GBLUP_application_code.R

zhezhang

2020-02-21

```r
setwd("~/Desktop/icloud/0-tongbu/6-teacher/1-book/gblup")

library(xbreed)
```

```
## ("|-----------------------------------------------------|")

## ("|                      xbreed                         |")

## ("|    Genomic simulation of purebreds and crossbreds   |")

## ("|               March 2017 Version 1.0.1              |")

## ("|                                                     |")

## ("|             H.Esfandyari,A.C.Sorensen               |")

## ("| Center for Quantitative Qenetics and Genomics (QGG) |")

## ("|             Aarhus University,Denmark               |")

## ("|                                                     |")

## ("|-----------------------------------------------------|")

## ("|Questions and bugs: esfandyari.hadi@gmail.com        |")

## ("|Development of xbreed was supported by GenSAP.       |")

## ("|-----------------------------------------------------|")
```

```r
library(BGLR)
library(Matrix)

genome<-data.frame(matrix(NA, nrow=3, ncol=6))
names(genome)<-c("chr","len","nmrk","mpos","nqtl","qpos")
genome$chr<-c(1:3)
genome$len<-c(80,60,50)
genome$nmrk<-c(500,1000,250)
genome$mpos<-c('rnd','rnd','rnd')
genome$nqtl<-c(40,50,45)
genome$qpos<-c('rnd','rnd','rnd')
genome
```

```
##   chr len nmrk mpos nqtl qpos
## 1   1  80  500  rnd   40  rnd
## 2   2  60 1000  rnd   50  rnd
## 3   3  50  250  rnd   45  rnd
```

```
hp<-make_hp(hpsize=200,
            ng=500,h2=0.3,d2=0.1,phen_var=1,
            genome=genome,mutr=5*10**-4,sel_seq_qtl=0.1,sel_seq_mrk=0.05,laf=0.5)
```

```
## Historical pop is initialized...
## Extracting segregating qtl loci ...
## ----No. segregating QTL: 111 out of 135
## Extracting segregating markers ...
## ----No. segregating markers: 1583 out of 1750
## Simulating trait ...
## Output data preparation ...
## Establishment of historical population completed
```

```
Male_founders<-data.frame(number=50,select='rnd')
Female_founders<-data.frame(number=100,select='rnd')

Selection<-data.frame(matrix(NA, nrow=2, ncol=2))
names(Selection)<-c('Number','type')
Selection$Number[1:2]<-c(60,100)
Selection$type[1:2]<-c('rnd','rnd')
Selection
```

```
##   Number type
## 1     60  rnd
## 2    100  rnd
```

```
sh_output<-data.frame(matrix(NA, nrow=5, ncol=5))
names(sh_output)<-c("data","qtl","marker","freq_mrk","freq_qtl")
sh_output[,1]<-c(0:4) # Save data for generations 0,3,4
sh_output[,2]<-c(0:4) # Save qtl genotype for generations 1,2,4
sh_output[,3]<-c(0:4) # Save marker genotype for generations 3,4,5
sh_output[,4]<-c(0:4) # Save marker frequencies for generations 3,4,5
sh_output[,5]<-c(0:4) # Save qtl frequencies for generations 3,4,5
sh_output
```

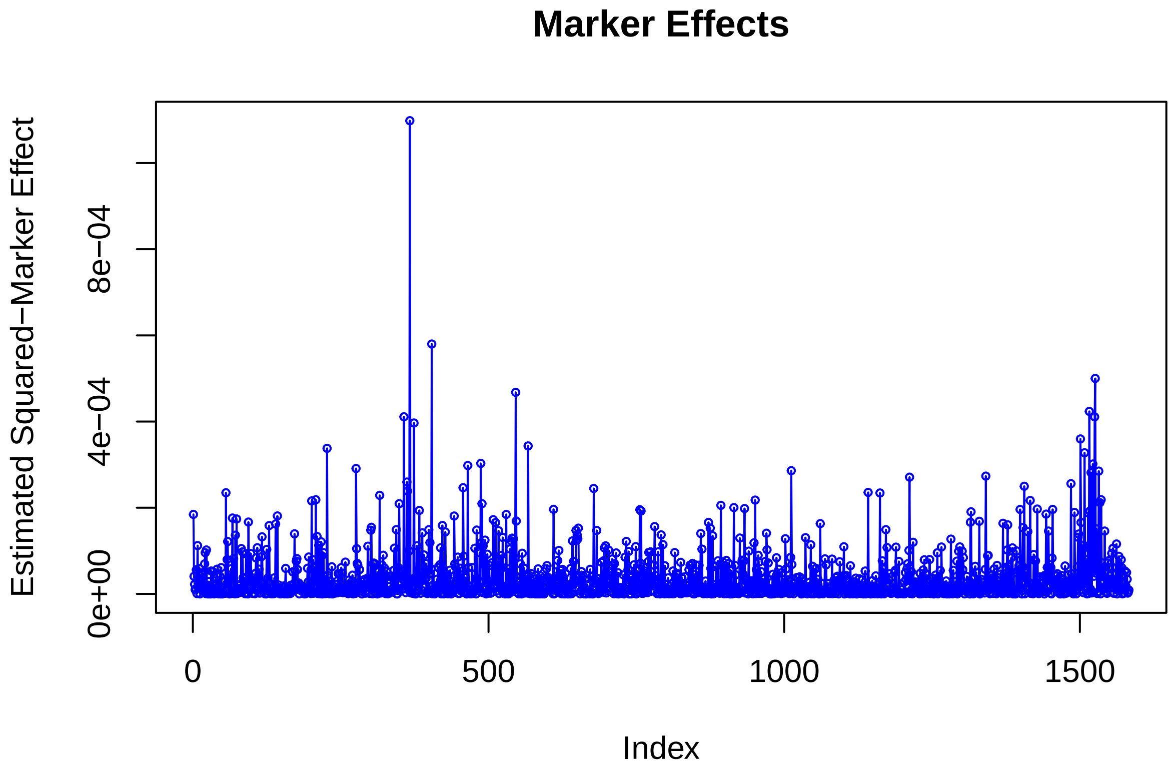
```
##   data qtl marker freq_mrk freq_qtl
## 1    0   0      0        0        0
## 2    1   1      1        1        1
## 3    2   2      2        2        2
## 4    3   3      3        3        3
## 5    4   4      4        4        4
```

```
RP<-sample_hp(hp_out=hp,Male_founders=
                Male_founders,Female_founders=Female_founders,
                ng=4,Selection=Selection,litter_size=5,saveAt="RP",
                sh_output=sh_output,Display=TRUE)
```

