Efficient estimation of evolutionary rates by covariance aware regression

Richard A. Neher

[1] *Biozentrum, University of Basel and*
[2] *SIB Swiss Institute of Bioinformatics, Basel, Switzerland*

(Dated: September 4, 2018)

Shared ancestry among individuals results in correlated traits and these dependencies need to be accounted for in probabilistic inference. In strictly asexual populations, the covariances have a particularly simple block-like structure imposed by the phylogenetic tree. Ho and Ane showed how this block-like structure can be exploited to efficiently invert covariance matrices and fit linear models on trees in linear time. In this short note, I use these methods to estimate evolutionary rates and to find the root of the tree that optimizes the time-divergence relationship. The algorithm is implemented in TreeTime and can be used to estimate evolutionary rates and their confidence intervals with computational cost scaling linearly in the number of tips.

Individuals related by a phylogenetic tree and share ancestral lineages to various degrees: All individuals share the history prior to the most recent common ancestor (MRCA) and each split in the tree partitions the populations into groups that subsequently evolve independently along different paths. Any heritable property that changes through time along branches of the tree is therefore correlated among individuals. Closely related individuals tend to have similar properties and biggest differences will be found among individuals whose MRCA lived far in the past. Such phylogenetic correlations are common confounders in inference from sequence data, for example in contact map prediction from covarying amino acids (Dunn *et al.*, 2008), genome wide association studies (Price *et al.*, 2010), or in the analysis of quantitative traits in comparative phylogenetics (Freckleton, 2012). Correction of population structure and phylogenetic correlations generically requires inversion of the covariance matrix of the trait among individuals in the population.

In general, matrix inversion is a computationally expensive operation, but (Ho *et al.*, 2014) have shown that tree-structured covariance matrices can be inverted recursively. Furthermore, linear models don't require matrix inversion but can be fit in linear time by recursively calculating weighted moments that account for covariation. The purpose of this note is to give a simplified albeit less general derivation of this algorithm and use it to estimate evolutionary rates from serially sampled sequence data.

Divergence from the root of the tree, i.e. the number of mutations per genome length L that accumulated in the sampled individuals since the MRCA of the entire sample (aka root-to-tip distance, d), is a simple example of a property with strong phylogenetic correlation. According to the molecular clock hypothesis, one expects d_i to increase linearly in time as $d_i \approx \beta(t_i - T_{MRCA})$, where β is the evolutionary rate (Zuckerkandl and Pauling, 1965). The actual values d_i of a specific tip i will fluctuate around the expectation $\beta(t_i - T_{MRCA})$ since mutations accumulate stochastically, e.g. according to a Poisson process. Divergences of closely related tips are strongly correlated because they share most of their history, while divergences of tips that only merge at the root are independent. The root-to-tip distances are sums of independent contributions of branches p_i in the tree connecting the tip to the root. The mean and variance of divergence d_i can be expressed as

$$\langle d_i \rangle = \sum_{k \in p_i} \beta \Delta t_k \quad \langle \delta d_i^2 \rangle = \sum_{k \in p_i} \sigma_k^2 \,, \tag{1}$$

where Δt_k is the length of the branch in calendar time and σ_k^2 is the variance in divergence on a branch of length Δt_k (in the simplest Poisson model $\sigma_k^2 = \beta \Delta t_k / L$). Divergences of different tips i of the tree, however, are clearly not independent observations and the covariance of d_i and d_j is given by

$$c_{ij} = \mathrm{cov}(d_i, d_j) = \sum_{k \in p_i \cap p_j} \sigma_k^2 \tag{2}$$

where the sum runs over all branches k in the intersection $p_i \cap p_j$ of the paths to the root of tips i and j. Fig. 1 shows this covariance matrix for a small example. The covariance matrix has a block like structure where each block corresponds to a branch in the tree that adds an identical contribution to all pairs of its descendants. This property naturally generates the three-point structure defined by Ho *et al.* (2014).

This covariation complicates inference from tree-structured data. A common objective is to estimate the evolutionary rate β and T_{MRCA} of measurably evolving populations such as rapidly evolving RNA viruses (Drummond *et al.*, 2003b). The simplest approach would be to regress the root-to-tip distances d_i against sampling data t_i of all available tips by minimizing the squared deviation

$$R = \frac{1}{2} \sum_i (d_i - (\alpha + \beta t_i))^2 \tag{3}$$

where $\alpha = -\beta / T_{MRCA}$ (Drummond *et al.*, 2003a; Paradis *et al.*, 2004; Rambaut *et al.*, 2016; Sagulenko *et al.*, 2018; To *et al.*, 2016; Volz and Frost, 2017). However, it is obvious that different data points (tips of the tree) are not independent observations and that simple least-squares regression will give noisy estimates of the evolutionary rate β or the T_{MRCA} without meaningful confidence intervals.

