Package ‘GeneSelectMMD’

January 16, 2026

Type Package

Title Gene selection based on the marginal distributions of gene profiles that characterized by a mixture of three-component multivariate distributions

Version 2.55.0

Date 2020-04-03

Author Jarrett Morrow <remdj@channing.harvard.edu>, Weiliang Qiu <weiliang.qiu@gmail.com>, Wenqing He <whe@stats.uwo.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>, Ross Lazarus <ross.lazarus@channing.harvard.edu>.

Maintainer Weiliang Qiu <weiliang.qiu@gmail.com>

Depends R (>= 2.13.2), Biobase

Imports MASS, graphics, stats, limma

Suggests ALL

Description Gene selection based on a mixture of marginal distributions.

License GPL (>= 2)

biocViews DifferentialExpression

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

git_branch devel

git_last_commit 6c8a3d6

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-01-15

Contents

errRates *Calculating FDR, FNDR, FPR, and FNR for a real microarray data set*

Description

Calculating FDR, FNDR, FPR, and FNR for a real microarray data set based on the mixture of marginal distributions.

Usage

```
errRates(obj.gsMMD)
```

Arguments

`obj.gsMMD`	an object returned by `gsMMD`, `gsMMD.default`, `gsMMD2`, or `gsMMD2.default`

Details

We first fit the real microarray data set by the mixture of marginal distributions. Then we calculate the error rates based on the posterior distributions of a gene belonging to a gene cluster given its gene profiles. Please refer to Formula (7) on the page 6 of the paper listed in the Reference section.

Value

A vector of 4 elements:

`FDR`	the percentage of nondifferentially expressed genes among selected genes.
`FNDR`	the percentage of differentially expressed genes among unselected genes.
`FPR`	the percentage of selected genes among nondifferentially expressed genes
`FNR`	the percentage of un-selected genes among differentially expressed genes

Author(s)

Jarrett Morrow <remdj@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Wenqing He <whe@stats.uwo.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>, Ross Lazarus <ross.lazarus@channing.harvard.edu>

References

Qiu, W.-L., He, W., Wang, X.-G. and Lazarus, R. (2008). A Marginal Mixture Model for Selecting Differentially Expressed Genes across Two Types of Tissue Samples. *The International Journal of Biostatistics. 4(1):Article 20.* http://www.bepress.com/ijb/vol4/iss1/20

Examples

```
  ## Not run: 
    library(ALL)
    data(ALL)
    eSet1 <- ALL[1:100, ALL$BT == "B3" | ALL$BT == "T2"]

    mem.str <- as.character(eSet1$BT)
    nSubjects <- length(mem.str)
    memSubjects <- rep(0,nSubjects)
    # B3 coded as 0, T2 coded as 1
    memSubjects[mem.str == "T2"] <- 1

    obj.gsMMD <- gsMMD(eSet1, memSubjects, transformFlag = TRUE,
      transformMethod = "boxcox", scaleFlag = TRUE, quiet = FALSE)
    round(errRates(obj.gsMMD), 3)

## End(Not run)
```

gsMMD *Gene selection based on a mixture of marginal distributions*

Description

Gene selection based on the marginal distributions of gene profiles that characterized by a mixture of three-component multivariate distributions. Input is an object derived from the class `ExpressionSet`. The function will obtain initial gene cluster membership by its own.

Usage

```
gsMMD(obj.eSet,
      memSubjects,
      maxFlag = TRUE,
      thrshPostProb = 0.5,
      geneNames = NULL,
      alpha = 0.05,
      iniGeneMethod = "Ttest",
      transformFlag = FALSE,
      transformMethod = "boxcox",
      scaleFlag = TRUE,
      criterion = c("cor", "skewness", "kurtosis"),
      minL = -10,
      maxL = 10,
      stepL = 0.1,
      eps = 0.001,
      ITMAX = 100,
      plotFlag = FALSE,
      quiet=TRUE)
```

Arguments

`obj.eSet` an object derived from the class `ExpressionSet` which contains the matrix of gene expression levels. The rows of the matrix are genes. The columns of the matrix are subjects.

`memSubjects` a vector of membership of subjects. `memSubjects[i]=1` means the i-th subject belongs to diseased group, 0 otherwise.

`maxFlag` logical. Indicate how to assign gene class membership. `maxFlag`=TRUE means that a gene will be assigned to a class in which the posterior probability of the gene belongs to this class is maximum. `maxFlag`=FALSE means that a gene will be assigned to class 1 if the posterior probability of the gene belongs to class 1 is greater than `thrshPostProb`. Similarly, a gene will be assigned to class 1 if the posterior probability of the gene belongs to class 1 is greater than `thrshPostProb`. If the posterior probability is less than `thrshPostProb`, the gene will be assigned to class 2 (non-differentially expressed gene group).

`thrshPostProb` threshold for posterior probabilities. For example, if the posterior probability that a gene belongs to cluster 1 given its gene expression levels is larger than `thrshPostProb`, then this gene will be assigned to cluster 1.

`geneNames` an optional character vector of gene names

`alpha` significant level which is equal to `1-conf.level`, `conf.level` is the argument for the function `t.test`.

`iniGeneMethod` method to get initial 3-cluster partition of genes. Available methods are: “Ttest”, “Wilcox”.

`transformFlag` logical. Indicate if data transformation is needed

`transformMethod`
method for transforming data. Available methods include "boxcox", "log2", "log10", "log", "none".

`scaleFlag` logical. Indicate if gene profiles are to be scaled to have mean zero and variance one. If `transformFlag=TRUE` and `scaleFlag=TRUE`, then scaling is performed after transformation. To avoid linear dependence of tissue samples after scaling gene profiles, we delete one tissue sample after scaling (c.f. details).

`criterion` if `transformFlag=TRUE`, `criterion` indicates what criterion to determine if data looks like normal. “cor” means using Pearson’s correlation. The idea is that the observed quantiles after transformation should be close to theoretical normal quantiles. So we can use Pearson’s correlation to check if the scatter plot of theoretical normal quantiles versus observed quantiles is a straightline. “skewness” means using skewness measure to check if the distribution of the transformed data are close to normal distribution; “kurtosis” means using kurtosis measure to check normality.

`minL` lower limit for the `lambda` parameter used in Box-Cox transformation `maxL` upper limit for the `lambda` parameter used in Box-Cox transformation

`stepL` tolerance when searching the optimal `lambda` parameter used in Box-Cox transformation

`eps` a small positive value. If the absolute value of a value is smaller than `eps`, this value is regarded as zero.

`ITMAX` maximum iteration allowed for iterations in the EM algorithm

`plotFlag` logical. Indicate if the Box-Cox normality plot should be output.

`quiet` logical. Indicate if intermediate results should be printed out.

