Supplemental Information
Likelihood-based inference in isolation-by-distance models using the spatial distribution of low-frequency alleles

John Novembre and Montgomery Slatkin

Supplementary Methods

To perform likelihood-based inference for this problem we need to perform an integration over missing data:

$$P_\theta(\mathbf{X}_l) = \int\int\int P_\theta(\mathbf{X}_l|G_l, \mathbf{T_l}, \mathbf{Z}_l)P(G_l)P(\mathbf{T}_l)P(\mathbf{Z}_l)d\mathbf{Z}_l dG_l d\mathbf{T}_l.$$

To integrate over $\mathbf{T}_l$ we use straightforward Monte Carlo sampling, i.e. for the ith Monte Carlo replicate we sample the vector $\mathbf{t}_i$ from the distribution $P(\mathbf{T}_l)$. To integrate over G_l we use importance sampling, so that for the ith Monte Carlo replicate we sample g_i from an importance sampling distribution $P^*(G_l)$ and weight its contribution by $w(g_i)$. We describe both of these steps in more detail below. The resulting approximation is:

$$P_\theta(\mathbf{X}_l) \approx \frac{1}{m}\sum_{i=1}^{m}\int P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z})w(g_i)P(\mathbf{Z}_l = \mathbf{z})d\mathbf{z} \quad (1)$$

where m is the number of Monte Carlo replicates.

Further simplification can be achieved by approximating the integration over $\mathbf{z}$, so that the above becomes:

$$P_\theta(\mathbf{X}_l) \approx \frac{1}{m}\frac{1}{A}\sum_{i=1}^{m} P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i)w(g_i) \quad (2)$$

The following sections explain in more detail the steps involved in computing the approximation to $P_\theta(\mathbf{X}_l)$. First though we take a slight digression and review Felsenstein's pruning algorithm for computing $P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z})$ because its structure plays an important role in our methods for integrating out $\mathbf{Z}_l$ and G_l.

Felsenstein's 1973 pruning algorithm

We first note that due to the Brownian motion assumptions regarding dispersal,
$P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z})$ is the product of independent multivariate normals, one for each spatial dimension. The parameters of the multivariate normal can be seen to be functions of g_i, $\mathbf{t}_i$, and $\mathbf{z}$. First, let $\mu(x)$ be a vector of length j_l and with each element equal to x. Let $\Sigma(g, \mathbf{t})$ be a $j_l \times j_l$ matrix constructed so that the element at row r and column c contains the total branch length of shared ancestry between the derived allele copies r and c. On the diagonals, the elements should contain t_1, the time since the mutation arose. Finally, letting $f_{MVN}(x, \mu, B)$ represent the probability density of a multivariate normal distribution with a vector of observations x, a mean vector μ, and a variance-covariance matrix B, we have

$$
\begin{aligned}
P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z}) &= \prod_{d=1}^{2} P(X_{ld\cdot}|N\sigma_d^2, G_l = g_i, \mathbf{T}_l = \mathbf{t}_l, Z_{ld} = z_d) \\
&= \prod_{d=1}^{2} f_{MVN}(X_{ld\cdot}, \mu(z_d), N\sigma_d^2\Sigma(g, \mathbf{t})).
\end{aligned}
$$

Using the pruning algorithm of Felsenstein (1973) we can compute each multivariate normal as the product of j independent univariate normal distributions, where $j-1$ of the normal distributions correspond to computations on the interior nodes of the tree topology and the last normal distribution is computed for the root of the tree.

The general procedure for the pruning algorithm for locus l would be:

1. For each node k and each dimension d, define $\overline{x}_k^{(d)}$, S_k^2, L_k, v'_k.

2. For the nodes at the tips of the tree ($i = 1 \dots j_l$), assign $\overline{x}_i^{(d)} = X_{ldi}$, $S_i^2 = 0$, $L_i = 1$,$v'_i = v_i$ where v_i is the length of the branch immediately ancestral to node i.