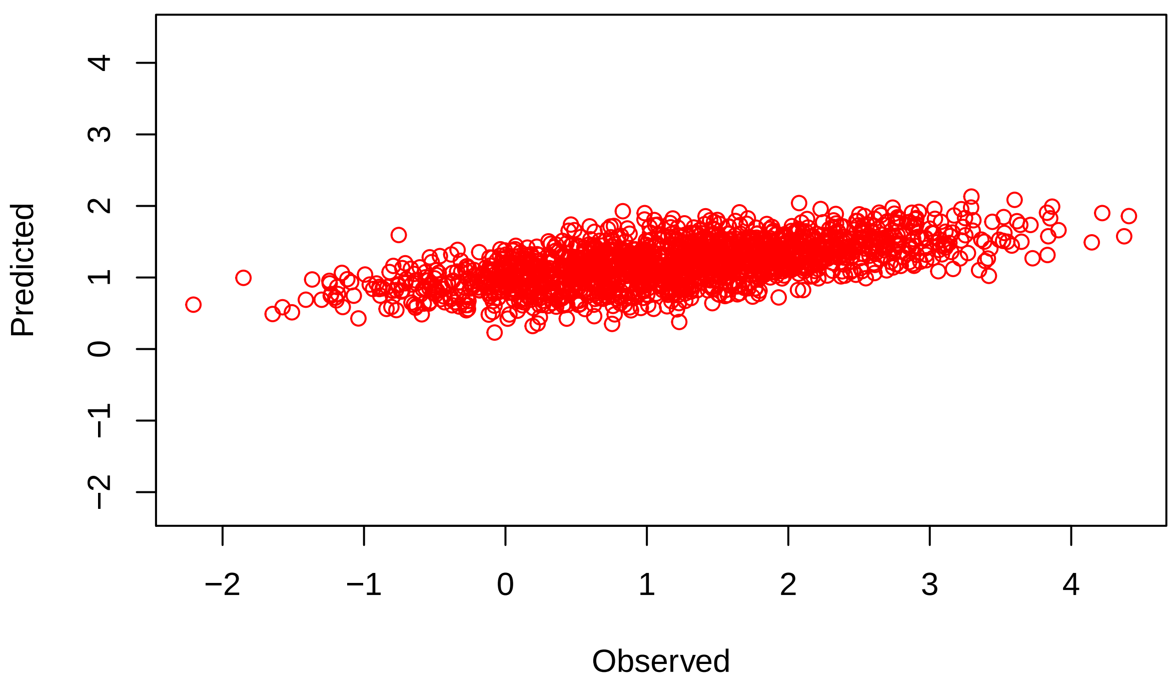
```
## Controlling input data ...
## Intializing base population ...
## Generation 0 started .........
## Generation 0 is finished. Time taken: 1.444055
## Generation 1 started .........
## Generation 1 is finished. Time taken: 1.07679
## Generation 2 started .........
## Generation 2 is finished. Time taken: 1.022432
## Generation 3 started .........
## Generation 3 is finished. Time taken: 0.9342191
## Generation 4 started .........
## Generation 4 is finished. Time taken: 0.9519191
## Output data preparation ...
##   Generation Phenotype   TrueBV M_accuracy F_accuracy heritability
## 1          1  1.255735 1.232595  0.4359257  0.5282426    0.1990000
## 2          2  1.180071 1.196310  0.5114829  0.4944380    0.2244696
## 3          3  1.164660 1.160853  0.5058930  0.5286205    0.2363670
## 4          4  1.208514 1.180589  0.5197019  0.4978231    0.2791541
## Writing output files ...
## Sampling hp is done!
```

```r
write.table(hp$linkage_map_qtl_mrk,file="linkage_map.txt",row.names = F,quote=F)
write.table(hp$allele_effcts,file="allele_effcts.txt",row.names = F,quote=F)

####data preparation
dat_train<-list()
mrk_train<-list()
for(i in 0:3){
  dat_train[[i+1]]<-read.table(paste0("RP_data_",i,".txt"),h=T)
  mrk_train[[i+1]]<-read.table(paste0("RP_mrk_",i,".txt"),skip=1,h=F)
}
dat_train<-do.call(rbind,dat_train)
mrk_train<-do.call(rbind,mrk_train)

i<-4
dat_test<-read.table(paste0("RP_data_",i,".txt"),h=T)
mrk_test<-read.table(paste0("RP_mrk_",i,".txt"),skip=1,h=F)


n<-nrow(dat_train)
geno_train<-mrk_train[,-c(1,2)]
geno_test<-mrk_test[,-c(1,2)]
m<-ncol(geno_train)/2

geno_train<-geno_train-1
geno_test<-geno_test-1
geno_train<-as.matrix(geno_train)
geno_test<-as.matrix(geno_test)
multip<-do.call(bdiag,rep(list(c(1,1)),m))
geno_train<-geno_train%*%multip
geno_test<-geno_test%*%multip

##SNP-BLUP using BGLR
sex<-as.numeric(dat_train$sex)
```

