Package 'BUScorrect'

May 20, 2024

Type Package

Title Batch Effects Correction with Unknown Subtypes

Version 1.23.0

Date 2018-06-07

Author Xiangyu Luo <xyluo1991@gmail.com>, Yingying Wei <yweicuhk@gmail.com>

Maintainer Xiangyu Luo <xyluo1991@gmail.com>

Depends R (>= 3.5.0)

Description High-throughput experimental data are accumulating exponentially in public databases. However, mining valid scientific discoveries from these abundant resources is hampered by technical artifacts and inherent biological heterogeneity. The former are usually termed ``batch effects,'' and the latter is often modelled by ``subtypes.'' The R package BUScorrect fits a Bayesian hierarchical model, the Batch-effects-correction-with-Unknown-Subtypes model (BUS), to correct batch effects in the presence of unknown subtypes. BUS is capable of (a) correcting batch effects explicitly, (b) grouping samples that share similar characteristics into subtypes, (c) identifying features that distinguish subtypes, and (d) enjoying a linear-order computation complexity.

Imports gplots, methods, grDevices, stats, SummarizedExperiment

VignetteBuilder knitr

Suggests BiocStyle, knitr, RUnit, BiocGenerics

biocViews GeneExpression, StatisticalMethod, Bayesian, Clustering, FeatureExtraction, BatchEffect

LazyLoad yes

License GPL (>= 2)

git_url https://git.bioconductor.org/packages/BUScorrect

git_branch devel

git_last_commit c3ba6b2

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-05-20

Contents

`BUScorrect-package` *Batch Effects Correction with Unknown Subtypes*

Description

High-throughput experimental data are accumulating exponentially in public databases. However, mining valid scientific discoveries from these abundant resources is hampered by technical artifacts and inherent biological heterogeneity. The former are usually termed "batch effects," and the latter is often modelled by "subtypes." The R package BUScorrect fits a Bayesian hierarchical model, the Batch-effects-correction-with-Unknown-Subtypes model (BUS), to correct batch effects in the presence of unknown subtypes. BUS is capable of (a) correcting batch effects explicitly, (b) grouping samples that share similar characteristics into subtypes, (c) identifying features that distinguish subtypes, and (d) enjoying a linear-order computation complexity.

Details

BUS is capable of (a) correcting batch effects explicitly, (b) grouping samples that share similar characteristics into subtypes, (c) identifying features that distinguish subtypes, (d) allowing the number of subtypes to vary from batch to batch, (e) integrating batches from different platforms, and (f) enjoying a linear-order computation complexity.

Author(s)

Xiangyu Luo <xyluo1991@gmail.com>, Yingying Wei <yweicuhk@gmail.com>

Maintainer: Xiangyu Luo <xyluo1991@gmail.com>

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
##############################################################################
#Generate Simulation Data
##############################################################################
rm(list = ls(all = TRUE))
set.seed(123)

B <- 3
#total number of batches

K <- 3
#total number of subtypes

G <- 3000
#total number of genes

pi <- matrix(NA, B, K)
# pi[b,k] stands for the proportion of the kth subtype in the bth batch

pi[1, ] <- c(0.2, 0.3, 0.5)
pi[2, ] <- c(0.4, 0.2, 0.4)
pi[3, ] <- c(0.3, 0.4, 0.3)


#total number of samples in each bacth.
n_vec <- rep(NA, B)

#n_vec[b] represents the total number of samples in batch b.
n_vec <- c(100, 110, 120)

#Data list
example_Data <- list()

#baseline expression level
alpha <- rep(2, G)



#subtype effect
mu <- matrix(NA, G, K)
#subtype effect, mu[g,k] stands for the kth-subtype effect of gene g
```

```
mu[ ,1] <- 0
#the first subtype is taken as the baseline subtype
#the subtype effect of subtype 1 is set to zero

mu[ ,2] <- c(rep(2,G/20), rep(0,G/20),rep(0, G-G/20-G/20))
mu[ ,3] <- c(rep(0,G/20), rep(2,G/20),rep(0, G-G/20-G/20))

