# Package: BAREB (via r-universe)

October 23, 2026

**Type** Package

**Title** A Bayesian Repulsive Biclustering Model for Periodontal Data

**Version** 0.1.2

**Description** Simultaneously clusters the Periodontal diseases (PD) patients and their tooth sites after taking the patient- and site-level covariates into consideration. 'BAREB' uses the determinantal point process (DPP) prior to induce diversity among different biclusters to facilitate parsimony and interpretability. Essentially, 'BAREB' is a cluster-wise linear model based on Yuliang (2020) <doi:10.1002/sim.8536>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**LinkingTo** Rcpp, RcppArmadillo

**Imports** Rcpp

**NeedsCompilation** yes

**Depends** R (>= 3.5.0)

**Author** Yuliang Li [aut], Yanxun Xu [aut], Dipankar Bandyopadhyay [aut], John Burkardt [ctb], Qingyang Liu [aut, cre]

**Maintainer** Qingyang Liu <rh8liuqy@gmail.com>

**Repository** https://cran.r-universe.dev

**Date/Publication** 2024-12-18 16:42:16 UTC

**RemoteUrl** https://github.com/cran/BAREB

**RemoteRef** HEAD

**RemoteSha** 6112efd66d49c8e5b673c457ee564f3b333fb620

# Contents

BAREB . . . 2
const . . . 3
Init . . . 4
kernelC . . . 4
likeli_theta . . . 5
obs . . . 6
RJi . . . 6
RJi_empty . . . 8
truth . . . 9
update_RJ . . . 9
update_sigma_squre . . . 14
update_theta_beta . . . 15
update_theta_gamma . . . 15
update_w . . . 16
update_w_beta . . . 17
updateBeta . . . 18
updatec . . . 19
updateC . . . 20
updateE . . . 21
updateGamma . . . 22
updatemu . . . 24
updatemustar . . . 25
updateR . . . 25
updateZstar . . . 27

**Index** **28**

---

BAREB *The BAREB package: summary information*

---

## Description

This package implements BAREB (A Bayesian Repulsive Biclustering Model for Periodontal Data)

## Value

This is the summary page. No return value.

## Main Features

The functions in the pacakge implement BAREB. BAREB can simultaneously cluster periodontal disease patients and their tooth site after taking the patient- and site-level covariates into consideration. BAREB uses the determinantal point process prior to induce diversity among different biclusters to facilitate parsimony and interpretability. In addition, since periodontal diseases are the leading cause for tooth loss, the missing data mechanism is non-ignorable. Such nonrandom missingness is incorporated into BAREB.

### Functions

The main functions are `updateBeta`, `update.theta.beta`, `update.theta.gamma`, `updatec`, `updateE`, `updateGamma`, `updatemu`, `updatemustar`, `updateR`, `updateZstar`, `update_RJ`, `update_sigma_square`, `update_w`, and `update_w_beta`; other functions intended for direct access by the user are: `kernelC` and `updateC`. There are undocumented functions which are called by these.

### Requirements

R version >= 3.4.3. Packages `Rcpp` and `RcppArmadillo` are used so that complicated functions are implemented in C++ to speed up.

### Version

This is version 0.1.2.

### Licence

This package and its documentation are usable under the terms of the "GNU General Public License", a copy of which is distributed with the package.

### Author(s)

Yuliang Li (Dept Applied Mathematics and Statistics, Johns Hopkins University, USA) and Yanxun Xu (Dept Applied Mathematics and Statistics, Johns Hopkins University, USA) and Dipankar Bandyopadhyay (Dept Biostatistics, Virginia Commonwealth University, USA) . Please send comments, error reports, etc. to the maintainer (Yuliang) via email.

---

## `const` *const*

---

### Description

The function to compute the normalizing constant for Determinantal Point Process (DPP)

### Usage

```
const(theta, tau)
```

### Arguments

| | |
|---|---|
| `theta` | The parameter that controls how repulsive the DPP is |
| `tau` | The parameter we fixed at a large number |

### Value

The computed normalizing constant

## Init *The initial value of patient- and site-level covariates for simulation*

### Description

These data record the initial value of patient- and site-level covariates in simulation in the paper "BAREB: A Bayesian Repulsive Biclustering Model for Periodontal Data". It is obtained by simple linear regression.

