

Ecological modeling, phylogeography and R tools to analyze biological processes and patterns

Ecological modeling, phylogeography and R tools to analyze biological processes and patterns

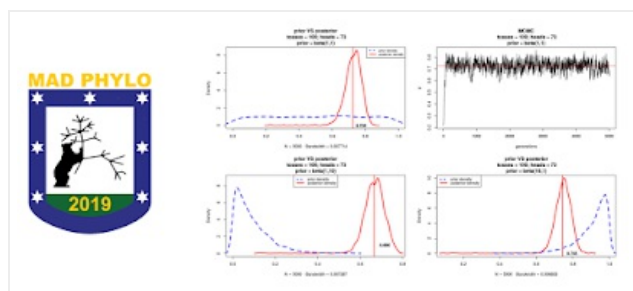
main my research

domingo, 14 de abril de 2019

Understanding Bayesian Inference with a simple example in R!

Hi there!

Last summer, the Royal Botanical Garden (Madrid, Spain) hosted the first edition of MadPhylo, a workshop about Bayesian Inference in phylogeny using RevBayes. It was a pleasure for me to be part of the organization staff with John Huelsenbeck, Brian Moore, Sebastian Hoena, Mike May, Isabel Sanmartin and Tamara Villaverde. Next edition of MadPhylo will be held June 10, 2019 to June 19, 2019 at the Real Jardín Botánico de Madrid! If you are interested in Bayesian Inference and phylogeny just can't miss it! You'll learn the RevBayes language, a programming language to perform phylogeny (and other) analyses under a Bayesian framework! Apply [here](#)!



The first days were focused to explain how we can use the Bayesian framework to estimate the parameters of a model. I'm not an expert in Bayesian Inference at all, but in this post I'll try to reproduce one of the first Madphylo [tutorials](#) in R language. As a simple example, we'll use a coin flipping experiment. We can model this experiment with a binomial distribution:

$\text{binomial}(n, p)$

where n is the number of tosses and the parameter p is the probability to obtain a head. Given the number of tosses and the number of heads (h), we can use Bayesian Inference to compute the probability to obtain a head (p).

If we use Bayes' theorem, we have that the probability of a specific value of p given the number of heads (h) is:

$$Pr(p|h) = \frac{Pr(h|p) \times Pr(p)}{Pr(h)}$$

where $Pr(h|p)$ is the likelihood of the data given a value of p , $Pr(p)$ is the prior probability for p and $Pr(h)$ is the marginal likelihood for the data. The prior probability of our parameter represent our believe about what values can take the parameter. If we think our parameter can take whatever value with the same probability, we use an uniform (flat) prior.

We can use a Markov Chain Monte Carlo (MCMC) to introduce many different values of p and get the posterior probability of these values. To do this, we can use the Metropolis-Hastings MCMC algorithm, with the next steps (simplified):

in Twitter!

Follow @javi_ferlop

 Javier Fernández-López

Ver todo mi perfil

downloads

- [niceOverPlot code](#)
- [Podarcis dataset 1](#)
- [rWind demo code](#)
- [rWind reference manual](#)
- [nichEvolve Netlogo model](#)

topics

- 2016 (5)
- 2017 (3)
- 2018 (5)
- ▼ 2019 (1)
- ▼ 04 (1)

Understanding Bayesian Inference with a simple exa...

about me...

PhD student (Predoctoral Grant from the Spanish Ministry of Economy and Competitiveness) at Royal Botanic Garden (Spanish National Research Council, CSIC) and member of the Evolution and Conservation Biology Research Group of Complutense University of Madrid (UCM)

e-mail:
jflopez.bio@gmail.com

views...

- Step 1) Set an initial value for p .

```
p <- runif(1, 0, 1)
```

- Step 2) Propose a new value of p , called p' .

```
p_prime <- p + runif(1, -0.05, 0.05)
```

- Step 3) Compute the acceptance probability of this new value for the parameter. We have to check if the new value improves the posterior probability given our data. This can be seen as the ratio: $\Pr(p'|h) / \Pr(p|h)$.

$$R = \min \left(1, \frac{\Pr(p'|h) = \frac{\Pr(h|p') \times \Pr(p')}{\Pr(h)}}{\Pr(p|h) = \frac{\Pr(h|p) \times \Pr(p)}{\Pr(h)}} \right)$$

It can be simplify as:

$$R = \min \left(1, \frac{\Pr(h|p') \times \Pr(p')}{\Pr(h|p) \times \Pr(p)} \right)$$

The advantage of this method is that we avoid to compute the marginal likelihood, that is often difficult to obtain with more complex models. Let's stop here a little bit to explain each term of this equation.

Since our main model is a binomial model (coin toss), the likelihood function $\Pr(h|p)$ can be defined in R language as:

```
# help(dbinom)
likelihood <- function(h, n, p){
  lh <- dbinom(h, n, p)
  lh
}
```

$\Pr(p)$ is our prior probability, that is, our believe about what values can take p . The only thing that we know is that it must be a value between 0 and 1, since it is a probability. If we think that all values have the same probability, we can define a flat prior using the *beta* distribution. A *beta* distribution with parameters $\beta(1,1)$ is a flat distribution between 0 and 1 (you can learn more about *beta* distribution [here](#)). Then, in R we can obtain $\Pr(p)$ under a $\beta(1,1)$ as:

```
# help(dbeta)
dbeta(p, 1, 1)
```

Now, the acceptance probability (R , see equations in Step 3) will be the minimum value: 1 or the ratio of posterior probabilities given the different p . We express this equation in R language as follows:

```
R <- likelihood(h,n,p_prime)/likelihood(h,n,p) * (dbeta(p_prime,1,1)/dbeta(p,1,1))
```

- Step 4) Next, we generate a uniform random number between 0 and 1. If this number is < R , we will accept the new value for p (p') and we update the value of $p = p'$. If not, the change is rejected.

```
if (R > 1) {R <- 1}
random <- runif (1, 0, 1)
if (random < R) {
  p <- p_prime
}
```

- Step 5) Now we record the current value of p .

```
posterior[i,1] <- log(likelihood(h, n, p))
posterior[i,2] <- p
```

Finally, we should repeat this loop many times to obtain a good estimate of p . This can be easily done in R using a *For* loop (check the full code below).

