Biophysical Journal, Volume *111*

Supplemental Information

Bayesian Statistical Inference in Ion-Channel Models with Exact Missed Event Correction

Michael Epstein, Ben Calderhead, Mark A. Girolami, and Lucia G. Sivilotti

Bayesian Statistical Inference in Ion-Channel Models with Exact Missed Event Correction

Dr M. Epstein[1,2], Dr B. Calderhead[1], Prof. M.A. Girolami[3], and Prof. L.G. Sivilotti[4]

[1]Department of Mathematics, Imperial College London, London, UK
[2]CoMPLEX, University College London, London, UK
[3]Department of Statistics, University of Warwick, Coventry, UK
[4]Department of Neuroscience, Physiology & Pharmacology, University College London, London, UK

June 14, 2016

Supporting Materials

S1 Bayesian Overview

Bayesian approaches in Systems Biology are increasingly popular for characterising the inherent uncertainty observed in biological systems. Such a formalism provides a mathematically consistent framework for reasoning with and propagating uncertainty, from model parameterisation through to model prediction.

Bayesian reasoning may be implemented through the specification of a probabilistic model assumed to be the generative process of the data and the specification of prior probability distributions that incorporate existing knowledge regarding the model parameters which are to be estimated are unknown. In the Bayesian framework we model the parameters and the uncertainty regarding their true, fixed values as random variables. The aim is to update our knowledge of the model parameters by gathering additional experimental data and applying Bayes' theorem, which combines the prior and the model likelihood to produce a posterior distribution over the parameter values. These distributions express our updated knowledge of the model's parameters after taking into account the experimental data. Our use of Uniform prior distributions results in point estimates being derived

from estimating the mode of the posterior being the same values as the point estimates derived from an ML analysis.

Mathematically, Bayes' rule is simply the application of the rules of joint and conditional probability, which may be stated as,

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)} = \frac{p(y \mid \theta)p(\theta)}{\int p(y \mid \theta)p(\theta)d\theta} \tag{1}$$

where θ is the vector of unknown parameters, and y is the observed data. The main benefit of the Bayesian approach is that uncertainty about parameter estimates is directly encoded within the well defined posterior probability distribution obtained after examining the experimental data, rather than estimating a single "true" parameter value with accompanying assumptions about estimation errors. Posterior distributions can then be sampled to propagate parameter uncertainty through to the predictions that the model makes about observable quantities. This can be achieved by sampling parameter values from the posterior distribution and calculating model predictions from these samples to see how they vary. A specified model can be judged not only through its predictive power of the "best" set of model parameters but also through the certainty of its predictions given the uncertainty of the parameter values. The posterior distribution represents the uncertainty in our knowledge regarding the true values of the rate constants.

S2 Sampling from the Posterior Distribution of Ion-channel models

Computing posterior probability distributions, as opposed to maximum likelihood estimation, often faces significant statistical challenges as the posterior distributions of interest are often known pointwise up to a normalising constant. Many challenges in Bayesian statistics, for example the estimation of the normalising constant for model evidence, arise from the need to calculate expectations with respect to probability distributions whose analytic form may not be known.

Consider an arbitrary probability distribution $g(\theta)$ from which we would want to estimate an expectation of the form $\mathbb{E}_{g(\theta)}(f(\theta)) = \int f(x)g(x)dx$. If we do not know the analytical distribution of $g(\theta)$ but are able to draw realisations from $g(\theta)$, we can calculate the function $f(\theta)$ and hence the integral can be numerically estimated. In such instances we can use the weak law of large numbers, that is, for any ϵ,

$$P(\mid \bar{f}_n - \mu_f \mid < \epsilon) \to 1 \quad as \quad n \to \infty \tag{2}$$

where μ_f is the true expectation of $f(\theta)$ and $\bar{f}_n$ is the estimator for the expectation after the nth sample. An ideal estimator for $\bar{f}_n$ is the Monte Carlo integrator $\bar{f}_n = \frac{1}{n}\sum_{i=1}^{n} f(\theta_i)$. This requires the ability to draw independent and identically distributed samples from $g(\theta)$. In practice, it is very difficult to draw such samples but the Monte Carlo estimator remains the ideal estimator as the rate of convergence of the estimator scales favourably with the dimensionality of the sampled probability distribution.

