

Bayesian modelling - low strain, two component (2C): GSI+GSS

Dr Sheng Fan

2024-07-22

1. Libraries and Data

The following libraries are essential for the modelling and visualisation

- JAGS <https://sourceforge.net/projects/mcmc-jags/>
- R2jags <https://cran.r-project.org/web/packages/R2jags/index.html>
- lattice <https://cran.r-project.org/web/packages/lattice/index.html>
- ggplot2 <https://cran.r-project.org/web/packages/ggplot2/index.html>
- gridExtra <https://cran.r-project.org/web/packages/gridExtra/index.html>

```
# Import Libraries

library(R2jags)
library(lattice)
library(ggplot2)
library(gridExtra)

# Import data
file_path <- "D:\\Bayesian_modelling\\Input_data\\Supplement_S3_input_low_strain.csv"

data <- read.csv(file_path)

# Assignment of each column to a variable
Source <- data$Source
GS_ini_mean <- data$GS_ini_mean
GS_ini_upper <- data$GS_ini_upper
GS_ini_lower <- data$GS_ini_lower
rate_c <- data$rate_c
strain_c <- data$strain_c
stress_c <- data$stress_c
T_mean <- data$T_mean
T_upper <- data$T_upper
T_lower <- data$T_lower
exp_type <- data$exp_type
```

2. Bayesian Modelling

2.1 Set Bayesian modelling parameters and output dir

```
## Bayesian model parameters
# Iterations
n_iter <- 4000000
n_iter_str <- format (n_iter, scientific=FALSE)

# Burn-in
n_burnin <- 10000
n_burnin_str <- format (n_burnin, scientific=FALSE)

# Thinning
n_thin <- 20
n_thin_str <- format (n_thin, scientific=FALSE)

model <- "low_strain_two_component"
extension <- ".jags"
model_file <- paste0(model, extension)

## Output dir

strain_condition_input = "Low-strain"

law <- "Two-component"

# Get the current date
date_time_str <- format(Sys.time(), "%Y-%m-%d-%H-%M-%S")

# Create the folder name by concatenating the date and variable name
sub_folder_name <- paste(date_time_str, strain_condition_input, law, n_iter_str, n_burnin_str, n_thin_str)

# Create the folder under the master directory
new_dir <- file.path("D:\\Bayesian_modelling\\Results", sub_folder_name)

dir.create(new_dir)
```

2.2 Modelling

```
R <- 8.314*1e-3 #kJmol-1K-1

# number of observations
N <- length(stress_c)

rate_exp = log10(rate_c)

# model 0
# model 0
data1 <- list("N"=N,
             "rate_exp" = rate_exp,
```

```

        "Stress"=stress_c,
        "R" = R,
        "Init_GS"=GS_ini_mean,
        "Init_GS_upper"=GS_ini_upper,
        "Init_GS_lower"=GS_ini_lower,
        "T_mean"=T_mean,
        "T_upper"=T_upper,
        "T_lower"=T_lower)

vinit <- function(){
  list("n_dis" = 4)
}

params <- c("n_GBS", "n_dis", "p", "Q_dis", "Q_GBS", "A_dis_log", "A_GBS_log")

# Start modelling
s1 <- Sys.time()
m0 = jags(data=data1, inits = vinit, parameters.to.save=params, n.chains = 3,
          n.iter = n_iter, n.burnin=n_burnin, n.thin=n_thin,
          model.file=model_file)

```

```
## module glm loaded
```

```

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 305
##   Unobserved stochastic nodes: 617
##   Total graph size: 6924
##
## Initializing model

```

```

s2 <- Sys.time()
s2-s1

```

```
## Time difference of 8.095596 hours
```

```
m0
```

```

## Inference for Bugs model at "low_strain_two_component.jags", fit using jags,
## 3 chains, each with 4e+06 iterations (first 10000 discarded), n.thin = 20
## n.sims = 598500 iterations saved
##           mu.vect sd.vect  2.5%   25%   50%   75%  97.5%  Rhat  n.eff
## A_GBS_log  2.182   0.535  1.072  1.835  2.203  2.549  3.174 1.001  86000
## A_dis_log  6.569   1.040  4.372  5.922  6.616  7.271  8.489 1.001  51000
## Q_GBS     63.360   2.590 58.172 61.640 63.395 65.112 68.355 1.001 600000
## Q_dis     70.183   4.922 60.019 67.056 70.330 73.479 79.486 1.001  67000
## n_GBS      2.282   0.108  2.065  2.211  2.285  2.357  2.487 1.001  33000
## n_dis      3.769   0.158  3.521  3.659  3.747  3.855  4.141 1.001  60000
## p          1.106   0.074  0.966  1.056  1.105  1.156  1.256 1.001  20000
## deviance   30.758   6.225 19.300 26.473 30.498 34.753 43.733 1.001  93000