#batch effect
gamma <- matrix(NA, B, G)
#'location' batch effect of gene g in batch b
gamma[1, ] <- 0
#the first batch is taken as the reference batch without batch effects
#the batch effect of batch 1 is set to zero
gamma[2, ] <- c(rep(3,G/5),rep(2,G/5),rep(1,G/5),
rep(2,G/5),rep(3,G/5))
gamma[3, ] <- c(rep(1,G/5),rep(2,G/5),rep(3,G/5),
rep(2,G/5),rep(1,G/5))

sigma_square <- matrix(NA, B,G)
#sigma_square[b,g] denotes the error variance of gene g in batch b.

sigma_square[1,] <- rep(0.1, G)
sigma_square[2,] <- rep(0.2, G)
sigma_square[3,] <- rep(0.15, G)


Z <- list()
#subtype indicator. Z[b,j] represents the subtype of sample j in batch b

Z[[1]] <- as.integer(c(rep(1,floor(pi[1,1]*n_vec[1])),rep(2,floor(pi[1,2]*n_vec[1])),
 rep(3,floor(pi[1,3]*n_vec[1]))))

Z[[2]] <- as.integer(c(rep(1,floor(pi[2,1]*n_vec[2])),rep(2,floor(pi[2,2]*n_vec[2])),
rep(3,floor(pi[2,3]*n_vec[2]))))

Z[[3]] <- as.integer(c(rep(1,floor(pi[3,1]*n_vec[3])),rep(2,floor(pi[3,2]*n_vec[3])),
rep(3,floor(pi[3,3]*n_vec[3]))))


for(b in 1:B){ #generate data
num <- n_vec[b]
example_Data[[b]] <-  sapply(1:num, function(j){
tmp <- alpha + mu[ ,Z[[b]][j]] + gamma[b, ] +
rnorm(G, sd = sqrt(sigma_square[b, ]))

tmp
})

}

#############################################################################
#Apply the BUSgibbs Function
```

```
##############################################################################
set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
summary(BUSfits)


## Not run:
#estimated subtypes
est_subtypes <- Subtypes(BUSfits)

#estimated subtype effects
est_subtype_effects <- subtype_effects(BUSfits)

#estimated location batch effects
est_location_batch_effects <- location_batch_effects(BUSfits)

#BIC
BIC_val <- BIC(BUSfits)

#adjusted expression values
adjusted_data <- adjusted_values(BUSfits, Data)

#estimate intrinsic gene indicators by using the default postprob_DE_threshold = 0.5
est_L <- estimate_IG_indicators(BUSfits, postprob_DE_threshold = 0.5)

#obtain the intrinsic gene indicators
intrinsic_gene_indices <- IG_index(est_L)
## End(Not run)
```

`adjusted_values` *Obtain adjusted expression values from the output by BUSgibbs*

Description

Call the function to obtain the corrected expression values with batch effects removed from the original input data.

Usage

```
adjusted_values(BUSfits, original_data)
```

Arguments

`BUSfits`	The BUSfits object output by the function `BUSgibbs`.
`original_data`	The original gene expression data list with length equal to the batch number. Each of its element is a gene expression matrix for a specific batch, in which each row corresponds to a gene and each column corresponds to a sample.

Value

`adjusted_data` An R list with length equal to the batch number. Each of its element is a corrected gene expression matrix for a specific batch, in which each row corresponds to a gene and each column corresponds to a sample.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect

set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)

adjusted_data <- adjusted_values(BUSfits, example_Data)
```

---

baseline_expression_values

Obtain the baseline expression values from the output by BUSgibbs

---

Description

Call the function to obtain the baseline expression values for subtype 1 from the output by BUSgibbs.

