Outline of this document:

1. Description of MTM function;
2. Models Implemented, (this section is incomplete and should be extended to describe models programmed in the REC, FA, UN and DIAG options);
3. Syntaxes;
4. Example.

***A) DESCRIPTION***

This program does inference of parameters from the posterior distribution of a multivariate normal model with structured co-variance matrices.

***B) MODELS IMPLEMENTED***

1. The Univariate Linear Additive Genetic Model

1. Model with one trait (Y with *n* subjects)

:a vector of fixed effects

**X** and **Z**: incidence matrices associating andwith **y**

**I**: an identity matrix of order *n × n*

: the residual variance of the model

: a vector of genetic values for trait Y

* **:** the additive genetic relationship matrix (expected proportion of allele sharing)
* : the genetic variance
* : Genomic Best Linear Unbiased Predictor (G-BLUP)
* **:** the marker-based measure of allele sharing between individuals (realized proportion of allele sharing)
* : the genomic variance

1. Prior distribution for variances
2. Proper uniform distribution
3. Scaled inverse chi-square distribution
4. Joint posterior density

* Assuming that and are independent a priori

2. The Multivariate Linear Additive Genetic Model

1. Model for two traits (Y and Z)

:a vector of fixed effects affecting each trait

and : incidence array relating location effects for each trait

(): a vector of residual effects for each trait

, where **v** contains **y** and **z**

,

, the residual variance-covariance matrix

* : *n* × *n* identity matrix
* , 2 × 2 residual dispersion matrix

(): a vector of additive genetic values for each trait

* **:** Additive genetic relationship matrix
* , 2 x 2 matrix with additive genetic (co)variance components
* **:** the marker-based measure of allele sharing between individuals
* , 2 x 2 matrix with additive genetic (co)variance components

1. Prior distributions for variances
2. Two-dimensional scaled inverted Wishart distribution for and , where *p* = number of traits
3. Hyper-parameters of the distribution

* , degrees of freedom
* , scale matrix

1. Bivariate model with

Set the mode of the prior to be at half of variance of phenotype. Therefore,

***C) PROGRAM SYNTAXES***

Phenotypes: Y matrix (*n* × *q*, where *q* is the number of traits, if some individuals have some traits that were not measured introduce NA's in Y, the program handles this).

It handles a variable number of random effects, which are specified in the argument K (for kernel). K is a two level list. Each of the inner elements of the list is used to specify one random effect. For each random effect you need to provide a co-variance structure, see below K=G, and a type argument that specifies the form of the within subject covariance. The program supports several, the most important ones are type='UN', type='DIAG' and type='FA', which stands for unstructured, diagonal and factor-analytic, respectively. You also need to specify hyper-parameters for prior distribution. For UN we use an inverse-Whishart prior; therefore, you need to specify prior degrees of freedom (df0, real number, df0 > *q* - 1) and a scale matrix (S0, *q* × *q* positive definite, I usually use a diagonal scale matrix, see example below). For DIAG we assign scaled inverse chi-squares to each of the diagonal elements; therefore, df0 and S0 must be vectors (*q*-dimensional each), and each entry defines the degrees of freedom and scale parameter of a scaled inverse chi-square distribution. For FA you need to specify the number of factors (nF, integer, nF > 1). For this model you also need to specify the degrees of freedom and scale parameters of the variances of the specific factors, these are specified with a vector df0 and a vector S0.

For the residual variance we indicate the structure and the hyper-parameters using the argument resCov, which is a list (see below).

The program will introduce one intercept by default (fm$mu), if you want to include fixed effects you need to provide an incidence matrix (Xf, *n* × *r*, *r* = number of fixed effects). The program applies these effects to all traits.

***D) EXAMPLE***

Example of the use of the script with the wheat data set available at BGLR library. The data has only 4 traits, the example uses un-structured co-variance matrix for the genetic part, and, since these records come from different sites, uses a diagonal co-variance matrix for the residuals.

|  |
| --- |
| rm(list=ls())  source('~/Dropbox/MTM/code/MTM7.r')  library(BGLR)  data(wheat)  G<-tcrossprod(scale(wheat.X,center=TRUE,scale=TRUE))/ncol(wheat.X)  nTraits<-ncol(wheat.Y)  df0<-3  R2<-0.5  ## NOTE: K is similar to ETA in BGLR, but only for RKHS in the case of this program.  ## One can provide K or EVD via V and d, I believe...  ## Use much longer chains, this is just an example!  fm<-MTM(Y=wheat.Y,  K=list(  list(K=G,  COV=list(type='UN',df0=(df0+nTraits),S0=diag(R2\*(df0+2\*nTraits+1),nTraits))  )  ),  resCov=list(type='DIAG',df0=rep(df0,nTraits),S0=diag(R2\*df0/(df0+2),nTraits)),  nIter=500,burnIn=100  )  ## Some estimates  fm$K[[1]]$G # estimated genomic covariance matrix  fm$K[[1]]$U # predicted genomic values  fm$resCov$R # estimated residual covariance matrix |