# Package ‘mlm4omics’

June 15, 2019

**Version** 1.3.0

**Title** Multilevel Model for Multivariate Responses with Missing Values

**Description** To conduct Bayesian inference regression for responses with multilevel explanatory variables and missing values; It uses function from 'Stan', a software to implement posterior sampling using Hamiltonian MC and its variation Non-U-Turn algorithms. It implements the posterior sampling of regression coefficients from the multilevel regression models. The package has two main functions to handle not-missing-at-random missing responses and left-censored with not-missing-at random responses. The purpose is to provide a similar format as the other R regression functions but using 'Stan' models.

**Maintainer** Irene SL Zeng <i.zeng@auckland.ac.nz>

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**ByteCompile** true

**Depends** R (>= 3.5.0), Rcpp (>= 0.12.17), methods, stats

**Imports** rstan (>= 2.17.3),rstantools (>= 1.5.0),MASS,Matrix,stats4,ggplot2

**LinkingTo** StanHeaders (>= 2.17.2), rstan (>= 2.17.3), BH (>= 1.66.0-1), Rcpp (>= 0.12.17), RcppEigen (>= 0.3.3.4.0)

**Suggests** testthat, BiocStyle, knitr, rmarkdown, roxygen2 (>= 5.0.0)

**URL** https://doi.org/10.1101/153049

**RoxygenNote** 6.0.1

**biocViews** ImmunoOncology, Bayesian,CopyNumberVariation,Classification,Regression,MassSpectrometry,Proteomics,Software

**bugReport** https://github.com/ireneslzeng/mlmm/issues

**VignetteBuilder** knitr

**SystemRequirements** GNU make

**NeedsCompilation** yes

**git_url** https://git.bioconductor.org/packages/mlm4omics

**git_branch** master

**git_last_commit** 754d8b6

**git_last_commit_date** 2019-05-02

**Date/Publication** 2019-06-14

**Author** Irene Zeng [aut, cre],
Thomas Lumley [ctb]

# R topics documented:

## mlm4omics-package *The 'mlm4omics' package.*

### Description

To conduct Bayesian inference regression for responses with multilevel explanatory variables and missing values

### References

Stan Development Team (2018). RStan: the R interface to Stan. R package version 2.17.3. http://mc-stan.org

## mlmc *The multilevel function for missing and censored dependents: mlmc().*

### Description

mlmc() handles Bayesian multilevel model with response variable that has left-censored values, and missing values that depends on the response value itself. Apart from the response value, the missingness is also known to associate with the other variables. The method is created for analyzing mass-spectrometry data when it has abundance-dependant missing and censored values, and there are prior information available for the associations between the probability of missing and the known variables. The imputed values for the censored response are outputted as part of the parameters.

### Usage

```
mlmc(formula_completed, formula_missing, formula_censor = NULL,
    formula_subject, pdata, respond_dep_missing = TRUE,
    response_censorlim = NULL, pidname, sidname, prec_prior = NULL,
    alpha_prior = NULL, iterno = 100, chains = 3, thin = 1, seed = 125,
    algorithm = "NUTS", warmup = floor(iterno/2), adapt_delta_value = 0.9,
    savefile = FALSE)
```

## Arguments

`formula_completed`
The main regression model formula; It has the same formula format as lmr() and it is used to define the first level response and its explanatory variables.

`formula_missing`
The logistic regression model formula; It has the same formula as formula_completed.

`formula_censor` The formula used in the program to define the observations with censored values.

`formula_subject`
The second level formula in the multilevel model which is used to define responses such as subject and its explanatory variables.

`pdata` The dataset contains response and predictors in a long format. Response is a vector with an indicator variable to define the corresponding unit. The data needs to have the following rudimental variables: the indicator variable for first level response, second level indicator variable for subject such as subject id or a sampling unit, an indicator for missingness and indicator of censoring. Missingness and censored are two different classifications, when these two variables are tabulated, there must not have any observation defined as censored and missing. Data structure can be referred from the example and vignette.

`respond_dep_missing`
A logical variable to indicate whether response value is missing-dependant.

