Package 'integIRTy'

April 8, 2025

Type Package

LazyData true

Title Integrating Multiple Modalities of High Throughput Assays Using
Item Response Theory

Version 1.0.8

Date 2025-04-08

Description Provides a systematic framework for
integrating multiple modalities of assays profiled on the same set of
samples. The goal is to identify genes that are altered in cancer
either marginally or consistently across different assays. The
heterogeneity among different platforms and different samples are
automatically adjusted so that the overall alteration magnitude can
be accurately inferred. See Tong and Coombes (2012)
<doi:10.1093/bioinformatics/bts561>.

License Apache License (== 2.0)

Depends R (>= 2.10), ltm, foreach, doParallel

Imports mclust, MASS, abind

Suggests KernSmooth

URL http://oompa.r-forge.r-project.org/

NeedsCompilation no

Author Pan Tong [aut],
Kevin R. Coombes [aut, cre]

Maintainer Kevin R. Coombes <krc@silicovore.com>

Repository CRAN

Date/Publication 2025-04-08 14:40:02 UTC

Contents

Index   **18**

---

calculatePermutedScoreByGeneSampling

Calculate the permuted latent trait by gene sampling

Description

Given the original binary matrix and item parameters, calculate the permuted latent trait by gene sampling. Basically this function permutes within columns and recompute the latent trait using pre-specified item parameters and the permuted binary matrix.

Usage

```
calculatePermutedScoreByGeneSampling(originalMat, dscrmn = dscrmn,
      dffclt = dffclt, c = rep(0, length(dffclt)), fold = 1, parallel=FALSE)
```

Arguments

originalMat	The original response matrix
dscrmn	The estimated item discrimination parameter
dffclt	The estimated item difficulty parameter
c	The estimated item guessing parameter if available
fold	The fold relative to the number of genes present should gene sampling achieve. Default is 1, meaning equal number of genes are sampled. Increasing fold would increase the precesion in estimating empirical P value
parallel	Logical indicating whether to use parallel computing with foreach package as backend.

Details

Both gene sampling and sample label permutation can be used to infer the null distribution of altent traits. For sample label permutation, one can simply first construct the binary matrix after permuting the sample labels and feed it to computeAbility() function together with item parameters

Value

A vector of null latent traits by gene sampling

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

David Magis, Gilles Raiche (2012). Random Generation of Response Patterns under Computerized Adaptive Testing with the R Package catR. Journal of Statistical Software, 48(8), 1-31.

See Also

fitOnSinglePlat, intIRTeasyRun, computeAbility

Examples

```
# number of items and number of genes
nSample <- 10
nGene <- 2000
set.seed(1000)
a <- rgamma(nSample, shape=1, scale=1)
b <- rgamma(nSample, shape=1, scale=1)
# true latent traits
theta <- rnorm(nGene, mean=0)

# probability of correct response (P_ij) for gene i in sample j
P <- matrix(NA, nrow=nGene, ncol=nSample)
for(i in 1:nSample){
P[, i] <- exp(a[i]*(theta-b[i]))/(1+exp(a[i]*(theta-b[i])))
}
# binary matrix
X <- matrix(NA, nrow=nGene, ncol=nSample)
for(i in 1:nSample){
X[, i] <- rbinom(nGene, size=1, prob=P[, i])
}
# IRT fitting
fit2PL <- fitOnSinglePlat(X, model=3)
dffclt <- coef(fit2PL$fit)[, 'Dffclt']
dscrmn <- coef(fit2PL$fit)[, 'Dscrmn']
# estimated null latent trait by gene sampling
scoreNull <- calculatePermutedScoreByGeneSampling(X, dffclt=dffclt,
  dscrmn=dscrmn, fold=1)
```

---

computeAbility | *Calculate latent traits for a given response matrix and item parameters using MLE*

Description

This function calculates the MLE of latent traits for a given response matrix with rows being examinees and columns being items for given item parameters.

