Package 'repfdr'

September 28, 2017

Type Package

Title Replicability Analysis for Multiple Studies of High Dimension

Version 1.2.3

Description Estimation of Bayes and local Bayes false discovery rates for replicability analysis (Heller & Yekutieli, 2014 <doi:10.1214/13-AOAS697> ; Heller at al., 2015 <doi: 10.1093/bioinformatics/btu434>).

License GPL (>= 2)

Depends R (>= 2.10)

Imports splines,Rcpp (>= 0.12.6)

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

URL https://github.com/barakbri/repfdr

BugReports https://github.com/barakbri/repfdr/issues

Author Ruth Heller [cre, aut],
Shachar Kaufman [aut],
Shay Yaacoby [aut],
David Israeli [aut],
Barak Brill [aut],
Daniel Yekutieli [aut],
Stephen Turner [cph]

Maintainer Ruth Heller <ruheller@gmail.com>

Date/Publication 2017-09-28 10:52:07 UTC

Suggests R.rsp

VignetteBuilder R.rsp

R topics documented:

`binned_zmat` *Three GWAS studies - input objects to main function*

Description

This data was created from the `zmat` matrix (see `SNPlocations`) using `ztobins` function. It contain two objects to be input to the main function `repfdr`.

Format

The file includes two objects - a matrix and 3d array:

`bz` is a matrix of binned 249024 z-scores (in rows) in each of the 3 studies (columns).

`pbz` is a 3-dimensional array which contains for each study (first dimension), the probabilities of a z-score to fall in the bin (second dimension), under each hypothesis status (third dimension).

Examples

```
## Not run:
download.file('http://www.math.tau.ac.il/~ruheller/repfdr_RData/binned_zmat.RData',
  destfile = "binned_zmat.RData")
load(file = "binned_zmat.RData")

bz[1:5,]
pbz[,1:5,]

## End(Not run)
```

binned_zmat_sim *Simulated data set - input objects to main function*

Description

This data was created from the zmat_sim matrix using ztobins function. It contain two objects to be input to the main function repfdr.

Usage

```
data(binned_zmat_sim)
```

Format

The file includes two objects - a matrix and 3d array:

bz_sim is a matrix of binned 10000 z-scores (in rows) in each of the 3 studies (columns).

pbz_sim is a 3-dimensional array which contains for each study (first dimension), the probabilities of a z-score to fall in the bin (second dimension), under each hypothesis status (third dimension).

Examples

```
data(binned_zmat_sim)
bz_sim[1:5,]
pbz_sim[,1:5,]
```

em.control *Control Parameters for the EM algorithm*

Description

Input parameters for the EM algorithm.

Usage

```
em.control(pi.initial = NULL, max.iter = 10000, tol = 1e-12,
           nr.threads = 0, verbose = TRUE)
```

Arguments

pi.initial	Initial guess for the probabilities of the vectors of associations status. If NULL then 0.9 is assigned for the c(0,...,0) configuration and 0.1 is distributed uniformly for all other configurations.
max.iter	Maximum number of EM iterations.
tol	Tolerance (in maximum absolute difference between two EM iterations in estimated probabilities) before declaring convergence and stopping.

nr.threads	Number of processing threads to use. If zero (the default), will automatically detect the number of compute cores available and spawn one thread per core.
verbose	An indicator of whether to report progress (running iteration number) during computation.

Details

The function is used inside the control argument in repfdr and piem.

Value

A list with the input values.

See Also

repfdr piem

Examples

```
## Not run:
download.file('http://www.math.tau.ac.il/~ruheller/repfdr_RData/binned_zmat.RData',
  destfile = "binned_zmat.RData")
load(file = "binned_zmat.RData")
out <- repfdr(pbz,bz,"replication",
              control = em.control(pi.initial = c(0.48,rep(0.02,26)),
              verbose = TRUE, nr.threads = 1))
# iterations are printed; run bit slower (1 thread)

## End(Not run)
```

---

hconfigs *Enumeration of all possible vectors of association status.*

Description

The function generates a matrix with all possible vectors of association status (in rows), given the number of studies and number of possible association status states in each study (2 or 3).

Usage

```
hconfigs(n.studies, n.association.status = 3, studies.names = NULL)
```

Arguments

n.studies	Number of studies in the analysis.
n.association.status	either 2 for no-association\association or 3 for no-association\negative-association\positive-association.
studies.names	Optional study names to display.

Details

This matrix should be used when selecting the rows indices for the association status vectors that are in the non-null set, specified by the used in `non.null.rows` in the function `repfdr`.

Value

Matrix with rows indicating all the possible vectors of association status.