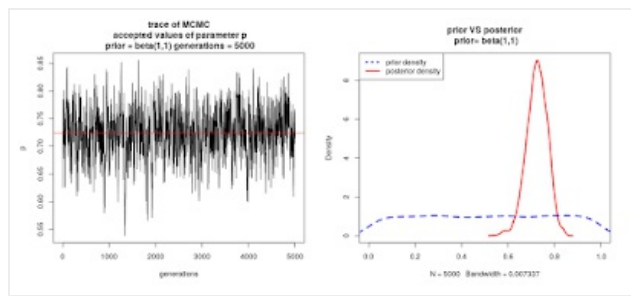
Following the [example](#) studied in MadPhylo with RevBayes, I wrote some code to reproduce

this example with R. We toss a coin 100 times, and we obtain 73 heads. Is my coin biased? Let's check what is the probability to obtain a head given the data:

```
# Set the number of tosses.
n <- 100
# Set the number of heads obtained.
h <- 73
# Define our likelihood function.
# Since our model is a binomial model, we can use:
likelihood <- function(h,n,p){
  lh <- dbinom(h,n,p)
  lh
}
# Set the starting value of p
p <- runif(1,0,1)
# Create an empty data.frame to store the accepted p values for each iteration.
# Remember: "the posterior probability is just an updated version of the prior"
posterior <- data.frame()
# Set the length of the loop (Markov Chain, number of iterations).
nrep <- 5000
# Start the loop (MCMC)
for (i in 1:nrep) {
  # Obtain a new proposal value for p
  p_prime <- p + runif(1, -0.05,0.05)
  # Avoid values out of the range 0 - 1
  if (p_prime < 0) {p_prime <- abs(p_prime)}
  if (p_prime > 1) {p_prime <- 2 - p_prime}
  # Compute the acceptance probability using our likelihood function and the
  # beta(1,1) distribution as our prior probability.
  R <- likelihood(h,n,p_prime)/likelihood(h,n,p) * (dbeta(p_prime,1,1)/dbeta(p,1,1))
  # Accept or reject the new value of p
  if (R > 1) {R <- 1}
  random <- runif(1,0,1)
  if (random < R) {
    p <- p_prime
  }
  # Store the likelihood of the accepted p and its value
  posterior[i,1] <- log(likelihood(h, n, p))
  posterior[i,2] <- p
  print(i)
}
```

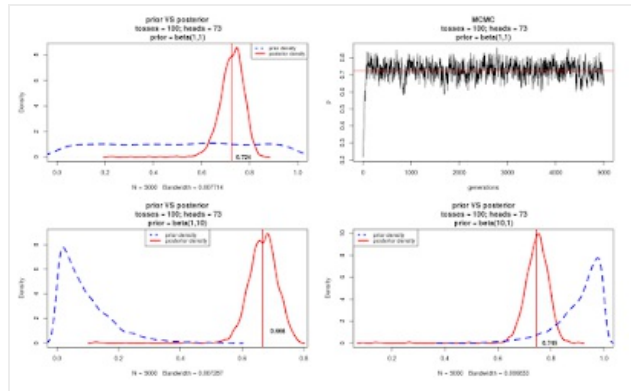
Let's plot some results...

```
par(mfrow= c(1,2))
prior <- rbeta(5000, 1,1)
plot(1:5000 ,posterior$V2, cex=0, xlab = "generations", ylab = "p",
     main = "trace of MCMC\n accepted values of parameter p\n prior = beta(1,1) generations = 5000")
lines(1:5000, posterior$V2, cex=0)
abline(h=mean(posterior$V2), col="red")
plot(density(posterior$V2), xlim = c(min(min(prior),min((posterior$V2))), max(max(prior),max((posterior$
V2)))),
     ylim = c(0, max(max(density(prior)$y),max((density(posterior$V2)$y)))), main= "prior VS posterior\n pr
ior= beta(1,1)",
     lwd=3, col="red")
lines(density(prior), lwd=3, lty=2, col="blue")
legend("topleft", legend=c("prior density", "posterior density"),
     col=c("blue", "red"), lty=c(3,1), lwd=c(3,3), cex = 1)
```



Well, we can see that the probability to obtain a head given our data is around 0.7, so our coin must be a fake!

You can play with the code and explore with a different number of tosses, or the effect of a different prior for p ...



If you want to learn more about Bayesian Inference, I recommend you these YouTube [videos](#) by Rasmus Bååth, or this wonderful [book/course](#) by Richard McElreath

That's all!

Enjoy!

Publicado por [Javier Fernández-López](#) en 14:36:00



No hay comentarios:

Publicar un comentario

Introduce tu comentario...



Comentar como:

Cuenta de Goog

Publicar

Vista previa

[Página principal](#)

[Entrada antigua](#)

Suscribirse a: [Enviar comentarios \(Atom\)](#)