File S1

S1.1 One step process, Q matrix

We denote by L the generator of the diffusion process Y_τ. We have that

$$L = \frac{1}{2}a(y)\frac{d^2}{dy^2} + b(y)\frac{d}{dy} \tag{1}$$

where $a(y)$ and $b(y)$ are the infinitesimal variance and mean of our diffusion process. For the WF model with additive selection (see main text) those functions are:

$$a(y) = y(1-y) \tag{2}$$

$$b(y) = \gamma y(1-y)(y + h(1-2y)). \tag{3}$$

By definition, the generator can also be written as

$$\lim_{\tau \downarrow 0} \frac{\mathbb{E}^y[f(Y_\tau)] - f(y)}{\tau} = Lf(y). \tag{4}$$

Ignoring the $\Delta\tau^2$ terms, we have for the infinitesimal mean:

$$\mathbb{E}^y[Y_{s+\Delta\tau} - Y_s \mid Y_s] \cong \gamma Y_s(1-Y_s)(Y_s + h \cdot (1-2Y_s))\Delta\tau = b(Y_s) \cdot \Delta\tau. \tag{5}$$

Similarly, the infinitesimal variance is:

$$\mathbb{E}^y\left[\{Y_{s+\Delta t} - Y_s - \gamma Y_s(1-Y_s)(Y_s + h \cdot (1-2Y_s))\}^2 \mid Y_s\right] \cong Y_s(1-Y_s)\Delta\tau = a(Y_s) \cdot \Delta\tau. \tag{6}$$

We want to choose the Markov chain Z such that $Z \simeq Y$, in the sense that the probability distribution governing the samples of Z is close to the probability distribution governing the samples of Y. To achieve that, we can match the infinitesimal mean and variance of Z and

Y (see Durrett (2008)). By definition of the generator of Z_τ (see equation 6), we know the probabilities of transition in time $\Delta\tau$. Assuming the process starts at $Z_s = z_i$:

$$Z_{s+\Delta\tau} = \begin{cases} z_i & \text{with probability } 1-(\beta_i+\delta_i)\Delta\tau + \mathcal{O}(\Delta\tau^2) \\ z_{i+1} & \text{with probability } \beta_i\Delta\tau + \mathcal{O}(\Delta\tau^2) \\ z_{i-1} & \text{with probability } \delta_i\Delta\tau + \mathcal{O}(\Delta\tau^2) \end{cases} \tag{7}$$

We can rewrite equations 5 and 6 replacing Y_τ by Z_τ. We have for the infinitesimal mean

$$\begin{aligned} \mathbb{E}^{z_i}\left[\{Z_{s+\Delta t} - z_i\}\right] &\cong z_i \cdot (1-(\beta_i+\delta_i)) + z_{i+1}(\beta_i\Delta\tau) + z_{i-1}(\delta_i\Delta\tau) - z_i \\ &= (\beta_i(z_{i+1}-z_i) + \delta_i(z_i - z_{i-1}))\Delta\tau \\ &= b(z_i)\cdot\Delta\tau, \end{aligned} \tag{8}$$

and for the infinitesimal variance:

$$\begin{aligned} \mathrm{Var}(Z_{s+\Delta t} - z_i) &= \mathbb{E}^{z_i}\left[\{Z_{\Delta t} - z_i\}^2\right] - \mathbb{E}^{z_i}\left[\{Z_{\Delta t} - z_i\}\right]^2 \\ &\cong (z_{i+1}-z_i)^2\cdot\beta_i\Delta\tau + (z_{i-1}-z_i)^2\cdot(\delta\Delta\tau - (z_i-z_i)^2(1-\beta_i-\delta_i)\Delta\tau \\ &= (\beta_i(z_{i+1}-z_i)^2) + (\delta_i(z_i-z_{i-1})^2)\Delta\tau \\ &= a(z_i)\cdot\Delta\tau. \end{aligned} \tag{9}$$

We have therefore two equations 8 and 9 with two unknowns δ_i and β_i. Solving the system we get:

$$\beta_i = \frac{(-1+z_i)\cdot z_i\cdot(-1-z_i^2\cdot\gamma + h\cdot(-1+2\cdot z_i)\cdot(z_i - z_{i-1})\cdot\gamma + z_i\cdot z_{i-1}\cdot\gamma)}{(z_i - z_{i+1})\cdot(z_{i-1}-z_{i+1})} \tag{10}$$

$$\delta_i = \frac{-((-1+z_i)\cdot z_i\cdot(-1-z_i^2\cdot\gamma + h\cdot(-1+2\cdot z_i)\cdot(z_i - z_{i+1})\cdot\gamma + z_i\cdot z_{i+1}\cdot\gamma))}{(z_i - z_{i-1})\cdot(z_{i-1}-z_{i+1})}. \tag{11}$$