3. Consider the first pair of nodes (a, b) to join looking backward in time, and label the new node they create as node c. Then set:

$$
\begin{aligned}
&v'_a = v_a + S_a^2 \\
&v'_b = v_b + S_b^2 \\
&L_c = f_N(\overline{x}_a^{(1)} - \overline{x}_b^{(1)}, 0, (v'_a + v'_b)N\sigma_1^2) f_N(\overline{x}_a^{(2)} - \overline{x}_b^{(2)}, 0, (v'_a + v'_b)N\sigma_2^2) L_a L_b \\
&S_c^2 = 1/(\tfrac{1}{v_a} + \tfrac{1}{v_b}) \\
&\overline{x}_c^{(d)} = (\tfrac{1}{v'_a}\overline{x}_a^{(d)} + \tfrac{1}{v'_b}\overline{x}_b^{(d)})/(\tfrac{1}{v'_a} + \tfrac{1}{v'_b})
\end{aligned}
$$

4. Repeat step 2 for the next pair to join, until the final node r remains. Then we have the final calculation:

$$
P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z}) = f_N(\overline{x}_r^{(1)}, z_1, v'_r) f_N(\overline{x}_r^{(2)}, z_2, v'_r) L_r.
$$

Example using figure 1

As an example we consider the intra-allelic genealogy in Figure 1. Let $\mathbf{t}$ be the vector of event times, g be the topology, and z the geographic position of the mutation event. To demonstrate the pruning algorithm we focus on a single locus and single dimension, so we momentarily suppress the d and l indices for clarity. For this tree we would have:

$$
P(X_\cdot|N\sigma^2, G = g, \mathbf{T} = \mathbf{t}, Z = z) = f_{MVN}((x_1, x_2, x_3, x_4, x_5), (z, z, z, z, z), N\sigma^2\Sigma(g, \mathbf{t}))
$$

where

$$
\Sigma(g, \mathbf{t}) = \begin{bmatrix}
t_1 & v_7 + v_9 & v_9 & v_9 & v_9 \\
v_7 + v_9 & t_1 & v_9 & v_9 & v_9 \\
v_9 & v_9 & t_1 & v_8 + v_9 & v_8 + v_9 \\
v_9 & v_9 & v_8 + v_9 & t_1 & v_6 + v_8 + v_9 \\
v_9 & v_9 & v_8 + v_9 & v_6 + v_8 + v_9 & t_1
\end{bmatrix}.
$$

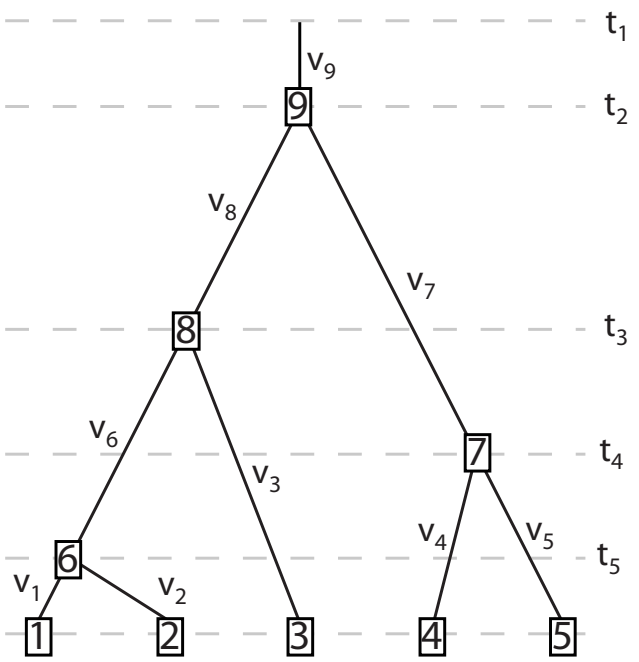


Figure 1: Example intra-allelic genealogy.