Details

We assume that the distribution of gene expression profiles is a mixture of 3-component multivariate normal distributions $\sum_{k=1}^{3} \pi_k f_k(x|\theta)$. Each component distribution f_k corresponds to a gene cluster. The 3 components correspond to 3 gene clusters: (1) up-regulated gene cluster, (2) non-differentially expressed gene cluster, and (3) down-regulated gene cluster. The model parameter vector is $\theta = (\pi_1, \pi_2, \pi_3, \mu_{c1}, \sigma^2_{c1}, \rho_{c1}, \mu_{n1}, \sigma^2_{n1}, \rho_{n1}, \mu_2, \sigma^2_2, \rho_2, \mu_{c3}, \sigma^2_{c3}, \rho_{c3}, \mu_{n3}, \sigma^2_{n3}, \rho_{n3}$. where π_1, π_2, and π_3 are the mixing proportions; μ_{c1}, σ^2_{c1}, and ρ_{c1} are the marginal mean, variance, and correlation of gene expression levels of cluster 1 (up-regulated genes) for diseased subjects; μ_{n1}, σ^2_{n1}, and ρ_{n1} are the marginal mean, variance, and correlation of gene expression levels of cluster 1 (up-regulated genes) for non-diseased subjects; μ_2, σ^2_2, and ρ_2 are the marginal mean, variance, and correlation of gene expression levels of cluster 2 (non-differentially expressed genes); μ_{c3}, σ^2_{c3}, and ρ_{c3} are the marginal mean, variance, and correlation of gene expression levels of cluster 3 (up-regulated genes) for diseased subjects; μ_{n3}, σ^2_{n3}, and ρ_{n3} are the marginal mean, variance, and correlation of gene expression levels of cluster 3 (up-regulated genes) for non-diseased subjects.

Note that genes in cluster 2 are non-differentially expressed across abnormal and normal tissue samples. Hence there are only 3 parameters for cluster 2.

To make sure the identifiability, we set the following contraints: $\mu_{c1} > \mu_{n1}$ and $\mu_{c3} < \mu_{n3}$.

To make sure the marginal covariance matrices are poisitive definite, we set the following contraints: $-1/(n_c - 1) < \rho_{c1} < 1$, $-1/(n_n - 1) < \rho_{n1} < 1$, $-1/(n - 1) < \rho_2 < 1$, $-1/(n_c - 1) < \rho_{c3} < 1$, $-1/(n_n - 1) < \rho_{n3} < 1$.

We also has the following constraints for the mixing proportion: $\pi_3 = 1 - \pi_1 - \pi_2$, $\pi_k > 0$, $k = 1, 2, 3$.

We apply the EM algorithm to estimate the model parameters. We regard the cluster membership of genes as missing values.

To facilitate the estimation of the parameters, we reparametrize the parameter vector as $\theta^* = (\pi_1, \pi_2, \mu_{c1}, \sigma^2_{c1}, r_{c1}, \delta_{n1}, \sigma^2_{n1}, r_{n1}, \mu_2, \sigma^2_2, r_2, \mu_{c3}, \sigma^2_{c3}, r_{c3}, \delta_{n3}, \sigma^2_{n3}, r_{n3})$, where $\mu_{n1} = \mu_{c1} - \exp(\delta_{n1})$, $\mu_{n3} = \mu_{c3} + \exp(\delta_{n3})$, $\rho_{c1} = (\exp(r_{c1}) - 1/(n_c - 1))/(1 + \exp(r_{c1}))$, $\rho_{n1} = (\exp(r_{n1}) - 1/(n_n - 1))/(1 + \exp(r_{n1}))$, $\rho_2 = (\exp(r_2) - 1/(n - 1))/(1 + \exp(r_2))$, $\rho_{c3} = (\exp(r_{c3}) - 1/(n_c - 1))/(1 + \exp(r_{c3}))$, $\rho_{n3} = (\exp(r_{n3}) - 1/(n_n - 1))/(1 + \exp(r_{n3}))$.

Given a gene, the expression levels of the gene are assumed independent. However, after scaling, the scaled expression levels of the gene are no longer independent and the rank $r^* = r - 1$ of the covariance matrix for the scaled gene profile will be one less than the rank r for the un-scaled gene profile Hence the covariance matrix of the gene profile will no longer be positive-definite. To avoid this problem, we delete a tissue sample after scaling since its information has been incorrporated by other scaled tissue samples. We arbitrarily select the tissue sample, which has the biggest label number, from the tissue sample group that has larger size than the other tissue sample group. For example, if there are 6 cancer tissue samples and 10 normal tissue samples, we delete the 10-th normal tissue sample after scaling.

Value

A list contains 18 elements.

`dat` the (transformed) microarray data matrix. If tranformation performed, then `dat` will be different from the input microarray data matrix.

`memSubjects` the same as the input `memSubjects`.

`memGenes` a vector of cluster membership of genes. 1 means up-regulated gene; 2 means non-differentially expressed gene; 3 means down-regulated gene.

`memGenes2` an variant of the vector of cluster membership of genes. 1 means differentially expressed gene; 0 means non-differentially expressed gene.

`para` parameter estimates (c.f. details).

`llkh` value of the loglikelihood function.

`wiMat` posterior probability that a gene belongs to a cluster given the expression levels of this gene. Column i is for cluster i.

`wiArray` posterior probability matrix for different initial gene selection methods.

`memIniMat` a matrix of initial cluster membership of genes.

`paraIniMat` a matrix of parameter estimates based on initial gene cluster membership.

`llkhIniVec` a vector of values of loglikelihood function.

`memMat` a matrix of cluster membership of genes based on the mixture of marginal models with initial parameter estimates obtained initial gene cluster membership.

`paraMat` a matrix of parameter estimates based on the mixture of marginal models with initial parameter estimates obtained initial gene cluster membership.

`llkhVec` a vector of values of loglikelihood function based on the mixture of marginal models with initial parameter estimates obtained initial gene cluster membership.

`lambda` the parameter used to do Box-Cox transformation

`paraRP` parameter estimates for reparametrized parameter vector (c.f. details).

`paraIniMatRP` a matrix of parameter estimates for reparametrized parameter vector based on initial gene cluster membership.

`paraMatRP` a matrix of parameter estimates for reparametrized parameter vector based on the mixture of marginal models with initial parameter estimates obtained initial gene cluster membership.

Note

The speed of the program can be slow for large data sets, however it has been improved using Fortran code.