See Also

`repfdr`

Examples

```
(H <- hconfigs(n.studies = 3))
# in replication analysis the non-null vectors are:
H[apply(H,1,function(y){ sum(y==1)>1 | sum(y==-1)>1 }),]
# in meta-analysis there is only one null vector (c(0,0,0)):
H[rowSums(abs(H))!=0,]

hconfigs(n.studies = 3, n.association.status= 2)
```

---

hmat_sim *Simulated data set - indicators of association status matrix*

Description

A matrix of size 10000x3 of indicators of whether each z-score from `zmat_sim` belongs to a non-null hypothesis for the feature in the study (1) or to a null hypothesis for the feature in the study (0).

Usage

```
data(hmat_sim)
```

Format

`hmat_sim` is a matrix of 10000 rows, each row a vector of the true association status from which the z-scores in the same row in `zmat_sim` was generated. Specifically, for a zero entry in hmat_sim the corresponding z-score in `zmat_sim` was generated from the standard normal distribution, and for a unit entry in hmat_sim the corresponding z-score in `zmat_sim` was generated from the normal distribution with mean 3 and variance one.

Examples

```
#### use hmat_sim to generate the simulated z-scores:

data(hmat_sim)
m <- nrow(hmat_sim)
set.seed(12)
zmat_sim1  <- matrix(rnorm(n=3*m,mean=hmat_sim*3),nrow=m,ncol=3)
rm(m,H)

data(zmat_sim)
stopifnot(all.equal(zmat_sim1,zmat_sim))

#### hmat_sim was generated by the following code:

H <- hconfigs(n.studies= 3, n.association.status=2)
f <- c(0.895,0.005,0.005,0.02,0.005,0.02,0.02,0.03) # frequencies for the association status vectors
m = 10000 # number of tests in each study
hmat_sim1 <- matrix(rep(x = H, times = m*cbind(f,f,f)),ncol=3)

data(hmat_sim)
stopifnot(all.equal(hmat_sim1,hmat_sim))

# the simulation design
cbind(H,f)
sum(f)      # all sum to 1?
```

`ldr` *Estimation of posterior probabilities for the vectors of association status*

Description

The function finds the posterior probabilities ofeach vector of association status for each feature, given the feature's vector of binned z-scores.

Usage

```
ldr(pdf.binned.z, binned.z.mat, Pi, h.vecs = NULL)
```

Arguments

`pdf.binned.z`	Same input as in `repfdr`. A 3-dimensional array which contains for each study (first dimension), the probability of a z-score to fall in the bin (second dimension), under each hypothesis status (third dimension). The third dimension can be of size 2 or 3, depending on the number of association states: if the association can be either null or only in one direction, the dimension is 2; if the association can be either null, or positive, or negative, the dimension is 3. Element `[[1]]` in the output of `ztobins`.

binned.z.mat	Same input as in repfdr. A matrix of the bin numbers for each of the z-scores (rows) in each study (columns). Element [[2]] in the output of ztobins.
Pi	The estimated prior probabilities for each association status vector. Can be extracted from the output of repfdr or piem, see Example section.
h.vecs	The row indices in H (see hconfigs), corresponding to the association status vectors. By default the posterior probabilities of all possible vectors of association status are computed.

Details

A subset of features (e.g most significant) can be specified as the rows in binned.z.mat, so the posterior probabilities of the vectors of association status are computed for this subset of features. See Example section.

Value

Matrix with rows that contain for each of the vectors of association status the posterior probabilities. The columns are the different feature.

See Also

repfdr, piem, hconfigs

Examples

```
## Not run:
download.file('http://www.math.tau.ac.il/~ruheller/repfdr_RData/binned_zmat.RData',
  destfile = "binned_zmat.RData")
load(file = "binned_zmat.RData")

data(Pi)

# Fdr calculation:
output3 <- repfdr(pbz, bz, "replication",Pi.previous.result = Pi)

BayesFdr <- output3$mat[,"Fdr"]
sum(BayesFdr <= 0.05)

# The posterior probabilities for the the first five features with Bayes FDR at most 0.05:
post <- ldr(pbz,bz[which(BayesFdr <= 0.05)[1:5],],Pi)
round(post,4)

# posteriors for a subset of the association status vectors can also be reported,
# here the subset is the four first association status vectors:
post <- ldr(pbz,bz[which(BayesFdr <= 0.05)[1:5],],Pi,h.vecs= 1:4)
round(post,4)

## End(Not run)
```

piem *Estimation of the prior probabilities for each association status vector.*

Description

The function calls an expectation-maximization (EM) algorithm to estimate the prior probabilities of each association status vector. It is also used internally in repfdr.

Usage

```
piem(pdf.binned.z, binned.z.mat, control = em.control())
```

Arguments

pdf.binned.z	Same input as in repfdr. A 3-dimensional array which contains for each study (first dimension), the probabilities of a z-score to fall in the bin (second dimension), under each hypothesis status (third dimension). The third dimension can be of size 2 or 3, depending on the number of association states: if the association can be either null or only in one direction, the dimension is 2; if the association can be either null, or positive, or negative, the dimension is 3. Element [[1]] in the output of ztobins.
binned.z.mat	Same input as in repfdr. A matrix of the bin numbers for each the z-scores (rows) in each study (columns). Element [[2]] in the output of ztobins.
control	List of control parameters to pass to the EM algorithm. See em.control.