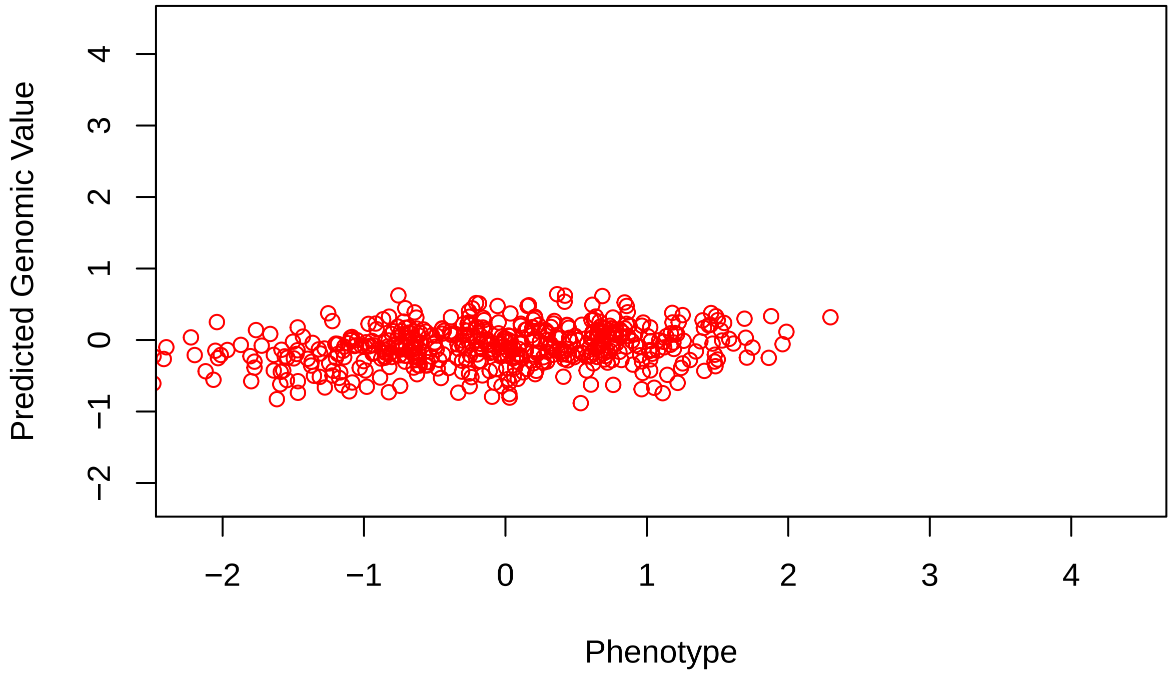
```
sex<-sex-1
y<-dat_train$phen

nIter=12000
burnIn=2000
saveAt='SNP-BLUP'

# Setting the linear predictor
ETA<-list( list(~factor(sex),
                model='FIXED'),
           list(X=geno_train, model='BRR')
)

fm<-BGLR(y=y,ETA=ETA,nIter=nIter, burnIn=burnIn,saveAt=saveAt,verbose = F)
ghat<-fm$ETA[[2]]$b
write.table(fm$ETA[[2]]$b,file="snp_eff.txt",
            row.names = F,col.names = F,
            quote = F)

plot(ghat^2, ylab='Estimated Squared-Marker Effect',
     type='o',cex=.5,col=4,main='Marker Effects')
```



```
yHat<-fm$yHat
tmp<-range(c(y,yHat))
plot(yHat~y,xlab='Observed',ylab='Predicted',col=2,
      xlim=tmp,ylim=tmp)
```



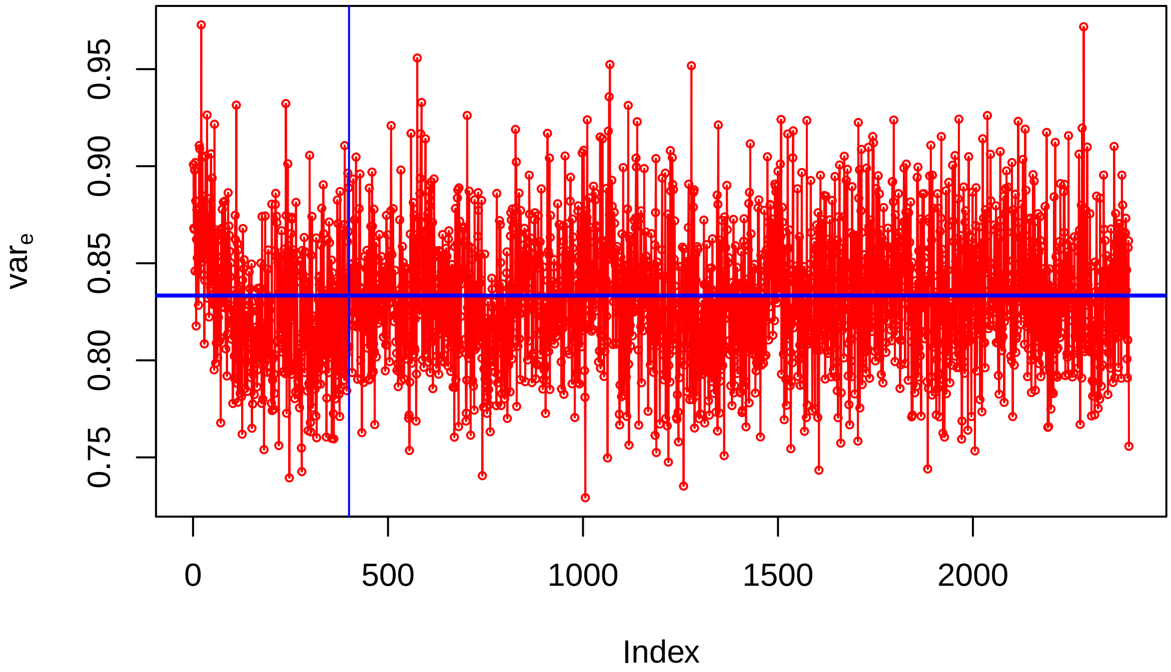
```
gebv<-geno_test%*%ghat
gebv<-gebv[,1]

sex_test<-as.numeric(dat_test$sex)-1
y_test<-dat_test$phen-fm$mu-fm$ETA[[1]]$b*sex_test
plot(gebv~y_test,xlab='Phenotype',
      ylab='Predicted Genomic Value',col=2,
      xlim=tmp,ylim=tmp)
```



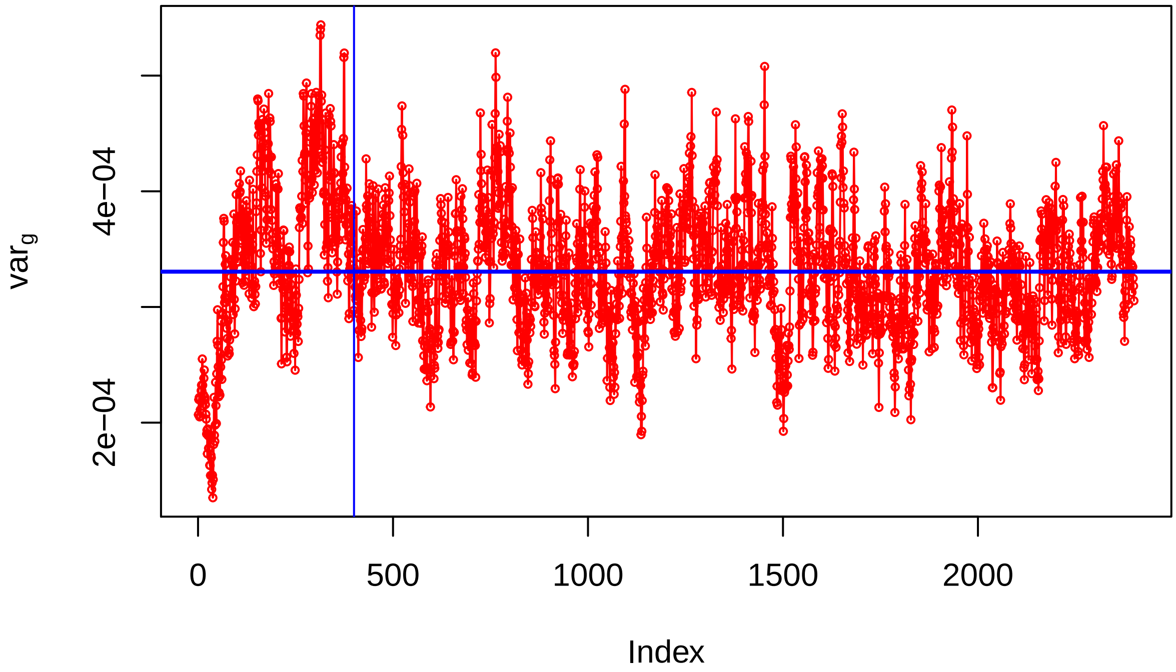
```
cor(gebv,y_test) #0.3438969
```

```
## [1] 0.2257575
```

```
varE<-scan(paste0(saveAt,'varE.dat'))
plot(varE,type='o',col=2,cex=.5,ylab=expression(var[e]));
abline(h=fm$varE,col=4,lwd=2);
abline(v=fm$burnIn/fm$thin,col=4)
```