Note that since we require that $\delta_i, \beta_i > 0 \; \forall i$, the range of the possible parameters γ depends on the choice of the states z_{i-1}, z_i, z_{i+1}, or on the grid. In particular if we use a uniform grid we get: $\{z_0, z_1, ..., z_{H-1}\} = \{0, \frac{1}{H-1}, ..., \frac{H-2}{H-1}, 1\}$ and $\beta_i = \frac{(-1+H-k)k(1+H^2+k\gamma+H(-2+h\gamma)-h(\gamma+2k\gamma))}{2(-1+H)^2}$ and $\delta_i = \frac{(-1+H-k)k(1+H^2-k\gamma-H(2+h\gamma)+h(\gamma+2k\gamma))}{2(-1+H)^2}$. Most likely the locus of interest is either dominant, co-dominant of recessive, i.e. $h \in \{0, \frac{1}{2}, 1\}$. In those three cases for a uniform grid the range of γ is easy to compute. If $h = \frac{1}{2}$ then $-2(H-1) < \gamma < 2(H-1)$, if $h = 0$, $-(H-1) < \gamma < (H-1)$, and if $h = 1$, $-\frac{(H-1)^2}{H-2} < \gamma < \frac{(H-1)^2}{H-2}$. In other words, we will need a large grid for high values of γ.

S1.2 Numerics

S1.2.1 Matrix exponentiation

We would like to compute the matrix exponential of the matrix Q and the matrix q^C for the conditional process. We will focus on the non conditional process as the conditional process follows easily. We use the convention of numbering the elements of a matrix starting from 0 to $H-1$ for the unconditional process, and from 1 to $H-2$ for the conditional process. We seek to compute

$$\exp(Qt),$$

where the $H \times H$ matrix Q is a tridiagonal matrix with all entries above and below the diagonal strictly positive. We implement two different approaches to compute the matrix exponentiation.

The first approach is a scaling and squaring algorithm with a Padé approximation. This approach is described in detail in Moler and Van Loan (2003) and is implemented in *SciPy*. This method works for a general matrix and takes advantage of the properties of the matrix Q.

The matrix Q is in general not symmetric ($\delta_i \neq \beta_i$ when $s \neq 0$). Nevertheless all eigenvalues are real. In particular two eigenvalues are 0 and the others are negative. Thus, when we

remove the first and last column and row, the resulting matrix is the tridiagonal matrix q^C. We can transform the matrix q^C into a symmetric matrix with a similarity transformation. More precisely, there exists a diagonal matrix

$$d = \begin{pmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ ... & ... & ... & ... \\ 0 & 0 & d_{H-3} & 0 \\ 0 & 0 & 0 & d_{H-2} \end{pmatrix} \tag{12}$$

such that $s = d^{-1}q^C d$ is a symmetric matrix. The d_i can be defined recursively as follows: $d_1 = 1$,$d_2 = \sqrt{\delta_2/\beta_1} \cdot d_1$, $d_3 = \sqrt{\delta_3/\beta_2} \cdot d_2$, ... Note that the square root exists since $\beta_i, \delta_i > 0$. The matrices q^C and s have the same eigenvalues, and the eigenvalues of a symmetric matrix are all real. In particular they are also eigenvalues of the original matrix Q. The two remaining eigenvalues of Q are the two zero eigenvalues (this can be seen writing the characteristic polynomials). Therefore all eigenvalues are real. We can build a matrix D adding a first and last row and column to the matrix d:

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_2 & 0 & 0 & 0 \\ 0 & ... & ... & ... & ... & 0 \\ 0 & 0 & 0 & d_{H-3} & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{H-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{13}$$

It follows that $R = D^{-1}QD$ symmetries the interior part of Q (the matrix q^C) and is a tridiagonal matrix as well. Since $s = d^{-1}qd$ is symmetric, there exists an orthogonal matrix,

o, such that $\ell = o^T s o$ is diagonal. This matrix ℓ has the following form:

$$\ell = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ ... & ... & ... & ... \\ 0 & 0 & \lambda_{H-3} & 0 \\ 0 & 0 & 0 & \lambda_{H-2} \end{pmatrix} \tag{14}$$

