Bayesian model evidence as a practical alternative to deviance information criterion - Appendices

C. M. Pooley\textsuperscript{1,2} and G. Marion\textsuperscript{2}

1 The Roslin Institute, The University of Edinburgh, Midlothian, EH25 9RG, UK.

2 Biomathematics and Statistics Scotland, James Clerk Maxwell Building, The King's Buildings, Peter Guthrie Tait Road, Edinburgh, EH9 3FD UK.

Corresponding author: Dr C. M. Pooley
Appendix A: Details of steppingstone sampling (SS)

Theory

Equation (2) can be approximated by Monte Carlo integration through importance sampling [1]. This is a general technique in which the expression within an integral is written in terms of two parts: (a) a normalised distribution which can be sampled from and (b) whatever remains. Successive samples are then taken from (a) and used to evaluate (b). The integral can then be approximated by averaging over these evaluations, with the estimate becoming increasingly accurate as the sample number is increased. In the case of Eq.(2), we imagine that \( S \) samples are taken from \( \pi(\theta | x) \pi(\theta) \) (which are typically generated by first drawing \( \theta \) from the prior distribution \( \pi(\theta) \) then simulating \( x \) from \( \pi(x | \theta) \)) leading to the estimate

\[
P(y) = \frac{1}{S} \sum_{s=1}^{S} P(y | \theta_s, x_s), \quad (A1)
\]

which converges on \( P(y) \) in the limit \( S \rightarrow \infty \). Directly calculating \( P(y) \) using this approach is problematic for all but the simplest systems because the vast majority of samples \( \theta_s, x_s \) have a negligible observed data likelihood. This necessitates the use of more sophisticated techniques such as SS.

We define the shorthand \( p=P(y | \theta, x) \) and introduce the probability density function \( g(p)dp \), which represents the probability that a sample from the model has an observed data likelihood between \( p \) and \( p+dp \). In the limit \( S \rightarrow \infty \), Eq.(A1) can be rewritten as

\[
P(y) = \int_{0}^{\infty} p \ g(p) \ dp. \quad (A2)
\]

Consider running \( K \) different MCMC chains with inverse temperatures \( \phi_1 > \phi_2 > \ldots > \phi_K = 0 \). The probability density function for \( p=P(y | \theta, x) \) in the \( k \)-th chain is taken to be \( f_k(p) \). The Metropolis-Hastings condition in Eq.(4) ensures that states with higher observed data likelihood are preferentially selected compared to states sampled directly from the model. In particular, the relationship between \( f_k \) and \( g \) is given by

\[
w_k f_k = g p^\phi_k, \quad (A3)
\]

where \( w_k \) are unknown normalisation factors. Considering chain \( k=1 \), substituting Eq.(A3) into Eq.(A2) and integrating leads to \( P(y)=w_1 \). This can be written in the suggestive form

\[
P(y) = \frac{w_1}{w_2} \frac{w_2}{w_3} \ldots \frac{w_{K-1}}{w_K}, \quad (A4)
\]

where \( w_K=1 \) follows from Eq.(A3) for \( k=K \) because \( g \) and \( f_k \) are both normalised. Next we establish how to estimate the separate ratios in Eq.(A4). Multiplying Eq.(A3) by \( p^{-\phi_k} \) and equating \( g \) between adjacent chains gives

\[
w_k f_k p^{-\phi_k} = w_{k+1} f_{k+1} p^{-\phi_{k+1}}, \quad (A5)
\]

Multiplying this by \( p^{\phi_k} \), rearranging and integrating gives
\[
\frac{w_k}{w_{k+1}} = \int f_{k+1} P_{k+1} \, dp.
\] (A6)

MCMC from chain \( k+1 \) produces \( N \) samples labelled \( \theta_i^{k+1} \) and \( x_i^{k+1} \) (where \( i \) goes from 1 up to a total \( N \)) which can be used to calculate \( p = P(y | \theta_i^{k+1}, x_i^{k+1}) \). These represent correlated draws from the distribution \( f_{k+1} \). Consequently, Monte Carlo integration through importance sampling can again be used to estimate the integral in Eq.(A6)

\[
\frac{w_k}{w_{k+1}} = \frac{1}{N} \sum_{i=1}^{N} P(y | \theta_i^{k+1}, x_i^{k+1}) \theta_{k+1}^{k+1}.
\] (A7)

Substituting Eq.(A7) into Eq.(A4) yields the final estimator for the model evidence

\[
P(y) = \prod_{k=2}^{K} \left( \frac{1}{N} \sum_{i=1}^{N} P(y | \theta_i^{k}, x_i^{k}) \theta_{k+1}^{k+1} \right).
\] (A8)

where \( P(y) \to P(y) \) in the limit \( N \to \infty \).

**Implementation**

An “update” constitutes sequentially going over a complete set of possible proposals (e.g. in the simplest case we might go through each model parameter in turn and then each latent variable in turn) and accepting or rejecting these using the Metropolis-Hastings probability in Eq.(4). The SS algorithm can be described as follows:

**STEPPINGSTONE SAMPLING ALGORITHM:**

**Initialization:** We set \( \theta_i^1, x_i^1 \) to be some initial state at \( i = 1 \) on chain \( k=1 \) (typically this involves generating a sample from the model). This initial state is then updated \( N^{\text{init}} \) times which constitutes a burn-in phase. During this phase proposal distributions are optimally tuned (see Appendix B). The final state on chain \( k=1 \) is then used as the initial state for chain \( k=2 \), which then undergoes a burn-in period itself. This process is repeated for all \( K \) chains. The following two steps are then iterated \( N \) times:

**Step 1:** Update chains 1 to \( K-1 \). For chain \( K \) directly sample from the model (if this is possible to do, otherwise it can also be updated).

