Jags example

Patrick Brown

Wednesday 14 September 2016

```
> library('glmmBUGS')
> haveR2jags = require('R2jags', quietly=TRUE)
> print(haveR2jags)
```

```
[1] TRUE
```

```
> haveRjags = requireNamespace('rjags', quietly=TRUE)
> print(haveRjags)
```

```
[1] TRUE
```

```
> haveRjags = try(
+                loadNamespace('rjags'),
+          silent=TRUE)
> if(class(haveRjags) == 'try-error') {
+        haveR2jags = FALSE
+ }
```

If Jags isn't available, the `rjags` namespace will give an error when loaded.

1 Muscle data

```
> data('Muscle', package='nlme')
```

```
> muscleRagged = glmmBUGS(conc ~ length, data=Muscle,
+        effects="Strip", family="gaussian",
+        modelFile='model.bug', reparam='Strip')
> startingValues = muscleRagged$startingValues
```

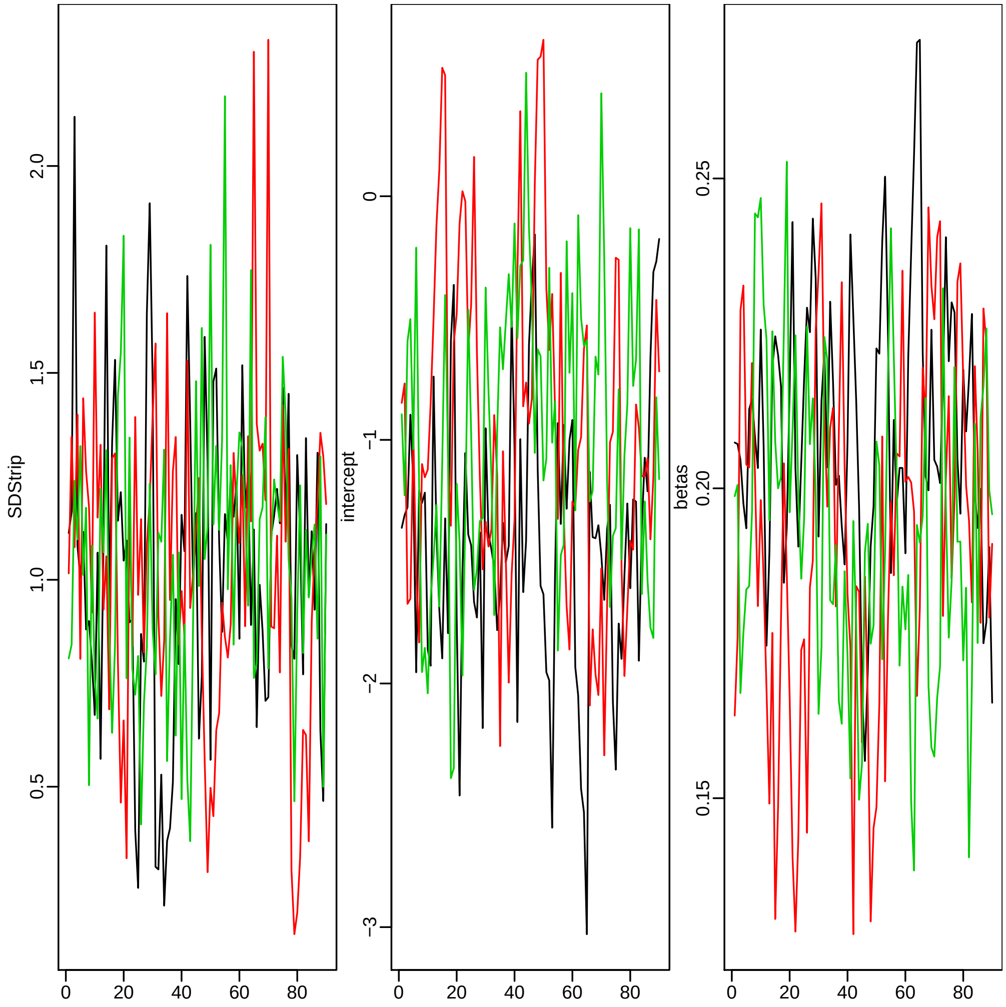
```
> source("getInits.R")
> if(haveR2jags) {
+        muscleResult = jags(
+          muscleRagged$ragged, getInits,
+          parameters.to.save = names(getInits()),
+      model.file="model.bug", n.chain=3, n.iter=1000,
+       n.burnin=100, n.thin=10, refresh=200,
+       working.directory=getwd())
+ } else {
+         muscleResult = NULL
+ }
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 60
   Unobserved stochastic nodes: 25
   Total graph size: 285

Initializing model
```

```
> if(!is.null(muscleResult)) {
+         muscleParams = restoreParams(
+         muscleResult$BUGSoutput,
+         muscleRagged$ragged)
+ summaryChain(muscleParams)$scalars[,c('mean','2.5pct','97.5pct')]
+ }
```