Instead, the d_i from all nodes are drawn approximately from a multivariate Gaussian distribution with covariance matrix $\mathbf{C}$ given by Eq. (2). An approximately most likely clock

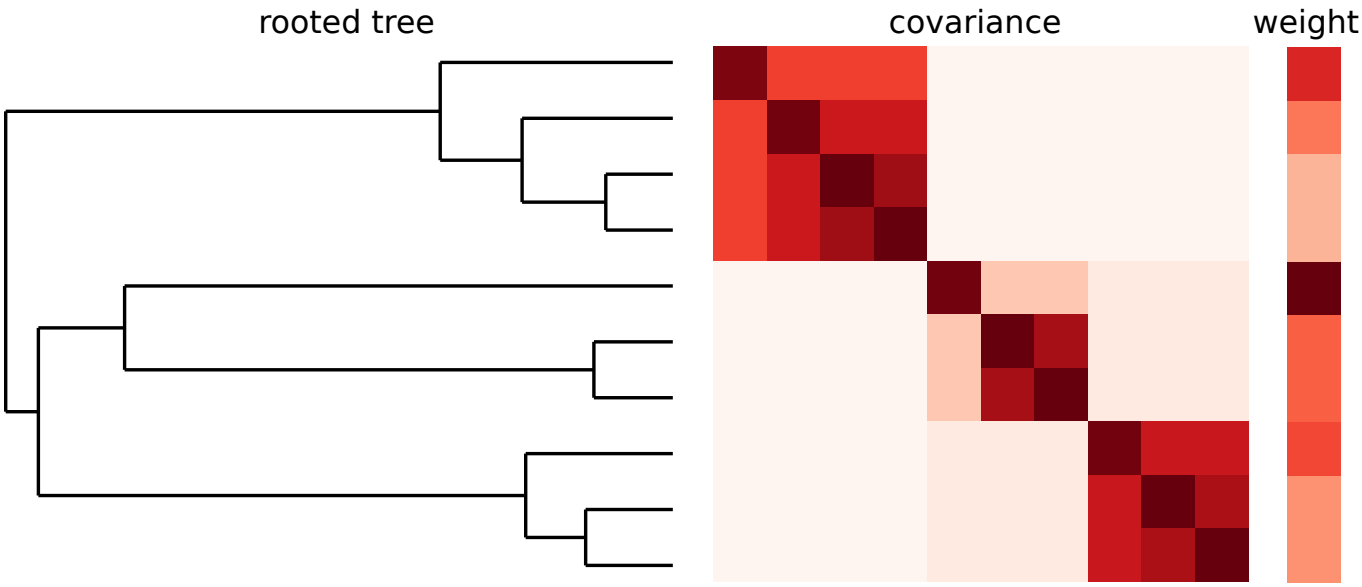

FIG. 1 **Covariance induced by a phylogenetic tree.** The closer two tips are in the tree (left), the stronger are correlations among quantities that evolve along the tree (middle, darker colors). In an inference problem, tips need to be down-weighted if they are strongly correlated with other tips. The weight of a tip i is denoted by r_i in the text and is shown in the column on the right. The tip with the longest terminal branch has the largest weight r_i.

model should therefore minimize

$$\chi^2 = \frac{1}{2}\sum_{ij} \epsilon_i h_{ij} \epsilon_j \ . \tag{4}$$

where $h_{ij}, \mathbf{H}$ is the inverse of the covariance matrix $\mathbf{C}$ and $\epsilon_i = d_i - \alpha - t_i\beta$ are the residuals. (Note that this ignores the prefactor involving determinant of $\mathbf{C}$, but this prefactor is insignificant for sufficiently large dat sets.) Solving for α, β that minimize χ^2 is straightforward but requires inversion of a potentially large covariance matrix which is a computationally expensive operation. Most implementations for matrix inversion scale as $\mathcal{O}(n^3)$ with the rank n of the matrix. Furthermore, the position of the root is typically unknown and the optimal model parameters have to be determined along with an optimization of the root. Naively, this would require repeating matrix inversion and model estimation for many choices of the root. The typical approach to circumvent this problem is to estimate to model and tree topology jointly by computationally expensive sampling of parameter and tree space (Drummond *et al.*, 2012).

