Package 'BaDGE'

August 16, 2011

Title BaDGE (Bayesian model for Detecting Gene Environment interaction)

Version 1.0.0

Date 2011-08-16

Author Kai Yu

Description A flexible Bayesian model for studying gene-environment interaction

Maintainer Kai Yu <yuka@mail.nih.gov>

Depends R (>= 2.10.1), fields, cluster

License GPL-2

R topics documented:

`BaDGE` *BaDGE (Bayesian model for Detecting Gene Environment interaction)*

Description

A flexible Bayesian model for studying gene-environment interaction.

Details

A more detailed analysis investigating how a gene or a chromosomal region and an established environment risk factor interact to influence the disease risk is an important follow up step in characterizing the effects of genetic markers found to be associated with a disease outcome. The standard approach that considers one genetic marker at a time could misrepresent and underestimate the genetic contribution to the joint effect when one or more functional loci, some of which might not be genotyped, exist in the region, and interact with the environment risk factor in a complex way. `BaDGE` implements a more global approach based on a Bayesian model that uses a latent genetic profile variable to capture the gene's effect and allows the environment effect to vary across different genetic profile categories. `BaDGE` also uses a resampling-based test derived from the developed Bayesian model for the detection of gene-environment interaction. The main function is `badge`.

Author(s)

Kai Yu <yuka@mail.nih.gov>

References

Yu K, Wacholder S, Wheeler W, Wang Z, Caporaso N, Landi M Liang F. A Flexible Bayesian Model for Studying Gene-Environment Interaction (Submitted manuscript).

`badge` *Bayesian model for Detecting Gene Environment interaction*

Description

A flexible Bayesian model for studying gene-environment interaction.

Usage

```
badge(data, cc.var, exposure.var, group.var, out.dir, op=NULL)
```

Arguments

`data`	Data frame containing the disease status, exposure variable, group variable and possibly covariates.
`cc.var`	Variable name for the disease status. This variable should be coded as 0 for no disease and 1 for disease.
`exposure.var`	Variable name for the exposure. This variable should be numerically coded.
`group.var`	Variable name for the groups. This variable should be coded as integers from 1 to the number of groups.
`out.dir`	Directory where the output files will be written.
`op`	List of options. See `details` for all possible options.

Details

Then input data should only contain finite values for the variables to be used in the analysis.

Options list:

Below are the names for the options list `op`. All names have default values if they are not specified.

- `sim.mat` A matrix defining the similarity between groups. This matrix must be symmetric with zeros on the main diagonal. The ith row and jth column of this matrix refers to groups i and j. The default is a matrix of zeros.
- `covars` Character vector of variable names to be used as covariates. Example: `covars=c("x1", "x2", "x3")`. The default is that no covariates will be used in the analysis.
- `out.string` Character string to be appended to the output file names. The default is "".
- `n_iter` The number of iterations. The default is 200000.
- `n_sep_out` Integer specifiying output to be written every `n_sep_out` iteration(s). The default is 1.
- `k_max` Maximum number of clusters. The default is 2.
- `random_seed` Positive integer. The default is 12345.
- `w_m` Number of auxiliary samples for updating the interaction parameter. The default is 50.
- `update_prop` Proportion of groups to be updated when updating the allocation vector. The default is 1.0.
- `update_prop_w` Proportion of groups to be updated when updating the auxiliary sample. The default is 1.0.
- `method` 0 (uniform) or 1 (normal) for the distribution of alpha, beta, and tao. The default is 0.
- `alpha_min` Minimum value for each alpha when `method` = 0. The default is -4.
- `alpha_max` Maximum value for each alpha when `method` = 0. The default is 4.
- `alpha_mu` Mean parameter for each alpha when `method` = 1. The default is 0.
- `alpha_sigma2` Variance parameter for each alpha when `method` = 1. The default is 4.
- `beta_min` Minimum value for each beta when `method` = 0. The default is -4.
- `beta_max` Maximum value for each beta when `method` = 0. The default is 4.
- `beta_mu` Mean parameter for each beta when `method` = 1. The default is 0.
- `beta_sigma2` Variance parameter for each beta when `method` = 1. The default is 4.
- `psi_min` Minimum value of the interaction parameter. The default is 0.
- `psi_max` Maximum value of the interaction parameter. The default is 1.2.
- `tao_min` Minimum value for each tao when `method` = 0. The default is -4.
- `tao_max` Maximum value for each tao when `method` = 0. The default is 4.
- `tao_mu` Mean parameter for each tao when `method` = 1. The default is 0.
- `tao_sigma2` Variance parameter for each tao when `method` = 1. The default is 4.
- `sigma2_alpha` Variance of a normal distribution with mean 0 used to update alpha. The default is 0.005.
- `sigma2_beta` Variance of a normal distribution with mean 0 used to update beta. The default is 0.005.
- `sigma2_tao` Variance of a normal distribution with mean 0 used to update tao. The default is 0.005.

