Package 'GMRP'

November 17, 2026

Type Package

Title GWAS-based Mendelian Randomization and Path Analyses

Version 1.41.0

Date 2018-04-15

Author Yuan-De Tan

Maintainer Yuan-De Tan <tanyuande@gmail.com>

Description Perform Mendelian randomization analysis of multiple SNPs to determine risk factors causing disease of study and to exclude confounding variabels and perform path analysis to construct path of risk factors to the disease.

License GPL (>= 2)

Depends R(>= 3.3.0),stats,utils,graphics, grDevices, diagram, plotrix, base,GenomicRanges

Suggests BiocStyle, BiocGenerics

LazyLoad yes

biocViews Sequencing, Regression, SNP

NeedsCompilation no

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

git_branch devel

git_last_commit a31ee76

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-11-16

Contents

GMRP-package 2
beta.data 4
cad.data 5
chrp 6
fmerge 7
lpd.data 8
mktable 10
path 13

pathdiagram 15
pathdiagram2 16
SNP358.data 17
SNP368annot.data 18
snpPositAnnot 19
ucscannot 20

Index **22**

GMRP-package *GWAS-based Mendelian Randomization Path Analysis*

Description

GMRP is used to perform Mendelian randomization analysis of causal variables on disease of study using SNP beta data from **GWAS** or **GWAS** meta analysis and furthermore execute path analysis of causal variables onto the disease.

Details

GMRP can perform analyses of Mendelian randomization (MR),correlation, path of causal variables onto disease of interest and SNP annotation summarization analysis. MR includes SNP selection with given criteria and regression analysis of causal variables on the disease to generate beta values of causal variables on the disease. Using the beta values, **GMRP** performs correlation and path analyses to construct path diagrams of causal variables to the disease. **GMRP** consists of 8 functions: *chrp, fmerge, mktable, pathdiagram2, pathdiagram, path, snpPositAnnot, ucscannot* and 5 datasets: `beta.data`, `cad.data`, `lpd.data`, `SNP358.data` and `SNP368_annot.data`. Function *chrp* is used to separate string vector hg19 into two numeric vectors: chromosome number and SNP position on chromosomes. Function *fmerge* is used to merge two datasets into one dataset. Function *mktable* performs SNP selection and creates a standard beta table for function path to do path analyses. Function *pathdiagram* is used to create a path diagram of causal variables onto the disease or onto outcome. Function *pathdiagram2* can merge two-level pathdiagrams into one nested pathdiagram where inner path diagram is a path diagram of causal variables contributing onto outcome and the outside path diagram is a diagram of path of causal variables including outcome onto the disease. The five datasets provide examples for running these functions. `lpd.data` and `cad.data` provide an example to create a standard beta dataset for *path* function to do path analysis and SNP data for SNP annotation analysis by performing *mktable* and *fmerge*. beta.data are a standard beta dataset for path analysis. `SNP358.data` provide an example for function *snpPositAnnot* to do SNP position annotation analysis and `SNP368_annot.data` are for function *ucscannot* to do SNP function annotation analysis. It is specially emphasized that except for that making standard beta table using *mktable* must be done in Unix/Linux system,GMRP can be performed in Windows or Mac OS. This is because GWAS datasets usually are very huge but standard beta table is small. If users'Unix/Linux system has X11 or the other graphics system, then user should perform GMRP in Unix/Linux system, otherwise, user should transfer a standard beta table to a local computer and run GMRP in it.

Author(s)

Yuan-De Tan <tanyuande@gmail.com>
Maintainer: Yuan-De Tan

References

Do, R. et al. 2013. Common variants associated with plasma triglycerides and risk for coronary artery disease. Nat Genet 45: 1345-1352.
Sheehan, N.A. et al. 2008. Mendelian randomisation and causal inference in observational epidemiology. PLoS Med 5: e177.
Sheehan, N.A.,et al. 2010. Mendelian randomisation: a tool for assessing causality in observational epidemiology. Methods Mol Biol 713: 153-166.\Wright, S. 1921. Correlation and causation. J. Agricultural Research 20: 557-585.
Wright, S. 1934. The method of path coefficients Annals of Mathematical Statistics 5 (3): 161-215 5: 161-215.