We can construct the matrix O as the matrix D before, with o in its center and adding first and last rows and columns with zeros everywhere but the diagonal entries $(0,0)$ and $(H-1,H-1)$. Then we see that $T = O^T R O$ has an inner part equal to ℓ the coefficients of the first and last lines remain equal to 0, and the coefficients on the first and last columns are non-zero. We denote $T(0,j) = v_{0,j}$ with $j = 1,\ldots,H-2$ and $T(H-1,j) = v_{H-1,j}$ with $j = 1,\ldots,H-2$. That is,

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ v_{0,1} & \lambda_1 & 0 & 0 & 0 & v_{H-1,1} \\ v_{0,2} & 0 & \lambda_2 & 0 & 0 & v_{H-1,2} \\ v_{0,..} & ... & ... & ... & ... & v_{H-1,..} \\ v_{0,H-3} & 0 & 0 & \lambda_{H-3} & 0 & v_{H-1,H-3} \\ v_{0,H-2} & 0 & 0 & 0 & \lambda_{H-2} & v_{H-1,H-2} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{15}$$

where the $v_{i,j} \neq 0$. We rewrite $T = \Lambda + V$ where

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 & 0 & 0 \\ 0 & ... & ... & ... & ... & 0 \\ 0 & 0 & 0 & \lambda_{H-3} & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{H-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{16}$$

and

$$V = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ v_{0,1} & 0 & 0 & 0 & 0 & v_{H-1,1} \\ v_{0,2} & 0 & 0 & 0 & 0 & v_{H-1,2} \\ v_{0,..} & ... & ... & ... & ... & v_{H-1,..} \\ v_{0,H-3} & 0 & 0 & 0 & 0 & v_{H-1,H-3} \\ v_{0,H-2} & 0 & 0 & 0 & 0 & v_{H-1,H-2} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{17}$$

We note that V is nilpotent and that $V\Lambda = 0$. It follows that for $k \geq 1$, $(\Lambda + V)^k = \Lambda^k + \Lambda^{k-1}V$, which we can see by induction. There is another identity that will be useful. We define:

$$\Lambda' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1/\lambda_1 & 0 & 0 & 0 \\ 0 & 0 & ... & 0 & 0 \\ 0 & 0 & 0 & 1/\lambda_{H-2} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{18}$$

where λ_1, λ_2... are the diagonal entries of l. We see that for $k \geq 2$, $\Lambda^{k-1} = \Lambda^k \Lambda'$. Since

$$\begin{aligned} T &= (DO)^{-1} Q (DO) \\ Q &= (DO) T (DO)^{-1} \\ Qt &= (DO) Tt (DO)^{-1}, \end{aligned} \tag{19}$$

we have:

$$\exp(Qt) = \sum_{k=0}^{\infty} \frac{1}{k!} (Qt)^k = \sum_{k=0}^{\infty} \frac{1}{k!} ((DO) Tt (DO)^{-1})^k = DO \left(\sum_{k=0}^{\infty} \frac{1}{k!} (Tt)^k \right) (DO)^{-1}. \tag{20}$$

Then,

$$\begin{aligned} \sum_{k=0}^{\infty} \frac{1}{k!} (Tt)^k &= \mathtt{I} + \sum_{k=1}^{\infty} \frac{1}{k!} ((\Lambda + V)t)^k \\ &= \mathtt{I} + \sum_{k=1}^{\infty} \frac{1}{k!} (\Lambda t)^k + \sum_{k=1}^{\infty} \frac{t^k}{k!} (\Lambda^{k-1} V) \\ &= \exp(\Lambda t) + tV + \left(\sum_{k=2}^{\infty} \frac{t^k}{k!} \Lambda^{k-1} \right) V \\ &= \exp(\Lambda t) + tV + \left(\sum_{k=0}^{\infty} \frac{t^k}{k!} \Lambda^k - \mathtt{I} - \Lambda t \right) \Lambda' V \\ &= \exp(\Lambda t) + tV + (\exp(\Lambda t) - \mathtt{I} - \Lambda t) \Lambda' V. \end{aligned} \tag{21}$$

Finally:

$$\exp(Qt) = DO \left(\exp(\Lambda t) + tV + (\exp(\Lambda t) - \mathtt{I} - \Lambda t) \Lambda' V \right) (DO)^{-1}. \tag{22}$$

In terms of computing time, this requires us to compute o using an algorithm for hermitian matrices, then to compute d by recursion and the rest should follow from matrix

multiplications. The advantage compared to the Padé approach described above is that most of the work is done once D and O are computed (only once) and reused for all time intervals.