The variables are:

`Init` It has two parts: the initial value of patient-level covariates, `Beta`; the initial value of site-level covariates, `Gamma`

### Usage

```
data("Init")
```

### Examples

```
# output patient level covariates
data("Init")
Init$Beta
Init$Gamma
```

## kernelC *The function to get the kernel function value*

### Description

This function take two configurations x and y, two parameters of the kernel function and returns its kernel function value.

### Usage

```
kernelC(x,y,theta,tau)
```

### Arguments

| | |
|---|---|
| `x` | a numeric vector, representing one configuration |
| `y` | a numeric vector, representing one configuration |
| `theta` | a parameter of the DPP's kernel function |
| `tau` | a parameter of the DPP's kernel function |

### Value

`kernelC(x,y,theta,tau)` returns the value of the kernel function

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

## Examples

```
x <- rnorm(5)
y <- rnorm(5) + 1
kernelC(x,y,1,1)
```

---

`likeli_theta` *likeli_theta*

---

## Description

The function computes the likelihood of the hyperparameter

## Usage

```
likeli_theta(theta, tau, Beta)
```

## Arguments

| | |
|---|---|
| `theta` | The parameter that controls how repulsive the DPP is |
| `tau` | The parameter we fixed at a large number |
| `Beta` | The matrix that records the samples |

## Value

The likelihood

## obs *The simulation observation*

### Description

These data record the simulation observation (observed value and missing indicator) and covariates in the paper "BAREB: A Bayesian Repulsive Biclustering Model for Periodontal Data".

The variables are:

| | |
|---|---|
| delta | the missing indicator matrix |
| X | the patient level covariates |
| Y | the observed (CAL) value matrix |
| Z | the site level covariates |

### Usage

```
data("obs")
```

### Examples

```
# output patient level covariates
data("obs")
obs$X
```

## RJi *Function to update the number of site level clusters for one patient level cluster via RJMCMC*

### Description

This function updates the number of site level clusters for one patient level cluster via RJMCMC. Only used for updateRJ function.

### Usage

```
RJi(w, K, Gamma, Beta, X, Y, Z, R, delta, mu, mu_star, c, sigma_square, C,
             theta, tau, m, n,q,T0, hyper_delta)
```

### Arguments

| | |
|---|---|
| w | The weights for site level clusters. Should be a vector. |
| K | The number of site level cluster for the patient level cluster. |
| Gamma | The linear coefficients for site level covariates. Should be a matrix. |
| Beta | The linear coefficients for patient level covariates. Should be a vector. |
| X | The design matrix for patient level clusters. |

| | |
|---|---|
| Y | The observed CAL value matrix |
| Z | The design matrix for site level clusters. |
| R | The current site level clustering membership |
| delta | The missing indicator matrix |
| mu | The current CAL mean vector |
| mu_star | The latent value vector for missingness model |
| c | The linear coefficients for missingness model |
| sigma_square | The current noise variance |
| C | The DPP related kernel matrices. Should be an array of 3 dimensions |
| theta | The DPP hyper-parameter for site level |
| tau | A fixed DPP hyper-parameter, which we suggest high value, say 10^5 |
| m | The number of sites |
| n | The number of patients |
| q | The number of site level covariates |
| T0 | The number of teeth |
| hyper_delta | The hyper-parameter with default value being 1 |

## Value

RJi(w, K, Gamma, Beta, X, Y, Z, R, delta, mu, mu_star, c, sigma_square, C, theta, tau, m, n,q,T0, hyper_delta) returns a list with following variables:

| | |
|---|---|
| K | The numbers of site level clusters |
| w | The weights for site level clusters |
| Gamma | Linear coefficients for site level covariates |
| R | The site level clusteirng membership |
| C | The DPP related kernel matrice |

## Author(s)

Yuliang Li

RJi_empty *Function to update the number of site level clusters for one patient level cluster via RJMCMC, when there is no patient in this cluster*

## Description

This function updates the number of site level clusters for one patient level cluster via RJMCMC, when there is no patient in this cluster. Only used for updateRJ function.

