Supplementary Material to

Full Likelihood Inference from the Site Frequency Spectrum based on the Optimal Tree Resolution

Raazesh Sainudiin, Amandine Véber

1 Pseudo-code

First, the global process MakeHistory producing a particle is given in Function 1. For the update of the topology, it calls Sstep to insert the edges subtending $n-1$ down to $\lfloor n/2 \rfloor + 1$ leaves, Hstep to insert the edges of size $n/2$ when n is even, Lstep to insert the edges subtending $\lceil n/2 \rceil - 1$ down to 3 leaves, and finally the functions Twostep and Onestep (devoid of randomness) to place the 2- and 1-edges. See the Procedures 4, 5, 6, 7 and 8. All these procedures use the functions IndexSplit, finding the largest block (larger than some quantity given as an input) in the epoch for which it is called, and BetaSplit which computes the required conditioned Beta-splitting probabilities. They are described here in Functions 2 and 3.

Function 1: MakeHistory(A, β, n, S, θ)

Input: A, vector of prior rates for epoch times; β, parameter for Aldous' Beta-splitting model; n, sample size; S, observed SFS; and θ, scaled mutation rate
Output: (F, M, T), an SFS-history of S; and its proposal weights (w, w_1, w_2)

Initialize : $F \leftarrow 0 \in \mathbb{R}^{(n+1)\times(n+1)}$; $M \leftarrow 0 \in \mathbb{R}^{(n+1)\times(n+1)}$; $T \leftarrow 0 \in \mathbb{R}^{n+1}$; $w \leftarrow 1$; $w_1 \leftarrow 1$; $w_2 \leftarrow 1$
foreach $k \in \{2, \dots, n\}$ do T[k] $\leftarrow$ a sample from exponential($A[k]$) random variable;
foreach $k \in \{1, 2, \dots, n-1\}$ do /* get control sequence C from S */
  $C[k] \leftarrow 0$; if $S[k] > 0$ then $C[k] \leftarrow 1$;
if $n == 2$ then
  Onestep (β, n, C, F, w); Mutate $(A, n, 1, S[1], \theta, F, M, T, w_1, w_2)$
else if $n == 3$ then
  Twostep (β, n, C, F, w); Mutate $(A, n, 1, S[2], \theta, F, M, T, w_1, w_2)$
  Onestep (β, n, C, F, w); Mutate $(A, n, 2, S[1], \theta, F, M, T, w_1, w_2)$
else
  foreach $j \in \{1, 2, \dots, \lceil n/2 \rceil - 1\}$ do
    Sstep (β, n, j, C, F, w); Mutate $(A, n, j, S[n-j], \theta, F, M, T, w_1, w_2)$
  if (n *is even*) then
    Hstep (β, n, C, F, w); Mutate $(A, n, j, S[n/2], \theta, F, M, T, w_1, w_2)$
  foreach $j \in \{\lfloor n/2 \rfloor + 1, \dots, n-3\}$ do
    Lstep (β, n, j, C, F, w); Mutate $(A, n, j, S[n-j], \theta, F, M, T, w_1, w_2)$
  Twostep (β, n, C, F, w); Mutate $(A, n, n-2, S[2], \theta, F, M, T, w_1, w_2)$
  Onestep (β, n, C, F, w); Mutate $(A, n, n-1, S[1], \theta, F, M, T, w_1, w_2)$
return (F, M, T, w, w_1, w_2);

Function 2: IndexSplit(n, k, V)

Input: n, sample size; k, index; V, vector;
Output: m, largest index greater than or equal to k with $V[k] > 0$, $k-1$ otherwise

$m \leftarrow k - 1$;
foreach $i \in \{n-1, n-2, \dots, k\}$ do
  if $V[i] > 0$ then $m \leftarrow i$; break ;
return m;

Finally, the first half of the procedure Mutate places the S_{n-j} mutations carried by $n-j$ individuals on the newly inserted $(n-j)$-edges, in a multinomial way (if $S_{n-j} > 0$), and updates the importance weight w_1 accordingly. The second half of the procedure samples new Gamma-values for the lengths of the epochs in which there are some $(n-j)$-edges, and updates w_2.