Usage

```
computeAbility(respMat, dscrmn = dscrmn, dffclt = dffclt,
c = rep(0, length(dffclt)), parallel=FALSE)
```

Arguments

respMat   The response matrix of 0 and 1 with rows being examinees and columns being items.

dscrmn   A vector of item discrimination parameter.

dffclt   A vector of item difficulty parameter.

c   A vector of guessing parameter. Default is set to all 0 indicating no guessing allowed.

parallel   Logical indicating whether to use parallel computing with foreach package as backend.

Details

This function is a wrapper of the thetaEst() function from catR package (Magis, 2012).

Value

A vector of latent trait estimates for each examinee.

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

David Magis, Gilles Raiche (2012). Random Generation of Response Patterns under Computerized Adaptive Testing with the R Package catR. Journal of Statistical Software, 48(8), 1-31.

See Also

fitOnSinglePlat, intIRTeasyRun, calculatePermutedScoreByGeneSampling

Examples

```
# number of items and number of genes
nSample <- 10
nGene <- 2000
set.seed(1000)
a <- rgamma(nSample, shape=1, scale=1)
b <- rgamma(nSample, shape=1, scale=1)
```

```
# true latent traits
theta <- rnorm(nGene, mean=0)

# probability of correct response (P_ij) for gene i in sample j
P <- matrix(NA, nrow=nGene, ncol=nSample)
for(i in 1:nSample){
P[, i] <- exp(a[i]*(theta-b[i]))/(1+exp(a[i]*(theta-b[i])))
}
# binary matrix
X <- matrix(NA, nrow=nGene, ncol=nSample)
for(i in 1:nSample){
X[, i] <- rbinom(nGene, size=1, prob=P[, i])
}
# IRT fitting
fit2PL <- fitOnSinglePlat(X, model=3)
dffclt <- coef(fit2PL$fit)[, 'Dffclt']
dscrmn <- coef(fit2PL$fit)[, 'Dscrmn']
# estimated latent trait
score <- computeAbility(X, dffclt=dffclt, dscrmn=dscrmn)
```

dichotomize	*A wrapper that is able to dichotomize expression, methylation and CN data*

Description

This function provides a common interface for the user so that data dichotomization can be done easily.

Usage

```
dichotomize(mat, matCtr, assayType = c("Expr", "Methy", "CN"), ...)
```

Arguments

mat	A matrix, either expression, methylation or CN
matCtr	A matrix corresponding to normal controls.
assayType	A character string specifying the assay type. It can only be any of "Expr", "Methy", "CN". For assays none of these types, the program will quite. To run intIRT, the user can manually dichotomize the data and feed them into intIRTeasyRun function.
...	Additional parameters to be passed to the specific dichotomization function.

Value

Abinary matrix of the same dimension as input mat.

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

Tong P, Coombes KR. *integIRTy: a method to identify altered genes in cancer accounting for multiple mechanisms of regulation using item response theory*. Bioinformatics, 2012 Nov 15; 28(22):2861–9.

See Also

[dichotomizeCN](), [dichotomizeExpr](), [dichotomizeMethy]()

Examples

```
data(OV)
binDat_expr <- dichotomize(Expr_T[1:20, ], Expr_N[1:20, ], assayType='Expr')
binDat_methy <- dichotomize(Methy_T[1:20, ], Methy_N[1:20, ], assayType='Methy')
```

---

dichotomizeCN	*Dichotomizing copy number data based on segmented data (i.e. log2ratio).*

Description

A simple dichotomization procedure is implemented for CN data that only requires two cutoffs.

Usage

```
dichotomizeCN(CN, CNctr = NULL, tau1 = -0.3, tau2 = 0.3)
```

Arguments

CN	A matrix of gene wise copy number data for tumor samples. Rows are the genes; columns are the samples.
CNctr	A matrix of copy number data for normal samples. The program first guess if this is paired data by comparing if tumor and normal sample has equal sample size. If TRUE, then normal samples will be subtracted element by element to correct for germline CN change. Otherwise, no correction is performed. The program proceeds with only tumor data.
tau1	The lower bound for log2Ratio when converting to a code as 0.
tau2	The upper bound for log2Ratio when converting to a code as 0. Log2ratio between tau1 and tau2 is converted to 0 and 1 otherwise.

