Winbugs code for Hierarchical Bayesian method for multilocus association analysis in general population sample

Written by:
Madhuchhanda Bhattacharjee (http://www.rni.helsinki.fi/~mab/)

Reference:
Bayesian Association-Based Fine Mapping in Small Chromosomal Segments
Mikko J. Sillanpää and Madhuchhanda Bhattacharjee
To appear in Genetics

Data:
Multi-allelic (or bi-allelc) markers, unknown phase, missing data (to some extent),
Continuous or binary phenotype
Small chromosomal segment spanned by a dense set of closely linked markers,
(alternately data from long segment can also be used and accordingly independent prior for marker indicators should be used)

Few model assumptions:
Covariance between consecutive loci accounted using genetic or physical map distances between markers
Dependent variable selection model for markers using joint prior distribution on indicators
Mapping assuming putative genes only at marker points

Winbugs code

```
model {

# Modelling missing data
 for(j in 1 : m) {
   for(k in 1:n2[j] ){
     p1[ j, k ] <- 1/n2[j]                    # Uniform allele frequencies assumed for missing data augmentation
   }
   for(k in n2[j]+1:n2max ){
     p1[ j, k ] <- 0                          # To complete probability matrix in case number of alleles differ from marker to marker
   }
 }
 for(i in 1 : n1) {
   for(j in 1 : m ) {
     y[i, 2*j-1 ]  ~ dcat( p1[ j, 1:n2[j] ] )   # Model for first allele at locus j
     y[i, 2*j ]  ~ dcat( p1[ j, 1:n2[j] ] )     # Model for second allele at locus j
#      y[i, j ]  ~ dcat( p1[ j, 1:n2[j]] )      # Model for single allele data at locus j
   }
 }

# Model for the indicators
 lambda ~ dgamma(1, 0.01)                     # Smoothing parameter
 p[1] <- 1/m                                  # Stringent prior belief on number of markers independently selected in the model
 x[1] ~dbern( p[1] )                          # Indicator for first marker
 for(j in 2 : m ) {
   p[ j ] <- 1/m                              # Stringent prior belief on number of markers independently selected in the model
   q[j-1]  <- 1/exp( lambda*d[j-1] )          # Smoothing parameter and distances used to generate probability of repeating previous indicat
   w[ j-1 ] ~dbern( q[ j-1 ] )                # LD indicator generated using distance information
   x0[ j ] ~dbern( p[ j ] )                   # Independent indicator
   x[j] <- (1-w[j-1] ) * x0[ j ] + w[j-1]*x[j-1]   # Mixture of independent indicator and previous indicator, using LD indicator
 }
 x1 <- sum( x[1:m] )                          # Total number of markers selected in the model
 for( j in 1 : m+1 ) {
   x2[ j ] <- equals( x1,  j-1)               # To estimate distribution of number of markers selected in the model
 }

# Modelling phenotype data
 for(j in 1 : m ) {
   tau[j] ~dgamma( 1, 1)                      # Precision parameter (i.e. 1/ variance) for random variance model
   for( k in 1 : n2[j] ) {
     beta[ j, k ] ~ dnorm( 0, tau[j] )        # Allelic coefficients for random variance model
#      beta[ j, k ] ~ dnorm( 0, 1)            # Allelic coefficients for fixed variance model
   }
   for( k in n2[j]+1:n2max ) {
     beta[ j, k ] <- 0                        # To complete coefficient matrix in case number of alleles differ from marker to marker
   }
   b1[ j ] <- x[ j ]/tau[ j ]                 # Weighted genetic variance for multiallelic data
   b1[ j ] <- x[ j ]/(abs( beta[j,1] - beta[j,2])   # Weighted absolute genetic effect for biallelic data
   for( k in 1 : n2[j] ) {
```

```
      b0[ j, l ] <- x[j]*beta[j, l]                          # Weighted allelic effects
    }
  }
  mu0 ~dnorm( 0, 0.1 )                                       # Overall mean
#  tau0 ~dgamma( 1, 1)                                       # For continuous phenotype: Precision parameter corresponding to residual variance

  for(i in 1 : n1) {
    for(j in 1 : m ) {
      beta1[i, j] <- x[j]*(beta[ j, y[i, 2*j-1] ] + beta[ j, y[i, 2*j] ] )   # For two allele data
#       beta1[i, j] <- x[j]*beta[ j, y[i, j] ]                # For single allele data
    }
    logit(p2[i]) <- mu0+ sum( beta1[i, 1:m] )                # For binary phenotype, transformation using logit link function
    z1[i] ~ dbern( p2[i] )                                   # Model for binary phenotype data
#     p2[i] <- mu0+ sum( beta1[i, 1:m] )                     # For continuous phenotype, variable selection model for mean
#     z1[i] ~ dnorm( p2[i], tau0 )                           # Model for continuous phenotype data
  }

}
```

Input data:

n1: Number of individuals
m: Number of candidate markers
n2: Vector of number of alleles at each marker
n2max: A number larger than the maximum number of alleles at any locus
d: Vector of pair wise distances between loci
z1: phenotypic data
y: genotype data