

Package ‘HEM’

June 5, 2023

Version 1.73.0

Date 2005-09-16

Title Heterogeneous error model for identification of differentially expressed genes under multiple conditions

Author HyungJun Cho <hcho@virginia.edu> and Jae K. Lee
<jaeklee@virginia.edu>

Maintainer HyungJun Cho <hcho@virginia.edu>

Depends R (>= 2.1.0)

Imports Biobase, grDevices, stats, utils

Description This package fits heterogeneous error models for analysis of microarray data

License GPL (>= 2)

URL <http://www.healthsystem.virginia.edu/internet/hes/biostat/bioinformatics/>

biocViews Microarray, DifferentialExpression

git_url <https://git.bioconductor.org/packages/HEM>

git_branch devel

git_last_commit 2525d8c

git_last_commit_date 2023-04-25

Date/Publication 2023-06-05

R topics documented:

am.tran	2
am.tran.half	2
base.ASE.Olig	3
base.error.Olig	3
base.error.Olig.quanOnly	3
base.PSE.Olig	3
boot.base.ASE.Olig	4
boot.base.error.Olig	4

boot.base.PSE.Olig	4
fixbound.predict.smooth.spline	4
hem	5
hem.eb.prior	7
hem.fdr	10
hem.null.no	11
hem.null.one	12
hem.null.two	12
hem.preproc	12
mubcp	13
nonpar.error.Olig	13
nonpar.no.error.Olig	14
nonpar.rep.error.Olig	14
par.error.Olig	14
par.no.error.Olig	14
par.rep.error.Olig	15
pbrain	15
permut	15
quant.norm	16
quant.normal	16
quant.normal2	16
quant.normalize	16
remove.sig.genes	17

Index 18

am.tran	<i>AM transformation for LPE</i>
---------	----------------------------------

Description

Computes AM for LPE

Author(s)

HyungJun Cho and Jae K. Lee

am.tran.half	<i>AM transformation for LPE</i>
--------------	----------------------------------

Description

Computes AM for LPE

Author(s)

HyungJun Cho and Jae K. Lee

base.ASE.Olig	<i>Baseline ASE estimation for oligonucleotide arrays</i>
---------------	---

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

base.error.Olig	<i>Baseline error estimation for oligonucleotide arrays</i>
-----------------	---

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

base.error.Olig.quanOnly	<i>Baseline error estimation for oligonucleotide arrays</i>
--------------------------	---

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

base.PSE.Olig	<i>Baseline PSE estimation for oligonucleotide arrays</i>
---------------	---

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

boot.base.ASE.Olig	<i>Baseline error bootstrap estimation for oligonucleotide arrays</i>
--------------------	---

Description

Estimates baseline error using bootstrap samples for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

boot.base.error.Olig	<i>Baseline error bootstrap estimation for oligonucleotide arrays</i>
----------------------	---

Description

Estimates baseline error using bootstrap samples for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

boot.base.PSE.Olig	<i>Baseline error bootstrap estimation for oligonucleotide arrays</i>
--------------------	---

Description

Estimates baseline error using bootstrap samples for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

fixbound.predict.smooth.spline	<i>Prediction using smoothing spine</i>
--------------------------------	---

Description

Makes predictions using smoothing spine

Author(s)

HyungJun Cho and Jae K. Lee

hem	<i>Heterogeneous Error Model for Identification of Differential Expressed Genes Under Multiple Conditions</i>
-----	---

Description

Fits an error model with heterogeneous experimental and biological variances.

Usage

```
hem(dat, probe.ID=NULL, n.layer, design, burn.ins=1000, n.samples=3000,
    method.var.e="gam", method.var.b="gam", method.var.t="gam",
    var.e=NULL, var.b=NULL, var.t=NULL, var.g=1, var.c=1, var.r=1,
    alpha.e=3, beta.e=.1, alpha.b=3, beta.b=.1, alpha.t=3, beta.t=.2,
    n.digits=10, print.message.on.screen=TRUE)
```

