Bayesian R for Project

Ray Stokes

2025-09-23

Equations

As a reminder, the main Bayesian equation for this project is:

$$P(S|ID, AQ, RD, AN) = \frac{P(AQ|S)P(AN|S)P(ID|S)P(RD|S)P(S)}{P(AQ)P(AN)P(ID)P(RD)}$$

Where:
S is the species
ID is the percentage ID
RD is the read count
AN is the assay name
AQ is the aquamaps probability (probability at that location, doesn't have to be aquamaps. We might be able to rename variable to L if it is clearer)

If we include the genus G, this becomes:

$$P(S|G, ID, AQ, RD, AN) = \frac{P(G|AQ, S)P(AQ|S)P(AN|S)P(ID|S)P(RD|S)P(S)}{P(G|AQ)P(AQ)P(AN)P(ID)P(RD)}$$

I believe $P(G|AQ, S) = 1$, correct me if I am wrong. If you know the species, you know the genus

If we don't care about the terms not including S, or G, then we have:

$$f(S|G, ID, AQ, RD, AN) \propto \frac{f(AQ|S) \times f(AN|S) \times f(ID|S) \times f(RD|S) \times f(S)}{f(G|AQ)}$$

Meaning of the terms:

$f(S)$ is the proportion of that particular species. Can be either from Phillips list, a database, or all databases (need to choose)
$f(AQ|S)$ probability of finding at that location given that we know the species. Would depend on closeness of species in database to that location, and the amount. For example, what proportion of points (where the species was observed) are closer than 10km? Can experiment with different thresholds, e.g 5km, 10km, 20km.
Or, if just using 1 location, can something like a function of 1/distance, e.g logit (still need to choose a threshold)
$f(ID|S)$ is the precision (TP/(TP+FP)), multiplied by % ID. So if 100% ID, it is just the precision
$f(RD|S)$ is the number of read counts, perhaps divided by the maximum number of read counts, to make it a proportion
$f(AN|S)$ is the value we assign for the assay name, closer to 1 if more trustworthy
$f(G|AQ)$ is the proportion of that genus at the location (area of interest, compared to other genus in the area)
Also need to add something about the missing genus/species in the databases (how well the genus is covered in databases

We can start by trying to model each as a normal distribution with sample proportions with mean p and standard deviation $\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$. We can update these later as we come up with better distributions for each variable. If we want to work with counts directly, we can view each sighting of the species in the database a 'success' that is 'evidence' we can use for Bayesian inference, e.g binomial, geometric, poisson, etc.