Value

Returns a binary matrix of the same dimension. Missing value would be propogated into the binary matrix.

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

Tong P, Coombes KR. *integIRTy: a method to identify altered genes in cancer accounting for multiple mechanisms of regulation using item response theory*. Bioinformatics, 2012 Nov 15; 28(22):2861–9.

See Also

dichotomizeExpr, dichotomizeMethy

Examples

```
data(OV)
binDat <- dichotomizeCN(CN_T[1:20, ], CN_N[1:20, ])
binDat[15:20, 1:2]
```

---

dichotomizeExpr	*Dichotomize the expression data given both tumor and normal samples.*

Description

This function implements the z-like metric described in the paper.

Usage

```
dichotomizeExpr(expr, exprCtr, refUseMean = FALSE, BIthr = NULL,
                tau1 = -2.5, tau2 = 2.5, parallel = FALSE)
```

Arguments

expr	The expression matrix for tumor samples. Rows are genes and columns are samples.
exprCtr	Expression matrix of normal controls. Genes should exactly the same as the tumor sample. The sample size are not necessarily the same as tumor sample.
refUseMean	Logical indicating whether to use mean of normal sample as reference. Default is set to FALSE which means to use median as it is more robust.
BIthr	Threshold of bimodality index to flag bimodal genes. If not specified, it will be set according to the sample size of tumor samples. Specifically, if tumor sample size is over 100, BIthr=1.1. If sample size is between 50 and 100, BIthr=1.5. If sample size is below 50, BIthr=2.0.
tau1	Lower bound of z-like metric to be coded as 0.

tau2	Upper bound of z-like metric to be coded as 0. The z-like metric between tau1 and tau2 will be finally converted to 0 and otherwise.
parallel	Logical indicating whether to use parallel backend provided by foreach and related packages.

Details

The parallelism is written to speedup BI computation. If the number of genes is not large, i.e. below 4000, we recommend not to use parallel since this will only slow down the computation. In fact, except BI computation, all other operations are written with vector operation.

Value

A binary matrix of the same dimension of input expr. Missing values will be propogated into binary matrix.

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

Tong P, Coombes KR. *integIRTy: a method to identify altered genes in cancer accounting for multiple mechanisms of regulation using item response theory*. Bioinformatics, 2012 Nov 15; 28(22):2861–9.

See Also

dichotomizeCN, dichotomizeMethy, dichotomize

Examples

```
data(OV)
binDat <- dichotomizeExpr(Expr_T[1:200, ], Expr_N[1:200, ])
#binDat <- dichotomizeExpr(Expr_T[1:200, ], Expr_N[1:200, ], parallel=TRUE)
binDat[15:20, 1:2]
```

---

dichotomizeMethy	*Dichotomize the methylation data given both tumor and normal controls.*

Description

This function implements the procedure for dichotomizing methylation data described in the paper.

Usage

```
dichotomizeMethy(methy, methyCtr, refUseMean = FALSE)
```

Arguments

methy	The methylation matrix for tumor samples. Each element represents the beta value which is bounded between 0 and 1. Rows are genes and columns are samples.
methyCtr	Methylation matrix of normal controls. Genes should exactly the same as the tumor sample. The sample size are not necessarily the same as tumor sample.
refUseMean	Logical indicating whether to use mean of normal sample as reference. Default is set to FALSE which means to use median as it is more robust.

Value

A binary matrix of the same dimension of input methy.

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

Tong P, Coombes KR. *integIRTy: a method to identify altered genes in cancer accounting for multiple mechanisms of regulation using item response theory*. Bioinformatics, 2012 Nov 15; 28(22):2861–9.