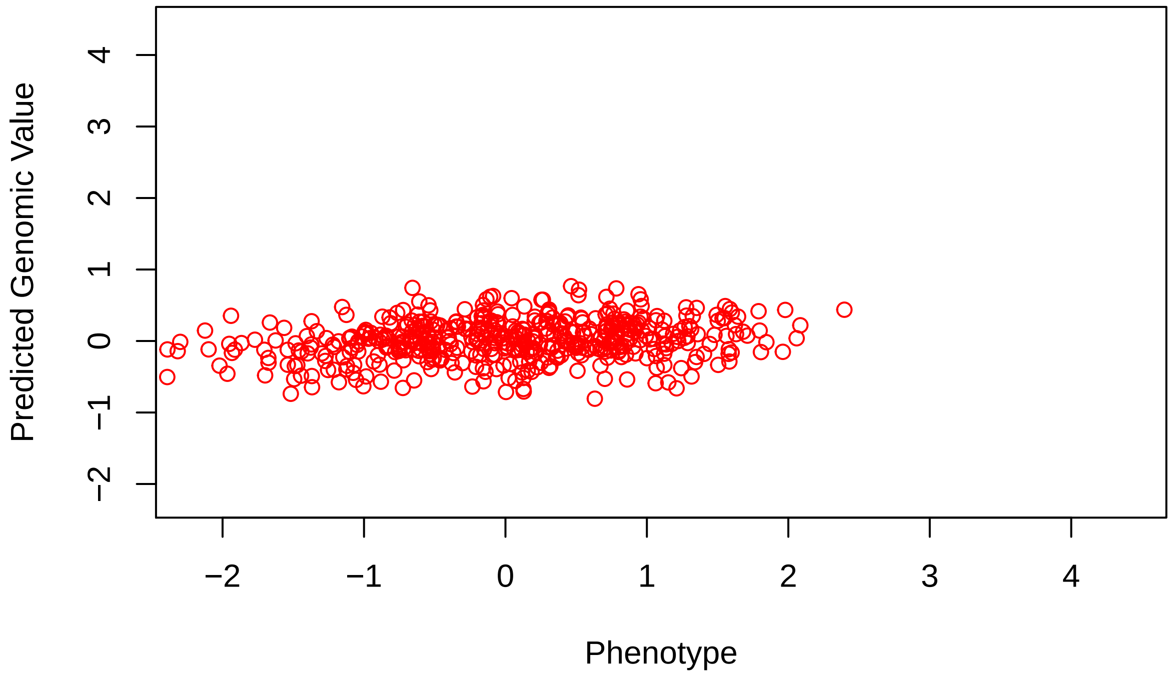


```
varg<-scan(paste0(saveAt,'ETA_2_varB.dat'))
plot(varg,type='o',col=2,cex=.5,ylab=expression(var[g]));
abline(h=fm$ETA[[2]]$varB,col=4,lwd=2);
abline(v=fm$burnIn/fm$thin,col=4)
```



```
### GBLUP
geno<-do.call(rbind,list(geno_train,geno_test))
dat<-do.call(rbind,list(dat_train,dat_test))
sex<-as.numeric(dat$sex)-1

#calculate gmatrix using VaRanden method
maf<-colMeans(geno)/2

z<-scale(geno,scale=F)
gmat<-tcrossprod(z)/sum(2*maf*(1-maf))

#### mask generation 4's phenotype
y<-dat$phen
y[which(dat$generation==4)]<-NA

## run BGLR
nIter=12000
burnIn=2000
saveAt='GBLUP'

# Setting the linear predictor

ETA<-list( list(~factor(sex),
                model='FIXED'),
           list(K=gmat, model='RKHS')
)
```

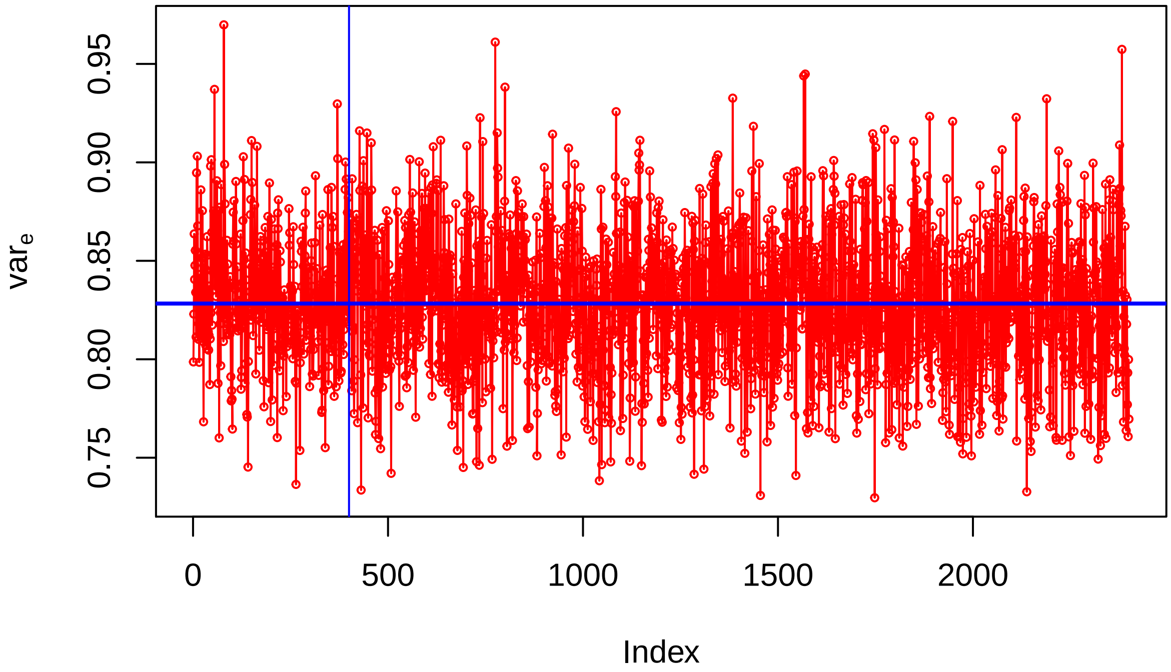
```
fm2<-BGLR(y=y,ETA=ETA,nIter=nIter, burnIn=burnIn,saveAt=saveAt,verbose = F)
gebv<-fm2$ETA[[2]]$u[fm2$whichNa]

y_test<-dat_test$phen-fm2$mu-fm2$ETA[[1]]$b*sex_test
plot(gebv~y_test,xlab='Phenotype',
     ylab='Predicted Genomic Value',col=2,
     xlim=tmp,ylim=tmp)
```