As Ho *et al.* (2014) have shown, covariance matrices that have the tree-structure defined in Eq. (2) can be inverted recursively in $\mathcal{O}(n^2)$ operations. Furthermore, model parameters that minimize χ^2 can be obtained in $\mathcal{O}(n)$ operations without ever inverting the full matrix. And the recursive nature of the problem allows to evaluate the optimal model parameters for every possible choice of the root simultaneously without any additional computational burden. I will first rederive the recursive matrix inversion for a specific choice of root, then show how this calculation be efficiently done for every possible choice of root, and finally show how χ^2 can be minimized without explicit calculation of the inverse covariance matrix for a linear model.

Tree-covariance matrix inversion

Due to the structure of the covariance in Eq. (2) induced by the tree, $\mathbf{C}$ can be inverted recursively (Ho *et al.*, 2014). Consider first only the correlation $\mathbf{C}_p$ between leaves of node p induced by the child-subtrees of node p, see Fig. 2. Relative to node p, leaves that descend from the different child branches of p are uncorrelated and $\mathbf{C}_p$ has two (or more, one for each child) blocks on the diagonal. These blocks are sum of the analogous partial correlation matrices $\mathbf{C}_{c_i}$ of the children c_i and the variation $\sigma^2_{c_i}$ associated with the branch leading to child i:

$$\mathbf{C}_p = \begin{pmatrix} \mathbf{C}_{c_1} + \sigma^2_{c_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{c_2} + \sigma^2_{c_2} \end{pmatrix} \ . \tag{5}$$

Inverting this matrix is cheap for two reasons: (i) the matrix is block diagonal, hence its inverse is the block diagonal of the inverse of its blocks. (ii) the individual blocks are a sum of a matrix with known inverse (calculated in the analogous step for child nodes) and a constant. The inverse of a the sum of a matrix with known inverse and constant can be calculated using the Sherman-Morrison-Woodbury Formula (Hager, 1989):

$$(\mathbf{A} + uv^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}uv^T\mathbf{A}^{-1}}{1 + v^T\mathbf{A}^{-1}u} \tag{6}$$

In our case, the outer product uv^T is simply a product of vectors whose elements are all equal to the variance added on the branches leading to the children. Hence the inverse of an individual blocks of $\mathbf{C}_p$ corresponding to child c is

$$(\sigma^2_c + \mathbf{C}_c)^{-1} = \mathbf{H}_c - \frac{\sigma^2_c r_c {r_c}^T}{1 + \sigma^2_c s_c} \tag{7}$$

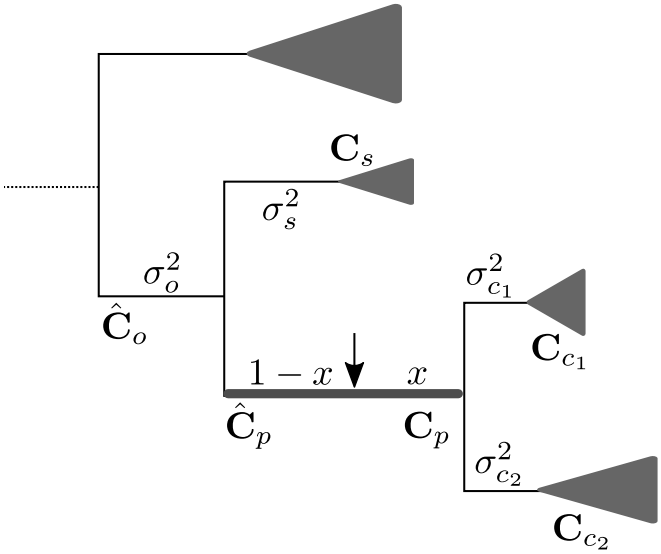

FIG. 2 **Illustration of the recursive inversion algorithm.** Each internal node p is dressed with a covariance matrix $\mathbf{C}_p$ and its inverse $\mathbf{H}_p$ (not shown) of divergence of its leaves relative to p. These matrices can be calculated recursively from the analogous quantities of the children and the branch variances $\sigma^2_{c_i}$, see Eq. (5) and Eq. (7). The covariance matrix of all leaves in the outgroup can be calculated in an analogous manner. Instead of the child nodes, this matrix includes one block for each sister clade s and one block for the outgroup of the parent node. To calculate the total leaf-covariance (or its inverse) for a choice of root – indicated by an arrow – at position x along the branch p, $\mathbf{C}_p$ and $\hat{\mathbf{C}}_p$ (or $\mathbf{H}_p$ and $\hat{\mathbf{H}}_p$) have to be combined in one matrix with additional variance $\sigma_p^2(x)$ and $\sigma_p^2(1-x)$.