Usage

```
baseline_expression_values(BUSfits)
```

Arguments

BUSfits	The BUSfits object output by the function BUSgibbs.

Value

est_baseline_expression_values
: The estimated baseline expression values for subtype 1, a vector with length equal to the gene number.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect

set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
est_baseline_expression_values <- baseline_expression_values(BUSfits)
```

BIC_BUS *Obtain BIC form the output by BUSgibbs*

Description

The BIC value can be used to determine the subtype number if it is unknown to the users.

Usage

```
BIC_BUS(BUSfits)
```

Arguments

BUSfits	The BUSfits object from the function BUSgibbs.

Value

BIC_val	The BIC value for the BUS model with the subtype number being n.subtypes, the input subtype number for the BUSgibbs function that generates the BUSfits object.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
```

```
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect

set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
BIC_val <- BIC_BUS(BUSfits)
```

BUSexample_data *A simulated data set*

Description

A simulated data set for demonstrating how to use the BUScorrect package

Examples

```
## Not run:
#This data set is simulated according to the following R code
rm(list = ls(all = TRUE))
set.seed(123456)

B <- 3
#total number of batches

K <- 3
#total number of subtypes

G <- 2000
#total number of genes

pi <- matrix(NA, B, K)
# pi[b,k] stands for the proportion of kth subtype in bth batch

pi[1, ] <- c(0.2, 0.3, 0.5)
pi[2, ] <- c(0.4, 0.2, 0.4)
pi[3, ] <- c(0.3, 0.4, 0.3)


#total number of samples in each bacth.
n_vec <- rep(NA, B)

#n_vec[b] represents the total number of samples in batch b.
n_vec <- c(70, 80, 70)

#Data list
example_Data <- list()


#baseline expression level
alpha <- rep(2, G)
```

```
#subtype effect
mu <- matrix(NA, G, K)
#subtype effect, mu[g,k] stands for the kth-subtype effect of gene g

mu[ ,1] <- 0
#the first subtype is taken as the baseline subtype
#the subtype effect of subtype 1 is set to zero

mu[ ,2] <- c(rep(2,G/20), rep(0,G/20),rep(0, G-G/20-G/20))
mu[ ,3] <- c(rep(0,G/20), rep(2,G/20),rep(0, G-G/20-G/20))

#batch effect
gamma <- matrix(NA, B, G)
#'location' batch effect of gene g in batch b
gamma[1, ] <- 0
#the first batch is taken as the reference batch without batch effects
#the batch effect of batch 1 is set to zero
gamma[2, ] <- c(rep(3,G/5),rep(2,G/5),rep(1,G/5),
rep(2,G/5),rep(3,G/5))
gamma[3, ] <- c(rep(1,G/5),rep(2,G/5),rep(3,G/5),
rep(2,G/5),rep(1,G/5))

sigma_square <- matrix(NA, B,G)
#sigma_square[b,g] denotes the error variance of gene g in batch b.

sigma_square[1,] <- rep(0.1, G)
sigma_square[2,] <- rep(0.2, G)
sigma_square[3,] <- rep(0.15, G)


Z <- list()
#subtype indicator. Z[b,j] represents the subtype of sample j in batch b

Z[[1]] <- as.integer(c(rep(1,floor(pi[1,1]*n_vec[1])),rep(2,floor(pi[1,2]*n_vec[1])),
 rep(3,floor(pi[1,3]*n_vec[1]))))

Z[[2]] <- as.integer(c(rep(1,floor(pi[2,1]*n_vec[2])),rep(2,floor(pi[2,2]*n_vec[2])),
rep(3,floor(pi[2,3]*n_vec[2]))))

Z[[3]] <- as.integer(c(rep(1,floor(pi[3,1]*n_vec[3])),rep(2,floor(pi[3,2]*n_vec[3])),
rep(3,floor(pi[3,3]*n_vec[3]))))


for(b in 1:B){ #generate data
num <- n_vec[b]
example_Data[[b]] <-  sapply(1:num, function(j){
tmp <- alpha + mu[ ,Z[[b]][j]] + gamma[b, ] +
rnorm(G, sd = sqrt(sigma_square[b, ]))

tmp
```

```
})

}
BUSexample_data <- example_Data

## End(Not run)
```

