A very brief introduction to computational Bayesian analysis

Zach Gompert and Alex Buerkle (last updated April 2016)

Much of modern biology is based on formal analyses of data. Consequently, in addition to learning the particulars of different forms of empiricism, there is a substantial need to understand approaches to analysis. This is not new and students of biology have utilized and developed statistics since the beginnings of these disciplines. Nevertheless, new forms of data and conceptual advances in thinking about proper analysis call for us to move well beyond traditional statistical approaches for analysis. One area of innovation includes Bayesian analysis and this document is meant as an introduction to this area.

There are many books devoted to Bayesian analysis. Our objective in introducing this subject ourselves is to provide an accurate, useful, and clear introduction, while avoiding some of the unnecessarily distracting issues that arise in some discussions of Bayesian analysis. The logic of this form of analysis is straightforward and we want to communicate this simple logic in our introduction.

1 Parameter estimation

A surprisingly under-appreciated fact is that much of analysis in biology is concerned with parameter estimation. We estimate population means, the recombination rate along a chromosome, migration rates between populations, the frequency of genotypes among diseased and healthy organisms, and so forth. Typical statistics have a long history as a basis for testing simple hypotheses about these parameters (what we think of as typical statistics are often referred to as *frequentist* statistics). For example, we are often interested in the equality of means in two populations, and use a t-test to determine the probability of observing the data, given a simple, single null hypothesis, namely that the difference in means (a parameter) is zero ($P(D|H_0)$, where D stands for the data and $H_0 : \bar{x}_1 - \bar{x}_2 = 0$). In a typical analysis, an estimate of the difference in means (the parameter of interest) is often not even reported, so perhaps it should not be surprising that many biologists are unaware that they are estimating parameters. Instead, researchers often only report $P(D|H_0)$ in the form of the p-value associated with the test statistic (t in this case). In reality, we are rarely interested in the null hypothesis itself ($H_0 : \bar{x}_1 - \bar{x}_2 = 0$), but instead we often would like an estimate of the magnitude of the difference $\bar{x}_1 - \bar{x}_2$ and whether the data suggest that true differences between populations populations are not neglible (including zero).

Formally, it would beneficial to have a method to calculate $P(H_i|D)$ for a full range of hypotheses or parameter values. In the example above, it would be an improvement to calculate the probability of all differences of means (over some reasonable interval). In the case of likelihood analysis, one calculates the likelihood of parameter estimates, given the data: $L(H_i|D) = cP(D|H_i)$ or after dropping the constant c, $L(H_i|D) \propto P(D|H_i)$. For many analysis problems, likelihood is a complete and satisfactory solution to parameter estimation. However, in many instances we seek estimates of $P(H_i|D)$ that are true probabilities and can be used accordingly and are based on better models of uncertainty, including more complex models that incorporate more unknowns and a hierarchy of parameters. For this, typically Bayesian inference will be the preferred solution.

2 Why use Bayesian inference?

Given the utility of hypothesis testing and parameter estimation in the context of frequentist statistics, and the existence of likelihood methods for parameter estimation and model comparison, it is natural to wonder about the need for something else. Skepticism is particularly warranted given the zealotry of some practitioners of Bayesian inference and the possibility that all this Bayesian business is a bandwagon. Reluctance about Bayesian inference is also certainly warranted because it is likely unfamiliar and implementation of the analyses can be non-trivial. One argument for Bayesian inference is that it is useful and makes better use of the data biologists work very hard to get. In particular, Bayesian methods allow us to properly model uncertainty (Section 2.2) and provide a coherent framework for model comparison (Section 2.3). Bayesian methods also result in true probabilities of parameters of interest, whereas the others do not. Other arguments are more philosophical and arcane, and often drive biologists away from learning about Bayesian approaches (e.g., arguments about the use of "prior" information). We will leave these aside and will focus first on what you can do with Bayesian methods.

2.1 A minimal Bayesian lexicon

In the context of Bayesian parameter estimation or model comparison, typically we will refer to the probability distribution for a parameter θ (or parameters, a vector of θs) or model $\mathcal{M}$, rather than for alternative hypotheses H_i (the notation used in Section 1). In Bayesian analysis, we wish to calculate the probability distribution $P(\theta|D)$; that is, the probability of each value of θ given the observed data. This conditional probability ($P(\theta|D)$) is referred to as the *posterior probability distribution* for θ.

