

# Package ‘BAMD’

February 14, 2012

**Type** Package

**Title** Bayesian Association Model for Genomic Data with Missing Covariates

**Version** 3.5

**Date** 2011-06-30

**Author** Gopal, V. and Li, Z. and Casella, G.

**Maintainer** Gopal, V. <viknesh@stat.ufl.edu>

**Depends** coda

**Description** This package fits a linear mixed model where the covariates for the random effects have missing values.

**SystemRequirements**

**Suggests** Rmpi

**License** GPL-2

**LazyLoad** yes

**URL** [http://www.stat.ufl.edu/~viknesh/assoc\\_model/assoc.html](http://www.stat.ufl.edu/~viknesh/assoc_model/assoc.html)

**Repository** CRAN

**Date/Publication** 2011-07-23 13:01:27

## R topics documented:

BAMD-package . . . . .	2
egMat . . . . .	3
gibbsSampler . . . . .	3
imputedGenotypes . . . . .	5
variableSelector . . . . .	7
variableSelectorBatchP . . . . .	8
variableSelectorInteractP . . . . .	10

<b>Index</b>	<b>12</b>
--------------	-----------

---

BAMD-package

*Bayesian Association Model for Genomic Data with Missing Covariates*

---

## Description

This package fits the following linear mixed model

$$Y = X\beta + Z\gamma + \epsilon$$

where the covariates for the random effects (in the Z-matrix) have missing values. The Z-matrix consists of Single Nucleotide Polymorphism (SNP) data and the Y-vector contains the phenotypic trait of interest. The X-matrix typically describes the family structure of the organisms.

## Details

Package:	BAMD
Type:	Package
Version:	3.5
Date:	2011-06-30
License:	GPL-2
LazyLoad:	yes

There are two functions in this package. The first, `gibbsSampler`, will fit and estimate the posterior parameters in the model above. This will allow the experimenter to pick out which covariates were significantly non-zero, since the routine will return  $(1 - \alpha)100\%$  confidence intervals. The imputed missing values at each iteration of the Gibbs sampler will be stored in a file for use by the second function.

The second function, `variableSelector`, is a variable selector that will pick out the “best” model, as measured by the Bayes Factor, using a stochastic search algorithm.

## Author(s)

Vik Gopal <viknesh@stat.ufl.edu>

Maintainer: Vik Gopal <viknesh@stat.ufl.edu>

## References

Gopal, V. "BAMD User Manual" [http://www.stat.ufl.edu/~viknesh/assoc\\_model/assoc.html](http://www.stat.ufl.edu/~viknesh/assoc_model/assoc.html)

## See Also

[gibbsSampler](#), [variableSelector](#)

---

egMat	<i>Example matrices for input to gibbsSampler() and variableSelector()</i>
-------	--

---

### Description

These are example datasets that demonstrate the format of input to the routines in BAMD.

### Usage

```
data(Y)
```

### Format

For this dataset, n=8, p=3 and s=5.

### References

Gopal, V. "BAMD User Manual" [http://www.stat.ufl.edu/~viknesh/assoc\\_model/assoc.html](http://www.stat.ufl.edu/~viknesh/assoc_model/assoc.html)

### Examples

```
data(Y,X,Z,R,Zprob)
write.csv(cbind(Y,X,Z,R), file="generatedData.csv", quote=FALSE, row.names=FALSE)
write.csv(Zprob, file="Zprob.csv", quote=FALSE, row.names=FALSE)
```

---

gibbsSampler	<i>Estimate posterior parameters in Bayesian Association Model</i>
--------------	--

---

### Description

This function fits the following linear mixed model

$$Y = X\beta + Z\gamma + \epsilon$$

where the covariates for the random effects (in the Z-matrix) have missing values. The Z-matrix consists of Single Nucleotide Polymorphism (SNP) data and the Y-vector contains the phenotypic trait of interest. The X-matrix typically describes the family structure of the organisms.

The model is fit by embedding it in a Bayesian framework and estimating the posterior parameters using a Gibbs sampler.

