# Package 'MatrixEpistasis'

November 23, 2017

**Type** Package

**Title** An ultrafast exhaustive epistasis scan for quantitative traits with covariate adjustment

**Version** 0.1.0

**Author** Shijia Zhu

**Maintainer** Shijia Zhu <shijia.zhu@mssm.edu>

**Description** MatrixEpistasis exhaustively scans all pairwise genetic interactions for quantitative traits. MatrixEpistasis works on both discrete genotype and continuous imputed genotype data. MatrixEpistasis can adjust covariates for epistasis detection, such as gender, age, and population structure. The excellent time-efficiency of MatrixEpistasis is achieved by the following innovations: 1) it expresses the intensive computation in terms of large matrix inner products (notably, no matrix inverse), avoiding separately calculating each epistasis model; 2) out of all regression coefficients (including two additive terms, one interaction term and multiple covariate terms), MatrixEpistasis only calculates the test statistic for the interaction term, largely alleviating the computational complexity; 3) the resulting test statistics from MatrixEpistasis are comparable, so that MatrixEpistasis can calculate p-values only for those exceeding the required significance level, therefore discarding a large number of incomplete beta or gamma functions.

**License** LGPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**NeedsCompilation** no

## R topics documented:

---

`as.dummy` *as.dummy*

---

## Description

as.dummy transforms a categorical variable to a dummy variable.

## Usage

```
as.dummy(x)
```

## Arguments

| | |
|---|---|
| `x` | a vector of categorical values |

## Value

a matrix of dummy variable, where each column represents a category.

## Examples

```
x <- rep( c('red','yellow','blue') , each=10 )
dx <- as.dummy(x)
```

---

`matrixEpistasis` *matrixEpistasis*

---

## Description

matrixEpistasis uses large matrix operation to perform the exhaustive epistasis scan for quantitative traits with covariate adjustment

## Usage

```
matrixEpistasis(snpA, snpB, trait, covariate = NULL)
```

## Arguments

| | |
|---|---|
| `snpA` | a matrix of numeric values in size of sample*snp representing the 1st group of SNPs, where the column names are the snp_ids |
| `snpB` | a matrix of numeric values in size of sample*snp representing the 2nd group of SNPs, where the column names are the snp_ids |
| `trait` | a vector of numeric values representing the quantitative trait |
| `covariate` | a matrix of numeric values in size of sample*covariate, by default, NULL |

## Value

A list containing the follow components:

- r a matrix of numeric values representing the partial correlation coefficients between snpA*snpB and traits conditioned on snpA, snpB and covariates
- df an integer value representing the degree of freedom

## Author(s)

Shijia Zhu, <shijia.zhu@mssm.edu>

## See Also

p2c; matrixPval; smoothQQplot

## Examples

```
# randomly generate a SNP matrix
snp <- sapply(1:100,function(i) rnorm(1000) )
# assign names to SNPs
colnames(snp) <- paste0('snp',1:100)
snpA = snp
snpB = snp

# radnomly generate a quantitative trait by simulating the relationship between SNPs and
trait <- snp %*% rnorm(100)

# use the top 5 PCs as the covariates
covariate <- prcomp(snp)$x[,1:5]

# run matrixEpistasis with covariates adjustment
res <- matrixEpistasis( snpA=snpA, snpB=snpB, trait=trait, covariate = covariate )
r <- res$r
df <- res$df

# run matrixEpistasis function with covariates adjustment
res <- matrixEpistasis( snpA=snpA, snpB=snpB, trait=trait, covariate = covariate )

# res is a list comprising two components: r and df. res$r is the matrix of partial corre
names(res)
r = res$r
df = res$df

# based on the degree of freedom, run matrixPval function to calculate p values for all p
p <- matrixPval( r , df )

# alternatively, users can calculate p-values only for those entries with p vlaues less t
# use p2c to covert p value threshold 1e-5 to the corresponding partial correlation coeff
corrThreshold <- p2c( pval=1e-2 , df )

# extract the index for those significant ones
index <- which( abs(r)>corrThreshold , arr.ind=TRUE )

# get the SNP names
snp1 <- colnames(snpA)[ index[,1] ]
snp2 <- colnames(snpB)[ index[,2] ]
```

```
# use matrixPval function to calculate p values for only those of interest
pvalue <- matrixPval( r[index] , df  )

# build the data frame
sig_res <- data.frame( snp1 , snp2 , pvalue )
head(sig_res)
```

---

`MatrixEpistasis_main` *matrixEpistasis*

---

## Description

matrixEpistasis uses large matrix operation to perform the exhaustive epistasis scan for quantitative traits with covariate adjustment

## Usage

```
MatrixEpistasis_main(snpA, snpB, trait, covariate = NULL,
  pvalThreshold = 1e-05, outputFileName)
```

## Arguments

| | |
|---|---|
| `snpA` | a matrix of numeric values in size of sample*snp representing the 1st group of SNPs, where the column names are the snp_ids |
| `snpB` | a matrix of numeric values in size of sample*snp representing the 2nd group of SNPs, where the column names are the snp_ids |
| `trait` | a vector of numeric values representing the quantitative trait |
| `covariate` | a matrix of numeric values in size of sample*covariate, by default, NULL |
| `pvalThreshold` | a numeric value representing the p-value threshold for matrixEpistasis output |
| `outputFileName` | a character value indicating the file name of matrixEpistasis output |

