The Bayestrat User's Guide

Contents

1 Introduction

The Bayestrat package is designed for Bayesian genetic association tests which identifies single-nucleotide polymorphisms (SNPs) associated with an outcome while using principle components (PCs) to account for population stratification. Because the sufficient number of PCs needed is unknown a priori, Bayestrat supports a large number of PCs to fully correct for the underlying population structures. Utilizing shrinkage priors, Bayestrat is able to filter out the irrelevant information, and select the highly confounded PCs. Bayestrat can be viewed as a compromise between the principle component regression of adding a few top PCs and the linear mixed model (LMM) which essentially includes all PCs. By shrinking the effects of irrelevant information, Bayestrat is able to achieve low type I error and high power.

The current implementation supports continuous and discrete covariates and continuous only outcome variables (forthcoming for discrete outcomes). Bayestrat first computes PCs based on a null data set (if PCs are not provided), then conducts association analysis for each SNP on the testing data set. Bayestrat provides inferences for effect sizes of SNPs, PCs, and other covariates based on the Markov Chain Monte Carlo (MCMC) samples from the posterior distribution. Individuals can be clustered based on PC scores and the results can be visualized in a 3-dimensional plot. Bayestrat also supports parallel computing with multiple cores.

2 Methods

2.1 The Regression Model

Let $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^T$ be a quantitative trait vector for n subjects, and $\boldsymbol{g} = (g_1, g_2, \cdots, g_n)^T$ be a vector of genotype scores, where $g_i, i = 1, 2, \cdots, n$, represents the number of minor alleles individual i has on a specific SNP with possible values of 0, 1 and 2. Let $\boldsymbol{Z} = (\boldsymbol{Z_1}, \boldsymbol{Z_2}, \cdots, \boldsymbol{Z_n})^T$ be a matrix of non-SNP covariates to be accounted for other than PCs. Let X be an $n \times K$ matrix of genotype scores for n individuals on K null markers. If PCs are not provided, Bayestrat uses functions in R or calls PLINK to calculate PCs based on the null data X. Suppose there are a total of L_{bs} PCs to be added to the model, and let $\boldsymbol{R} = (\boldsymbol{R_1}, \boldsymbol{R_2}, \cdots, \boldsymbol{R_n})^T$ be an $n \times L_{bs}$ matrix of PC scores for the corresponding L_{bs} PCs.

To achieve comparability among variables, standardization is performed on the outcome, PCs and other non-SNP covariates before analysis. Specifically, y_i is standardized by $y_i = \frac{y_i - \bar{y}}{sd(\boldsymbol{y})}$, where $\bar{y}$ is the mean of the outcome and $sd(\boldsymbol{y})$ is the standard deviation of the outcome. PCs are standardized to be comparable with SNP by $R_{il} = \frac{R_{il} - \bar{R}_{\cdot l}}{sd(R_{\cdot l})}\sqrt{\hat{p}(1-\hat{p})}$, $i = 1, \cdots n$, and $l = 1, \cdots L_{bs}$, where $\bar{R}_{\cdot l} = \frac{1}{n}\sum_{i=1}^{n} R_{il}$, $sd(R_{\cdot l}) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(R_{il} - \bar{R}_{\cdot l})^2}$, and $\hat{p}$ is the estimated minor allele frequency.

The analysis model is:

$$y_i = \mu + \boldsymbol{Z_i}^T\boldsymbol{\alpha} + g_i\beta + \sum_{l=1}^{L_{bs}} R_{il}\gamma_l + \epsilon_i, \text{ for } i = 1, 2, \cdots, n,$$

where μ is the intercept, $\boldsymbol{\alpha}$, β and $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_L)$ are the effects of covariates, the genetic variant, and PCs, respectively, and ϵ_i represents an random error term.

2.2 Prior Distributions and Inferences

We consider assigning μ, $\boldsymbol{\alpha}$ and β with independent and identically distributed (i.i.d) Uniform priors on the real line. ϵ_i's are assigned with i.i.d normal priors centered at zero, i.e. $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \cdots, \epsilon_n) \sim N(\boldsymbol{0}, \sigma_e^2\boldsymbol{I}_n)$ and $\boldsymbol{I}_n$ is the identity matrix of size n. σ_e^2 is assigned the conjugate Inverse-Gamma prior $IG(a_e, b_e)$ with probability density function (PDF): $\pi(\sigma_e^2 \mid a_e, b_e) = \frac{b_e^{a_e}}{\Gamma(a_e)}(\sigma_e^2)^{-a_e-1}\exp(-\frac{b_e}{\sigma_e^2}), 0 < \sigma_e^2 < \infty$.

To fully account for the underlying genetic relationships among individuals, a relatively large number of PCs are included in the model (instead of, for example, only the top 5 PCs). Because noisy information might be introduced by irrelevant PCs, we apply a penalization scheme on the PCs to shrink the effects of irrelevant PCs and boost the signal of highly confounded ones. Specifically, we impose a Laplace prior for each γ_l,

$$\pi(\gamma_l \mid \lambda, \sqrt{\sigma_e^2}) = \frac{\lambda}{2\sqrt{\sigma_e^2}}\exp(-\frac{\lambda}{\sqrt{\sigma_e^2}}|\gamma_l|)), \quad -\infty < \gamma_l < \infty, \quad l = 1, 2, \cdots, L_{bs},$$

where $\lambda > 0$ is a hyperparameter contributing to the degree of shrinkage. We let λ^2 follow $Gamma(a_\lambda, b_\lambda)$ with density function $\pi(\lambda^2) = \frac{b_\lambda^{a_\lambda}}{\Gamma(a_\lambda)}\lambda^{2(a_\lambda-1)}\exp(-\lambda^2 b_\lambda), 0 < \lambda^2 < \infty$.

Bayestrat also implements normal priors for the PC effects for completeness. $\pi(\gamma_l \mid \sigma_\gamma^2) \overset{i.i.d}{\sim} N(0, \sigma_\gamma^2)$. σ_γ^2 is assigned with the $IG(a_\gamma, b_\gamma)$ prior.

The default setting is $a_e = 2$, $b_e = 1$, $a_\lambda = 1.4$, $b_\lambda = 0.4$ and $a_\gamma = b_\gamma = 1$. Users can specify other values depending on specific situations. MCMC algorithms are used to sample from the posterior distributions. The implemented MCMC algorithm is adapted from R package BLR. (Pérez et al. (2010), Campos et al. (2013)) Significance is judged by the credible intervals not including zero under the given confidence level α.

3 Downloading Bayestrat

The Bayestrat package can be downloaded from Github using the devtools (link here) R package.

```
library(devtools)
install_github("Zilu-Liu/Bayestrat@main")
```

If the dependencies are not automatically downloaded (especially for Windows users), please manually download these R packages as dependencies: flashpcaR (link here), data.table, rgl, coda.

4 Using Bayestrat

We demonstrate the usage of Bayestrat with simulated data contained in the package as an example.

```
library(Bayestrat)
```

4.1 Data Input

The test data set is a data frame consisting of n rows (n =number of individuals) and $1 + c + m$ columns (c =number of covariates, m =number of candidate SNPs). The first column contains the values of a continuous phenotype for each individual. The following c columns consist of c covariates (other than PCs) to be accounted for. The next m columns consist of genotype scores (0,1 or 2) for m candidate SNPs. If race information is included, the corresponding column should be named as "Race".

```
data(data.test)
dim(data.test)
#> [1] 200  54
data.test[1:5,1:7]
#>          Y Age Sex  Race SNP1 SNP2 SNP3
#> 1 9.350663  29   F White    1    2    0
#> 2 8.642237  48   F White    0    1    0
#> 3 7.639163  37   M White    0    1    0
#> 4 8.226608  33   M White    0    0    0
#> 5 8.121156  32   F White    0    1    0
```

4.2 PC Computation

Users may provide PC data set directly if it is available and thus do not require calculation from Bayestrat. The PC data set is a data frame or matrix consisting of individuals as rows and PCs as columns.

```
data(pc)
dim(pc)
#> [1] 200 200
pc[1:5,1:5]
#>          PC1       PC2       PC3       PC4       PC5
#> 1 -12.024503 0.3583104 -5.547448  3.709040 -2.455794
#> 2  -4.947947 6.1862095 -5.646706  1.982633 -3.039221
#> 3  -7.818385 7.6065218 -6.986831  8.523370  8.569974
#> 4  -9.536486 4.3974174 -2.451673 12.685466  3.041754
#> 5  -9.597012 3.4612920 -4.703284 -2.466677 -5.732877
```

If a PC data set is not given, Bayestrat calculates PCs either internally or calls PLINK externally, depending on the size of data and the number of PCs required.