BUSgibbs *Batch Effects Correction and Subtype Discovery using Gibbs Sampler*

Description

The function "BUSgibbs" stands for fitting the Batch effects correction with Unknown Subtypes model (BUS) with the Gibbs Sampler. BUS is capable of (a) correcting batch effects explicitly, (b) grouping samples that share similar characteristics into subtypes, (c) identifying features that distinguish subtypes, and (d) enjoying a linear-order computation complexity. After correcting the batch effects with BUS, the corrected value can be used for other analysis as if all samples are measured in a single batch. We adopt the Bayesian framework and use the Gibbs sampler to conduct posterior inference for the BUS model.

Usage

```
BUSgibbs(Data, n.subtypes, n.iterations = 500, n.records = floor(n.iterations/2),
 hyperparameters = c(1, sqrt(5), sqrt(5), 2, 2, 1, 2, 0.005, 1, 3, 10),
 showIteration = TRUE)
```

Arguments

`Data`	`Data` is either an R list or a SummarizedExperiment object. If `Data` is an R list, it has the length equal to the batch number. The bth element of `Data` is the gene expression matrix in batch b, where the rows correspond to genes and the columns represent samples. If `Data` is a SummarizedExperiment object, `assays(Data)` must contain a gene expression matrix named "GE_matr", where one row represents a gene and one column corresponds to a sample. `colData(Data)` must include a vector named "Batch", which indicates the batch information for each sample.
`n.subtypes`	`n.subtypes` is the subtype number, which needs to be specified by the user.
`n.iterations`	`n.iterations` is the iteration number used in the Gibbs sampler. The default is 500.
`n.records`	The posterior samples in the last `n.records` iterations are used to conduct posterior inference. The default is one half of `n.iterations`.
`hyperparameters`	`hyperparameters` is a hyper-parameter vector with 11 elements used in the Gibbs sampler. The first element to the last element of `hyperparameters` are as follows. `eta_alpha`: the mean of the normal prior for alpha_g; `tau_alpha`: the

standard deviation of the normal prior for alpha_g; tau_gamma: the standard deviation of the normal prior for gamma; alpha_par: the parameter of the Dirichlet prior for subtype proportions; a_inv_gamma: the shape of the gamma prior for 1/sigma^2_bg; b_inv_gamma: the rate of the gamma prior for 1/sigma^2_bg; a_tau0: the shape of the gamma prior for 1/tau^2_mu 0; b_tau0: the rate of the gamma prior for 1/tau^2_mu 0; (a_p, b_p): parameters in the beta prior for p; tau_{mu1}: the standard deviation of the normal prior of mu_gk (k >= 2) when the gene expression level in subtype k is different from that in subtype one.

showIteration If TRUE, the iteration number will be displayed when conducting Gibbs sampler. The default is TRUE.

Details

Notice that Data, the input original gene expression values, are organized in the format of an R list with length equal to the batch number. Its bth element Data[[b]] is a G by n_b matrix, where G is the gene number and n_b is the sampler size of batch b.

Value

L_PosterSamp The posterior samples of the intrinsic gene indicators. The return is a G by K-1 by n.records array, where G is the gene number, K is the subtype number, and n.records is the number for recorded iterations.