Author(s)

Jarrett Morrow <remdj@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Wenqing He <whe@stats.uwo.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>, Ross Lazarus <ross.lazarus@channing.harvard.edu>

References

Qiu, W.-L., He, W., Wang, X.-G. and Lazarus, R. (2008). A Marginal Mixture Model for Selecting Differentially Expressed Genes across Two Types of Tissue Samples. *The International Journal of Biostatistics. 4(1):Article 20.* http://www.bepress.com/ijb/vol4/iss1/20

See Also

gsMMD.default, gsMMD2, gsMMD2.default

Examples

```
library(ALL)
data(ALL)
eSet1 <- ALL[1:100, ALL$BT == "B3" | ALL$BT == "T2"]

mem.str <- as.character(eSet1$BT)
nSubjects <- length(mem.str)
memSubjects <- rep(0,nSubjects)
# B3 coded as 0, T2 coded as 1
memSubjects[mem.str == "T2"] <- 1

obj.gsMMD <- gsMMD(eSet1, memSubjects, transformFlag = TRUE,
  transformMethod = "boxcox", scaleFlag = TRUE, quiet = FALSE)
round(obj.gsMMD$para, 3)
```

gsMMD.default *Gene selection based on a mixture of marginal distributions*

Description

Gene selection based on the marginal distributions of gene profiles that characterized by a mixture of three-component multivariate distributions. Input is a data matrix. The function will obtain initial gene cluster membership by its own.

Usage

```
gsMMD.default(X,
              memSubjects,
              maxFlag = TRUE,
              thrshPostProb = 0.5,
              geneNames = NULL,
              alpha = 0.05,
              iniGeneMethod = "Ttest",
              transformFlag = FALSE,
              transformMethod = "boxcox",
              scaleFlag = TRUE,
              criterion = c("cor", "skewness", "kurtosis"),
              minL = -10,
              maxL = 10,
              stepL = 0.1,
              eps = 0.001,
              ITMAX = 100,
              plotFlag = FALSE,
              quiet=TRUE)
```

Arguments

`X` a data matrix. The rows of the matrix are genes. The columns of the matrix are subjects.

`memSubjects` a vector of membership of subjects. `memSubjects[i]=1` means the i-th subject belongs to diseased group, 0 otherwise.

`maxFlag` logical. Indicate how to assign gene class membership. `maxFlag`=TRUE means that a gene will be assigned to a class in which the posterior probability of the gene belongs to this class is maximum. `maxFlag`=FALSE means that a gene will be assigned to class 1 if the posterior probability of the gene belongs to class 1 is greater than `thrshPostProb`. Similarly, a gene will be assigned to class 1 if the posterior probability of the gene belongs to class 1 is greater than `thrshPostProb`. If the posterior probability is less than `thrshPostProb`, the gene will be assigned to class 2 (non-differentially expressed gene group).

`thrshPostProb` threshold for posterior probabilities. For example, if the posterior probability that a gene belongs to cluster 1 given its gene expression levels is larger than `thrshPostProb`, then this gene will be assigned to cluster 1.

`geneNames` an optional character vector of gene names

`alpha` significant level which is equal to `1-conf.level`, `conf.level` is the argument for the function `t.test`.

`iniGeneMethod` method to get initial 3-cluster partition of genes. Available methods are: “Ttest”, “Wilcox”.

`transformFlag` logical. Indicate if data transformation is needed

`transformMethod`
method for transforming data. Available methods include "boxcox", "log2", "log10", "log", "none".

`scaleFlag` logical. Indicate if gene profiles are to be scaled. If `transformFlag=TRUE` and `scaleFlag=TRUE`, then scaling is performed after transformation. To avoid linear dependence of tissue samples after scaling gene profiles, we delete one tissue sample after scaling (c.f. details).

`criterion` if `transformFlag=TRUE`, `criterion` indicates what criterion to determine if data looks like normal. “cor” means using Pearson’s correlation. The idea is that the observed quantiles after transformation should be close to theoretical normal quantiles. So we can use Pearson's correlation to check if the scatter plot of theoretical normal quantiles versus observed quantiles is a straightline. "skewness" means using skewness measure to check if the distribution of the transformed data are close to normal distribution; "kurtosis" means using kurtosis measure to check normality.

`minL` lower limit for the `lambda` parameter used in Box-Cox transformation

`maxL` upper limit for the `lambda` parameter used in Box-Cox transformation

`stepL` tolerance when searching the optimal `lambda` parameter used in Box-Cox transformation

`eps` a small positive value. If the absolute value of a value is smaller than `eps`, this value is regarded as zero.

`ITMAX` maximum iteration allowed for iterations in the EM algorithm

`plotFlag` logical. Indicate if the Box-Cox normality plot should be output.

`quiet` logical. Indicate if intermediate results should be printed out.

Details

We assume that the distribution of gene expression profiles is a mixture of 3-component multivariate normal distributions $\sum_{k=1}^{3} \pi_k f_k(x|\theta)$. Each component distribution f_k corresponds to a gene cluster. The 3 components correspond to 3 gene clusters: (1) up-regulated gene cluster, (2) non-differentially expressed gene cluster, and (3) down-regulated gene cluster. The model parameter vector is $\theta = (\pi_1, \pi_2, \pi_3, \mu_{c1}, \sigma_{c1}^2, \rho_{c1}, \mu_{n1}, \sigma_{n1}^2, \rho_{n1}, \mu_2, \sigma_2^2, \rho_2, \mu_{c3}, \sigma_{c3}^2, \rho_{c3}, \mu_{n3}, \sigma_{n3}^2, \rho_{n3}$. where π_1, π_2, and π_3 are the mixing proportions; μ_{c1}, σ_{c1}^2, and ρ_{c1} are the marginal mean, variance, and correlation of gene expression levels of cluster 1 (up-regulated genes) for diseased subjects; μ_{n1}, σ_{n1}^2, and ρ_{n1} are the marginal mean, variance, and correlation of gene expression levels of cluster 1 (up-regulated genes) for non-diseased subjects; μ_2, σ_2^2, and ρ_2 are the marginal mean, variance, and correlation of gene expression levels of cluster 2 (non-differentially expressed genes); μ_{c3}, σ_{c3}^2, and ρ_{c3} are the marginal mean, variance, and correlation of gene expression levels of cluster 3 (up-regulated genes) for diseased subjects; μ_{n3}, σ_{n3}^2, and ρ_{n3} are the marginal mean, variance, and correlation of gene expression levels of cluster 3 (up-regulated genes) for non-diseased subjects.

Note that genes in cluster 2 are non-differentially expressed across abnormal and normal tissue samples. Hence there are only 3 parameters for cluster 2.

To make sure the identifiability, we set the following contraints: $\mu_{c1} > \mu_{n1}$ and $\mu_{c3} < \mu_{n3}$.

To make sure the marginal covariance matrices are poisitive definite, we set the following contraints: $-1/(n_c - 1) < \rho_{c1} < 1$, $-1/(n_n - 1) < \rho_{n1} < 1$, $-1/(n-1) < \rho_2 < 1$, $-1/(n_c - 1) < \rho_{c3} < 1$, $-1/(n_n - 1) < \rho_{n3} < 1$.