4.2.1 Internal PC Computation

Users need to provide the null data from which PCs will be calculated. The null data set should be in data frame or matrix format, consisting of n rows and K columns (K =number of null SNPs) representing genotype scores (0,1 or 2).

```
data(data.null)
dim(data.null)
#> [1]  200 5000
data.null[1:5,1:5]
#>      SNP51 SNP52 SNP53 SNP54 SNP55
#> [1,]     2     0     1     0     1
#> [2,]     0     1     0     2     1
#> [3,]     0     1     0     1     0
#> [4,]     1     0     1     2     1
#> [5,]     0     1     0     0     2
```

SNPs with missing values or no sequence variation will be deleted. Therefore, users may pre-impute the missing data using any imputation software before using Bayestrat. The internal computation (set $plink = F$) is appropriate in the following two cases:

1. $n.pc < min(n, K)/2$ ($n.pc$: the number of PCs added to model; n: the number of individuals; K: the number of null SNPs). In this case, Bayestrat utilizes the flashpca R package (link here) to calculate PCs, which is designed for fast PCA computation. (Abraham, Qiu, and Inouye (2017))
2. $n \times K \leq 10^7$. In this case, Bayestrat uses *prcomp* (Holland (2008)) to calculate PCs. The limination on the dimensions of data is needed such that the computational cost is acceptable.

4.2.2 External PC Computation

If the internal computation is not applicable, Bayestrat calls PLINK (Purcell et al. (2007)) to perform fast PC calculation for large data (set $plink = T$). Under the current working directory, PLINK1.9 (download here) needs to be installed, PED and MAP (introduction here) files for the null data are required. See section Plink Users for the details of generating PED and MAP files. The PED and MAP files should share a common filename prefix as an argument input to Bayestrat. For example, in the example data they are named as "data.null.ped" and "data.null.map", which will then be called by PLINK via the name prefix "data.null".

4.3 Running Bayestrat

Individuals with missing values will be deleted in the analysis. Bayestrat is equipped with Laplace and normal priors for PC effects. The defalut is the shrinkage Laplace prior. Below is an example of running Bayestrat with Laplace priors and calculating PCs by calling PLINK on a 2.3 GHz Intel Core i5 processor. Since PC score is used as a measure for population stratification, race information will not be inlcuded in the model by setting $race = F$ and $n.cov = 2$.

```
setwd("~/Downloads/plink1.9")
out1<-bayestrat(data.test,data.null=NULL,data.pc=NULL,race=F,plink=T,
                                data.null.name="data.null",n.pc=40,n.cov=2,priorType="Laplace",
                                priorError=c(2,1),priorLaplace=list(type="random",G=c(1.01,0.01),

                                priorNormal=list(type="random",IG=c(1,1),startvarPC=NULL),
                                nIter=5000,burnIn=500,minAbsBeta=1e-09,alpha=c(0.95,0.999,0.9999),
                                n.core=1,save.pc=F,checkConvergence=F,plot.interval=NULL,n.chains=1,
                                check.whichPC=c(1,2,3),check.whichSNP=1,seed=30)
#> Column Race deleted
```

```
#> Start PC calculation using PLINK
#> End PC calculation
#> Start testing
#> End testing
#> Times elapsed 37.152
```

Example of running Bayestrat by calculating PCs internally and using Laplace priors.

```
out2<-bayestrat(data.test,data.null,race=F,plink=F,n.pc=40,n.cov=2,
                                priorLaplace=list(type="random",G=c(1.01,0.01),startlambda=NULL),
                                nIter=5000,burnIn=500,save.pc=F,seed=30)
#> Column Race deleted
#> Start PC calculation using flashpca
#> End PC calculation
#> Start testing
#> End testing
#> Times elapsed 35.66
```

Example of running Bayestrat with given PC data and normal priors.

```
out3<-bayestrat(data.test,data.pc=pc,race=F,n.pc=40,n.cov=2,priorType="Normal",
                                nIter=5000,burnIn=500,seed=30)
#> Column Race deleted
#> Start testing
#> End testing
#> Times elapsed 23.717
```

Example of running Bayestrat without PCs, but with three covariates including race.

```
out4<-bayestrat(data.test,race=T,n.pc=0,n.cov=3,nIter=5000,burnIn=500,seed=30)
#> No PC added to model
#> Start testing
#> End testing
#> Times elapsed 10.458
```

Example of running Bayestrat with no adjustment for population stratification.

```
out5<-bayestrat(data.test[,-4],race=F,n.pc=0,n.cov=2,nIter=5000,burnIn=500,seed=30)
#> No PC added to model
#> Start testing
#> End testing
#> Times elapsed 9.784
```

4.4 Summarizing the Output

The output from *Bayestrat* is a list consisting of the sample sizes for testing each SNP (n), the estimated (posterior means) μ and σ_e^2 (varE), credible intervals and standard errors for SNP, covariate and PC effects, the estimate and standard error for λ (if $priorType =$ "$Laplace$") or σ_γ^2 (i.e. $varPC$, if $priorType =$ "$Normal$"). Except credible intervals and standard errors for SNP, other estimates are averaged over tesing all SNPs.

```
names(out2)
#>  [1] "n"        "mu"       "varE"     "CIsnp"    "SDsnp"    "CIcov"
#>  [7] "SDcov"    "CIpc"     "SDpc"     "lambda"   "SDlambda"
```

The function *bayestratSummary* summarizes the output from *Bayestrat*, and returns a list consisting of the estimated scaled effect sizes (raw effect size/standard error), raw effect sizes, standard errors and confidence

levels for significant SNPs, covariates and PCs.

```
result<-bayestratSummary(out2) #40PCs
names(result)
#> [1] "snp" "cov" "pc"
result$snp
#>       ScaledEffect  EffectSize         SD  alpha
#> SNP20     3.926991  0.25036508 0.06375494 0.9999
#> SNP21     3.262971  0.17231436 0.05280904 0.9990
#> SNP7      2.885031  0.15618219 0.05413536 0.9500
#> SNP24     2.822452  0.13055503 0.04625590 0.9500
#> SNP23     2.450208  0.18769826 0.07660502 0.9500
#> SNP8      2.405013  0.13368307 0.05558517 0.9500
#> SNP1      2.317339  0.14673192 0.06331916 0.9500
#> SNP10     2.279464  0.27402740 0.12021574 0.9500
#> SNP39    -2.092860 -0.09603291 0.04588596 0.9500
#> SNP12     2.073033  0.12067462 0.05821162 0.9500
#> SNP16     2.071658  0.09940434 0.04798298 0.9500
#> SNP5      2.067379  0.12212675 0.05907322 0.9500
```