**Step 2:** Swap states between chains (see Appendix C for details).

**Output:** Use Eq.(6) to estimate the model evidence.

**Appendix B: Tuning proposal parameters**

The proposals used in this paper separately make changes to each individual parameter within the model. This is the simplest approach to take (although not necessarily the best, because potential correlations between parameters are ignored which can lead to poor mixing). Suppose \( \theta \) is one parameter within \( \Theta \). A new proposed value \( \theta_{i,p} \) on chain \( k \) is sampled from the normal distribution
\[ \theta_{j,p} \sim N(\theta_{j,p}^k, \lambda_{k,j}^2) \] (B1)

centered at the parameter’s current value on the chain \( \theta_{j}^k \) with standard deviation \( \lambda_{k,j} \). Tuning \( \lambda_{k,j} \) is important. If it is too large, very few proposals will be accepted and if too small, mixing will be slow. Motivated by adaptive MCMC \([2, 3]\), a robust heuristic method for optimising \( \lambda_{k,j} \) within the burn-in period is as follows (note, after burn-in the proposal distributions are fixed to ensure that detailed balance is strictly satisfied). Initially, \( \lambda_{k,j} \) is set to a small quantity. Each time a proposed change on parameter \( j \) is accepted \( \lambda_{k,j} \) is updated according to

\[ \lambda_{k,j}^{\text{new}} = \lambda_{k,j} \times 1.01, \] (B2)

and when rejected

\[ \lambda_{j}^{\text{new}} = \lambda_{k,j} \times 0.995. \] (B3)

These numerical factors are chosen for two reasons: Firstly, the updates in Eqs. (B2) and (B3) balance each other out when acceptance occurs around 33% of the time, leading to a steady state solution for \( \lambda_{k,j} \). Secondly, they are chosen to be sufficiently close to 1 to prevent large fluctuations in \( \lambda_{k,j} \) but sufficiently far to allow the steady state solution to be found within the burn-in period.

**Appendix C: Swapping between chains**

One way to improving mixing when running multi-temperature MCMC is to perform swapping of states between adjacent chains. This works in the following way. From Eq.(5) the probability for sample \( i \) in chain \( k \) having a particular state \( \theta_{i}^k, x_{i}^k \) is proportional to

\[ L_k \propto p_{k,i} \pi(x_{i}^k \mid \theta_{i}^k) \pi(\theta_{i}^k), \] (C1)

where we use the shorthand \( p_{k,i} = P(y \mid \theta_{i}^k, x_{i}^k) \). Taking all chains simultaneously, the joint probability is proportional to

\[ L_{\text{tot}} \propto \prod_{k=1}^{K} p_{k,i} \pi(x_{i}^k \mid \theta_{i}^k) \pi(\theta_{i}^k). \] (C2)

We now consider swapping states between two adjacent chains \( k \) and \( k+1 \). The total probability for this proposed state is proportional to

\[ L_{\text{tot}}^{\text{prop}} \propto p_{k+1,i} \pi(x_{i}^{k+1} \mid \theta_{i}^{k+1}) \pi(\theta_{i}^{k+1}) \times p_{k,i} \pi(x_{i}^k \mid \theta_{i}^k) \pi(\theta_{i}^k) \times \prod_{k \neq k, k+1} p_{k,i} \pi(x_{i}^k \mid \theta_{i}^k) \pi(\theta_{i}^k). \] (C3)

The corresponding Metropolis-Hastings probability is given by
\[
\max \left\{ \frac{L_{\text{prop}}}{L_{\text{tot}}}, 1 \right\} = \max \left\{ \left( \frac{P_{k_{s},i_{i},d}}{P_{k_{s},i_{i},d}} \right)^{k_{s} - k_{s+1}}, 1 \right\}.
\]

(C4)

Step 2 of the algorithm in Appendix A is implemented by looping from \(k_{s}=1\) to \(k_{s}=K-1\) and swapping states \(\theta_{k_{s}+1}^{i}, x_{k_{s}+1}^{i}\) and \(\theta_{k_{s}}^{i}, x_{k_{s}}^{i}\) with the probability given in Eq. (C4).

Swapping is generally computationally fast and almost always worth doing. It is of particular value to improve the mixing of chains with high temperature (low \(\phi\)). This is because the chain \(\phi_{K}=0\) typically samples directly from the model (i.e. it does not suffer from correlations between successive samples that are usually problematic for efficient MCMC) and so swapping these uncorrelated states into the higher temperature chains is especially beneficial.

**Appendix D: Model selection measures given MVN distributions**

Multiplying Eqs. (13) and (14) gives

\[
P(y | \theta)\pi(\theta) = P(y | \langle \theta \rangle)e^{-\frac{1}{2}(\theta-\langle \theta \rangle)\Sigma^{-1}(\theta-\langle \theta \rangle)} e^{-\frac{1}{2}(\theta-\langle \theta \rangle)^{\intercal}(\theta-\langle \theta \rangle)\Omega^{-1}(\theta-\langle \theta \rangle)} \frac{1}{(2\pi)^{\frac{N}{2}} \sqrt{\mid \Omega \mid}}.
\]

(D1)

In anticipation of this being MVN we write

\[
P(y | \theta)\pi(\theta) = C e^{-\frac{1}{2}(\theta-\mu)^{\intercal}\psi^{-1}(\theta-\mu)} \frac{1}{(2\pi)^{\frac{N}{2}} \sqrt{\mid \psi \mid}},
\]

where the constant \(C\), mean \(\mu\) and covariance matrix \(\psi\) need to be established. Equating Eqs. (D1) and (D2) leads to

\[
P(y | \theta)\pi(\theta) = P(y | \langle \theta \rangle)e^{-\frac{1}{2}(\theta-\langle \theta \rangle)\Sigma^{-1}(\theta-\langle \theta \rangle)} e^{-\frac{1}{2}(\theta-\langle \theta \rangle)^{\intercal}(\theta-\langle \theta \rangle)\Omega^{-1}(\theta-\langle \theta \rangle)} \frac{1}{(2\pi)^{\frac{N}{2}} \sqrt{\mid \Omega \mid}}
\]

\[= P(y | \langle \theta \rangle) e^{-\frac{1}{2}(\theta-\langle \theta \rangle)^{\intercal}(\Sigma^{-1}+\Omega^{-1})\theta+\frac{1}{2}(\langle \theta \rangle)^{\intercal}(\Sigma^{-1}+\Omega^{-1})\langle \theta \rangle+\frac{1}{2}(\langle \theta \rangle)^{\intercal}\Omega^{-1}\langle \theta \rangle+\frac{1}{2}(\langle \theta \rangle)^{\intercal}(\Sigma^{-1}+\Omega^{-1})\theta+\frac{1}{2}(\langle \theta \rangle)^{\intercal}(\Sigma^{-1}+\Omega^{-1})\langle \theta \rangle)} \frac{1}{(2\pi)^{\frac{N}{2}} \sqrt{\mid \Omega \mid}}.
\]