Function 3: BetaSplit(β, m, J)

Input: β, tree shape parameter; m, size of edge split; J, size of largest daughter edge;
Output: I, indicator of whether an m-edge was split into J-edge and $(m-J)$-edge; w, probability of this event

if $J == \lceil m/2 \rceil$ then
    $I \leftarrow 1$; $w \leftarrow 1$;
else
    $p \leftarrow \dfrac{\lambda_{m,J}}{1 - \sum_{\ell=J+1}^{n-1} \lambda_{m,\ell}}$; $U \sim$ uniform$(0,1)$;
    if $U < p$ then $I \leftarrow 1$; $w \leftarrow p$;
    else $I \leftarrow 0$; $w \leftarrow 1-p$;
return $[I, w]$;

Procedure 4: Sstep(β, n, j, C, F, w)

Data: β, tree shape parameter; n, sample size; j, j-th step; C, control sequence; F, tree topology; and w, weight of F
Result: C, F and w are updated by Sstep

$J \leftarrow n - j$; $F[2, n] \leftarrow n$;
if $\sum_{i=J+1}^{n-1} F[2,i] == 1$ then
    $F[2, J] \leftarrow 0$
else
    if $(C[J] > 0)$ or $(\,(j == \lfloor n/2 \rfloor)$ and $(n \ is \ odd\,))$ then
        $F[2, J] \leftarrow 1$
    else
        $B \leftarrow$ BetaSplit (β, n, J); $F[2, J] \leftarrow B[0]$; $w \leftarrow w \times B[1]$
if $F[2, J] > 0$ then
    $F[2, 0] \leftarrow J$; $C[J] \leftarrow 0$;
foreach $k \in \{3, j+1\}$ do
    if $\sum_{i=J+1}^{n-1} F[k,i] > 0$ then
        $F[k, J] \leftarrow 0$
    else
        $m \leftarrow$ IndexSplit$(n, J, F[k-1, 0:n])$;
        if $C[J] > 0$ then
            $F[k, J] \leftarrow 1$; $F[k, n] \leftarrow m$; $C[J] \leftarrow 0$;
            if $F[k-1, J] == 0$ then $F[k, 0] = J$;
        else
            if $m < J$ then $F[k, J] \leftarrow 0$;
            else if $m == J$ then
                $U \leftarrow$ sample from uniform$(0,1)$ random variable;
                $q \leftarrow \dfrac{m-1}{n-k+1}$;
                if $U < q$ then
                    $F[k, J] \leftarrow 0$; $F[k, n] \leftarrow J$; $w \leftarrow w \times q$
                else
                    $F[k, J] \leftarrow 1$; $w \leftarrow w \times (1-q)$
            else if $J == \lceil m/2 \rceil$ then
                $F[k, J] \leftarrow 1$
            else
                $F[k, n] \leftarrow m$; $B \leftarrow$ BetaSplit (β, m, J); $F[k, J] \leftarrow B[0]$;
                if $F[k, J] == 1$ then $F[k, 0] \leftarrow J$;
                $w \leftarrow w \times B[1]$

Procedure 5: Hstep(β, n, C, F, w)

Data: β, tree shape parameter; n, sample size; C, control sequence; F, tree topology; and w, weight of F
Result: C, F and w are updated by Hstep