The example data was generated such that the first 25 SNPs have some causal effects on the outcome, whereas the rest SNPs have no effect. Without adjustment for population stratification, there were 7 true positives and 4 false positives. After adjustment using 40 PCs, there were 11 true positives and 1 false positive. Notice that in this example, using race as an adjustment instead of PCs performs the best due to the data generation procedure, which detects 14 true positives and 0 false positive. In real cases, PCs may outperform races and are suggested as a way to correct for population strtification. These results can be reproduced by setting $seed = 30$, but seeds shouldn't always be set in real data analysis.

```
bayestratSummary(out5)$snp #noAdj
#>       ScaledEffect EffectSize         SD alpha
#> SNP29     3.252200  0.3316552 0.10197875 0.999
#> SNP1      2.549344  0.3495639 0.13711919 0.950
#> SNP8      2.512149  0.2783063 0.11078415 0.950
#> SNP48     2.493848  0.5020119 0.20130012 0.950
#> SNP10     2.315449  0.5942721 0.25665525 0.950
#> SNP43    -2.228665 -0.3960070 0.17768798 0.950
#> SNP27     2.134932  0.2054217 0.09621933 0.950
#> SNP5      2.060143  0.2601738 0.12628921 0.950
#> SNP20     1.989218  0.2823036 0.14191688 0.950
#> SNP23     1.967432  0.3246675 0.16502101 0.950
#> SNP7      1.961140  0.2422982 0.12354966 0.950
bayestratSummary(out4)$snp #race
#>       ScaledEffect EffectSize         SD  alpha
#> SNP24     4.284180 0.16060102 0.03748699 0.9999
#> SNP21     3.916219 0.16185995 0.04133066 0.9990
#> SNP1      3.490384 0.18190162 0.05211508 0.9990
#> SNP25     3.214795 0.14205925 0.04418920 0.9990
#> SNP20     3.174043 0.17193790 0.05417000 0.9500
#> SNP8      3.151226 0.13108972 0.04159960 0.9500
#> SNP5      3.134664 0.14777425 0.04714198 0.9500
#> SNP4      3.091088 0.12439027 0.04024158 0.9500
#> SNP9      3.020272 0.21736930 0.07197010 0.9500
#> SNP10     2.772257 0.26089484 0.09410919 0.9500
#> SNP3      2.737176 0.18878436 0.06897050 0.9500
#> SNP7      2.638929 0.12005104 0.04549234 0.9500
```

```
#> SNP23     2.620168 0.16080164 0.06137074 0.9500
#> SNP12     2.059268 0.09729274 0.04724628 0.9500
```

4.5 Parallel Computation

Bayestrat supports parallelism by using multiple cores for SNP analyses. Simply set *n.core* to be the number of cores used. Parallelism is performed by the *foreach* package. (R. Calaway, Weston, and Calaway (2015))

```
out6<-bayestrat(data.test,data.null,plink=F,n.pc=40,n.cov=2,
                                priorLaplace=list(type="random",G=c(1.01,0.01),startlambda=NULL),
                                nIter=5000,burnIn=500,save.pc=F,n.core=2,seed=30)
#> Column Race deleted
#> Start PC calculation using flashpca
#> End PC calculation
#> Start testing
#> End testing
#> Times elapsed 20.376
#> Start summarizing results
#> End summarizing results
```

4.6 Convergence Diagnostics

By setting $checkConvergence = T$, convergence diagnostics will be performed using trace plots and the Raftrey diagnostic for one MCMC chain or the Gelman diagnostic for multiple MCMC chains. (Raftery and Lewis (1995),Brooks and Gelman (1998)) Numerical and graphical summaries will be reported under the current working directory.

```
setwd("~/Documents/Genetics/bayestratDraft")
bayestrat(data.test,data.pc=pc,n.pc=40,n.cov=2,nIter=5000,burnIn=500,checkConvergence=T,
                    n.chains=2,check.whichPC=c(1:4,seq(5,40,5)),check.whichSNP=1,seed=30)
#> Column Race deleted
#> Diagnostic results saved
```

The Raftery dependence factors and the Gelman reduction (shrink) factors are reported. Dependence factors greater than 5 or reduction (shrink) factors substantially above 1 often indicate lack of convergence. Trace plots will also be generated as graphical summaries, to evaluate convergence and mixing of multiple chains. If the sampler converges well, one chain should show one horizontal band without long-term trends or drifts, and multiple chains with different starting values should finally reach similar results. Figure 1 indicates the convergence for the example data is not problematic. The full diagnostic outputs are in section Appendix.

4.7 Data Clustering and Visualization

The function *pcCluster* is designed for data clustering and visualization. Individuals are clustered by the k-means algorithm based on PC scores. Users need to specify which PCs to be clustered on. If the race information is provided as a column in *data.test* with name "Race", a confusion matrix comparing races and the clustering results will also be provided by setting $race = T$.

```
data(pc)
out6<-pcCluster(data.test,data.pc=pc,race=T,cluster=T,cluster.whichPC=c(1,2,3),
                                n.cluster=4,iter.max=10,algorithm="Hartigan-Wong",plotPC=F,
                                plot.whichPC=c(1,2,3),colorby="clusters",colors=NULL)
out6$confusion.matrix
```

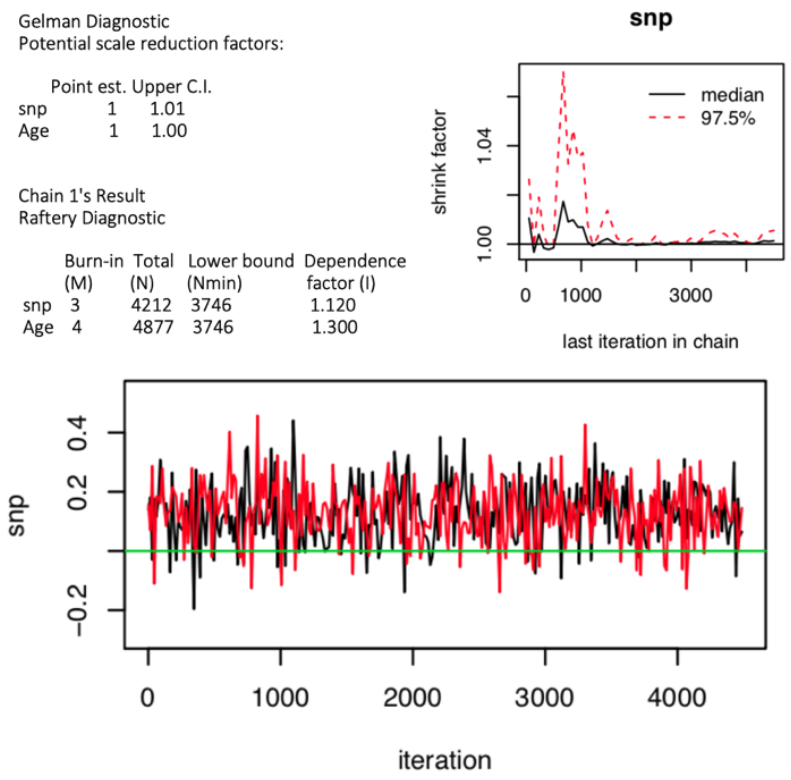


Figure 1: Diagnostics

```
#>           3  4  2  1 total
#> Asian    50  0  0  0    50
#> Black     0 50  0  0    50
#> Hispanic  0  0 50  0    50
#> White     0  0  0 50    50
#> total    50 50 50 50   200
```

PCs can be visualized by a 3-dimensional plot in the pop-up window (set $plotPC = T$). Users need to specify which three PCs to plot. The default PCs for clustering and plotting are the first three.

```
out7<-pcCluster(data.test,data.pc=pc,race=T,cluster=T,n.cluster=4,plotPC=T,
                                plot.whichPC=c(1,2,3),colorby="clusters")
```

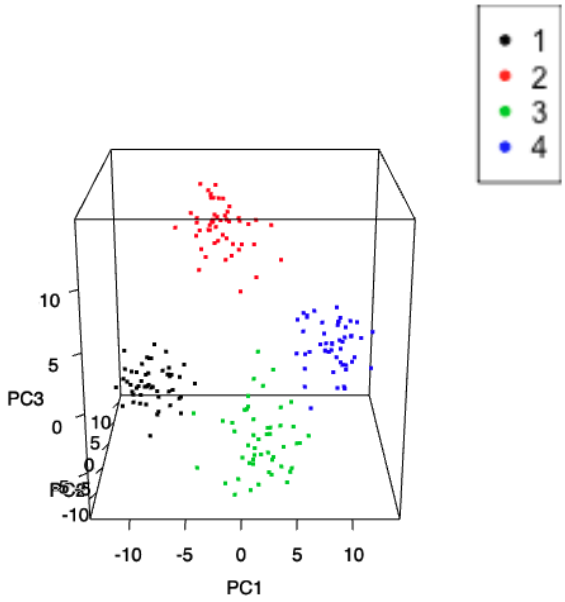


Figure 2: PC 3D plot

5 Plink Users

The PED file has n rows and $6 + 2 \times K$ columns. The first 6 columns consist of:

1. Family ID: A number indicates which family this individual belongs. Because Bayestrat is designed assuming independency, individuals should be unrelated and have different family IDs.
2. Individual ID: The unique identifier of this individual within each family.
3. Paternal ID: ID of the father of this individual. For Bayestrat, paternal ID should be 0 indicating founders.
4. Moternal ID: ID of the mother of this individual. For Bayestrat, maternal ID should be 0 indicating founders.
5. Sex: The gender of an individual. 1=male, 2=female and other=unknown.
6. Phenotype: The quantitative trait value of this individual.

