

# Package ‘GEInfo’

July 21, 2025

**Type** Package

**Title** Gene-Environment Interaction Analysis Incorporating Prior Information

**Version** 1.0

**Author** Xiaoyan Wang, Hongduo Liu, and Shuangge Ma

**Maintainer** Xiaoyan Wang <xywang@hnu.edu.cn>

**Description** Realize three approaches for Gene-Environment interaction analysis. All of them adopt Sparse Group Minimax Concave Penalty to identify important G variables and G-E interactions, and simultaneously respect the hierarchy between main G and G-E interaction effects. All the three approaches are available for Linear, Logistic, and Poisson regression. Also realize to mine and construct prior information for G variables and G-E interactions.

**License** GPL

**Encoding** UTF-8

**Imports** MASS, glmnet, stats, rvest, dplyr, pheatmap, grDevices, graphics

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-01-10 17:52:50 UTC

## Contents

CGEInfo . . . . .	2
coef.GEInfo . . . . .	4
cv.CGEInfo . . . . .	4
cv.GEInfo . . . . .	7
GEInfo . . . . .	10
matW . . . . .	12
plot.GEInfo . . . . .	13
plot.PubMed . . . . .	13
predict.GEInfo . . . . .	14
PubMed.search . . . . .	14
TypeS . . . . .	15

CGEInfo

*CGEInfo and GEsgMCP approaches with fixed tunings***Description**

Realize to estimate CGEInfo and GEsgMCP approaches at fixed tunings.

**Usage**

```
CGEInfo(
  E,
  G,
  Y,
  family,
  lam1,
  lam2,
  xi = 6,
  epsilon = 0,
  max.it = 500,
  thresh = 0.001,
  S_G = NULL,
  S_GE = NULL
)
```

**Arguments**

E	Observed matrix of E variables, of dimensions n x q.
G	Observed matrix of G variables, of dimensions n x p.
Y	Response variable of length n. Quantitative for family="gaussian", or family="poisson" (non-negative count). For family="binomial" should be a factor with two levels.
family	Model type: one of ("gaussian", "binomial", "poisson").
lam1	A user supplied lambda1.
lam2	A user supplied lambda2.
xi	Tuning parameter of MCP penalty. Default is 6.
epsilon	Tuning parameter of Ridge penalty which shrinks the coefficients of variables having prior information. Default is 0.
max.it	Maximum number of iterations (total across entire path). Default is 500.
thresh	Convergence threshold for group coordinate descent algorithm. The algorithm iterates until the change for each coefficient is less than thresh. Default is 1e-3.
S_G	A user supplied vector, denoting the subscript of G variables which have prior information. Default is NULL.

**S\_GE** A user supplied matrix, denoting the subscript of G-E interactions which have prior information. The first and second columns of S\_GE represent the subscript of G variable and the subscript of E variable, respectively. For example, S\_GE = matrix( c(1, 2), ncol = 2), which indicates that the 1st G and the 2nd E variables have an interaction effect on Y. Default is NULL. If both S\_G and S\_GE are NULL, no prior information is incorporated in the model, in which case function CGEInfo realizes GEsgMCP approach.

### Value

An object of class "GEInfo" is returned, which is a list including the estimation results at fixed tunings.

<b>a</b>	Coefficient vector of length q for E variables.
<b>b</b>	Coefficient vector of length (q+1)p for W (G variables and G-E interactions).
<b>beta</b>	Coefficient vector of length p for G variables.
<b>gamma</b>	Coefficient matrix of dimensions p*q for G-E interactions.
<b>alpha</b>	Intercept.
<b>coef</b>	A coefficient vector of length (q+1)*(p+1), including the estimates for $\alpha$ (intercept), $a$ (coefficients for all E variables), and $b$ (coefficients for all G variables and G-E interactions).