In its simplest form[1] the posterior probability is: $P(\theta|D) \propto P(D|\theta)P(\theta)$. The posterior probability in this case consists of two terms. Somewhat confusingly, the first term ($P(D|\theta)$) is referred to as the *likelihood* (but note from the discussion of likelihood above that $L(H_i|D) = cP(D|H_i)$, so the label makes some sense). The second term $P(\theta)$ is the *prior distribution* (or simply *prior*) for the parameter of interest, θ (e.g., a migration rate, an assignment probability, a population mean, etc.). The choice of prior distribution is an important and powerful aspect of Bayesian inference. Unfortunately, the use of priors at all has been a point of distracting contention among those with a philosophical bent. We have chosen to focus on the utility of a prior rather than trying to recount the philosophical debates (most books on Bayesian data analysis cover this thoroughly), which we consider to be resolved to our satisfaction. Note that by incorporating the prior, the result of Bayesian analysis is a probability distribution for *theta* and it is the only analytical approach that results in this desired probability.

[1] In this case we are disregarding the denominator, $P(D)$, that appears in the full formula for Bayes' Theorem $P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$, because very often the denominator does not come into play.

2.2 Modeling uncertainty

2.2.1 Specification and parameterization of unknowns

In Bayesian analysis all unknowns, including missing data, parameters, and covariates, are modeled as random variables using probability theory. This means that it is possible to model uncertainty in all unknowns, and that a probability distribution for each unknown can be selected that is consistent with the processes or data associated with that unknown (i.e., some unknowns may be described well by Normal distributions, whereas others might be better described by Gamma, Beta, or binomial distributions). It is also possible to combine uncertainty for multiple parameters into a coherent analysis. For example, genetic mapping in natural populations requires estimates of genetic ancestry and kinship, which are then used as covariates when estimating regression coefficients describing the relationship between a genotype and phenotype of interest. In a classic statistical framework, point estimates of ancestry and kinship proportions would be obtained first and then used for estimating regression coefficients, which would ignore the uncertainty associated with the covariates and potentially lead to poor inference. In a Bayesian context it is possible to propogate uncertainty in covariates, while estimating regression coefficients that reflect uncertainty in all other parameters of the analysis.

2.2.2 Complex and hierarchical models

Many processes, entities, and data in biology have natural hierarchical structure. For examples, species are composed of populations, which are composed of individuals that possess genomes that are composed of chromosomes and with allelic variation at genetic loci. Bayesian analysis provides a coherent framework for estimating parameters at each of these hierarchical levels while accounting for uncertainty at each level. Similarly, studies of quantitative genetics often have a hierarchy (families, sibships, replicates, treatments) that could be modeled appropriately in a Bayesian framework, rather than under constraints associated with typical analyses of variance (e.g., the requirement of equal sample size without missing data).

2.2.3 Credible intervals

Bayesian credible intervals summarize posterior uncertainty in all parameters and other unknowns. Credible intervals are easily calculated from posterior distributions and have a clear interpretation. For example a 95% credible interval can be interpreted as having a 95% chance of containing the parameter's true value.

This is quite different than a 95% confidence interval, which is interpreted as an interval constructed by a method that generates intervals containing the true parameter value at least 95% of the time (i.e., the confidence is associated with the method not the actual interval and there is no probability associated with whether the interval actually contains the parameter).

2.3 Model comparison

2.3.1 Bayes factors

The ratio of the marginal likelihood of two models ($\frac{P(\theta_1|P(\text{Data}))}{P(\theta_2|P(\text{Data}))}$) is referred to as a Bayes factor (it is analogous to a likelihood ratio in likelihood analysis). Bayes factors provide a simple and clear framework for comparing models, including models with different parameters constrained to different values or models with entirely different parameters. Unlike likelihood ratio tests, Bayes factors do not require that nested models are compared.

2.3.2 Models as unknown parameters

Recent Bayesian techniques provide a means for treating different models made up of different parameters as a parameter and seek to estimate the posterior probability of each of these alternative models. Thus, models with different parameters can be treated in the same manner as different values of a parameter (e.g., Zhou *et al.*, 2013), in that the posterior probability of alternative models with different numbers of parameters (a discrete distribution) can be quantified.

3 A simple example of computation with MCMC

In practice, typically we utilize a computer to gather samples from the posterior distribution by simulation, rather than calculating the posterior distribution directly and analytically. Whereas calculating the posterior analytically is interesting and there are many remarkable aspects of this approach, we are simply going to skip it here ($>$ 200 pages in at least one standard book on the subject), simply because the opportunity to use an analytical solution is seemingly rare and we want to present first what is likely to be useful. Secondly we focus on using simulations to sample from the posterior because it is a very flexible tool that can be adapted to anything from simple to very complex models.