Column 7 onwards describe the genotype information of the individual. All markers should be biallelic, i.e., each marker taking up 2 columns. The genotype data can be either alphabetic or numeric (e.g. 1,2,3,4 or A,C,G,T or anything else) except 0 which is, by default, the missing genotype character. Below is an example of creating a PED file based on 100 SNPs from the null data set.

```
data.null<-data.null[,1:100]
r<-nrow(data.null)
c<-ncol(data.null)
ped<-matrix(nrow=r,ncol=c*2+6)
ped<-as.data.frame(ped)
ped[,1]<-1:r
ped[,2]<-rep(1,r)
ped[,3]<-rep(0,r)
ped[,4]<-rep(0,r)
ped[,5]<-rep(1,r)
ped[,6]<-data.test$Y
for(i in 1:r){
    for(j in 1:c){
        if(data.null[i,j]=="0"){
            ped[i,2*j-1+6]<-"C" #C major allele, A minor allele
            ped[i,2*j+6]<-"C"
        }else if(data.null[i,j]=="1"){
            ped[i,2*j-1+6]<-"A"
            ped[i,2*j+6]<-"C"
        }else if(data.null[i,j]=="2"){
            ped[i,2*j-1+6]<-"A"
            ped[i,2*j+6]<-"A"
        }
    }
}
ped[1:5,1:10]
#>   V1 V2 V3 V4 V5       V6 V7 V8 V9 V10
#> 1  1  1  0  0  1 9.350663  A  A  C   C
#> 2  2  1  0  0  1 8.642237  C  C  A   C
#> 3  3  1  0  0  1 7.639163  C  C  A   C
#> 4  4  1  0  0  1 8.226608  A  C  C   C
#> 5  5  1  0  0  1 8.121156  C  C  A   C
write.table(ped,file="data.null.ped",row.names=F,col.names = F,quote=F)
```

The MAP file has K rows and 4 columns:

1. Chromosome: The chromosome for this SNP. Should be numbers from 1 to 22, X or Y.
2. SNP ID: Unique SNP identifier, such as rs number.
3. Genetic distance (morgans)
4. Base-pair position (bp units)

Below is an example of creating a MAP file.

```
map<-matrix(0,nrow=c,ncol=4)
map<-as.data.frame(map)
map[,1]<-rep("1",c)
name<-c()
for(i in 1:c){
    name<-c(name,paste("SNP",i,sep=""))
}
```

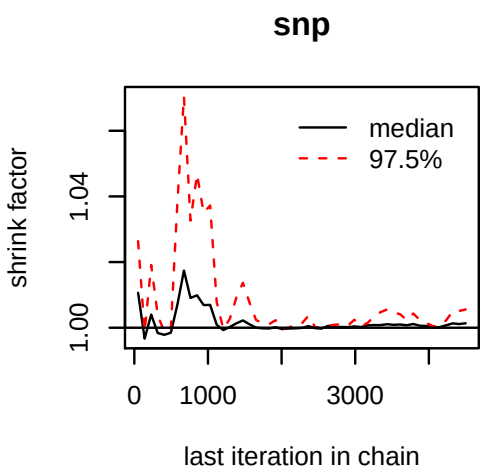
```r
map[,2]<-name
map[,3]<-rep(0,c)
map[,4]<-seq(5000,5990,10)
head(map)
#>   V1   V2 V3   V4
#> 1  1 SNP1  0 5000
#> 2  1 SNP2  0 5010
#> 3  1 SNP3  0 5020
#> 4  1 SNP4  0 5030
#> 5  1 SNP5  0 5040
#> 6  1 SNP6  0 5050
write.table(map,file="data.null.map",row.names=F,col.names = F,quote=F)
```

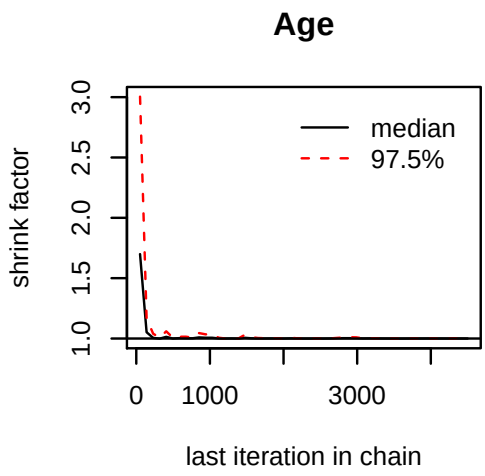
6 Appendix

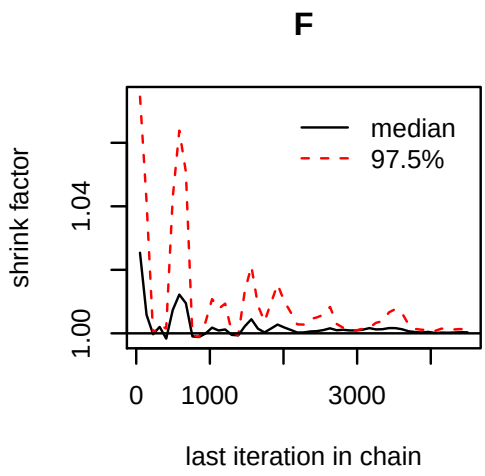
```
Gelman Diagnostic
Potential scale reduction factors:

       Point est. Upper C.I.
snp             1       1.01
Age             1       1.00
F               1       1.00
lambda          1       1.00
PC1             1       1.00
PC2             1       1.01
PC3             1       1.00
PC4             1       1.00
PC5             1       1.01
PC10            1       1.00
PC15            1       1.00
PC20            1       1.01
PC25            1       1.01
PC30            1       1.00
PC35            1       1.01
PC40            1       1.00

Multivariate psrf

1.01



Chain 1's Result
Raftery Diagnostic

Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

        Burn-in  Total Lower bound  Dependence
        (M)      (N)   (Nmin)       factor (I)
 snp    3        4212  3746         1.120
```