where $\mathbf{H}_c$ is the inverse of $\mathbf{C}_c$, $r_c = \sum_j h^c_{ij}$ is the row or column sum of the symmetric matrix $\mathbf{H}_c$ and s_c is the sum of all elements of $\mathbf{H}_c$. This recursive update can be computed in $\mathcal{O}(n_c^2)$ operations where n_c is number of leaves of the child c. In case the child node c_i is a terminal node, the respective block diagonal is the scalar σ_c^{-2}. By traversing the tree from its leaves to root, each internal node can be dressed with a covariance matrix of its leaf nodes until we obtain the full covariance matrix between all leaves at the root. For a balanced tree, there are $\mathcal{O}(\log n)$ levels in the tree and the computational cost is dominated by the last operation at the root. Hence the overall computational complexity is $\mathcal{O}(n^2)$. For a maximally unbalanced tree, the $\mathcal{O}(n^2)$ operation has to be done $\mathcal{O}(n)$ times such that the overall complexity in this worse case scenario is $\mathcal{O}(n^3)$.

Speed and accuracy

To get a sense of the numerical accuracy and the empirical scaling of the time required for recursive matrix inversion, I generated trees and the associated covariance matrices from a Kingman coalescent process. I inverted these matrices using the matrix inversion routing in the `numpy.linalg` package and using the recursive algorithm outlined above. The recursive algorithm clearly scales more favorably with matrix size and is faster for matrices larger than 500×500. The simply python/numpy implementation is dominated by tree-traversal and memory allocation up to the matrix size of about 1000×1000 and scales as $\sim n^2$ afterwards. Optimized memory management would likely reduce runtime. The standard matrix inversion scales as $\sim n^3$ as expected.

The numerical accuracy of the recursive algorithm is lower than that of `numpy.linalg.inv`. Individual elements of $\mathbf{C} \cdot \mathbf{H}$ differ by about 10^{-13} from the identity matrix for matrices of rank $n = 1000$ while these residuals are about 5-fold lower for `numpy.linalg.inv`.

Simultaneous calculation of $\mathbf{H}$ for every choice of root

Our discussion above singled out a specific node as the root of the tree. In general, however, the root is not known and to determine a plausible root χ^2 is optimized with respect to the position of the root on the tree. Minimizing χ^2 directly would require repeated calculation of $\mathbf{H}$ for various choices of the root. However, in analogy to message-passing approaches for inference on trees (Mézard and Montanari, 2009), the inverse covariance matrix can be calculated for every node of the tree in one post-order followed by one pre-order tree traversal.

Assume we have calculated the inverse 'leaf-covariance matrices' $\mathbf{H}_p$ for each internal node p in a post-order traversal as described above. We can now calculate the 'outgroup-covariance matrices', that is the covariance $\hat{\mathbf{C}}$ of all outgroup leaves of node p relative to the base the branch p and its inverse $\hat{\mathbf{H}}$, see Fig. 2. This calculation is analogous to Eq. (7): Instead of the children of the node, the individual blocks are formed by the covariance matrices of the sister clades $\mathbf{C}_s$ and the outgroup-covariance matrix of the parent node $\mathbf{C}_o$, see Fig. 2.

With the two inverse covariance matrix $\mathbf{H}_p$ and $\hat{\mathbf{H}}_p$ of tips on either end of branch p, we can now calculate the inverse covariance matrix for an arbitrary choice of root along the branch by again using Eq. (7) with the outgroup and in-group as children, see Fig. 2.

Efficient linear regression with tree-like covariance matrices

If the only objective is to minimize χ^2 with respect to model parameters, the matrix inverse is not necessary and the optimal model parameters can be determined in linear time as shown by (Ho *et al.*, 2014). Differentiating Eq. (4) with respect to β and α, one finds:

$$\begin{aligned} 0 &= -\sum_{ij} t_i h_{ij} \epsilon_j = -t_i h_{ij} d_j + \beta t_i h_{ij} t_j + \alpha t_i r_i \\ 0 &= -\sum_{ij} h_{ij} \epsilon_j = -r_j d_j + \beta r_j t_j + \alpha s \end{aligned} \tag{8}$$

where repeated indices imply summation and r_i and s are the column sum and complete sum of h_{ij} as before. Straightfor-