In practice, the condition number of the matrix o can be very high leading to instabilities in the matrix exponentiation. Indeed the higher the condition number, the more sensitive the matrix will be to numerical operation. The condition number of our matrix can be of the order of 10^6 for large γ and is therefore ill-conditioned. Note that for the approximation of the diffusion process to the WF model, γ has to be on the order of 1. Thus, the matrix exponentiation becomes harder when the conditions for approximating the WF model with the diffusion are not necessarily met.

In order to overcome this problem, we implemented the matrix exponentiation in *C++* using a library, *mpack* (Nakata, 2010), for multiple precision arithmetic. The library *mpack* is a multiple precision arithmetic version of *LAPACK* and *BLAS*. Although this allows us to exponentiate the matrix for any γ in principle, it makes the matrix exponentiation step much slower. We therefore empirically test for which parameter range we require more precision than the double precision of *numpy* or *SciPy* that rely on *LAPACK*.

To do so, for a particular matrix $Q = Q(H, h, \gamma)$ we compute

$$\text{test}(Q) = \text{norm}\left((D \cdot O) \cdot (O^T D^{-1})\right) - \text{trace}\left((D \cdot O) \cdot (O^T D^{-1})\right), \tag{23}$$

where $\text{norm}(A) = \text{norm}((a_{ij})) = \sum_{i,j} |a_{ij}|$. The value of $\text{test}(Q)$ should be equal to 0. We choose a threshold value ϵ such that if $\text{test}(Q) > \epsilon$, we do not trust the default *SciPy* implementation and we invoke the higher precision computation. For this paper we used $\epsilon = 10^{-5}$.

We plot on Figure S1 the Boolean $\text{test}(Q) > \epsilon$ for different values of N_e and γ for $h = 0$. We can see on those plots that the matrix instability does depend on γ but not on the population size. For all the population sizes, the default implementation becomes unstable

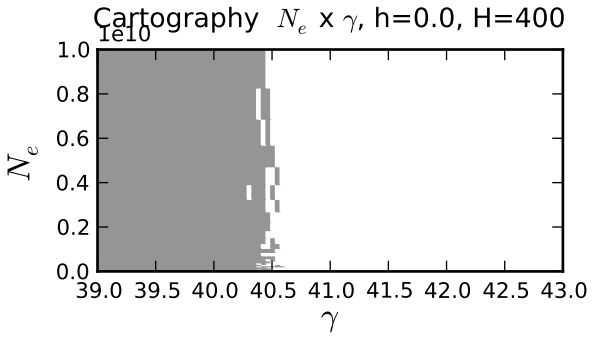

Figure S1: One example cartography of the parameter combination that require higher precision for $\epsilon = 10^{-5}$. We plot the result of the Boolean operation test($Q(H, h, \gamma, N_e) \leq \epsilon$ The legend is True for gray and white for False. We fix $H = 400$ and we plot N_e versus γ.

for $\gamma \gtrsim 40$.

To conclude, we use one existing method to exponentiate the matrix (Padé) and implemented one more method, with the possibility of increasing the double precision. Which method to use depends on the type of dataset and the parameter range one needs to explore. For high values of γ, if there are many time intervals, a method based on the spectral decomposition would be faster, otherwise the Padé approximation works well.

S1.3 Choice of grids

We investigated several grids inspired by Gutenkunst et al. (2009). No matter the parameters, to compute the likelihood we need to approximate the transition probabilities between the original frequency of the A allele, $\frac{1}{2N_e}$, and another frequency between 0 and 1. Although we could extrapolate, we instead use grids that all include the point $\frac{1}{2N_e}$.

The first is a uniform grid with a point added at $\frac{1}{2N_e}$. We call this grid the "uniform grid". Then we investigate a quadratic grid and an exponential grid. The last two grids were chosen so that, as opposed to the uniform grid, the distance between adjacent points changes smoothly.