See Also

path, mktable, pathdiagram,pathdiagram2, plotmat, plotweb

Examples

```
data(beta.data)
mybeta<-DataFrame(beta.data)
CAD<-beta.data$cad
LDL<-beta.data$ldl
HDL<-beta.data$hdl
TG<-beta.data$tg
TC<-beta.data$tc
#par(mfrow=c(2,2))
plot(LDL,CAD,pch=19,col="blue",xlab="beta of SNPs on LDL",ylab="beta of SNP on CAD",
 main="A",cex.lab=1.5,cex.axis=1.5,cex.main=2)
abline(lm(CAD~LDL),col="red",lwd=2)
plot(HDL,CAD,pch=19,col="darkgreen",xlab="beta of SNPs on HDL",ylab="beta of SNP on
 CAD", main="B",cex.lab=1.5,cex.axis=1.5,cex.main=2)
abline(lm(CAD~HDL),col="red",lwd=2)
plot(TG,CAD,pch=19,col=colors()[96],xlab="beta of SNPs on TG",ylab="beta of SNP on
 CAD",main="C",cex.lab=1.5,cex.axis=1.5,cex.main=2)
abline(lm(CAD~TG),col="red",lwd=2)
plot(TC,CAD,pch=19,col=colors()[123],xlab="beta of SNPs on TC",ylab="beta of SNP on
 CAD",main="D",cex.lab=1.5,cex.axis=1.5,cex.main=2)
abline(lm(CAD~TC),col="red",lwd=2)

mod<-cad~ldl+hdl+tg+tc
pathvalue<-path(betav=mybeta,model=mod,outcome="cad")

mypath<-matrix(NA,3,4)
mypath[1,]<-c(1.000000,-0.066678, 0.420036,0.764638)
mypath[2,]<-c(-0.066678,1.000000,-0.559718,0.496831)
mypath[3,]<-c(0.420036,-0.559718,1.000000,0.414346)
colnames(mypath)<-c("ldl","hdl","tg","path")
mypath<-DataFrame(mypath)
#mypath
#DataFrame with 3 rows and 4 columns
#        ldl       hdl        tg      path
#  <numeric> <numeric> <numeric> <numeric>
#1  1.000000 -0.066678  0.420036  0.764638
#2 -0.066678  1.000000 -0.559718  0.496831
#3  0.420036 -0.559718  1.000000  0.414346

#> pathdiagram(pathdata=mypath,disease="cad",R2=0.988243,range=c(1:3))
```

```
#Loading required package: shape
#Error in pathcad$path : $ operator is invalid for atomic vectors
mypath<-as.data.frame(mypath)
pathdiagram(pathdata=mypath,disease="cad",R2=0.988243,range=c(1:3))
```

`beta.data` *Beta Data Of SNP Regressed on Causal Variables and Disease*

Description

Beta data are a matrix dataset consisting of 5 columns: cad, ldl, hdl, tg, and tc with 368 rows.

Usage

```
data("beta.data")
```

Format

A data frame with 368 observations on the following 5 variables.

`cad` a numeric vector

`ldl` a numeric vector

`hdl` a numeric vector

`tg` a numeric vector

`tc` a numeric vector

Details

Beta data are a matrix consisting of regression coefficients of 368 SNPs on cad, ldl, hdl, tg, tc where `cad` is coronary artery disease, `ldl` is low-density lipoprotein cholesterol, `hdl`, high-density lipoprotein cholesterol, `tg`,triglycerides and `tc`, total cholesterol in plasma. These 368 SNPs were obtained by using `mktable` from **GWAS** meta-analyzed data.

Value

A set of real regression coefficients of 368 SNPs on disease and causal variables.

Source

http://csg.sph.umich.edu//abecasis/public/lipids2013/

\http://www.cardiogramplusc4d.org/downloads/

References

Willer CJ et al. Discovery and refinement of loci associated with lipid levels. Nat. Genet. 2013. doi:10.1038/ng.2797.

\Schunkert, H. et al. 2011. Large-scale association analysis identifies 13 new susceptibility loci for coronary artery disease. Nat Genet 43: 333-338.[online]

\Schunkert H, Konig IR, Kathiresan S, Reilly MP, Assimes TL, Holm H et al. Large-scale association analysis identifies 13 new susceptibility loci for coronary artery disease. Nat Genet. 2011 43: 333-338.

Examples

```
data(beta.data)
## maybe str(beta.data) ; plot(beta.data) ...
```

---

cad.data *boldGWAS Meta-analyzed Data of Coronary Artery Disease*

Description

cad.data are a matrix dataset consisting of 12 variables such as SNPID, SNP position on chromosomes, allele and alternative allele, allelic frequencies and 1069 SNPs.

Usage

```
data("cad.data")
```

Format

A data frame with 1609 observations on the following 12 variables.

SNP a character vector

chr_pos_b36 a character vector

reference_allele a character vector

other_allele a character vector

ref_allele_frequency a numeric vector

pvalue a numeric vector

het_pvalue a numeric vector

log_odds a numeric vector

log_odds_se a numeric vector

N_case a numeric vector

N_control a numeric vector

model a character vector

Details

cad.data, also called CARDIoGRAM **GWAS**, are a meta-analyzed GWAS data from **GWAS** studies of European descent imputed to **HapMap2** involving 22,233 cases and 64,762 controls. The data were downloaded from the following website.

Value

A data sheet consisting of 1609 rows(SNPs) and 12 columns(character vectors such as SNPID and allele, numeric vector such as allele frequency and beta coefficient. See data format above).

Source

http://www.cardiogramplusc4d.org/downloads/

References

Schunkert H, Konig IR, Kathiresan S, Reilly MP, Assimes TL, Holm H et al. Large-scale association analysis identifies 13 new susceptibility loci for coronary artery disease. **Nat Genet.** 2011 **43**: 333-338.

Examples

```
data(cad.data)
## maybe str(cad.data) ; plot(cad.data) ...
```

`chrp` *Split hg19*

Description

Split string hg19 into two numeric columns: chr and posit.