We also has the following constraints for the mixing proportion: $\pi_3 = 1 - \pi_1 - \pi_2$, $\pi_k > 0$, $k = 1, 2, 3$.

We apply the EM algorithm to estimate the model parameters. We regard the cluster membership of genes as missing values.

To facilitate the estimation of the parameters, we reparametrize the parameter vector as $\theta^* = (\pi_1, \pi_2, \mu_{c1}, \sigma_{c1}^2, r_{c1}, \delta_{n1}, \sigma_{n1}^2, r_{n1}, \mu_2, \sigma_2^2, r_2, \mu_{c3}, \sigma_{c3}^2, r_{c3}, \delta_{n3}, \sigma_{n3}^2, r_{n3})$, where $\mu_{n1} = \mu_{c1} - \exp(\delta_{n1})$, $\mu_{n3} = \mu_{c3} + \exp(\delta_{n3})$, $\rho_{c1} = (\exp(r_{c1}) - 1/(n_c - 1))/(1 + \exp(r_{c1}))$, $\rho_{n1} = (\exp(r_{n1}) -$

$1/(n_n - 1))/(1 + \exp(r_{n1}))$, $\rho_2 = (\exp(r_2) - 1/(n-1))/(1 + \exp(r_2))$, $\rho_{c3} = (\exp(r_{c3}) - 1/(n_c - 1))/(1 + \exp(r_{c3}))$, $\rho_{n3} = (\exp(r_{n3}) - 1/(n_n - 1))/(1 + \exp(r_{n3}))$.

Given a gene, the expression levels of the gene are assumed independent. However, after scaling, the scaled expression levels of the gene are no longer independent and the rank $r^* = r - 1$ of the covariance matrix for the scaled gene profile will be one less than the rank r for the un-scaled gene profile Hence the covariance matrix of the gene profile will no longer be positive-definite. To avoid this problem, we delete a tissue sample after scaling since its information has been incorrporated by other scaled tissue samples. We arbitrarily select the tissue sample, which has the biggest label number, from the tissue sample group that has larger size than the other tissue sample group. For example, if there are 6 cancer tissue samples and 10 normal tissue samples, we delete the 10-th normal tissue sample after scaling.

Value

A list contains 18 elements.

`dat` the (transformed) microarray data matrix. If tranformation performed, then `dat` will be different from the input microarray data matrix.

`memSubjects` the same as the input `memSubjects`.

`memGenes` a vector of cluster membership of genes. 1 means up-regulated gene; 2 means non-differentially expressed gene; 3 means down-regulated gene.

`memGenes2` an variant of the vector of cluster membership of genes. 1 means differentially expressed gene; 0 means non-differentially expressed gene.

`para` parameter estimates (c.f. details).

`llkh` value of the loglikelihood function.

`wiMat` posterior probability that a gene belongs to a cluster given the expression levels of this gene. Column i is for cluster i.

`wiArray` posterior probability matrix for different initial gene selection methods.

`memIniMat` a matrix of initial cluster membership of genes.

`paraIniMat` a matrix of parameter estimates based on initial gene cluster membership.

`llkhIniVec` a vector of values of loglikelihood function.

`memMat` a matrix of cluster membership of genes based on the mixture of marginal models with initial parameter estimates obtained initial gene cluster membership.

`paraMat` a matrix of parameter estimates based on the mixture of marginal models with initial parameter estimates obtained initial gene cluster membership.

`llkhVec` a vector of values of loglikelihood function based on the mixture of marginal models with initial parameter estimates obtained initial gene cluster membership.

`lambda` the parameter used to do Box-Cox transformation

`paraRP` parameter estimates for reparametrized parameter vector (c.f. details).

`paraIniMatRP` a matrix of parameter estimates for reparametrized parameter vector based on initial gene cluster membership.

`paraMatRP` a matrix of parameter estimates for reparametrized parameter vector based on the mixture of marginal models with initial parameter estimates obtained initial gene cluster membership.

Note

The speed of the program can be slow for large data sets, however it has been improved using Fortran code.

Author(s)

Jarrett Morrow <remdj@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Wenqing He <whe@stats.uwo.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>, Ross Lazarus <ross.lazarus@channing.harvard.edu>

References

Qiu, W.-L., He, W., Wang, X.-G. and Lazarus, R. (2008). A Marginal Mixture Model for Selecting Differentially Expressed Genes across Two Types of Tissue Samples. *The International Journal of Biostatistics. 4(1):Article 20.* http://www.bepress.com/ijb/vol4/iss1/20

See Also

gsMMD, gsMMD2, gsMMD2.default

Examples

```
  ## Not run: 
    library(ALL)
    data(ALL)
    eSet1 <- ALL[1:100, ALL$BT == "B3" | ALL$BT == "T2"]
    mat <- exprs(eSet1)

    mem.str <- as.character(eSet1$BT)
    nSubjects <- length(mem.str)
    memSubjects <- rep(0,nSubjects)
    # B3 coded as 0, T2 coded as 1
    memSubjects[mem.str == "T2"] <- 1

    obj.gsMMD <- gsMMD.default(mat, memSubjects, iniGeneMethod = "Ttest",
            transformFlag = TRUE, transformMethod = "boxcox", scaleFlag = TRUE)
    round(obj.gsMMD$para, 3)

## End(Not run)
```

gsMMD2 *Gene selection based on a mixture of marginal distributions*

Description

Gene selection based on the marginal distributions of gene profiles that characterized by a mixture of three-component multivariate distributions. Input is an object derived from the class ExpressionSet. The user needs to provide initial gene cluster membership.

Usage

```
gsMMD2(obj.eSet,
       memSubjects,
       memIni,
       maxFlag = TRUE,
       thrshPostProb = 0.5,
       geneNames = NULL,
       alpha = 0.05,
       transformFlag = FALSE,
       transformMethod = "boxcox",
       scaleFlag = TRUE,
       criterion = c("cor", "skewness", "kurtosis"),
       minL = -10,
       maxL = 10,
       stepL = 0.1,
       eps = 0.001,
       ITMAX = 100,
       plotFlag = FALSE,
       quiet=TRUE)
```

Arguments

obj.eSet	an object derived from the class ExpressionSet which contains the matrix of gene expression levels. The rows of the matrix are genes. The columns of the matrix are subjects.
memSubjects	a vector of membership of subjects. memSubjects[i]=1 means that the i-th subject belongs to diseased group, 0 otherwise.
memIni	a vector of user-provided gene cluster membership.
maxFlag	logical. Indicate how to assign gene class membership. maxFlag=TRUE means that a gene will be assigned to a class in which the posterior probability of the gene belongs to this class is maximum. maxFlag=FALSE means that a gene will be assigned to class 1 if the posterior probability of the gene belongs to class 1 is greater than thrshPostProb. Similarly, a gene will be assigned to class 1 if the posterior probability of the gene belongs to class 1 is greater than thrshPostProb. If the posterior probability is less than thrshPostProb, the gene will be assigned to class 2 (non-differentially expressed gene group).
thrshPostProb	threshold for posterior probabilities. For example, if the posterior probability that a gene belongs to cluster 1 given its gene expression levels is larger than thrshPostProb, then this gene will be assigned to cluster 1.
geneNames	an optional character vector of gene names
alpha	significant level which is equal to 1-conf.level, conf.level is the argument for the function t.test.
transformFlag	logical. Indicate if data transformation is needed
transformMethod	method for transforming data. Available methods include "boxcox", "log2", "log10", "log", "none".