### Usage

```
gibbsSampler(fname, fprob, n, p, s, a = 2.01, b = 0.99099, c = 2.01, d = 0.99099,
beta.in = 1, gamma.in = 1, sig2.in = 1, phi2.in = 1, alpha = 0.05,
nsim = 1000, keep = 100, codaOut="CodaChain.txt",
codaIndex="CodaIndex.txt")
```

**Arguments**

fname	fname should be the name of a .csv file. This file should contain the Y, X, Z and R matrices for the model, in that particular order. Hence it should contain $n \times (1 + p + s + n)$ values. There should be a header row in the input file as well. The Z matrix should use the values 1,2,3 for the SNPs and 0 for any missing SNPs. The program will convert the SNP codings to -1,0,1 and work with those.
fprob	fprob should also be a .csv file. It should contain one probability vector for each entry in the Z-table. Hence it should be a matrix of dimension $n \times 3s$ . The program will read in the entire table, but only store the distributions corresponding to the missing values. If uniform priors are to be used, there is no need to specify anything.
n	n refers to the length of the Y-vector; equivalent to the number of observations in the dataset.
p	p is the number of columns of the X-matrix.
s	s is the number of columns of the Z-matrix.
a,b,c,d	ab,c,d are hyperparameters in the Bayesian set-up.
beta.in	beta.in is the initial value for the Gibbs sampler. It should be a vector of length p.
gamma.in	gamma.in is the initial value for the Gibbs sampler. It should be a vector of length s.
sig2.in	sig2.in is the initial value for the Gibbs sampler. It should be a vector of length 1.
phi2.in	phi2.in is the initial value for the Gibbs sampler. It should be a vector of length 1.
alpha	alpha refers to the $(1 - \alpha)100\%$ confidence intervals that the program should output.
nsim	nsim specifies the number of iterations of the Gibbs sampler to carry out.
keep	keep specifies which values from the Gibbs sampler chain to keep and use when computing the mean and confidence intervals. This allows user to allow for burn-in.
codaOut	File to store Gibbs sample parameters in coda format
codaIndex	File to store description of sampled and stored values.

**Details**

For further details on the prior distributions used, please refer to the User Guide in the reference(s) given below.

**Value**

There will be no R object returned. Instead, as the routine is running, it will print debugging statements to show the user which iteration of the Gibbs sampler it is currently at. This would allow the user to detect if something is going wrong with the routine.

The values sampled from the full conditionals will be stored in the files CodaChain.txt and Imputed\_missing\_vals by default.

**Author(s)**

Vik Gopal <viknesh@stat.ufl.edu>

Maintainer: Vik Gopal <viknesh@stat.ufl.edu>

**References**

Gopal, V. "BAMD User Manual" [http://www.stat.ufl.edu/~viknesh/assoc\\_model/assoc.html](http://www.stat.ufl.edu/~viknesh/assoc_model/assoc.html)

**See Also**

[variableSelector](#)

**Examples**

```
# Load example matrices and write to csv files.
data(Y, X, Z, R, Zprob)
write.csv(cbind(Y,X,Z,R), file="generatedData.csv", quote=FALSE, row.names=FALSE)
write.csv(Zprob, file="Zprob.csv", quote=FALSE, row.names=FALSE)

# Run the gibbs sampler with 100 iterations, keeping the last 800
gibbsSampler(fname="generatedData.csv", fprob="Zprob.csv", n=8, p=3, s=5, nsim=1000, keep=800)

#remove all generated csv files
unlink("*.txt")
unlink("*.csv")
```

---

imputedGenotypes

*Estimate posterior distribution of missing SNPs*

---

**Description**

After the Gibbs sampler has been run, this function retrieves the values that were imputed for the missing SNP values and prints out the frequency with which the particular values were sampled. Assuming the chain has reached stationarity (that sufficient burn-in has been discarded), this corresponds to the posterior distribution of the SNPs.