### References

Wang X, Xu Y, and Ma S. (2019). Identifying gene-environment interactions incorporating prior information. Statistics in medicine, 38(9): 1620-1633. doi: [10.1002/sim.8064](https://doi.org/10.1002/sim.8064)

### Examples

```
n <- 30; p <- 5; q <- 2
E <- MASS::mvrnorm(n, rep(0,q), diag(q))
G <- MASS::mvrnorm(n, rep(0,p), diag(p))
W <- matW(E, G)
alpha <- 0; a <- seq(0.4, 0.6, length=q);
beta <- c(seq(0.2, 0.5, length=3),rep(0, p-3)) # coefficients of G variables
vector.gamma <- c(0.8, 0.5, 0, 0)
gamma <- matrix(c(vector.gamma, rep(0, p*q - length(vector.gamma))), nrow=p, byrow=TRUE)
mat.b.gamma <- cbind(beta, gamma)
b <- as.vector (t(mat.b.gamma)) # coefficients of G and G-E interactions
Y <- alpha + E %*% a + W %*% b + rnorm (n, 0, 0.5)
S_G <- c(1)
S_GE <- cbind(c(1), c(1))
fit1 <- CGEInfo(E, G, Y,family='gaussian', S_G=S_G, S_GE=S_GE,lam1=0.4,lam2=0.4)
```

---

coef.GEInfo	<i>Extract coefficients from a fitted object</i>
-------------	--

---

### Description

Report the estimate of all coefficients from a fitted "CGEInfo" or "GEInfo" model object.

### Usage

```
## S3 method for class 'GEInfo'
coef(object, ...)
```

### Arguments

object	A fitted "CGEInfo" or "GEInfo" model object for which the estimate of coefficients is extracted.
...	Other arguments.

### Value

A coefficient vector of length  $(q+1) \times (p+1)$ , including the estimates for  $\alpha$  (intercept),  $a$  (coefficients for all E variables), and  $b$  (coefficients for all G variables and G-E interactions).

---

cv.CGEInfo	<i>Cross-validation for CGEInfo</i>
------------	-------------------------------------

---

### Description

Does k-fold cross-validation for CGEInfo, returns the estimation results at best tunings, and produces a heatmap for the identification results.

### Usage

```
cv.CGEInfo(
  E,
  G,
  Y,
  family,
  nfolds = 3,
  xi = 6,
  epsilon = 0,
  max.it = 500,
  thresh = 0.001,
  criterion = "BIC",
  lam1 = NULL,
```

```

    lam2 = NULL,
    S_G = NULL,
    S_GE = NULL
)

```

### Arguments

E	Observed matrix of E variables, of dimensions n x q.
G	Observed matrix of G variables, of dimensions n x p.
Y	Response variable, of length n. Quantitative for family="gaussian", or family="poisson" (non-negative counts). For family="binomial" should be a factor with two levels.
family	Model type: one of ("gaussian", "binomial", "poisson").
nfolds	Number of folds. Default is 3. Although nfolds can be as large as the sample size n (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is nfolds=3. See Details.
xi	Tuning parameter of MCP penalty. Default is 6.
epsilon	Tuning parameter of Ridge penalty which shrinks the coefficients having prior information. Default is 0.
max.it	Maximum number of iterations (total across entire path). Default is 500.
thresh	Convergence threshold for group coordinate descent algorithm. The algorithm iterates until the change for each coefficient is less than thresh. Default is 1e-3.
criterion	Criterion used for cross-validation. Currently five options: MSE, AIC, BIC, EBIC, GCV. Default is BIC. See Details.
lam1	A user supplied lambda1 sequence. Typical usage is to have the program compute its own lambda1 sequence. Supplying a value of lam1 overrides this. Default is lam1=NULL.
lam2	A user supplied lambda2 sequence. Default is lam2=NULL. Typical usage is to have the program compute its own lambda2 sequence. Supplying a value of lam2 overrides this. Default is lam2=NULL.
S_G	A user supplied vector, denoting the subscript of G variables which have prior information. Default is NULL. See Details.
S_GE	A user supplied matrix, denoting the subscript of G-E interactions which have prior information. The first and second columns of S_GE represent the subscript of G variable and the subscript of E variable, respectively. For example, S_GE = matrix( c(1, 2), ncol = 2), indicating that the 1st G variable and the 2nd E variables have an interaction effect on Y. Default is NULL. If both S_G and S_GE are NULL, no prior information is incorporated in the model, in which case this function realizes GEsgMCP approach. See Details.