`scaleFlag` logical. Indicate if gene profiles are to be scaled. If `transformFlag=TRUE` and `scaleFlag=TRUE`, then scaling is performed after transformation. To avoid linear dependence of tissue samples after scaling gene profiles, we delete one tissue sample after scaling (c.f. details).

`criterion` if `transformFlag=TRUE`, `criterion` indicates what criterion to determine if data looks like normal. "cor" means using Pearson's correlation. The idea is that the observed quantiles after transformation should be close to theoretical normal quantiles. So we can use Pearson's correlation to check if the scatter plot of theoretical normal quantiles versus observed quantiles is a straightline. "skewness" means using skewness measure to check if the distribution of the transformed data are close to normal distribution; "kurtosis" means using kurtosis measure to check normality.

`minL` lower limit for the `lambda` parameter used in Box-Cox transformation

`maxL` upper limit for the `lambda` parameter used in Box-Cox transformation

`stepL` tolerance when searching the optimal `lambda` parameter used in Box-Cox transformation

`eps` a small positive value. If the absolute value of a value is smaller than `eps`, this value is regarded as zero.

`ITMAX` maximum iteration allowed for iterations in the EM algorithm

`plotFlag` logical. Indicate if the Box-Cox normality plot should be output.

`quiet` logical. Indicate if intermediate results should be printed out.

Details

We assume that the distribution of gene expression profiles is a mixture of 3-component multivariate normal distributions $\sum_{k=1}^{3} \pi_k f_k(x|\theta)$. Each component distribution f_k corresponds to a gene cluster. The 3 components correspond to 3 gene clusters: (1) up-regulated gene cluster, (2) non-differentially expressed gene cluster, and (3) down-regulated gene cluster. The model parameter vector is $\theta = (\pi_1, \pi_2, \pi_3, \mu_{c1}, \sigma^2_{c1}, \rho_{c1}, \mu_{n1}, \sigma^2_{n1}, \rho_{n1}, \mu_2, \sigma^2_2, \rho_2, \mu_{c3}, \sigma^2_{c3}, \rho_{c3}, \mu_{n3}, \sigma^2_{n3}, \rho_{n3}$. where π_1, π_2, and π_3 are the mixing proportions; μ_{c1}, σ^2_{c1}, and ρ_{c1} are the marginal mean, variance, and correlation of gene expression levels of cluster 1 (up-regulated genes) for diseased subjects; μ_{n1}, σ^2_{n1}, and ρ_{n1} are the marginal mean, variance, and correlation of gene expression levels of cluster 1 (up-regulated genes) for non-diseased subjects; μ_2, σ^2_2, and ρ_2 are the marginal mean, variance, and correlation of gene expression levels of cluster 2 (non-differentially expressed genes); μ_{c3}, σ^2_{c3}, and ρ_{c3} are the marginal mean, variance, and correlation of gene expression levels of cluster 3 (up-regulated genes) for diseased subjects; μ_{n3}, σ^2_{n3}, and ρ_{n3} are the marginal mean, variance, and correlation of gene expression levels of cluster 3 (up-regulated genes) for non-diseased subjects.

Note that genes in cluster 2 are non-differentially expressed across abnormal and normal tissue samples. Hence there are only 3 parameters for cluster 2.

To make sure the identifiability, we set the following contraints: $\mu_{c1} > \mu_{n1}$ and $\mu_{c3} < \mu_{n3}$.

To make sure the marginal covariance matrices are poisitive definite, we set the following contraints: $-1/(n_c-1) < \rho_{c1} < 1$, $-1/(n_n-1) < \rho_{n1} < 1$, $-1/(n-1) < \rho_2 < 1$, $-1/(n_c-1) < \rho_{c3} < 1$, $-1/(n_n-1) < \rho_{n3} < 1$.

We also has the following constraints for the mixing proportion: $\pi_3 = 1 - \pi_1 - \pi_2$, $\pi_k > 0$, $k = 1, 2, 3$.

We apply the EM algorithm to estimate the model parameters. We regard the cluster membership of genes as missing values.

To facilitate the estimation of the parameters, we reparametrize the parameter vector as $\theta^* = (\pi_1$, π_2, μ_{c1}, σ^2_{c1}, r_{c1}, δ_{n1}, σ^2_{n1}, r_{n1}, μ_2, σ^2_2, r_2, μ_{c3}, σ^2_{c3}, r_{c3}, δ_{n3}, σ^2_{n3}, $r_{n3})$, where $\mu_{n1} = \mu_{c1} - \exp(\delta_{n1})$, $\mu_{n3} = \mu_{c3} + \exp(\delta_{n3})$, $\rho_{c1} = (\exp(r_{c1}) - 1/(n_c - 1))/(1 + \exp(r_{c1}))$, $\rho_{n1} = (\exp(r_{n1}) - 1/(n_n - 1))/(1 + \exp(r_{n1}))$, $\rho_2 = (\exp(r_2) - 1/(n - 1))/(1 + \exp(r_2))$, $\rho_{c3} = (\exp(r_{c3}) - 1/(n_c - 1))/(1 + \exp(r_{c3}))$, $\rho_{n3} = (\exp(r_{n3}) - 1/(n_n - 1))/(1 + \exp(r_{n3}))$.

Given a gene, the expression levels of the gene are assumed independent. However, after scaling, the scaled expression levels of the gene are no longer independent and the rank $r^* = r - 1$ of the covariance matrix for the scaled gene profile will be one less than the rank r for the un-scaled gene profile Hence the covariance matrix of the gene profile will no longer be positive-definite. To avoid this problem, we delete a tissue sample after scaling since its information has been incorrporated by other scaled tissue samples. We arbitrarily select the tissue sample, which has the biggest label number, from the tissue sample group that has larger size than the other tissue sample group. For example, if there are 6 cancer tissue samples and 10 normal tissue samples, we delete the 10-th normal tissue sample after scaling.

Value

A list contains 13 elements.

`dat` the (transformed) microarray data matrix. If tranformation performed, then `dat` will be different from the input microarray data matrix.