Usage

```
chrp(hg)
```

Arguments

hg	character vector represented with codechr##.####### where chr## is chromosome number and .####### is SNP site (bp).

Details

chrp can convert chr##.####### into two numeric columns: chr(chromosome number) and posit(SNP position)

Value

Return two numeric vectors: chrosomome number and SNP position

Note

If there is chrX.######### in the data sheet, then user should change chrX.######## to chr23.#######.

Note

hg may also be hg18. User can also use packages GenomicRanges to retrieve chromosome # and SNP position.

Author(s)

Yuan-De Tan <tanyuande@gmail.com>

\Dajiang Liu

See Also

mktable,link{GenomicRanges}[Granges],link{GenomicRanges}[IRangs],link{GenomicRanges}[DataFrame]

Examples

```
data(lpd.data)
lpd<-lpd.data
hg19<-lpd$SNP_hg19.HDL
chr<-chrp(hg=hg19)
```

`fmerge` *Merge Two **GWAS** Result Data Sheets*

Description

fmerg can be used to merge two **GWAS** result data sheets with the same key `ID(SNP ID)` into one data sheet.

Usage

```
fmerge(fl1, fl2,ID1, ID2, A, B, method)
```

Arguments

`fl1`	R object: data file 1
`fl2`	R object: data file 2
`ID1`	key id (SNP ID such as rsid) in file 1
`ID2`	key id (SNP ID such as rsid) in file 2
`A`	postfix for file 1: A=".W1". W1 may be any identifier in file 1. Default is A="".
`B`	postfix for file 2: B=".W2". W2 may be any identifier in file2. Default is B="".
`method`	method for merging. See details.

Details

fl1 and fl2 are two **GWAS** result data files from different studies or with different risk variables. They contain `SNPID`, `hg18`, `hg19`(positions), beta values, allele, frequency, and so on. The method has four options: method="No","NO" or "no" means that all data with unmatch `SNPs` are not saved in the merged file; method="All","ALL" or "all" lets fmerge save all the data with unmatched `SNPs` from two files but they are not paired one-by-one. This is different from R *merge* function. method="file1" will save the data with unmatched `SNPs` only from file 1 in the merged file and method="file2" allows function *fmerge* to save the data with unmatched `SNPs` from file2 in the merged file.

Value

Return a joined data sheet.

Note

Function fmerg can also be applied to the other types of data.

Author(s)

Yuan-De Tan <tanyuande@gmail.com>

See Also

merge

Examples

```
data1<-matrix(NA,20,4)
data2<-matrix(NA,30,7)
SNPID1<-paste("rs",seq(1:20),sep="")
SNPID2<-paste("rs",seq(1:30),sep="")
data1[,1:4]<-c(round(runif(20),4),round(runif(20),4),round(runif(20),4),round(runif(20),4))
data2[,1:4]<-c(round(runif(30),4),round(runif(30),4),round(runif(30),4),round(runif(30),4))
data2[,5:7]<-c(round(seq(30)*runif(30),4),round(seq(30)*runif(30),4),seq(30))
data1<-cbind(SNPID1,as.data.frame(data1))
data2<-cbind(SNPID2,as.data.frame(data2))
dim(data1)
dim(data2)
colnames(data1)<-c("SNP","var1","var2","var3","var4")
colnames(data2)<-c("SNP","var1","var2","var3","var4","V1","V2","V3")
data12<-fmerge(fl1=data1,fl2=data2,ID1="SNP",ID2="SNP",A=".dat1",B=".dat2",method="No")
#data12[1:3,]
#  SNP.dat1 var1.dat1 var2.dat1 var3.dat1 var4.dat1 SNP.dat2 var1.dat2 var2.dat2
#1      rs1    0.9152    0.9853    0.9879    0.9677      rs1    0.5041    0.5734
#2     rs10    0.3357     0.116    0.3408    0.1867     rs10    0.9147    0.9294
#3     rs11    0.8004    0.8856    0.2236    0.4642     rs11    0.9262    0.5831
#  var3.dat2 var4.dat2     V1     V2 V3
#1    0.4933    0.6766 0.1864 0.6836  1
#2    0.4104    0.1327 3.2192 1.4166 10
#3    0.8541    0.6228 1.1803 1.9044 11
```

---

lpd.data ***GWAS** Meta-analyzed Data of Lipoprotein Cholesterols*

Description

lpd.data are standard **GWAS** Meta-analyzed dataset of lipoprotein cholesterols. It was constructed by merging four datasets: `Mc_HDL.txt`,`Mc_LDL.txt`, `Mc_TC.txt` and `Mc_TG.txt`.

Usage

```
data("lpd.data")
```

Format

A data frame with 1609 observations on the following 40 variables.

