
Algorithm 1: LOGDEPTH: pseudocode for equation Eq. (4)

Input: Count vector C ; interval boundaries T ;

Parent time t_p ; child time t_c ;

Initial index i_0 ; recombination rate r ;

Coalescence rate c ; recombination flag R

Output: log likelihood contribution in time dimension

- 1 Initialize cumulative area $A \leftarrow 0$;
- 2 Initialize result $S \leftarrow 0$;
- 3 Set index $i \leftarrow i_0$;
- 4 Define helper function $f(f_0, f_1)$:
 - If $f_0 = f_1$, return 0
 - Else return $\frac{c}{r - cf_1} \cdot \frac{c}{r - cf_0} \cdot (f_1 - f_0) \cdot \frac{r}{c}$

$t_{\text{stop}} \leftarrow t_p$

if $R = \text{False}$ **then**

└ $t_{\text{stop}} \leftarrow t_c$

while $T_i > t_{\text{stop}}$ **do**

└ $A \leftarrow A + (T_i - T_{i-1}) \cdot C_{i-1}$

└ $i \leftarrow i - 1$

└ **if** $i < 1$ **then**

└└ break

if $R = \text{False}$ **then**

└ **return** $-c \cdot A$

$B \leftarrow -r / (r - cC_{i-1}) \cdot \exp(-rT_i - cA)$

while $i > 0$ **do**

└ $A \leftarrow A + (T_i - T_{i-1}) \cdot C_{i-1}$

└ $i \leftarrow i - 1$

└ **if** $T_i = t_c$ **then**

└└ break

└ $S \leftarrow S + f(C_{i-1}, C_i) \cdot \exp(-rT_i - cA)$

$S \leftarrow B + S + r(r - cC_i) \cdot \exp(-rT_i - cA)$

return $\log(S)$

603

Algorithm 2: LOGLIKELIHOOD: pseudocode for Eq. (5)

Input: Time vector I for all nodes;
Edge arrays containing parent, child, left and right coordinate of each edge: `edges_parent`,
`edges_child`, `edges_left`, `edges_right`;
Recombination rate r , coalescence rate c ;
treesequence;
Output: Total log-likelihood value

- 1 Initialize log-likelihood: $LL \leftarrow 0$;
- 2 Initialize lineage count vector $C \leftarrow 0$;
- 3 Initialize arrays to track last parent and visited nodes ;
- 4 **while** *next tree exists in treesequence* **do**
 - // Process edge removals*
 - 5 **foreach** *edge e being removed* **do**
 - 6 Identify parent p and child c from e ;
 - 7 Get node times t_p for p , and t_c for c ;
 - 8 Decrease lineage count C between t_c and t_p ;
 - 9 Record p as last parent of c ;
 - // Process edge insertions*
 - 10 **foreach** *edge e being inserted* **do**
 - 11 Identify parent p and child c from e ;
 - 12 Get node times t_p for p , and t_c for c ;
 - 13 Set `rec_event` \leftarrow False ;
 - 14 Set `left_parent_time` $\leftarrow \infty$;
 - 15 **if** *child c had a previous parent p'* **then**
 - 16 Retrieve previous parent time $t_{p'}$;
 - 17 **if** $p' \neq p$ and p not visited **then**
 - 18 Set `rec_event` \leftarrow True ;
 - 19 Set $t_{p_{\min}} \leftarrow \min(t_p, \text{left_parent_time})$ as effective parent time ;
 - // Add depth contribution*
 - 20 $LL \leftarrow LL + \text{LOGDEPTH}(C, I, t_{p_{\min}}, t_c, p, r, c, \text{rec_event})$
 - // Add span contribution*
 - 21 $LL \leftarrow LL - r \cdot (t_p - t_c) \cdot (\text{edges_right}[e] - \text{edges_left}[e])$
 - // Update lineage count*
 - 22 Increase C between t_c and t_p ;
- 23 Number of nodes with out-degree 0: n_o ;
- 24 Contribution coalescent events: $ll_c = \text{num_edges} - \text{num_nodes} + n_o$;
- 25 Number of nodes with in-degree 0: n_r ;
- 26 Contribution recombination events: $ll_r = \text{num_edges} - \text{num_nodes} + n_r$;
- 27 Update likelihood: $LL \leftarrow LL + ll_c \cdot \log(c) + ll_r \cdot \log(r)$;
- 28 **return** LL