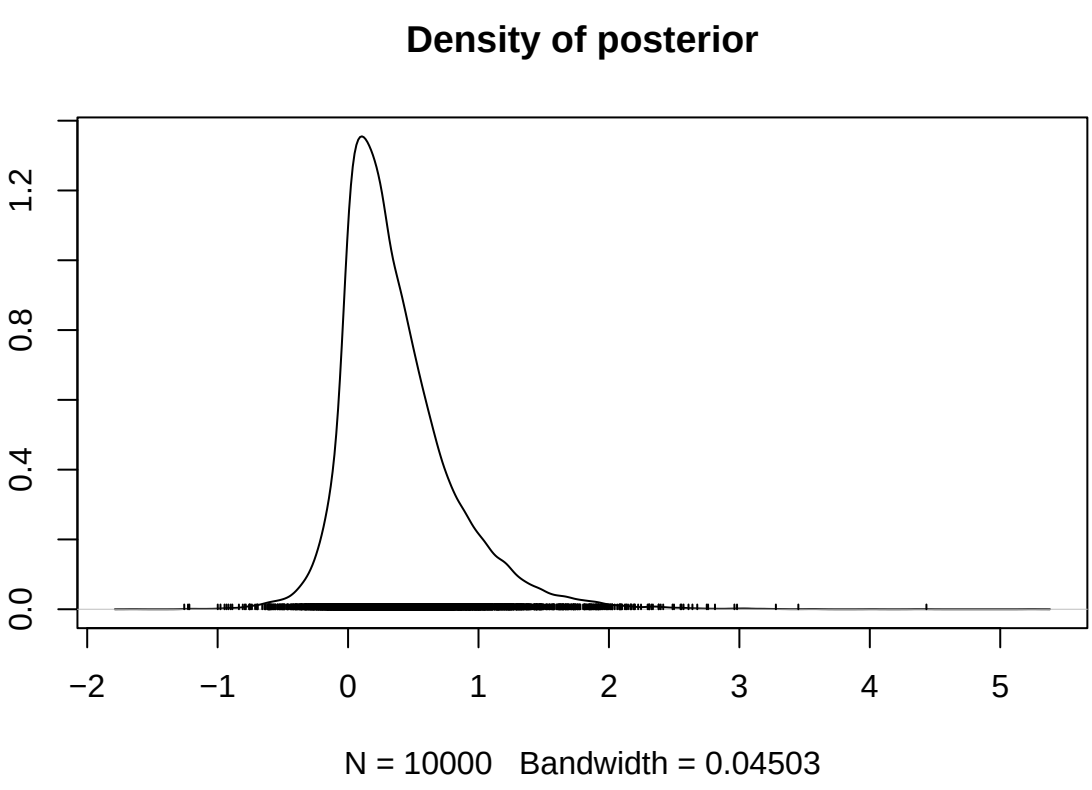
Main function for Open BUGS:

```
eDNA_OpenBUGS <- function(p_species, n_species, p_reads, sigma_reads, p_ID, sigma_ID, p_location, sigma
                          p_assay, sigma_assay, p_genus_location, n_genus_location, show_trace=F, show_
{
  # Given one DNA sample and the relevant data from the same row, including related data from external
  #  make the distributions in OpenBUGS

  # ARGS:
    # p_species - proportion of that species
    # n_species - amount of data used (length, more length is stronger evidence)
    # p_reads - probability given reads
    # sigma_reads - how much you expect reads to deviate (standard deviation)
    # p_ID - precision multiplied by percentage match
    # sigma_ID - expected deviation in p_ID
    # p_location - probability of species being at that particular location
    # sigma_location - perceived deviation in location
    # p_assay - probability score assigned to assay type
    # sigma_assay - expected deviation in p_assay
    # p_genus_location - proportion of that genus at the location, compared to other genus
    # n_genus_location - number of samples/data size used to calculate p_genus_location
    # show_trace - whether you want the trace plots to be shown (TRUE/FALSE)
    # show_density - whether you want the density plots to be shown (TRUE/FALSE)
    # show_stats - shows the Bayesian estimates if TRUE
    # show_all_vars - If you want to see the other variables in the density plots, set to TRUE



  # load libraries. Might be better to load once, from another function instead of loading each time fu
  library(R2OpenBUGS)
  library(coda)

  # Write the OpenBUGS code:
  writeLines("
    model{

      S ~ dnorm(mu_s, sigma_s)
      ID ~ dnorm(mu_id, sigma_id)
      RD ~ dnorm(mu_rd, sigma_rd)
      AN ~ dnorm(mu_an, sigma_an)
      G ~ dnorm(mu_g, sigma_g)
      AQ ~ dnorm(mu_aq, sigma_aq)

      posterior <- S * ID * RD * AN * AQ / G


    }", con="OpenBUGS_eDNA.txt")
```

```
# Need to pass all the variables so BUGS can use them. Remember, BUGS uses 1/sigma^2 for its dnorm() .
# First find the sigmas from p and n, where applicable

sigma_species = sqrt(p_species*(1-p_species)/n_species)
sigma_genus_location = sqrt(p_genus_location*(1-p_genus_location)/n_genus_location)

if (show_all_vars)
{
  res <- bugs(data=list(mu_s = p_species, sigma_s = 1/sigma_species^2,
              mu_id = p_ID , sigma_id = 1/sigma_ID^2,
              mu_rd = p_reads, sigma_rd = 1/sigma_reads^2,
              mu_an = p_assay, sigma_an = 1/sigma_assay^2,
              mu_g = p_genus_location, sigma_g = 1/sigma_genus_location^2,
              mu_aq = p_location, sigma_aq = 1/sigma_location^2),
            inits = NULL,
            n.chains = 4, n.iter = 11000, n.burnin = 1000,
            parameters.to.save = c("posterior,S,ID,RD,AQ,AN,G"),
            model.file = "OpenBUGS_eDNA.txt",
            DIC = FALSE, codaPkg = TRUE)
} else{
  res <- bugs(data=list(mu_s = p_species, sigma_s = 1/sigma_species^2,
              mu_id = p_ID , sigma_id = 1/sigma_ID^2,
              mu_rd = p_reads, sigma_rd = 1/sigma_reads^2,
              mu_an = p_assay, sigma_an = 1/sigma_assay^2,
              mu_g = p_genus_location, sigma_g = 1/sigma_genus_location^2,
              mu_aq = p_location, sigma_aq = 1/sigma_location^2),
            inits = NULL,
            n.chains = 4, n.iter = 11000, n.burnin = 1000,
            parameters.to.save = c("posterior"),
            model.file = "OpenBUGS_eDNA.txt",
            DIC = FALSE, codaPkg = TRUE)
}



codaobj <- read.bugs(res, quiet=TRUE)

if (show_stats)
{
  summary(codaobj)
}

if (show_density)
{
  plot(codaobj, trace=FALSE)
}

if (show_trace)
{
  plot(codaobj, density=FALSE)
```

```
  }



}
```

```
# Test everything:


p_species = 0.7
n_species = 35
p_reads = 0.8
sigma_reads = 0.4
p_ID = 0.95
sigma_ID = 0.1
p_location = 0.6
sigma_location = 0.5
p_assay = 0.8
sigma_assay = 0.2
p_genus_location = 0.7
n_genus_location = 44
show_trace=F
show_density=T
show_stats=T
show_all_vars=F

test_probs = eDNA_OpenBUGS(p_species, n_species, p_reads, sigma_reads, p_ID, sigma_ID, p_location, sigma
                           p_assay, sigma_assay, p_genus_location, n_genus_location, show_trace=F, show_
```


Density of posterior
0.0
0.4
0.8
1.2
−2
−1
0
1
2
3
4
5
N = 10000 Bandwidth = 0.04503