Fortunately, Markov chain Monte Carlo methods are an invaluable tool that can be used to perform sampling in this context. These methods employ a probabilistic process whose stationary distribution is our target probability distribution and hence allows us to draw samples from it. The most widely known algorithm for such a process is the Metropolis-Hastings algorithm [1, 2].

MCMC methods typically require significantly more computational power than optimisation methods for Maximum Likelihood inference. This means that the efficient design of these sampling algorithms is of crucial importance particularly as the dimension of models increase. Although the converged Markov chain defined by the Metropolis-Hastings algorithm will draw samples from the exact target distribution, they will not be uncorrelated as the next proposed point depends on the current position for the chain. The presence of autocorrelation within the samples drawn increases the variance of the Monte Carlo estimator. We can assess the impact of autocorrelation on our sampling by considering the Effective Sample Size (ESS) of our samples. This is the equivalent number of independent samples given the autocorrelation present in the samples. The ESS for a given parameter θ is given by:

$$ESS(\theta) = \frac{N}{(1 + 2\sum_{l=1}^{L} \gamma(l))} \tag{3}$$

where $\sum_{l=1}^{L} \gamma(l)$ is the sum of the L significant autocorrelation lag coefficients of the converged chain and N is the number of samples drawn. A lag is determined to be significant if its coefficient is different from zero. The total number of significant lags can be established by visually examining the autocorrelation plot or with by calculating a confidence interval for the autocorrelation sequence of a white noise process with which to compare with the empirical coefficients.

The auxiliary proposal distribution $Q(. \mid x)$ is often manipulated to speed up the convergence of the chain and reduce the level of autocorrelation in collected samples. Such schemes often use information that is available at the current point in the parameter space. These schemes often integrate gradient

information about the log-target density e.g. HMC[3] or MALA [4] or utilise Riemannian geometry [5] in order to make more informed proposals. These schemes work well when analytic derivatives of the log-likelihood are available to use, or at least can be approximated in a computationally timely fashion. Unfortunately the missed events correction in Equation 3 (main text) does not have analytical expressions for the derivatives and the computational expense of evaluating derivative information numerically outweighs the gain in sampling efficiency for these models.

S3 Algorithms

Algorithm S1 Multiplicative Metropolis-within-Gibbs algorithm [6] with scaling during burn-in

$i = 0$
$burnin = N/2$ ▷ Set number of burn-in samples
$AC(1:K) = 0$ ▷ Number of acceptances over each adjust period
$adjust(1:K) = X$ ▷ Consider adjusting SF every X samples during burn-in
$SF(1:K) = 1$ ▷ Initialise scale factor for proposal covariance
$\theta_i = \theta$
$\Sigma = \mathbf{I}$
while $i < N$ **do**
  for each k in K **do**
    $q = \mathcal{N}(\theta(k)_i, SF(k) * \Sigma(k,k))$
    $Y \sim q(\theta \mid \theta(k)_i)$
    $\theta(k)^* = \theta(k)e^Y$
    $\alpha = min\left(1, \frac{p(\theta(k)^*)q(\theta(k)_i|\theta(k)^*)[\prod_{j\neq k}^{K}\theta(j)]\theta(k)^*}{p(\theta(k))q(\theta(k)^*|\theta(k)_i)\prod_{j}^{K}(\theta(j)_i)}\right)$
    if $u \sim U[0,1] < \alpha$ **then**
      $\theta(k)_{i+1} = \theta(k)^*$
      $AC = AC + 1$
    else
      $\theta(k)_{i+1} = \theta(k)_i$
    end if
    if $mod(i, adjust(k)) == 0$ and $i <= burnin$ **then**
      if $AC/adjust(k) < 0.1$ **then** ▷ reduce the step size
        $SF(k) = SF(k) * 0.9$
      else if $AC/adjust(k) > 0.5$ **then** ▷ increase the step size
        $SF(k) = SF(k) * 1.1$
      end if
      $AC(k) = 0$
    end if
  end for
  $i = i + 1$
end while

In Algorithm S1, an iteration of the MCMC sampler proceeds as follows. At the ith step for parameter k, the proposal distribution q is constructed as a univariate Gaussian distribution with mean $\theta(k)$, the current parameter

value, and standard deviation $\Sigma(k,k)$ where $\Sigma(k,k)$ is the diagonal element of the covariance matrix Σ scaled by the individual scaling factor for parameter k.