Subtypes The estimated subtypes, an R list with length equal to the batch number B, in which Subtypes[[b]] is an integer vector showing the subtype indicators of samples in batch b.

tau_mu_zero The estimated tau_mu 0, which is the prior normal distribution's standard deviation of the subtype effects when there is no differential expression.

p The estimated proportion of intrinsic genes.

pi The estimated subtype proportions across batches, a B by K matrix, whose [b,k] element is the estimated proportion of subtype k in the batch b.

alpha The estimated baseline expression levels, a G-dimension vector, whose gth element is the estimated mean gene expression level of gene g in subtype one.

gamma_PosterSamp The posterior samples of location batch effects, a G by B by n.records array.

gamma The estimated location batch effects, a G by B matrix, where gamma_gb is the "location" batch effect on gene g in the batch b. Note that the first column is zero as the first batch is taken as the reference batch without batch effects.

sigma_sq_PosterSamp The posterior samples of variances, a G by B by n.records array.

sigma_sq The estimated variance, a G by B matrix, whose [g,b] element is the variance of gene g's expression in the batch b.

mu_PosterSamp The posterior samples of subtype effects, a G by K by n.records array.

mu The estimated subtype effects, a G by K matrix, whose [g,k] element is the subtype k effect on gene g. Note that the first column is zero as the fist subtype is taken as the baseline subtype.

BIC the BIC value when K = n.subtypes, which is used to determine the subtype number by varying the value of K.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
################################################################################
#Generate Simulation Data
################################################################################
rm(list = ls(all = TRUE))
set.seed(123)

B <- 3
#total number of batches

K <- 3
#total number of subtypes

G <- 3000
#total number of genes

pi <- matrix(NA, B, K)
# pi[b,k] stands for the proportion of kth subtype in bth batch

pi[1, ] <- c(0.2, 0.3, 0.5)
pi[2, ] <- c(0.4, 0.2, 0.4)
pi[3, ] <- c(0.3, 0.4, 0.3)


#total number of samples in each bacth.
n_vec <- rep(NA, B)

#n_vec[b] represents the total number of samples in batch b.
n_vec <- c(100, 110, 120)

#Data list
example_Data <- list()

#baseline expression level
alpha <- rep(2, G)



#subtype effect
mu <- matrix(NA, G, K)
#subtype effect, mu[g,k] stands for the kth-subtype effect of gene g

mu[ ,1] <- 0
```

```
#the first subtype is taken as the baseline subtype
#the subtype effect of subtype 1 is set to zero

mu[ ,2] <- c(rep(2,G/20), rep(0,G/20),rep(0, G-G/20-G/20))
mu[ ,3] <- c(rep(0,G/20), rep(2,G/20),rep(0, G-G/20-G/20))

#batch effect
gamma <- matrix(NA, B, G)
#'location' batch effect of gene g in batch b
gamma[1, ] <- 0
#the first batch is taken as the reference batch without batch effects
#the batch effect of batch 1 is set to zero
gamma[2, ] <- c(rep(3,G/5),rep(2,G/5),rep(1,G/5),
rep(2,G/5),rep(3,G/5))
gamma[3, ] <- c(rep(1,G/5),rep(2,G/5),rep(3,G/5),
rep(2,G/5),rep(1,G/5))

sigma_square <- matrix(NA, B,G)
#sigma_square[b,g] denotes the error variance of gene g in batch b.

sigma_square[1,] <- rep(0.1, G)
sigma_square[2,] <- rep(0.2, G)
sigma_square[3,] <- rep(0.15, G)


Z <- list()
#subtype indicator. Z[b,j] represents the subtype of sample j in batch b

Z[[1]] <- as.integer(c(rep(1,floor(pi[1,1]*n_vec[1])),rep(2,floor(pi[1,2]*n_vec[1])),
 rep(3,floor(pi[1,3]*n_vec[1]))))

Z[[2]] <- as.integer(c(rep(1,floor(pi[2,1]*n_vec[2])),rep(2,floor(pi[2,2]*n_vec[2])),
rep(3,floor(pi[2,3]*n_vec[2]))))