`SNP_hg18.HDL` a character vector

`SNP_hg19.HDL` a character vector

`rsid.HDL` a character vector

`A1.HDL` a character vector

`A2.HDL` a character vector

`beta.HDL` a numeric vector

`se.HDL` a numeric vector

`N.HDL` a numeric vector

`P.value.HDL` a numeric vector

`Freq.A1.1000G.EUR.HDL` a numeric vector

`SNP_hg18.LDL` a character vector

`SNP_hg19.LDL` a character vector

`rsid.LDL` a character vector

`A1.LDL` a character vector

`A2.LDL` a character vector

`beta.LDL` a numeric vector

`se.LDL` a numeric vector

`N.LDL` a numeric vector

`P.value.LDL` a numeric vector

`Freq.A1.1000G.EUR.LDL` a numeric vector

`SNP_hg18.TG` a character vector

`SNP_hg19.TG` a character vector

`rsid.TG` a character vector

`A1.TG` a character vector

`A2.TG` a character vector

`beta.TG` a numeric vector

`se.TG` a numeric vector

`N.TG` a numeric vector

`P.value.TG` a numeric vector

`Freq.A1.1000G.EUR.TG` a numeric vector

`SNP_hg18.TC` a character vector

`SNP_hg19.TC` a character vector

`rsid.TC` a character vector

`A1.TC` a character vector

`A2.TC` a character vector

`beta.TC` a numeric vector

`se.TC` a numeric vector

`N.TC` a numeric vector

`P.value.TC` a numeric vector

`Freq.A1.1000G.EUR.TC` a numeric vector

Details

These four **GWAS** Meta-analyzed data of lipoprotein cholesterols were downloaded from the following website.

Value

A data sheet consisting of 1609 rows (SNPs) and 40 columns(character vectors such as SNPID and allele, numeric vector such as allele frequency, beta coefficient. See data format above).

Source

http://csg.sph.umich.edu//abecasis/public/lipids2013/

References

Willer CJ et al. Discovery and refinement of loci associated with lipid levels. *Nat. Genet.* 2013. doi:10.1038/ng.2797.

Examples

```
data(lpd.data)
## maybe str(lpd.data) ; plot(lpd.data) ...
```

`mktable` *Selection of SNPs and Creation of A Standard Table for Mendelian Randomization and Path Analyses*

Description

mktable is used to choose SNPs with `LG`, `Pv`, `Pc` and `Pd` and create a standard `SNP` beta table for `Mendelian randomization` and `path analysis`, see details.

Usage

```
mktable(cdata, ddata,rt, varname, LG, Pv, Pc, Pd)
```

Arguments

`cdata`	causal variable **GWAS** data or **GWAS** meta-analysed data containing `SNP ID`, `SNP` position, chromosome, allele, allelic frequency, beta value, `sd`, sample size, etc.
`ddata`	disease **GWAS** data or **GWAS** meta-analysed data containing `SNP ID`, `SNP` position, chromosome, allele, allelic frequency, beta value, `sd`, sample size, etc.
`rt`	a string that specifies type of returning table. It has two options: `rt="beta"` returns beta table or `rt="path"` returns `SNP` direct path coefficient table. Default is "beta".
`varname`	a required string set that lists names of undefined causal variables for Mendelian randomization and path analyses. The first name is disease name. Here an example given is `varname <-c("CAD","LDL","HD","TG","TC")`.
`LG`	a numeric parameter. `LG` is a given minimum interval distnce between `SNP`s and used to choose `SNP`s with. Default `LG=1`
`Pv`	a numeric parameter. `Pv` is a given maximum p-value that is used to choose `SNP`s. Default Pv=5e-8
`Pc`	a numeric parameter. `Pc` is a given proportion of sample size to maximum sample size in causal variable data and used to choose `SNP`s. Default `Pc=0.979`
`Pd`	a numeric parameter. Pd is a given proportion of sample size to the maximum sample size in disease data and used to choose `SNP`s. Default `Pd =0.979`.

Details

The standard **GWAS** cdata set should have the format with following columns: chrn, posit, rsid, a1.x1, a1.x2, ..., a1.xn, freq.x1, freq.x2, ..., freq.xn, beta.x1, beta.x2, ..., beta.xn, sd.x1, sd.x2, ..., sd.xn, pvj, N.x1, N.x2, ..., N.xn, pcj. The standard **GWAS** *ddata* set should havehg.d, SNP.d,a1.d, freq.d, beta.d, N.case,N.ctr,freq.case where x1, x2, ..., xn are causal variables. See example.

beta is a numeric vector that is a column of beta values for regression of SNPs on variable vector X={x1, x2, ..., xn}.

freq is a numeric vector that is a column of frequencies of allele 1 with respect to variable vector X={x1, x2, ..., xn}.

sd is a numeric vector that is a column of standard deviations of variable x1,x2, ..., xn specific to SNP. Note that here sd is not beta standard deviation. If sd is not specifical to SNPs, then sd.xi has the same value for all SNPs in variable i.

d denotes disease.

N is sample size.

freq.case is frequency of disease.

chrn is a numeric vector for chromosome #.

posit is a numeric vector for SNP positions on chromosome #. Some time, chrn and posit are combined into string vector: hg19/hg18.

pvj is defined as p-value, pcj and pdj as proportions of sample size for SNP j to the maximum sample size in the causal variable data and in disease data, respectively.

Value

Return a standard SNP beta or SNP path table containing m SNPs chosen with LG, Pv, Pc and Pd and n variables and disease for Mendelian randomization and path analysis.

