GWAS-based Mendelian Randomization Path Analysis

Yuan-De Tan and Dajiang Liu tanyuande@gmail.com

October 17, 2016

Abstract

GMRP can perform analyses of Mendelian randomization (*MR*),correlation, path of causal variables onto disease of interest and *SNP* annotation 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 vectors, *GMRP* performs correlation and path analyses to construct path diagrams of causal variables to the disease. *GMRP* consists of 8 R functions: `chrp`,`fmerg`,`mktable`,`pathdiagram`,`pathdiagram2`,`path`,`snpposit`,`ucscannot` and 5 datasets: `beta.data`,`cad.data`,`lpd.data`,`SNP358.data`,`SNP368annot.data`. `chrp` is used to separate string vector `hg19` into two numeric vectors: chromosome number and *SNP* chromosome position. Function `fmerg` is used to merge two *GWAS* result datasets into one dataset. Function `mktable` performs *SNP* selection and creates a standard beta table for function `path` to do *MR* and 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 *pathdiagram* is a *pathdiagram* of causal variables contributing to outcome and the outside *pathdiagram* is a path diagram 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 `fmerg`. `beta.data` are a standard beta dataset for path analysis. `SNP358.data` provide an example for `snpposit` to do *SNP* position annotation analysis and `SNP368annot.data` are for `ucscannot` to do *SNP* function annotation analysis.

Contents

1 Introduction

As an example of human disease, coronary artery disease (*CAD*) is one of the causes leading to death and infirmity worldwide [**?**]. Low-density lipoprotein cholesterol (*LDL*) and triglycerides (*TG*) are viewed as risk factors causing *CAD*. In epidemiological studies, plasma concentrations of increasing *TG* and *LDL* and deceasing *HDL* have been observed to be associated with risk for *CAD* [**?**, **?**]. However, from observational studies, one could not directly infer that these cholesterol concentrations in plasma are risk factors causing *CAD* [**?**, **?**, **?**, **?**]. A big limitation of observational studies is to difficultly distinguish between causal and spurious associations due to confounding [**?**]. An efficient approach to overcome this limitation is *Mendelian Randomization* (*MR*) analysis [**?**, **?**] where genetic variants are used as instrumental variables. For this reason, many investigators tried to use genetic variants to assess causality and estimate the causal effects on the diseases.

MR analysis can perfectly exclude confounding factors associated with disease. However, when we expand one causal variable to many, *MR* analysis becomes challenged and complicated because the genetic variant would have additional effects on the other risk factors, which violate assumption of no pleiotropy. An unknown genetic variant in *MR* analysis possibly provides a false instrument for causal effect assessment of risk factors on the disease. The reason is that if this genetic variant is in *LD* with another gene that is not used but has effect on the disease of study [**?**, **?**]. It then violates the third assumption. These two problems can be addressed by using multiple instrumental variables. For this reason, Do *et al* (2013)developed statistic approach to address this issue [**?**]. However, method of Do *et al* [**?**] cannot disentangle correlation effects among the multiple undefined risk factors on the disease of study. The beta values obtained from regression analyses are not direct causal effects because their effects are entangled with correlations among these undefined risk factors.

The best way to address the entanglement of multiple causal effects is path analysis that was developed by Wright [**?**, **?**]. This is because path analysis can dissect beta values into direct and indirect effects of causal variables on the

disease. However, path analysis has not broadly been applied to diseases because diseases are usually binary variable. The method of Do [**?**] makes it possible to apply path analysis to disentangle causal effects of undefined risk factors on diseases. For doing so, we here provide **R package** *GMRP* (*GWAS*-based *MR* and *path analysis*) to solve the above issues.

This vignette is intended to give a rapid introduction to the commands used in implementing *MR* analysis, regression analysis, and path analysis, including *SNP* annotation and chromosomal position analysis by means of the *GMRP* package.

We assume that user has the *GWAS* result data from *GWAS* analysis or *GWAS* meta analysis of *SNP*s associated with risk or confounding factors and a disease of study. If all studied causal variables of *GWAS* data are separately saved in different sheet files, then files are assumed to have the same sheet format and they are required to be merged by using function `fmerg` into one sheet file without disease *GWAS* data. After a standard beta table is created with `mktable`, user can use function `path` to perform *RM* and path analyses. Using the result of path analysis, user can draw path `plot`*(pathdiagram)* with functions `pathdiagram` and `pathdiagram2`. These will be introduced in detail in the following examples.