(D3)

Comparing terms on the second and third lines reveals the following relations hps:

\[
\psi = \left( \Sigma^{-1} + \Omega^{-1} \right)^{-1},
\]

\[
\mu = \psi \left( \Sigma^{-1} \langle \theta \rangle + \Omega^{-1} \bar{\theta} \right),
\]

\[
C = P(y | \langle \theta \rangle) e^{-\frac{1}{2}(\langle \theta \rangle)^{\intercal}(\Sigma^{-1}+\Omega^{-1})\langle \theta \rangle+\frac{1}{2}(\langle \theta \rangle)^{\intercal}\Omega^{-1}\langle \theta \rangle+\frac{1}{2}(\langle \theta \rangle)^{\intercal}(\Sigma^{-1}+\Omega^{-1})\langle \theta \rangle}.
\]

(D4)
Substituting these results into Eq. (D2) gives

\[
P(y | \theta) \pi(\theta) = P(y | \langle \theta \rangle) \sqrt{\frac{1}{2\pi}} e^{-\frac{1}{2} (y - \langle y \rangle)^T \Sigma^{-1} (y - \langle y \rangle)} \left( \frac{1}{2\pi} \right)^{n/2},
\]

(D5)

This expression can be integrated over parameter space (which effectively sets the square bracket to one) to give the final result in Eq. (15).

The expressions in Eq. (17) can be derived in the following way. Taking the log of the observed data likelihood in Eq. (13) gives

\[
\langle \log (P(y | \theta)) \rangle = \log \left( \frac{P(y | \langle \theta \rangle)}{P(y | \theta)} \right) - \frac{1}{2} \left( \theta - \langle \theta \rangle \right)^T \Sigma^{-1} \left( \theta - \langle \theta \rangle \right)
\]

(D6)

where \( \langle \ldots \rangle \) denotes an average over the posterior from Eq. (D5), \( \Delta \theta = \mu - \langle \theta \rangle \) is the difference in mean parameters between the posterior and observed data likelihood and \( \delta \theta = \theta - \mu \) represents the parameters relative to the posterior mean. Explicitly writing indices on matrices and using results such as \( \langle \delta \theta \rangle = 0 \) and \( \langle \delta \theta_i \delta \theta_j \delta \theta_k \rangle = \psi_{ij} \psi_{kl} + \psi_{ik} \psi_{jl} + \psi_{il} \psi_{jk} \) leads to

\[
\langle \log (P(y | \theta)) \rangle = \log \left( \frac{P(y | \langle \theta \rangle)}{P(y | \theta)} \right) - \frac{1}{2} \text{Tr} (\Sigma^{-1} \psi) + \Delta \theta^T \Sigma^{-1} \Delta \theta.
\]

(D7)

Substituting this into the expression for DIC in Eq. (9) leads to the analytical expression in Eq. (17).

The posterior variance in the observed data likelihood is given by

\[
\text{var} [\log (P(y | \theta))] = \frac{1}{4} \left( \left( \theta - \langle \theta \rangle \right)^T \Sigma^{-1} \left( \theta - \langle \theta \rangle \right) - \text{Tr} (\Sigma^{-1} \psi) + \Delta \theta^T \Sigma^{-1} \Delta \theta \right)^2
\]

(D8)

Using the results \( \langle \delta \theta_i \delta \theta_j \delta \theta_k \rangle = 0 \) and \( \langle \delta \theta_i \delta \theta_j \delta \theta_k \delta \theta_l \rangle = \psi_{ij} \psi_{kl} + \psi_{ik} \psi_{jl} + \psi_{il} \psi_{jk} \) leads to

\[
\text{var} [\log (P(y | \theta))] = \frac{1}{2} \text{Tr} (\Sigma^{-1} \psi \Sigma^{-1} \psi) + \Delta \theta^T \Sigma^{-1} \psi \Sigma^{-1} \Delta \theta.
\]

(D9)

Combining the results from Eqs. (D7) and (D9) and the expression for DIC in Eq. (10) gives the analytical expression in Eq. (17).

**Appendix E: Estimating computational efficiency**

This Appendix derives the number of MCMC updates \( U \) (where one “update” refers to performing Metropolis-Hastings proposals on each of the model parameters and latent variables) to estimate the model selection measures to within an uncertainty of \( \epsilon \).

**SS:** We first obtain the sampling error when the model evidence is estimated using Eq. (6). Taking the log of this expression gives the estimate
\[
\log(P(y)) = \sum_{k=2}^{K} h_k, \quad (E1)
\]

where

\[
h_k = \log\left(\frac{1}{N} \sum_{i=1}^{N} e^{l_{k,j} \phi_{k-1} - \phi_k}\right), \quad (E2)
\]

and \(l_{k,j} = \log\left(P(y \mid \theta_i^k, x_i^k)\right)\) is the log of the observed data likelihood for the \(i\)-th sample from chain \(k\).