Note

The order of column variables must be chrn posit rsid a1.x1 ...a1.xn freq.x1 ...freq.xn beta.x1 ...beta.x1 ...beta.xn sd.x1 ...sd.xn ...otherwise, mktable would have error. see example.

Author(s)

Yuan-De Tan <tanyuande@gmail.com>

References

Do, R. et al. 2013. Common variants associated with plasma triglycerides and risk for coronary artery disease. *Nat Genet* **45**: 1345-1352.
Sheehan, N.A. et al. 2008. Mendelian randomisation and causal inference in observational epidemiology. *PLoS Med* **5**: e177.
Sheehan, N.A.,et al. 2010. Mendelian randomisation: a tool for assessing causality in observational epidemiology. *Methods Mol Biol* **713**: 153-166.
Willer, C.J. Schmidt, E.M. Sengupta, S. Peloso, G.M. Gustafsson, S. Kanoni, S. Ganna, A. Chen, J.,Buchkovich, M.L. Mora, S. et al (2013) Discovery and refinement of loci associated with lipid levels. *Nat Genet* **45**: 1274-1283.

See Also

path

Examples

```
data(lpd.data)
#lpd<-DataFrame(lpd.data)
lpd<-lpd.data
data(cad.data)
#cad<-DataFrame(cad.data)
cad<-cad.data
# step 1: calculate pvj
pvalue.LDL<-lpd$P.value.LDL
pvalue.HDL<-lpd$P.value.HDL
pvalue.TG<-lpd$P.value.TG
pvalue.TC<-lpd$P.value.TC
pv<-cbind(pvalue.LDL,pvalue.HDL,pvalue.TG,pvalue.TC)
pvj<-apply(pv,1,min)

#step 2: construct beta table of undefined causal variables:
beta.LDL<-lpd$beta.LDL
beta.HDL<-lpd$beta.HDL
beta.TG<-lpd$beta.TG
beta.TC<-lpd$beta.TC
beta<-cbind(beta.LDL,beta.HDL,beta.TG,beta.TC)

#step 3: construct a matrix for allele 1 in each undefined causal variable:
a1.LDL<-lpd$A1.LDL
a1.HDL<-lpd$A1.HDL
a1.TG<-lpd$A1.TG
a1.TC<-lpd$A1.TC
alle1<-cbind(a1.LDL,a1.HDL,a1.TG,a1.TC)

#step 4: calculate sample sizes of causal variables and calculate pcj
N.LDL<-lpd$N.LDL
N.HDL<-lpd$N.HDL
N.TG<-lpd$N.TG
N.TC<-lpd$N.TC
ss<-cbind(N.LDL,N.HDL,N.TG,N.TC)
sm<-apply(ss,1,sum)
pcj<-sm/max(sm)

#step 5: construct a matrix for frequency of allele1 in each undefined causal variable in 1000G.EUR
freq.LDL<-lpd$Freq.A1.1000G.EUR.LDL
freq.HDL<-lpd$Freq.A1.1000G.EUR.HDL
freq.TG<-lpd$Freq.A1.1000G.EUR.TG
freq.TC<-lpd$Freq.A1.1000G.EUR.TC
freq<-cbind(freq.LDL,freq.HDL,freq.TG,freq.TC)

#step 6: construct matrix for sd of each causal variable (here sd is not specific to SNPj)
# the sd values were averaged over 63 studies see reference Willer et al(2013)
sd.LDL<-rep(37.42,length(pvj))
sd.HDL<-rep(14.87,length(pvj))
sd.TG<-rep(92.73,length(pvj))
sd.TC<-rep(42.74,length(pvj))
sd<-cbind(sd.LDL,sd.HDL,sd.TG,sd.TC)
```

```
#step 7: retriev SNP ID and position:
hg19<-lpd$SNP_hg19.HDL
rsid<-lpd$rsid.HDL

#step 8: invoke chrp to separate chromosome number and SNP position:
chr<-chrp(hg=hg19)

#step 9: get new data of causal variables:
newdata<-cbind(freq,beta,sd,pvj,ss,pcj)
newdata<-cbind(chr,rsid,alle1,as.data.frame(newdata))
dim(newdata)
#[1] 120165     25

#step 10: retrieve cad data from cad and calculate pdj and frequency of cad in population
hg18.d<-cad$chr_pos_b36
SNP.d<-cad$SNP #SNPID
a1.d<-tolower(cad$reference_allele)
freq.d<-cad$ref_allele_frequency
pvalue.d<-cad$pvalue
beta.d<-cad$log_odds
N.case<-cad$N_case
N.ctr<-cad$N_control
N.d<-N.case+N.ctr
freq.case<-N.case/N.d


#step 11: get new cad data:
newcad<-cbind(freq.d,beta.d,N.case,N.ctr,freq.case)
newcad<-cbind(hg18.d,SNP.d,a1.d,as.data.frame(newcad))
dim(newcad)

#step 12: give variable list
varname<-c("CAD","LDL","HDL","TG","TC")
#step 3: create beta table with function mktable
mybeta<-mktable(cdata=newdata,ddata=newcad,rt="beta",varname=varname,LG=1, Pv=0.00000005,
Pc=0.979,Pd=0.979)

beta<-mybeta[,4:8] # save beta for path analysis
snp<-mybeta[,1:3] # save snp for annotation analysis
beta<-DataFrame(beta)
```

---

path *Path Analysis*

Description

path is used to perform path analysis of multiple causal or risk variables on an outcome or disease of study.