`memSubjects` the same as the input `memSubjects`.

`memGenes` a vector of cluster membership of genes. 1 means up-regulated gene; 2 means non-differentially expressed gene; 3 means down-regulated gene.

`memGenes2` an variant of the vector of cluster membership of genes. 1 means differentially expressed gene; 0 means non-differentially expressed gene.

`para` parameter estimates (c.f. details).

`llkh` value of the loglikelihood function.

`wiMat` posterior probability that a gene belongs to a cluster given the expression levels of this gene. Column i is for cluster i.

`memIni` the initial cluster membership of genes.

`paraIni` the parameter estimates based on initial gene cluster membership.

`llkhIni` the value of loglikelihood function.

`lambda` the parameter used to do Box-Cox transformation

`paraRP` parameter estimates for reparametrized parameter vector (c.f. details).

`paraIniRP` the parameter estimates for reparametrized parameter vector based on initial gene cluster membership.

Note

The speed of the program can be slow for large data sets, however it has been improved using Fortran code.

Author(s)

Jarrett Morrow <remdj@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Wenqing He <whe@stats.uwo.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>, Ross Lazarus <ross.lazarus@channing.harvard.edu>

References

Qiu, W.-L., He, W., Wang, X.-G. and Lazarus, R. (2008). A Marginal Mixture Model for Selecting Differentially Expressed Genes across Two Types of Tissue Samples. *The International Journal of Biostatistics. 4(1):Article 20.* http://www.bepress.com/ijb/vol4/iss1/20

See Also

gsMMD, gsMMD.default, gsMMD2.default

Examples

```
## Not run:
  library(ALL)
  data(ALL)
  eSet1 <- ALL[1:100, ALL$BT == "B3" | ALL$BT == "T2"]

  mem.str <- as.character(eSet1$BT)
  nSubjects <- length(mem.str)
  memSubjects <- rep(0,nSubjects)
  # B3 coded as 0, T2 coded as 1
  memSubjects[mem.str == "T2"] <- 1

  myWilcox <-
  function(x, memSubjects, alpha = 0.05)
  {
    xc <- x[memSubjects == 1]
    xn <- x[memSubjects == 0]

    m <- sum(memSubjects == 1)
    res <- wilcox.test(x = xc, y = xn, conf.level = 1 - alpha)
    res2 <- c(res$p.value, res$statistic - m * (m + 1) / 2)
    names(res2) <- c("p.value", "statistic")

    return(res2)
  }

  mat <- exprs(eSet1)
  tmp <- t(apply(mat, 1, myWilcox, memSubjects = memSubjects))
  colnames(tmp) <- c("p.value", "statistic")
  memIni <- rep(2, nrow(mat))
  memIni[tmp[, 1] < 0.05 & tmp[, 2] > 0] <- 1
  memIni[tmp[, 1] < 0.05 & tmp[, 2] < 0] <- 3

  cat("initial gene cluster size>>\n"); print(table(memIni)); cat("\n");
```

```
    obj.gsMMD <- gsMMD2(eSet1, memSubjects, memIni, transformFlag = TRUE,
         transformMethod = "boxcox", scaleFlag = TRUE, quiet = FALSE)
    round(obj.gsMMD$para, 3)

## End(Not run)
```

gsMMD2.default *Gene selection based on a mixture of marginal distributions*

Description

Gene selection based on the marginal distributions of gene profiles that characterized by a mixture of three-component multivariate distributions. Input is a data matrix. The user needs to provide initial gene cluster membership.

Usage

```
gsMMD2.default(X,
               memSubjects,
               memIni,
               maxFlag = TRUE,
               thrshPostProb = 0.5,
               geneNames = NULL,
               alpha = 0.05,
               transformFlag = FALSE,
               transformMethod = "boxcox",
               scaleFlag = TRUE,
               criterion = c("cor", "skewness", "kurtosis"),
               minL = -10,
               maxL = 10,
               stepL = 0.1,
               eps = 0.001,
               ITMAX = 100,
               plotFlag = FALSE,
               quiet=TRUE)
```

Arguments

`X` a data matrix. The rows of the matrix are genes. The columns of the matrix are subjects.

`memSubjects` a vector of membership of subjects. `memSubjects[i]=1` means the i-th subject belongs to diseased group, 0 otherwise.

`memIni` a vector of user-provided gene cluster membership.

`maxFlag` logical. Indicate how to assign gene class membership. `maxFlag`=TRUE means that a gene will be assigned to a class in which the posterior probability of the gene belongs to this class is maximum. `maxFlag`=FALSE means that a gene will be assigned to class 1 if the posterior probability of the gene belongs to class 1 is greater than `thrshPostProb`. Similarly, a gene will be assigned to class 1 if the posterior probability of the gene belongs to class 1 is greater than `thrshPostProb`. If the posterior probability is less than `thrshPostProb`, the gene will be assigned to class 2 (non-differentially expressed gene group).

`thrshPostProb` threshold for posterior probabilities. For example, if the posterior probability that a gene belongs to cluster 1 given its gene expression levels is larger than `thrshPostProb`, then this gene will be assigned to cluster 1.

`geneNames` an optional character vector of gene names

`alpha` significant level which is equal to `1-conf.level`, `conf.level` is the argument for the function `t.test`.

`transformFlag` logical. Indicate if data transformation is needed

`transformMethod`
method for transforming data. Available methods include "boxcox", "log2", "log10", "log", "none".

`scaleFlag` logical. Indicate if gene profiles are to be scaled. If `transformFlag=TRUE` and `scaleFlag=TRUE`, then scaling is performed after transformation. To avoid linear dependence of tissue samples after scaling gene profiles, we delete one tissue sample after scaling (c.f. details).

`criterion` if `transformFlag=TRUE`, `criterion` indicates what criterion to determine if data looks like normal. “cor” means using Pearson’s correlation. The idea is that the observed quantiles after transformation should be close to theoretical normal quantiles. So we can use Pearson’s correlation to check if the scatter plot of theoretical normal quantiles versus observed quantiles is a straightline. “skewness” means using skewness measure to check if the distribution of the transformed data are close to normal distribution; “kurtosis” means using kurtosis measure to check normality.

`minL` lower limit for the `lambda` parameter used in Box-Cox transformation

`maxL` upper limit for the `lambda` parameter used in Box-Cox transformation

`stepL` tolerance when searching the optimal `lambda` parameter used in Box-Cox transformation

`eps` a small positive value. If the absolute value of a value is smaller than `eps`, this value is regarded as zero.

`ITMAX` maximum iteration allowed for iterations in the EM algorithm

`plotFlag` logical. Indicate if the Box-Cox normality plot should be output.

`quiet` logical. Indicate if intermediate results should be printed out.