Details

The implementation of the EM algorithm is in C, and allows paralel processing. By default, the software automatically detects the number of available processing threads. See em.control for the option of providing the number of threads to use, as well as for the additional control parameters.

Value

all.iterations	Matrix with number of columns equal to the number of EM iterations, and each column is the estimated probability distribution of the vector of association status.
last.iteration	Matrix of the vectors of association status along with the column vector of the last EM iteration, which contains the estimated probabilities of the vectors of association status.

Author(s)

C implementation by Shachar Kaufman.

References

Heller, Ruth, and Daniel Yekutieli. "Replicability analysis for Genome-wide Association studies." *arXiv preprint arXiv:1209.2829* (2012).

See Also

repfdr

Examples

```
## Not run:

download.file('http://www.math.tau.ac.il/~ruheller/repfdr_RData/binned_zmat.RData',
  destfile = "binned_zmat.RData")
load(file = "binned_zmat.RData")
#binned_zmat can also be generated via
output_piem <- piem(pbz, bz)

# extract the last iteration to use it in repfdr (see help(repfdr)):
Pi1 <- output_piem$last.iteration
data(Pi)
stopifnot(all.equal(Pi,Pi1))

# simulation data:
data(binned_zmat_sim)
output_piem_sim <- piem(pbz_sim, bz_sim)
Pi_sim <- output_piem_sim$last.iteration

# following are the true proportions in the data: (see help(hmat_sim) for data generation details.)
f <- c(0.895,0.005,0.005,0.02,0.005,0.02,0.02,0.03)

# the estimation vs the true proportions:
cbind(round(Pi_sim,6),f)

## End(Not run)
```

---

repfdr *Bayes and local Bayes false discovery rate estimation for replicability analysis*

Description

Estimate Bayes and local Bayes false discovery rates (FDRs) from multiple studies, for replicability analysis and for meta-analysis, as presented in Heller and Yekutieli (see reference below).

Usage

```
repfdr(pdf.binned.z, binned.z.mat,
       non.null = c("replication", "meta-analysis",
       "user.defined"),
       non.null.rows = NULL,Pi.previous.result = NULL,
       control = em.control(), clusters = NULL,
       clusters.ldr.report=NULL, clusters.verbose=T)
```

Arguments

pdf.binned.z	A 3-dimensional array which contains for each study (first dimension), the probabilities of a z-score to fall in the bin (second dimension), under each hypothesis status (third dimension). The third dimension can be of size 2 or 3, depending on the number of association states: if the association can be either null or non-null (e.g. only in one direction), the dimension is 2; if the association can be either null, or positive, or negative, the dimension is 3. Element [[1]] in the output of ztobins.
binned.z.mat	A matrix of the bin numbers for each of the z-scores (rows) in each study (columns). Element [[2]] in the output of ztobins.
non.null	Indicates the desired analysis: replication, meta-analysis or user.defined. When user.defined is selected non.null.rows must be specified.
non.null.rows	Vector of row indices in H (see hconfigs), indicating which vectors of association status should be considered as non-null in the analysis. H is the output of hconfigs(dim(pdf.binned.z)[1], dim(pdf.binned.z)[3]), i.e. the matrix with rows indicating the possible vectors of association status, where dim(pdf.binned.z)[1] is the number of studies and dim(pdf.binned.z)[3] is the number of association states in each study (2 or 3).
Pi.previous.result	An optional Vector of probabilities for each association status. If NULL, then the probabilities are estimated with the EM algorithm. An estimation result from a previous run of repfdr or piem can be supplied to shorten the run-time of the function, see Example section.
control	List of control parameters to pass to the EM algorithm. See em.control.
clusters	Used for performing analysis in each cluster, and than aggregating results together.Default value is NULL (no clusters in data). To use clusters, argument must be vector of integer, filled with number from 1 to wanted number of clusters. Vector is of length of number of studies, where clusters[i] is the cluster membership of the ith study. NULL
clusters.ldr.report	Sets whether local fdr values (available through the function ldr for non clustered data) should be displayed with the output. Default value is NULL (no ldr values reported). Other options are 'ALL' (all ldr valeus are reported) or a vector of integers for the indices of the SNPs to be reported)
clusters.verbose	if set to TRUE, messages will be printed to screen regarding the state of the calculation (which cluster is currently being processes, and aggregation procecdure by SNP). Default is FALSE

Details

For N studies, each examining the same M features, the binned z-scores and the (estimated) probabilities under the null and non-null states in each study are given as input. These inputs can be produced from the z-scores using the function ztobins.