Arguments

<code>dat</code>	data
<code>probe.ID</code>	a vector of probe set IDs
<code>n.layer</code>	number of layers; 1=one-layer EM, 2=two-layer EM
<code>design</code>	design matrix
<code>burn.ins</code>	number of burn-ins for MCMC
<code>n.samples</code>	number of samples for MCMC
<code>method.var.e</code>	prior specification method for experimental variance; "gam"=Gamma(alpha,beta), "peb"=parametric EB prior specification, "neb"=nonparametric EB prior specification
<code>method.var.b</code>	prior specification method for biological variance; "gam"=Gamma(alpha,beta), "peb"=parametric EB prior specification
<code>method.var.t</code>	prior specification method for total variance; "gam"=Gamma(alpha,beta), "peb"=parametric EB prior specification, "neb"=nonparametric EB prior specification
<code>var.e</code>	prior estimate matrix for experimental variance
<code>var.b</code>	prior estimate matrix for biological variance
<code>var.t</code>	prior estimate matrix for total variance
<code>var.g</code>	$N(0, \text{var.g})$; prior parameter for gene effect
<code>var.c</code>	$N(0, \text{var.c})$; prior parameter for condition effect
<code>var.r</code>	$N(0, \text{var.r})$; prior parameter for interaction effect of gene and condition
<code>alpha.e, beta.e</code>	Gamma(alpha.e,alpha.e); prior parameters for inverse of experimental variance
<code>alpha.b, beta.b</code>	Gamma(alpha.b,alpha.b); prior parameters for inverse of biological variance

```

alpha.t, beta.t      Gamma(alpha.b,alpha.b); prior parameters for inverse of total variance
n.digits             number of digits
print.message.on.screen
                     if TRUE, process status is shown on screen.

```

Value

```

n.gene               numer of genes
n.chip               number of chips
n.cond               number of conditions
design                design matrix
burn.ins             number of burn-ins for MCMC
n.samples            number of samples for MCMC
priors               prior parameters
m.mu                 estimated mean expression intensity for each gene under each condition
m.x                  estimated unobserved expression intensity for each combination of genes, con-
                     ditions, and individuals (n.layer=2)
m.var.b              estimated biological variances (n.layer=2)
m.var.e              estimated experiemental variances (n.layer=2)
m.var.t              estimated total variances (n.layer=1)
H                     H-scores

```

Author(s)

HyungJun Cho and Jae K. Lee

References

Cho, H. and Lee, J.K. (2004) Bayesian Hierarchical Error Model for Analysis of Gene Expression Data, *Bioinformatics*, 20: 2016-2025.

See Also

[hem.eb.prior](#), [hem.fdr](#)

Examples

```

#Example 1: Two-layer HEM

data(pbrain)

##construct a design matrix
cond <- c(1,1,1,1,1,1,2,2,2,2,2,2) #condition
ind  <- c(1,1,2,2,3,3,1,1,2,2,3,3) #biological replicate
rep  <- c(1,2,1,2,1,2,1,2,1,2,1,2) #experimental replicate

```

```

design <- data.frame(cond,ind,rep)

##normalization
pbrain.nor <- hem.preproc(pbrain[,2:13])

##fit HEM with two layers of error
##using the small numbers of burn-ins and MCMC samples for a testing purpose;
##but increase the numbers for a practical purpose
#pbrain.hem <- hem(pbrain.nor, n.layer=2, design=design,
#                burn.ins=10, n.samples=30)

##print H-scores
#pbrain.hem$H

#Example 2: One-layer HEM

data(mubcp)

##construct a design matrix
cond <- c(rep(1,6),rep(2,5),rep(3,5),rep(4,5),rep(5,5))
ind  <- c(1:6,rep((1:5),4))
design <- data.frame(cond,ind)

##construct a design matrix
mubcp.nor <- hem.preproc(mubcp)

#fit HEM with one layers of error
#using the small numbers of burn-ins and MCMC samples for a testing purpose;
#but increase the numbers for a practical purpose
#mubcp.hem <- hem(mubcp.nor, n.layer=1,design=design, burn.ins=10, n.samples=30)

##print H-scores
#mubcp.hem$H

###NOTE: Use 'hem.fdr' for FDR evaluation
###NOTE: Use 'hem.eb.prior' for Empirical Bayes (EB) prior sepecification
###NOTE: Use EB-HEM ('hem' after 'hem.eb.prior') for small data sets