## Usage

```
RJi_empty(w, K, Gamma,  R,  C,theta, tau, m,  q, hyper_delta)
```

## Arguments

| | |
|---|---|
| w | The weights for site level clusters. Should be a vector. |
| K | The number of site level cluster for the patient level cluster. |
| Gamma | The linear coefficients for site level covariates. Should be a matrix. |
| R | The current site level clustering membership |
| C | The DPP related kernel matrices. Should be an array of 3 dimensions |
| theta | The DPP hyper-parameter for site level |
| tau | A fixed DPP hyper-parameter, which we suggest high value, say 10^5 |
| m | The number of sites |
| q | The number of site level covariates |
| hyper_delta | The hyper-parameter with default value being 1 |

## Value

RJi_empty(w, K, Gamma, R, C,theta, tau, m, q, hyper_delta) returns a list with following variables:

| | |
|---|---|
| K | The numbers of site level clusters |
| w | The weights for site level clusters |
| Gamma | Linear coefficients for site level covariates |
| R | The site level clusteirng membership |
| C | The DPP related kernel matrice |

## Author(s)

Yuliang Li

## truth *The simulation truth*

### Description

These data record the simulation truth in the paper "BAREB: A Bayesian Repulsive Biclustering Model for Periodontal Data". It includes the true simulated parameters.

The variables are:

| | |
|---|---|
| S | the number of patient level clusters |
| E | the clustering membership of patient level |
| K | the numbers of site level clusters |
| R | the site level clustering membership |
| Beta | the patient level linear coefficients |
| Gamma | the site level linear coefficients |
| mu | the underlying mean for CAL values |
| sigma_square | the variance of noise for CAL values |
| noise | the noise for CAL values |
| c | the parameter for missingness model |
| mu.star | the mean of latent values for missingness model |
| z.star | the latent values for missingness model |

### Usage

```
data("truth")
```

### Examples

```
# output true patient level clustering membership
data("truth")
truth$E

#get the details of the list
str(truth)
```

## update_RJ *update_RJ*

### Description

Update the number of site level clusters for each patient level cluster via RJMCMC

## Usage

```
update_RJ(
  w,
  K,
  Gamma,
  Beta,
  E,
  Z,
  X,
  R,
  mu,
  mu_star,
  Y,
  delta,
  c,
  sigma_square,
  C,
  S,
  theta,
  tau,
  q,
  m,
  T0,
  hyper_delta = 1
)
```

## Arguments

| | |
|---|---|
| w | The weights for site level clusters. Should be a matrix with S rows. |
| K | The number of site level cluster for each patient level cluster. Should be a vector of length S |
| Gamma | The linear coefficients for site level covariates. Should be a 3-dimensional array. |
| Beta | The linear coefficients for patient level covariates. Should be a matrix with S rows |
| E | A vector that records the current clustering membership. |
| Z | The design matrix for site level clusters. |
| X | The design matrix for patient level clusters. |
| R | The current site level clustering membership |
| mu | The current CAL mean matrix |
| mu_star | The latent value matrix for missingness model |
| Y | The observed CAL value matrix |
| delta | The missing indicator matrix |
| c | The linear coefficients for missingness model |
| sigma_square | The current noise variance |

| | |
|---|---|
| C | The DPP related kernel matrices. Should be an array of 3 dimensions |
| S | The number of patient level clusters. |
| theta | The DPP hyper-parameter for site level |
| tau | A fixed DPP hyper-parameter, which we suggest high value, say 10^5 |
| q | The number of site level covariates |
| m | The number of sites |
| T0 | The number of teeth |
| hyper_delta | The hyper-parameter with default value being 1 |

## Value

A list with following updated parameters:

| | |
|---|---|
| K | The numbers of site level clusters |
| w | The weights for site level clusters |
| Gamma | Linear coefficients for site level covariates |
| R | The site level clusteirng membership |
| C | The DPP related kernel matrice |