$j \leftarrow \lfloor n/2 \rfloor$; $F[2,n] \leftarrow n$;
if $\sum_{i=j+1}^{n-1} F[2,i] == 0$ then $F[2,j] \leftarrow 2$; $F[2,0] \leftarrow j$; $C[j] \leftarrow 0$;
else $F[2,j] \leftarrow 0$;
if $F[2,j] == 0$ then
  foreach $k \in \{3,4,\ldots,j+1\}$ do
    if $\sum_{i=j+1}^{n-1} F[k,i] > 0$ then $F[k,j] \leftarrow 0$;
    else
      $m \leftarrow$ IndexSplit $(n, j, F[k-1, 0:n])$;
      if $C[j] > 0$ then
        $F[k,j] \leftarrow 1$; $F[k,n] \leftarrow m$; $C[j] \leftarrow 0$;
        if $F[k-1,j] == 0$ then $F[k,0] \leftarrow j$;
      else
        if $m < j$ then $F[k,j] \leftarrow 0$;
        else if $m == j$ then
          $U \leftarrow$ sample from uniform$(0,1)$ random variable;
          $q \leftarrow \dfrac{m-1}{n-k+1}$;
          if $U < q$ then $F[k,j] \leftarrow 0$; $F[k,n] \leftarrow j$; $w \leftarrow w \times q$;
          else $F[k,j] \leftarrow 1$; $w \leftarrow w \times (1-q)$;
        else if $j == \lceil m/2 \rceil$ then
          $F[k,j] \leftarrow 1$; $F[k,0] \leftarrow j$;
        else
          $F[k,n] \leftarrow m$; $B \leftarrow$ BetaSplit (β, m, j); $F[k,j] \leftarrow B[0]$;
          if $F[k,j] == 1$ then $F[k,0] \leftarrow j$;
          $w \leftarrow w \times B[1]$;
else
  $F[3,j] \leftarrow 1$; $F[3,n] \leftarrow j$;
  foreach $k \in \{4,5,\ldots,j+1\}$ do
    $m \leftarrow$ IndexSplit $(n, j, F[k-1, 0:n])$;
    if $m < j$ then $F[k,j] \leftarrow 0$;
    else
      $F[k,n] \leftarrow m$; $U \leftarrow$ sample from uniform$(0,1)$ random variable;
      $q \leftarrow \dfrac{m-1}{n-k+1}$;
      if $U < q$ then $F[k,j] \leftarrow 0$; $w \leftarrow w \times q$;
      else $F[k,j] \leftarrow 1$; $w \leftarrow w \times (1-q)$;

Procedure 6: Lstep(β, n, j, C, F, w)

Data: β, tree shape parameter; n, sample size; j, j-th step; C, control sequence; F, tree topology; and w, weight of F

Result: C, F and w are updated by Lstep

$J \leftarrow n - j$; $F[2, n] \leftarrow n$; $F[2, J] \leftarrow F[2, j]$;
if $F[2, J] > 0$ then $C[J] \leftarrow 0$;
foreach $k \in \{3, 4, \ldots, j + 1\}$ do
  $m \leftarrow 0$;
  foreach $i \in \{J + 1, J + 2, \ldots, n - 1\}$ do `/* find size of the edge just split, if present */`
    if $F[k - 1, i] > F[k, i]$ then $m \leftarrow i$; break ;
  if $m == 0$ then
    if $\left(n - \sum_{i=J}^{n-1}(i \times F[k - 1, i]) - k + 1 + \sum_{i=J}^{n-1} F[k - 1, i]\right) == 0$ then
      $F[k, J] \leftarrow F[k - 1, J] - 1$; $F[k, n] \leftarrow J$
    else if $(C[J] == 0)$ or $\left(\sum_{i=J+1}^{n-1} F[k, i] > 0\right)$ or $(F[k - 1, J] > 1)$ then
      if $F[k - 1, J] == 0$ then $F[k, J] \leftarrow 0$;
      else
        $U \leftarrow$ sample from uniform$(0, 1)$ random variable;
        $q \leftarrow \dfrac{F[k - 1, J] \times (J - 1)}{n - k + 1 - \sum_{l=J+1}^{n-1}(F[k, l] \times (l - 1))}$;
        if $U < q$ then $F[k, J] \leftarrow F[k - 1, J] - 1$; $F[k, n] \leftarrow J$; $w \leftarrow w \times q$;
        else $F[k, J] \leftarrow F[k - 1, J]$; $w \leftarrow w \times (1 - q)$;
    else $F[k, J] \leftarrow F[k - 1, J]$;
  else if $m > 2 \times J$ then
    $F[k, n] \leftarrow m$; $F[k, J] \leftarrow F[k - 1, J] + (F[k, m - J] - F[k - 1, m - J])$;
  else
    $\delta \leftarrow \left(\sum_{i=J+1}^{m-1} F[k, i]\right) - \left(\sum_{i=J+1}^{m-1} F[k - 1, i]\right)$;
    if $(m == 2 \times J)$ and $(\delta == 0)$ then
      $F[k, n] \leftarrow m$; $F[k, J] \leftarrow F[k - 1, J] + 2$; $F[k, 0] \leftarrow J$;
    else if $(m <= 2 \times J)$ and $(\delta > 0)$ then
      $F[k, n] \leftarrow m$; $F[k, J] \leftarrow F[k - 1, J]$;
    else if $(J == \lceil m/2 \rceil)$ and $(\delta == 0)$ then
      $F[k, n] \leftarrow m$; $F[k, J] \leftarrow F[k - 1, J] + 1$; $F[k, 0] \leftarrow J$;
    else if $(m < 2 \times J)$ and $(\delta == 0)$ then
      $F[k, n] \leftarrow m$;
      if $(C[J] > 0)$ and $(F[k - 1, J] == 0)$ then $F[k, J] \leftarrow 1$; $F[k, 0] \leftarrow J$;
      else
        $B \leftarrow$ BetaSplit (β, m, J); $F[k, J] \leftarrow F[k - 1, J] + B[0]$;
        if $B[0] == 1$ then $F[k, 0] = J$;
        $w \leftarrow w \times B[1]$;