- `sigma2_psi` Variance of a normal distribution with mean 0 used to update psi. The default is 0.02.

Output files:

Below are the descriptions of the output files. Each line of output is written once every `n_sep_out` iterations.

- `out_psi` Estimate of the psi interaction parameter
- `out_alpha` Estimates of the alpha parameters
- `out_beta` Estimates of the beta parameters
- `out_alloc_z` Cluster number for each group
- `out_dev_i` Deviance
- `out_tao` Created only if there were covariates used in the analysis.

Value

The returned value is NULL. All output files are written to `out.dir`.

Author(s)

Kai Yu

See Also

`define.NB.geno`

Examples

```
set.seed(123)
n    <- 100
cc   <- rbinom(n, 1, 0.5)
x    <- rbinom(n, 1, 0.5)
grp  <- sample(1:20, n, replace=TRUE)
data <- data.frame(cc, x, grp)
dir  <- "K:/bayesian/R_package/temp/"

# Not run
#badge(data, "cc", "x", "grp", dir)

# Add some covariates
ncov <- 3
for (i in 1:ncov) data[, paste("cov", i, sep="")] <- runif(n)

# Define the options list. See above for all possible options.
op <- list(covars=c("cov1", "cov2", "cov3"))

# Not run
#badge(data, "cc", "x", "grp", dir, op=op)
```

`define.NB.geno` *Define a subject grouping and similarity matrix for the groups*

Description

Define a subject grouping and similarity matrix for the groups using genotype data

Usage

```
define.NB.geno(geno.mat, num.neighbor=4, method.tie="min")
```

Arguments

`geno.mat`	Matrix of genotype data. Genotype should be coded as 0, 1, or 2 for the number of copies of the minor allele. The dimension of this matrix must be the number of subjects by the number of SNPs.
`num.neighbor`	The number of neighbors for the similarity matrix.
`method.tie`	One of "average", "first", "random", "max", "min".

Details

This function can be called to obtain a vector of groups for the subjects and a similarity matrix for the groups that can be used as input for the `badge` function. Subject with a similar genotype structure will be put into the same group.

Value

A list containing the subject grouping (`grp.subj`) and similarity matrix (`NB.mat`). The order of `grp.subj` is the same as the order of the subjects in `geno.mat`. The groups will be coded as 1, 2, ..., Ngroups. The dimension of `NB.mat` will be Ngroups by Ngroups, where the ith row is for group number i.

Author(s)

Kai Yu

See Also

`badge`

Examples

```
# Create a matrix of 0, 1, and 2
set.seed(123)
nsub <- 100
nsnp <- 5
mat  <- rbinom(nsub*nsnp, 2, 0.45)
dim(mat) <- c(nsub, nsnp)

ret <- define.NB.geno(mat)
table(ret$grp.subj)
```

```
ret$NB.mat[1:5, 1:5]
```

---

heatMap *Heat Map*

Description

Create a heat map using the Krig function

Usage

```
heatMap(x, y, z, op=NULL)
```

Arguments

x	Vector of x-coordinates.
y	Vector of y-coordinates.
z	Vector of surface values.
op	List of options. See `details` for all possible options.

Details

The heat map is produced by first calling the `Krig` function and then using the using the output from `Krig` in the `surface` function.

Options list:
Below are the names for the options list `op`. All names have default values if they are not specified.