Equation (E2) can be re-expressed as

\[
h_k = \langle l_k \rangle (\phi_{k-1} - \phi_k) + \log\left(\frac{1}{N} \sum_{i=1}^{N} e^{(l_{k,j} - \langle l_k \rangle \phi_{k-1} - \phi_k)}\right), \quad (E3)
\]

where \(\langle l_k \rangle\) represents the average of \(l_{k,j}\) over a large number of MCMC iterations. Performing the Taylor series expansion

\[
e^{(l_{k,j} - \langle l_k \rangle \phi_{k-1} - \phi_k)} = 1 + (l_{k,j} - \langle l_k \rangle)(\phi_{k-1} - \phi_k) + \frac{1}{2} (l_{k,j} - \langle l_k \rangle)^2 (\phi_{k-1} - \phi_k)^2 + \ldots \quad (E4)
\]

and truncating up to the first two terms (which is valid in the limit when \(K\) is large because \(\phi_{k-1} - \phi_k\) becomes small) leads to Eq. (E3) being approximated by

\[
h_k \approx \langle l_k \rangle (\phi_{k-1} - \phi_k) + \log\left(1 + (\phi_{k-1} - \phi_k) \frac{1}{N} \sum_{i=1}^{N} (l_{k,j} - \langle l_k \rangle)\right). \quad (E5)
\]

In the limit of large \(N\), the MCMC based estimate for the average \(\frac{1}{N} \sum_{i=1}^{N} l_{k,j}\) becomes closer and closer to its true value \(\langle l_k \rangle\). Consequently, the second term within the log expression on the RHS of Eq. (E5) becomes smaller and smaller. Using the fact that \(\log(1+x) \approx x\) for small \(x\), Eq. (E5) can be written

\[
h_k \approx \langle l_k \rangle (\phi_{k-1} - \phi_k) + (\phi_{k-1} - \phi_k) \frac{1}{N} \sum_{i=1}^{N} (l_{k,j} - \langle l_k \rangle). \quad (E6)
\]

The first term on the RHS is exact (and on substitution into Eq. (E1) would lead to the exact value for the model evidence). The second term, on the other hand, comes from noise associated with MCMC sampling. An estimate for the variance of this noise is given by

\[
\text{var}[h_k] \approx \frac{1}{N} \left( \langle \sum_{i=1}^{N} (l_{k,i} - \langle l_k \rangle) \langle \sum_{i=1}^{N} (l_{k,i} - \langle l_k \rangle) \rangle \right) (\phi_{k-1} - \phi_k)^2. \quad (E7)
\]

When \(i=i'\), the expectation on the RHS is simply given by the variance in \(l_{k,i}\), which we denote \(\sigma_i^2\). On the other hand, when \(i \neq i'\) the expectation is still non-zero due to correlations in \(l_{k,i}\) along the MCMC chain. Taking the limit in which \(N \to \infty\) (and using the concept of effective sample size as explained in Appendix F), Eq. (E7) can be written

\[
\text{var}[h_k] \approx \frac{1}{N} \left( \langle \sum_{i=1}^{N} (l_{k,i} - \langle l_k \rangle) \langle \sum_{i=1}^{N} (l_{k,i} - \langle l_k \rangle) \rangle \rangle \right) (\phi_{k-1} - \phi_k)^2. \quad (E8)
\]
where

\[ n_{k}^{\text{cor}} = 1 + 2 \sum_{r=1}^{\infty} F_r, \tag{E9} \]

represents a characteristic number of samples over which MCMC becomes uncorrelated and \( F_r \) is the autocorrelation function for \( l_i \) (see Appendix F for a definition).

Summing up the variances for the individual chains in Eq.(E1) leads to

\[ \text{var} \left[ \log \left( P(y) \right) \right] = \frac{1}{N} \sum_{i=1}^{N} \frac{\sigma^2_i}{N} \left( \phi_{i-1} - \phi_i \right)^2. \tag{E10} \]

\( N \) updates on \( K \) chains requires \( U = NK \) updates in all. Since the model selection measure is given by \(-2\log(P(y))\), equating 4 times Eq.(E10) with the desired variance \( \epsilon^2 \) leads to

\[ U = \frac{4}{\epsilon^2} K \sum_{i=2}^{\infty} n_{k}^{\text{cor}} \sigma^2_i \left( \phi_{i-1} - \phi_i \right)^2. \tag{E11} \]

**DIC_1**: The term that dominates the sampling error in Eq.(9) is \( \langle \log(P(y|\theta)) \rangle \). When running the MCMC algorithm this average is estimated from

\[ \langle \log(P(y|\theta)) \rangle = \frac{1}{N} \sum_{i=1}^{N} l_{i,i}. \tag{E12} \]

The variance in this quantity is given by

\[ \text{var} \left[ \langle \log(P(y|\theta)) \rangle \right] = \left( \left( \frac{1}{N} \sum_{i=1}^{N} \left( l_{i,i} - \langle l_i \rangle \right) \right)^2 \right) \]

\[ = n_{k}^{\text{cor}} \sigma^2_i \frac{1}{N}. \tag{E13} \]

Remembering that DIC_1 contains a contribution of \(-4\langle \log(P(y|\theta)) \rangle \) and that \( U = N \) because only one chain is used, we finally obtain

\[ U = \frac{16 n_{k}^{\text{cor}} \sigma^2_i}{\epsilon^2}. \tag{E14} \]

**DIC_2**: For large \( \sigma_i \) the term which dominates the uncertainty in Eq.(10) is the variance \( \text{var}[\log(P(y|\theta))] \). Within MCMC this is estimated using

\[ \text{var} \left[ \log \left( P(y | \theta) \right) \right] = \frac{1}{N} \sum_{i=1}^{N} \left( l_{i,i} - \langle l_i \rangle \right)^2. \tag{E15} \]

The expected variance in this quantity is given by

\[ \text{var} \left[ \text{var} \left[ \log \left( P(y | \theta) \right) \right] \right] = \left( \left( \frac{1}{N} \sum_{i=1}^{N} \left( l_{i,i} - \langle l_i \rangle \right)^2 - \sigma_i^2 \right) \right) \]

\[ = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'}^{N} \left( \left( l_{i,i} - \langle l_i \rangle \right)^2 - \sigma_i^2 \right) \left( \left( l_{i,i'} - \langle l_i \rangle \right)^2 - \sigma_i^2 \right). \tag{E16} \]
Following the same arguments as for Eq. (E8), this can be expressed as

\[
\text{var}
\left[
\text{var}
\left[
\log
\left(\frac{P(y | \theta)}{}\right)
\right]
\right] = \frac{1}{\tau^2} n_k^\text{cor} \sum_{i=1}^{N} \left(\left( \frac{\ell_{ij} - \langle \ell_i \rangle}{\sigma_i^2} \right)^2 \right),
\tag{E17}
\]

where correlations in the MCMC are accounted for by a new quantity

\[
h_k^\text{cor} = 1 + 2 \sum_{\tau=1}^{\infty} F_{\tau}^i
\tag{E18}
\]

which uses \( F_{\tau}^i \), the autocorrelation function for \( \frac{\ell_{ij} - \langle \ell_i \rangle}{\sigma_i^2} \) (instead of simply \( \ell_{ij} \)).