In practice, a computational Bayesian analysis can performed easily with JAGS (Plummer, 2003, or other BUGS implementations) and can be done from an interface to JAGS from R (the `rjags` library). What follows is an explanation of how computational Bayesian analysis does it work; this will be hidden by use of JAGS and similar software.

Recall that the posterior distribution of the parameters of interest, given the data is given by: $P(\theta|D) \propto P(D|\theta)P(\theta)$. To utilize simulation to characterize the posterior distribution, we need first to propose a candidate value of θ, which we will designate as θ_{cand}. Then we calculate the prior probability of θ_{cand} (the $P(\theta)$ term above) and the likelihood of the data given θ_{cand} ($P(D|\theta_{cand})$; their joint probability is proportional to the posterior distribution ($P(\theta_{cand}|D)$).

The term used to refer to simulations and algorithms to draw samples from the posterior distribution is *Markov Chain Monte Carlo*. Stochastic sampling occurs in the choice of a proposed θ_{cand} and in the algorithm for determining whether the proposed value will be accepted into the chain (e.g., the Metropolis-Hastings method).

Perhaps the best way to introduce these methods is by way of a simple example that describes the computational steps. From there we can work towards generalization and recognition of the method's components.

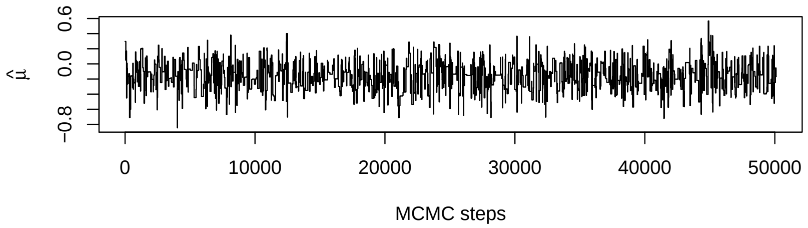

Figure 1: History plot for a Markov Chain for estimates of the mean (μ) of the data.

3.1 Computational example

In the following computational example, which is written in R, we have a data set containing 30 values. This example is for teaching and learning purposes and in practice we would likely use the R interface to JAGS (the `rjags` library) as the computational tool to get MCMC samples from the posterior. The objective is to estimate the posterior distribution of the mean of these data. We can summarize the posterior distribution of the mean through various statistics, including the distribution's mean, as well as quantiles of interest (e.g., 2.5% and 97.5% for the 95% equal tail credible interval for the mean). For illustration, we compare these estimates to the sample mean and the standard 95% confidence interval for the mean at $\bar{x} \pm 1.96\frac{s}{\sqrt{n-1}}$ (typically referred to as the standard error, it is the standard deviation of the estimates of the mean).

The example utilizes the Metropolis-Hastings algorithm to determine whether proposed new values are incorporated into the chain. Plots of the posterior distribution and the full Markov Chain are in Figs. 1 and 2.

```
## 30 data points that happen to come from a standard normal distribution (mean = 0, sd = 1)

observed.data<-c(1.61546274, -0.86954608, -0.20670916, -1.73755818, ...
                 1.27636970, -0.19511416)

## Number of steps for MCMC
mcmcL<-50100

## Vectors to save estimate of mu and P(D|M)
mu <- numeric(mcmcL)
pr.DataModel <- numeric(mcmcL)

## random seed for mu from a uniform between -10 and 10; we could use almost anything here
mu[1] <- runif(1,-10,10)

## initial probability: first term = likelihood, second term = prior,
## we assume a normal prior on mu with mean 0 and sd 1000, which is
## an uniformative prior
pr.DataModel[1] <- sum(dnorm(observed.data, mu[1], 1, log=T) + dnorm(mu[1], 0, 1000, log=T))
```

```
## Enter MCMC
for(i in 2:mcmcL){
  ## proposal of candidate value for mu, we will use a normal with mean = 0, and sd = 10.
  ## Note that this proposal is independent of the value in previous
  ## MCMC step, so this is an "independence chain", as opposed to a random walk
  mu[i] <- rnorm(1, 0, 10)


  ## calculate probability of the data given the candidate value, mu[i]
  pr.M_cand<-sum(dnorm(observed.data, mu[i], 1, log=T) + dnorm(mu[i], 0, 1000, log=T))
  ## grab the probability of the data given mu[i-1] (the previous value in the chain)
  pr.M_old <- pr.DataModel[i-1]

  ## calculate probablity of drawing the mu[i] and mu[i-1] from the
  ## candidate generating or jump function
  pr.sam_cand <- dnorm(mu[i], 0, 10, log=T)
  pr.sam_old <- dnorm(mu[i-1], 0, 10, log=T)

  ## calculate Metropolis Hastings ratio
  ## (use a difference in logs rather than ratio to maintain numerical precision)
  mh <- (pr.M_cand - pr.sam_cand) - (pr.M_old - pr.sam_old)

  ## sample U=log(runif(1)) from random uniform (with defaults min=0, and max=1) and
  ## accept or reject proposed value according to comparison to mh ratio
  if( log(runif(1)) < mh){    ## accept candidate value
    pr.DataModel[i]<-pr.M_cand
  }
  else{ ## reject candidate value
    mu[i]<-mu[i-1]
    pr.DataModel[i]<-pr.M_old
  }
}
```