We begin by loading the *GMRP* package.

```
> library(GMRP)
```

2 Loading Data

GMRP provides five data files: `beta.data`, `cad.data`, `lpd.data`, `SNP358.data` and `SNP368annot.data` where

`lpd.data` was a subset (1069 SNPs) of four GWAS result datasets for *LDL*, *HDL*, *TG* and *TC*. These *GWAS* result data sheets were downloaded from the website[1] where there are 120165 SNPs scattered on 23 chromosomes and 40 variables. Four GWAS result datasets for *LDL*, *HDL*, *TG* and *TC* were merged into one data sheet by using `fmerg`*(fl1,fl2,ID1,ID2,A,B,method)* where *fl1* and *fl2* are two *GWAS* result data sheets. $ID1$ and $ID2$ are key id in files *fl1* and *fl2*, respectively, and required. A and B are postfix for *fl1* and *fl2*. Default values are A="" and B="". *method* is method for merging . In the current version, there are four methods: *method*="No" or "no" or "NO" or "N" or "n" means that the data with unmatched *SNP*s in *file1* and *file 2* are not saved in the merged file; *method*="ALL" or "All" or "all" or "A" or "a" indicates that the data with all unmatched *SNP*s in *file 1* and *file 2* are saved in the unpaired way in the merged data file; if *method*="*file1*", then those with unmatched *SNP*s only from file1 are saved or if *method*="*file2*", `fmerg` will save the data with unmatched *SNP*s only from *file2*". Here is a simple example:

```
> 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")
> data1<-DataFrame(data1)
> data2<-DataFrame(data2)
> data12 <- fmerg(fl1=data1, fl2=data2, ID1="SNP", ID2="SNP", A=".dat1", B=".dat2", method="No")
```

User can take the following approach to merge all four lipid files into a data sheet: *LDLHDL<-fmerg(fl1=LDLfile,fl2=HDLfile, ID1="SNP", ID2="SNP", A=".LDL", B=".HDL", method="No")*

TGTC<-fmerg(fl1=TGfile,fl2=TCfile, ID1="SNP", ID2="SNP", A=".TG", B=".TC", method="No")

lpd<-fmerg(fl1=LDLHDL,fl2=TGTC,ID1="SNP",ID2="SNP",A="",B="",method="No")

[1] http://csg.sph.umich.edu//abecasis/public/lipids2013/

`cad.data` was also a subset (1069 SNPs) of original GWAS meta-analyzed dataset that was downloaded from the website[2] and contains 2420360 *SNP*s and 12 variables.

`beta.data` that was created by using function `mktable` and `fmerg` from `lpd.data` and `cad.data` is a standard beta table for *MR* and path analyses.

`SNP358.data` contains 358 *SNP*s selected by `mktable` for *SNP* position annotation analysis.

`SNP368annot.data` is the data obtained from function analysis with *SNP* Annotation Tool and provides example of performing function `ucscanno` to draw a 3D pie and output the results of proportions of *SNP*s coming from gene function various elements.