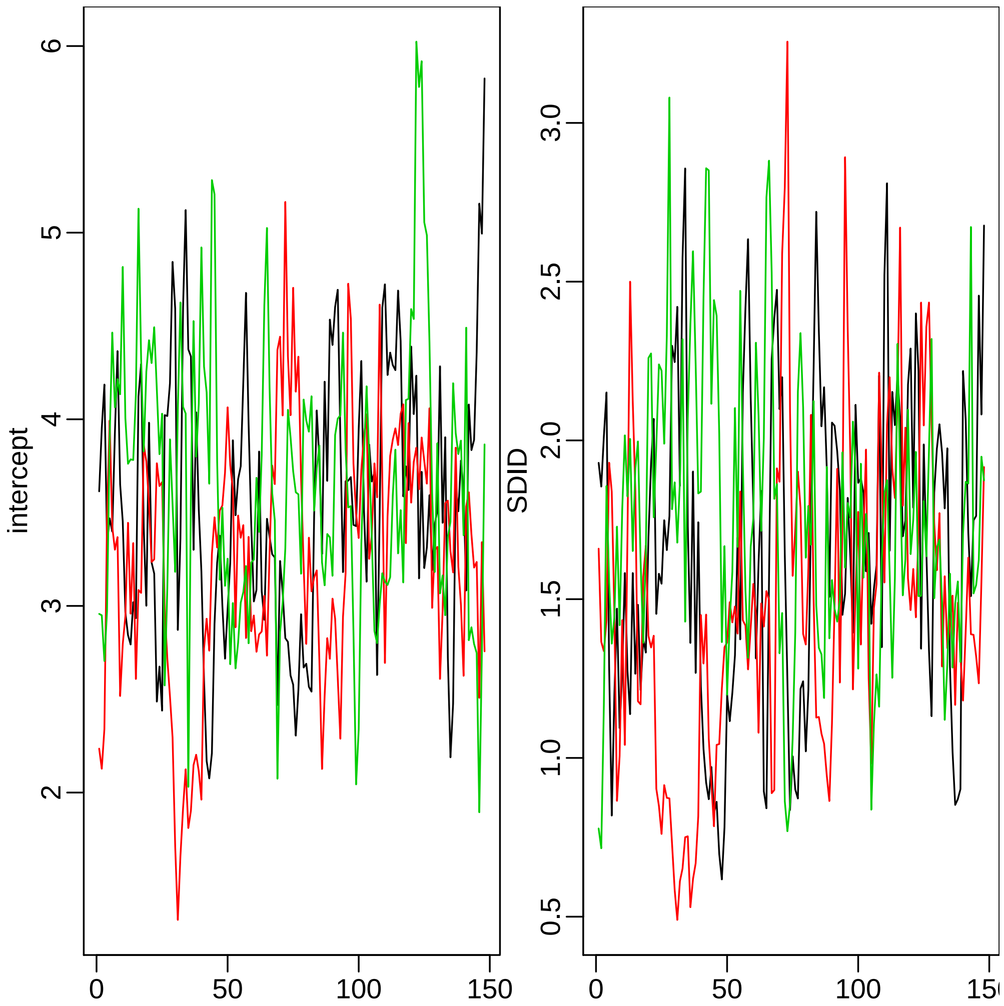
```
                mean     2.5pct   97.5pct
SDStrip    1.0348909  0.3046083 1.8081782
intercept -1.1148962 -2.3487627 0.3694371
betas      0.1987191  0.1427875 0.2455235
```

```
> if(!is.null(muscleResult)) {
+         checkChain(muscleParams)
+ } else {
+         plot(1, main='Not run, some packages missing')
+ }
```



2 Bacteria

```
> data('bacteria', package='MASS')
> bacterianew <- bacteria
> bacterianew$yInt = as.integer(bacterianew$y == "y")
> levels(bacterianew$trt) <- c("placebo", "drug", "drugplus")

> bacrag <- glmmBUGS(formula = yInt ~ trt + week,
+     data = bacterianew,
+     effects = "ID", modelFile = "bacteria.txt",
+     family = "bernoulli",brugs=TRUE)
> source("getInits.R")
> startingValues = bacrag$startingValues

> if(haveR2jags) {
+   bacResult = R2jags::jags(bacrag$ragged,
+       inits=getInits,
+       model.file = "bacteria.txt", n.chain = 3,
+       n.iter = 600, n.burnin = 10, refresh=200,
+       parameters = names(getInits()),
+       n.thin = 4)
+ } else {
+         bacResult = NULL
+ }

Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 220
   Unobserved stochastic nodes: 55
   Total graph size: 1156

Initializing model

> if(!is.null(bacResult)) {
+ bacParams = restoreParams(bacResult$BUGSoutput,
+      bacrag$ragged)
+ }

> if(!is.null(bacResult)) {
+         bacsummary = summaryChain(bacParams)
+         bacsummary$betas[,c('mean', 'sd')]
+ }

                       mean         sd
trtdrug          -1.4037641 0.76605160
trtdrugplus      -0.8977213 0.81224717
betaobservations -0.1564980 0.05750264

> if(!is.null(bacResult)) {
+         checkChain(bacParams, c("intercept", "SDID"),oneFigure=TRUE)
+ } else {
+         plot(1, main='Not run, some packages missing')
+ }
```


intercept
6
5
4
3
2
0
50
100
150
SDID
3.0
2.5
2.0
1.5
1.0
0.5
0
50
100
150