As before, let's denote by $\{z_0, z_1, ..., z_{H-1}\}$ the state space of the one step process or

the grid. The quadratic grid is described by a cubic equation, i.e., the difference between adjacent points is quadratic. We will assume for simplicity of notation that H is a multiple of 20 (it is straightforward to generalize), and that $G = \frac{H}{10}$. We set the first $G+1$ points to form a uniform grid between 0 and $\frac{2}{2N_e}$, so that the median of this grid is $\frac{1}{2N_e}$. In other words, $z_j = \frac{j}{N_e G}$ for $0 \leq j \leq G$. Now we assume first that $\{q_0, \ldots, q_{H-G-1}\}$ is a uniform grid between 0 and 1. In other words, $q_0 = 0$, $q_{H-G-1} = 1$ and $q_j = \frac{j}{H-G-1}$. The remaining points are described by

$$z_{G+j} = aq_j^3 + bq_j^2 + cq_j + d \tag{24}$$

where $d = \frac{2}{2N_e}$, $c = \frac{1}{2N_e G}$, $b = -3(\frac{1}{H-G-1} + c + \frac{d}{H-G-1})\frac{1}{H-G-1}$, $a = -\frac{2}{3}b$.

The exponential grid will be defined as follows. If $\{u_0, ..., u_{H-1}\}$ is a uniform grid between -1 and 1 (i.e., $u_0 = -1$, $u_{H-1} = 1$ and $u_j = -1 + j\frac{2}{H-1}$), then the grid is

$$z_j = \frac{\frac{1}{1+\exp(-\xi u_j)} - \frac{1}{1+\exp(\xi)}}{\frac{1}{1+\exp(-\xi)} - \frac{1}{1+\exp(\xi)}}, \tag{25}$$

where ξ is a parameter that defines the density of the grid around the boundaries. We pick ξ such as $z_{\left[\frac{H}{10}\right]} = \frac{1}{2N_e}$, with [] denoting the integer part. To do so, we solve numerically the equation 25 for $j = \left[\frac{H}{10}\right]$.

We plot the grids of interest versus uniform grids and the spacing between each point in Figure S2.

For the neutral case, it is possible to compute the likelihood since the transition probabilities are known for the diffusion process (see e.g. Ewens (2004)). We plot the results in Figure S3 for a quadratic grid of size 100 for two samples of size $M = (4, 4)$ and number of A alleles $I = (1, 3)$, sampled at times $T = (-200, 0)$ for several values of N_e and t_0. The plots suggest that even for a grid of size 100 the one step process is a very good approximation of the diffusion process.

We compare the relative error between the diffusion and the one step process and demonstrate that, when we increase the grid size the one step process converges towards the diffusion

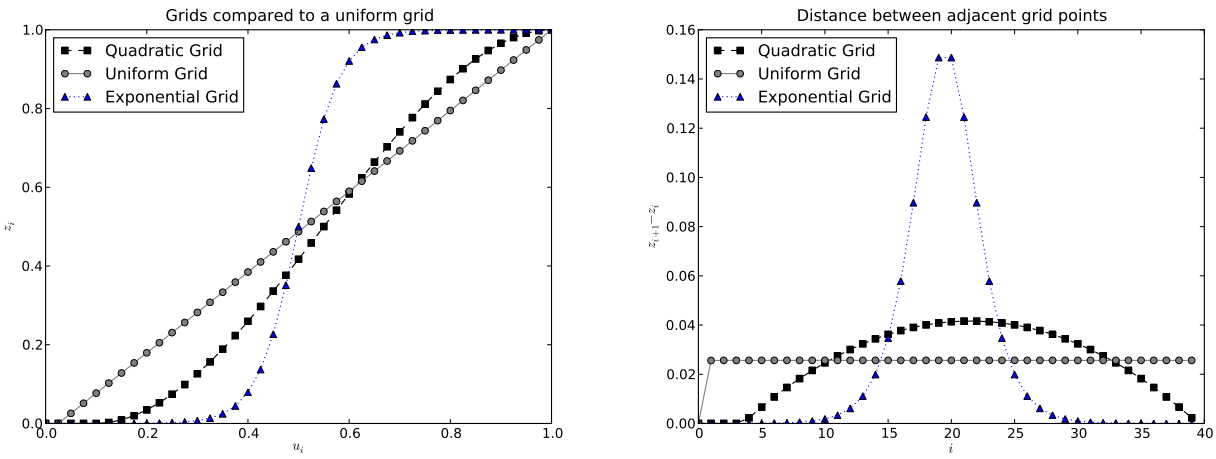

Figure S2: Description of three different grids tested of size H=41 and $N_e = 10^4$. Left: the grids are plotted against a uniform grid of points between 0 and 1. Right: the spacing of adjacent points.