Z[[3]] <- as.integer(c(rep(1,floor(pi[3,1]*n_vec[3])),rep(2,floor(pi[3,2]*n_vec[3])),
rep(3,floor(pi[3,3]*n_vec[3]))))



for(b in 1:B){ #generate data
num <- n_vec[b]
example_Data[[b]] <-  sapply(1:num, function(j){
tmp <- alpha + mu[ ,Z[[b]][j]] + gamma[b, ] +
rnorm(G, sd = sqrt(sigma_square[b, ]))

tmp
})

}

#############################################################################
#Apply the BUSgibbs Function
#############################################################################
```

```
set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
```

`calculate_EPSR_gamma` *Calculate the estimated potential scale reduction factors (EPSR factors) for location batch effects*

Description

When Gibbs sampler attains stationary, the distances between multiple chains (with multiple initial values) should be small. The EPSR factors are calculated to help decide the iteration number of Gibbs sampler.

Usage

```
calculate_EPSR_gamma(gamma_PosterSamp_chain1, gamma_PosterSamp_chain2)
```

Arguments

`gamma_PosterSamp_chain1`
: posterior samples of location batch effects from chain 1.

`gamma_PosterSamp_chain2`
: posterior samples of location batch effects from chain 2.

Value

`EPSR_gamma`
: The EPSR factors for gamma, a G by B matrix. Note that EPSR_gamma[,1] is a NA vector. gamma[,1] are fixed at zero, so their EPSR factors are not taken into account.

Author(s)

Xiangyu Luo

Examples

```
#2 batches, 10 genes, 100 posterior samples per parameter
chain1 <- 1+array(rnorm(10*2*100,sd=0.05), dim=c(2,10,100))
chain2 <- 1+array(rnorm(10*2*100,sd=0.05), dim=c(2,10,100))
calculate_EPSR_gamma(chain1,chain2)
```

calculate_EPSR_mu *Calculate the estimated potential scale reduction factors (EPSR factors) for subtype effects*

Description

When Gibbs sampler attains stationary, the distances between multiple chains (with multiple initial values) should be small. The EPSR factors are calculated to help decide the iteration number of Gibbs sampler.

Usage

```
calculate_EPSR_mu(mu_PosterSamp_chain1, mu_PosterSamp_chain2)
```

Arguments

mu_PosterSamp_chain1	posterior samples of subtype effects from chain 1.
mu_PosterSamp_chain2	posterior samples of subtype effects from chain 1.

Value

EPSR_gamma	The EPSR factors for mu, a G by K matrix. Note that EPSR_mu[,1] is a NA vector. mu[,1] are fixed at zero, so their EPSR factors are not taken into account.

Author(s)

Xiangyu Luo

Examples

```
#10 genes, 2 subtypes, 100 posterior samples per parameter
chain1 <- 1+array(rnorm(10*2*100,sd=0.05), dim=c(10,2,100))
chain2 <- 1+array(rnorm(10*2*100,sd=0.05), dim=c(10,2,100))
calculate_EPSR_mu(chain1,chain2)
```

calculate_EPSR_sigma_sq *Calculate the estimated potential scale reduction factors (EPSR factors) for the variances of expression values*

Description

When Gibbs sampler attains stationary, the distances between multiple chains (with multiple initial values) should be small. The EPSR factors are calculated to help decide the iteration number of Gibbs sampler.

Usage

```
calculate_EPSR_sigma_sq(sigma_sq_PosterSamp_chain1, sigma_sq_PosterSamp_chain2)
```

Arguments

sigma_sq_PosterSamp_chain1	posterior samples of variances from chain 1.
sigma_sq_PosterSamp_chain2	posterior samples of variances from chain 2.

Value

EPSR_sigma_sq	The EPSR factors for sigma_sq, a G by B matrix.

Author(s)

Xiangyu Luo

Examples

```
#2 batches, 10 genes, 100 posterior samples per parameter
chain1 <- 1+array(rnorm(10*2*100,sd=0.05), dim=c(2,10,100))
chain2 <- 1+array(rnorm(10*2*100,sd=0.05), dim=c(2,10,100))
calculate_EPSR_sigma_sq(chain1,chain2)
```

---

estimate_IG_indicators

Estimate the intrinsic gene indicators

Description

Call the function to estimate the intrinsic gene indicators.

Usage