Usage

```
path(betav,model,outcome)
```

Arguments

betav	a matrix numeric data with p rows and q columns in which the first column must be outcome and other columns are risk variables.
model	a linear model for multivariate linear regression analysis. The model must be given in R Console. For example, mymodel<-CAD~LDL+HDL+TG+TC.
outcome	a string that is required to give outcome name or disease name. For example, outcome="CAD".

Details

path is originally planed to perform causal analysis of risk variables on disease of study based on the results of the Mendelian randomization analysis of SNPs on these risk variables and disease. In the **GMRP** package, the betav is a matrix of beta coefficients of linear regression analyses of chosen SNPs on the risk (or causal) variables and disease or outcome. The beta values are equivalently quantitative values, so this path function can also be used to analyze direct and indirect contributions of quantitative traits to economic traits.

Value

Return three matrices: beta coefficients of regressions of risk variables on outcome or disease, correlation matrix and path matrix and also return director path coefficients and R-square.

Note

betav may also be a matrix of SNP path coefficients onto risk variables and disease.

Author(s)

Yuan-De Tan <tanyuande@gmail.com>

References

Do, R. et al. 2013 Common variants associated with plasma triglycerides and risk for coronary artery disease. *Nat Genet* **45**: 1345-1352.
Sheehan, N.A. et al. 2008 Mendelian randomisation and causal inference in observational epidemiology. PLoS Med 5: e177.
Sheehan, N.A.,et al. 2010 Mendelian randomisation: a tool for assessing causality in observational epidemiology. *Methods Mol Biol* **713**: 153-166.
Wright, S. 1921 Correlation and causation. *J.Agricultural Research* **20**: 557-585.
Wright, S. 1934 The method of path coefficients. *Annals of Mathematical Statistics* **5** (3): 161-215.

See Also

link[base]{lm},link[stats]{cor}

Examples

```
data(beta.data)
mybeta<-DataFrame(beta.data)
#mybeta<-as.data.frame(beta.data)
mod<-cad~ldl+hdl+tg+tc
pathvalue<-path(betav=mybeta,model=mod,outcome="cad")
```

pathdiagram *Path Diagram*

Description

Create a directed acyclic diagram to represent causal effects of risk factors on the disease of study.

Usage

```
pathdiagram(pathdata, disease, R2, range)
```

Arguments

pathdata	R object that is dataset consisting of correlation matrix of risk factors and a numeric vector of direct path coefficients.
disease	a string that specifies outcome name or disease name. If disease name is long or has multiple words, then we suggest an abbreviated name, for example, coronary artery disease may be shortened as CAD.
R2	a numeric parameter, is R-square obtained from path analysis.
range	range of specified columns for correlation matrix. For example, range = c(2:4) means the correlation coefficient begins with column 2 and end at column 4.

Details

The *pathdata* contains correlation matrix of risk factors and a numeric vector of direct path coefficients obtained from path analysis of beta data of SNPs on risk factors and disease. Columns must have risk factor names and path.

Value

NULL. pathdiagram will create one-level path diagram labeled with colors.

Author(s)

Yuan-De Tan <tanyuande@gmail.com>

See Also

pathdiagram2, plotmat, plotweb

Examples

```
mypath<-matrix(NA,3,4)
mypath[1,]<-c(1.000000,-0.066678, 0.420036,0.764638)
mypath[2,]<-c(-0.066678,1.000000,-0.559718,0.496831)
mypath[3,]<-c(0.420036,-0.559718,1.000000,0.414346)
colnames(mypath)<-c("ldl","hdl","tg","path")

#mypath
#        ldl       hdl        tg      path
#1  1.000000 -0.066678  0.420036  0.764638
#2 -0.066678  1.000000 -0.559718  0.496831
```

```
#3  0.420036 -0.559718  1.000000  0.414346

mypath<-as.data.frame(mypath)
pathdiagram(pathdata=mypath,disease="cad",R2=0.988243,range=c(1:3))
```

---

pathdiagram2 *Two-level Nested Pathdiagram*

Description

This function is used to create two-level nested pathdiagram to represent causal effects of risk factors on outcome and on the disease of study. The nested path is a child path, which is related to outcome and the outside path is parent path with respect to disease.