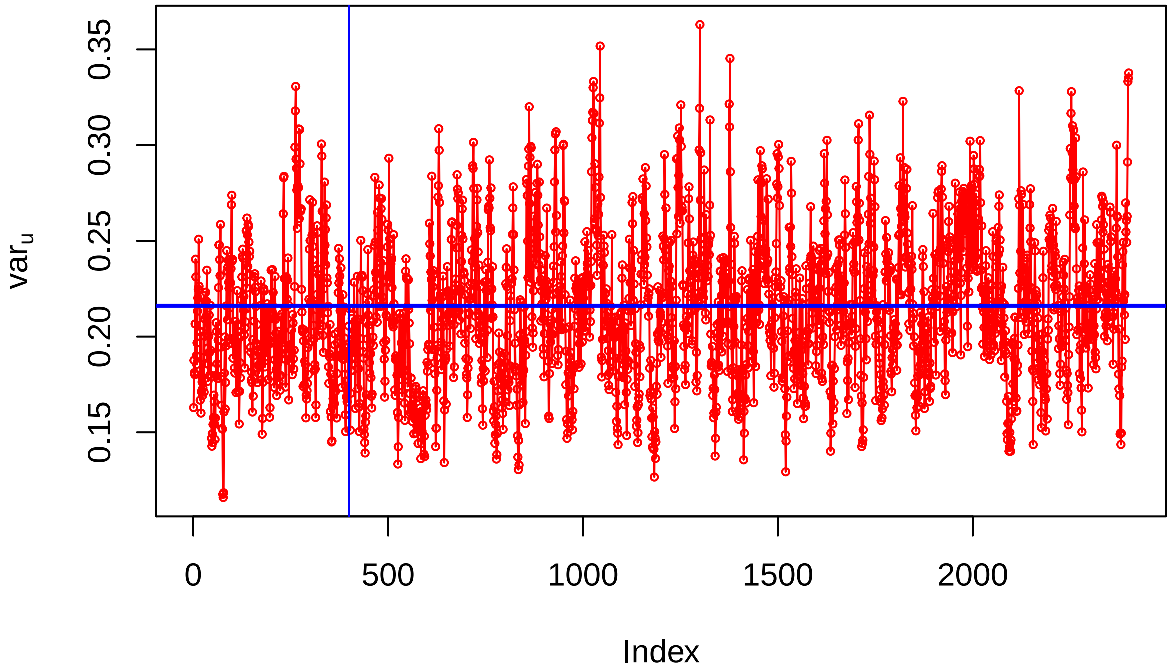
```
cor(gebv,y_test) #0.344162
```

```
## [1] 0.2277904
```

```
varE<-scan(paste0(saveAt,'varE.dat'))
plot(varE,type='o',col=2,cex=.5,ylab=expression(var[e]));
abline(h=fm2$varE,col=4,lwd=2);
abline(v=fm2$burnIn/fm2$thin,col=4)
```

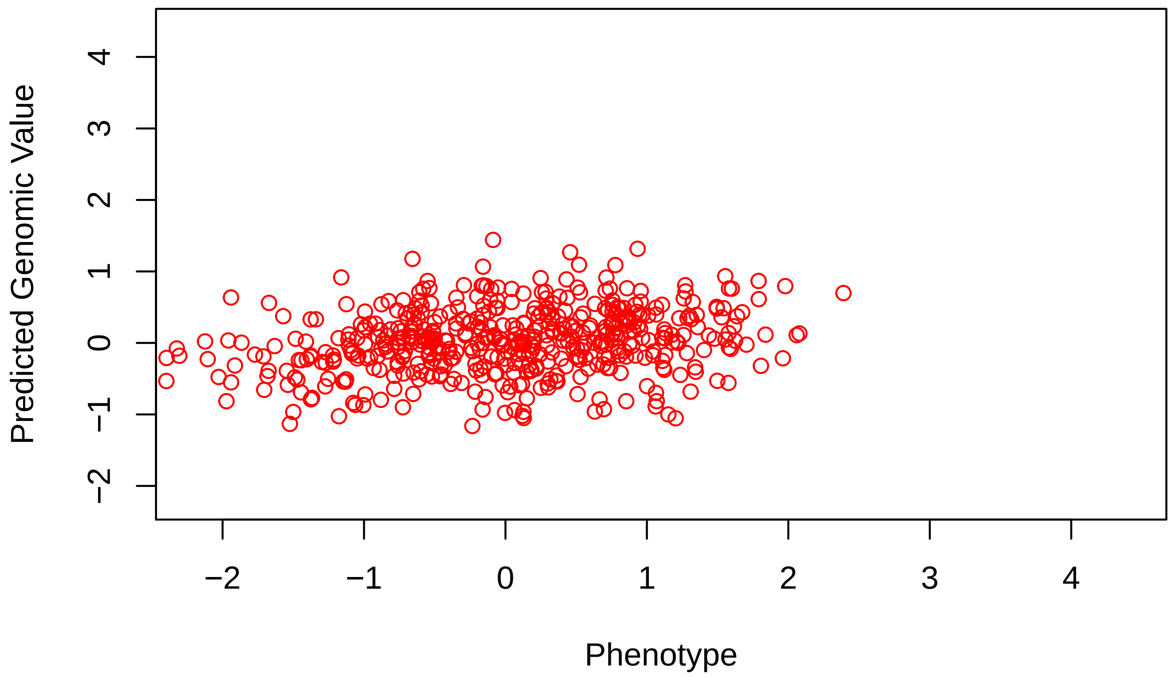


```
varu<-scan(paste0(saveAt,'ETA_2_varU.dat'))
plot(varu,type='o',col=2,cex=.5,ylab=expression(var[u]));
abline(h=fm2$ETA[[2]]$varU,col=4,lwd=2);
abline(v=fm2$burnIn/fm2$thin,col=4)
```



```
### TABLUP

#calculate gmatrix using Zheng et al.(2010) method
snp_eff<-read.table("snp_eff.txt",h=F)[,1]

maf<-colMeans(geno)/2
weight<-diag(2*maf*(1-maf)*snp_eff^2)

z<-scale(geno,scale=F)
gmat2<-z%*%weight%*%t(z)/sum(2*maf*(1-maf))

## run BGLR
nIter=12000
burnIn=2000
saveAt='TABLUP'

# Setting the linear predictor

ETA<-list( list(~factor(sex),
                model='FIXED'),
           list(K=gmat2, model='RKHS')
)

fm3<-BGLR(y=y,ETA=ETA,nIter=nIter, burnIn=burnIn,saveAt=saveAt,verbose = F)
gebv<-fm3$ETA[[2]]$u[fm3$whichNa]
```