Procedure 7: Twostep(β, n, C, F, w)

Data: β, tree shape parameter; n, sample size; C, control sequence; F, topology; and w, weight of F
Result: F and C are updated by Twostep

$j \leftarrow n-2$; $F[2,n] \leftarrow n$; $F[2,2] \leftarrow F[2,j]$;
foreach $k \in \{3,4,\ldots,j+1\}$ do
  $m \leftarrow 0$;
  foreach $i \in \{3,4,\ldots,n-1\}$ do `/* find size of the edge just split, if present */`
    if $F[k-1,i] > F[k,i]$ then $m \leftarrow i$; break;
  if $m == 0$ then
    $F[k,2] \leftarrow F[k-1,2]-1$; $F[k,n] \leftarrow 2$;
  else if $m > 4$ then
    $F[k,n] \leftarrow m$; $F[k,2] \leftarrow F[k-1,2]+F[k,m-2]-F[k-1,m-2]$;
  else if $(m == 4)$ and $(F[k,3]-F[k-1,3] == 0)$ then
    $F[k,n] \leftarrow m$; $F[k,2] \leftarrow F[k-1,2]+2$; $F[k,0] \leftarrow 2$;
  else if $(m == 4)$ and $(F[k,3]-F[k-1,3] > 0)$ then
    $F[k,n] \leftarrow m$; $F[k,2] \leftarrow F[k-1,2]$;
  else if $m == 3$ then
    $F[k,n] \leftarrow m$; $F[k,2] \leftarrow F[k-1,2]+1$; $F[k,0] \leftarrow 2$;
$C[j] \leftarrow 0$;

Procedure 8: Onestep(β, n, C, F, w)

Data: β, tree shape parameter; n, sample size; C, control sequence; F, topology; and w, weight of F
Result: F and C are updated by Onestep

$j \leftarrow n-1$; $F[2,n] \leftarrow n$; $F[2,1] \leftarrow F[2,j]$;
foreach $k \in \{3,4,\ldots,j\}$ do
  $m \leftarrow 0$;
  foreach $i \in \{3,4,\ldots,n-1\}$ do
    if $F[k-1,i] > F[k,i]$ then $m \leftarrow i$; break;
  if $m > 2$ then
    $F[k,1] \leftarrow F[k-1,1]+(F[k,m-1]-F[k-1,m-1])$; $F[k,n] \leftarrow m$;
  else
    $F[k,1] \leftarrow F[k-1,1]+2$; $F[k,0] \leftarrow 1$; $F[k,n] \leftarrow 2$;
  $F[n,1] \leftarrow n$;
$F[n,n] \leftarrow 2$; $F[n,0] \leftarrow 1$; $C[j] \leftarrow 0$;

Procedure 9: Mutate$(A, n, j, s, \theta, F, M, T, w_1, w_2)$

Data: A, rates of *a priori* exponential epoch times; n, sample size; j, j-th step; s, mutations carried by $n-j$ individuals; θ, scaled mutation rate; F, topology; M, mutation matrix; T, epoch times; w_1, weight of M and w_2, weight of T