By using the pruning algorithm we arrive at a form that is the product of univariate normal distributions:

$$P(X.|N\sigma^2, G = g, \mathbf{T} = \mathbf{t}, Z = z) = \begin{array}{l} f_N(x_5 - x_4, 0, (v_4 + v_5)N\sigma^2)\times \\ f_N(x_2 - x_1, 0, (v_1 + v_2)N\sigma^2)\times \\ f_N(\bar{x}_6 - x_3, 0, (v_3 + v'_6)N\sigma^2)\times \\ f_N(\bar{x}_7 - \bar{x}_8, 0, (v'_7 + v'_8)N\sigma^2)\times \\ f_N(\bar{x}_9, z, v'_9 N\sigma^2) \end{array}$$

or equivalently:

$$P(X.|N\sigma^2, G = g, \mathbf{T} = \mathbf{t}, Z = z) = \begin{array}{l} L_6 \times L_7\times \\ L_8 \times L_9\times \\ f_N(\bar{x}_9, z, v'_9 N\sigma^2) \end{array}$$

Key features of the pruning algorithm

There are two key features of the algorithm that are useful to note. The first is that $P(X.|N\sigma^2, G = g, \mathbf{T} = \mathbf{t}, Z = z)$ is a product of terms that each represent a single node. The terms are not independent, however their dependency structure is simple and is such that L_k only depends on $\{L_i, i \in \mathcal{C}_k\}$ where $\mathcal{C}_k$ is the set of descendant nodes of k.

The second important feature is that z only enters the equation during the final step as part of the calculation of a single univariate normal distribution.

Sampling intra-allelic coalescent times $\mathbf{T}_l$

As part of the approximation to $P_\theta(\mathbf{X}_l)$ we need the ability to simulate values from $P(\mathbf{T}_l)$. Simulation from $P(\mathbf{T}_l)$ is achieved by sorting a sample of j_l values from the probability density $h(t)$ (Equation 1, main text). To simulate from $h(t)$ we transform a uniform [0,1] random deviate u by $H^{-1}(u)$, the inverse cumulative distribution function that corresponds to the density $h(t)$:

$$H^{-1}(u) = \frac{2u}{n_l(1-u)}.$$

The observation that $P(\mathbf{T}_l)$ can be computed as an ordered sample from the probability density $h(t)$ is a result that arises naturally by considering the distributions $P(T_{l1})$ and $P(T_{l1}, \ldots, T_{lj_l}|T_{l1})$ derived by Slatkin and Rannala (1997). To follow their notation we temporarily drop the subscript l and let the value f be the fraction of all lineages sampled in the population. For neutral alleles in constant-sized populations they derived the forms:

$$P(T_1 = t_1) = \frac{2jf(ft_1)^{j-1}}{(2+ft_1)^{j+1}}$$

and

$$Pr(T_2 = t_2, \ldots, T_j = t_j|T_1 = t_1) = (j-1)! \prod_{i=2}^{j} \frac{2f(2+ft_1)}{ft_1(2+ft_i)^2}.$$

If we compute the joint distribution $P(T_1 = t_1, T_2 = t_2, \ldots, T_{j_l} = t_{j_l})$ we find after simplification that:

$$P(T_1 = t_1, T_2 = t_2, \ldots, T_{j_l} = t_{j_l}) = j! \prod_{i=1}^{j} \frac{2f}{(2+ft_i)^2}$$

which immediately suggests the whole vector of times can be obtained by sorting j replicates from the single density $\frac{2f}{(2+ft)^2}$. The density $h(t)$ (Equation 1, main text) is equivalent to $\frac{2f}{(2+ft)^2}$, but with the definition that $f = \frac{n}{N}$ and with time scaled in units of N generations.

Sampling tree topologies using importance sampling

The importance sampling distribution $P^*(G_l)$ is motivated by considering the function $P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z})$. As mentioned above, the peeling algorithm of Felsenstein (1973) defines how to compute $P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z})$ as the product of independent univariate normal distributions. From the pruning algorithm, we can see that any node k will contribute to the probability of the data in proportion to L_k. This suggests an importance sampling algorithm that samples the nodes of a topology in proportion to their contribution to the likelihood.

1. For the time-point t_k, $2 \leq k \leq n$, there are $k(k-1)/2$ possible pairs of lineages that can be joined. Each of these pairs defines a new node on the tree and let the set of all possible new nodes be $\mathcal{N}$. Using step 3 of the pruning algorithm with $\theta = \theta_0$ calculate the value of $L_i, i = 1 \ldots k(k-1)/2$ for each node. Choose a node i to add to the tree with probability:

   $$\frac{L_i^H}{\sum_{a \in \mathcal{N}} L_a^H}$$

   where H is a "heat" parameter that takes values greater than or equal to one, and hence flattens the sampling distribution over potential nodes.