## Examples

```
library(BAREB)
data("obs")
X <- obs$X
Y <- obs$Y
Z <- obs$Z
delta <- obs$delta
data("truth")

set.seed(1)

n<-80
m<-168
T0<-28
q<-3
p<-3
S<-3
theta1 <- theta2 <- 5
tau <- 100000
D<-matrix(0, nrow = T0, ncol = m)
for(i in 1:T0){
  indi<-1:6
  indi<-indi+6*(i-1)
  D[i,indi]<-rep(1/6,6)
}
nu_gamma<-0.05
nu_beta <- 0.05
data("Init")
```

```
Beta0 <- Init$Beta
Gamma0 <- Init$Gamma

Niter<-10
record <- NULL
record$E<-matrix(NA,nrow = Niter, ncol = n)
record$R<-array(NA, dim = c(Niter, S, m))
record$Gamma <-array(NA,dim = c(Niter, 10, q, S))
record$Beta <- array(NA,dim = c(Niter, S, p))
record$K <- matrix(NA,nrow = Niter, ncol = S)
record$sigma_square <-rep(NA,Niter)
record$theta1<-rep(0,Niter)
record$theta2<-rep(0,Niter)
record$c<-matrix(0,nrow =  Niter,ncol = T0)
record$mu<-array(NA,dim = c(Niter,n,m))
record$w_beta <- array(NA, dim = c(Niter, S))
record$w <- array(NA, dim = c(Niter, S, 10))
set.seed(1)
E<-sample.int(S,n,TRUE)
Beta <- matrix(NA,nrow = S, ncol = p)
Beta[1,] <- Beta[2,] <- Beta[3,]  <- Beta0
Beta<- Beta + matrix(rnorm(S*p, 0, 1) , nrow = S, ncol = p)
Gamma <- array(NA,dim = c(10,q,S))
Gamma[1,,1] <- Gamma[2,,1] <- Gamma[3,,1] <- Gamma0
Gamma[,,2] <- Gamma[,,3]   <- Gamma[,,1]
Gamma <- Gamma + array(rnorm(10*q*S, 0, 5), dim = c(10, q, S))
K <- rep(3,S)
R <- matrix(NA,nrow = S, ncol = m)
R[1,]<- R[2,]<- R[3,]  <- sample.int(3,m,TRUE)
mu<-updatemu(R,Z,X,Gamma,K,Beta,E,m,n,p,q)
mu_star<-updatemustar(mu,rep(0.01,2),n,T0,D)
z_star<-updateZstar(mu_star,delta,n,T0)
sigma_square <- 10
C<-array(NA,dim = c(10,10,S))
w<-matrix(NA,nrow = S, ncol = 10)
w_beta<-rep(1/S,S)
for(i in 1:S){
 C[1:K[i],1:K[i],i]<-updateC(Gamma[1:K[i],,i],theta2,tau)
  w[i, 1:K[i]]<-rep(1/K[i],K[i])
}
c<-c(0,0.01)
start <- Sys.time()
for(iter in 1:Niter){
  c<-updatec(z_star, mu,D, 100,sigma_square, n, T0)
  mu_star<-updatemustar(mu,c,n,T0,D)
  z_star<-updateZstar(mu_star,delta,n, T0)
  w <- update_w(K, R, S)
  R <- updateR(w, Gamma, Beta,
               Y, Z, delta, mu, mu_star, c[2], S,
               sigma_square, K, E, X,
               m, n, q, p, T0)
  for(i in unique(E)){
    ind<-sort(unique(R[i,]))
```

```
  KK<-length(ind)
  Gamma_temp<-Gamma[ind,,i]
  Gamma[,,i]<-NA
  Gamma[1:KK,,i]<-Gamma_temp
  w_temp<-w[i,ind]
  w_temp<-w_temp/sum(w_temp)
  w[i,]<-NA
  w[i,1:KK]<-w_temp
  for(k in 1:KK){
    R[i,which(R[i,]==ind[k])]<-k
  }
  K[i]<-KK
}
mu<-updatemu(R,Z,X,Gamma,K,Beta,E,m,n,p,q)
mu_star<-updatemustar(mu,c,n,T0,D)
step <- array(rnorm(max(K) * S *q, 0, nu_gamma),dim=c( max(K), q,S))
run<- array(runif(max(K) * S *q, 0, 1),dim=c( max(K), q,S))
A<-updateGamma(X,Y, Z, delta, Beta, Gamma, E, R, S, K , mu, mu_star, sigma_square, rep(c,T0),
               step,  run,  n,  m, T0,  p,  q,  D,theta2,  tau)
Gamma<-A$Gamma
mu<-A$mu
mu_star<-A$mustar
for(i in 1:S){
  if(K[i]==1){
    Gammai = t(as.matrix(Gamma[1:K[i],,i]))
    C[1:K[i],1:K[i],i]<-updateC(Gammai,theta2,tau)
  }
  else{
    C[1:K[i],1:K[i],i]<-updateC(Gamma[1:K[i],,i],theta2,tau)
  }
}
A<-update_RJ(w, K, Gamma,Beta, E,
             Z, X, R, mu, mu_star, Y, delta, c,sigma_square, C,
             S, theta2, tau, q, m, T0)
K<-A$K
w<-A$w
Gamma<-A$Gamma
R<-A$R
C<-A$C
mu<-updatemu(R,Z,X,Gamma,K,Beta,E,m,n,p,q)
mu_star<-updatemustar(mu,rep(c,T0),n,T0,D)
C_beta<-updateC(Beta,theta1,tau)
step<-matrix(rnorm(S*p,0,nu_beta),nrow = S)
runif<-matrix(runif(S*p,0,1),nrow = S)
A<- updateBeta( X,Y, Z, delta,
           Beta, Gamma, E,R,S,K,mu_star, mu,sigma_square,rep(c,T0), C_beta, step,runif,
                n, m, T0,  p,  q,  D,
                theta1, tau)
Beta<-A$Beta
mu<-A$mu
mu_star<-A$mustar
record$Beta[iter,,]<-Beta
A<-updateE( Beta,Gamma, w_beta, X,Y,Z,delta,E, R, S,K, mu, mu_star,sigma_square,rep(c,T0),
```

```
            n,  m,  T0,  p,  q,  D)
  E<-A$E
  mu<-A$mu
  mu_star<-A$mustar
  K<-A$Ds
  w_beta<- update_w_beta(S, w_beta, E)
  theta1<-update_theta_beta(theta1,tau,Beta)
  theta2<-update_theta_gamma(theta2,tau,Gamma,S,K)
  sigma_square <- update_sigma_squre(Y,mu)
}
```