Assuming the observed data likelihood is approximately log-normally distributed, Eq. (E17) simplifies to

\[
\text{var}
\left[
\text{var}
\left[
\log
\left(\frac{P(y | \theta)}{}\right)
\right]
\right] = \frac{2 n_k^\text{cor} \sigma_i^4}{N}.
\tag{E19}
\]

Consequently the number of updates needed to calculate DIC \(_2\) to within an uncertainty of \( \varepsilon \) is

\[
U = \frac{8 \sigma_i^4 n_k^\text{cor}}{\varepsilon^2}.
\tag{E20}
\]

**Figure F:** (a) A typical trace plot and (b) a typical autocorrelation function.
Appendix F: Effective sample number

Given \( N \) correlated MCMC samples of some quantity \( \nu \), where in this paper the focus is on the log of the observed data likelihood \( \nu = \log(p(x|\theta^*, \Sigma^*)) \), the autocorrelation function can be approximated using

\[
F_\tau = \frac{1}{(N-\tau)\sigma^2} \sum_{i=1}^{N-\tau} (\nu_i - \langle \nu \rangle)(\nu_{i+\tau} - \langle \nu \rangle),
\]

where estimates for the average and variance of \( \nu \) are given by

\[
\langle \nu \rangle = \frac{1}{N} \sum_{i=1}^{N} \nu_i, \quad \sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\nu_i - \langle \nu \rangle)^2.
\]

The effective sample size is given by the actual sample number \( N \), correcting for correlations between successive samples:

\[
N_{\text{eff}}^k = \frac{N}{1 + 2 \sum_{\tau=1}^{\infty} F_\tau}.
\]

Correspondingly, we define the characteristic number of samples \( n_{\text{cor}}^k \) over which MCMC becomes uncorrelated is given by

\[
n_{\text{cor}}^k = 1 + 2 \sum_{\tau=1}^{\infty} F_\tau.
\]

When calculating \( N_{\text{eff}}^k \) or \( n_{\text{cor}}^k \) based on actual MCMC results, however, clearly the sums in Eqs. (F3) and (F4) cannot go to infinity. We now discuss how these quantities can be estimated in practice. Figure F shows a typical trace plot along with the corresponding autocorrelation function. Note, this function exhibits considerable fluctuations for large \( \tau \), and these can generate unwanted bias. The simplest way to deal with them is to truncate the sums in Eqs. (F3) and (F4) up to a maximum size \( \tau_{\text{max}} \), which is defined to be the largest even value of \( \tau \) for which the following condition holds\(^1\) (see [4]):

\[
F_\tau + F_{\tau+1} > 0.
\]

Essentially this represents the point at which the autocorrelation function intersects the \( x \)-axis, as illustrated by the vertical line in Fig. H(b).

Appendix G: Evidence-based model selection using MCMC

In cases in which the number of potential models is large, one possibility to tackle model selection is through applying an MCMC scheme over model space (see, for example, [5]). This makes use of the

\(^1\) The reason Eq. (F5) is not simply \( F_{\tau+1} > 0 \) is down to a curious quirk of autocorrelation functions generated from MCMC: in the limit of large sample size the sum of adjacent autocorrelation values is guaranteed to be positive, whereas individual values are not.
model evidence (which itself is calculated using a separate MCMC scheme such as SS) as the basis for jumping from one model to another (see algorithm below). Note, this is in contrast to reversible-jump MCMC that relies on a single MCMC scheme containing all models and specially designed proposals for jumping between models (which in practice can often be hard to implement\(^2\)).

The following example is for the linear regression model in section 5.1:

| MODEL SELECTION FOR LINEAR REGRESSION MODEL: |
|-----------------------------------------------|
| **Initialisation:** Set index \(i=1\) and start with a model which incorporates all regressors, \(i.e. \kappa_j' = 1\). Calculate an estimate of the model evidence \(P_i\) using SS. |
| **Step 1:** Randomly select a regressor \(j_s\) and flip its status to form a proposed state \((i.e. \kappa_{p,j_s} = 1 - \kappa_{j_s}, \text{and } \kappa_{p,j} = \kappa_j' \text{ for } j \neq j_s)\). |
| **Step 2:** Calculate an estimate of the model evidence \(P_p\) for the proposed model. |
| **Step 3:** Accept the proposal with a Metropolis-Hastings probability \[
\max \left\{ \frac{P_p \pi(\kappa^p)}{P_i \pi(\kappa^i)} ; 1 \right\} \tag{F6} \]
\((\pi(\kappa)\text{ is the prior probability for a model defined by vector } \kappa)\) and set \(P_{i+1} = P_p, \kappa^{i+1} = \kappa^p\text{ else set } P_{i+1} = P_i, \kappa^{i+1} = \kappa^i\). |
| **Step 4:** Increment \(i\) by 1 and jump to step 1. |

The validity of this algorithm rests on the fact that SS generates an unbiased estimate of the true model evidence (see [6] for a discussion of this).

**Appendix H: DIC-based model selection using stepwise regression**
Here the idea is to start at a particular model and make a succession of stepwise changes which aim to minimise the model selection measure [7], in this case DIC. Implementation can be achieved in one of three ways:

- **Forward selection** - Start with \(\kappa \equiv 0\) for \(j=1...J'\) and calculate the DIC. Go though each \(j\) in turn and calculate the DICs for new models for which \(\kappa_j = 1\). Select which of these changes \(j_s\) produces the greatest reduction in DIC and permanently set \(\kappa_{j_s} = 1\). Repeat this procedure until no further reduction in DIC is possible.

