Supplemental Information:
Anticipating the Chemistry of Live (ACOL)

Marco Visani[1], Madleina Caduff[1,2], Christoph Leuenberger[1], Daniel Wegmann[1,2], and Pierre-Marie Allard[1,3]

[1]Department of Biology, University of Fribourg, 1700 Fribourg, Switzerland
[2]Swiss Institute of Bioinformatics, 1700 Fribourg, Switzerland
[3]Corresponding author, pierre-marie.allard@unifr.ch

Contents

1 Computational tricks **1**
1.1 Discretization of branch lengths 1
1.2 Discretization of rate parameters 2

1 Computational tricks

1.1 Discretization of branch lengths

The parameters of the rate matrix μ_{c1} and μ_{c2} and the branch lengths $b(n)$ are non identifiable: doubling all rate parameters and halving all branch lengths will lead to the exact same solution. We therefore introduce the constraint

$$\sum_n b(n) = 1,$$

such that $0 \leq b(n) \leq 1$ for all branch lengths $b(n)$.
We further note that calculating the matrix exponential $\boldsymbol{P}(n) = \exp(\boldsymbol{\Lambda}_c b(n))$ for every possible branch length is computationally prohibitive. To reduce the number of calculations, we bin the branch lengths to predefined values. Specifically, let there be a regularly spaced grid on the interval $[a, b]$ consisting of K bins. The width of each bin is given by $\Delta = \frac{b-a}{K}$. Let us further define by $k(n) \in 0, \ldots, K-1$ the bin where node n is assigned to.
The transition matrix of the first three bins is given by:

$$\begin{aligned} \boldsymbol{P}(0) &= \exp(\boldsymbol{\Lambda}_c a) \\ \boldsymbol{P}(1) &= \exp(\boldsymbol{\Lambda}_c (a + \Delta)) \\ \boldsymbol{P}(2) &= \exp(\boldsymbol{\Lambda}_c (a + 2\Delta)). \end{aligned}$$

More generally, the transition matrix for bin k is given by:

$$\boldsymbol{P}(k) = \exp(\boldsymbol{\Lambda}_c (a + k\Delta)).$$

For all $k = 1, \ldots, K-1$, this term can be calculated efficiently using a recursion:

$$\begin{aligned}\boldsymbol{P}(k) &= \exp(\boldsymbol{\Lambda}_c(a+k\Delta))\\ &= \exp(\boldsymbol{\Lambda}_c(a+(k-1)\Delta)+\boldsymbol{\Lambda}_c\Delta)\\ &= \exp(\boldsymbol{\Lambda}_c(a+(k-1)\Delta))\exp(\boldsymbol{\Lambda}_c\Delta),\end{aligned}$$

where $\exp(\boldsymbol{\Lambda}_c(a+(k-1)\Delta))$ corresponds to the transition matrix of the previous bin $k-1$, and $\boldsymbol{\alpha} = \exp(\boldsymbol{\Lambda}_c\Delta)$ is a scaling matrix that needs to be calculated once. Therefore, the matrix exponential needs to be calculated only twice: once for calculating first transition matrix $\boldsymbol{P}(0)$ and once for calculating the scaling matrix $\boldsymbol{\alpha}$. The transition matrices of all subsequent bins are obtained by a recursive matrix multiplication, which is very cheap to calculate.

Since the sum of all branch lengths is constrained to one, most branch lengths will likely be very small. We therefore set $a = 0$ and $b = 0.1$ by default, assuming that the longest branch length of the tree will not exceed 10% of the total length. We further set $K = 100$ bins by default. However, all default values can be changed by the user.

To respect the sum-one-constraint, we update the branch lengths in pairs. Specifically, we select two nodes n_1 and n_2, pick a sign (+ or -) and propose moving to an adjacent bin: either $k(n_1)' = k(n_1) + 1$ and $k(n_2)' = k(n_2) - 1$ or $k(n_1)' = k(n_1) - 1$ and $k(n_2)' = k(n_2) + 1$. If the bin of a node corresponds to the first or the last bin, $k(n) = 0$ or $k(n) = K - 1$, there is only one possible direction for proposing.

1.2 Discretization of rate parameters

When updating the transition rate parameters μ_{c1} and μ_{c2} for a clique c, the transition matrix $\boldsymbol{P}(k)$ for all $k = 0, \dots, K-1$ bins must be re-calculated. Despite the above approach, this becomes computationally prohibitive when considering there to be millions of cliques (one per molecule). We therefore propose to discretize the values of the rate parameters as well. We will use a logarithmically spaced grid in the interval $[x, y]$ with a total of M bins. Not clear how many bins we will need. It is then possible to pre-calculate and store all combinations of K branch lengths and M^2 values of μ_{c1} and μ_{c2}, such that the update of these parameters will be very fast.