- `ncolors` The number of colors used in `heat.colors`. The default is 20.
- `xlab` X-axis label. The default is "".
- `ylab` Y-axis label. The default is "".

Value

The returned value is the Krig model fit.

See Also

`plot_xy_hist`

Examples

```
set.seed(123)
n <- 100
x <- -4 + 4*runif(n)
y <- -4 + 4*runif(n)
z <- rnorm(n) + x + y
fit <- heatMap(x, y, z)
```

plot_xy_hist *Scatter plot of histograms*

Description

Performs a scatter plot where each "point" is a histogram

Usage

```
plot_xy_hist(veclist, x, y, nbars=5, force=0, xscale=1, yscale=1,
             xlab="", ylab="", title="", col="black")
```

Arguments

veclist	List of vectors from which each histogram is determined. The vectors can be of different lengths.
x	Vector of x-coordinates for each histogram. The length of this vector must be equal to length(veclist).
y	Vector of y-coordinates for each histogram. The length of this vector must be equal to length(veclist).
nbars	Number of bars in each histogram (the breaks option in hist)
force	0 or 1 to force the number of bars in each histogram to be nbars.
xscale	Scaling factor for the width of each histogram.
yscale	Scaling factor for the height of each histogram.
xlab	X-axis label
ylab	Y-axis label
title	Title of plot
col	Color of each histogram

Details

The lower left point of the histogram defined by veclist[[i]] has coordinates (x[i], y[i]). The options nbars, force, and col are allowed to be vectors of the same length as x.

Value

The returned value is a list of sublists of information about each histogram. The order is the same as veclist.

Author(s)

Kai Yu

See Also

heatMap

Examples

```
set.seed(123)
nr <- 10000
nc <- 50
dlist <- list()
for (i in 1:nc) dlist[[i]] <- rnorm(ceiling(1000+nr*runif(1)))
x <- 2*runif(nc) + 10*runif(nc)
y <- 2*runif(nc) + 5*runif(nc)

ret <- plot_xy_hist(dlist, x, y, xscale=0.25, yscale=0.5)
```

`post_badge` *Post processing badge output*

Description

A function to summarize the output from the badge function

Usage

```
post_badge(geno.mat, data, cc.var, exposure.var, group.var, out.dir, op=NULL)
```

Arguments

`geno.mat`	Matrix of genotype data. Genotype should be coded as 0, 1, or 2 for the number of copies of the minor allele. The dimension of this matrix must be the number of subjects by the number of SNPs.
`data`	Data frame containing the disease status, exposure variable, group variable and possibly covariates.
`cc.var`	Variable name for the disease status. This variable should be coded as 0 for no disease and 1 for disease.
`exposure.var`	Variable name for the exposure. This variable should be numerically coded.
`group.var`	Variable name for the groups. This variable should be coded as integers from 1 to the number of groups.
`out.dir`	Directory where the output files will be written.
`op`	List of options. See `details` for all possible options.

Details

`geno.mat` and `data` should be the same objects that were used in `define.NB.geno` and `badge`.

Options list:

Below are the names for the options list `op`. All names have default values if they are not specified.

- `covars` A character vector of variable names to be used as covariates. Example: `covars=c("x1", "x2", "x3")`. The default is that no covariates will be used in the analysis.

- `out.string` Character string to be appended to the output file names. The default is "".
- `M1` Starting iteration to use. The default is 1.
- `M2` Final iteration to use. The default is Inf.
- `everyN` Integer to use every `everyN` iterations. The default is 1.

Output file:

The output file will contain 3 plots:

- `1` Plot of the first 2 principal components
- `2` Heat map of the median odds(alpha) parameters
- `2` Heat map of the median odds(beta) parameters

Value

The returned value is list containing the deviance information criteria (`dic`), the cluster assignment for each subject (`subj.assign`), the first 5 principal components (`pc.mat`), the median odds(alpha) for each subject (`alpha.med.odds`), and the median odds(beta) for each subject (`beta.med.odds`) An output file of plots is written to `out.dir`.

Author(s)

Kai Yu

See Also

`define.NB.geno` `badge`

Index