frac{1}{\sigma^2(\text{ord}[i,j;k,l])} + \frac{1}{\sigma^2(\text{ord}[i,j;l,k])}} - d_{ij} - d_{kl}]/2 \tag{44}$$

and

$$\sigma^2_{PQ\text{new}}(ij,kl) \equiv [\frac{1}{\frac{1}{\sigma^2(\text{ord}[i,j;k,l])} + \frac{1}{\sigma^2(\text{ord}[i,j;l,k])}} + (\sigma^2_{kl;i} + \sigma^2_{kl;j})/2]/4 \tag{45}$$

```
   i         j
    \       /
     \     /
      \   /
       \ /
        | P
        |
        |
        |
        |
        ^ Q
       / \
      /   \
     /     \
    /\      /\
   / /\    /  \
  / /  \  /\  l/\
 k
```

where

$$\sigma^2(\mathrm{ord}[i,j;k,l]) \equiv \sigma^2(\mathrm{ordered}[i,j;k,l]) \equiv \sigma^2_{ik;j} + \sigma^2_{ik;l} - \sigma^2_{ik;jl} + \sigma^2_{jl;i} + \sigma^2_{jl;k} - \sigma^2_{jl;ik} \tag{46}$$

(I don't care for this term "ordered", but let's keep it for now; note this quantity should never be negative) and

$$\sigma^2_{ik;jl} \equiv {\sigma_{\mathrm{t}}}^2(d_{ik} + c_i + c_k) - {\sigma_{\mathrm{t}}}^2(d_{iP} + c_i) - {\sigma_{\mathrm{t}}}^2(d_{PQ}) - {\sigma_{\mathrm{t}}}^2(d_{Qk} + c_k) \tag{47}$$

and this should also be non-negative. This is just like I had way back in a versions called 7-9 in the first numbering system (around August 1997), except that we've added the c's in this last formula (solving the renormalization problem), and I changed a σ^2_{na} to something closer to what we have been doing lately, and I'm changing the following definitions.

For the purposes of computing $\sigma^2_{ik;jl}$ only, let us take $d_{iP} = b_{i;j}$, $d_{jP} = b_{j;i}$, d_{PQ} is the old calculation, and

$$d_{Qk} = \frac{\frac{d_{ik} - d_{iP}}{\sigma^2_{ik;j}} + \frac{d_{jk} - d_{jP}}{\sigma^2_{jk;i}}}{1/\sigma^2_{ik;j} + 1/\sigma^2_{jk;i}} - d_{PQ} \tag{48}$$

and likewise d_{Ql}. If any of these becomes negative, just set it to zero.

-S Again a new option; and -S is an OK name with me if it's OK with you.

Once before I had an option to improve the σ's, but I now think I took the wrong tack. What bothers me about the σ's now is that it's possible for the error in the error bar to be much bigger than the error bar. So I think we should reflect the error in the error bar in the error bar itself.

So here's what I think should be done.

Start with the current formulas

$$d_{il} = \begin{cases} 0 & \text{when } d_{jk} > d_{ij} + d_{ik} \\ d_{ik} & \text{when } d_{ij} > d_{ik} + d_{jk} \\ (d_{ij} + d_{ik} - d_{jk})/2 & \text{otherwise} \end{cases} \tag{49}$$

$$d_{lj} = d_{ij} - d_{il} \tag{50}$$

Let $d_m = \min(d_{il}, d_{lj})$. Replace

$$d_m \to \min(d_m + foo, d_{ij}/2) \tag{51}$$

where foo is

$$foo = (\sqrt{\sigma^2_{jk;i} + \sigma^2_{ik;j}})/2 \tag{52}$$

and the other distance (d_{il} or d_{lj}) is replaced by $d_{ij} - d_m$. A debug check is to make sure the new $\sigma^2_{ij;k}$ that results is larger than the original. (Actually, if we have BMIN's in the current version of the above d_{il} formula, then this debug check won't work; if we do have them, we might think about whether they should really be there and whether to put them in these new formulas).