The function calls piem for the computation of the probabilities for each vector of association status. The number of probabilies estimated is x^N, where x=2,3 is the number of possible association states in each study.

The function calls ldr for the computation of the conditional probability of each of the vectors of association status in the null set given the binned z-scores. The null set contains the rows in hconfigs(N,x) that: are excluded from non.null.rows if non.null is user.defined; that are non-zero if non.null is meta-analysis; that contain at most one 1 if non.null is replication and x=2; that contain at most one 1 or one -1 if non.null is replication and x=3.

The local Bayes FDR is estimated to be the sum of conditional probabilities in the null set for each feature. The empirical Bayes FDR is the average of all local Bayes FDRs that are at most the value of the local Bayes FDR for each feature. The list of discoveries at level q are all features with empirical Bayes FDR at most q.

If many studies are available, one may not be able to compute RepFDR directly at the original data. If however, different groups of studies are known to be independent (e.g., if a SNP is non null for studies 1,2 is independent of the SNP being non null in studies 3,4) one may Run RepFDR in each cluster seperatly and then aggregate the results. This is done by providing a vector for the clusters argument, with an integer value stating the cluster membership for each study. See the values section below for the results returned from this function, when partitioning the data to clusters. See vignette('RepFDR') for a complete example, on how to run RepFDR in clusters.

Value

mat
: An Mx2 Matrix with a row for each feature (M rows) and two columns, the estimated local Bayes FDR (fdr) and the estimated Bayes FDR (Fdr).

Pi
: Vector of the estimated probabilities for each of the x^N possible vectors of association status. If clusters is not NULL, A matrix with number of rows being choose(2+nr_studies,2) (for n.association being 3) or choose(1+nr_studies,1) (for n.association being 3). The last column represents an aggregated probability over all combintaions for row.

repfdr.mat.percluster
: Returned if clusters is not NULL. A list of the mat values returned from the RepFDR analysis, per cluster.

repfdr.Pi.percluster
: Returned if clusters is not NULL. A list of the Pi values returned from the RepFDR analysis, per cluster.

ldr
: Returned if clusters.ldr.report is not NULL. A matrix with number of rows being choose(2+nr_studies,2) (for n.association being 3) or choose(1+nr_studies,1) (for n.association being 3). Each row holds the local fdr values for a combination of non null values, for the reported SNPs. The first three columns count the number of studies for each reported state (number of null studies, number of non null studies). Other values in the row give the local fdr value, per reported SNP, for the specificed system of hypotheses.

Author(s)

Ruth Heller, Shachar Kaufman, Shay Yaacoby, Barak Brill, Daniel Yekutieli.

References

Heller, R., & Yekutieli, D. (2014). Replicability analysis for genome-wide association studies. *The Annals of Applied Statistics*, 8(1), 481-498.

Heller, R., Yaacoby, S., & Yekutieli, D. (2014). repfdr: a tool for replicability analysis for genome-wide association studies. *Bioinformatics*, btu434.