2. Update the tree topology to reflect the chosen node c. This includes setting $\bar{x}_c$ and S_c^2 according to the equations in step 3 of the pruning algorithm above.

3. Define the weight factor for that node as:

$$w_{r,k} = \frac{\frac{2}{k(k-1)}}{\frac{L_c^H}{\sum_{a \in \mathcal{N}} L_a^H}}$$

   where c is the index of the chosen node.

4. Repeat step 1 for the next branch-point back in time, until a complete topology is constructed.

5. The resulting topology is g_i which we store along with $w(g_i) = \prod_{k=2}^{n} w_{i,k}$.

The sampling algorithm is closely related to the importance sampling algorithm proposed by Slatkin (2002) except that the present algorithm is designed for continuous characters that evolve according to Brownian motions, rather than discrete state space characters evolving according to continuous-time Markov chains. In addition, in Slatkin (2002) the equivalent of θ_0 is fixed to be θ in order to allow for a large cancellation and simplification of the Monte Carlo estimate of the probability of the data. While the Slatkin (2002) approach leads to computationally efficient evaluation of the likelihood at a single point, it has the drawback of requiring a new set of importance sampling replicates for evaluating every point. The approach given here is more computationally expensive for evaluating the probability of the data at a single point; however, once the first set of topologies has been sampled, the algorithm allows for very efficient calculation of the probability of the data in the region around θ_0. One final difference is that the algorithm presented here includes a heat parameter H that can be used to flatten the importance sampling distribution.

Approximate analytical integration of $\mathbf{Z}_l$

The goal of this step is to compute $P_\theta(\mathbf{X}_l | G_l = g_i, \mathbf{T}_l = \mathbf{t}_i)$ by quickly performing the integral on the right-side here:

$$P_\theta(\mathbf{X}_l | G_l = g_i, \mathbf{T}_l = \mathbf{t}_i) = \int P_\theta(\mathbf{X}_l | G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z}) P(\mathbf{Z}_l = \mathbf{z}) d\mathbf{z}.$$

As noted above, the calculation of $P_\theta(\mathbf{X} | G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = \mathbf{z})$ involves $\mathbf{z}$ only through the product of two univariate normals, $\prod_{d=1}^{2} f_N(\bar{x}_r^{(d)}, z_d, v_r' N \sigma_d^2)$ where r is the index of the node representing the MRCA of the intra-allelic genealogy and $\bar{x}_r^{(d)}$ and v_r' are quantities that arise from performing the initial steps of the pruning algorithm. In addition we recall that $P(\mathbf{Z}_l = \mathbf{z}) = \frac{1}{A}$. As a result, our main challenge is to compute the right-hand side:

$$P_\theta(\mathbf{X}_l | G_l = g_i, \mathbf{T}_l = \mathbf{t}_i) = \frac{K}{A} \int \prod_{d=1}^{2} f_N(\bar{x}_{dr}, z_d, v_r' N \sigma_d^2) d\mathbf{z}$$

where K is a constant with respect to the integration representing the $j-1$ L_i terms that arise in the calculation of $P_\theta(\mathbf{X} | G_l = g_i, \mathbf{T}_l = \mathbf{t}_i, \mathbf{Z}_l = z_d)$.

Because there is no analytical solution to this integral for arbitrary habitat shapes, we make the following approximation. We assume that if the habitat is sufficiently large relative to $\bar{x}_{dr}$ and $v_r' N \sigma_d^2$ then the integration will evaluate to approximately one and thus we can make the approximation:

$$\begin{aligned} P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i) &\approx \frac{K}{A} \\ &\approx \frac{1}{A}\prod_{k=1}^{j_l-1} L_k \end{aligned}$$

Using this approach we are in effect only considering independent contrasts between the positions of lineages in the genealogy. In turn, the approach is related to the restricted maximum likelihood approach taken in Felsenstein (1981). In cases where the assumption regarding a relatively large habitat size is violated, the approximate likelihood surface will be flatter than the true likelihood surface. Thus, violations of the assumption should lead to overly broad confidence intervals and conservative inferences.

