File S2

Details on the algorithm to sample from the inference model

In order to provide a decision criterion for discriminating between neutral and selected markers, we calibrate the Kullback–Leibler divergence (KLD) using simulations from a predictive distribution based on the observed data set. To that end, we generate pseudo-observed data as follows.

We set the hyperparameters M_i, π_j and λ to their respective posterior means $\bar{M}_i$, $\bar{\pi}_j$ and $\bar{\lambda}$, as estimated from the MCMC. Then we draw δ_j from an exponential distribution $\sim \exp(\bar{\lambda}^{-1})$ and we draw σ_{ij} from an exponential distribution $\sim \exp\left(\delta_j^{-1}\right)$. Last, the parameter κ_{ij} is drawn from a Bernoulli distribution (with parameter the posterior mean $\bar{\kappa}_{ij}$).

We aim at sampling the allele frequency p_{ij} from the distribution with density $f(p_{ij})$ defined by equations 2 and 3 in the main text. Because the cumulative distribution function of the distribution with density $f(p_{ij})$ is not tractable, we use a rejection-sampling algorithm. To that end, we define an instrumental distribution $g(p_{ij}) \sim \mathrm{Beta}(M_i\pi_j, M_i(1-\pi_j))$, with density:

$$g(p_{ij}) = \frac{\Gamma(M_i)}{\Gamma(M_i\pi_j)\Gamma(M_i(1-\pi_j))} p_{ij}^{M_i\pi_j - 1}(1-p_{ij})^{M_i(1-\pi_j)-1} \tag{S2.1}$$

We further need to define a constant u, such that $f(p_{ij}) \leq [ug(p_{ij})]$ over the support $[0, 1]$. Noting that:

$$\frac{f(p_{ij})}{g(p_{ij})} = \frac{\exp(\sigma_{ij}\tilde{p}_{ij})}{{}_1F_1(M_i\tilde{\pi}_{ij}; M_i; \sigma_{ij})} \tag{S2.2}$$

then, if we define $u \equiv \exp(\sigma_{ij})/{}_1F_1(M_i\tilde{\pi}_{ij}; M_i; \sigma_{ij})$ we get:

$$\frac{f(p_{ij})}{ug(p_{ij})} = \exp(\sigma_{ij}(\tilde{p}_{ij} - 1)) \tag{S2.3}$$

Since $0 \leq \tilde{p}_{ij} \leq 1$ and $\sigma_{ij} \geq 0$, by definition, we have $\exp(\sigma_{ij}(\tilde{p}_{ij} - 1)) \leq 1$ and therefore $f(p_{ij}) \leq [ug(p_{ij})]$. A straightforward algorithm to sample from the distribution with density $f(p_{ij})$ is then:

(1) Sample x from a beta distribution $\mathrm{Beta}(M_i \pi_j, M_i(1-\pi_j))$ and y from $\mathcal{U}(0,1)$ (the uniform distribution over the unit interval).

(2) Check whether or not $y < f(x)/[ug(x)]$ or equivalently (see equation S2.3) if $\log(y) < \sigma_{ij}(\tilde{p}_{ij} - 1)$:

- If this holds, accept x and set $\tilde{p}_{ij} = x$;
- if not, reject the value of x and repeat the sampling step (1).

(3) Compute $p_{ij} = \tilde{p}_{ij}(1-\kappa_{ij}) + (1-\tilde{p}_{ij})\kappa_{ij}$.

Finally, we draw the allele counts $\mathbf{n}_{ij}$ in the ith deme at the jth locus by a random draw from the binomial distribution $\sim \mathcal{B}(n_{ij}, p_{ij})$. We repeat this procedure for each locus j in each deme i.

This algorithm is computationally efficient, since it avoids computing ${}_1F_1(M_i\tilde{\pi}_{ij}; M_i; \sigma_{ij})$ (see equations 2 and 3 in the main text). However, the efficiency of the algorithm may be very low for large values of σ_{ij}. This is so because the expected number of iterations required until an x is successfully generated is exactly the bounding constant $u \equiv \exp(\sigma_{ij})/{}_1F_1(M_i\tilde{\pi}_{ij}; M_i; \sigma_{ij})$. Therefore, to avoid the algorithm getting stuck in very long loops, we adopt an alternative strategy whenever $u > 10^4$: in such case, we draw x from a beta distribution $\mathrm{Beta}(\alpha, \beta)$ with the same first two moments as the target distribution (equations 2 and 3 in the main text). Little algebra shows that: $\alpha = m_1(m_2 - m_1)/(m_1^2 - m_2)$ and $\beta = \alpha(1/m_1 - 1)$, where

$$m_1 = \tilde{\pi}_{ij} \left(\frac{{}_1F_1(M_i\tilde{\pi}_{ij} + 1; M_i + 1; \sigma_{ij})}{{}_1F_1(M_i\tilde{\pi}_{ij}; M_i; \sigma_{ij})} \right) \tag{S2.4}$$

and

$$m_2 = \tilde{\pi}_{ij} \left(\frac{M_i\tilde{\pi}_{ij} + 1}{M_i + 1} \right) \left(\frac{{}_1F_1(M_i\tilde{\pi}_{ij} + 2; M_i + 2; \sigma_{ij})}{{}_1F_1(M_i\tilde{\pi}_{ij}; M_i; \sigma_{ij})} \right) \tag{S2.5}$$