Result: M, T, w_1 and w_2 are updated by Mutate

$J \leftarrow n - j$;

foreach $i \in \{0, 1, \ldots, n\}$ do $M[i, J] \leftarrow 0$;

if $s \neq 0$ then

$$M[2 : j+1, J] \sim \text{multinomial}\left(s, \left(\frac{F[2,J] \times T_2}{\sum_{\ell=2}^{j+1} (F[\ell, J] \times T_\ell)}, \frac{F[3,J] \times T_3}{\sum_{\ell=2}^{j+1} (F[\ell, J] \times T_\ell)}, \ldots, \frac{F[j+1,J] \times T_{j+1}}{\sum_{\ell=2}^{j+1} (F[\ell, J] \times T_\ell)}\right)\right);$$

$$w_1 \leftarrow w_1 \times s! \prod_{i=2}^{j+1} \frac{1}{M[i,J]!} \left(\frac{F[i,J] \times T_i}{\sum_{\ell=2}^{j+1} (F[\ell, J] \times T_\ell)}\right)^{M[i,J]};$$

foreach $k \in \{2, 3, \ldots, j+1\}$ do

if $F[k, J] > 0$ then

$$a \leftarrow 1 + \sum_{i=J}^{n-1} M[k, i]; \quad b \leftarrow A[k] + \theta \sum_{i=J}^{n-1} F[k, i];$$

$T[k] \sim \text{gamma}(a, b)$;

$$w2 \leftarrow w2 \times \frac{b^a}{\Gamma(a)} T[k]^{a-1} \exp(-bT[k])$$

2 Exposition of the Algorithm when $n = 8$

Let us detail how MakeHistory (the full procedure constructing a tree topology, a mutation matrix and an epoch time vector compatible with a given SFS) works.

Suppose $n = 8$ and the observed SFS is $S = (5, 2, 0, 0, 1, 0, 2)$. Let us see how our sampler constructs a tree with mutations based on this information. We assume that $\beta = 0$ to simplify the expression of the probabilities related to the topology of the tree. The control sequence created at the beginning of the procedure tells us which edge sizes need to be seen in the tree. Here, it is thus equal to $C = (1, 1, 0, 0, 1, 0, 1)$.

Recall that during step j, the edges subtending $n - j$ leaves are placed in the tree.

2.1 Topology Matrix F.

We start from an $(n+1) \times (n+1)$ matrix whose entries are all equal to 0 (indexed from 0 to n), and a proposal weight $w = 1$.

j = 1: Since $C(7) = 1$, Sstep forces the presence of a 7-edge in the only epoch at which such an edge is possible, that is epoch 2. Hence, $F(2, 7) := 1$ and since a 7-edge now exists in the tree, $C(7) := 0$. The largest edge created during the first split has size 7 and the edge split during this step subtended 8 leaves by construction, and so $F(2, 0) := 7$ and $F(2, 8) := 8$. On the other hand, we do not know yet the size of the largest edge created by the split of the 7-edge, so that $F(3, 8) := 7$ but $F(3, 0)$ remains equal to 0 for now. This call of Sstep ends here.

j = 2: Because of the presence of a 7-edge at epoch 2 (i.e., $F(2, 7) > 0$), there cannot be an edge of size 6 at this epoch and $F(2, 6) = 0$. Next, $C(6) = 0$ and so the algorithm may or may not split the 7-edge into a 6- and a 1-edge. Let us say that it creates no 6-edges, which happens with probability 1/3 when $\beta = 0$. Hence, $F(3, 6) = 0$ and $F(3, 0)$ (the size

of the largest edge created by the split of the 7-edge) remains equal to 0 too. Also, the weight w associated to the tree is multiplied by the above probability, that is $w := 1/3$ after this step. This call of Sstep ends here.

$\mathbf{j = 3}$: Again, there can be no 5-edge at epoch 2. Next, $C(5) = 1$ and so the 7-edge needs to be split into a 5-edge and a 2-edge. Since at this step Sstep updates only the entries corresponding to the 5-edges, we obtain $F(3,5) := 1$, $F(3,0) := 5$ and $C(5) := 0$. In epoch 4, IndexSplit (see Section A) gives the size of the largest edge present in epoch 3, that is 5. Since a 5-edge has already been placed in the previous epoch, the presence or absence of this 5-edge in epoch 4 is random. With probability 4/5, we decide that it is absent, and so $F(4,5) = 0$ and $w := 1/3 \times 4/5 = 4/15$. This means that the 5-edge at epoch 3 was split and so $F(4,8) := 5$. This call of Sstep stops here.