Before continuing we note one property of our approximation to $P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i)$. By collecting the terms in the univariate normal distributions that comprise the L_k terms, we have:

$$P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i) \approx \frac{1}{A}\prod_{d=1}^{2}(2\pi N\sigma_d^2)^{-(j_l-1)/2} D_i^{-1/2} \exp\{-SS_i/(2N\sigma_d^2)\}$$

where SS_i is a "sum of squares" term that is based on $\mathbf{X}_l$, g_i, and $\mathbf{t}_i$. D_i is a normalization term based on g_i and $\mathbf{t}_i$. An attractive feature here is that the terms in the above equation that depend on g_i and $\mathbf{t}_i$ are independent of θ. This suggests that we only need to store SS_i and D_i for each g_i and $\mathbf{t}_i$ sampled to calculate $P_\theta(\mathbf{X}_l|G_l = g_i, \mathbf{T}_l = \mathbf{t}_i)$ for any value of θ.

Pulling it all together to approximate $P_\theta(\mathbf{X}_l)$

The complete algorithm then proceeds as following:

1. Take as user input the value of θ_0 and H for the importance sampler $P^*(G)$.
2. For $i = 1, \ldots, m$
   (a) Sample $\mathbf{t}_i$ from $P(\mathbf{T})$ using the algorithm described above.
   (b) Sample g_i from $P^*(G)$ using the algorithm described above.
   (c) Calculate SS_i, D_i, $w(g_i)$ and store the values.
3. For any desired θ compute $P_\theta(\mathbf{X}_l)$ as:

$$P_\theta(\mathbf{X}_l) \approx \frac{1}{m}\sum_{i=1}^{m}\frac{w(g_i)}{A}\left[\prod_{d=1}^{2}(2\pi N\sigma_d^2)^{-(j_l-1)/2} D_i^{-1/2} \exp\{-SS_i/(2N\sigma_d^2)\}\right].$$

In practice, we drop the $\frac{1}{A}$ term because it only scales $P_\theta(\mathbf{X}_l)$ and by dropping it we no longer have to specify A.

Obtaining the maximum likelihood estimate and confidence intervals for θ.

Using the method described above to approximate $P_\theta(\mathbf{X}_l)$ we obtain the MLE by using numerical optimization routines provided by the GNU Scientific Library. The steps are as follows:

1. We first begin with the constraint that $N\sigma_1^2 = N\sigma_2^2 = N\sigma^2$, and we let $\theta^* = (N\sigma^2, N\sigma^2)$. In this case, $P_{\theta^*}(\mathbf{X}_l)$ is a one-dimensional function of $N\sigma^2$ and we use the Brent minimization algorithm to find the value $N\sigma^2$ that maximizes $P_{\theta^*}(\mathbf{X}_l)$. Let the maximum value of $P_{\theta^*}(\mathbf{X}_l)$ be M and the MLE be $\widehat{\theta^*}$.

2. Then starting from $\widehat{\theta^*}$ we use the Brent root solver algorithm to find one solution to the equation $M - 2 - \log(P_{\theta^*}(\mathbf{X}_l)) = 0$ that is less than $\widehat{\theta^*}$ and a second that is greater than $\widehat{\theta^*}$. These correspond to the lower and upper two log-likelihood confidence intervals for $\widehat{\theta^*}$.

3. We next relax the constraint that $N\sigma_1^2 = N\sigma_2^2$ and use the Nelder Mean simplex algorithm to find the MLE $\widehat{\theta} = (\widehat{N\sigma_1^2}, \widehat{N\sigma_2^2})$. We found it helpful to initialize the search to $\widehat{\theta^*}$ to increase the speed of the search and avoid failed convergence of the simplex algorithm.

4. Finally, we use the Brent root solver algorithm to find the two log-likelihood confidence intervals on the profile likelihood curve for $N\sigma_1^2$ and $N\sigma_2^2$.

In a series of trials, we compared the MLEs and confidence intervals obtained using the GSL routines versus those obtained by calculating a fine grid over values of θ. In all cases we found that if the GSL algorithm exited successfully, the resulting MLE and confidence intervals were accurate. For datasets on which the GSL algorithms do not converge on an MLE, manual searches using a gridded likelihood surface is a less efficient, but feasible approach to find the MLEs and CIs.