Usage

```
pathdiagram2(pathD, pathO, rangeD, rangeO, disease, R2D, R2O)
```

Arguments

pathD	R object that is disease path result data consisting of correlation matrix of causal variables to be identified in Mendelian randomization analysis and path coefficient vector of these variables directly causing the disease of study.
pathO	R object that is outcome path result data consisting of correlation matrix of causal variables and path coefficient vector of these variables directly contributing to outcome. This outcome variable may be or not be one of risk factors or causal variables in disease path data. These variables are the same with those in *pathD*.
rangeD	specifies column range for correlation coefficient matrix in *pathD*, for example, `rangeD=c(2:4)` means the correlation coefficient begins with column 2 and end at column 4.
rangeO	specifies column range for correlation coefficient matrix in *pathO*, see example in `rangeD`.
disease	a string that specifies disease name. If disease name is long or has multiple words, then we suggest an abbreviated name, for example, "coronary artery disease" can be shortened as `CAD`.
R2D	a required numeric parameter and its value is R-square obtained from path analysis of disease data.
R2O	a required numeric parameter and its value is R-square obtained from path analysis of outcome data.

Details

Two path datasets must contain correlation matrix of variables detected to be risk factor of disease and a vector of direct path coefficients obtained from path analysis of beta data of `SNPs` on causal variables and disease. Columns must have shortened variable names and path word (see examples). *pathdiagram2* requires two path data have the same causal variable names and the same name order. The outcome in the outcome path data must be the last variable in the correlation matrix in disease path data (see examples). Otherwise, *pathdiagram2* would give an error message.

Value

Null. Function *pathdiagram2* creates a nested two-level path diagram labeled with color.

Author(s)

Yuan-De Tan <tanyuande@gmail.com>

See Also

pathdiagram, plotmat, plotweb

Examples

```
mypath<-matrix(NA,3,4)
mypath[1,]<-c(1.000000,-0.066678, 0.420036,0.764638)
mypath[2,]<-c(-0.066678,1.000000,-0.559718,0.496831)
mypath[3,]<-c(0.420036,-0.559718,1.000000,0.414346)
colnames(mypath)<-c("ldl","hdl","tg","path")
mypath<-DataFrame(mypath)
#mypath
#DataFrame with 3 rows and 4 columns
#        ldl       hdl        tg      path
#  <numeric> <numeric> <numeric> <numeric>
#1  1.000000 -0.066678  0.420036  0.764638
#2 -0.066678  1.000000 -0.559718  0.496831
#3  0.420036 -0.559718  1.000000  0.414346

#In this pathdiagram, the outcome is TC
pathD<-matrix(NA,4,5)
pathD[1,]<-c(1, -0.070161,0.399038,0.907127,1.210474)
pathD[2,]<-c(-0.070161, 1,-0.552106,0.212201,0.147933)
pathD[3,]<-c(0.399038,-0.552106,1,0.44100,0.64229)
pathD[4,]<-c(0.907127 ,0.212201,0.441007,1,-1.035677)
colnames(pathD)<-c("ldl","hdl","tg","tc","path")

#pathD
#        LDL       HDL        TG        TC      path
#1  1.000000 -0.070161  0.399038  0.907127  1.210474
#2 -0.070161  1.000000 -0.552106  0.212201  0.147933
#3  0.399038 -0.552106  1.000000  0.441000  0.642290
#4  0.907127  0.212201  0.441007  1.000000 -1.035677

pathD<-as.data.frame(pathD)
## tc is outcome in my path
pathdiagram2(pathD=pathD,pathO=mypath,rangeD=c(1:4),rangeO=c(1:3),disease="CAD",
R2D=0.536535,R2O=0.988243)
```

SNP358.data *Data of 358 SNPs*

Description

SNP358.data were obtained from **GWAS** Meta-analyzed datasets of lipoprotein cholesterols and coronary artery disease. The data contain three numeric vectors (columns): SNPID(rsid), chromosome number (chr) and SNP position on chromosome (posit).

Usage

```
data("SNP358.data")
```

Format

A data frame with 358 observations on the following 3 variables.

rsid a character vector

chr a numeric vector

posit a numeric vector

Details

These 358 SNPs were chosen by using mktable with Pv = 5x10e-08,Pc = Pd =0.979 from lpd.data and cad.data. They provide a data example for how to perform annotation analysis of SNP positions.

Value

A set of data with 358 rows(SNPs) and 3 columns(SNP ID, chromosome # and SNP position on chromosomes).

Examples

```
data(SNP358.data)
## maybe str(SNP358.data) ; plot(SNP358.data) ...
```

SNP368annot.data *Annotation data of* 368SNPs

Description

The annotation data of 368SNPs are used to construct SNP distribution in gene elements (coding region, introne,UTR, etc). The data contain 12 vectors or variables but only Symbol and function_unit are used by ucscannot.R to build SNP distribution in gene elements.

Usage

```
data("SNP368annot.data")
```

Format

A data frame with 1053 observations on the following 6 variables.

`SNP` a string vector

`Allele` a string vector

`Strand` a numeric vector

`Symbol` a string vector

`Gene` a string vector

`function_unit` a string vector

Details

`SNP368annot.data` were obtained by performing mktable with `PV=5x10e-08`,`Pc=Pd=0.979` on lpd.data and cad.data and SNP tools. `SNP368annot.data` provides an practical example for constructing distribution of SNPs in gene elements. Note that `function_units` are gene elements.

Value

A dataset with 1053 rows and 6 columns for results of SNP annotation analysis. See format above.