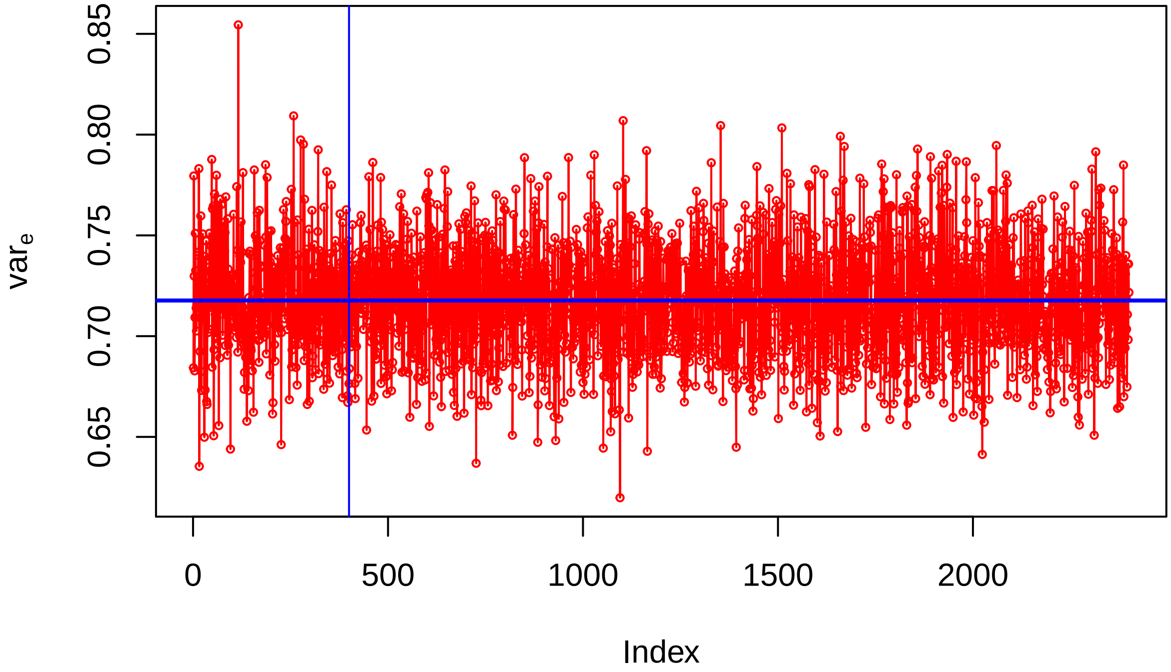
```r
y_test<-dat_test$phen-fm3$mu-fm3$ETA[[1]]$b*sex_test
plot(gebv~y_test,xlab='Phenotype',
     ylab='Predicted Genomic Value',col=2,
     xlim=tmp,ylim=tmp)
```



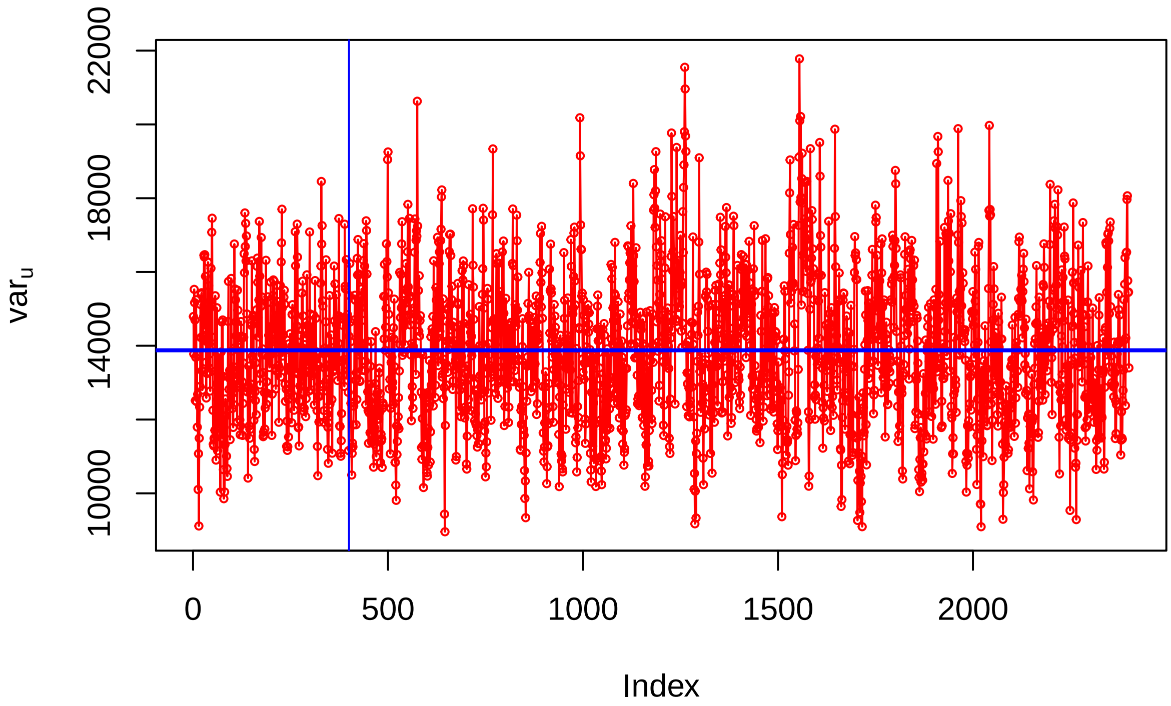
```r
cor(gebv,y_test) #0.3453815
```

```
## [1] 0.2250914
```