`response_censorlim`
The detectable limit for the response value, i.e. 1 mg per Liter for intensity value.

`pidname` Variable name to define the multilevel response unit, i.e. protein name or gene name.

`sidname` Variable name to define the subject unit, i.e. patient id or sampling id.

`prec_prior` prior precision matrix of the explanatory variables at the first level unit in the multilevel model, for example, the variables to predict the ion intensity. The dimension will be q x q, where q is the number of explanatory variables at the right-hand side of formula_completed. The default is a matrix with diagonal values of 0.01 and off-diagonal values of 0.005.

`alpha_prior` prior for coefficients of predictors to missing probability in the logistic regression. Its length will be equal to the number of variables at the right-hand side of the formula_missing. Default is a vector of zeros.

`iterno` Number of iterations for the posterior samplings.

`chains` rstan parameter to define number of chains of posterior samplings.

`thin` rstan parameter to define the frequency of iterations saved.

`seed` random seed for rstan function.

`algorithm` rstan parameter which has three options NUTS, HMC, Fixed param.

`warmup` Number of iterations for burn-out in stan.

`adapt_delta_value`
Adaptive delta value is an adaptation parameters for sampling algorithms, default is 0.85, value between 0-1.

`savefile` A logical variable to indicate if the sampling files are to be saved.

### Value

Return of the function is the result fitted by stan. It will have the summarized parameters from all chains and summary results for each chain.

### Examples

```
## Not run:
set.seed(150)
library(MASS)
var2 <- abs(rnorm(800,0,1)); treatment <- c(rep(0,400), rep(1,400));
var1 <- (1/0.85)*var2+2*treatment;
geneid <- rep(seq_len(50),16);
sid <- c(rep(seq_len(50),8), rep(seq_len(50)+50,8))
cov1 <- rWishart(1,df=50, Sigma <- diag(rep(1,50)))
u <- rnorm(50,0,1);mu <- mvrnorm(n=1, mu=u, cov1[,,1])
sdd <- rgamma(1, shape=1, scale=1/10);
for (i in seq_len(800)) {var1[i] <- var1[i]+rnorm(1, mu[geneid[i]], sdd)}
miss_logit <- var2*(-0.9)+var1*(0.001);
miss <- rbinom(800, 1, exp(miss_logit)/(exp(miss_logit)+1));
censor <- rep(0,800)
for (i in seq_len(800)) {if (var1[i]<0.002) censor[i]=1}
pdata <- data.frame(var1, var2, treatment, miss, censor, geneid, sid);
for ( i in seq_len(800))
{if ((pdata$miss[i]==1) & (pdata$censor[i]==1)) pdata$miss[i]=0};
for ( i in seq_len(800)) {
if (pdata$miss[i]==1) pdata$var1[i]=NA;
if (pdata$censor[i]==1) pdata$var1[i]=0.002};
pidname="geneid";sidname="sid";
#copy and paste the following formulas to the mlmm() function respectively
formula_completed=var1~var2+treatment;
formula_missing=miss~var2;
formula_censor=censor~1;
formula_subject=~treatment;
response_censorlim=0.002;

model1 <- mlmc(formula_completed=var1~var2+treatment,
formula_missing=miss~var2,
formula_censor=censor~1,
formula_subject=~treatment,
pdata=pdata,
response_censorlim=0.002,
respond_dep_missing=TRUE,
pidname="geneid",sidname="sid",
iterno=50,
chains=2,
savefile=FALSE)

## End(Not run)
```

## mlmm *The multilevel model to handle missing dependants: mlmm().*

## Description

mlmm() handles Bayesian multilevel model with response variable that has missing values that depends on the response value itself. Apart from the response value, the missingness is also known to associate with the other variables. The method is created for analyzing mass-spectrometry data when it has abundance-dependant missing and censored values, and there are prior information available for the associations between the probability of missing and the known variables. The function mlmm is written for response variable has no censored values while mlmc function include imputing censored values.