See Also

[dichotomizeCN](#), [dichotomizeExpr](#), [dichotomize](#)

Examples

```
data(OV)
binDat <- dichotomizeMethy(Methy_T[1:200, ], Methy_N[1:200, ])
binDat[15:20, 1:2]
```

---

fitOnSinglePlat	*Fit IRT model on a single platform*

Description

This function fits the Item Response Model for one platform. It assumes the user has already dichotomized the data.

Usage

```
fitOnSinglePlat(data, model = 2, guessing = FALSE,
    sampleIndices = 1:ncol(data), geneIndices = 1:nrow(data), ...)
```

Arguments

`data`	A matrix of 0's and 1's with rows being genes (treated as examinees) and columns being samples (treated as items).
`model`	IRT model. 1-Rasch model where all item discrination are set to 1; 2-all item discrimation are set to be equal but not necessarily as 1; 3-the 2PL model where no constraint is put on the item difficulty and discrimination parameter.
`guessing`	A logical variable indicating whether to include guessing parameter in the model.
`sampleIndices`	Indices of the samples to be feeded into the model. Default is set to use all samples.
`geneIndices`	Indices of the genes to be feeded into the model. Default is to use all genes.
`...`	Additional options available in ltm package. Currently not used in intIRT package.

Value

A list giving the estimated IRT model and related information

`fit`	An object returned by calling ltm package. Item parameters and other auxillary inforamtion (i.e. loglikelihood, convergence, Hessian) can be accessed from this object. For more details, please refer to ltm package
`model`	The model type
`guessing`	The guessing parameter
`sampleIndices`	The sample indices used in the model
`geneIndices`	The gene indices used in the model

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

Rizopoulos, D. (2006) ltm: An R package for latent variable modelling and item response theory analyses. Journal of Statistical Software, 17(5), 1-25.

See Also

[computeAbility](#), [intIRTeasyRun](#), [calculatePermutedScoreByGeneSampling](#)

Examples

```
# number of items and number of genes
nSample <- 10
nGene <- 2000
set.seed(1000)
a <- rgamma(nSample, shape=1, scale=1)
b <- rgamma(nSample, shape=1, scale=1)
# true latent traits
```

```
theta <- rnorm(nGene, mean=0)

# probability of correct response (P_ij) for gene i in sample j
P <- matrix(NA, nrow=nGene, ncol=nSample)
for(i in 1:nSample){
P[, i] <- exp(a[i]*(theta-b[i]))/(1+exp(a[i]*(theta-b[i])))
}
# binary matrix
X <- matrix(NA, nrow=nGene, ncol=nSample)
for(i in 1:nSample){
X[, i] <- rbinom(nGene, size=1, prob=P[, i])
}
# IRT fitting
fit2PL <- fitOnSinglePlat(X, model=3)
```

---

intIRTeasyRun	*The easyrun function for integrating multiple modalities of high throughput assays using binary input matrix.*

Description

It fits IRT models on each of the specified platform and calculate integrated latent trait. If required, permuted latent trait by gene sampling will also be calculated. An option for parallel computing is implemented to speed up the computation.

Usage

```
intIRTeasyRun(platforms, model = 3, guessing = FALSE,
    addPermutedScore = FALSE, fold = 1, echo = TRUE, parallel = FALSE)
```

Arguments

platforms	A list of response matrices representing different platforms. It assumes that the number of rows (genes) must be equal whiel the number of columns (samples) can be different.
model	The model type as described in fitOnSinglePlat.
guessing	A logical variable indicating whether to include guessing parameter in the model.
addPermutedScore	A logical variable indicating whether to also calculate permuted latent trait by gene sampling.
fold	The fold of sampling to calculate permuted score as used in calculatePermutedScoreByGeneSampling(). Only relevant when addPermutedScore=TRUE is used.
echo	A logical variable indicating whether to print out the progress information.
parallel	Logical indicating whether to use parallel computing with foreach package as backend.