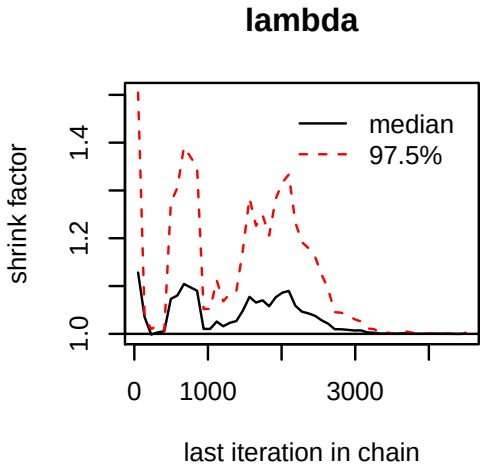
```
 Age    4        4877  3746         1.300
 F      3        4212  3746         1.120
 lambda 7        7921  3746         2.110
 PC1    3        4143  3746         1.110
 PC2    2        3916  3746         1.050
 PC3    2        3987  3746         1.060
 PC4    2        3987  3746         1.060
 PC5    2        3987  3746         1.060
 PC10   2        3845  3746         1.030
 PC15   2        3916  3746         1.050
 PC20   2        3845  3746         1.030
 PC25   2        3776  3746         1.010
 PC30   2        3845  3746         1.030
 PC35   2        3708  3746         0.990
 PC40   2        3641  3746         0.972

Acceptance Rate
   snp    Age      F lambda    PC1    PC2    PC3    PC4    PC5   PC10   PC15   PC20
     1      1      1      1      1      1      1      1      1      1      1      1
  PC25   PC30   PC35   PC40
     1      1      1      1



Chain 2's Result
Raftery Diagnostic

Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

        Burn-in  Total Lower bound  Dependence
        (M)      (N)   (Nmin)       factor (I)
 snp    2        3776  3746         1.010
 Age    5        5771  3746         1.540
 F      3        4531  3746         1.210
 lambda 7        7758  3746         2.070
 PC1    2        3987  3746         1.060
 PC2    2        3916  3746         1.050
 PC3    3        4061  3746         1.080
 PC4    2        3776  3746         1.010
 PC5    2        3745  3746         1.000
 PC10   2        3845  3746         1.030
 PC15   2        3576  3746         0.955
 PC20   2        3916  3746         1.050
 PC25   2        3845  3746         1.030
 PC30   2        3776  3746         1.010
 PC35   2        3883  3746         1.040
 PC40   2        3708  3746         0.990

Acceptance Rate
   snp    Age      F lambda    PC1    PC2    PC3    PC4    PC5   PC10   PC15   PC20
     1      1      1      1      1      1      1      1      1      1      1      1
  PC25   PC30   PC35   PC40
```

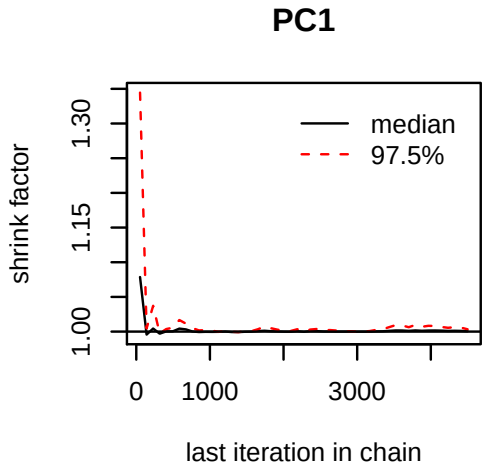
1 1 1 1

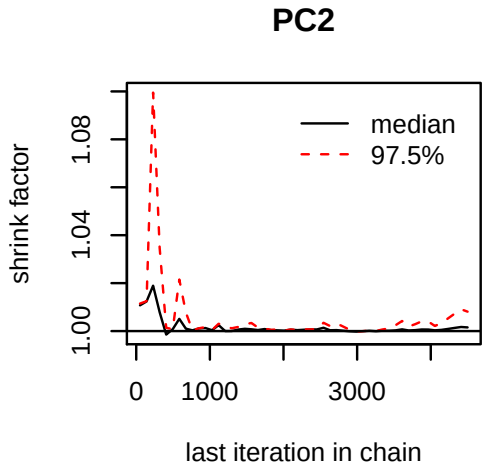

snp
median
97.5%
shrink factor
1.00
1.04
0
1000
3000
last iteration in chain

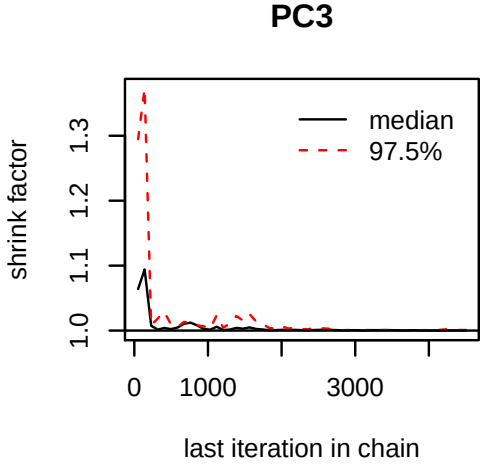

Age
median
97.5%
shrink factor
1.0
1.5
2.0
2.5
3.0
0
1000
3000
last iteration in chain

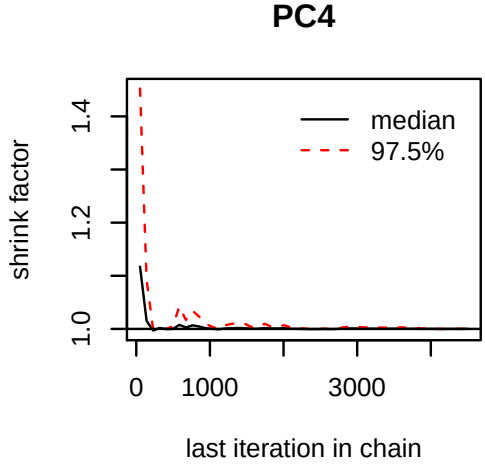

F
median
97.5%
shrink factor
1.00
1.04
0
1000
3000
last iteration in chain

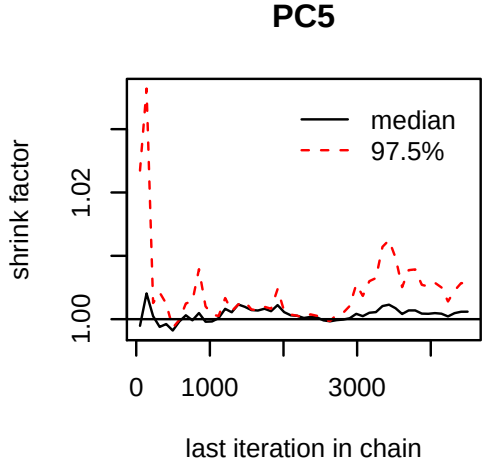

lambda
median
97.5%
shrink factor
1.0
1.2
1.4
0
1000
3000
last iteration in chain

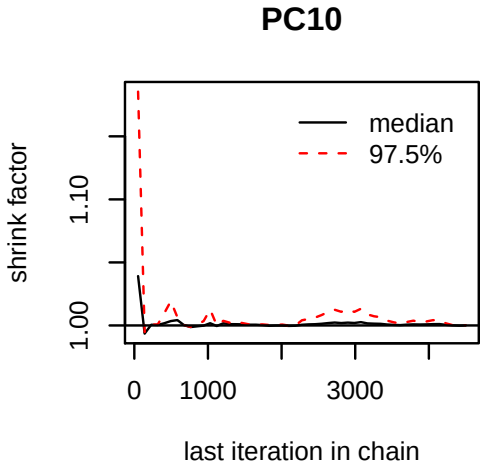

PC1
median
97.5%
shrink factor
1.00
1.15
1.30
0
1000
3000
last iteration in chain

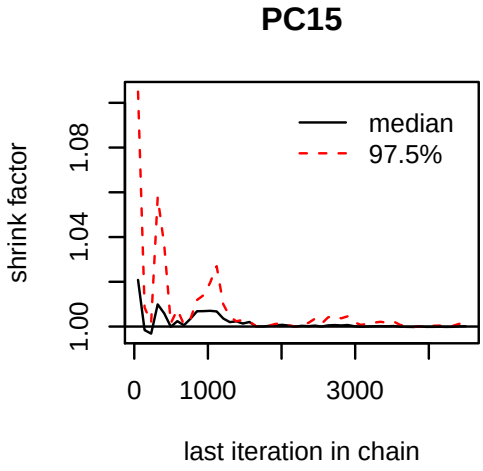

PC2
median
97.5%
shrink factor
1.00
1.04
1.08
0
1000
3000
last iteration in chain