```
estimate_IG_indicators(BUSfits, postprob_DE_threshold = 0.5)
```

Arguments

BUSfits	The BUSfits object output by the function BUSgibbs.
postprob_DE_threshold	the threshold to call an intrinsic gene indicator to be one or not according to whether its posterior probability is higher than postprob_DE_threshold or not. The default is 0.5.

Value

est_L	the estimated intrinsic gene indicators, a matrix where the rows represent genes and the columns correspond to subtypes k=2,...,K

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect

set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
#select the posterior probability threshold to estimate the intrinsic gene indicators
thr0 <- postprob_DE_thr_fun(BUSfits, fdr_threshold=0.1)
est_L <- estimate_IG_indicators(BUSfits, postprob_DE_threshold=thr0)

#obtain the intrinsic gene indicators
intrinsic_gene_indices <- IG_index(est_L)
```

IG_index *Obtain the intrinsic gene indices*

Description

Call the function to obtain the indices of the intrinsic genes.

Usage

```
IG_index(EstIGindicators)
```

Arguments

EstIGindicators
: The estimated intrinsic gene indicators.

Value

intrinsic_gene_indices
: The intrinsic gene indices

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect

set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
#select the posterior probability threshold to estimate intrinsic gene indicators
thr0 <- postprob_DE_thr_fun(BUSfits, fdr_threshold=0.1)
est_L <- estimate_IG_indicators(BUSfits, postprob_DE_threshold=thr0)

#obtain the intrinsic gene indicators
intrinsic_gene_indices <- IG_index(est_L)
```

location_batch_effects *Obtain the location batch effects from the output by BUSgibbs*

Description

Call the function to obtain the location batch effects from the output by BUSgibbs.

Usage

```
location_batch_effects(BUSfits)
```

Arguments

BUSfits	The BUSfits object output by the function BUSgibbs.

Value

est_location_batch_effects
: The estimated location batch effects, a matrix whose rows represent genes and columns correspond to batches.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
```

```
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect

set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
est_location_batch_effects <- location_batch_effects(BUSfits)
```

---

postprob_DE *Calculate the posterior probability of being differentially expressed*

Description

Calculate the posterior probability of being differentially expressed for genes in subtypes k (k>=2) compared to subtype 1.

Usage

```
postprob_DE(BUSfits)
```

Arguments

BUSfits	The BUSfits object output by the function BUSgibbs.

Value

postprob_DE_matr
: the matrix of posterior probabilities of being differentially expressed

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
```

```
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect

set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
postprob_DE(BUSfits)
```

`postprob_DE_thr_fun` *Select the the posterior probability threshold to control the false discovery rate*

Description

To control the false discovery rate at the targeted level, call postprob_DE_thr_fun to obtain the threshold for the posterior probability of being differentially expressed.

Usage

```
postprob_DE_thr_fun(BUSfits, fdr_threshold = 0.1)
```

Arguments

`BUSfits`	The BUSfits object output by the function `BUSgibbs`.
`fdr_threshold`	the false discovery rate level we want to control.

Value

`thre0`	the posterior probability threshold that controls the false discovery rate.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect
set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
#select kappa to estimate intrinsic gene indicators
thr0 <- postprob_DE_thr_fun(BUSfits, fdr_threshold=0.1)
est_L <- estimate_IG_indicators(BUSfits, postprob_DE_threshold=thr0)

#obtain the intrinsic gene indicators
intrinsic_gene_indices <- IG_index(est_L)
```

`print.BUSfits` *Print the output by BUSgibbs*

Description

Call the function to print the output by BUSgibbs.

Usage

```
## S3 method for class 'BUSfits'
print(x, ...)
```

Arguments

`x`	The BUSfits object output by the function `BUSgibbs`.
`...`	not used.