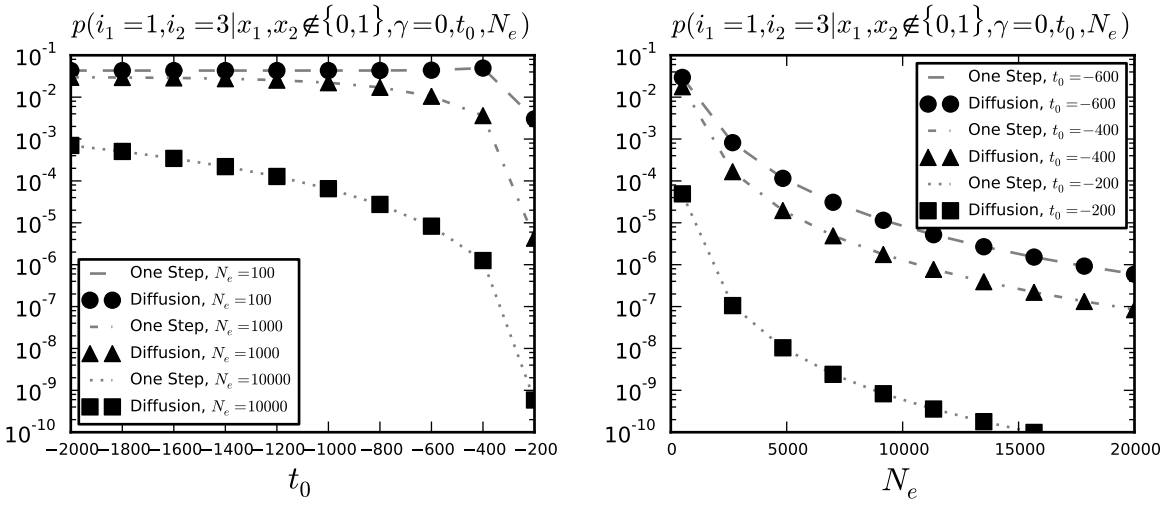

Figure S3: Likelihood for the neutral case for several values of N_e and t_0. The likelihood is for two samples taken at times -200 and 0 generations of size $M = (4, 4)$ and with $I = (1, 3)$ derived alleles. On the left (right), we fix N_e (respectively t_0) to several values and plot the likelihood versus t_0 (respectively N_e).

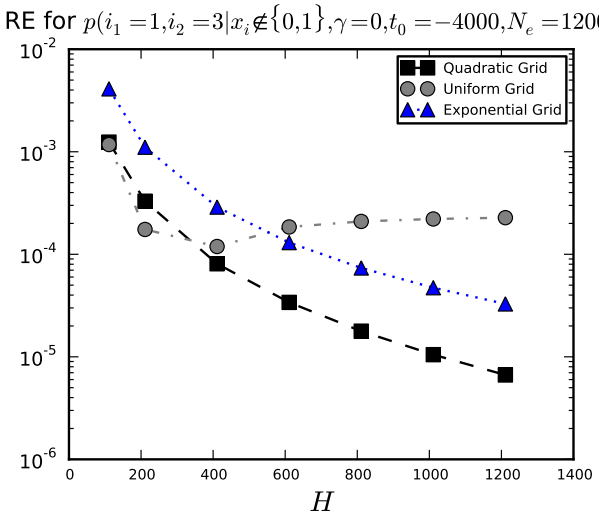

Figure S4: Relative error (RE) for the three grids discussed in 3.1 for the likelihood of 2 samples taken at times -3000 and 0, with $M = (4, 4)$. The parameter H describes the size of the grid. The y-axis is in logarithmic scale. In this example, the one step process converges towards the diffusion process faster when using the quadratic grid rather than the other two grids.

process. The results for a particular choice of parameters is shown in Figure S4 for the three grids. First we note that the one step process does converge as expected with increasing grid size. In this example, the convergence is faster for the quadratic grid. We looked at several combinations of parameters, and we observe that the quadratic grid and the exponential grid perform better than the uniform grid in general but that the ordering between the other two grids depends on the parameters. Indeed, if the allele age is close to the first sampling time a grid more refined around the frequency $\frac{1}{2N_e}$ performs better. In the applications below we will use a quadratic grid of size between 100 and 400.