3.2 Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm is a tool for sampling from a proposal distribution and deciding whether proposed, or candidate value is accepted into the chain. There are four steps in the algorithm and this algorithm is implemented in our example.

Step 1 Generate a proposed value of the parameter of interest, θ_{cand} by sampling from a proposal distribution ($J(\theta)$, `rnorm(1, 0, 10)` in our example, at the top of loop for the Markov Chain; also referred to as a candidate generating or jump function).

Step 2 Calculate

$$\text{Metropolis Hastings ratio } (r) = \frac{\frac{P(\theta_{cand})}{J(\theta_{cand}|\theta_{t-1})}}{\frac{P(\theta_{t-1})}{J(\theta_{t-1}|\theta_{cand})}}$$

where the $J(\theta|\theta_*)$ terms might simply be $J(\theta)$ and not be conditional on the previous step in the chain (as in our example).

Step 3 Sample a random value (U) from a uniform distribution on the interval $(0, 1)$.

Step 4 If $U < r$, then $\theta_t = \theta_{cand}$, otherwise $\theta_t = \theta_{t-1}$. That is, accept the proposed value with probability given by the Metropolis Hastings ratio (note the ratio can be greater than one).

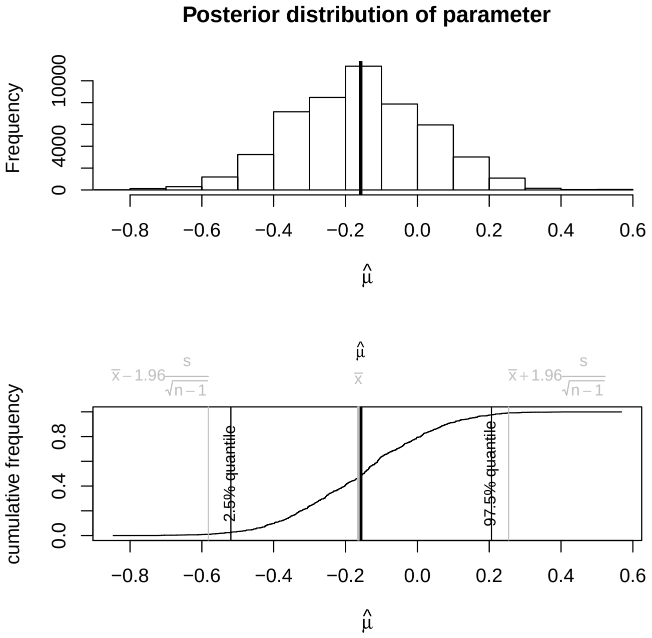

Figure 2: Posterior probability distributions for our parameter of interest, the mean (μ) of the data. Note that in this case the credible interval includes the true value, but the sampling has not lead to an estimate where the mean is very close to the true value. More or different (random walk) sampling might do so.

If we use the Metropolis-Hastings Algorithm or other, related MCMC methods, theory shows that the samples will converge on the posterior distribution for the parameters of interest. We might need to discard initial samples as a "burn-in" of the model, before the sampling has converged to the stationary, posterior distribution. Furthermore, samples in the chain should be monitored for mixing to insure an adequate number of independent samples to characterize the distribution. Multiple, independent chains should be run to ensure their convergence on the same posterior distribution. For many problems JAGS can determine the proper algorithms to use for a particular model and the details of updates are hidden to the user (although proper burn-in, mixing and convergence need to be evaluated by the user). Regardless, it is useful to know the mechanics of how MCMC samples are obtained.

References

Plummer M (2003) JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling.

Zhou X, Carbonetto P, Stephens M (2013) Polygenic modeling with Bayesian sparse linear mixed models. *PLoS Genetics*, **9**, e1003264.