References

Felsenstein J (1973). Maximum-likelihood estimation of evolutionary trees from continuous characters. Am J Hum Genet 25: 471–492.

Felsenstein J (1981). Evolutionary trees from gene frequencies and quantitative characters: Finding maximum likelihood estimates. Evolution 35: 1229–1242.

Slatkin M (2002). A vectorized method of importance sampling with applications to models of mutation and migration. Theor Popul Biol 62: 339–348.

Slatkin M, Rannala B (1997). Estimating the age of alleles by use of intraallelic variability. American Journal of Human Genetics 60: 447–458.

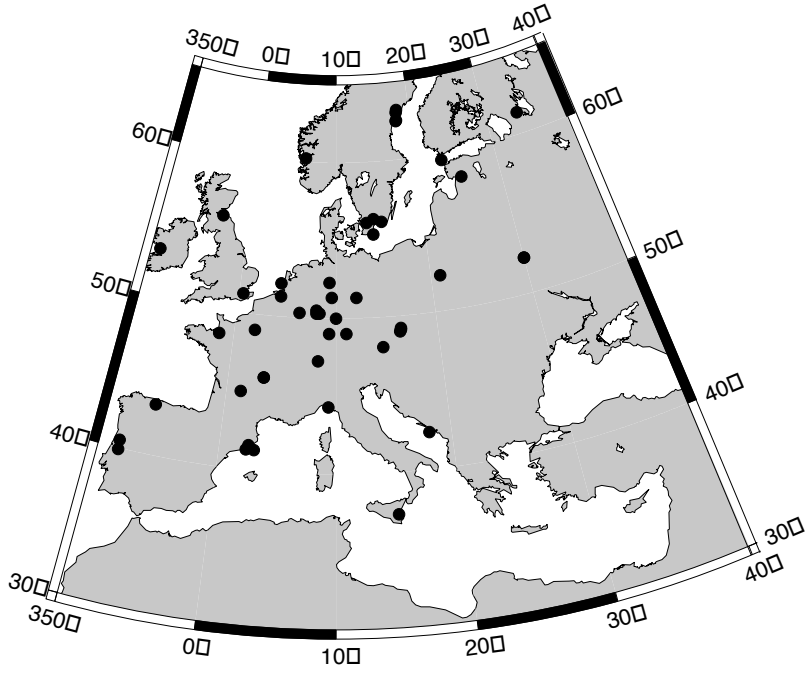
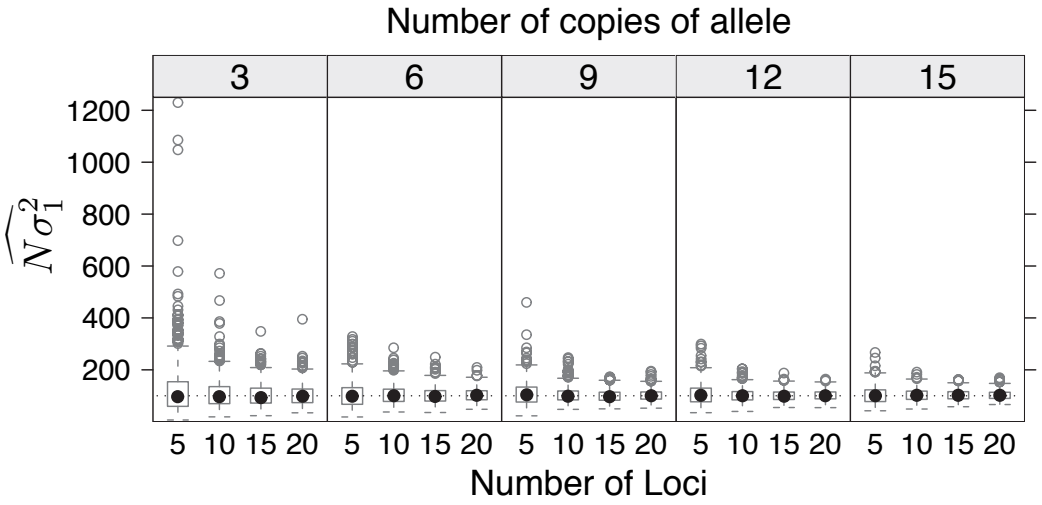