S1.4 Simulations

We plot in Figure S5 the root mean square error (RMSE) for the simulations for the two sampling schemes, *i.e.*, 6 and 12 sampling times. See main text for discussion.

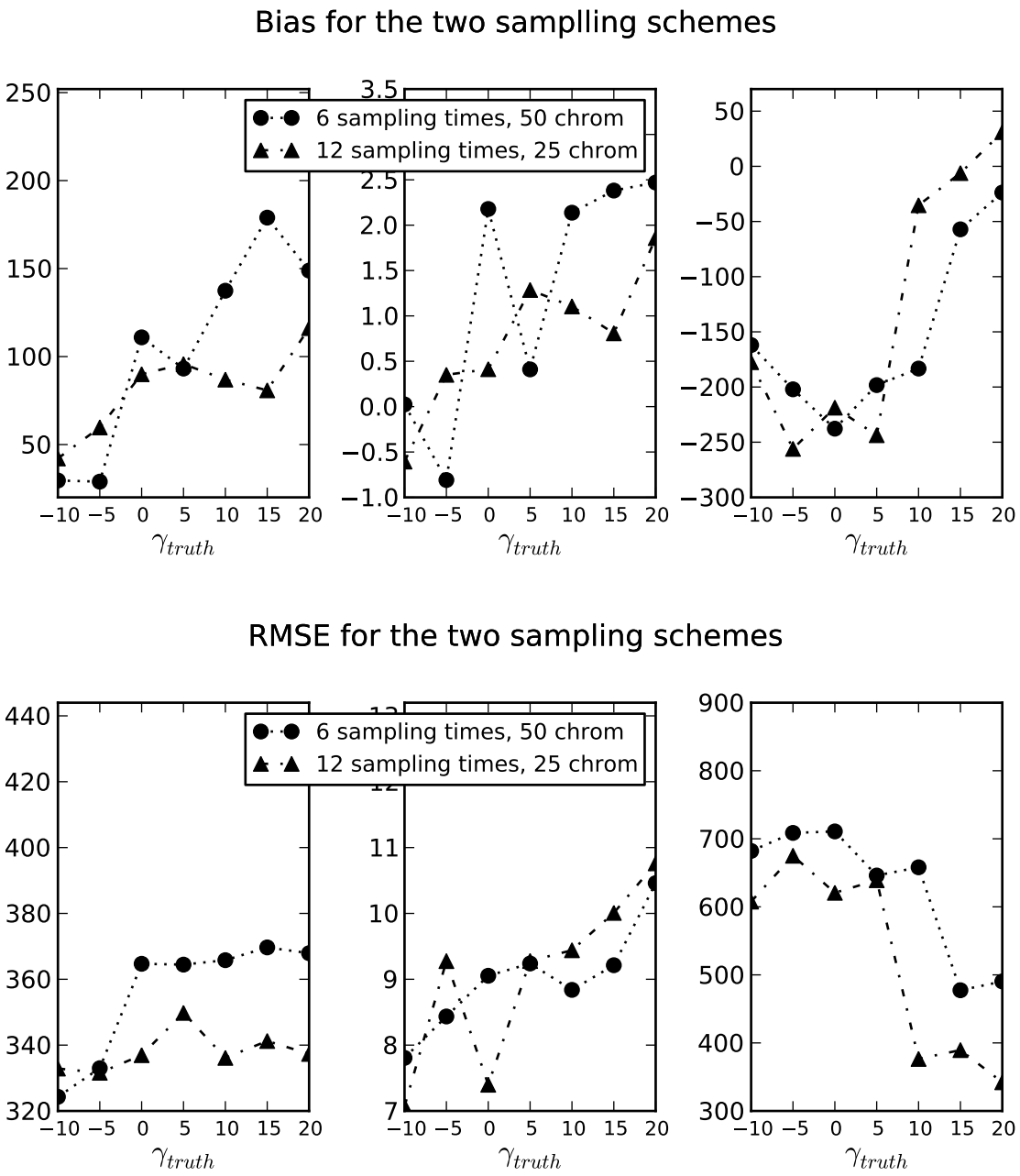

Figure S5: Bias (top plot) and RMSE (bottom plot) results for the MLEs for seven different sets of simulations presented in Figure 2. The left panel corresponds to the N_e estimates, the middle panel to results for γ estimates and the right panel for t_0 estimates, the order matching Figure 2.