## Usage

```
mlmm(formula_completed, formula_missing, formula_subject, pdata,
    respond_dep_missing = TRUE, pidname, sidname, prec_prior = NULL,
    alpha_prior = NULL, iterno = 100, chains = 3, thin = 1, seed = 125,
    algorithm = "NUTS", warmup = floor(iterno/2), adapt_delta_value = 0.9,
    savefile = FALSE)
```

## Arguments

`formula_completed`
: The main regression model formula. It has the same formula format as lmr() and it is used to define the first level response and its explanatory variables.

`formula_missing`
: The logistic regression model formula. It has the same formula as formula_completed.

`formula_subject`
: The second level formula in the multilevel model which is used to define second level unit such as subject and explanatory variables.

`pdata`
: The dataset contains response and predictors in a long format. Response is a vector with an indicator variable to define the corresponding unit. The data needs to have the following rudimental variables: the indicator variable for first level response, the indicator variable for second level unit such as subject or a sampling unit, an indicator for missingness and indicator of censoring. Missingness and censored are two different classification, there should not have any overlap between missingness and censored. Data structure can be referenced from the example and reference papers.

`respond_dep_missing`
: An indicator of whether response value is missing-dependant.

`pidname`
: Variable name to define the multilevel response unit, i.e. protein name or gene name

`sidname`
: Vriable name to define the subject unit, i.e. patient id or sampling id

`prec_prior`
: prior precision matrix of the explanatory variables at the first level unit in the multilevel model, for example, the variables to predict the ion intensity. The dimension will be q x q, where q is the number of explanatory variables at the right-hand side of formula_completed. The default is a matrix with diagonal value of 0.01 and off-diagonal value of 0.005.

`alpha_prior`
: prior for coefficients of predictors to missing probability in the logistic regression. Its length will be equal to the number of variables at the right-hand side of the formula_missing. Default is a vector of zeros.

`iterno`
: Number of iterations for the posterior samplings

`chains`
: rstan() parameter to define number of chains of posterior samplings.

`thin` rstan() parameter to define the frequency of iterations saved.

`seed` random seed for rstan() function

`algorithm` rstan() parameter which has three options c(NUTS,HMC, Fixed_param).

`warmup` Number of iterations for burn-out in stan.

`adapt_delta_value`
Adaptive delta value is an adaptation parameters for sampling algorithms,default is 0.85, value between 0-1.

`savefile` A logical variable to indicate if the sampling files are to be saved.

## Value

Return of the function is the result fitted by stan(). It will have the summarized parameters from all chains and summary results for each chain. Plot() function will return the visualization of the mean and parameters.

## Examples

```
library(MASS)
set.seed(150)
var2 <- abs(rnorm(1000,0,1)); treatment <- c(rep(0,500),rep(1,500))
geneid <- rep(seq_len(20),50);
sid <- c(rep(seq_len(25),20),rep(seq_len(25)+25,20))
cov1 <- rWishart(1,df=100,Sigma <- diag(rep(1,100)))
u <- rnorm(100,0,1)
mu <- mvrnorm(n=1,mu=u,cov1[,,1])
sdd <- rgamma(1,shape=1,scale=1/10)
var1=(1/0.85)*var2+2*treatment
for (i in seq_len(1000)) {var1[i]=var1[i]+rnorm(1,mu[geneid[i]],sdd)}
miss_logit <- var2*(-0.9)+var1*(0.01)
probmiss <- exp(miss_logit)/(exp(miss_logit)+1)
miss <- rbinom(1000,1,probmiss); table(miss)
pdata <- data.frame(var1,var2,treatment,miss,geneid,sid)
for ( i in seq_len(1000)) if (pdata$miss[i]==1) pdata$var1[i]=NA;
pidname="geneid"; sidname="sid";
#copy and paste the following formulas to the mmlm() function respectively
formula_completed=var1~var2+treatment
formula_missing=miss~var2
formula_censor=censor~1
formula_subject=~treatment
model3 <- mlmm(formula_completed=var1~var2+treatment,
formula_missing=miss~var2,
formula_subject=~treatment, pdata=pdata, respond_dep_missing=TRUE,
pidname="geneid", sidname="sid", iterno=10, chains=2,
savefile=FALSE)
```

---

`pdata` *pdata for examples and testthat() pdata has 7 variables and var1 is the response variable, var2 is a continuous explanatory variable, treatment is another explanatory variable, miss and censor are indicator for missing and censored, geneid and sid represents gene id and subject id respectively.*

---

### Description

pdata for examples and testthat() pdata has 7 variables and var1 is the response variable, var2 is a continuous explanatory variable, treatment is another explanatory variable, miss and censor are indicator for missing and censored, geneid and sid represents gene id and subject id respectively.