Figure 2: **Geographic distribution of samples from *Arabidopsis thaliana.***

A)



B)

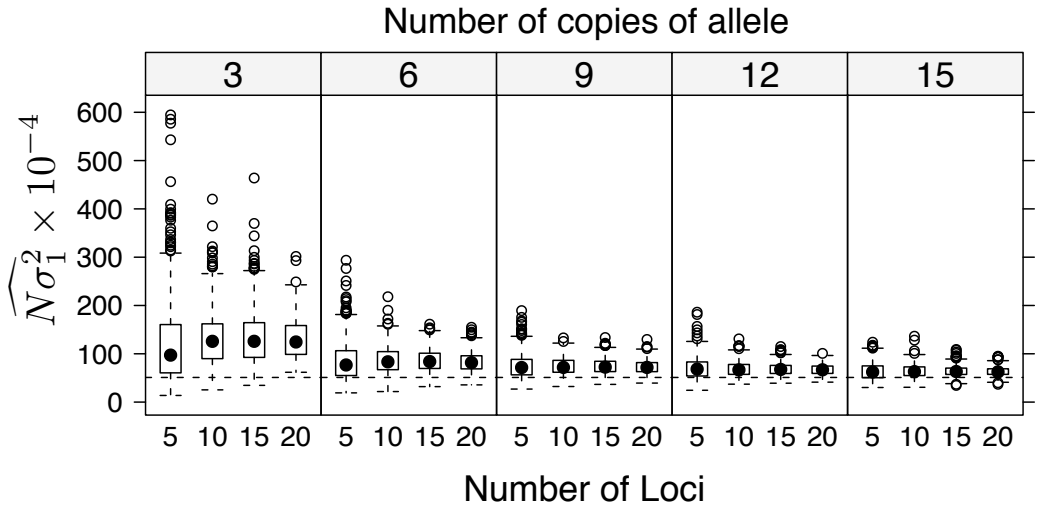


Figure 3: **Box-plot summaries of the sampling distribution of $\widehat{N\sigma_1^2}$.** Summaries are plotted across a range for the number of loci and the number of copies of the minor allele observed, and the true value of $N\sigma_1^2$ is indicated by a horizontal dashed line in each panel. (A) Brownian Birth Process results: Each summary is based on the results of applying the importance sampling algorithm with $M = 2000$, $\theta_0 = (200, 200)$ and $H = 2$ to 500 datasets obtained by independent simulations from the birth process model with $\theta = (100, 100)$. (B) Lattice-model results: Each summary is based on the results of applying the importance sampling algorithm with $M = 20000$, $\theta_0 = (102 \times 10^4, 102 \times 10^4)$ and $H = 2$ to 500 datasets obtained by independent simulations from a 101×101 lattice with $\sigma_1 = \sigma_2 = 5$, such that $theta = (51 \times 10^4, 51 \times 10^4)$.

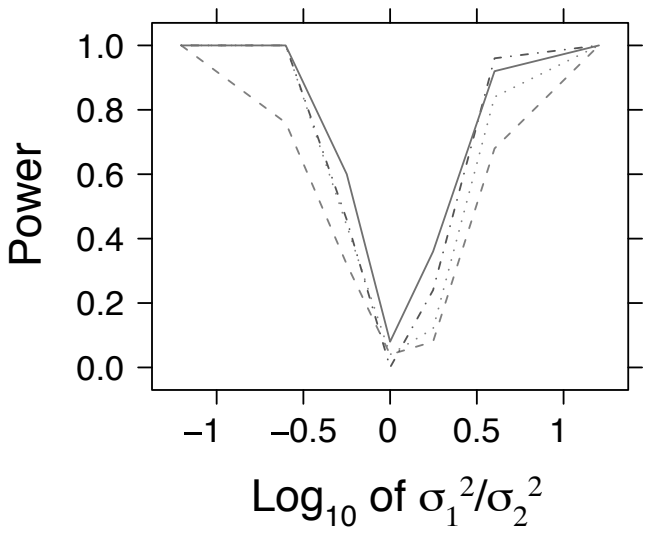


Figure 4: **Lattice model results: Power of the asymptotic likelihood-ratio test to detect departures from the null hypothesis that $\sigma_1^2 = \sigma_2^2$.** The results are based on inference performed on simulated data for a 401×401 sized lattice, where σ_1^2 was fixed at 100 and σ_2^2 was varied across the values 6.25, 25, 56.25, 100, 177, 400, 1600. All simulations were fixed at $L = 10$. Power is estimated from 25 replicate simulations.