Details

We assume that the distribution of gene expression profiles is a mixture of 3-component multivariate normal distributions $\sum_{k=1}^{3}\pi_k f_k(x|\theta)$. Each component distribution f_k corresponds to a gene cluster. The 3 components correspond to 3 gene clusters: (1) up-regulated gene cluster, (2) non-differentially expressed gene cluster, and (3) down-regulated gene cluster. The model parameter vector is $\theta = (\pi_1, \pi_2, \pi_3, \mu_{c1}, \sigma_{c1}^2, \rho_{c1}, \mu_{n1}, \sigma_{n1}^2, \rho_{n1}, \mu_2, \sigma_2^2, \rho_2, \mu_{c3}, \sigma_{c3}^2, \rho_{c3}, \mu_{n3}, \sigma_{n3}^2, \rho_{n3}.$

where π_1, π_2, and π_3 are the mixing proportions; μ_{c1}, σ_{c1}^2, and ρ_{c1} are the marginal mean, variance, and correlation of gene expression levels of cluster 1 (up-regulated genes) for diseased subjects; μ_{n1}, σ_{n1}^2, and ρ_{n1} are the marginal mean, variance, and correlation of gene expression levels of cluster 1 (up-regulated genes) for non-diseased subjects; μ_2, σ_2^2, and ρ_2 are the marginal mean, variance, and correlation of gene expression levels of cluster 2 (non-differentially expressed genes); μ_{c3}, σ_{c3}^2, and ρ_{c3} are the marginal mean, variance, and correlation of gene expression levels of cluster 3 (up-regulated genes) for diseased subjects; μ_{n3}, σ_{n3}^2, and ρ_{n3} are the marginal mean, variance, and correlation of gene expression levels of cluster 3 (up-regulated genes) for non-diseased subjects.

Note that genes in cluster 2 are non-differentially expressed across abnormal and normal tissue samples. Hence there are only 3 parameters for cluster 2.

To make sure the identifiability, we set the following contraints: $\mu_{c1} > \mu_{n1}$ and $\mu_{c3} < \mu_{n3}$.

To make sure the marginal covariance matrices are poisitive definite, we set the following contraints: $-1/(n_c - 1) < \rho_{c1} < 1$, $-1/(n_n - 1) < \rho_{n1} < 1$, $-1/(n-1) < \rho_2 < 1$, $-1/(n_c - 1) < \rho_{c3} < 1$, $-1/(n_n - 1) < \rho_{n3} < 1$.

We also has the following constraints for the mixing proportion: $\pi_3 = 1 - \pi_1 - \pi_2$, $\pi_k > 0$, $k = 1, 2, 3$.

We apply the EM algorithm to estimate the model parameters. We regard the cluster membership of genes as missing values.

To facilitate the estimation of the parameters, we reparametrize the parameter vector as $\theta^* = (\pi_1$, π_2, μ_{c1}, σ_{c1}^2, r_{c1}, δ_{n1}, σ_{n1}^2, r_{n1}, μ_2, σ_2^2, r_2, μ_{c3}, σ_{c3}^2, r_{c3}, δ_{n3}, σ_{n3}^2, $r_{n3})$, where $\mu_{n1} = \mu_{c1} - \exp(\delta_{n1})$, $\mu_{n3} = \mu_{c3} + \exp(\delta_{n3})$, $\rho_{c1} = (\exp(r_{c1}) - 1/(n_c - 1))/(1 + \exp(r_{c1}))$, $\rho_{n1} = (\exp(r_{n1}) - 1/(n_n - 1))/(1 + \exp(r_{n1}))$, $\rho_2 = (\exp(r_2) - 1/(n-1))/(1 + \exp(r_2))$, $\rho_{c3} = (\exp(r_{c3}) - 1/(n_c - 1))/(1 + \exp(r_{c3}))$, $\rho_{n3} = (\exp(r_{n3}) - 1/(n_n - 1))/(1 + \exp(r_{n3}))$.

Given a gene, the expression levels of the gene are assumed independent. However, after scaling, the scaled expression levels of the gene are no longer independent and the rank $r^* = r - 1$ of the covariance matrix for the scaled gene profile will be one less than the rank r for the un-scaled gene profile Hence the covariance matrix of the gene profile will no longer be positive-definite. To avoid this problem, we delete a tissue sample after scaling since its information has been incorrporated by other scaled tissue samples. We arbitrarily select the tissue sample, which has the biggest label number, from the tissue sample group that has larger size than the other tissue sample group. For example, if there are 6 cancer tissue samples and 10 normal tissue samples, we delete the 10-th normal tissue sample after scaling.

Value

A list contains 13 elements.

`dat` the (transformed) microarray data matrix. If tranformation performed, then `dat` will be different from the input microarray data matrix.

`memSubjects` the same as the input `memSubjects`.

`memGenes` a vector of cluster membership of genes. 1 means up-regulated gene; 2 means non-differentially expressed gene; 3 means down-regulated gene.

`memGenes2` an variant of the vector of cluster membership of genes. 1 means differentially expressed gene; 0 means non-differentially expressed gene.

`para` parameter estimates (c.f. details).

`llkh` value of the loglikelihood function.

`wiMat` posterior probability that a gene belongs to a cluster given the expression levels of this gene. Column i is for cluster i.

`memIni` the initial cluster membership of genes.

`paraIni` the parameter estimates based on initial gene cluster membership.

`llkhIni` the value of loglikelihood function.

`lambda` the parameter used to do Box-Cox transformation

`paraRP` parameter estimates for reparametrized parameter vector (c.f. details).

`paraIniRP` the parameter estimates for reparametrized parameter vector based on initial gene cluster membership.

Note

The speed of the program can be slow for large data sets, however it has been improved using Fortran code.

Author(s)

Jarrett Morrow `<remdj@channing.harvard.edu>`, Weiliang Qiu `<Weiliang.Qiu@gmail.com>`, Wenqing He `<whe@stats.uwo.ca>`, Xiaogang Wang `<stevenw@mathstat.yorku.ca>`, Ross Lazarus `<ross.lazarus@channing.harvard.edu>`

References

Qiu, W.-L., He, W., Wang, X.-G. and Lazarus, R. (2008). A Marginal Mixture Model for Selecting Differentially Expressed Genes across Two Types of Tissue Samples. *The International Journal of Biostatistics. 4(1):Article 20.* `http://www.bepress.com/ijb/vol4/iss1/20`

