

# Package ‘upmfit’

February 19, 2017

**Type** Package

**Title** Unified Probability Model Fitting

**Version** 0.1.0

**Description** Fitting a Unified Probability Model for household-community tuberculosis transmission dynamics.

**URL** <http://people.bu.edu/aimcinto/>

**Depends** R (>= 3.2.1)

**License** GPL-3

**LazyData** TRUE

**Imports** stats, R2jags, mcmcplots

**Suggests** doMC, knitr, rmarkdown

**SystemRequirements** JAGS (<http://mcmc-jags.sourceforge.net>)

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Avery McIntosh [aut, cre]

**Maintainer** Avery McIntosh <[aimcinto@bu.edu](mailto:aimcinto@bu.edu)>

**Repository** CRAN

**Date/Publication** 2017-02-19 10:53:41

## R topics documented:

upmbuilder	2
upmdata	3
upmfit	3
upmrun	4
<b>Index</b>	<b>6</b>

---

 upmbuilder

*Bayesian Markov Chain Monte Carlo model building for household-community tuberculosis transmission models*


---

## Description

Bayesian Markov Chain Monte Carlo model building for household-community tuberculosis transmission models

## Usage

```
upmbuilder(design.matrix, categorical.columns = 0, prior.beta.means = rep(x
  = 0, times = ncol(design.matrix[, which(names(design.matrix) != "cluster")])),
  prior.beta.sd = rep(x = 3.162278, times = ncol(design.matrix[,
  which(names(design.matrix) != "cluster")])), prior.alpha = c(0, 0.44))
```

## Arguments

`design.matrix` A data frame type design matrix of observations for rows, variables on each observation for columns. If there are natural clusters (households), then a variable titled 'cluster' must be included. The design matrix must have exactly one column titled 'y' for outcome on each observation.

`categorical.columns`

A scalar (vector) specification of any column(s) of type "factor" to be coded as a dummy variable regression. Default is 0: no columns are factor variables.

`prior.beta.means`

A scalar (vector) specification of Gaussian prior distribution mean(s) for the coefficient(s) in the household component of the logistic regression model. Default is noninformative on the inverse logit transformed (odds ratio) scale.

`prior.beta.sd`

A scalar (vector) specification of Gaussian prior distribution standard deviation(s) for the coefficient(s) in the household component of the logistic regression model. Default is noninformative on the inverse logit transformed (odds ratio) scale.

`prior.alpha`

A vector of prior probability parameters for the alpha term in the community component of model, a logistic regression with intercept only. Default is noninformative, with  $p^C$  being essentially uniform on the unit interval.

## Details

The package contains long-form documentation in the form of a vignette that cover the use of the main functions. Use `browseVignettes(package="upmfit")` to access them.

## Value

A list of: a JAGS model script for specification of the unified probability model to be used; the (possibly recoded) design matrix; a vector of the beta coefficient names to be used in the household component of the UPM.

**Examples**

```
upmbuilder(design.matrix=upmdata)

upmbuilder(design.matrix=upmdata, prior.alpha=c(-2.2,round(1/sqrt(3),3)))
suppressWarnings(cat(upmbuilder(design.matrix=upmdata,
prior.alpha=c(-2.2,round(1/sqrt(3),3)))[[1]]))
```

---

upmdata	<i>The dataset 'upmdata.Rdata' contains example data from a theoretical tuberculosis household contact study of 800 households contacts of an infectious pulmonary TB case from 108 households. The first four variables are numeric. The final predictor variable is binary. Next is the data defining household: cluster, and finally the measured outcome of a Tuberculin Skin Test (TST/PPD) for each contact, y. In these data, the true proportion of contacts contracting TB from outside the home is 15.5 percent. The true coefficient values are, in order <math>b_0 = -1.5</math>, <math>b_1 = 0.15</math>, <math>b_2 = -2.25</math>, <math>b_3 = 0</math> (no effect), <math>b_4 = 0.2</math>, <math>b_5 = -0.1</math>.</i>
---------	---

---

**Description**

See the introductory vignette for this package for more information.

**Usage**

```
upmdata
```

**Format**

An object of class `data.frame` with 800 rows and 7 columns.

---

upmfit	<i>upmfit: Bayesian MCMC sampling with the Unified Probability Model</i>
--------	--

---

**Description**

The upmfit package uses methods described by McIntosh, et al. "Extensions to Bayesian Generalized Linear Mixed Effects Models for Household Tuberculosis Transmission," *Statistics in Medicine* (2017), to implement functions that generate model script in the JAGS language for the Unified Probability Model (UPM), and to run a JAGS MCMC sampler for posterior probability density estimation with the model script.