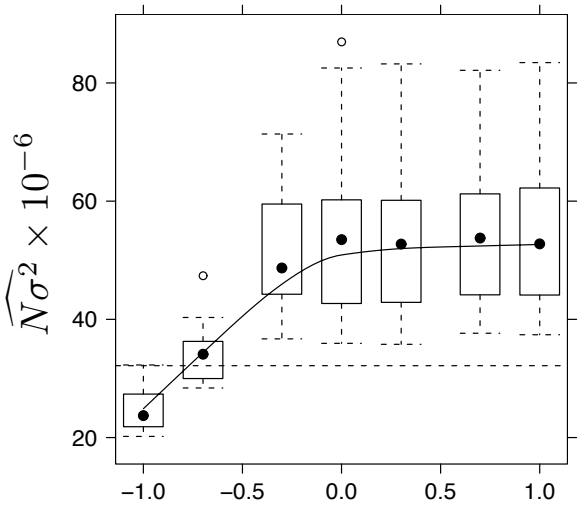


Figure 5: **Lattice model results: Effect of driving value parameter on bias.** The distribution of MLEs as a function of the ratio between the driving value θ_0 and the true value of $\theta = N\sigma^2$. The lattice size was 401×401 and $\sigma^2 = 100$ so that the true underlying value of $N\sigma^2 = 32 \times 10^6$. 25 replicates per value of θ_0. All simulations had $L = 10$, $j = 9$.

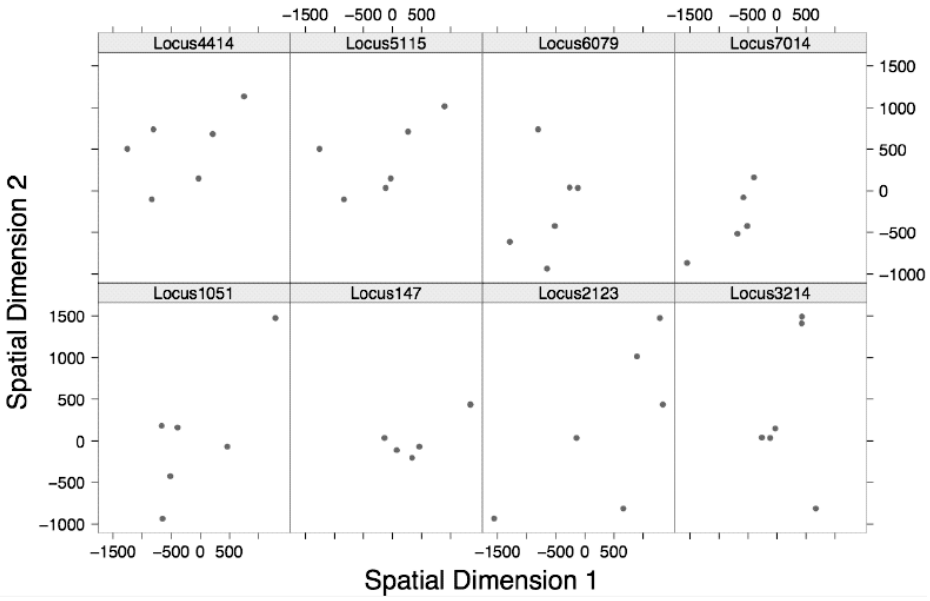


Figure 6: **The geographic positions of the minor alleles at each of the eight loci in the example dataset.** Each circle represents a location at which the minor allele was observed in the sample shown in Supplemental Figure 1. The geographic positions are given on a km scale with the origin centered arbitrarily at 50N, 10E.