---

`update_sigma_squre` *update_sigma_squre*

---

## Description

Update the noise variance

## Usage

```
update_sigma_squre(Y, mu, a = 1, b = 1)
```

## Arguments

| | |
|---|---|
| `Y` | The CAL observation matrix, with missing values |
| `mu` | Current estimated mean matrix for CAL |
| `a` | The hyper-parameter with default value being 1 |
| `b` | The hyper-parameter with default value being 1 |

## Value

Updated noise variance

## See Also

update_RJ for a complete example for all functions in this package.

update_theta_beta *update_theta_beta*

## Description

Update the DPP hyper-parameter for patient level

## Usage

```
update_theta_beta(theta, tau, Beta, sig = 10)
```

## Arguments

| | |
|---|---|
| theta | The DPP hyper-parameter for patient level |
| tau | A fixed DPP hyper-parameter, which we suggest high value, say 10^5 |
| Beta | The linear coefficients for patient level covariates. Should be a matrix with S rows |
| sig | The hyper-parameter with default value being 10 |

## Value

updated DPP hyper-parameter for patient level

## See Also

update_RJ for a complete example for all functions in this package.

update_theta_gamma *update_theta_gamma*

## Description

Update the DPP hyper-parameter for site level

## Usage

```
update_theta_gamma(theta, tau, Gamma, S, Ds, sig = 10)
```

## Arguments

| | |
|---|---|
| theta | The DPP hyper-parameter for patient level |
| tau | A fixed DPP hyper-parameter, which we suggest high value, say 10^5 |
| Gamma | The linear coefficients for site level covariates. Should be a 3-dimensional array. |
| S | The number of patient level clusters |
| Ds | The number for site level clusters for each patient level cluster |
| sig | The hyper-parameter with default value being 10 |

## Value

updated DPP hyper-parameter for site level

## See Also

update_RJ for a complete example for all functions in this package.

---

`update_w` *update_w*

---

## Description

This function updates the weights for site level clusters

## Usage

```
update_w(K, R, S, hyper_delta = 1)
```

## Arguments

| | |
|---|---|
| `K` | The number of site level cluster for each patient level cluster. Should be a vector of length `S` |
| `R` | The current site level clustering membership. Should be a matrix with `S` rows. |
| `S` | The number of patient level clusters. |
| `hyper_delta` | The hyper-parameter with default value being 1 |

## Details

It returns a matrix with 10 columns. For example, first patient cluster has 2 site level clusters. The first row's first 2 elements give the weights for site level clusters in patient cluster 1. Last 8 elements are NA's

## Value

The updated weights for site level clusters. Should be a matrix with `S` rows.