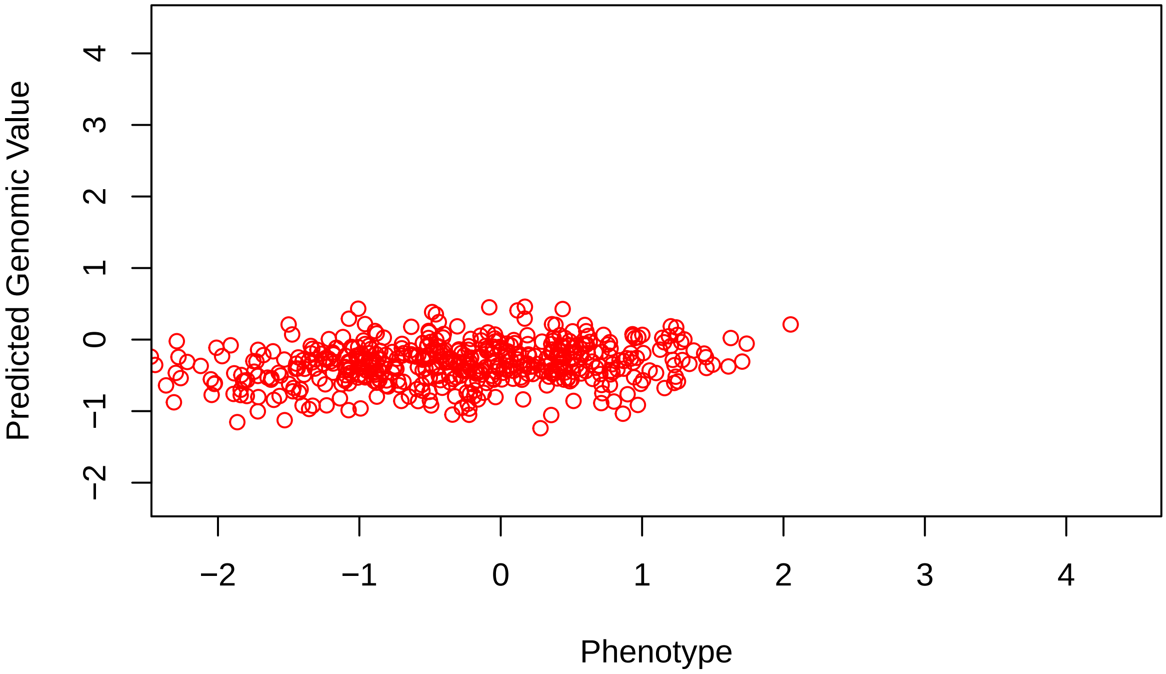
```r
varE<-scan(paste0(saveAt,'varE.dat'))
plot(varE,type='o',col=2,cex=.5,ylab=expression(var[e]));
abline(h=fm3$varE,col=4,lwd=2);
abline(v=fm3$burnIn/fm3$thin,col=4)
```



```
varu<-scan(paste0(saveAt,'ETA_2_varU.dat'))
plot(varu,type='o',col=2,cex=.5,ylab=expression(var[u]));
abline(h=fm3$ETA[[2]]$varU,col=4,lwd=2);
abline(v=fm3$burnIn/fm3$thin,col=4)
```



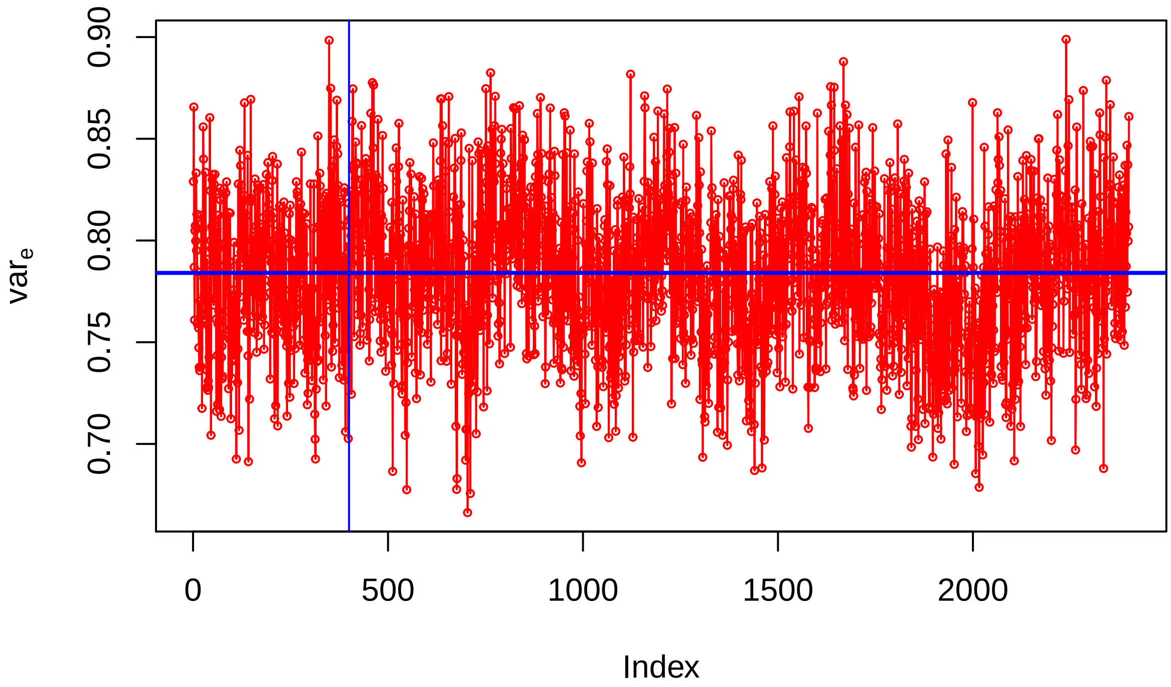
```
### GBLUP with dominance
n<-nrow(dat)
m<-ncol(geno)
#calculate dmatrix accounting for the dominance relationship between individuals
d0<--2*(1-maf)^2
d1<-2*maf*(1-maf)
d2<--2*maf^2
d<-matrix(0,n,m)
for(i in 1:m){
  tem_vec<-c(d0[i],d1[i],d2[i])
  d[,i]<-tem_vec[geno[,i]+1]
}

dmat<-tcrossprod(d)/sum((2*maf*(1-maf))^2)

## run BGLR
nIter=12000
burnIn=2000
saveAt='GBLUPandDominance'

# Setting the linear predictor

ETA<-list( list(~factor(sex),
                model='FIXED'),
           list(K=gmat, model='RKHS'),
           list(K=dmat, model='RKHS')
)

fm4<-BGLR(y=y,ETA=ETA,nIter=nIter, burnIn=burnIn,saveAt=saveAt,verbose = F)
gebv<-fm4$ETA[[2]]$u[fm4$whichNa]+fm4$ETA[[3]]$u[fm4$whichNa]

y_test<-dat_test$phen-fm4$mu-fm4$ETA[[1]]$b*sex_test
plot(gebv~y_test,xlab='Phenotype',
     ylab='Predicted Genomic Value',col=2,
     xlim=tmp,ylim=tmp)
```