\(^2\) For example, adding an extra compartment into the epidemic models from section 5.3 means revising the event time lines for all the individuals. Ideally this would be achieved by sampling these new events from the posterior of the new model. Such sampling, however, is usually not possible (or sufficiently accurate) leading to negligibly small acceptance probabilities for switching between models.
**Backward elimination** - Start with $\kappa_j=1$ for $j=1..J'$ and calculate the DIC. Go though each $j$ in turn and calculate the DICs for new models for which $\kappa_j=0$. Select which of these changes $j$, produces the greatest reduction in DIC and permanently set $\kappa_j=1$. Repeat this procedure until no further reduction in DIC is possible.

**Bidirectional elimination** – A combination of the two approach above which allows for both addition (i.e. $\kappa_j=1$) and removal (i.e. $\kappa_j=0$) of regressors.

**Appendix I: Calculating CPU time**

Here we outline a procedure for calculating the CPU time needed to estimate a model selection measure within an uncertainty of 0.2. For illustrative purposes we consider SS for estimating $-2\log(P(y))$, but the same basic method is also used for the DIC measures.

As the number of MCMC iterations $N$ increases in Eq. (6) the more accurate the estimate for $P(y)$ becomes. The black line in Fig. I(a) shows that $-2\log(P(y))$ converges to the true solution (horizontal dashed line) as a function of computational time. In isolation, this curve cannot be used to estimate the time taken to reach a certain degree of uncertainty. Instead, we consider running the algorithm multiple times (100 in total), with some of these different runs illustrated by the grey curves in Fig. I(a). The standard deviation between these runs can then be used to estimate the uncertainty in the estimate for $-2\log(P(y))$ as a function of CPU time. An example of this is shown in Fig. I(b).

MCMC is a sampling procedure, so uncertainty in the model selection measure is expected to scale as the square root of the number of samples. Because Fig. I(b) is a log-log plot, such a scaling is represented by a line of slope $-\frac{1}{2}$ (solid red line), which can be fitted to the data. The point where this line intersects the desired uncertainty of 0.2 (blue filled circle) is used as the estimate for CPU time. To generate the figures in the results section, four data sets were first simulated and the average CPU time for inference was calculated (with $2\times10^7$ burn-in updates and $N=10^6$ sampling updates for SS and $N=10^6$ sampling updates for DIC).

**Figure 1**: (a) Shows estimates for $-2\log(P(y))$ as a function of CPU time for a linear regression model with $J_{sim}=J_{sel}=5$ regressors. (b) The uncertainty in the approximations for $-2\log(P(y))$ across 100 independent runs.
Appendix J: Details of MCMC for the linear regression model

The following sets of proposals are used to perform a single “update” (note the explicit chain index \( k \) is omitted for clarity).

**UPDATE**

**Regressors** – Each regressor \( \beta_j \) is considered in turn (where \( j \) runs from 1 to \( J_{\text{sel}} \)) and a new value is proposed from the normal distribution

\[
\beta_{j,p} \sim N(\beta_{j,i}, \lambda^2_{\beta_j})
\]  

(G1)

centred at \( \beta_{j} \) (the current value for \( \beta_j \) on the chain being updated) with variance \( \lambda^2_{\beta_j} \).

**Residual Variance** – A new value is proposed according to

\[
\eta^2_p \sim N(\eta^2_i, \lambda^2_{\eta}).
\]  

(G2)

The parameters \( \lambda^2_{\beta_j} \) and \( \lambda^2_{\eta} \) are automatically tuned during burn-in to give an average acceptance probability of approximately 33% (see Appendix B for details).

In the case of DIC, updates are sequentially performed on the posterior chain and the proposals are accepted with the usual Metropolis-Hastings probability

\[
\left\{ \frac{P(y \mid \theta_{p,1}, x_p) \pi(x_p \mid \theta_{p,1}) \pi(\theta_{p,1})}{P(y \mid \theta_{1,p}, x_1) \pi(x_1 \mid \theta_1) \pi(\theta_1)} \frac{j_{p \to 1}}{j_{1 \to p}}, 1 \right\],
\]  

(G2)

where \( \theta_i, x_i \) are the current model parameters and latent variables and \( \theta_{p,1}, x_p \) are the proposed set.

In the case of SS each chain \( k \) is updated sequentially, but this time proposals are accepted with a modified Metropolis-Hastings probability

\[
\left\{ \left( \frac{P(y \mid \theta_{p,1}, x_p)}{P(y \mid \theta_{1,p}, x_1)} \right)^{\phi_k} \frac{\pi(x_p \mid \theta_{p,1}) \pi(\theta_{p,1})}{\pi(x_1 \mid \theta_1) \pi(\theta_1)} \frac{j_{p \to 1}}{j_{1 \to p}}, 1 \right\},
\]  

(G2)

which incorporates the chain’s inverse temperature \( \phi_k \).

To generate the results in the paper \( 2 \times 10^5 \) burn-in updates were followed by \( N=10^5 \) sampling updates in the case of SS and \( N=10^5 \) sampling updates in the case of DIC.

Appendix K: Calculating the relationship matrix \( A \)

To generate the population we first consider \( P=50 \) individuals in the founder population. We then go through each of the 50 individuals in the 1st generation and randomly select their sire and dam \((s, \text{ and } d)\) from the founder population (avoiding cases in which sire and dam are identical). This process is repeated for the 2nd and 3rd generations to make up a randomly mated population.
Calculation of the inverse relationship matrix $A^{-1}$ is actually easier than calculating $A$ (and considerably computationally faster for large matrixes because it is sparse). The following procedure is used:

1) Set all element of $A^{-1}$ to zero.
2) Loop over founder population members $r$ and increment $A_{rr}$.
3) Loop over non-founder population members $r$ and add 2 to $A_{rr}$, -1 to $A_{s,r}$, $A_{r,s}$, $A_{d,r}$, and $A_{d,s}$.

Because the relationship matrix is constant it becomes unnecessary to calculate the determinant of $A$ in Eq.(24), as it cancels out during Metropolis-Hastings acceptance probabilities.

Appendix L: Details of the MCMC for the mixed model
The following sets of proposals are used to perform a single update (note the explicit chain index $k$ is omitted for clarity).