## See Also

update_RJ for a complete example for all functions in this package.

## Examples

```
#Suppose we know the number of patient level cluster is 2,
#one has 2 site level clusters and one has 3.
#Use the default value, 1, for hyper-parameter
update_w(K=c(2,3),R=matrix(c(1,2,2,1,1,2,3,2),nrow=2,byrow=TRUE), S=2)

#To change the hyper-parameter to, for example 2
update_w(K=c(2,3),R=matrix(c(1,2,2,1,1,2,3,2),nrow=2,byrow=TRUE), S=2, hyper_delta = 2)
```

update_w_beta *update_w_beta*

## Description

This function updates the weights for each patient level cluster

## Usage

```
update_w_beta(S, E, hyper_delta = 1)
```

## Arguments

| | |
|---|---|
| S | The number of patient level clusters |
| E | A vector that records the current clustering membership. |
| hyper_delta | The hyper-parameter with default value being 1 |

## Value

The updated weights for each patient level cluster

## See Also

update_RJ for a complete example for all functions in this package.

## Examples

```
#Suppose we know the number of patient level cluster is 4
#Suppose the current clustering membership indicates 3 patients in cluster 1,
#2 patients in cluster 2, 3 patinets in cluster 3, 1 patient in cluster 4
#Use the default value, 1, for hyper-parameter
update_w_beta(S=4,E=c(1,1,1,2,2,3,3,3,4))


#To change the hyper-parameter to, for example 2
update_w_beta(S=4,E=c(1,1,1,2,2,3,3,3,4),hyper_delta = 2)
```

updateBeta *Function to update patient level linear coefficients in the BAREB model*

## Description

This function takes current parameters and observed data, gives an updated patient level linear coefficients.

## Usage

```
updateBeta(X, Y, Z,
    delta, Beta, Gamma, E, R,
    S, Ds, mustar, mu,
    sigma, c, C,
    step, runif,
    n, m, T0, p, q, D,
    theta,tau)
```

## Arguments

| | |
|---|---|
| X | the patient level covariate matrix |
| Y | the CAL observation matrix, with missing values |
| Z | the site level covariate matrix |
| delta | the missing indicator matrix, with 1 means missing |
| Beta | current patient level linear coefficients matrix |
| Gamma | current site level linear coefficients array |
| E | current patient level clustering vector |
| R | current site level clustering matrix |
| S | number of patient level clusters |
| Ds | a vector recording numbers of site level clusters |
| mustar | current matrix of latent value for missingness model |
| mu | current estimated mean matrix for CAL |
| sigma | current estimated noise variance |
| c | current c for missingness model. It is a vector |
| C | current kernel matrix for DPP |
| step | a matrix of steps for M-H |
| runif | a matrix of uniform random variables for deciding whether to accept new proposed point in M-H |
| n | number of patients |
| m | number of sites |

| | |
|---|---|
| T0 | number of teeth |
| p | dimension of patient level covariates |
| q | dimension of site level covariates |
| D | the D matrix in the paper |
| theta | parameter for DPP |
| tau | parameter for DPP |

## Value

updateBeta(X, Y, Z, delta, Beta, Gamma, E, R, S, Ds, mustar, mu, sigma, c, C, step, runif, n, m, T0, p, q, D, theta,tau) returns a list with following variables:

| | |
|---|---|
| C | the updated kernel matrix computed by updated Beta |
| Beta | the updated patient level linear coefficients |
| mu | the updated mu computed by updated Beta |
| mustar | the updated mustar computed by updated Beta |

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

---

# updatec *Function to update c in missingess model*

---

## Description

This function takes current parameters, gives updated c in missingess model. Note a double type of value is returned

## Usage

```
updatec(Zstar, mu,  D, sigmac,sigma_square, n, T0)
```

## Arguments

| | |
|---|---|
| Zstar | generated latent value for missingness model |
| mu | current estimated mean matrix for CAL |
| D | the D matrix in the paper |
| sigmac | hyperparamter for c (variance) |
| sigma_square | current estimated noise variance |
| n | number of patients |
| T0 | number of teeth |

## Value

`updatec(Zstar, mu, D, sigmac, n, T0)` returns the updated c in missingess model.