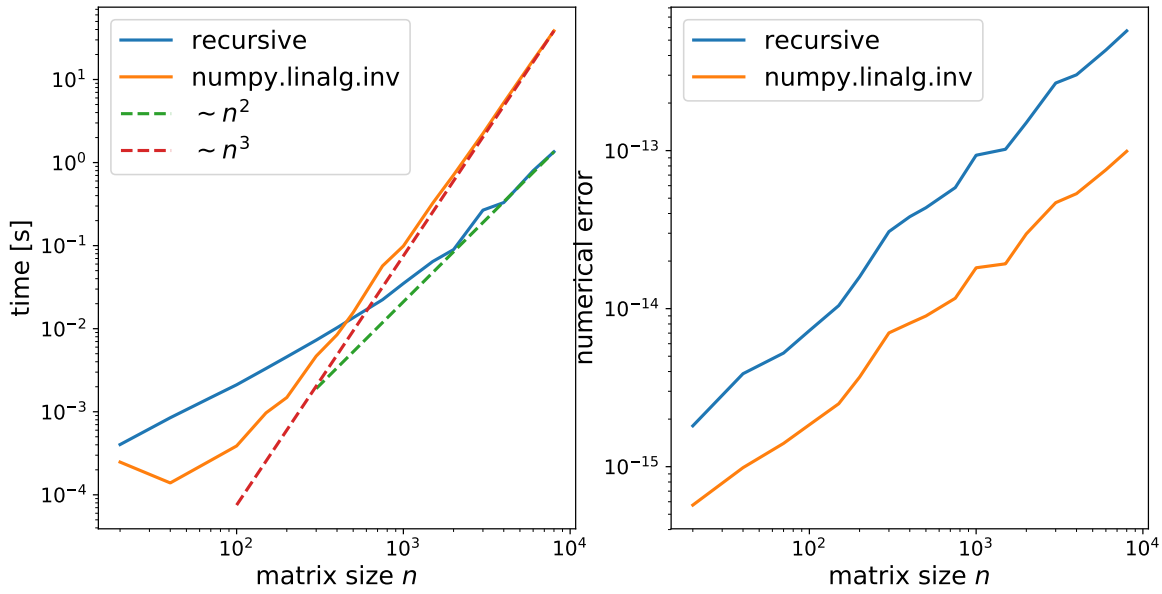

FIG. 3 **Speed and accuracy**: While standard matrix inversion scales as n^3 with rank of the matrix, the recursive algorithm scales much more favorably. In fact, up to $n = 1000$ the simply python/numpy implementation is dominated by tree-traversal and allocation and has not reached the asymptotic $\sim n^2$ scaling yet. The right panel shows the numerical accuracy of matrix inversion quantified as $n^{-2}\sum_{ij}\left|\delta_{ij} - \sum_k c_{ik}h_{kj}\right|$. The recursive inverse is less accurate than `numpy.linalg.inv` but behaves similarly when the rank of the matrix changes.

ward algebra yields

$$\alpha = \frac{r_j(d_j - \beta t_j)}{s} \tag{9}$$

$$\beta = \frac{t_i h_{ij} d_j - \frac{t_i r_i r_j d_j}{s}}{t_i h_{ij} t_j - \frac{t_i r_i r_j t_j}{s}} \tag{10}$$

for the intercept and for the slope of the regression. The form of these expression is analogous to ordinary least squares regression $\beta = \langle d_i \delta t_i\rangle / \langle \delta t_i^2\rangle$ and $\alpha = \langle d_i\rangle - \beta\langle t_i\rangle$. The quantities $\sum_i t_i r_i$ and $\sum_i d_i r_i$ are essentially weighted averages of times and divergences with weights r_i (after dividing by $s = \sum_i r_i$) . These weights r_i are lower if tip i is strongly correlated with several other tips j, see Fig. 1. The quantity $t_i h_{ij} t_j$ and $t_i h_{ij} d_j$, in turn, are analogs of second moments of time and distance. Since the matrix h_{ij} has block diagonal structure, these weighted sums can again be calculated recursively from the independent contributions of the children. The weighted sum at node p of sampling times obeys

$$\tau_p = \sum_i t_i r_i = \sum_{c\in p} \frac{\tau_c}{1+\sigma_c^2 s_c} \tag{11}$$

where τ_c is the equivalent sum of the child c. The only quantity that explicitly depends on the covariance matrix is the normalization s_c which obeys he recursive update $s_p = \sum_{c\in p} s_c/(1+\sigma_c^2 s_c)$. These relations are analogous to those by Ho *et al.* (2014).