$\mathbf{j = n/2 = 4}$: Because $\sum_{l=5}^{7} F(2,l) > 0$ and $\sum_{l=5}^{7} F(3,l) > 0$, there cannot be a 4-edge in epochs 2 and 3. Next IndexSplit returns 5, the size of the largest edge present in epoch 3. Since $\sum_{l=5}^{7} F(4,l) = 0$ and $C(4) = 0$, the presence of a 4-edge in epoch 4 is random. Let us say that such an edge is created by the split of the 5-edge, which happens with probability 1/2. Thus, $F(4,4) := 1$, $F(4,0) := 4$ and $w := 4/15 \times 1/2 = 2/15$. Using the same procedure (with IndexSplit returning 4 now), the 4-edge is not split at the beginning of the next epoch with probability 1/4, so that $F(5,4) := 1$ and $w := 1/30$. Finally, since there cannot be a 4-edge in epoch $k \geq 6$, Hstep forces the split of this edge, $F(6,4) = 0$ and $F(6,8) := 4$ (while w remains the same). This call of Hstep stops here.

$\mathbf{j = 5}$: First, $F(2,3) := F(2,5) = 0$. Next, in each epoch, Lstep looks for the size m of the edge split just before this epoch, if it has been already decided. This size is $m = 7$ for epoch 3. Since $7 > 2 \times 3$, the largest edge created by this split has already been decided and $F(3,3) := F(2,3) + F(3,7-3) - F(2,7-3) = 0$. In epoch 4, $m = 5$ and $F(4,4) - F(3,4) > 0$ (a 4-edge has been created), and so $F(4,3) := F(3,3) = 0$. Then, $m < 4$ and $F(4,3) = 0$, hence $F(5,3)$ remains equal to 0 with probability 1. The edge split at the beginning of epoch 6 has size $m = 4$, we do not know yet the size of the largest edge created by this split and $C(3) = 0$, hence the presence of a 3-edge in epoch 6 is random. Let us say that it is absent, which happens with probability 1/3: we thus have $F(6,3) = 0$ and $w := 1/90$. This call of Lstep stops here.

The last two steps (placing 2- and 1-edges) are fully deterministic and so the final weight of the tree topology obtained is $w = 1/90$.

$\mathbf{j = 6}$: First, $F(2,2) := F(2,6) = 0$. Next, in each epoch, Twostep again looks for the size m of the edge split at its beginning, if such an edge already exists (otherwise $m = 0$). Hence, in epoch 3 we have $m = 7 > 4$ and so $F(3,2) := F(2,2) + F(3,5) - F(2,5) = 1$. In epoch 4, $m = 5$ and $F(4,2) := F(3,2) + F(4,3) - F(3,3) = 1$. In epoch 5, $m = 0$ and so a 2-edge needs to be split: $F(5,2) = 0$ and $F(5,8) = 2$. In epoch 6, $m = 4$ and since no 3-edge was created by this split, we have $F(6,2) := F(5,2) + 2 = 2$ and $F(6,0) = 2$. In epoch 7, $m = 0$ and so $F(7,2) = F(6,2) - 1 = 1$ and $F(7,8) = 2$. Finally, $F(8,2)$ remains equal to 0 (there are only 1-edges) and $F(8,8) := 2$.

$\mathbf{j = 7}$: Onestep considers each split, epoch by epoch, and checks whether the number of 1-edges remains the same, or increases by 1 or 2 (the latter being the consequence of the split of a 2-edge). Hence, $F(2,1) = F(3,1) := 1$, $F(4,1) := 2$, $F(5,1) = F(6,1) := 4$, $F(7,1) := 6$ and $F(8,1) := 8$. Also, $F(5,0) = F(7,0) = F(8,0) := 1$.