A parameter value for $\theta(k)^*$ is then proposed by exponentiating a sample drawn from the proposal distribution and multiplying it by the current parameter value $\theta(k)_i$. This proposal is accepted with probability α according to the Metropolis-Hastings acceptance ratio which has been adjusted to account for the fact that proposals are being made in log-space. If the move is accepted, the value of the $i+1$th iteration of parameter k is set to the proposed parameter $\theta(k)^*$ and the acceptance counter for parameter k, $AC(k)$ is iterated. If the move is not accepted, the value of the kth parameter set to the current value. At this point, if we are still in the burnin phase and the current iteration is a multiple of the adjustment period (denoted by the modulus of i and $adjust(k)$), then a decision is made as to whether to adjust the scale factor for the proposal distribution for parameter k. Briefly, if the proportion of acceptances in the last X samples is less than 0.1 then the scale factor for parameter k is reduced and if it is greater than 0.5 the scale factor is increased.

Regardless of whether an adjustment is made, the acceptance counter is reset to 0 for the next set of X samples. If we are not in the burn-in phase or the current iteration is not a multiple of the adjustment period then the algorithm updates to the $i+1$th iteration and the next sample is drawn. The process continues until N iterations have been performed.

Algorithm S2 Adaptive algorithm of [7]

$i = 0$
$\theta = \theta_i$
while $i < N$ **do**
  if $i <= 2K$ **then**
    $q = N(\theta, \frac{(0.1^2)}{K}\mathbf{I})$ ▷ where K is number of params
  else
    $q = (1-\beta)N(\theta, \frac{(2.38^2)}{K}\hat{\mathbf{\Sigma}} + \beta N(\theta, \frac{(0.1^2)}{K}\mathbf{I})$ ▷ where β is the mixture parameter
  end if
  $\theta^* \sim q(\theta \mid \theta_i)$
  $\alpha = min\left(1, \frac{p(\theta^*)q(\theta_i|\theta^*)}{p(\theta)q(\theta^*|\theta_i)}\right)$
  if $u \sim U[0,1] < \alpha$ **then**
    $\theta_{i+1} = \theta^*$
  else
    $\theta_{i+1} = \theta_i$
  end if
  $\hat{\mathbf{\Sigma}} = cov(\theta(.))$ ▷ Update the covariance matrix with all the samples
  $i = i + 1$
end while

In Algorithm S2, an iteration of the MCMC algorithm proceeds as follows. If the current iteration i is less that $2K$ where K is the number of parameters, then the proposal distribution q is set to a multivariate Gaussian distribution with mean vector of θ, the current set of parameter values, and a scaled identity matrix for the covariance matrix. This is to ensure that the process proposes moves with a small step size at the start, in order to build up the initial estimation of the sample covariance. If the current iteration is greater than $2K$ then the proposal distribution is made up of a multivariate Gaussian mixture distribution composed of an estimate of the current covariance distribution $\hat{\mathbf{\Sigma}}$ and an uncorrelated Gaussian. The fraction of each is decided by the β parameter which is fixed at $\beta = 0.05$ as per [7]. A vector of parameters θ is then proposed from this distribution, and the vector is accepted according to the probability derived from the Metropolis-Hastings ratio. The final step of the iteration is to update the sample covariance $\hat{\mathbf{\Sigma}}$ with all of the samples derived so far. The process terminates after N samples have been drawn. Similarly to Algorithm S1, a burn-in phase was also incorporated where a global scaling factor is used to increase or decrease the proposal step size according to the global parameter acceptance rate. This has been omitted from Algorithm S2 for clarity.

References

[1] Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

[2] Hastings, W. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[3] Neal, R. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, pages 113–162. CRC Press, 2011.

[4] Roberts, G. and R. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363, 1996.

[5] Girolami, M. and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 73(2):123–214, 2011.

[6] Sherlock, C., P. Fearnhead and G. Roberts. The random walk metropolis: Linking theory and practice through a case study. *Statist. Sci.*, pages 172–190, 2010.

[7] Roberts, G. and J. Rosenthal. Examples of adaptive MCMC. *J. Comp. Graph. Stat.*, 18(2):349–367, 2009.