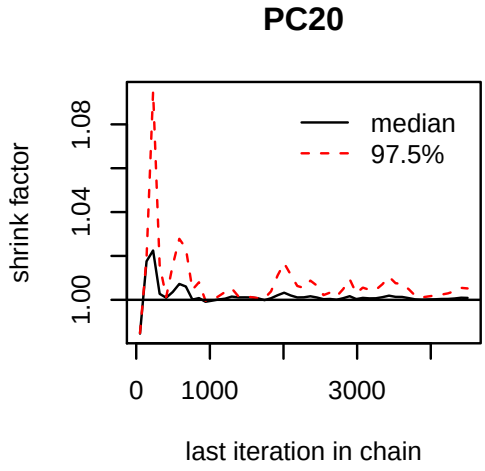

PC3
median
97.5%
shrink factor
1.0
1.1
1.2
1.3
0
1000
3000
last iteration in chain

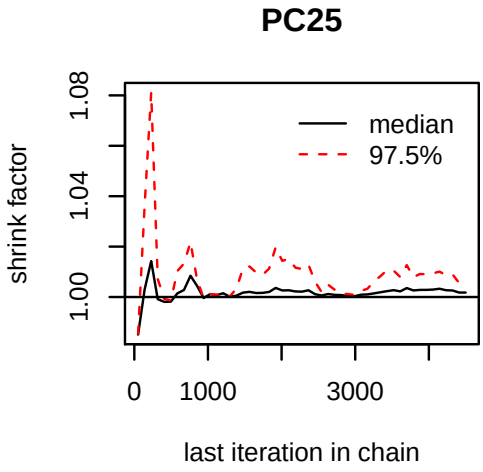

PC4
median
97.5%
shrink factor
1.0
1.2
1.4
0
1000
3000
last iteration in chain

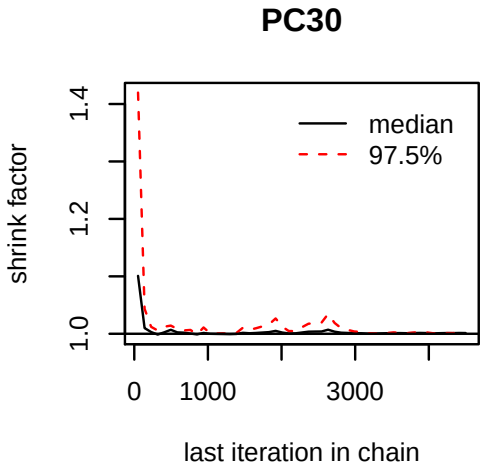

PC5
median
97.5%
shrink factor
1.00
1.02
0
1000
3000
last iteration in chain

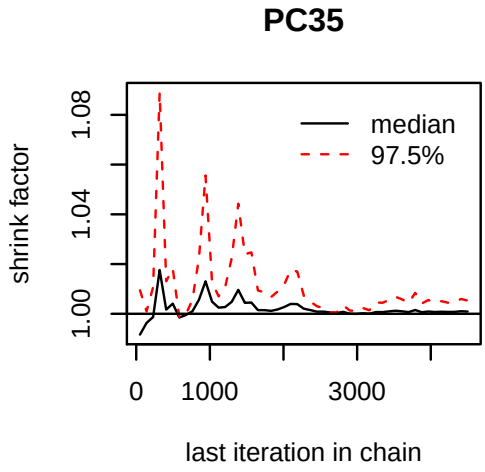

PC10
median
97.5%
shrink factor
1.00
1.10
0
1000
3000
last iteration in chain

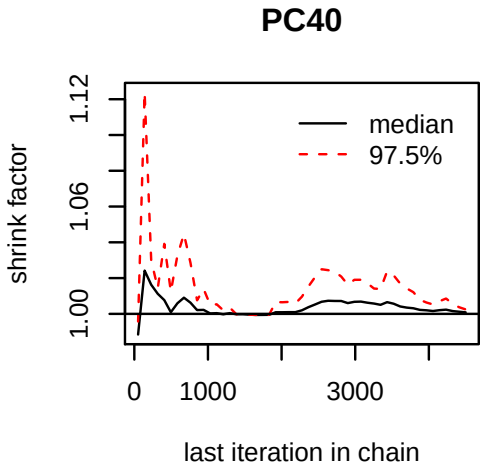

PC15
median
97.5%
shrink factor
1.00
1.04
1.08
0
1000
3000
last iteration in chain

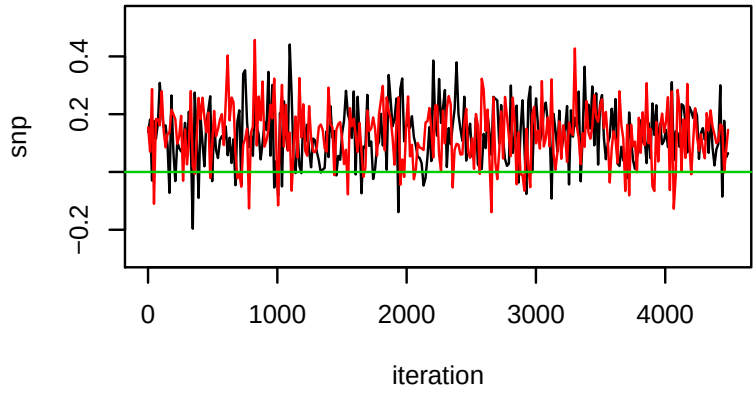

PC20
median
97.5%
shrink factor
1.00
1.04
1.08
0
1000
3000
last iteration in chain

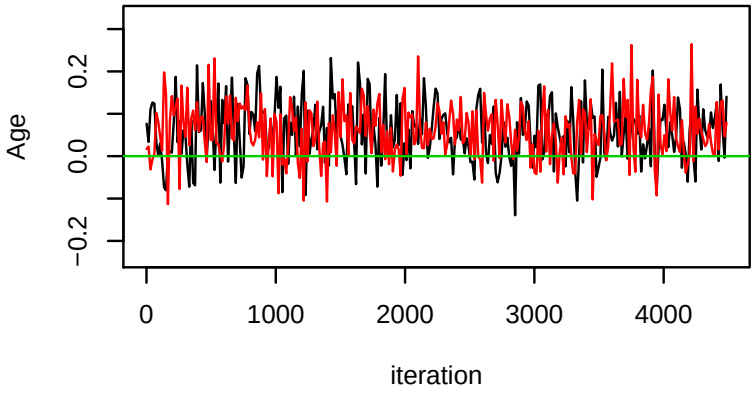

PC25
median
97.5%
shrink factor
1.00
1.04
1.08
0
1000
3000
last iteration in chain

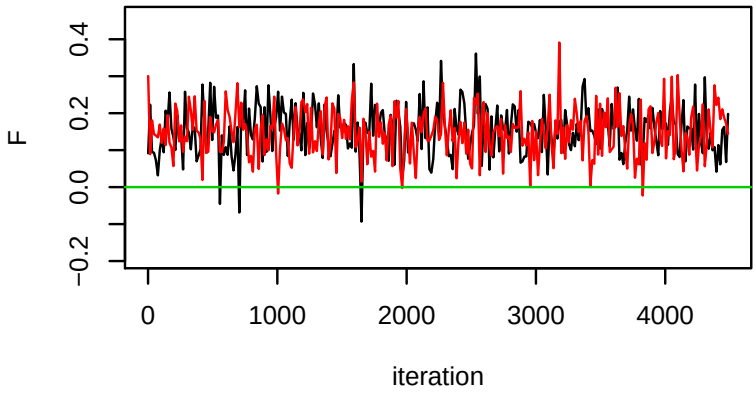

PC30
median
97.5%
shrink factor
1.0
1.2
1.4
0
1000
3000
last iteration in chain

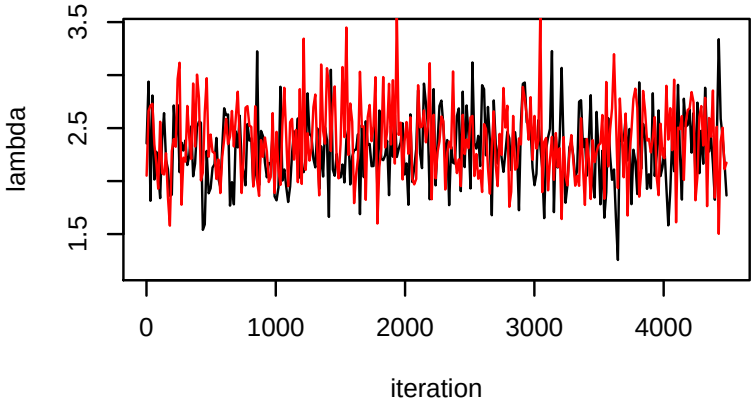

PC35
median
97.5%
shrink factor
1.00
1.04
1.08
0
1000
3000
last iteration in chain