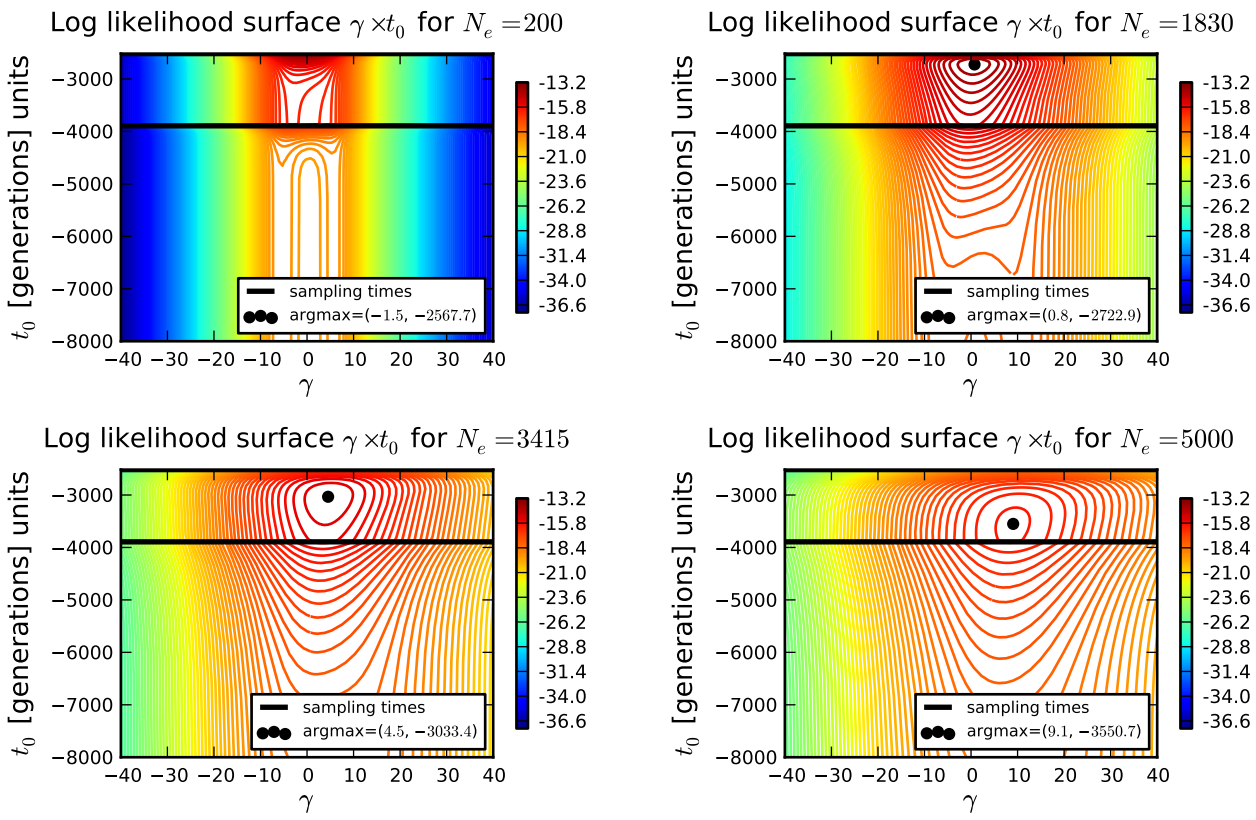

Figure S6: Likelihood surfaces for various values of N_e when analyzing the *ASIP* locus. In each case the local maximum is indicated.

S1.5 Real Data

We plot the likelihood surface for four values of N_e on Figure S6. As discussed in the main text, the higher the population size, the higher the selection coefficient, and the older the allele age that maximize the likelihood surface.

References

Durrett, R. (2008). *Probability Models for DNA Sequence Evolution* (2nd ed. ed.). Springer.

Ewens, W. J. (2004). *Mathematical Population Genetics* (second edition ed.). Springer.

Gutenkunst, R. N., R. D. Hernandez, S. H. Williamson, and C. D. Bustamante (2009). Inferring the joint demographic history of multiple populations from multidimensional SNP frequency data. *PLoS Genet 5*(10), e1000695.

Moler, C. and C. Van Loan (2003). Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later*. *SIAM Review 45*(1), 3–000.

Nakata, M. (2010, 6 August). The MPACK (MBLAS/MLAPACK); a multiple precision arithmetic version of BLAS and LAPACK. URL: http://mplapack.sourceforge.net/. Enter text here.