```

hem.eb.prior

Empirical Bayes (EB) Prior Specification

Description

Estimates experimental and biological variances by LPE and resampling

Usage

```
hem.eb.prior(dat, n.layer, design,
             method.var.e="neb", method.var.b="peb", method.var.t="neb",
             rep=TRUE, baseline.var="LPE", p.remove=0, max.chip=4,
             q=0.01, B=25, n.digits=10, print.message.on.screen=TRUE)
```

Arguments

<code>dat</code>	data
<code>n.layer</code>	number of layers
<code>design</code>	design matrix
<code>method.var.e</code>	prior specification method for experimental variance; "peb"=parametric EB prior specification, "neb"=nonparametric EB prior specification
<code>method.var.b</code>	prior specification method for biological variance; "peb"=parametric EB prior specification
<code>method.var.t</code>	prior specification method for total variance; "peb"=parametric EB prior specification, "neb"=nonparametric EB prior specification
<code>rep</code>	no replication if FALSE
<code>baseline.var</code>	baseline variance estimation method: LPE for replicated data and BLPE, PSE, or ASE for unreplicated data
<code>p.remove</code>	percent of removed rank-variance genes for BLPE
<code>max.chip</code>	maximum number of chips to estimate errors
<code>q</code>	quantile for partitioning genes based on expression levels
<code>B</code>	number of iterations for resampling
<code>n.digits</code>	number of digits
<code>print.message.on.screen</code>	if TRUE, process status is shown on screen.

Value

<code>var.b</code>	prior estimate matrix for biological variances (n.layer=2)
<code>var.e</code>	prior estimate matrix for experiemtnal variances (n.layer=2)
<code>var.t</code>	prior estimate matrix for total variances (n.layer=1)

Author(s)

HyungJun Cho and Jae K. Lee

See Also

[hem](#), [hem.fdr](#)

Examples

#Example 1: Two-layer HEM with EB prior specification

```
data(pbrain)
```

```
##construct a design matrix
cond <- c(1,1,1,1,1,1,2,2,2,2,2,2)
ind  <- c(1,1,2,2,3,3,1,1,2,2,3,3)
rep  <- c(1,2,1,2,1,2,1,2,1,2,1,2)
design <- data.frame(cond,ind,rep)
```

```
##normalization
pbrain.nor <- hem.preproc(pbrain[,2:13])
```

```
##take a subset for a testing purpose;
##use all genes for a practical purpose
pbrain.nor <- pbrain.nor[1:1000,]
```

```
##estimate hyperparameters of variances by LPE
#pbrain.eb <- hem.eb.prior(pbrain.nor, n.layer=2, design=design,
#                           method.var.e="neb", method.var.b="peb")
```

```
#fit HEM with two layers of error
#using the small numbers of burn-ins and MCMC samples for a testing purpose;
#but increase the numbers for a practical purpose
#pbrain.hem <- hem(pbrain.nor, n.layer=2, design=design, burn.ins=10, n.samples=30,
#                  method.var.e="neb", method.var.b="peb",
#                  var.e=pbrain.eb$var.e, var.b=pbrain.eb$var.b)
```

#Example 2: One-layer HEM with EB prior specification

```
data(mubcp)
```

```
##construct a design matrix
cond <- c(rep(1,6),rep(2,5),rep(3,5),rep(4,5),rep(5,5))
ind  <- c(1:6,rep((1:5),4))
design <- data.frame(cond,ind)
```

```
##normalization
mubcp.nor <- hem.preproc(mubcp)
```

```
##take a subset for a testing purpose;
##use all genes for a practical purpose
mubcp.nor <- mubcp.nor[1:1000,]
```

```
##estimate hyperparameters of variances by LPE
#mubcp.eb <- hem.eb.prior(mubcp.nor, n.layer=1, design=design,
#                          method.var.t="neb")
```

```
#fit HEM with two layers of error
#using the small numbers of burn-ins and MCMC samples for a testing purpose;
```

```
#but increase the numbers for a practical purpose
#mubcp.hem <- hem(mubcp.nor, n.layer=1, design=design, burn.ins=10, n.samples=30,
#               method.var.t="neb", var.t=mubcp.eb$var.t)
```

hem.fdr

FDR Evaluation

Description

Computes resampling-based False Discovery Rate (FDR)