**Usage**

```
imputedGenotypes(fname, n, p, s, missingfile = "Imputed_missing_vals")
```

**Arguments**

**fname**            fname should be the name of a .csv file. This file should contain the Y, X, Z and R matrices for the model, in that particular order. Hence it should contain  $n \times (1 + p + s + n)$  values. There should be a header row in the input file as well. The Z matrix should use the values 1,2,3 for the SNPs and 0 for any missing SNPs. The program will convert the SNP codings to -1,0,1 and work with those.

n	n refers to the length of the Y-vector; equivalent to the number of observations in the dataset.
p	p is the number of columns of the X-matrix.
s	s is the number of columns of the Z-matrix.
missingfile	Contains the missing SNP values that were output from <a href="#">gibbsSampler</a> .

**Details**

This posterior distribution is different from the "average" Z-matrix that is used in the variable selector method. In that situation, we ignore the fact that at every iteration of the Gibbs sampler, only one column is updated and the majority of the Z-matrix remains the same.

In this situation, we consider a sample of the genotype to be new only every time it is updated, not every Gibbs sampler iteration. For example, if there were 2 columns with missing values and 7000 Gibbs sampler iterations were kept, only ~3500 samples would be taken for each column.

**Value**

A matrix specifying the posterior distribution for each missing SNP.

**Author(s)**

Vik Gopal <viknesh@stat.ufl.edu>

Maintainer: Vik Gopal <viknesh@stat.ufl.edu>

**References**

Gopal, V. "BAMD User Manual" [http://www.stat.ufl.edu/~viknesh/assoc\\_model/assoc.html](http://www.stat.ufl.edu/~viknesh/assoc_model/assoc.html)

**See Also**

[gibbsSampler](#)

**Examples**

```
# Load example matrices and write to csv files.
data(Y, X, Z, R, Zprob)
write.csv(cbind(Y,X,Z,R), file="generatedData.csv", quote=FALSE, row.names=FALSE)
write.csv(Zprob, file="Zprob.csv", quote=FALSE, row.names=FALSE)

# Run the gibbs sampler with 100 iterations, keeping the last 800
gibbsSampler(fname="generatedData.csv", fprob="Zprob.csv", n=8, p=3, s=5, nsim=1000, keep=800)
imputedGenotypes("generatedData.csv", n=8, p=3, s=5)

#remove all generated csv files
unlink("*.txt")
unlink("*.csv")
```

---

variableSelector      *Variable Selection in Bayesian Association Model*

---

### Description

This function carries out variable selection on the following linear mixed model

$$Y = X\beta + Z\gamma + \epsilon$$

where the covariates for the random effects (in the Z-matrix) have missing values. The Z-matrix consists of Single Nucleotide Polymorphism (SNP) data and the Y-vector contains the phenotypic trait of interest. The X-matrix typically describes the family structure of the organisms.

The best models are determined by their Bayes Factor, and uses the imputed values from the [gibbsSampler](#) function.

### Usage

```
variableSelector(fname, n, p, s, nsim, keep = 5, prop = 0.75,
  codaOut="CodaChain.txt", codaIndex="CodaIndex.txt",
  missingfile = "Imputed_missing_vals", SNPsubset)
```

### Arguments

fname	fname should be the name of a .csv file. This file should contain the Y, X, Z and R matrices for the model, in that particular order. Hence it should contain $n \times (1 + p + s + n)$ values. There should be a header row in the input file as well. The Z matrix should use the values 1,2,3 for the SNPs and 0 for any missing SNPs. The program will convert the SNP codings to -1,0,1 and work with those.
n	n refers to the length of the Y-vector; equivalent to the number of observations in the dataset.
p	p is the number of columns of the X-matrix.
s	s is the number of columns of the Z-matrix. Note that this is the total number of original SNPs put through the Gibbs sampler.
nsim	nsim specifies the number of iterations of the Metropolis-Hastings chain to carry out.
keep	keep specifies the number of models to store. The top keep models will be retained.
prop	As the candidate distribution for the Metropolis-Hastings chain is a mixture, one of whose components is a random walk, prop will determine the percentage of time that the random walk distribution is chosen.
codaOut	This is the name of the file that was output from <a href="#">gibbsSampler</a> . It contains the values obtained from the Gibbs sampler.
codaIndex	This is the name of the file that describes the format of the variables in codaOut.
missingfile	Contains the missing SNP values that were output from <a href="#">gibbsSampler</a> .
SNPsubset	A 0-1 vector of length s, indicating the SNPs that should be considered as possible variables.