## Details

It should be noted that, while the UPM and R package `upmfit` were designed specifically with tuberculosis household contact studies in mind, the method is applicable to any scenario where there are two competing risks for a single outcome with unobserved linkage between the sources of risk and the outcome. Example situation where this method could be applicable are in profiling the risk of MRSA infection from a nosocomial versus other source, or in modeling risk of TB infection among populations utilizing homeless shelters. Non-clinical examples could be in an industrial process control setting where a manufactured item has two sources of degradation, one observed and one unobserved, or in a chemical mixing procedure where catalysis can occur from the mixing of an exogenous agent or autocatalysis.

## upmfit functions

`upmfit` has: function `upmbuilder()` to create the JAGS model script and display a (possibly redefined) design matrix from a design matrix input by the user; `upmrun()` to generate posterior parameter estimates via a Markov Chain Monte Carlo sampler; and synthetic dataset `upmdata`.

For a longer introduction, see the introductory vignette for this package. Use command `vignette("upm-primer", package="upmfit")` or `browseVignettes(package = "upmfit")` to access the vignettes.

---

upmrun

*Bayesian Markov Chain Monte Carlo model building for household-community tuberculosis transmission models*

---

## Description

Bayesian Markov Chain Monte Carlo model building for household-community tuberculosis transmission models

## Usage

```
upmrun(design.matrix, categorical.columns = 0, prior.beta.means = rep(x = 0,
  times = ncol(design.matrix[, which(names(design.matrix) != "cluster")])),
  prior.beta.sd = rep(x = 3.162278, times = ncol(design.matrix[,
  which(names(design.matrix) != "cluster")])), prior.alpha = c(0, 0.44),
  n.chains = 3, n.iter = 50000, n.burnin = n.iter/2, n.thin = max(1,
  floor((n.iter - n.burnin)/1000))
```

## Arguments

`design.matrix` A data frame type design matrix of observations for rows, variables on each observation for columns. If there are natural clusters (households), then a variable titled 'cluster' must be included. The design matrix must have exactly one column titled 'y' for outcome on each observation and at least one column of predictors.

`categorical.columns`

A scalar (vector) specification of any column(s) of type 'factor' to be coded as a dummy variable regression. Default is 0: no columns are factor variables.

<code>prior.beta.means</code>	A scalar (vector) specification of Gaussian prior distribution mean(s) for the coefficient(s) in the household component of the logistic regression model. Default is noninformative on the inverse logit transformed (odds ratio) scale.
<code>prior.beta.sd</code>	A scalar (vector) specification of Gaussian prior distribution standard deviation(s) for the coefficient(s) in the household component of the logistic regression model. Default is noninformative on the inverse logit transformed (odds ratio) scale.
<code>prior.alpha</code>	A vector of prior probability parameters for the alpha term in the community component of model, a logistic regression with intercept only. Default is noninformative, with $p^C$ being essentially uniform on the unit interval.
<code>n.chains</code>	From JAGS documentation: number of Markov chains (default: 3).
<code>n.iter</code>	From JAGS documentation: number of total iterations per chain (including burn in; JAGS default is 2000, updated here to 50000).
<code>n.burnin</code>	From JAGS documentation: length of burn in, i.e. number of iterations to discard at the beginning. Default is $n.iter/2$ , that is, discarding the first half of the simulations. If <code>n.burnin</code> is 0, <code>jags()</code> will run 100 iterations for adaptation.
<code>n.thin</code>	From JAGS documentation: thinning rate. Must be a positive integer. Set <code>n.thin &gt; 1</code> to save memory and computation time if <code>n.iter</code> is large. Default is $\max(1, \text{floor}(n.chains * (n.iter - n.burnin) / 1000))$ which will only thin if there are at least 2000 simulations.

### Details

The package contains long-form documentation in the form of a vignette that cover the use of the main functions. Use `browseVignettes(package="upmfit")` to access them.

### Value

Returns a JAGS object of posterior samples for: household predictors (betas); if clusters are present, the posterior distribution of the hierarchical effects mean and standard deviation; the probability of community-acquired infection (`post.comm.risk`); the probability of household-acquired infection (`pHHinfection`); and model deviance.

### Examples

```
upmrun(design.matrix=upmdata, n.iter=100, prior.alpha = c(-2.2, 1/sqrt(3)))
```

# Index

## \*Topic **datasets**

upmdata, [3](#)

upmbuilder, [2](#)

upmdata, [3](#)

upmfit, [3](#)

upmfit-package (upmfit), [3](#)

upmrun, [4](#)