The divergence d_{pi} of tip i, one the other hand, depends on the reference node p. As the recursion moves up the tree towards the root, we can express the d_{pi} of tip i by the contribution of the branch ℓ_c leading to child node c and the d_{ci}. The weighted sum therefore can be expressed as

$$\delta_p = \sum_i d_{pi} r_i = \sum_{c\in p} \frac{\delta_c + s_c \ell_c}{1+\sigma_c^2 s_c} \tag{12}$$

The second order quantities can be calculate analogously

$$\mathcal{T}_p = t_i h_{ij} t_j = \sum_c \left[\mathcal{T}_c - \sum_{ij\in c} \frac{\sigma_c^2 \tau_c^2}{1+\sigma_c^2 s_c}\right] \tag{13}$$

The situation is slightly more complicated for $t_i h_{ij} d_j$ since as before the d_j change as we add another branch.

$$\begin{aligned}\Phi_p &= \sum_{ij} t_i h_{ij} d_{pj} = \sum_c \sum_{ij\in c} t_i h_{ij}(d_{cj} + \ell_c) \\ &= \sum_c \left[\Phi_c + \tau_c \ell_c - \sigma_c^2 \frac{\tau_c(\delta_c + s_c \ell_c)}{1+\sigma_c^2 s_c}\right]\end{aligned} \tag{14}$$

After evaluating these equations recursively in a post-order traversal, the quantities at the root node are the full weighted averages $\tau = \sum_i r_i t_i$ etc. With these quantities in place, the optimal slope and intercept for a given choice of the root are given by

$$\begin{aligned}\beta &= \frac{\Phi - \tau\delta/s}{\mathcal{T} - \tau^2/s} \\ \alpha &= (\delta - \beta\tau)/s\end{aligned} \tag{15}$$

If the assumption that divergences are distributed as multivariate Gaussian is correct, the covariance matrix of the estimates is given by the inverse of the Hessian:

$$\begin{bmatrix} \frac{d\chi^2}{d\alpha^2} & \frac{d\chi^2}{d\alpha d\beta} \\ \frac{d\chi^2}{d\alpha d\beta} & \frac{d\chi^2}{d\beta^2} \end{bmatrix} = \begin{bmatrix} s & \tau \\ \tau & \mathcal{T} \end{bmatrix} \tag{16}$$

The covariance of the estimates can therefore be expressed as in terms of three scalar quantities that are already calculated.

$$\begin{bmatrix} \sigma_\alpha^2 & \mathrm{cov}(\beta,\alpha) \\ \mathrm{cov}(\beta,\alpha) & \sigma_\beta^2 \end{bmatrix} = \frac{1}{\mathcal{T}s - \tau^2} \begin{bmatrix} \mathcal{T} & -\tau \\ -\tau & s \end{bmatrix} \tag{17}$$

To obtain accurate estimates with tight confidence intervals, the weighted variance in tip dates $\mathcal{T}/s - \tau^2/s^2$ needs to be large, as one would intuitively expect.

Optimizing the position of the root

The optimal root will rarely be an existing node, but will typically be an intermediate point on an existing branch at distance ℓx and $\ell(1-x)$ between two nodes, see Fig. 2.

In analogy to the calculations for the covariance matrix, we can calculate the quantities $\hat{\delta}_p$, $\hat{\tau}_p$, $\hat{\mathcal{T}}_p$ and $\hat{\Phi}_p$ for all tips in the outgroup of branch p. If the variance σ_p^2 is linear in branch length, we can then calculate the weighted averages $\tau(x)$, $\delta(x)$, $\mathcal{T}(x)$ and $\Phi(x)$ for any x by treating the outgroup and the in-group as two child nodes with branch length $x\ell_p$ and $(1-x)\ell_p$. The resulting conditions are polynomial equations in x that don't seem to have a convenient solution but are easily solved numerically.

Practical issues

We have so far assumed that the variance contributions of the branches are known. However, these contributions are proportional to elapsed calendar time along the branch which is not known a priori. What we are given is branch length measured in the expected number of mutations which is proportional to time but fluctuates. Equating one with the other leads to systematic biases since less diverged parts of the tree will be associated with smaller variance such that they exert undue influence on the total estimate. This problem can be addressed with more complicated non-linear optimziation or by first estimating a time-scaled phylogeny using a rough rate estimate and then refining this estimate using branch length as measured in calendar time.

Fits of divergence time relationships with covariance aware methods can be sensitive to outliers that violate model assumptions. Since closely related tips are down-weighted due to their presumed covariation, outlier sequences have a larger weight in the covariance aware fit than in a simple least square fit. If the outliers are due to sequencing errors, culture adaptation, or similar artifacts, they should be removed.