**Details**

A Metropolis-Hastings algorithm is used to conduct a stochastic search through the model space to find the best models.

**Value**

A matrix consisting of the best keep models and their Bayes Factors is returned.

**Author(s)**

Vik Gopal <viknesh@stat.ufl.edu>

Maintainer: Vik Gopal <viknesh@stat.ufl.edu>

**References**

Gopal, V. "BAMD User Manual" [http://www.stat.ufl.edu/~viknesh/assoc\\_model/assoc.html](http://www.stat.ufl.edu/~viknesh/assoc_model/assoc.html)

**See Also**

[gibbsSampler](#)

**Examples**

```
# Load example matrices and write to csv files.
data(Y, X, Z, R, Zprob)
write.csv(cbind(Y,X,Z,R), file="generatedData.csv", quote=FALSE, row.names=FALSE)
write.csv(Zprob, file="Zprob.csv", quote=FALSE, row.names=FALSE)

# Run the gibbs sampler with 100 iterations, keeping the last 800
gibbsSampler(fname="generatedData.csv", fprob="Zprob.csv", n=8, p=3, s=5, nsim=1000, keep=800)

# Imputed values from gibbs sampler will be used in Variable Selector
variableSelector(fname="generatedData.csv", n=8, p=3, s=5, nsim=100, keep = 5)

#remove all generated csv files
unlink("*.csv")
unlink("*.txt")
```

---

variableSelectorBatchP

*Variable Selection in Parallel Batch Mode BAMD*

---

**Description**

This function runs [variableSelector](#) in parallel in Batch mode.

**Usage**

```
variableSelectorBatchP(fname, n, p, s, nsim, keep = 5, prop = 0.75,
  codaOut = "CodaChain.txt", codaIndex = "CodaIndex.txt",
  missingfile = "Imputed_missing_vals", SNPsubset, prefix,
  pathToLog, outfile = "out1.rdt")
```

**Arguments**

fname	fname should be the name of a .csv file. This file should contain the Y, X, Z and R matrices for the model, in that particular order. Hence it should contain $n \times (1 + p + s + n)$ values. There should be a header row in the input file as well. The Z matrix should use the values 1,2,3 for the SNPs and 0 for any missing SNPs. The program will convert the SNP codings to -1,0,1 and work with those.
n	n refers to the length of the Y-vector; equivalent to the number of observations in the dataset.
p	p is the number of columns of the X-matrix.
s	s is the number of columns of the Z-matrix. Note that this is the total number of original SNPs put through the Gibbs sampler.
nsim	nsim specifies the number of iterations of the Metropolis-Hastings chain to carry out.
keep	keep specifies the number of models to store. The top keep models will be retained.
prop	As the candidate distribution for the Metropolis-Hastings chain is a mixture, one of whose components is a random walk, prop will determine the percentage of time that the random walk distribution is chosen.
codaOut	This is the name of the file that was output from <a href="#">gibbsSampler</a> . It contains the values obtained from the Gibbs sampler.
codaIndex	This is the name of the file that describes the format of the variables in codaOut.
missingfile	Contains the missing SNP values that were output from <a href="#">gibbsSampler</a> .
SNPsubset	A 0-1 vector of length s, indicating the SNPs that should be considered as possible variables.
prefix	A prefix to name the log files from each processor. For example, if prefix is specified as "rank" and there are 3 processors, then there will be 3 files with names "rank00.log", "rank01.log" and "rank02.log"
pathToLog	A path to where the log files should be stored.
outfile	A character string - the file name to store the output table to.