See Also

`gsMMD`, `gsMMD.default`, `gsMMD2`

Examples

```
## Not run:
  library(ALL)
  data(ALL)
  eSet1 <- ALL[1:100, ALL$BT == "B3" | ALL$BT == "T2"]
  mat <- exprs(eSet1)

  mem.str <- as.character(eSet1$BT)
  nSubjects <- length(mem.str)
  memSubjects <- rep(0, nSubjects)
  # B3 coded as 0, T2 coded as 1
  memSubjects[mem.str == "T2"] <- 1

  myWilcox <-
  function(x, memSubjects, alpha = 0.05)
  {
      xc <- x[memSubjects == 1]
      xn <- x[memSubjects == 0]

      m <- sum(memSubjects == 1)
      res <- wilcox.test(x = xc, y = xn, conf.level = 1 - alpha)
      res2 <- c(res$p.value, res$statistic - m * (m + 1) / 2)
      names(res2) <- c("p.value", "statistic")

      return(res2)
    }

    tmp <- t(apply(mat, 1, myWilcox, memSubjects = memSubjects))
    colnames(tmp) <- c("p.value", "statistic")
    memIni <- rep(2, nrow(mat))
    memIni[tmp[, 1] < 0.05 & tmp[, 2] > 0] <- 1
    memIni[tmp[, 1] < 0.05 & tmp[,2] < 0] <- 3

    cat("initial gene cluster size>>\n"); print(table(memIni)); cat("\n");

    obj.gsMMD <- gsMMD2.default(mat, memSubjects, memIni = memIni,
            transformFlag = TRUE, transformMethod = "boxcox", scaleFlag = TRUE)
    round(obj.gsMMD$para, 3)

## End(Not run)
```

`obtainResi` *Replace expression levels by the residuals of regression analysis to remove the confounding effects.*

Description

Replace expression levels by the residuals of regression analysis in which predictor of interest is not in the regression model. The purpose of this function is to remove potential confounding factors.

Usage

```
obtainResi(es, fmla)
```

Arguments

`es` An `ExpressionSet` object.

`fmla` A formula object that specifies the covariates of the linear regression model. The variable of interest should not be included. No response variable should be specified in `fmla` since the response variable is always the expression level. See function `lmFit` of R Bioconductor package `limma`.

Details

To remove confounding effects, we can replace the expression level by the residuals of a linear regression model with response variable the expression level and covariates the potential confounders. The functions `lmFit` and `eBayes` will be used to obtain regression coefficients.

Value

An ExpressionSet object with expression levels replaced by residuals of linear regression analysis.

Note

The number of arrays of the returned ExpressionSet object might be smaller than that of the original ExpressionSet object, due to missing values in covariates.

Author(s)

Jarrett Morrow `<remdj@channing.harvard.edu>`, Weiliang Qiu `<Weiliang.Qiu@gmail.com>`, Wenqing He `<whe@stats.uwo.ca>`, Xiaogang Wang `<stevenw@mathstat.yorku.ca>`, Ross Lazarus `<ross.lazarus@channing.harvard.edu>`

`plotHistDensity` *Plot of histogram and density estimate of the pooled gene expression levels.*

Description

Plot of histogram of pooled gene expression levels, composited with density estimate based on the mixture of marginal distributions. The density estimate is based on the assumption that the marginal correlations between subjects are zero.

Usage

```
plotHistDensity(obj.gsMMD,
                plotFlag = "case",
                plotComponent = FALSE,
                myxlab = "expression level",
                myylab = "density",
                mytitle = "Histogram with estimated density (case)",
                x.legend = NULL,
                y.legend = NULL,
                numPoints = 500,
                mycol = 1:4,
                mylty = 1:4,
                mylwd = rep(3,4),
                cex.main = 2,
                cex.lab = 1.5,
                cex.axis = 1.5,
                cex = 2,
                bty = "n")
```

Arguments

`obj.gsMMD` an object returned by `gsMMD`, `gsMMD.default`, `gsMMD2`, or `gsMMD2.default`

`plotFlag` logical. Indicate the plot will based on which type of subjects.

`plotComponent` logical. Indicate if components of the mixture of marginal distribution will be plotted.

`myxlab` label for x-axis

`myylab` label for y-axis

`mytitle` title of the plot

`x.legend` the x-corrdiates of the legend

`y.legend` the y-corrdiates of the legend

`numPoints` logical. Indicate how many genes will be plots.

`mycol` color for the density estimates (overall and components)

`mylty` line styles for the density estimates (overall and components)

`mylwd` line width for the density estimates (overall and components)

`cex.main` font for main title

`cex.lab` font for x- and y-axis labels

`cex.axis` font for x- and y-axis

`cex` font for texts

`bty` the type of box to be drawn around the legend. The allowed values are ’"o"’ and ’"n"’ (the default).

Details

For a given type of subjects, we pool their expression levels together if the marginal correlations among subjects are zero. We then draw a histogram of the pooled expression levels. Next, we composite density estimates of gene expression levels for the overal distribution and the 3 component distributions.

Value

A list containing coordinates of the density estimates:

`x` sorted pooled gene expression levels for cases or controls.

`x2` a subset of `x` specified by the sequence: `seq(from = 1,to = len.x, by = delta)`, where `len.x` is the length of the vector `x`, and `delta = floor(len.x/numpoints)`.

`y` density estimate corresponding to `x2`

`y1` weighted density estimate for gene cluster 1

`y2` weighted density estimate for gene cluster 2

`y3` weighted density estimate for gene cluster 3

Note

The density estimate is obtained based on the assumption that the marginal correlation among subjects is zero. If the estimated marginal correlation obtained by gsMMD is far from zero, then do not use this plot function.

Author(s)

Jarrett Morrow <remdj@channing.harvard.edu>, Weiliang Qiu <Weiliang.Qiu@gmail.com>, Wenqing He <whe@stats.uwo.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>, Ross Lazarus <ross.lazarus@channing.harvard.edu>

References

Qiu, W.-L., He, W., Wang, X.-G. and Lazarus, R. (2008). A Marginal Mixture Model for Selecting Differentially Expressed Genes across Two Types of Tissue Samples. *The International Journal of Biostatistics. 4(1):Article 20.* http://www.bepress.com/ijb/vol4/iss1/20

Examples

```
  ## Not run: 
library(ALL)
data(ALL)
eSet1 <- ALL[1:100, ALL$BT  ==  "B3" | ALL$BT  ==  "T2"]

mem.str <- as.character(eSet1$BT)
nSubjects <- length(mem.str)
memSubjects <- rep(0,nSubjects)
# B3 coded as 0 (control), T2 coded as 1 (case)
memSubjects[mem.str  ==  "T2"] <- 1

obj.gsMMD <- gsMMD(eSet1, memSubjects, transformFlag  =  TRUE,
  transformMethod  =  "boxcox", scaleFlag  =  TRUE, quiet  =  FALSE)

plotHistDensity(obj.gsMMD, plotFlag  =  "case",
    mytitle  =  "Histogram of for T2 imposed with estimated density (case)",
    plotComponent  =  TRUE,
    x.legend  =  c(0.8, 3),
    y.legend  =  c(0.3, 0.4),
    numPoints  =  500)

## End(Not run)
```

Index