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

---

`updateC` *Function to obtain the kernel matrix of the determinantal point process*

---

## Description

This function takes a matrix and two parameters of kernel function for the determinantal point process (DPP) and gives the kernel matrix for that DPP.

## Usage

```
updateC(Z,theta,tau)
```

## Arguments

| | |
|---|---|
| `Z` | a matrix, whose rows stand for configurations of the DPP. |
| `theta` | a parameter of the DPP's kernel function |
| `tau` | a parameter of the DPP's kernel function |

## Value

`updateC(Z,theta,tau)` returns the kernel matrix

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

## Examples

```
Z <- matrix(rnorm(15), nrow = 5)
updateC(Z,1,1)
```

updateE *Function to update patient level clustering in the BAREB model*

## Description

This function takes current parameters and observed data, gives an updated patient level clustering.

## Usage

```
updateE( Beta, Gamma,w,
    X, Y, Z, delta,
    E, R, S, Ds,
    mu, mustar,
    sigma, c,
    n, m, T0, p, q, D)
```

## Arguments

| | |
|---|---|
| Beta | current patient level linear coefficients matrix |
| Gamma | current site level linear coefficients array |
| w | current patient level clustering prior prob, a vector |
| X | the patient level covariate matrix |
| Y | the CAL observation matrix, with missing values |
| Z | the site level covariate matrix |
| delta | the missing indicator matrix, with 1 means missing |
| E | current patient level clustering vector |
| R | current site level clustering matrix |
| S | number of patient level clusters |
| Ds | a vector recording numbers of site level clusters |
| mu | current estimated mean matrix for CAL |
| mustar | current matrix of latent value for missingness model |
| sigma | current estimated noise variance |
| c | current c for missingness model. It is a vector |
| n | number of patients |
| m | number of sites |
| T0 | number of teeth |
| p | dimension of patient level covariates |
| q | dimension of site level covariates |
| D | the D matrix in the paper |

## Value

updateE( Beta, Gamma,w, X, Y, Z, delta,E, R, S, Ds, mu, mustar, sigma, c, n, m, T0, p, q, D) returns a list with following variables:

| | |
|---|---|
| E | the updated patient level clustering |
| Ds | new vector recording the numbers of site level clusters |
| mu | the updated mu computed by updated E |
| mustar | the updated mustar computed by updated E |

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

---

# updateGamma *Function to update site level linear coefficients in the BAREB model*

---

## Description

This function takes current parameters and observed data, gives an updated site level linear coefficients.

## Usage

```
updateGamma(X, Y, Z, delta,
    Beta, Gamma, E, R,
    S, Ds, mu, mustar,
    sigma, c, step, runif,
    n, m, T0, p, q, D, theta,tau)
```

## Arguments

| | |
|---|---|
| X | the patient level covariate matrix |
| Y | the CAL observation matrix, with missing values |
| Z | the site level covariate matrix |
| delta | the missing indicator matrix, with 1 means missing |
| Beta | current patient level linear coefficients matrix |
| Gamma | current site level linear coefficients array |
| E | current patient level clustering vector |
| R | current site level clustering matrix |
| S | number of patient level clusters |

| | |
|---|---|
| Ds | a vector recording numbers of site level clusters |
| mu | current estimated mean matrix for CAL |
| mustar | current matrix of latent value for missingness model |
| sigma | current estimated noise variance |
| c | current c for missingness model. It is a vector |
| step | an array of steps for M-H |
| runif | an array of uniform random variables for deciding whether to accept new proposed point in M-H |
| n | number of patients |
| m | number of sites |
| T0 | number of teeth |
| p | dimension of patient level covariates |
| q | dimension of site level covariates |
| D | the D matrix in the paper |
| theta | parameter for DPP |
| tau | parameter for DPP |

## Value

updateGamma(X, Y, Z, delta, Beta, Gamma, E, R, S, Ds, mu, mustar, sigma, c, step, runif, n, m, T0, p, q, D, theta,tau) returns a list with following variables:

| | |
|---|---|
| Gamma | the updated site level linear coefficients |
| mu | the updated mu computed by updated Gamma |
| mustar | the updated mustar computed by updated Gamma |

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

`updatemu` *Function to update estimated mean CAL values based on current parameters*

## Description

This function takes current parameters, gives an estimated mean CAL values.