**Details**

A Metropolis-Hastings algorithm is used to conduct a stochastic search through the model space to find the best models. nsim steps of the chain will be run on each available processor. Each of them will return the best keep models they found to the master. The master will strip away the duplicates and return the top keep models found.

See the scripts in demo/ directory for full examples.

**Value**

No value is returned as it is run in Batch mode. The output object is stored in the binary output file.

**Note**

Remember to copy the appropriate Rprofile that is provided in the inst/ directory to the directory to you are working in!

**Author(s)**

Vik Gopal <viknesh@stat.ufl.edu>

**See Also**

[variableSelector](#), [variableSelectorInteractP](#)

---

variableSelectorInteractP

*Variable Selection in Parallel Interactive Mode BAMD*

---

**Description**

This function runs [variableSelector](#) in parallel in an interactive R session.

**Usage**

```
variableSelectorInteractP(fname, n, p, s, nsim, keep, prop,
  codaOut, codaIndex, missingfile, SNPsubset)
```

**Arguments**

fname	fname should be the name of a .csv file. This file should contain the Y, X, Z and R matrices for the model, in that particular order. Hence it should contain $n \times (1 + p + s + n)$ values. There should be a header row in the input file as well. The Z matrix should use the values 1,2,3 for the SNPs and 0 for any missing SNPs. The program will convert the SNP codings to -1,0,1 and work with those.
n	n refers to the length of the Y-vector; equivalent to the number of observations in the dataset.
p	p is the number of columns of the X-matrix.
s	s is the number of columns of the Z-matrix. Note that this is the total number of original SNPs put through the Gibbs sampler.
nsim	nsim specifies the number of iterations of the Metropolis-Hastings chain to carry out.
keep	keep specifies the number of models to store. The top keep models will be retained.

prop	As the candidate distribution for the Metropolis-Hastings chain is a mixture, one of whose components is a random walk, prop will determine the percentage of time that the random walk distribution is chosen.
codaOut	This is the name of the file that was output from <a href="#">gibbsSampler</a> . It contains the values obtained from the Gibbs sampler.
codaIndex	This is the name of the file that describes the format of the variables in codaOut.
missingfile	Contains the missing SNP values that were output from <a href="#">gibbsSampler</a> .
SNPsubset	A 0-1 vector of length s, indicating the SNPs that should be considered as possible variables.

### Details

A Metropolis-Hastings algorithm is used to conduct a stochastic search through the model space to find the best models. nsim steps of the chain will be run on each slave. The master does not do anything except consolidate the models they return. Each slave will return the best keep models that it found to the master. The master will strip away the duplicates and return the top keep models found.

See the scripts in demo/ directory for full examples.

### Value

The consolidated table with the best keep models is returned.

### Note

Remember to copy the appropriate Rprofile that is provided in the inst/ directory to the directory to you are working in!

Also, only for this function, all arguments have to be specified, and named. Please see the script in demo/ for a full example.

### Author(s)

Vik Gopal <viknesh@stat.ufl.edu>

### See Also

[variableSelector](#), [variableSelectorBatchP](#)

# Index

\*Topic **datasets**

egMat, [3](#)

\*Topic **htest**

variableSelectorBatchP, [8](#)

variableSelectorInteractP, [10](#)

\*Topic **models**

gibbsSampler, [3](#)

imputedGenotypes, [5](#)

variableSelector, [7](#)

\*Topic **package**

BAMD-package, [2](#)

BAMD (BAMD-package), [2](#)

BAMD-package, [2](#)

egMat, [3](#)

gibbsSampler, [2](#), [3](#), [6–9](#), [11](#)

imputedGenotypes, [5](#)

R (egMat), [3](#)

variableSelector, [2](#), [5](#), [7](#), [8](#), [10](#), [11](#)

variableSelectorBatchP, [8](#), [11](#)

variableSelectorInteractP, [10](#), [10](#)

X (egMat), [3](#)

Y (egMat), [3](#)

Z (egMat), [3](#)

Zprob (egMat), [3](#)