### Details

The function calls CGEInfo nfolds times, each time leaving out 1/nfolds of the data. The cross-validation error is based on the user given "criterion". cv.CGEInfo supports to construct two methods: GEInfo and GEsgMCP, depending on whether S\_G and S\_GE are NULL. When either S\_G or

`S_GE` is not NULL, CGEInfo approach is realized, which completely trusts the prior information. Otherwise, GEsgMCP approach is constructed, in which no prior information is incorporated.

In order to select the optimal tunings, there are five criteria available, which are MSE, AIC, BIC, GCV, and EBIC. Let  $L$  be the loss function of the model,  $MSE = L$ ,  $AIC = 2L + 2df$ ,  $BIC = 2L + \ln(n)df$ ,  $GCV = 2L/(1 - df/n)^2$ , and  $EBIC = 2L + \ln(n)df + 2df\ln(nvar)(1 - \ln(n)/(2\ln(nvar)))$ . In most cases, BIC is a good choice. In the case of high dimension, EBIC criterion is recommended first, which has demonstrated satisfactory performance in high-dimensional studies.

## Value

An object of class "GEInfo" is returned, which is a list with the ingredients of the cross-validation fit.

<code>best.tuning</code>	A vector of length 2, containing the best <code>lambda1</code> and <code>lambda2</code> selected by cross-validation.
<code>a</code>	Coefficient vector of length <code>q</code> for all E variables.
<code>beta</code>	Coefficient vector of length <code>p</code> for all G variables.
<code>gamma</code>	Coefficient matrix of dimensions <code>p*q</code> for G-E interactions.
<code>b</code>	Coefficient vector of length $(q+1)*p$ for <code>W</code> (G variables and G-E interactions).
<code>alpha</code>	Intercept.
<code>coef</code>	A coefficient vector of length $(q+1)*(p+1)$ , including the estimates for $\alpha$ (intercept), $a$ (coefficients for all E variables), and $b$ (coefficients for all G variables and G-E interactions).
<code>nvar</code>	Number of non-zero coefficients at the best tunings.

## References

Wang X, Xu Y, and Ma S. (2019). Identifying gene-environment interactions incorporating prior information. *Statistics in medicine*, 38(9): 1620-1633. doi: [10.1002/sim.8064](https://doi.org/10.1002/sim.8064)

## Examples

```
n <- 30; p <- 5; q <- 2
E <- MASS::mvrnorm(n, rep(0,q), diag(q))
G <- MASS::mvrnorm(n, rep(0,p), diag(p))
W <- matW(E, G)
alpha <- 0; a <- seq(0.4, 0.6, length=q);
beta <- c(seq(0.2, 0.5, length=3), rep(0, p-3))
vector.gamma <- c(0.8, 0.5, 0, 0)
gamma <- matrix(c(vector.gamma, rep(0, p*q - length(vector.gamma))), nrow=p, byrow=TRUE)
mat.b.gamma <- cbind(beta, gamma)
b <- as.vector(t(mat.b.gamma))
Y <- alpha + E %*% a + W %*% b + rnorm(n, 0, 0.5)
S_G <- c(1)
S_GE <- cbind(c(1), c(1))
fit2 <- cv.CGEInfo(E, G, Y, family='gaussian', S_G=S_G, S_GE=S_GE, lam1=0.4, lam2=0.4)
```

cv.GEInfo

*Cross-validation for GEInfo***Description**

Does k-fold cross-validation for GEInfo approach, which adaptively accommodates the quality of the prior information and automatically detects the false information. Tuning parameters are chosen based on a user given criterion.

**Usage**

```
cv.GEInfo(
  E,
  G,
  Y,
  family,
  S_G,
  S_GE,
  nfolds = 3,
  xi = 6,
  epsilon = 0,
  max.it = 500,
  thresh = 0.001,
  criterion = "BIC",
  Type_Y = NULL,
  kappa1 = NULL,
  kappa2 = NULL,
  lam1 = NULL,
  lam2 = NULL,
  tau = c(0, 0.25, 0.5, 0.75, 1)
)
```

**Arguments**

E	Observed matrix of E variables, of dimensions $n \times q$ .
G	Observed matrix of G variables, of dimensions $n \times p$ .
Y	Response variable, of length $n$ . Quantitative for family="gaussian", or family="poisson" (non-negative counts). For family="binomial" should be a factor with two levels.
family	Model type: one of ("gaussian", "binomial", "poisson").
S_G	A user supplied vector, denoting the subscript of G variables which have prior information.
S_GE	A user supplied matrix, denoting the subscript of GE interactions which have prior information. The first and second columns of S_GE represent the subscript of G variable and the subscript of E variable, respectively. For example, S_GE