Value

print the results from the output by BUSgibbs.

Author(s)

Xiangyu Luo

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect
set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
print(BUSfits)
```

---

scale_batch_effects *Obtain the scale batch effects from the output by* BUSgibbs

---

Description

Call the function to obtain the scale batch effects from the output by BUSgibbs.

Usage

```
scale_batch_effects(BUSfits)
```

Arguments

BUSfits	The BUSfits object output by the function BUSgibbs.

Value

est_scale_batch_effects
: The estimated scale batch effects, a matrix where rows are genes and columns are batches.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect
set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
est_scale_batch_effects <- scale_batch_effects(BUSfits)
```

Subtypes *Obtain the subtype indicators from the output by* BUSgibbs

---

Description

Call the function to obtain the subtype indicators from the output by BUSgibbs.

Usage

```
Subtypes(BUSfits)
```

Arguments

BUSfits	The BUSfits object output by the function BUSgibbs.

Value

`est_subtypes` The estimated subtypes, an R list with length equal to the batch number. The bth element is the estimated subtype indicator vector for samples in batch `b`.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect
set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
est_subtypes <- Subtypes(BUSfits)
```

---

subtype_effects *Obtain the subtype effects from the output by* BUSgibbs

---

Description

Call the function to obtain the subtype effects from the output by `BUSgibbs`.

Arguments

`BUSfits` The BUSfits object output by the function `BUSgibbs`.

Value

est_subtype_effects

The estimated subtype effects, a matrix where rows are genes and columns are subtypes.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect
set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
est_subtype_effects <- subtype_effects(BUSfits)
```

---

summary.BUSfits *Summary the output by* BUSgibbs

Description

Call the function to summarize the output object by BUSgibbs.

Usage

```
## S3 method for class 'BUSfits'
summary(object, ...)
```

Arguments

object	The BUSfits object output by the function BUSgibbs.
...	not used.

Value

summarize the results from the output by BUSgibbs.

Author(s)

Xiangyu Luo

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect
set.seed(123)
BUSfits <- BUSgibbs(example_Data, n.subtypes = 3, n.iterations = 100, showIteration = FALSE)
summary(BUSfits)
```

---

visualize_data *Visualize the gene expression data from multiple batches*

Description

Use "heatmap.2" in R package "gplots" to visualize the gene expression data across multiple batches.

Usage

```
visualize_data(Data, title_name="Heatmap", gene_ind_set, color_key_range=seq(-0.5,8.5,1))
```

Arguments

Data	The gene expression data, an R list with length equal to the batch number. Each of its element is a gene expression matrix, where rows are genes and columns represent samples.
title_name	The title name of the heatmap.
gene_ind_set	The indices of the set of genes the user wants to display in the heatmap.
color_key_range	The color range in the color key.

Details

The values displayed in the heatmap are the raw values in the argument Data without scaling.

Value

visualize the gene expression data matrix, where one row is a gene and one column represents a sample.

Author(s)

Xiangyu Luo

References

Xiangyu Luo, Yingying Wei. Batch Effects Correction with Unknown Subtypes. Journal of the American Statistical Association. Accepted.

Examples

```
rm(list = ls(all = TRUE))
set.seed(123)
#a toy example, there are 6 samples and 20 genes in each batch
example_Data <- list()

#batch 1
example_Data[[1]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6))

#batch 2
batch2_effect <- c(2,2,2,1,1)
example_Data[[2]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch2_effect

#batch 3
batch3_effect <- c(3,2,1,1,2)
example_Data[[3]] <- rbind(matrix(c(1,1,5,5,10,10,
3,3,7,7,12,12), ncol=6, byrow=TRUE), matrix(c(1,2),nrow=18, ncol=6)) + batch3_effect

visualize_data(example_Data, title_name="Heatmap", gene_ind_set = 1:20, color_key_range=seq(0,10,2))
```

Index