### Usage

```
data(pdata)
```

### Format

An object of class `data.frame` with 100 rows and 7 columns.

---

## `setinitvalues` *The function to set initial values for parameters: setinitvalues().*

---

### Description

Generate initial values for parameters

### Usage

```
setinitvalues(npred, np, npred_miss, npred_sub, nmiss, nsid,
    censor_lim_upp = 0.008, ita_a = 1, ita_b = 1/10, g_mu = 0,
    g_sig = 1, alpha_mu_u = 0, alpha_mu_s = 1, alpha_theta_a = 1,
    alpha_theta_b = 1/10, beta2_theta_a = 1, beta2_theta_b = 1/10)
```

### Arguments

| | |
|---|---|
| `npred` | number of predictors for the regression model |
| `np` | number of protein/metabolite units comprised of the response values (i.e. which represents peptides' ion-intensities used to construct protein/metabolite's abundance) |
| `npred_miss` | number of predictors for missingness |
| `npred_sub` | number of predictors for the second level units such as subjects |
| `nmiss` | number of observations with missing responses values |
| `nsid` | number of second level units i.e. subjects |
| `censor_lim_upp` | upper-limit of censored value of responses. The default value is 0.001 according to an experiment device. User can change it according to the data. |
| `ita_a` | shape parameter for gamma distributed prior-ita (std of response value). The default is set to 1. |
| `ita_b` | rate parameter for gamma distributed prior-ita. The default is set to 1/10. The default values of shape and rate parameters provide a reasonable wide range of initial value for ita. Users can change it accordingly. |
| `g_mu` | mean of normal distributed location parameter g for re-parameterising U (regression coefficient of model in the completed data). The default value is set to 0 for the mean of standard normal distribution. |

`g_sig` std of normal distributed location parameter g. The default is set to 1 for the std of normal distribution.

`alpha_mu_u` mean of normal distributed location parameter alpha_mu for re-parameterising alpha (regression coefficient of logistic regression model for missing prob). The default is set to 0.

`alpha_mu_s` std of normal distributed location parameter alpha_mu

`alpha_theta_a` shape parameter of gamma-distributed dispersion parameter alpha_theta for re-parameterising alpha. Default value is set to 1 as a natural starting value.

`alpha_theta_b` rate parameter of gamma-distributed alpha_theta. Default value uses 1/10 as for ita_b. Both default values of shape(_a) and rate(_b) of alpha_theta can be changed to give a wider range (_b=1/10) or a narrower range (_b=0.5).

`beta2_theta_a` shape parameter of gamma-distributed beta2_theta for re-parameterizing beta2 (regression coefficient for second level units,i.e. subject). Default value uses 1.

`beta2_theta_b` rate parameter of gamma distributed dispersion parameter beta2_theta. Default value used 1/10, same as for ita_b.

## Value

pVAR precision matrix for predictors in completed data model

U_latent standardized multinormal distributed latern variable to re-parameterise regression coefficient U.

g location parameter to re-parameterise U.

alpha_mu mean value for alpha(regression coefficient of model for missing probability).

alpha_latent standardized normal distributed latent variable to re-parameterize alpha.

beta2_latent standardized multivariate normal distributed latent variable to re-parameterising beta2.

beta2_mu mean of the multivariate normal distributed beta2

y_m_latent standardized normal distributed latent variable to re-parameterise response variable.

## Examples

```
testexmp <- setinitvalues(npred=2,np=3,npred_miss=3,npred_sub=2,nmiss=10,
nsid=30)
```

# Index