Examples

```
#### Example 1: a simulation; each feature in each study has two association states,
####            null and positive, prior is known
#This example generates the Z scores for two studies, with 0.05 probability to have
# non - null signal in each study.
# The prior matrix is being pregenerated to show the optimal values.
# if this matrix was not supplied, the repfdr method would estimate it
# using an EM algorithm. See the next examples for estimating the prior as well using repfdr.

set.seed(1)
n = 2 #two studies
m=10000 # ten thounsand, SNPs
H_Study_1 = rbinom(m,1,prob = 0.05) #signal of 1, for SNPS with association in the first study
H_Study_2 = rbinom(m,1,prob = 0.05) #signal of 1, for SNPS with association in the second study
Zmat = matrix(rnorm(n*m),nrow = m) #generate matrix

#insert signal (mean shift of 3) for the first study
Zmat[which(H_Study_1==1),1] = Zmat[which(H_Study_1==1),1]  + 4

#insert signal to the second study
Zmat[which(H_Study_2==1),2] = Zmat[which(H_Study_2==1),2]  + 4

#estimate densities via ztobins:
ztobins_res = ztobins(Zmat,n.association.status = 2,plot.diagnostics = FALSE,n.bin= 100)

#writing out the prior explicitly. If this was not supplied,
#the repfdr would try to estimate this prior from the data.
Precomputed_Pi = matrix(NA,ncol = 3,nrow = 4)
Precomputed_Pi[,1] = c(0,1,0,1)
Precomputed_Pi[,2] = c(0,0,1,1)
Precomputed_Pi[,3] = c(0.95^2,0.95*0.05,0.95*0.05,0.05^2)
colnames(Precomputed_Pi) = c('Study 1','Study 2','Pi')

#run repfdr
repfdr_res = repfdr(ztobins_res$pdf.binned.z,
                    ztobins_res$binned.z.mat,
                    non.null = 'replication',
                    Pi.previous.result = Precomputed_Pi)

#The precomputed prior matrix. if this would not
repfdr_res$Pi

#local fdr0 and Fdr for each SNP
head(repfdr_res$mat)

Non_Null = which(H_Study_1 ==1 & H_Study_2 == 1)
Reported = which(repfdr_res$mat[,2] <= 0.05)
TP = length(intersect(Reported,  Non_Null))
```

```
TP
FP = length(Reported) - TP
FP
FN = length(Non_Null - TP)
FN


#### Example 2: a simulation; each feature in each study has two association states,
####            null and positive, prior is estimated
## Not run:
# a) Replicablity analysis:
data(binned_zmat_sim) # this loads the binned z-scores as well as the (estimated) probabilities
# in each bin for each state
output.rep <- repfdr(pbz_sim, bz_sim, "replication")
BayesFdr.rep <- output.rep$mat[,"Fdr"]
Rej <- (BayesFdr.rep <= 0.05)
sum(Rej)

# which of the tests are true replicability findings? (we know this since the data was simulated)
data(hmat_sim)
true.rep   <- apply(hmat_sim,1,function(y){ sum(y==1)>1 })


# Compute the false discovery proportion (FDP) for replicability:
sum(Rej * !true.rep) / sum(true.rep)

# we can use the previously calculated Pi for further computations (e.g meta-analysis):
Pi_sim <- output.rep$Pi

# b) meta-analysis:
output.meta <- repfdr(pbz_sim, bz_sim, "meta-analysis", Pi.previous.result = Pi_sim)

BayesFdr.meta <- output.meta$mat[,"Fdr"]
Rej <- (BayesFdr.meta <= 0.05)
sum(Rej)

# which of the tests are true association findings? (we know this since the data was simulated)
true.assoc <- rowSums(hmat_sim) >= 1

# Compute the false discovery proportion (FDP) for association:
sum(Rej * !true.assoc) / sum(true.assoc)


## End(Not run)

## Not run:
#### Example 3: SNPs data; each SNP in each study has three association states,
####            negative, null, or positive:

# load the bins of the z-scores and their probabilities.
download.file('http://www.math.tau.ac.il/~ruheller/repfdr_RData/binned_zmat.RData',
  destfile = "binned_zmat.RData")
load(file = "binned_zmat.RData")
```

```
# can also be generated from SNPlocation - see ztobins documentation.

# load the prior probabilities for each association status vector.
data(Pi)
Pi # the proportions vector was computed using piem()
   # with the following command: Pi <- piem(pbz, bz)$last.iteration

# a) replicablity analysis:
output.rep <- repfdr(pbz, bz, "replication",Pi.previous.result=Pi)
BayesFdr.rep <- output.rep$mat[,"Fdr"]
Rej <- sum(BayesFdr.rep <= 0.05)
sum(Rej)

# The posterior probabilities for the first five features with Bayes FDR at most 0.05:
post <- ldr(pbz,bz[order(BayesFdr.rep)[1:5],],Pi)
round(post,4)

# posteriors for a subset of the association status vectors can also be reported:
H <- hconfigs( dim(bz)[2], 3)
h.replicability = apply(H, 1, function(y) {sum(y == 1)> 1 | sum(y == -1) >1})
post <- ldr(pbz,bz[order(BayesFdr.rep)[1:5],],Pi,h.vecs= which(h.replicability==1))
round(post,4)

# b) meta-analysis:
output.meta <- repfdr(pbz, bz, "meta-analysis", Pi.previous.result = Pi)
BayesFdr.meta <- output.meta$mat[,"Fdr"]
Rej <- sum(BayesFdr.meta <= 0.05)
sum(Rej)

## End(Not run)

## manhattan plot (ploting can take a while):
# code for manhattan plot by Stephen Turner (see copyrights at the source code manhattan.r)

## Not run:
  data(SNPlocations)
  par(mfrow=c(2,1))
  # Replication
  manhattan(dataframe=cbind(SNPlocations,P=BayesFdr.rep),ymax=10.5,pch=20,
            limitchromosomes=1:4,suggestiveline=-log(0.05,10),genomewideline=F,cex=0.25,
            annotate=SNPlocations$SNP[BayesFdr.rep<=0.05],main="Replication")
  # Association
  manhattan(dataframe=cbind(SNPlocations,P=BayesFdr.meta),ymax=10.5,cex=0.25,
            limitchromosomes=1:4,suggestiveline=-log(0.05,10),genomewideline=F,pch=20,
            annotate=SNPlocations$SNP[BayesFdr.rep<=0.05],main="Meta-analysis")
  par(mfrow=c(1,1))

## End(Not run)
```

SNPlocations *Three GWAS studies SNPs locations and data*

Description

SNPlocations includes the locations of SNPs in chromosomes 1 to 4. Data was simulated to the SNPs with HAPGEN2 for three studies and a sample of it was taken (Chromosomes 1 to 4) for the examples. The data is summarized as z-scores(transformed p-values, with inverse standard normal cumulative distribution). The z-scores matrix can be download from the web (see example).