Details

Parallel computing uses foreach and related packages for backend. The parallelism assumes computation on each platform individually takes similar time; the latent trait computation of the integrated data is assumed to be comparable to computation on individual platform. By default, all parallel options are set to be FALSE. Parallelism happens on the individual assay and combined data level; No parallelism happens on genes since it would only slow donw the computation due to data transfering!

Value

A list with following elements:

fits   Model fits for each platform as returned by fitOnSinglePlat function

estimatedScoreMat

                A matrix of estimated latent traits. The first several columns correspond to the
                individual assays; the last column represents the integrated latent trait with all
                data.

permutedScoreMat

                A matrix of latent trait estimates after permuting the binary matrix within columns.
                This is only available if addPermutedScore is set to TRUE. The first several
                columns correspond to the individual assays; the last column represents the integrated data.

dscrmnList   A list of discrimination parameters. Each element contains all of the discrimination parameters as a vector for each assay. The last element contains the discrimination parameters for the integrated data which is formed by combining discrimination parameters from each assay sequentially.

dffcltList   Same format as dscrmnList except it contains difficulty parameter.

gussngList   Same format as dscrmnList except it contains guessing parameter. Be default, this is just all 0's.

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

Tong P, Coombes KR. *integIRTy: a method to identify altered genes in cancer accounting for multiple mechanisms of regulation using item response theory*. Bioinformatics, 2012 Nov 15; 28(22):2861–9.

See Also

intIRTeasyRunFromRaw, fitOnSinglePlat, calculatePermutedScoreByGeneSampling

intIRTeasyRunFromRaw | *The easyrun function for integrating multiple modalities of high throughput assays using raw data.*

Description

This function performs data dichotomization, IRT fitting on individual assay, latent trait estimation for integrated data and significance assessment of latent trait by permutation. An option for parallel computing is implemented to speed up the computation.

Usage

```
intIRTeasyRunFromRaw(platforms, platformsCtr,
assayType = c("Expr", "Methy", "CN"),
model = 3, guessing = FALSE, permutationMethod = NULL,
fold = 1, nPerm = 200, echo = TRUE,
parallel = FALSE, ...)
```

Arguments

platforms
: A list of matrices of the raw data for tumor samples. The matrices should have equal row number corresponding to the same set of genes. The columns representing the tumor samples can differ.

platformsCtr
: A list of matrices of the raw data for normal control samples. The matrices should have equal row number corresponding to the same set of genes. The column number of each matrix can differ. When normal control is not available, i.e. in CN data, use NA instead.

assayType
: A vector of assay types. Candidates can only be a combination of "Expr", "Methy", "CN" in the order of the assays specified in the input platforms. When assays other then these three types, we recommend the user to dichotomize the data first and use the intIRTeasyRun function.

model
: The model type as described in fitOnSinglePlat. 1: Rasch model where all item discrination are set to 1; 2: all item discrimation are set to be equal but not necessarily as 1; 3: the 2PL model where no constraint is put on the item difficulty and discrimination parameter.

guessing
: A logical variable indicating whether to include guessing parameter in the model.

permutationMethod
: What permutation method to use. It can only be 'gene sampling', 'sample label permutation' or NULL. if NULL, no permutation is performed

fold
: The fold of sampling to calculate permuted score as used in calculatePermutedScoreByGeneSampling(). Only relevant when permutationMethod=gene sampling is used.

nPerm
: Number of permutations for sample label permutation. It is effective only when permutationMethod='sample label permutation'.

echo
: A logical variable indicating whether to print out the progress information.

parallel	Logical indicating whether to use parallel computing with foreach package as backend.
...	Additional parameters for dichotomizing binary data.