■= data(cad.data) cad <- DataFrame(cad.data) head(cad)

```
> data(lpd.data)
> lpd <- DataFrame(lpd.data)
> head(lpd)
```

```
DataFrame with 6 rows and 40 columns
     SNP_hg18.HDL    SNP_hg19.HDL    rsid.HDL      A1.HDL      A2.HDL  beta.HDL    se.HDL
      <character>     <character> <character> <character> <character> <numeric> <numeric>
1  chr19:19379316  chr19:19518316   rs1000237           a           t    0.0083    0.0051
2   chr4:88283455   chr4:88064431  rs10023050           g           a    0.0174    0.0049
3  chr5:156366229  chr5:156433651  rs10036890           t           a    0.0044    0.0062
4  chr19:11085181  chr19:11224181   rs1003723           t           c    0.0051    0.0050
5   chr5:74753409   chr5:74717653  rs10038723           t           c    0.0043    0.0059
6 chr11:116227036 chr11:116721826  rs10047459           t           c    0.0203    0.0070
      N.HDL P.value.HDL Freq.A1.1000G.EUR.HDL    SNP_hg18.LDL    SNP_hg19.LDL    rsid.LDL
  <numeric>   <numeric>             <numeric>     <character>     <character> <character>
1  92630.00    0.127300                0.3272  chr19:19379316  chr19:19518316   rs1000237
2  92768.00    0.001186                0.4222   chr4:88283455   chr4:88064431  rs10023050
3  92644.98    0.551000                0.8417  chr5:156366229  chr5:156433651  rs10036890
4  86071.10    0.337800                0.4578  chr19:11085181  chr19:11224181   rs1003723
5  92790.02    0.488600                0.2375   chr5:74753409   chr5:74717653  rs10038723
6  76329.02    0.013370                0.8074 chr11:116227036 chr11:116721826  rs10047459
       A1.LDL      A2.LDL  beta.LDL    se.LDL     N.LDL P.value.LDL Freq.A1.1000G.EUR.LDL
  <character> <character> <numeric> <numeric> <numeric>   <numeric>             <numeric>
1           t           a    0.0373    0.0054  82988.90   2.755e-11                0.6728
2           a           g    0.0141    0.0052  83123.00   1.457e-02                0.5778
3           t           a    0.0245    0.0066  83022.99   3.848e-04                0.8417
4           t           c    0.0335    0.0053  77777.00   1.508e-09                0.4578
5           t           c    0.0605    0.0063  83135.02   1.381e-19                0.2375
6           c           t    0.0056    0.0073  71581.06   5.302e-01                0.1926
      SNP_hg18.TG     SNP_hg19.TG     rsid.TG       A1.TG       A2.TG   beta.TG     se.TG
      <character>     <character> <character> <character> <character> <numeric> <numeric>
1  chr19:19379316  chr19:19518316   rs1000237           t           a    0.0368    0.0049
2   chr4:88283455   chr4:88064431  rs10023050           a           g    0.0293    0.0048
3  chr5:156366229  chr5:156433651  rs10036890           t           a    0.0160    0.0060
4  chr19:11085181  chr19:11224181   rs1003723           t           c    0.0004    0.0048
5   chr5:74753409   chr5:74717653  rs10038723           t           c    0.0013    0.0058
6 chr11:116227036 chr11:116721826  rs10047459           c           t    0.0560    0.0066
      N.TG P.value.TG Freq.A1.1000G.EUR.TG    SNP_hg18.TC    SNP_hg19.TC    rsid.TC
  <numeric>  <numeric>             <numeric>     <character>     <character> <character>
1  86629.00  2.116e-12                0.6728  chr19:19379316  chr19:19518316   rs1000237
2  86765.00  2.449e-09                0.5778   chr4:88283455   chr4:88064431  rs10023050
3  86658.99  6.558e-03                0.8417  chr5:156366229  chr5:156433651  rs10036890
4  81143.90  8.068e-01                0.4578  chr19:11085181  chr19:11224181   rs1003723
5  86781.02  9.066e-01                0.2375   chr5:74753409   chr5:74717653  rs10038723
6  74837.95  1.523e-14                0.1926 chr11:116227036 chr11:116721826  rs10047459
       A1.TC       A2.TC   beta.TC     se.TC      N.TC P.value.TC Freq.A1.1000G.EUR.TC
```

[2] http://www.cardiogramplusc4d.org/downloads/

	<character>	<character>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
1	t	a	0.0374	0.0051	92528.90	1.557e-12	0.6728
2	a	g	0.0162	0.0049	92675.00	1.499e-03	0.5778
3	t	a	0.0351	0.0063	92553.21	3.974e-08	0.8417
4	t	c	0.0288	0.0051	85814.00	1.681e-08	0.4578
5	t	c	0.0515	0.0060	92698.92	4.354e-17	0.2375
6	c	t	0.0079	0.0069	80331.01	2.155e-01	0.1926

3 Preparation of Standard Beta Table

The standard beta table for *MR* and path analyses must have the standard format. It has columns: *chrn*, *posit*, *rsid*, $a1.x_1$, $a1.x_2$, $\cdots$, $a1.x_n$, $freq.x_1$, $freq.x_2$, $\cdots$, $freq.x_n$, $beta.x_1$, $beta.x_2$, $\cdots$, $beta.x_n$,$sd.x_1$, $sd.x_2$, $\cdots$, $sd.x_n$, pv_j, $N.x_1$, $N.x_2$, $\cdots$, $N.x_n$, pc_j, $hg.d$, $SNP.d$, $freq.d$, $beta.d$, $N.d$, $freq.case$, pd_j where x_1,x_2, $\cdots$, x_n are variables.

$beta$ is vector of beta values of *SNP*s on variable vector X=(x_1,x_2,$\cdots$,x_n).