Usage

```
hem.fdr(dat, n.layer, design, rep=TRUE, hem.out, eb.out=NULL, n.iter=5, q.trim=0.9,
        target.fdr=c(0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.20, 0.30, 0.40, 0.50),
        n.digits=10, print.message.on.screen=TRUE)
```

Arguments

dat	data
n.layer	number of layers: 1=one-layer EM; 2=two-layer EM
design	design matrix
rep	no replication if FALSE
hem.out	output from hem function
eb.out	output from hem.eb.prior function
n.iter	number of iterations
q.trim	quantile used for estimating the proportion of true negatives (π_0)
target.fdr	Target FDRs
n.digits	number of digits
print.message.on.screen	if TRUE, process status is shown on screen.

Value

fdr	H-values and corresponding FDRs
π_0	estimated proportion of true negatives
H.null	H-scores from null data
targets	given target FDRs, corresponding critical values and numbers of significant genes are provided

Author(s)

HyungJun Cho and Jae K. Lee

See Also

[hem.eb.prior hem](#)

Examples

```
data(pbrain)

##construct a design matrix
cond <- c(1,1,1,1,1,1,2,2,2,2,2)
ind  <- c(1,1,2,2,3,3,1,1,2,2,3,3)
rep  <- c(1,2,1,2,1,2,1,2,1,2,1,2)
design <- data.frame(cond,ind,rep)

##normalization
pbrain.nor <- hem.preproc(pbrain[,2:13])

##take a subset for a testing purpose;
##use all genes for a practical purpose
pbrain.nor <- pbrain.nor[1:1000,]

##estimate hyperparameters of variances by LPE
#pbrain.eb <- hem.eb.prior(pbrain.nor, n.layer=2, design=design,
#                           method.var.e="neb", method.var.b="peb")

##fit HEM with two layers of error
##using the small numbers of burn-ins and MCMC samples for a testing purpose;
##but increase the numbers for a practical purpose
#pbrain.hem <- hem(pbrain.nor, n.layer=2, design=design,burn.ins=10, n.samples=30,
#                  method.var.e="neb", method.var.b="peb",
#                  var.e=pbrain.eb$var.e, var.b=pbrain.eb$var.b)

##Estimate FDR based on resampling
#pbrain.fdr <- hem.fdr(pbrain.nor, n.layer=2, design=design,
#                      hem.out=pbrain.hem, eb.out=pbrain.eb)
```

hem.null.no

Generation of null data

Description

Generates null data by resampling

Author(s)

HyungJun Cho and Jae K. Lee

hem.null.one	<i>Generation of null data</i>
--------------	--------------------------------

Description

Generates null data by resampling

Author(s)

HyungJun Cho and Jae K. Lee

hem.null.two	<i>Generation of null data</i>
--------------	--------------------------------

Description

Generates null data by resampling

Author(s)

HyungJun Cho and Jae K. Lee

hem.preproc	<i>Preprocessing</i>
-------------	----------------------

Description

Performs IQR normalization, thesholding, and log2-transformation

Usage

```
hem.preproc(x, data.type = "MAS5")
```

Arguments

x	data
data.type	data type: MAS5 or MAS4

Author(s)

HyungJun Cho and Jae K. Lee

See Also

[hem](#), [hem.eb.prior](#), [hem.fdr](#)

Examples

```
data(pbrain)
pbrain.nor <- hem.preproc(pbrain[,2:13])
```

mubcp

Gene expression data for mouse B cell development

Description

This data set consists of gene expression of the five consecutive stages (pre-B1, large pre-B2, small pre-B2, immature B, and mature B cells) of mouse B cell development. The data were obtained with high-density oligonucleotide arrays, Affymetrix Mu11k GeneChips, from flow-cytometrically purified cells.