	= matrix( c(1, 2), ncol = 2), which indicates that the 1st G variable and the 2nd E variables have an interaction effect on Y.
nfolds	Number of folds. Default is 3. Although nfolds can be as large as the sample size n (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is nfolds=3
xi	Tuning parameter of MCP penalty. Default is 6.
epsilon	Tuning parameter of Ridge penalty which shrinks on the coefficients having prior information. Default is 0.
max.it	Maximum number of iterations (total across entire path). Default is 500.
thresh	Convergence threshold for group coordinate descent algorithm. The algorithm iterates until the change for each coefficient is less than thresh. Default is 1e-3.
criterion	Criterion used for tuning selection via cross-validation. Currently five options: MSE, AIC, BIC, EBIC, GCV. Default is BIC. See Details.
Type_Y	A vector of Type_Y prior information, having the same length with Y. Default is NULL. For family="gaussian", Type_Y is continuous. For family="binomial", Type_Y is binary. For family="poisson", Type_Y is count. If users supply a Type_Y prior information, this function will use it to estimate a GEInfo model. If Type_Y=NULL, the function will incorporate the prior information included in S_G and S_GE to realize a GEInfo model.
kappa1	A user supplied kappa1 sequence. Default is kappa1=NULL. Typical usage is to have the program compute its own kappa1 sequence. Supplying a value of kappa1 overrides this. See Details.
kappa2	A user supplied kappa2 sequence. Default is kappa2=NULL. Typical usage is to have the program compute its own kappa2 sequence. Supplying a value of kappa2 overrides this. See Details.
lam1	A user supplied lambda1 sequence. Default is lam1=NULL. Typical usage is to have the program compute its own lambda1 sequence. Supplying a value of lam1 overrides this. See Details.
lam2	A user supplied lambda2 sequence. Default is lam2=NULL. Typical usage is to have the program compute its own lambda1 sequence. Supplying a value of lam2 overrides this. See Details.
tau	A user supplied tau sequence ranging from 0 to 1. Default is tau = c(0, 0.25, 0.5, 0.75, 1). See Details.

## Details

The function contains five tuning parameters, namely kappa1, kappa2, lambda1, lambda2, and tau. kappa1 and kappa2 are used to estimate model and select variables. lambda1 and lambda2 are used to calculate the prior-predicted response based on S\_G and S\_GE. tau is used for balancing between the observed response Y and the prior-predicted response. When tau=0 and tau=1, this function realizes cross-validation for GEsgMCP and CGEInfo approaches, respectively.

In order to select the optimal tuning combination, there are five criteria available, which are MSE, AIC, BIC, GCV, and EBIC. Let L be the loss function of the model,  $MSE = L$ ,  $AIC = 2L + 2df$ ,  $BIC = 2L + \ln(n)df$ ,  $GCV = 2L/(1 - df/n)^2$ , and  $EBIC = 2L + \ln(n)df + 2df\ln(nvar)(1 -$



$\ln(n)/(2\ln(nvar)))$ . In most cases, BIC is a good choice. In the case of high dimension, EBIC criterion is recommended first, which has demonstrated satisfactory performance in high-dimensional studies.

## Value

An object of class "GEInfo" is returned, which is a list with the ingredients of the cross-validation fit.

<code>coef.all.tau</code>	A matrix of coefficients, of dimensions $(p+1)(q+1) \times \text{length}(\text{tau})$ .
<code>best.tuning</code>	A list containing the optimal tau, kappa1, and kappa2.
<code>a</code>	Coefficient vector of length q for E variables.
<code>beta</code>	Coefficient vector of length p for E variables.
<code>gamma</code>	Coefficient matrix of dimensions $p \times q$ for G-E interactions.
<code>b</code>	Coefficient vector of length $(q+1)p$ for W (G variables and G-E interactions).
<code>alpha</code>	Intercept.
<code>coef</code>	A coefficient vector of length $(q+1)(p+1)$ , including the estimates for $\alpha$ (intercept), $a$ (coefficients for all E variables), and $b$ (coefficients for all G variables and G-E interactions).
<code>nvar</code>	Number of non-zero coefficients at the best tunings.