The tree topology we obtain is thus (recall that $F(k,8)$ gives the size of the edge split at the beginning of epoch k and $F(k,0)$ that of the largest edge created by this split):

$$\begin{array}{c} \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 7 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 8 \\ 5 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 7 \\ 4 & 2 & 1 & 0 & 1 & 0 & 0 & 0 & 5 \\ 1 & 4 & 0 & 0 & 1 & 0 & 0 & 0 & 2 \\ 2 & 4 & 2 & 0 & 0 & 0 & 0 & 0 & 4 \\ 1 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \end{array}$$

See Figure 1 for a tree representation of this topology.

2.2 Mutation Matrix M and Epoch Time Vector T.

We present the construction of the mutation matrix M and of the epoch time vector T in a separate paragraph for the sake of clarity, but in fact the mutations carried by $n-j$ individuals in the sample are placed just after the j-th partial update of the topology matrix F.

We start from an $(n+1)\times(n+1)$ matrix M whose entries are all 0, and an $(n+1)$ vector T such that T_k is a realization of an exponential random variable with parameter A_k.

For j ranging from 1 to n, after the j-th update of the topology we first check whether there are $(n-j)$-mutations to place (i.e., $S(n-j)>0$). If it is the case, we use the distribution of the $(n-j)$-edges just obtained and the current value of the epoch time vector T to give a weight W to each epoch and distribute the mutations in a multinomial way. For example, for $j=1$, there is only one edge in epoch 2 subtending $n-1=7$ leaves, and this edge is split at the beginning of epoch 3. Consequently, the only possible allocation of the $S_7=2$ mutations carried by 7 individuals is to declare that $M(2,7)=2$ and $M(k,7)=0$ for $k>2$. The time $T(2)$ is then updated by taking an independent sample from a $\mathcal{G}(1+2, A_2+\theta)$ distribution. The importance weight w_2 is multiplied by the likelihood of the sampled value, and no other epoch times are updated.

Likewise, the only edge subtending 5 leaves is placed during step $j=3$ in epoch 3 and it is split at the beginning of epoch 4. This imposes that $M(3,5)=1$ and $M(k,5)=0$ for $k\neq 3$, and $T(3)$ is replaced by an independent sample from a $\mathcal{G}(1+1, A_3+\theta)$ distribution. The weight w_2 is updated accordingly.

As concerns the $S_2=2$ mutations carried by 2 individuals, during the $(n-2)nd$ update of the topology we inserted 1 2-edge in epoch 3, 1 in epoch 4 (the continuation of that in epoch 3), 2 in epoch 6, 1 in epoch 7 and 0 in epochs 2, 5 and 8. This gives the weights

$$W(3)=T(3),\ W(4)=T(4),\ W(6)=2T(6),\ W(7)=T(7)$$

to the epochs in which we see some 2-edges, and $W(k)=0$ otherwise. Writing $L_2=\sum_{k=2}^{n} W(k)$ for the total length of 2-edges in the partial topology constructed up to step

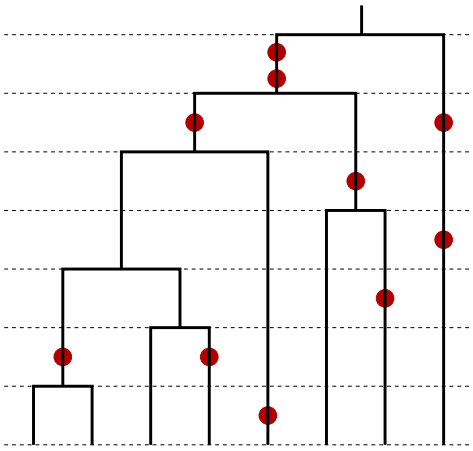

Figure 1: Tree with mutations corresponding to the result of the sampling described in the example. The F-matrix fully characterizes the tree topology, while the mutation pattern shown here is only one instance of the possible mutation placements corresponding to the matrix M. For example, the single mutation carried by a singleton lineage in epoch 8 ($M(8,1) = 1$) may actually be carried by any of the 8 extant branches in epoch 8.

