**Maximum Likelihood estimate for the supertree edge lengths and gene evolution rates**

Consider a rooted supertree (ST) with a fixed topology. The ST encompasses a set of edges ***e*** defined by the ST topology and a set of unknown edge lengths ***t***. Consider a set of unrooted GTs reduced to MAST with the given ST. Each GT encompasses a set of edges with known edge lengths and an unknown gene-specific evolution rate (***b****k*, ***l****k* and *rk* for the *k*-th GT, respectively). Each edge of each GT uniquely maps to an ST path ***e****j*, that is a subset of adjacent edges in the ST (*bk,i* ≡ ***e****j* where ***e****j* ⊆ ***e*** for the *i*-th edge of the *k*-th GT).

Let be the length of the path ***e****j*. We assume that the length of the *i*-th edge of the *k*-th GT is related to the length of the corresponding ST path ***e****j*:

where ε*i,k* is the multiplicative deviation factor for the given edge. We further assume that the deviation is random, independent for branches both within and between GTs, and comes from a lognormal distribution with the mean of 1 and an arbitrary variance, translating to a model with an additive normally distributed deviation in the logarithmic scale (i.e. ln ε*i,k* ~ *N*(0,σ2)).

Given ***t*** and ***r***, the expectation for the logarithm of the length of the *i*-th edge of the *k*-th GT is:

and the likelihood of observing the length *li,k* is:

where *E*2*i,k* = (ln *li,k* ‑ ln *tj* ‑ ln *rk*)2. For all observed edge lengths in all GTs (***l***), the likelihood function is

In the logarithmic scale:

where *n* is the total number of GT edges (). Designating the residual sum of squares and substituting the estimate for σ2

for large *n*, we obtain:

Because *n* is constant for a given data set, finding the maximum of *L*(***l*** | ***t***,***r***) is equivalent to finding the minimum of *E*2.

**Optimization procedure**

Least Squares (LS) is called linear if the residuals are linear for all unknowns. Linear LS can be represented in a matrix format which has a closed form solution (given that the columns of the matrix are linearly independent). However, our formulation requires taking logs over sums of unknowns in the case where a GT edge corresponds to a path in ST (). Then, the problem becomes non-linear with respect LS and can be solved only using numerical algorithms where the solution is obtained by iteratively refining the parameter values. This approach requires supplying initial values for the parameters. The goodness of the initial value estimation is critical for the convergence time of the iterative method and the risk of being trapped in local maximum points. We employed the following strategy for determining the initial values: For each ST edge, we computed the mean value of the sum over all GT edges that uniquely correspond to the given edge. The latter also creates a ranking between the gene rates (even when edges are missing). Therefore, if we assign one gene a specific rate value (e.g. the length of some edge), we obtain initial rate values for all genes. It can be easily shown that, if there are no errors in rates (i.e. σ2 = 0), the above procedure yields the accurate (ML) values for all unknowns.

To perform the LS optimization, we used the function *fmin\_slsqp()* that is part of the *scipy.optimize* package of Python which minimizes a function using sequential least squares programming. The function admits also a set of constraints that are necessary for our calculation. In both the MC and the UPM models, both the ST edges and the GT rates were constrained to positive values. For the UPM model, the distances from a node to any leaf in a subtree under that node were set equal for all subtrees under that node. It can be shown by induction that this constraint implies an ultrametric tree. Thus, we have a constraint for every internal node; in a rooted binary tree with *m* leaves, there are *m* ‑ 1 such nodes.