## Usage

```
updatemu(R, Z, X, Gamma, K, Beta, E,  m,n,p, q)
```

## Arguments

| | |
|---|---|
| `R` | current site level clustering matrix |
| `Z` | the site level covariate matrix |
| `X` | the patient level covariate matrix |
| `Gamma` | current site level linear coefficients array |
| `K` | a vector recording numbers of site level clusters |
| `Beta` | current patient level linear coefficients matrix |
| `E` | current patient level clustering vector |
| `m` | number of sites |
| `n` | number of patients |
| `p` | dimension of patient level covariates |
| `q` | dimension of site level covariates |

## Value

`updatemu(R, Z, X, Gamma, K, Beta, E, m,n,p, q)` returns the updated estimated mean CAL matrix.

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

## updatemustar *Function to update mean latent values for missingness model*

### Description

This function takes current parameters, gives updated mean latent values for missingness model

### Usage

```
updatemustar(mu, c, n, T0, D)
```

### Arguments

| | |
|---|---|
| mu | current estimated mean matrix for CAL |
| c | current c for missingness model |
| n | number of patients |
| T0 | number of teeth |
| D | the D matrix in the paper |

### Value

`updatemustar(mu, c, n, T0, D)` returns the updated mean latent values for missingness model.

### Author(s)

Yuliang Li

### See Also

update_RJ for a complete example for all functions in this package.

## updateR *Function to update site level clustering in the BAREB model*

### Description

This function takes current parameters and observed data, gives an updated site level clustering.

### Usage

```
updateR( w ,Gamma,Beta,
    Y, Z, delta,
    mu, mu_star,
    c, S, sigma_square,
    K, E, X,
    m,n, q, p, T0)
```

## Arguments

| | |
|---|---|
| w | current site level clustering prior prob, a matrix |
| Gamma | current site level linear coefficients array |
| Beta | current patient level linear coefficients matrix |
| Y | the CAL observation matrix, with missing values |
| Z | the site level covariate matrix |
| delta | the missing indicator matrix, with 1 means missing |
| mu | current estimated mean matrix for CAL |
| mu_star | current matrix of latent value for missingness model |
| c | current c for missingness model |
| S | number of patient level clusters |
| sigma_square | current estimated noise variance |
| K | a vector recording numbers of site level clusters |
| E | current patient level clustering vector |
| X | the patient level covariate matrix |
| m | number of sites |
| n | number of patients |
| p | dimension of patient level covariates |
| q | dimension of site level covariates |
| T0 | number of teeth |

## Value

updateR( w ,Gamma,Beta, Y, Z, delta, mu, mu_star, c, S, sigma_square, K, E, X, m,n, p, q, T0) returns the updated site level clustering.

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

`updateZstar` *Function to generate new latent values for missingness model*

## Description

This function takes current parameters, gives updated c in missingess model. Note a double type of value is returned

## Usage

```
updateZstar(mu_star, delta, n, T0)
```

## Arguments

| | |
|---|---|
| `mu_star` | current matrix of latent value for missingness model |
| `delta` | the missing indicator matrix, with 1 means missing |
| `n` | number of patients |
| `T0` | number of teeth |

## Value

`updateZstar(mu_star, delta, n, T0)` returns a matrix of new generated latent values.

## Author(s)

Yuliang Li

## See Also

update_RJ for a complete example for all functions in this package.

# Index

∗ **datasets**
Init, 4
obs, 6
truth, 9

BAREB, 2

const, 3

Init, 4

kernelC, 4

likeli_theta, 5

obs, 6

RJi, 6
RJi_empty, 8

truth, 9

update_RJ, *5*, 9, *14–17*, *19*, *20*, *22–27*
update_sigma_squre, 14
update_theta_beta, 15
update_theta_gamma, 15
update_w, 16
update_w_beta, 17
updateBeta, 18
updateC, 20
updatec, 19
updateE, 21
updateGamma, 22
updatemu, 24
updatemustar, 25
updateR, 25
updateZstar, 27