Usage

```
data(SNPlocations)
```

Format

SNPlocations data.frame of 249024 SNPs' names, chromosome number and location on the chromosomes. zmat Matrix of 249024 SNPs' z-scores (in rows) in each of the 3 studies (columns).

Source

See: Su, Zhan, Jonathan Marchini, and Peter Donnelly. "HAPGEN2: simulation of multiple disease SNPs." Bioinformatics 27.16 (2011): 2304-2305.

Examples

```
data(SNPlocations)
head(SNPlocations)

## Not run:
download.file('http://www.math.tau.ac.il/~ruheller/repfdr_RData/zmat.RData',destfile = "zmat.RData")
load(file = "zmat.RData")

input.to.repfdr <- ztobins(zmat, 3, df= 15)
pbz <- input.to.repfdr$pdf.binned.z
bz  <- input.to.repfdr$binned.z.mat

## End(Not run)
```

---

twosided.PValues.tobins

Binning of two sided P-Values and estimation of the probabilities in each bin for the null and non-null states.

Description

For each study, the function discretizes two sided P-values into bins and estimates the probabilities in each bin for the null and non-null states.

The function can plot diagnostic plots (disabled by default) for model fit. These should be monitored for misfit of model to data, before using function output in repfdr. See description of diagnostic plots below.

Usage

```
twosided.PValues.tobins(pval.mat, n.bins = 120, type = 0, df = 7,
                                    central.prop = 0.5,
                                    pi0=NULL,plot.diagnostics = FALSE,
                                    trim.z=FALSE,trim.z.upper = 8,
                                    trim.z.lower = -8, force.bin.number = FALSE,
                                    pi.plugin.lambda = 0.05)
```

Arguments

`pval.mat`	Matrix of two sided P-Values of the features (in rows) in each study (columns).
`n.bins`	Number of bins in the discretization of the z-score axis (the number of bins is `n.bins - 1`). If the number of z-scores per study is small, we set `n.bins` to a number lower than the default of 120 (about equals to the square root of the number of z-scores). To override the bin number cap (and create a discretization of the data that is sparse), use the `force.bin.number = TRUE` argument.
`type`	Type of fitting used for f; 0 is a natural spline, 1 is a polynomial, in either case with degrees of freedom `df` (so total degrees of freedom including the intercept is `df+1`).
`df`	Degrees of freedom for fitting the estimated density f(z).
`central.prop`	Central proportion of the z-scores used like the area of zero-assumption to estimate pi0.
`pi0`	Sets argument for estimation of proportion of null hypotheses. Default value is NULL (automatic estimation of pi0) for every study. Second option is to supply vector of values between 0 and 1 (with length of the number of studies/ columns of `zmat`. These values will be used for pi0.
`plot.diagnostics`	If set to `TRUE`, will show disgnostics plots for density estimation for each study. First plot is a histogram of counts for each bin (Displayed as white bars), along with fitted density in green. Pink bars represent the observed number of counts in each bins, minus the expected number of null hypotheses by the model (truncated at zero). Red and Orange dashed lines represent the estimated densities for non null distributions fitted by the spline. A blue dashed line represents the density component of Z scores for null SNPS, N(0,1). A second plot is the Normal Q-Q plot of Zscores, converted using `qnorm` to the normal scale. A valid graph should coincide with a the linear fit displayed. A misfit with the linear plot could indicate either a null distribution which is not standard normal (a problem), or an extreme number of non null P-Values (Signal is not sparse, output is still valid). A black dashed line markes the expected fit for the standard normal distribution (with a single black dot for the (0,0) point). If the linear fit for the Q-Q plot (red line) does not match the dashed black line, the null distribution of the data is not standard normal. Misfit in these two plots should be investigated by the user, before using output in `repfdr` Default value is `False`.

`trim.z` If set to `TRUE`, Z scores above `trim.z.upper` or below `trim.z.lower` will be trimmed at their respective limits. Default value if `FALSE`

`trim.z.upper` Upper bound for trimming Z scores. Default value is 8

`trim.z.lower` Lower bound for trimming Z scores. Default value is -8

`force.bin.number`
Set to `T` to be able to create a discretization with `n.bins>sqrt(nrow(zmat))`.

`pi.plugin.lambda`
The function makes use of the plugin estimator for the estimation of the proportion of null hypotheses. The plugin estimator is `(sum(Pvalues > pi.plugin.lambda) + 1)/(m * (1-pi`
where `m` is the number of P-values. Default value is 0.05. This should be set to the type 1 error used for hypothesis testing.

Details

This utility function outputs the first two arguments to be input in the main function `repfdr`.