Lastly, the accumulation of mutations is typically much more lumpy than predicted by a Poisson model. To improve stability of the rate estimation, it is advisable to include some extra variance in root-to-tip distance for each terminal branch in the model.

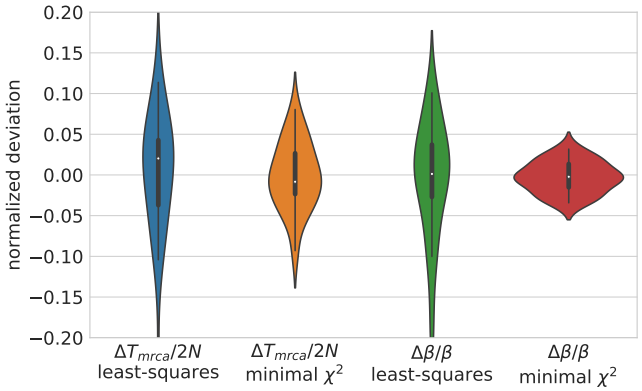

FIG. 4 Accounting for the covariance structure of reduces variation in estimates of the T_{MCRA} and the evolutionary rate β. The figure compares results from simple least-squares regression with minimization of χ^2 as defined in Eq. (4) on simulated data. Estimates that account for correlations in divergence among tips are more accurate.

Accounting for covariance reduces noise

To empirically investigate the improvement achieved when accounting for the covariance structure when estimating the rate of evolution and the time T_{MRCA} of the most recent common ancestor, I simulated evolution in population of size $N = 100$ under a neutral Wright-Fisher model using FFPopSim (Zanini and Neher, 2012). I sampled 200 individuals over a period of $T = 2N$ generations, and recorded the true tree of the entire sample including the T_{MRCA}. I reconstructed phylogenetic trees using IQ-TREE (Nguyen *et al.*, 2015) and estimated the substitution rate and T_{MRCA} using either root-to-tip vs time regression with naive least-squares or χ^2 minimization. Accounting for the covariance structure dramatically reduced the noise in the estimates, see Fig. 4.

Implementation

The algorithm to invert covariance matrices and perform regression on trees is implemented as a class `TreeRegression` in TreeTime. Estimation of root-to-tip regression and clock rate estimation is exposed as a commandline tool `treetime clock`. The output of this command when applied to a set of influenza virus NA sequences is shown in Fig. 5. The corresponding data set is part of the collections of treetime examples and tutorials .

Discussion

Phylogenetic correlations are a natural consequence of vertical descent and heritability and act as confounders in infer-

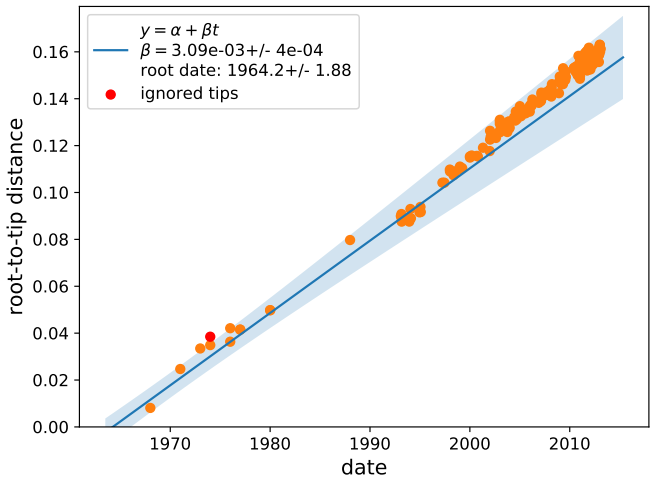

FIG. 5 Optimized divergence vs time regression for 200 influenza A/H3N2 NA sequences as produced by `treetime clock`.

ence problems. But in the simple case of additive accumulation of variance along branches, these phylogenetic correlations can be readily taken into account. The recursive block like structure of the covariance matrix allows inversion using the Sherman-Morrison-Woodbury formula that has been used in a number of related problems (Hager, 1989). Ho *et al.* (2014) have shown that analogous algorithms hold for models in which variance does not accumulate linearly in time by saturates, e.g. in an Ornstein-Uhlenbeck process.

Here, I showed how the algorithm by Ho *et al.* (2014) can be used to simultaneously evaluate all possible choices for the root of the tree and estimate evolutionary rates. I included a slightly simplified and extended re-derivations of the recursive matrix inversion and linear model fit by Ho *et al.* (2014) here in the hope that they are helpful.

Acknowledgements

I am very grateful to Vladimir Minin for pointing out the work by Ho and Ane.