Usage

```
data(mubcp)
```

Format

A matrix containing 13,207 probe sets and 26 chips; first 6 chips for pre-B1 cell and next 20 chips for other stages (5 chips for each)

Source

Hoffmann, R., Seidl, T., Neeb, M., Rolink, A. and Melchers, F. (2002). Changes in gene expression profiles in developing B cells of murine bone marrow, *Genome Research* 12:98-111.

nonpar.error.0lig

Baseline error nonparametric estimation for oligonucleotide arrays

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

nonpar.no.error.Olig *Baseline error nonparametric estimation for oligonucleotide arrays*

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

nonpar.rep.error.Olig *Baseline error nonparametric estimation for oligonucleotide arrays*

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

par.error.Olig *Baseline error parametric estimation for oligonucleotide arrays*

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

par.no.error.Olig *Baseline error parametric estimation for oligonucleotide arrays*

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

par.rep.error.Olig	<i>Baseline error parametric estimation for oligonucleotide arrays</i>
--------------------	--

Description

Estimates baseline error for oligonucleotide arrays

Author(s)

HyungJun Cho and Jae K. Lee

pbrain	<i>Gene expression data for primate brains</i>
--------	--

Description

This data set consists of gene expression of primate brains (Affymetrix U95A GeneChip). The frozen brains of three humans (H1, H2, H3) and three chimpanzees (C1, C2, C3) were used to take the postmortem tissue samples, and two independent tissue samples for each individual were taken.

Usage

```
data(pbrain)
```

Format

A matrix containing 12,600 probe sets and 12 chips (H1,H1,H2,H2,H3,H3,C1,C1,C2,C2,C3,C3); the first column is probe set ID

Source

Enard, W., Khaitovich, P., Klose, J., Zollner, S., Heissig, F., Giavalisco, P., Nieselt-Struwe, K., Muchmore, E., Varki, A., Ravid, R., Doxiadis, G.M., Bontrop, R.R., and Paabo, S. (2002) Intra- and interspecific variation in primate gene expression patterns, *Science* 296:340-343

permut	<i>Permutation</i>
--------	--------------------

Description

Permute

Author(s)

HyungJun Cho and Jae K. Lee

quant.norm	Quantile normalization
Description Performs quantile normalization	
Author(s) HyungJun Cho and Jae K. Lee	
quant.normal	Normalization
Description Normalization	
Author(s) HyungJun Cho and Jae K. Lee	
quant.normal2	Normalization
Description Normalization	
Author(s) HyungJun Cho and Jae K. Lee	
quant.normalize	Quantile normalization
Description Performs quantile normalization	
Author(s) HyungJun Cho and Jae K. Lee	

remove.sig.genes	<i>Remove significant genes</i>
------------------	---------------------------------

Description

Remove significant genes

Author(s)

HyungJun Cho and Jae K. Lee

Index

* datasets

mubcp, 13
pbrain, 15

* models

am.tran, 2
am.tran.half, 2
base.ASE.Olig, 3
base.error.Olig, 3
base.error.Olig.quanOnly, 3
base.PSE.Olig, 3
boot.base.ASE.Olig, 4
boot.base.error.Olig, 4
boot.base.PSE.Olig, 4
fixbound.predict.smooth.spline, 4
hem, 5
hem.eb.prior, 7
hem.fdr, 10
hem.null.no, 11
hem.null.one, 12
hem.null.two, 12
hem.preproc, 12
nonpar.error.Olig, 13
nonpar.no.error.Olig, 14
nonpar.rep.error.Olig, 14
par.error.Olig, 14
par.no.error.Olig, 14
par.rep.error.Olig, 15
permut, 15
quant.norm, 16
quant.normal, 16
quant.normal2, 16
quant.normalize, 16
remove.sig.genes, 17

am.tran, 2
am.tran.half, 2

base.ASE.Olig, 3
base.error.Olig, 3
base.error.Olig.quanOnly, 3

base.PSE.Olig, 3
boot.base.ASE.Olig, 4
boot.base.error.Olig, 4
boot.base.PSE.Olig, 4

fixbound.predict.smooth.spline, 4

hem, 5, 8, 11, 13
hem.eb.prior, 6, 7, 11, 13
hem.fdr, 6, 8, 10, 13
hem.null.no, 11
hem.null.one, 12
hem.null.two, 12
hem.preproc, 12

mubcp, 13

nonpar.error.Olig, 13
nonpar.no.error.Olig, 14
nonpar.rep.error.Olig, 14

par.error.Olig, 14
par.no.error.Olig, 14
par.rep.error.Olig, 15
pbrain, 15
permut, 15

quant.norm, 16
quant.normal, 16
quant.normal2, 16
quant.normalize, 16

remove.sig.genes, 17