Value

A list with:

`pdf.binned.z` A 3-dimensional array which contains for each study (first dimension), the probabilities of a z-score to fall in the bin (second dimension), under each hypothesis status (third dimension). The third dimension can be of size 2 or 3, depending on the number of association states: if the association can be either null or only in one direction, the dimension is 2; if the association can be either null, or positive, or negative, the dimension is 3.

`binned.z.mat` A matrix of the bin numbers for each the z-scores (rows) in each study (columns).

`breaks.matrix` A matrix with `n.bins + 1` rows and ncol(zmat) columns, representing for each study the discretization chosed. Values are the between bin breaks. First and last values are the edges of the outmost bins.

`df` Number of degrees of freedom, used for spline fitting of density.

`proportions` Matrix with `n.association.status` rows, and `ncol(zmat)` columns, giving the estimated proportion of each component, for each study.

`PlotWarnings` Vector of size `ncol{zmat}`, keeping the warnings given for each study (available here, in the plots for each study and printed to console). With no warnings given for study, value is `NA`

See Also

`repfdr`

Examples

```
# we generate a dataset with p=10000 pvalues for two studies,
# p1=300 of which are non null:
set.seed(1)
p = 10000
```

```
p1 = 300
z1 = (rnorm(p))
z2 = (rnorm(p))
temp = rnorm(p1, 3.5,0.5)
z1[1:p1] = temp + rnorm(p1,0,0.2)
z2[1:p1] = temp + rnorm(p1,0,0.2)

zmat.example = cbind(z1,z2)
pmat.example = 1-(pnorm(abs(zmat.example)) - pnorm(-1*abs(zmat.example)))

twosided.pval.res = twosided.PValues.tobins(pmat.example,
                                            plot.diagnostics = TRUE)

twosided.pval.res$proportions
```

---

zmat_sim *Simulated data set*

Description

A simulated data set from three studies, with 10000 "features" in each study, each of which yielded a z-score. The data comprises 10000x3 z-scores. See hmat_sim for the indicators of association status matrix.

Usage

```
data(zmat_sim)
```

Format

zmat_sim is a matrix of 10000 z-scores (in rows) in each of the 3 studies (columns).

Examples

```
data(zmat_sim)
head(zmat_sim)

## Not run:
  input.to.repfdr <- ztobins(zmat_sim, 2 )
  pbz_sim1 <- input.to.repfdr$pdf.binned.z
  bz_sim1  <- input.to.repfdr$binned.z.mat

  data(binned_zmat_sim)
  stopifnot(all.equal(pbz_sim1,pbz_sim))
  stopifnot(all.equal(bz_sim1,bz_sim))

## End(Not run)

#### zmat_sim was generated by the following code:
```

```
data(hmat_sim)
set.seed(12)
m <- nrow(hmat_sim)
zmat_sim1  <- matrix(rnorm(n=3*m,mean=hmat_sim*3),nrow=m,ncol=3)
data(zmat_sim)
stopifnot(all.equal(zmat_sim1,zmat_sim))
```

`ztobins` *Binning of z-scores and estimation of the probabilities in each bin for the null and non-null states.*

Description

For each study, the function discretizes the z-scores into bins and estimates the probabilities in each bin for the null and non-null states.

The function can plot diagnostic plots (disabled by default) for model fit. These should be monitored for misfit of model to data, before using function output in `repfdr`. See description of diagnostic plots below.

Usage

```
ztobins(zmat, n.association.status = 3, n.bins = 120, type = 0, df = 7,
                    central.prop = 0.5,
                    pi0=NULL,plot.diagnostics = FALSE,
                    trim.z=FALSE,trim.z.upper = 8,trim.z.lower = -8,
                    force.bin.number = FALSE,
                    pi.using.plugin = FALSE, pi.plugin.lambda = 0.05)
```

Arguments

`zmat`	Matrix of z-scores of the features (in rows) in each study (columns).
`n.association.status`	either 2 for no-association\association or 3 for no-associtation\negative-association\positive-association.
`n.bins`	Number of bins in the discretization of the z-score axis (the number of bins is `n.bins - 1`). If the number of z-scores per study is small, we set `n.bins` to a number lower than the default of 120 (about equals to the square root of the number of z-scores). To override the bin number cap (and create a discretization of the data that is sparse), use the `force.bin.number = TRUE` argument.
`type`	Type of fitting used for f; 0 is a natural spline, 1 is a polynomial, in either case with degrees of freedom `df` (so total degrees of freedom including the intercept is `df+1`).
`df`	Degrees of freedom for fitting the estimated density f(z).
`central.prop`	Central proportion of the z-scores used like the area of zero-assumption to estimate pi0.

`pi0` Sets argument for estimation of proportion of null hypotheses. Default value is NULL (automatic estimation of pi0) for every study. Second option is to supply vector of values between 0 and 1 (with length of the number of studies/ columns of `zmat`. These values will be used for pi0.