**UPDATE**

**Regressors** – Each regressor $\beta_j$ is considered in turn (where $j$ runs from 1 to $J_{sel}$) and a new value is proposed from the normal distribution

$$\beta_{j,p} \sim N(\beta_{j,i}, \lambda_{\beta_j}^2)$$  \hspace{1cm} (11)

centred at $\beta_{j,i}$ (the current value for $\beta_j$ on the chain being updated) with variance $\lambda_{\beta_j}^2$.

**Variances** – A new value for the environmental variance is proposed according to

$$\eta_p^2 \sim N(\eta_i^2, \lambda_{\eta}^2),$$  \hspace{1cm} (12)

and for the genetic variance

$$\omega_p^2 \sim N(\omega_i^2, \lambda_{\omega}^2).$$  \hspace{1cm} (13)

Note, we use here a simple Metropolis-Hastings random walk update for the variance components. Based on marginalisation of the posterior, it is possible to use Gibbs sampling for both of these quantities (as is usually done). However because of the prior distributions placed on $\eta^2$ and $\omega^2$ (uniform between 0.1 and 2) these Gibbs proposals frequently get rejected for high temperature chains, actually making this approach slower.

**Random effects** – When combining terms from the observed data likelihood in Eq.(23) and the latent process likelihood in Eq.(24), the posterior marginalised for a specific random effect $u_r$ is described by a normal distribution. Gibbs sampling from this distribution is achieved through

$$u_{r,j+1} \sim N \left( \frac{1}{\lambda_{u_r}^2} \left( \sum_j X_{rj} \beta_j - C_{r,r} u_{r,r} \right), \frac{\sigma^2}{\lambda_{u_r}^2} \right).$$  \hspace{1cm} (14)
where matrix \( \mathbf{C} \) is defined by

\[
\mathbf{C} = \mathbf{I} + \mathbf{A}^{-1} \frac{\eta_i^2}{\phi \omega_i^2}.
\]

(15)

Here, \( \mathbf{I} \) and \( \mathbf{A} \) are the identity and relationship matrices, respectively. The notation \( \mathbf{C}_{ir} \) denotes the row in matrix \( \mathbf{C} \) excluding the element \( \mathbf{C}_{ir} \), and \( \mathbf{u}_{ir} \) is a vector of random effects excluding \( u_{ir} \) for the current state \( i \).

The parameters \( \lambda_{\beta_i}^2, \lambda_{\eta_i}^2 \) and \( \lambda_{\omega_i}^2 \) are automatically tuned during burn-in to give an acceptance probability of approximately 33% (see Appendix B for details).

In the actual code some effort has been made to speed up proposals by pre-calculating computationally time consuming parts of the observed data likelihood and latent process likelihood, and then repeating certain types of proposal multiple times using a faster to calculate version of these quantities.

In the case of DIC, updates are sequentially performed on the posterior chain and the proposals are accepted with the usual Metropolis-Hastings probability

\[
\left\{ \frac{P(y \mid \theta_p, x_p) \pi(x_p \mid \theta_p) \pi(\theta_p) \, j_{p-1}}{P(y \mid \theta_i, x_i) \pi(x_i \mid \theta_i) \pi(\theta_i) \, j_{i-1}}, 1 \right\},
\]

(11)

where \( \theta_i, x_i \) are the current model parameters and latent variables and \( \theta_p, x_p \) are the proposed set.

In the case of SS each chain \( k \) is updated sequentially, but this time proposals are accepted with a modified Metropolis-Hastings probability

\[
\left\{ \frac{P(y \mid \theta_p^k, x_p^k)}{P(y \mid \theta_i^k, x_i^k)} \frac{\pi(x_p^k \mid \theta_p^k) \pi(\theta_p^k) \, j_{p-1} \phi_k}{\pi(x_i^k \mid \theta_i^k) \pi(\theta_i^k) \, j_{i-1} \phi_k}}, 1 \right\},
\]

(12)

which incorporates the chain’s inverse temperature \( \phi_k \).

To generate the results in the paper \( 2 \times 10^7 \) burn-in updates were followed by \( N=10^5 \) sampling updates in the case of SS and \( N=10^6 \) sampling updates in the case of DIC. In the case of DIC\(_3 \) and DIC\(_4 \), every twenty updates the parameter
Appendix M: The Doob-Gillespie algorithm

The Doob-Gillespie algorithm ([8]) provides a way to simulate from discrete state-space continuous time Markov processes such as the compartmental models shown in Fig. 5. For illustrative purposes we focus on the SEIR model. The algorithm builds up a sequence of events in the following way:

**Initialization:** At time $t_0=0$ all individuals are assumed to be susceptible apart from one that is taken to be infected. An index counting event number is set to $e=1$.

**Step 1:** Calculate the time to the next event. This is done by first calculating $W$, the total rate at which events (of any type) occur. Each individual $r$ is classified as either being susceptible $S$, exposed $E$, infectious $I$ or recovered $R$. The arrows in Fig. 5 denote transition rates between these states. An individual classified as $I$, therefore, has a transition rate $\rho_r=\gamma$ of recovering. The total event rate is found by summing over all individuals $W=\Sigma_r \rho_r$. Drawing a random inter-event time $\Delta t$ from the exponential distribution $W e^{-W \Delta t}$ is achieved by sampling a new event time from

$$t_e = t_{e-1} - \frac{\log(u)}{R},$$  \hspace{1cm} (K1)

where $u$ is a (uniform) randomly generated number between 0 and 1.

If $t_e$ is greater than the simulation time $t_{\text{max}}$ the algorithm is terminated.

**Step 2:** Select which individual $q$ undergoes a disease status transition with probability

$$\text{Prob}(q = r) = \frac{\rho_r}{W}. \hspace{1cm} (K2)$$

**Step 3:** Update the state of individual $q$.

**Step 4:** Increment $e$ and jump to step 1.

Appendix N: Details of the MCMC for epidemiological models

The following sets of proposals are used to perform a single update (note the explicit chain index $k$ is omitted for clarity).