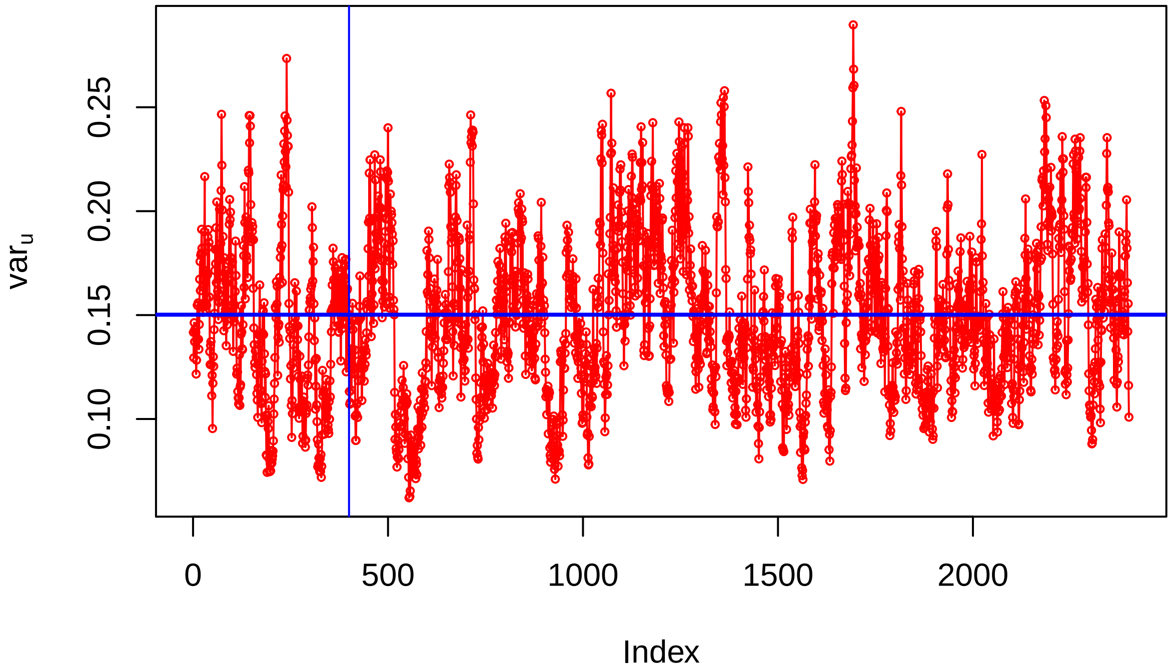
```
cor(gebv,y_test) #0.3371237
```

```
## [1] 0.2173347
```

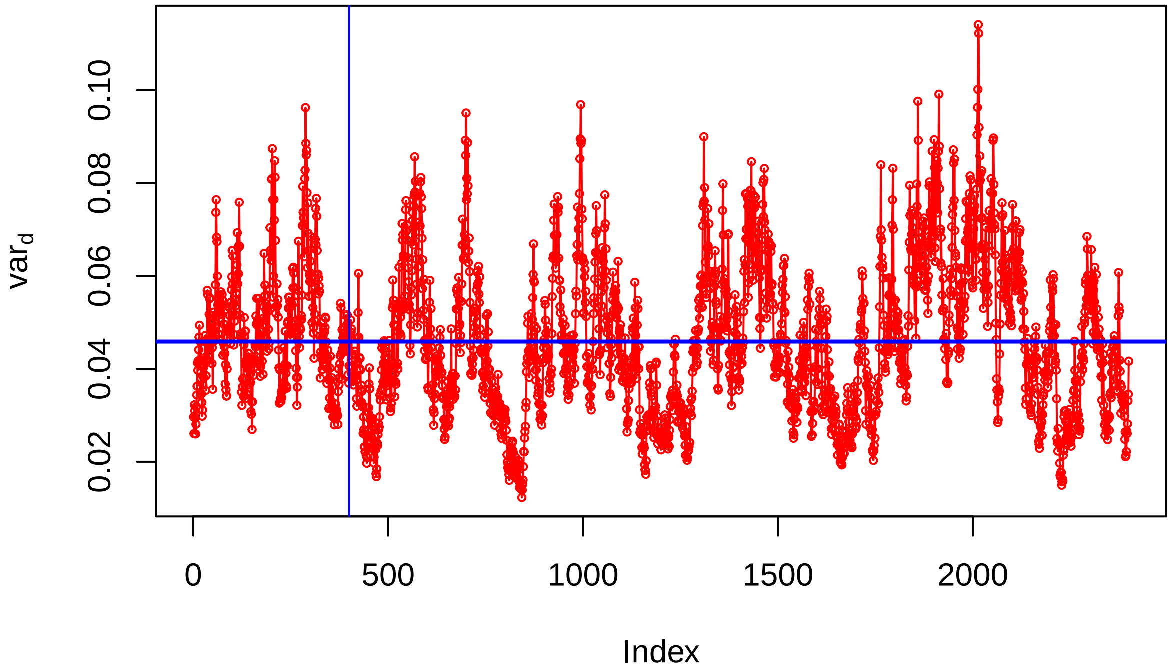
```
varE<-scan(paste0(saveAt,'varE.dat'))
plot(varE,type='o',col=2,cex=.5,ylab=expression(var[e]));
abline(h=fm4$varE,col=4,lwd=2);
abline(v=fm4$burnIn/fm4$thin,col=4)
```



```
varu<-scan(paste0(saveAt,'ETA_2_varU.dat'))
plot(varu,type='o',col=2,cex=.5,ylab=expression(var[u]));
abline(h=fm4$ETA[[2]]$varU,col=4,lwd=2);
abline(v=fm4$burnIn/fm4$thin,col=4)
```



```
vard<-scan(paste0(saveAt,'ETA_3_varU.dat'))
plot(vard,type='o',col=2,cex=.5,ylab=expression(var[d]));
abline(h=fm4$ETA[[3]]$varU,col=4,lwd=2);
abline(v=fm4$burnIn/fm4$thin,col=4)
```


0.02
0.04
0.06
0.08
0.10
var_d
0
500
1000
1500
2000
Index