```

```

##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 19.4 and DIC = 50.1
## DIC is an estimate of expected predictive error (lower deviance is better).

m0.mcmc <- as.mcmc(m0)

# Document the summary of output
jags_output_path <- file.path(new_dir, "jags_output.txt")
sink(file = jags_output_path)

print(m0)

## Inference for Bugs model at "low_strain_two_component.jags", fit using jags,
## 3 chains, each with 4e+06 iterations (first 10000 discarded), n.thin = 20
## n.sims = 598500 iterations saved
##      mu.vect sd.vect  2.5%  25%  50%  75%  97.5% Rhat  n.eff
## A_GBS_log  2.182  0.535  1.072  1.835  2.203  2.549  3.174 1.001  86000
## A_dis_log  6.569  1.040  4.372  5.922  6.616  7.271  8.489 1.001  51000
## Q_GBS      63.360  2.590 58.172 61.640 63.395 65.112 68.355 1.001 600000
## Q_dis      70.183  4.922 60.019 67.056 70.330 73.479 79.486 1.001  67000
## n_GBS      2.282  0.108  2.065  2.211  2.285  2.357  2.487 1.001  33000
## n_dis      3.769  0.158  3.521  3.659  3.747  3.855  4.141 1.001  60000
## p          1.106  0.074  0.966  1.056  1.105  1.156  1.256 1.001  20000
## deviance   30.758  6.225 19.300 26.473 30.498 34.753 43.733 1.001  93000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 19.4 and DIC = 50.1
## DIC is an estimate of expected predictive error (lower deviance is better).

summary(m0.mcmc)

##
## Iterations = 10001:3999981
## Thinning interval = 20
## Number of chains = 3
## Sample size per chain = 199500
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##      Mean      SD Naive SE Time-series SE
## A_dis_log  6.569 1.0401 1.345e-03    0.0157476
## A_GBS_log  2.182 0.5351 6.917e-04    0.0056090
## deviance  30.758 6.2255 8.047e-03    0.0209093
## n_dis      3.769 0.1577 2.039e-04    0.0006905
## n_GBS      2.282 0.1081 1.397e-04    0.0006121

```

```
## p          1.106 0.0740 9.566e-05      0.0003772
## Q_dis      70.183 4.9217 6.362e-03      0.0732330
## Q_GBS      63.360 2.5896 3.347e-03      0.0252190
##
## 2. Quantiles for each variable:
##
##           2.5%    25%    50%    75%   97.5%
## A_dis_log  4.3721  5.922  6.616  7.271  8.489
## A_GBS_log  1.0724  1.835  2.203  2.549  3.174
## deviance  19.3000 26.473 30.498 34.753 43.733
## n_dis      3.5209  3.659  3.747  3.855  4.141
## n_GBS      2.0645  2.211  2.285  2.357  2.487
## p          0.9663  1.056  1.105  1.156  1.256
## Q_dis      60.0194 67.056 70.330 73.479 79.486
## Q_GBS      58.1721 61.640 63.395 65.112 68.355
```

```
gelman.diag(m0.mcmc)
```

```
## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## A_dis_log          1          1
## A_GBS_log          1          1
## deviance           1          1
## n_dis              1          1
## n_GBS              1          1
## p                  1          1
## Q_dis              1          1
## Q_GBS              1          1
##
## Multivariate psrf
##
## 1
```

```
sink(file = NULL)

# Save raw outputs
all_samples <- as.matrix (m0.mcmc[,])
all_samples_path <- file.path(new_dir, "all_samples.RDS")
save (all_samples, file=all_samples_path)

m0_file_path <- file.path(new_dir, "m0.RDS")
m0mc_file_path <- file.path(new_dir, "m0mc.RDS")

save(m0,file=m0_file_path)
save(m0.mcmc,file=m0mc_file_path)
```