Now if -S in invoked with -Z (above) then we should probably change the -Z formulas in a similar way to what we have just done above. But I suppose I will put this off. If it turns out that -S has no benefits, and -Z has no benefits, then we can suppose (perhaps not with total confidence) that the combination -S -Z is not worth doing.

-L specifies the sequence length. Default is 500. We should probably make it harder for people not to use this option. We can talk about it.

-V gives the version.

-q turns off (?) the requirement that $q[q[i]] == i$ for pairs that have $R(i, q[i])$ evaluated. This issue should become less of a concern if we investigate the following extension options.

I believe all of the following options are now pretty much obsolete, except maybe -z.

-ec this extension would make sure that, before joining the pair $(i, q[i])$, that the tournament victory that defined $q[i]$ was really unambiguous. At the end of the tournament, we force the winner to compete against every taxon head-to-head. But if it loses to one, that one is declared the winner without having to compete further. This helps ensure N^3 execution. But, given that we have now settled on i, we can afford further investigation. I don't know if we want to store the fact that $q[i]$ did not win unambiguously the first time.

There are various proceedures we could consider. I don't even know if this situation ever comes up (although I imagine it will if the data are really ambiguous). The first one I can think of, which seems reasonable, is to do a tornament using $R(i, q[i], j)$ and whichever j beats $q[i]$ by the biggest margin is the winner (call it $q_{\text{new}}[i]$). Then likewise use $R(q[i], i, j)$ to define i_{new}. Most of the time, there are no new winners. If there is one winner, join that pair. If there are two winners, then look at whether $R(i, q_{\text{new}}[i])$ or $(i_{\text{new}}, q[i])$ has a better (lower) value, and join the corresponding pair.

-ez This extension is similar in spirit to -ec, except it assumes that $R(i, j)$ is more reliable and preferable to use than $R(i, j, k)$. In this case, before deciding to join the pair $(i, q[i])$, one finds the minimum over j of $R(i, j)$, and over j' of $R(j', q[i])$, and joins whichever is the smallest. The options -ec and -ez should not be used together.

-z This uses the fuller, N^2, original calculation of z. Instead of taking a $\min_{kl}$, this involves doing an average over l of a $\min_k$. Trying to get an N^3 version to do anything like this may be hopeless, but we should confirm this.

this slow $z(i, j)$ is defined by.

$$z(i, j) \equiv PQ(ij)/\sqrt{\sigma^2_{PQ}(ij)} \tag{53}$$

where

$$PQ(ij) \equiv \frac{1}{s_{PQ}(ij)} \sum_k \frac{1}{\sigma^2_{PQ}(ij, kl)} \min_l [PQ(ij, kl)] \tag{54}$$

where $s_{PQ}(ij)$ is the normalization for the weighted average, i.e., the sum over k of the $1/\sigma^2_{PQ}(ij, kl)$ for the l that minimizes the PQ distance, and

$$\sigma^2_{PQ}(ij, kl) \approx \left[\frac{1}{\frac{1}{\min(\sigma^2_{ik;j}, \sigma^2_{ik;l}) + \min(\sigma^2_{jl;i}, \sigma^2_{jl;k})} + \frac{1}{\min(\sigma^2_{il;j}, \sigma^2_{il;k}) + \min(\sigma^2_{jk;i}, \sigma^2_{jk;l})}} + (\sigma^2_{kl;i} + \sigma^2_{kl;j})/2\right]/4 \tag{55}$$