**UPDATE**

**Parameters** – The number of parameters is model dependent. For example the SI model only contains a single parameter $\theta=(\beta)$, whereas the SEIR model has three $\theta=(\beta, \nu, \gamma)$. During the MCMC procedure we consider each parameter $j$ in turn and propose a new value according to

$$\theta_{j,p} \sim N(\theta_{j,j}, \lambda^2_j). \hspace{1cm} (L1)$$

**Move events** – An event $e$ is randomly selected from the $E$ events within $\xi$. A new event time is proposed according to
Note, this new time does not include a subscript \( e \) because typically there is a reordering of events under this transformation. If the move is inconsistent (e.g., a recovery event in placed before an infection event) the proposal is immediately rejected.

**Add Event** – An individual \( r \) is randomly selected from the population. The final event \( e \) that affects \( r \) is established (if no such event exists we set \( e=0 \) to be the event at the beginning of the simulation). If that state is not terminal (i.e., there exists a transition from that state to another) then the transition is added to the time line at a point randomly selected between \( t_e \) and \( t_{\text{max}} \) (the maximum simulation time). The ratio of forward to reverse proposal probabilities in Eq.(4) is given by

\[
\frac{f_{i\to p}}{f_{p\to i}} = \frac{t_{\text{max}} - t_e}{t_e - t_{\text{max}}}. \tag{L3}
\]

**Remove Event** – An individual \( r \) is randomly selected from the population. The final event \( e \) that affects \( r \) is established (if no such event exists then no proposal is made). Event \( e \) is removed. In this new state the final event \( e' \) that affects \( r \) is established (if no such event exists then we set \( e'=0 \) to be the event at the beginning of the simulation). In this case, the ratio of forward to reverse proposal probabilities in Eq.(4) is given by

\[
\frac{f_{i\to p}}{f_{p\to i}} = \frac{1}{t_{\text{max}} - t_{e'}}. \tag{L4}
\]

The parameters \( \lambda^2_j \) and \( \lambda^2_i \) are automatically tuned during burn-in to give an acceptance probability of approximately 33% (see Appendix B for details).

A single “update” consists of twenty sets of proposals for each of the parameters (for which the likelihood calculation is sped up by using the fact that \( \xi \) is fixed) followed by five randomly selected move event proposals, and finally two randomly selected add/remove event proposals. This ensured an approximate 20/80 split in CPU time between updating model parameters and latent variables.

In the case of DIC, updates are sequentially performed on the posterior chain and the proposals are accepted with the usual Metropolis-Hastings probability

\[
\begin{pmatrix}
    P(y | \theta_p, x_p) \pi(x_p | \theta_p) \pi(\theta_p) \frac{f_{p\to i}}{f_{i\to p}} - 1 \\ P(y | \theta, x) \pi(x | \theta) \pi(\theta) \frac{f_{i\to p}}{f_{p\to i}} - 1
\end{pmatrix},
\tag{M1}
\]

where \( \theta, x \) are the current model parameters and latent variables and \( \theta_p, x_p \) are the proposed set.

In the case of SS each chain \( k \) is updated sequentially, but this time proposals are accepted with a modified Metropolis-Hastings probability
\[
\left\{ \frac{P(y | \theta^p, x^p)}{P(y | \theta, x^l)} \right\}^{k} \frac{\pi(x^p | \theta_p) \pi(\theta_p)}{\pi(x^l | \theta_l) \pi(\theta_l)} \frac{1}{j^p 
rightarrow j},
\]

which incorporates the chain’s inverse temperature \(\phi_k\).

To generate the results in the paper 10⁶ burn-in updates were followed by \(N=10^5\) sampling updates in the case of SS and \(N=10^6\) sampling updates in the case of DIC. The prior for \(\beta\) were chosen to be flat and in the range 0.0004-0.01, for \(\tau\) in the range 0.02-0.2, and for \(\gamma\) in the range 0.01-0.1.

### Appendix O: Table showing model selection measures

This table shows the model selection measures used in Fig. 4(a) and (b).

| \(J_{sel}\) | \(-2\log(P(y))\) | DIC \(_3\) | DIC \(_5\) |
|---|---|---|---|
| 0 | 327.4 | 254.0 | 485.9 |
| 1 | 303.7 | 276.3 | 337.8 |
| 2 | 296.6 | 265.8 | 316.5 |
| 3 | 286.3 | 248.1 | 316.5 |
| 4 | **278.9 - Selected** | 226.6 | 322.4 |
| 5 – True model | 281.5 | **230.5** | **314.5** |
| 6 | **281.4** | 224.7 - Selected | 313.2 - Selected |
| 7 | 285.6 | 226.2 | 316.2 |
| 8 | 288.7 | 226.7 | 317.3 |
| 9 | 293.0 | 228.9 | 320.2 |
| 10 | 294.1 | 221.5 | 324.9 |

Note, the red entries show that whilst \(J_{sel}=4\) is chosen to be the “best” model based on model evidence, those models with \(J_{sel}=5\) and \(J_{sel}=6\) are also candidates as they lie within 4.6 units of \(J_{sel}=4\).

### References

[1] Evans, M. & Swartz, T. 2000 *Approximating integrals via Monte Carlo and deterministic methods*. Oxford; New York, Oxford University Press; ix, 288 p.
[2] Andrieu, C. & Thoms, J. 2008 A tutorial on adaptive MCMC. *Stat Comput* **18**, 343-373.
[3] Roberts, G.O. & Rosenthal, J.S. 2009 Examples of Adaptive MCMC. *J Comput Graph Stat* **18**, 349-367.
[4] Geyer, C.J. 1992 Practical Markov Chain Monte Carlo. *Statist. Sci.* **7**, 473-483.
[5] Chipman, H., George, E.I. & McCulloch, R.E. 2001 The Practical Implementation of Bayesian Model Selection. In *Model selection* (ed. P. Lahiri), pp. 65-116. Beachwood, OH, Institute of Mathematical Statistics.
[6] Andrieu, C., Doucet, A. & Holenstein, R. 2010 Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B* **72**, 269-342.
[7] Hocking, R.R. 1976 A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear Regression. *Biometrics* **32**, 1-49.
[8] Gillespie, D.T. 1977 Exact Stochastic Simulation of Coupled Chemical Reactions. *J. Phys. Chem.* **81**, 2340-2361.