## References

Wang X, Xu Y, and Ma S. (2019). Identifying gene-environment interactions incorporating prior information. *Statistics in medicine*, 38(9): 1620-1633. doi: [10.1002/sim.8064](https://doi.org/10.1002/sim.8064)

## Examples

```
n <- 30; p <- 4; q <- 2
E <- MASS::mvrnorm(n, rep(0,q), diag(q))
G <- MASS::mvrnorm(n, rep(0,p), diag(p))
W <- matW(E, G)
alpha <- 0; a <- seq(0.4, 0.6, length=q);
beta <- c(seq(0.2, 0.5, length=2), rep(0, p-2))
vector.gamma <- c(0.8, 0.9, 0, 0)
gamma <- matrix(c(vector.gamma, rep(0, p*q - length(vector.gamma))), nrow=p, byrow=TRUE)
mat.b.gamma <- cbind(beta, gamma)
b <- as.vector(t(mat.b.gamma))
Y <- alpha + E %*% a + W %*% b + rnorm(n, 0, 0.5)
S_G <- c(1)
S_GE <- cbind(c(1), c(1))
fit4 <- cv.GEInfo(E, G, Y, family='gaussian', S_G=S_G,
  S_GE=S_GE, lam1=0.4, lam2=0.4, kappa1 = 0.4, kappa2=0.4, tau=0.5)
```

---

GEInfo

*GEInfo approach with fixed tunings*


---

### Description

Realize to estimate the GEInfo approach at fixed tunings. It is available for Linear, Logistic, and Poisson regressions.

### Usage

```
GEInfo(
  E,
  G,
  Y,
  family,
  S_G,
  S_GE,
  kappa1,
  kappa2,
  lam1,
  lam2,
  tau,
  xi = 6,
  epsilon = 0,
  max.it = 500,
  thresh = 0.001,
  Type_Y = NULL
)
```

### Arguments

E	Observed matrix of E variables, of dimensions $n \times q$ .
G	Observed matrix of G variables, of dimensions $n \times p$ .
Y	Response variable, of length $n$ . Quantitative for family="gaussian", or family="poisson" (non-negative counts). For family="binomial" should be a factor with two levels.
family	Model type: one of ("gaussian", "binomial", "poisson").
S_G	A user supplied vector, denoting the subscript of G variables which have prior information.
S_GE	A user supplied matrix, denoting the subscript of G-E interactions which have prior information. The first and second columns of S_GE represent the subscript of G variable and the subscript of E variable, respectively. For example, S_GE = matrix( c(1, 2), ncol = 2), which indicates that the 1st G variable and the 2nd E variable have an interaction effect on Y.
kappa1	A user supplied kappa1.

kappa2	A user supplied kappa2.
lam1	A user supplied lambda1.
lam2	A user supplied lambda2.
tau	A user supplied tau.
xi	Tuning parameter of MCP penalty. Default is 6.
epsilon	Tuning parameter of Ridge penalty which shrinks on the coefficients having prior information. Default is 0.
max.it	Maximum number of iterations (total across entire path). Default is 500.
thresh	Convergence threshold for group coordinate descent algorithm. The algorithm iterates until the change for each coefficient is less than thresh. Default is 1e-3.
Type_Y	A vector of Type_Y prior information, having the same length with Y. Default is NULL. For family="gaussian", Type_Y is continuous. For family="binomial", Type_Y is binary. For family="poisson", Type_Y is a count vector. If users supply a Type_Y prior information, the function will use it to estimate a GE-Info model. If Type_Y=NULL, the function will incorporate the Type_S prior information S_G and S_GE to realize a GEInfo model.

## Details

The function contains five tuning parameters, namely kappa1, kappa2, lambda1, lambda2, and tau. kappa1 and kappa2 are used to estimate model and select variables. lambda1 and lambda2 are used to calculate the prior-predicted response based on S\_G and S\_GE. tau is used for balancing between the observed response Y and the prior-predicted response.