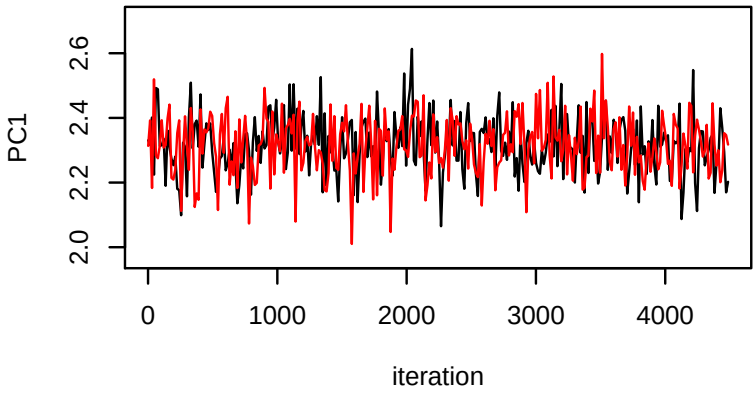

PC40
median
97.5%
shrink factor
1.00
1.06
1.12
0
1000
3000
last iteration in chain

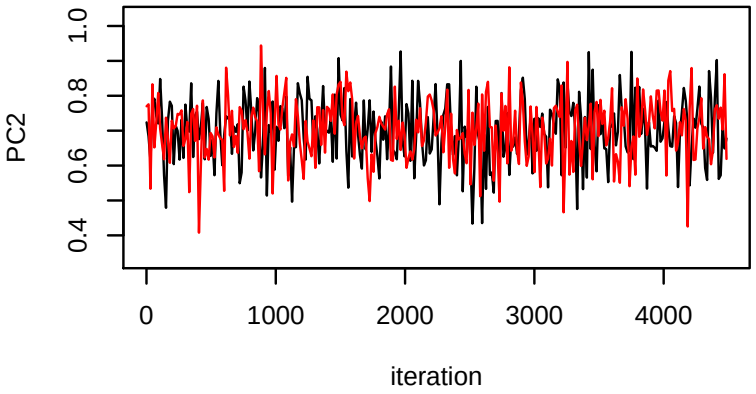

snp
−0.2
0.2
0.4
0
1000
2000
3000
4000
iteration

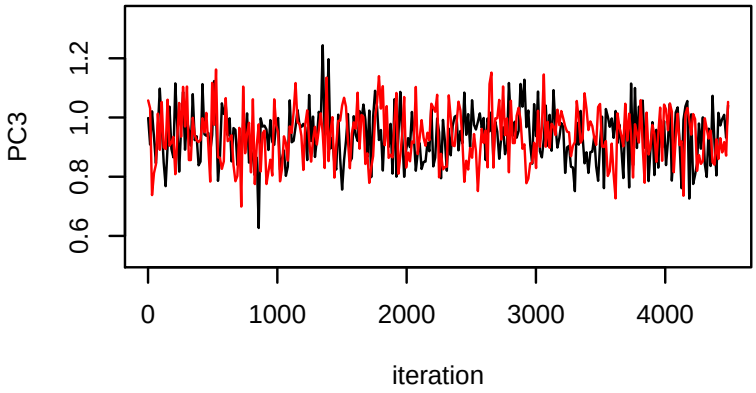

Age
−0.2
0.0
0.2
0
1000
2000
3000
4000
iteration

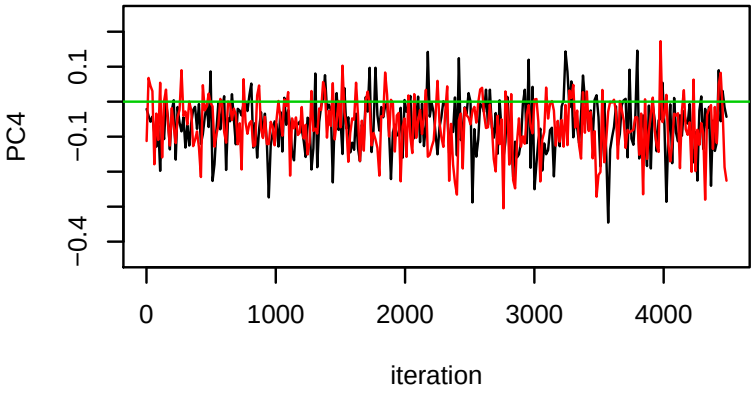

F
−0.2
0.0
0.2
0.4
0
1000
2000
3000
4000
iteration

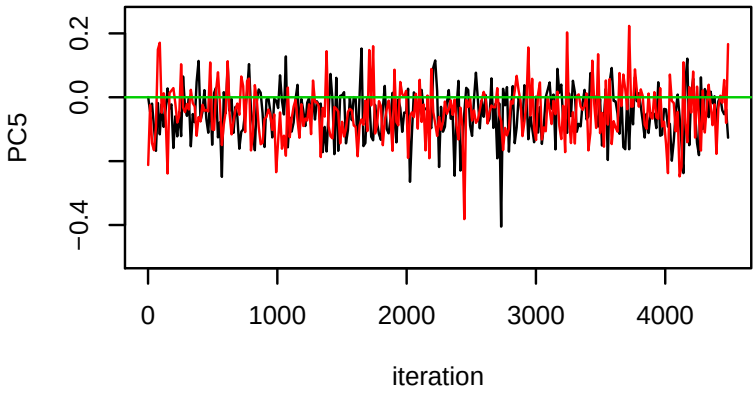

lambda
1.5
2.5
3.5
0
1000
2000
3000
4000
iteration

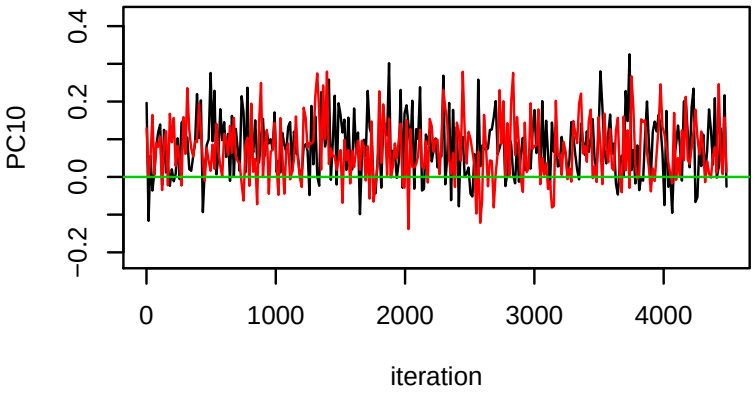

PC1
2.0
2.2
2.4
2.6
0
1000
2000
3000
4000
iteration

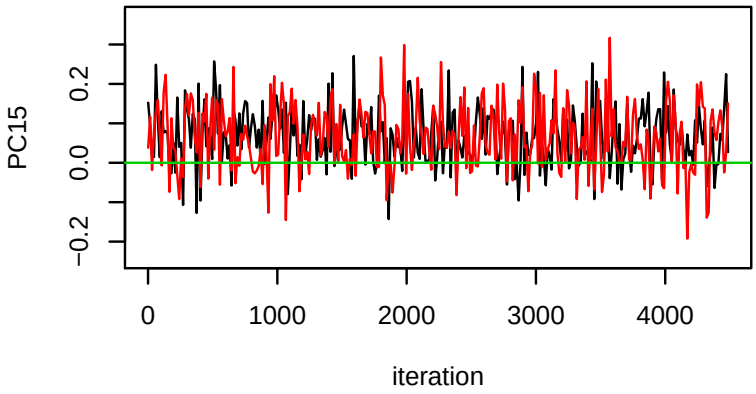

PC2
0.4
0.6
0.8
1.0
0
1000
2000
3000
4000
iteration

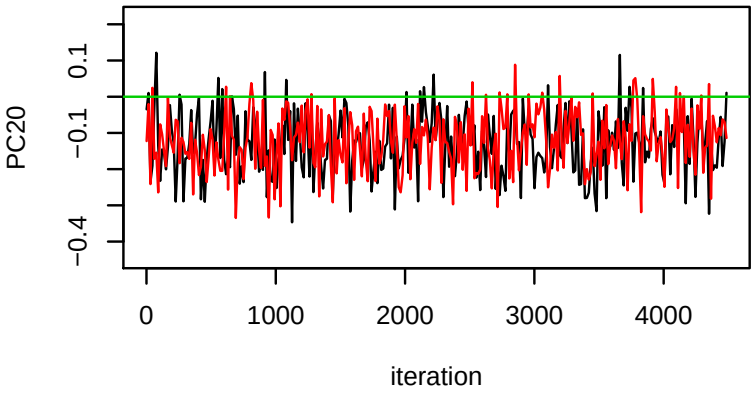

PC3
0.6
0.8
1.0
1.2
0
1000
2000
3000
4000
iteration

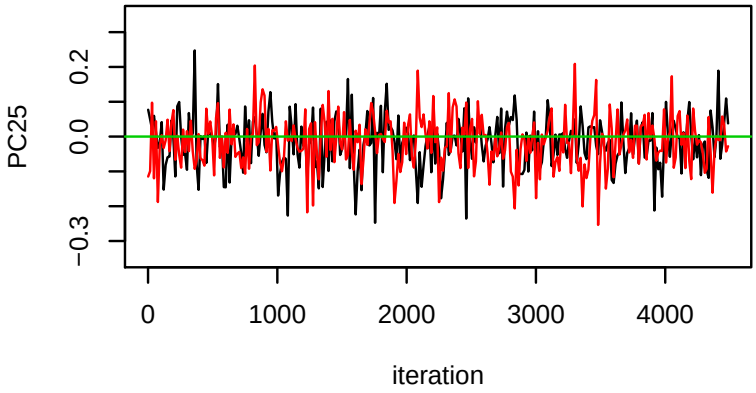

PC4
−0.4
−0.1
0.1
0
1000
2000
3000
4000
iteration

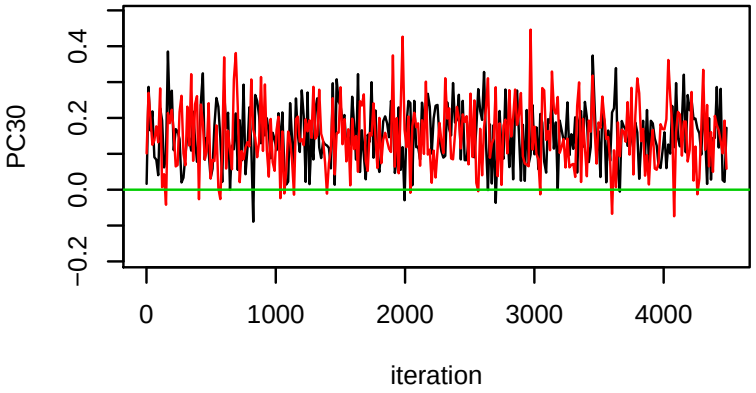

PC5
−0.4
0.0
0.2
0
1000
2000
3000
4000
iteration

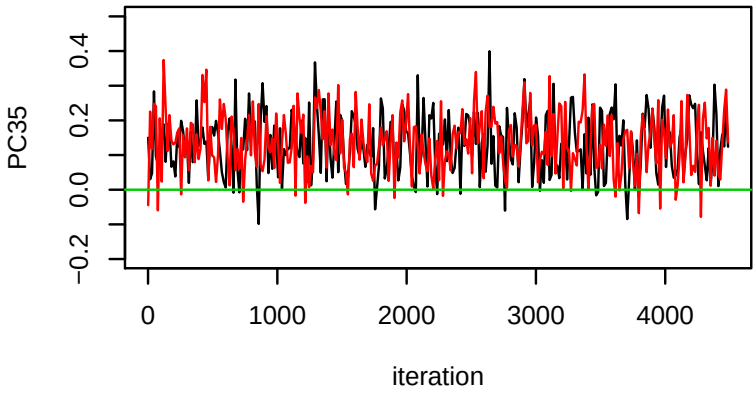

PC10
−0.2
0.0
0.2
0.4
0
1000
2000
3000
4000
iteration

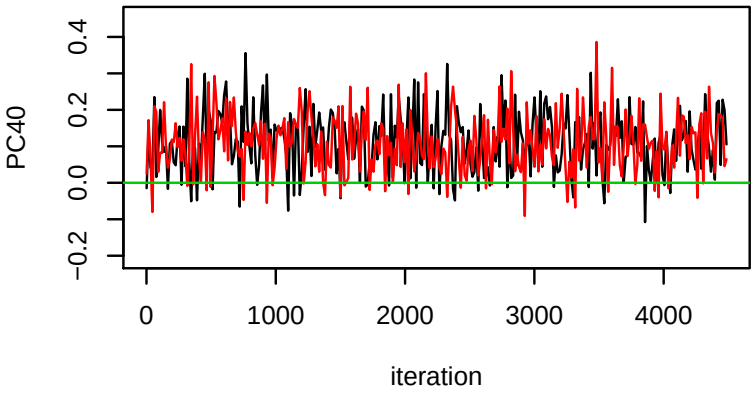

PC15
−0.2
0.0
0.2
0
1000
2000
3000
4000
iteration


PC20
−0.4
−0.1
0.1
0
1000
2000
3000
4000
iteration


PC25
−0.3
0.0
0.2
0
1000
2000
3000
4000
iteration


PC30
−0.2
0.0
0.2
0.4
0
1000
2000
3000
4000
iteration


PC35
−0.2
0.0
0.2
0.4
0
1000
2000
3000
4000
iteration


PC40
−0.2
0.0
0.2
0.4
0
1000
2000
3000
4000
iteration

References