Value

A list quite similar to the results returned by intIRTeasyRun. The following elements are included:

fits	Model fits for each platform as returned by fitOnSinglePlat function
estimatedScoreMat	
	A matrix of estimated latent traits. The first several columns correspond to the individual assays; the last column represents the integrated latent trait with all data.
permutedScoreMat	
	A matrix of latent trait estimates after permuting the binary matrix within columns. This is only available if permutationMethod='gene sampling' is used. The first several columns correspond to the individual assays; the last column represents the integrated data.
dscrmnList	A list of discrimination parameters. Each element contains all of the discrimination parameters as a vector for each assay. The last element contains the discrimination parameters for the integrated data which is formed by combining discrimination parameters from each assay sequentially.
dffcltList	Same format as dscrmnList except it contains difficulty parameter.
gussngList	Same format as dscrmnList except it contains guessing parameter. Be default, this is just all 0's.
permutedScoreMatWithLabelPerm	
	A matrix of latent trait estimates using sample label permutation. This is only available if permutationMethod='sample label permutation' is used. The first several columns correspond to the individual assays; the last column represents the integrated data.

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

References

Tong P, Coombes KR. *integIRTy: a method to identify altered genes in cancer accounting for multiple mechanisms of regulation using item response theory*. Bioinformatics, 2012 Nov 15; 28(22):2861–9.

See Also

[intIRTeasyRun](), [fitOnSinglePlat](), [calculatePermutedScoreByGeneSampling]()

Examples

```
data(OV)
#
controlList <- list(Expr_N, Methy_N, CN_N)
tumorList <- list(Expr_T, Methy_T, CN_T)
# not run as it takes time
#runFromRaw <- intIRTeasyRunFromRaw(platforms=tumorList,
# platformsCtr=controlList,
# assayType=c("Expr", "Methy", "CN"),
# permutationMethod="gene sampling")
```

OV   *Ovarian Cancer Datasets*

Description

Six matrices containing a subset of TCGA ovarian cancer data.

Usage

```
data(OV)
```

Format

Each of the six objects (CN_N, CN_T, Methy_N, Methy_T, Expr_N, Expr_T) is a matrix with rows for 1000 (matched) genes and columns as samples. Gene expression, methylation and copy number data for 30 tumor samples and around 10 normal samples are provided.

Source

This data is a subset of the TCGA ovarian cancer datasets. The full datasets can be downloaded through the TCGA data portal at: http://cancergenome.nih.gov/

References

Cancer Genome Atlas Research Network (2011). Integrated genomic analyses of ovarian carcinoma. Nature 474, 609-615

simulateBinaryResponseMat

> *Simulate binary response matrix according to 2-parameter Item Characteristic Function for given latent traits and item parameters.*

Description

This function generates binary response matrix according to the Item Characteristic Function for specified item parameter and latent traits. It can be used for simulation purposes.

Usage

```
simulateBinaryResponseMat(a = a, b = b, theta = theta)
```

Arguments

a	A vector of item discrimination parameter
b	A vector of item difficulty parameter
theta	A vector of true latent traits

Details

This function is not necessary for the integration purpose. It serves as a utility function to help the user conduct simulation.

Value

A matrix of 0's and 1's where rows are genes (examinees) and columns are samples (items).

Author(s)

Pan Tong (nickytong@gmail.com), Kevin R Coombes (krc@silicovore.com)

See Also

computeAbility, fitOnSinglePlat, intIRTeasyRun

Examples

```
# number of samples and genes to simulate
nSample <- 50
nGene <- 1000
# mean and variance of item parameters
meanDffclt_Expr <- 3; varDffclt_Expr <- 0.2
meanDscrmn_Expr <- 1.5; varDscrmn_Expr <- 0.1
# generate item parameters from gamma distribution
set.seed(1000)
Dffclt_Expr <-  rgamma(nSample, shape=meanDffclt_Expr^2/varDffclt_Expr,
```

```
                        scale=varDffclt_Expr/meanDffclt_Expr)
Dscrmn_Expr <-  rgamma(nSample, shape=meanDscrmn_Expr^2/varDscrmn_Expr,
                        scale=varDscrmn_Expr/meanDscrmn_Expr)
# generate latent trait
theta <- rnorm(nGene)
# the binary response matrix
binary_Expr <- simulateBinaryResponseMat(a=Dscrmn_Expr, b=Dffclt_Expr, theta=theta)
dim(binary_Expr)
```

Index