## Value

An object of class "GEInfo" is returned, which is a list with the ingredients of the cross-validation fit.

a	Coefficient vector of length q for E variables.
b	Coefficient vector of length (q+1)p for W (G variables and G-E interactions).
beta	Coefficient vector of length p for G variables.
gamma	Coefficient matrix of dimensions p*q for G-E interactions.
alpha	Intercept.
coef	A coefficient vector of length (q+1)*(p+1), including the estimates for $\alpha$ (intercept), $a$ (coefficients for all E variables), and $b$ (coefficients for all G variables and G-E interactions).

## References

Wang X, Xu Y, and Ma S. (2019). Identifying gene-environment interactions incorporating prior information. *Statistics in medicine*, 38(9): 1620-1633. doi: [10.1002/sim.8064](https://doi.org/10.1002/sim.8064)

### Examples

```
n <- 30; p <- 4; q <- 2
E <- MASS::mvrnorm(n, rep(0,q), diag(q))
G <- MASS::mvrnorm(n, rep(0,p), diag(p))
W <- matW(E, G)
alpha <- 0; a <- seq(0.4, 0.6, length=q);
beta <- c(seq(0.2, 0.5, length=2), rep(0, p-2))
vector.gamma <- c(0.8, 0.9, 0, 0)
gamma <- matrix(c(vector.gamma, rep(0, p*q - length(vector.gamma))), nrow=p, byrow=TRUE)
mat.b.gamma <- cbind(beta, gamma)
b <- as.vector(t(mat.b.gamma)) # coefficients of G and GE
Y <- alpha + E %*% a + W %*% b + rnorm (n, 0, 0.5)
S_G <- c(1)
S_GE <- cbind(c(1), c(1))
fit3 <- GEInfo(E, G, Y, family='gaussian', S_G=S_G,
S_GE=S_GE, kappa1 = 0.2, kappa2=0.2, lam1=0.2, lam2=0.2, tau=0.5)
```

---

matW

---

*Calculate matrix W*


---

### Description

Calculate observed matrix W for all G variables and G-E interactions. Denote  $W_j$  as the  $n \times (q+1)$  sub-matrix of W corresponding the  $j$ th G variable. The first column of  $W_j$  is the observation vector of the  $j$ th G variable, and the rest  $q$  columns of  $W_j$  are observations of G-E interactions.

### Usage

```
matW(E, G)
```

### Arguments

E	Observed matrix of E variables, of dimension $n \times q$ .
G	Observed matrix of G variables, of dimensions $n \times p$ .

### Value

A matrix of dimension  $n \times [p(q+1)]$ .

### Examples

```
n <- 30; q <- 3; p <- 5;
E <- MASS::mvrnorm (n, rep (0, q), diag (q))
G <- MASS::mvrnorm (n, rep (0, p), diag (p))
W <- matW (E, G)
```

---

plot.GEInfo	<i>Heatmap of the identification results</i>
-------------	--

---

**Description**

Plot the heatmap for all E variables, identified G variables, and their G-E interactions from a fitted (GEInfo) model.

**Usage**

```
## S3 method for class 'GEInfo'
plot(x, Gname = NULL, Ename = NULL, ...)
```

**Arguments**

x	A fitted "GEInfo" model object for which prediction is desired.
Gname	Names of all G variables. Default is NULL.
Ename	Names of all E variables. Default is NULL.
...	Other parameters.

**Value**

A Heatmap.

---

plot.PubMed	<i>Visualize the prior information</i>
-------------	--

---

**Description**

Visualize the prior counts for G variables and G-E interactions. It reports a bar chart for the top 40 G variables by prior count and a boxplot of prior counts for all G variables. For each E variables, it draws a bar chart for the corresponding top 20 G-E interactions by prior count.

**Usage**

```
## S3 method for class 'PubMed'
plot(x, G.count = NULL, GE.count = NULL, ...)
```

**Arguments**

x	A 'PubMed' object for which visualization is desired.
G.count	A numeric vector of length p, including prior counts for all G variables. Default is NULL.
GE.count	A numeric matrix of dimensions p*q, including prior counts for G-E interactions. Default is NULL.
...	Other parameters.

**Value**

The output includes bar chart for top G variables and G-E interactions by prior counts, and boxplot of prior counts for all G variables.

---

predict.GEInfo	<i>Make Predictions for a fitted model</i>
----------------	--

---

**Description**

Output predicted response values for new observations.