$n-2$ (included), we then sample the second column of the mutation matrix according to the following multinomial distribution:

$$\big(M(2,2), M(3,2), \ldots, M(8,2)\big) \sim \text{Multinomial}\left(S_2\,;\, \frac{W(2)}{L_2}, \frac{W(3)}{L_2}, \ldots, \frac{W(8)}{L_2}\right).$$

We multiply the importance weight w_1 by the probability of the mutation allocation sampled. We then update the values of $T(k)$ by taking independent samples from $\mathcal{G}\big(1 + \sum_{l=2}^{7} M(k,l), A_k + \theta \sum_{l=2}^{7} M(k,l)\big)$ distributions, only for $k = 3, 4, 6, 7$. The weight w_2 is multiplied by the likelihood of this 4-sample of times.

We proceed in the same way to arrange the 5 mutations carried by a single individual on the final topology and update the epoch times and importance weights accordingly. In the end, a possible mutation matrix created by the procedure is the following:

$$\begin{array}{c} \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

3 Conditioned Gamma variables

In this section, we provide a short (standard) proof of the fact that if T follows a Gamma distribution with parameters (k, λ), then the law of T conditional on $\text{Poisson}(\theta T) = m$ is again a Gamma distribution with parameters $(k + m, \lambda + \theta)$. Indeed, we have

$$\begin{aligned}
\mathbb{P}\big[\text{Poisson}(\theta T) = m\big] &= \frac{\lambda^k}{\Gamma(k)} \int_0^\infty t^{k-1} e^{-\lambda t} \mathbb{P}\big[\text{Poisson}(\theta t) = m\big] \mathrm{d}t \\
&= \frac{\lambda^k}{\Gamma(k)} \int_0^\infty t^{k-1} e^{-\lambda t} \frac{e^{-\theta t}(\theta t)^m}{m!} \, \mathrm{d}t \\
&= \frac{\lambda^k \theta^m}{\Gamma(k) m!} \int_0^\infty t^{k+m-1} e^{-(\lambda+\theta)t} \mathrm{d}t \\
&= \frac{\lambda^k \theta^m \Gamma(k+m)}{\Gamma(k)(m!)(\lambda+\theta)^{k+m}},
\end{aligned}$$

and so the density of T conditional on $\text{Poisson}(\theta T) = m$ is equal to

$$\frac{1}{\mathbb{P}\big[\text{Poisson}(\theta T) = m\big]} \frac{\lambda^k}{\Gamma(k)} t^{k-1} e^{-\lambda t} \frac{e^{-\theta t}(\theta t)^m}{m!} = \frac{(\lambda+\theta)^{k+m}}{\Gamma(k+m)} t^{k+m-1} e^{-(\lambda+\theta)t}$$

on $\mathbb{R}_+^*$. This is the density of the $\mathcal{G}(k + m, \lambda + \theta)$ distribution.

4 Example using `ms`

In this example, we considered a rather complex historical scenario of a stepping-stone model with a recent barrier (as given in Fig. 3 of the documentation for `ms`, available from `https://uchicago.box.com/s/l3e5uf13tikfjm7e1il1eujitlsjdx13`). There are six subpopulations that exchange migrants in a stepping-stone model. At a time $T = 2$ time units in the past a barrier to gene flow arose, such that no further gene flow occurs between subpopulation 3 and subpopulation 4. We quote the exact command we used in our simulation with explanation quoted directly from the `ms` documentation for concreteness.

```
ms 15 100 -t 10.0 -I 6 0 7 0 0 8 0 -m 1 2 2.5 -m 2 1 2.5 -m 2 3 2.5
-m 3 2 2.5 -m 4 5 2.5 -m 5 4 2.5 -m 5 6 2.5 -m 6 5 2.5 -em 2.0 3 4
2.5 -em 2.0 4 3 2.5
```

> The phrase, -I 6 0 7 0 0 8 0, sets up 6 subpopulations with zero migration rate between them and establishes that a sample of size 7 is taken from subpopulation 2 and a sample of size 8 is taken from subpopulation 5. (In the output the first 7 haplotypes are from subpopulation 2 and the next 8 are from subpopulation 5. The -m commands set up migration, 4Nm = 2.5 , between the neighboring subpopulations (except between subpopulation 3 and 4). The -em commands modify the migration matrix at time 2.0 in the past such that pastward of this time, migration at rate 4Nm = 2.5 occurs also between subpopulation 3 and 4.