Source

http://csg.sph.umich.edu//abecasis/public/lipids2013/

References

http://snp-nexus.org/test/snpnexus_19427/

Examples

```
data(SNP368annot.data)
```

`snpPositAnnot` SNP *Position Annotation*

Description

This function is used to perform position annotation analysis of `SNP`s chosen from `GWAS`.

Usage

```
snpPositAnnot(SNPdata, SNP_hg19="chr", main)
```

Arguments

SNPdata	SNPdata may be hg19 that is a string vector(chr##.########) or two numeric vectors (chromosome number and SNP position).
SNP_hg19	a string parameter. It may be "hg19" or "chr". If SNP_hg19="hg19",then SNPdata contain a string vector of hg19 or if SNP_hg19="chr", then SNPdata consist of at lest two columns: chr and posit. chr is chromosome number and posit is SNP physical position on chromosomes. If data sheet has chromosome X, then character "X" should be changed to 23 in chr vector or chr23.######## in hg19 vector.
main	a string which is title of graph. If no title is given, then main="".

Value

Return a set of numbers of SNPs between which interval length > **LG** on 23 chromosomes. This function also creates a histogram for averaged distances between SNPs and SNP numbers on chromosomes.

Note

This function can also be applied to hg18 data with SNP_hg19="hg18".

Author(s)

Yuan-De Tan <tanyuande@gmail.com>

See Also

barplot, text, chrp

Examples

```
data(SNP358.data)
SNP358<-DataFrame(SNP358.data)
snpPositAnnot(SNPdata=SNP358,SNP_hg19="chr",main="A")
```

---

ucscannot *Functional Annotation of* SNP*s Chosen*

Description

This function is used to give proportion of *SNP*s derived from functional elements of genes.

Usage

```
ucscannot(UCSCannot,SNPn,A=3,B=1.9,C=1.3,method=1)
```

Arguments

UCSCannot	annotation data obtained by performing SNP tools.
SNPn	numeric parameter for number of SNPs contained in UCSCannot
A	numeric parameter for title size, default=2.5
B	numeric parameter for label size, default=1.5
C	numeric parameter for labelrad distance,default=0.1
method	numeric parameter for choosing figure output methods. It has two options: method=1 has no legend but color and pie components are labeled with gene elements, method=2 has legend over pie. The default = 1.

Details

SNPs chosen by performing mktable should be copied to **Batch Query Box** in SNP annotation tool. After setting parameters and running by clicking run button, *SNP* annotation data will be obtained after running for a while. Consequence sheet of **UCSC** should be copied to excel sheet,"Predicted function" column name is changed to "function_unit" name and save it as csv format. These parametric defaults are used as graph image for publication, user can expand image to the maximum size and copy it to powerpoint that will give ideal effect. User also can use R packagelink{VariantAnnotation} to get SNP annotation result but the result must be constructed a table with function_unit column listing gene elements and Symbol column listing genes, otherwise, ucscannot will get an error.

Value

Create a color *pie3D* diagram and return a set of numeric values: proportions of code region, intron,3'and 5' UTRs and upstream and downstream etc.

Note

This function just need data of "Predicted function" and "symbol", so the other column data in UCSCannot do not impact the results of analysis.

Author(s)

Yuan-De Tan <tanyuande@gmail.com>

References

http://snp-nexus.org/index.html

See Also

mktable, pie3D,link{VariantAnnotion}

Examples

```
data(SNP368annot.data)
SNP368<-DataFrame(SNP368annot.data)
ucscannot(UCSCannot=SNP368,SNPn=368,A=1.5,B=1,C=1.3)
ucscannot(UCSCannot=SNP368,SNPn=368,A=1.5,B=1,C=1.3,method=2)
```

Index

∗ **Mendelian Randomiztion**
  mktable, 10
∗ **SNP position**
  snpPositAnnot, 19
∗ **SNP**
  chrp, 6
  ucscannot, 20
∗ **Selection of SNPs**
  mktable, 10
∗ **annotation**
  ucscannot, 20
∗ **chromosome**
  chrp, 6
∗ **datasets**
  beta.data, 4
  cad.data, 5
  lpd.data, 8
  SNP358.data, 17
  SNP368annot.data, 18
∗ **data**
  fmerge, 7
∗ **diagram**
  pathdiagram, 15
  pathdiagram2, 16
∗ **graphics**
  snpPositAnnot, 19
∗ **merge**
  fmerge, 7
∗ **package**
  GMRP-package, 2
∗ **path**
  path, 13
  pathdiagram, 15
  pathdiagram2, 16
∗ **structural equation model**
  path, 13

barplot, *20*
beta.data, 4

cad.data, 5
chrp, 6, *20*

fmerge, 7

GMRP *(*GMRP-package*)*, 2
GMRP-package, 2

lpd.data, 8

merge, *8*
mktable, *3*, *6*, 10, *21*

path, *3*, *12*, 13
pathdiagram, *3*, 15, *17*
pathdiagram2, *3*, *15*, 16
pie3D, *21*
plotmat, *3*, *15*, *17*
plotweb, *3*, *15*, *17*

SNP358.data, 17
SNP368annot.data, 18
snpPositAnnot, 19

text, *20*

ucscannot, 20