604

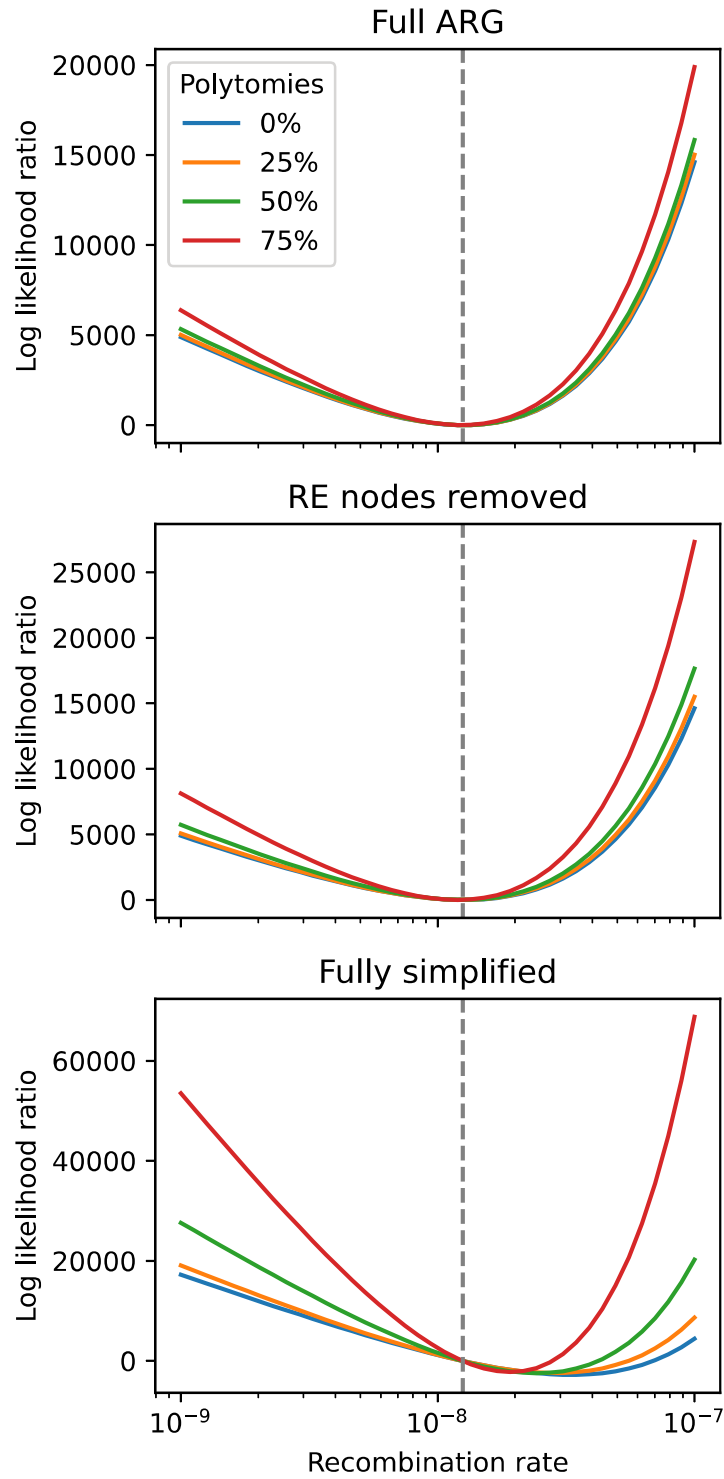


Figure S1: Log likelihood-ratio curves for three different ARGs with increasing fractions of polytomies. The base ARGs are as described in Fig. 3, but here we remove 25%, 50% and 75% of the internal nodes to create polytomies. The true value of the recombination parameter is indicated by the dashed line, as in Fig. 3.