## Author(s)

Shijia Zhu, <shijia.zhu@mssm.edu>

## See Also

matrixEpistasis; matrixPval; p2c;

## Examples

```
# randomly generate a SNP matrix
snp <- sapply(1:100,function(i) rnorm(1000) )
# assign names to SNPs
colnames(snp) <- paste0('snp',1:100)
snpA = snp
snpB = snp
# radnomly generate a quantitative trait by simulating the relationship between SNPs and
trait <- snp %*% rnorm(100)
# use the top 5 PCs as the covariates
covariate <- prcomp(snp)$x[,1:5]

MatrixEpistasis_main( snpA=snpA, snpB=snpB, trait=trait, covariate=covariate, pvalThresho
```

---

`matrixPar` *matrixPar*

---

## Description

matrixPar calculates the partial correlation using the iterative formula

## Usage

```
matrixPar(r12, r13, r23)
```

## Arguments

| | |
|---|---|
| `r12` | a matrix of correlation coefficients between the 1st and 2nd groups of varaibles |
| `r13` | a matrix of correlation coefficients between the 1st and 3rd groups of varaibles |
| `r23` | a matrix of correlation coefficients between the 2nd and 3rd groups of varaibles |

## Value

a matrix of partial correlation coefficients between the 1st and 2nd groups of variables conditioned on the 3rd

---

`matrixPval` *matrixPval*

---

## Description

matrixPval calculates the p values for the correlation coefficients based on t-statistics

## Usage

```
matrixPval(r, df)
```

### Arguments

| | |
|---|---|
| r | a vector or matrix of correlation coefficients in [-1,+1] |
| df | the degree of freedom |

### Value

a vector or matrix of p values in [0,1]

### Examples

```
r <- cor(USArrests)
df <- nrow(USArrests) - 2
pval1 <- matrixPval(r,df)

pval2 <- matrix(ncol=ncol(USArrests),nrow=ncol(USArrests),data=0)
for(i in 1:ncol(USArrests))
{
   for(j in 1:ncol(USArrests))
   {
     pval2[i,j] <- cor.test(USArrests[,i],USArrests[,j])$p.val
   }
}

head(pval1)
head(pval2)
```

---

## matrixQtl *matrixQtl*

---

### Description

matrixQtl calculates the correlation between snp and quantitative trait

### Usage

```
matrixQtl(snp, trait, covariate = NULL)
```

### Arguments

| | |
|---|---|
| snp | a matrix of numeric values in size of sample*snp |
| trait | a matrix of numeric values in size of sample*trait |
| covariate | a matrix of numeric values in size of sample*covariate |

### Value

A list containing two components:

- r a matrix of Pearson correlation between snp and trait in size of snp*trait
- df the degree of freedom

## Author(s)

Shijia Zhu, <shijia.zhu@mssm.edu>

## See Also

matrixPval;

## Examples

```
snp <- sapply(1:100,function(i) rnorm(1000) )
alpha <- sapply(1:10,function(i) rnorm(100) )
trait <- snp %*% alpha
covariates <- prcomp(snp)$x[,1:5]

qtl <- matrixQtl( snp, trait , covariates )
pval1 <- matrixPval( r=qtl$r , df=qtl$df)

pval2 <- matrix(nrow=ncol(snp),ncol=ncol(trait),data=0)
for(i in 1:ncol(snp))
{
  for(j in 1:ncol(trait))
  {
    pval2[i,j] <- summary(lm(trait[,j]~snp[,i]+covariates))$coef[2,4]
  }
}


head(pval1)
head(pval2)
```

---

# n *n*

---

## Description

n standardizes a variable to have zero mean and unit standard deviation

## Usage

```
n(x)
```

## Arguments

| | |
|---|---|
| x | a vector of numeric values |

## Value

a vector of numeric values

## Examples

```
x <- rnorm(100,mean=10,sd=10)
nx <- n(x)
mean(nx)
sd(nx)
```

---

p2c *p2c*

---

## Description

p2c converts the given p value to the absolute value of corresponding correlation coefficient

## Usage

```
p2c(pval, df)
```

## Arguments

| | |
|---|---|
| pval | a numeric value in [0,1] representing the p value |
| df | an integer value representing the degree of freedom |

## Value

a numeric value in [0,1] representing the absolute value of correlation coefficient

## Author(s)

Shijia Zhu, <shijia.zhu@mssm.edu>

## Examples

```
x <- rnorm(100)
y <- x + rnorm(100)
z <- cor.test(x,y)
z
p2c(z$p.value,98)
```

`removeNAN` *removeNAN*

## Description

removeNAN removes the missing values from the data matrix.

## Usage

```
removeNAN(x, v = 0)
```

## Arguments

| | |
|---|---|
| x | a matrix of numeric values |
| v | the value assigned to the missing values, by default, 0 |

## Value

a matrix of numeric values

## Examples

```
x <- matrix(c(0,1,NA,2),ncol=2)
x
removeNAN(x)
```

`smoothQQplot` *smoothQQplot*

## Description

smoothQQplot draws the QQplot between two groups of p values together with the smoothed density representation of the scatterplot

## Usage

```
smoothQQplot(x, y)
```

## Arguments

| | |
|---|---|
| x | a vector of numeric values in [0,1] representing the first group of p values |
| y | a vector of numeric values in [0,1] representing the second group of p values |

## Examples

```
x <- runif(10000)
y <- runif(10000)
smoothQQplot(x,y)
```

# Index