`plot.diagnostics`
If set to `TRUE`, will show disgnostics plots for density estimation for each study. First plot is a histogram of counts for each bin (Displayed as white bars), along with fitted density in green. Pink bars represent the observed number of counts in each bins, minus the expected number of null hypotheses by the model (truncated at zero). Red and Orange dashed lines represent the estimated densities for non null distributions fitted by the spline. A blue dashed line represents the density component of Z scores for null SNPS, N(0,1).

A second plot is the Normal Q-Q plot of Zscores, converted using `qnorm` to the normal scale. A valid graph should coincide with a the linear fit displayed. A misfit with the linear plot could indicate either a null distribution which is not standard normal (a problem), or an extreme number of non null P-Values (Signal is not sparse, output is still valid). A black dashed line markes the expected fit for the standard normal distribution (with a single black dot for the (0,0) point). If the linear fit for the Q-Q plot (red line) does not match the dashed black line, the null distribution of the data is not standard normal.

Misfit in these two plots should be investigated by the user, before using output in `repfdr`

Default value is `False`.

`trim.z` If set to `TRUE`, Z scores above `trim.z.upper` or below `trim.z.lower` will be trimmed at their respective limits. Default value if `FALSE`

`trim.z.upper` Upper bound for trimming Z scores. Default value is 8

`trim.z.lower` Lower bound for trimming Z scores. Default value is -8

`force.bin.number`
Set to `T` to be able to create a discretization with `n.bins>sqrt(nrow(zmat))`.

`pi.using.plugin`
Logical flag indicating whether estimation of the number of null hypotheses should be done using the plugin estimator.(Default is `F`). The plugin estimator is `(sum(Pvalues > pi.plugin.lambda) + 1)/(m * (1-pi.plugin.lambda))` where `m` is the number of P-values.

`pi.plugin.lambda`
Parameter used for estimation of proportion of null hypotheses, for one sided tests. Default value is 0.05. This should be set to the type 1 error used for hypothesis testing.

Details

This utility function outputs the first two arguments to be input in the main function `repfdr`.

Value

A list with:

pdf.binned.z	A 3-dimensional array which contains for each study (first dimension), the probabilities of a z-score to fall in the bin (second dimension), under each hypothesis status (third dimension). The third dimension can be of size 2 or 3, depending on the number of association states: if the association can be either null or only in one direction, the dimension is 2; if the association can be either null, or positive, or negative, the dimension is 3.
binned.z.mat	A matrix of the bin numbers for each the z-scores (rows) in each study (columns).
breaks.matrix	A matrix with n.bins + 1 rows and ncol(zmat) columns, representing for each study the discretization chosed. Values are the between bin breaks. First and last values are the edges of the outmost bins.
df	Number of degrees of freedom, used for spline fitting of density.
proportions	Matrix with n.association.status rows, and ncol(zmat) columns, giving the estimated proportion of each component, for each study.
PlotWarnings	Vector of size ncol{zmat}, keeping the warnings given for each study (available here, in the plots for each study and printed to console). With no warnings given for study, value is NA

See Also

repfdr

Examples

```
# Simulated example using both the central proportion estimator
# and the plug in estimator for the proportion of null hypotheses:

set.seed(1)
p = 10000
p1 = 300
z1 = (rnorm(p))
z2 = (rnorm(p))
temp = rnorm(p1, 3.5,0.5)
z1[1:p1] = temp + rnorm(p1,0,0.2)
z2[1:p1] = temp + rnorm(p1,0,0.2)

z1.abs = abs(z1)
z2.abs = abs(z2)
plot(z1,z2)
hist(z1)
hist(z2)

zmat.example = cbind(z1,z2)

ztobins.res = ztobins(zmat.example,
                      plot.diagnostics = TRUE)
ztobins.res$proportions

ztobins.res.plugin.estimator = ztobins(zmat.example,
                           pi.using.plugin = TRUE,
```

```
                            plot.diagnostics = TRUE)

ztobins.res.plugin.estimator$proportions

## Not run:

# three association states case (H in {-1,0,1}):
download.file('http://www.math.tau.ac.il/~ruheller/repfdr_RData/zmat.RData',destfile = "zmat.RData")
load(file = "zmat.RData")

input.to.repfdr3 <- ztobins(zmat, 3, df = 15)
pbz    <- input.to.repfdr3$pdf.binned.z
bz     <- input.to.repfdr3$binned.z.mat

# two association states case (H in {0,1}):
data(zmat_sim)

input.to.repfdr <- ztobins(zmat_sim, 2, n.bins = 100 ,plot.diagnostics = T)
pbz_sim    <- input.to.repfdr$pdf.binned.z
bz_sim     <- input.to.repfdr$binned.z.mat

## End(Not run)
```

Index