**Usage**

```
## S3 method for class 'GEInfo'
predict(object, Enew, Gnew, family, ...)
```

**Arguments**

object	A fitted "GEInfo" model object for which prediction is desired.
Enew	Matrix of dimensions $n_{test} \times q$ for E variables at which predictions are to be made.
Gnew	Matrix of dimensions $n_{test} \times p$ for G variables at which predictions are to be made.
family	Model type: one of ("gaussian", "binomial", "poisson").
...	Other arguments.

**Value**

Return a vector of length  $n_{test}$ , representing the fitted response value. For family= “gaussian”, the fitted values are returned; for family = “binary”, the fitted probabilities are returned; for family = “poisson”, the fitted means are returned.

---

PubMed.search	<i>Search prior counts for G variables and G-E interactions</i>
---------------	---

---

**Description**

Provide an available tool for mining prior counts for G variables and G-E interactions from PubMed database.

**Usage**

```
PubMed.search(Yname, Gname, Ename, Gnamefile)
```

**Arguments**

Yname	A user supplied character including disease name such as "breast".
Gname	A user supplied character vector including all G variable names.
Ename	A user supplied character vector including all E variable names.
Gnamefile	A newline-delimited text file uploaded by users that contains all the G variable names to be searched. Each row represents a G variable name. It provides another way to input G variable names besides from argument "Gname".

**Value**

Return the searched frequencies.

G.count	A numeric vector, presenting the prior counts for all searched G variables.
GE.count	A numeric matrix of dimensions $\text{length}(\text{Gname}) \times \text{length}(\text{Ename})$ , which presents the prior counts for all G variables (Gname) and E variables (Ename) comparisons

**Examples**

```
Yname <- c('breast')
Gname <- c('CAMP')
Ename <- c('Age')
res <- PubMed.search(Yname,Gname,Ename)
res
```

---

TypeS	<i>Construct Type_S prior information</i>
-------	---

---

**Description**

For G variables and G-E interactions, transform their prior information from counts(frequencies) into a set of significant variables (Type\_S)

**Usage**

```
TypeS(
  G.count,
  GE.count,
  eta_G = 0.95,
  eta_GE = 0.95,
  varphi_G = NULL,
  varphi_GE = NULL
)
```

**Arguments**

G.count	A numeric vector, including the prior counts (frequencies) for G variables.
GE.count	A numeric matrix, including the prior counts (frequencies) for G-E interactions.
eta_G	A probability. The (eta_G)th quantile of G.count is used as a count (frequency) threshold (denoted by varphi_G) for G variables. Default is 0.95.
eta_GE	A probability. The (eta_GE)th quantile of GE.count is used as a data-dependent count (frequency) threshold (denoted by varphi_GE) for G-E interactions. Default is 0.95.
varphi_G	A user supplied count threshold for G variables. It is used to determine which G variables will be finally included in the Type_S prior information set. Default is NULL. Typical usage is to have the program calculate the (eta_G)th quantile of G.count as the threshold. Supplying a varphi_G value will override this.
varphi_GE	A user supplied threshold value used for G-E interactions. It is used to determine which G-E interactions will be finally included in the Type_S prior information set. Default is NULL. Typical usage is to have the program calculate the (eta_GE)th quantile of GE.count as the threshold. Supplying a varphi_GE value will override this.

**Value**

The outputs include the Type\_S prior information sets for G variable and G-E interactions.

S_G	A numeric vector, denoting the Type_S set for G variables. For j in S_G, the jth G variable is suggested to be associated with the response.
S_GE	A numeric matrix, denoting the Type_S set for G-E interactions. For (l,k) in S_GE, the lth G variable and the kth E variable is suggested to have an interaction effect on the response.

**Examples**

```
G.count<-c(100,300)
GE.count<-matrix(c(130,356,8,30,87,2),nrow=2)
TypeS(G.count,GE.count)
```



# Index

CGEInfo, [2](#)

coef.GEInfo, [4](#)

cv.CGEInfo, [4](#)

cv.GEInfo, [7](#)

GEInfo, [10](#)

matW, [12](#)

plot.GEInfo, [13](#)

plot.PubMed, [13](#)

predict.GEInfo, [14](#)

PubMed.search, [14](#)

TypeS, [15](#)