$$PQ(ij,kl) \approx [\frac{\frac{(d_{ik}+d_{jl})}{\min(\sigma^2_{ik;j},\sigma^2_{ik;l})+\min(\sigma^2_{jl;i},\sigma^2_{jl;k})} + \frac{(d_{il}+d_{jk})}{\min(\sigma^2_{il;j},\sigma^2_{il;k})+\min(\sigma^2_{jk;i},\sigma^2_{jk;l})}}{\frac{1}{\min(\sigma^2_{ik;j},\sigma^2_{ik;l})+\min(\sigma^2_{jl;i},\sigma^2_{jl;k})} + \frac{1}{\min(\sigma^2_{il;j},\sigma^2_{il;k})+\min(\sigma^2_{jk;i},\sigma^2_{jk;l})}} - d_{ij} - d_{kl}]/2 \tag{56}$$

$$\sigma^2_{PQ}(ij) \equiv \frac{1}{s_{PQ}(ij)} + \sigma^2_{na}(i,P,j) \tag{57}$$

Actually, the second term above needs to be expressed as something that can be renormalized. If i and j are neighbors, then all $\sigma^2_{ij;k}$ should be equal and any one of them will do. I guess the obvious choice then is

$$\sigma^2_{PQ}(ij) \equiv \frac{1}{s_{PQ}(ij)} + \frac{1}{4(N-2)} \sum_{k \notin \{i,j\}} \sigma^2_{ij;k} \tag{58}$$

where $N-2$ is just number of terms in the sum.

-n I think the factor of $\frac{1}{N-3}$ in $R(i,j,j')$ should probably be replaced by $\frac{1}{N-3}\frac{N-2}{2}$. Note that for $N=4$ there is no difference, but the difference becomes large for large N. But I want to test this before doing anything. Note that I don't think we want to change $R(i,j)$.

-r Once we know Δb_{ij} then we can better estimate the variances. But this thing is messy to implement in N^3. Actually, I'm not sure what the prospects are here.

We currently have

$$\sigma^2_{ik;j} = \sigma^2_{na}(d_{il}+c_i, d_{kl}+c_k) \tag{59}$$

but we can replace this by

$$\sigma^2_{ik;j}{}' = \sigma^2_{na}(b_i+c_i, b_k+c_k) \tag{60}$$

and likewise

$$\sigma^2_{jk;i}{}' = \sigma^2_{na}(b_j+c_j, b_k+c_k) \tag{61}$$

where b_i, b_j and b_k are *locally* defined as

$$b_i = \max[(d_{ij}+\Delta b_{ij})/2, b_{\min}] \tag{62}$$

$$b_j = \max[(d_{ij}-\Delta b_{ij})/2, b_{\min}] \tag{63}$$

and

$$b_k = \frac{\max(d_{ik}-b_i, b_{\min})/\sigma^2_{ik;j} + \max(d_{jk}-b_j, b_{\min})/\sigma^2_{jk;i}}{1/\sigma^2_{ik;j} + 1/\sigma^2_{jk;i}} \tag{64}$$

Using these new σ's we can recompute Δb_{ij} and $\Delta^2 b_{ij}$, and just as important, s_{ij}. We also must remember to use these in computing $\Delta b''$. We obviously could iterate here using the new Δb_{ij} to get new σ's, but I don't think it's necessary. The main point is that if b_i and b_k are large, and b_j is small, then we get a much better estimate on the very first iteration. The convergence may be more gradual when the b's are neary equal, but the total improvement will also be negligible. Furthermore, it's really the change in the form of s that is most important.

If we are going for N^3 complexity, we cannot afford to do these two passes for every i,j pair every interation. So instead, we do the two passes on the initial iteration, and save the Δb_{ij} from the first pass as well (I assume we were saving this anyway, because we would need it, [or something equivalent, like these context dependent b_i's?] to compute the $\Delta b''_{ij}$). Then, when updating the s_{ij}, Δb_{ij}, and $\Delta^2 b_{ij}$ after renormalization, we can use the second pass formula in just one pass. This will give slightly different results then what we get updating the whole matrices with 2 passes, because of using the old Δb_{ij}. Of course if i or j a new taxon that has just been created from two old ones, then we can do the full two passes as described, and store the first-pass Δb_{ij} in the matrix containing the original values.