References

Drummond, A., O. G. Pybus, and A. Rambaut, 2003a, Advances in Parasitology **54**, 331, ISSN 0065-308X.

Drummond, A. J., O. G. Pybus, A. Rambaut, R. Forsberg, and A. G. Rodrigo, 2003b, Trends in Ecology & Evolution **18**(9), 481, ISSN 0169-5347, URL `http://www.sciencedirect.com/science/article/pii/S0169534703002167`.

Drummond, A. J., M. A. Suchard, D. Xie, and A. Rambaut, 2012, Mol Biol Evol **29**(8), 1969.

Dunn, S. D., L. M. Wahl, and G. B. Gloor, 2008, Bioinformatics **24**(3), 333, ISSN 1367-4803, URL `https://academic.oup.com/bioinformatics/article/24/3/333/253952`.

Freckleton, R. P., 2012, Methods in Ecology and Evolution **3**(5), 940, ISSN 2041-210X, URL `https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.2041-210X.2012.00220.x`.

Hager, W., 1989, SIAM Review **31**(2), 221, ISSN 0036-1445, URL `https://epubs.siam.org/doi/10.1137/1031049`.

Ho, T., L. Si, and C. An, 2014, Systematic Biology **63**(3), 397, ISSN 1063-5157, URL `https://academic.oup.com/sysbio/article/63/3/397/1649891`.

Mézard, M., and A. Montanari, 2009, *Information, Physics, and Computation* (OUP Oxford), ISBN 978-0-19-857083-7, google-Books-ID: jhCM7i0a6UUC.

Nguyen, L.-T., H. A. Schmidt, A. von Haeseler, and B. Q. Minh, 2015, Molecular Biology and Evolution **32**(1), 268, ISSN 0737-4038, URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4271533/`.

Paradis, E., J. Claude, and K. Strimmer, 2004, Bioinformatics **20**(2), 289, ISSN 1367-4803, URL `https://academic.oup.com/bioinformatics/article/20/2/289/204981`.

Price, A. L., N. A. Zaitlen, D. Reich, and N. Patterson, 2010, Nature reviews. Genetics **11**(7), 459, ISSN 1471-0056, URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2975875/`.

Rambaut, A., T. T. Lam, L. M. Carvalho, and O. G. Pybus, 2016, Virus Evolution **2**(1), vew007.

Sagulenko, P., V. Puller, and R. A. Neher, 2018, Virus Evolution **4**(1), URL `https://academic.oup.com/ve/article/4/1/vex042/4794731`.

To, T.-H., M. Jung, S. Lycett, and O. Gascuel, 2016, Syst. Biol. **65**(1), 82.

Volz, E. M., and S. D. W. Frost, 2017, Virus Evolution **3**(2), URL `https://academic.oup.com/ve/article/3/2/vex025/4100592`.

Zanini, F., and R. A. Neher, 2012, Bioinformatics **28**(24), 3332, ISSN 1367-4811 1367-4803.

Zuckerkandl, E., and L. Pauling, 1965.

Appendix A: The value of the objective function

The objective function at optimal α and β is given by

$$\begin{aligned} 2\chi^2 &= \sum_{ij}(d_i - (\beta t_i + \alpha))h_{ij}(d_j - (\beta t_j + \alpha)) \\ &= \mathcal{D} - 2(\beta\Phi + \alpha\delta) + \beta^2\mathcal{T} + 2\alpha\beta\tau + \alpha^2 s \\ &= \mathcal{D} - \delta^2/s - \frac{(\Phi - \delta\tau/s)^2}{\mathcal{T} - \tau^2/s} \end{aligned} \tag{A1}$$

where $\mathcal{D}$ is defined in analogy to $\mathcal{T}$ and Φ

$$\begin{aligned} \mathcal{D} = d_i h_{ij} d_j &= \sum_c \sum_{ij\in c}(d_{ci} + \ell_c)(h^c_{ij} - \sigma_c^2 \frac{r_{ci} r_{cj}}{1 + \sigma_c^2 s_c})(d_{cj} + \ell_c) \\ &= \sum_c \left[\mathcal{D}_c + 2\ell_c\delta_c + \ell_c^2 s_c - (\delta_c^2 + 2\delta_c\ell_c s_c + \ell_c^2 s_c^2)\frac{\sigma_c^2}{1 + \sigma_c^2 s_c}\right] \end{aligned} \tag{A2}$$

Here $\mathcal{D}_c$ is the analogous quantity of the child and the full $\mathcal{D}$ can be calculated recursively as before.