Abraham, Gad, Yixuan Qiu, and Michael Inouye. 2017. "FlashPCA2: Principal Component Analysis of Biobank-Scale Genotype Datasets." *Bioinformatics*.

Brooks, Stephen P, and Andrew Gelman. 1998. "General Methods for Monitoring Convergence of Iterative Simulations." *Journal of Computational and Graphical Statistics* 7 (4). Taylor & Francis Group: 434–55.

Calaway, Rich, Steve Weston, and Maintainer Rich Calaway. 2015. "Package ?Foreach?" *R Package*, 1–10.

Campos, Gustavo de los, Paulino Perez, Ana I Vazquez, and Jose Crossa. 2013. "Genome-Enabled Prediction Using the Blr (Bayesian Linear Regression) R-Package." In *Genome-Wide Association Studies and Genomic Prediction*, 299–320. Springer.

Holland, Steven M. 2008. "Principal Components Analysis (Pca)." *Department of Geology, University of Georgia, Athens, GA*, 30602–32501.

Pérez, Paulino, Gustavo de los Campos, José Crossa, and Daniel Gianola. 2010. "Genomic-Enabled Prediction Based on Molecular Markers and Pedigree Using the Bayesian Linear Regression Package in R." *The Plant Genome* 3 (2). Wiley Online Library: 106–16.

Purcell, Shaun, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Pamela Maller Julian an d Sklar, Paul IW De Bakker, Mark J Daly, and others. 2007. "PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses." *The American Journal of Human Genetics* 81 (3). Elsevier: 559–75.

Raftery, Adrian E, and Steven M Lewis. 1995. "The Number of Iterations, Convergence Diagnostics and Generic Metropolis Algorithms." *Practical Markov Chain Monte Carlo* 7 (98). Citeseer: 763–73.