$freq$ is vector of frequency of allele 1 with respect to variable vector X=(x_1,x_2,$\cdots$,x_n).

sd is vector of standard deviations of variable (x_1,x_2,$\cdots$,x_n) specific to *SNP*. If sd does not specifically correspond to *SNP*, then $sd.x_i$ has the same value for all *SNP*s.

d denotes disease.

N is sample size.

$freq.case$ is frequency of disease.

chrn is vector of chromosome number.

posit is position vector of *SNP*s on chromosomes. Some time, *chrn* and posit are combined into string *hg19* or *hg18*.

pv_j is defined as *p-value*, pc_j and pd_j as proportions of sample size for *SNP* j to the maximum sample size in causal variables and in disease, respectively.

We use function `mktable` to choose *SNP*s and make a standard beta table for *MR* and path analyses. For convenience, we first assign `lpd.data` to `lpd` and `cad.data` to `cad`:

The standard beta table will be created via 15 steps:

Step1: 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)
```

Step2: retrieve causal variables from data `lpd` and construct a matrix for beta:

```
> 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)
```

Step3: construct a matrix for allele1:

```
> 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)
```

Step4: give 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 <- round(sm/max(sm), 6)
```

Step5: Construct matrix for frequency of *allele1* in each 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)
```

Step6: construct matrix for sd of each causal variable (here sd is not specific to *SNP* j). The following sd values for *LDL, HDL,TG* and *TC* were means of standard deviations of these lipoprotein concentrations in plasma over 63 studies from Willer *et al* [**?**].

```
> 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)
```

Step7: *SNPID* and position:

```
> hg19 <- lpd$SNP_hg19.HDL
> rsid <- lpd$rsid.HDL
```

Step8: separate chromosome number and *SNP* position using `chrp`:

```
> chr<-chrp(hg=hg19)
```

Step9: get new data:

```
> newdata<-cbind(freq,beta,sd,pvj,ss,pcj)
> newdata<-cbind(chr,rsid,alle1,as.data.frame(newdata))
> dim(newdata)
```

Step10: retrieve data from cad and calculate pdj and frequency of coronary artery disease: *freq.case* 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
```

Step11: combine these cad variables into new data sheet using `cbind`

```
> 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)
```

Step12: give name vector of causal variables:

```
> varname <-c("CAD", "LDL", "HDL", "TG", "TC")
```

Step13: choose *SNP*s using parameters LG, Pv, Pc and Pd and create **standard beta table** using `mktable`*(cdata, ddata, rt,varname,LG, Pv, Pc,Pd)*where

cdata is beta data of SNPs regressed on causal variables. Here *cdata*=`newdata`.

ddata is beta data of SNPs regressed on the disease (here *CAD*). Here *ddata*=`newcad`.

LG: a numeric parameter. LG is used to choose *SNP*s with given interval threshold for linkage disequilibrium (*LD*). Default $LG = 10$.

Pv:a numeric parameter. Pv is used to choose *SNP*s with a given p-value cutoff. Default $Pv = 5 \times 10^{-}8$.

Pc: a numeric parameter. Pc is used to choose *SNP*s with a given cutoff for the proportion of sample size to maximum sample size in causal variable data. Default Pc=0.979.

Pd: a numeric parameter. Pd is used to choose *SNP* with a given cutoff for the proportion of sample size to the maximum sample size in disease data. Default Pd =0.979.

rt has two options: "beta" and "path". If *rt*="beta" or "Beta" or "B", then `mktable` return a beta coefficient matrix of *SNP*s regressed on causal variables and disease , if *rt*="path" or "Path" or "P" it returns a path coefficient matrix of *SNP*s directly contributing to causal variables and disease.

```
> mybeta <- mktable(cdata=newdata, ddata=newcad, rt="beta", varname=varname, LG=1, Pv=0.00000005, Pc=0.
> dim(mybeta)
> beta <- mybeta[,4:8]   #  standard beta table for path analysis
> snp <- mybeta[,1:3]   #  snp data for annotation analysis
> beta<-DataFrame(beta)
> head(beta)
```

4 Two-way Scatter Plots for Beta Values of Disease and Undefined Causal Variables

To roughly display relationship of the undefined causal variables to disease of study, we use simple **R** `plot` function to create two-way plots of beta of multiple *SNP* regressed on the undefined causal variable versus the disease.

```
> data(beta.data)
> beta.data<-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), mar=c(5.1, 4.1, 4.1, 2.1), oma=c(0, 0, 0, 0))
> plot(LDL,CAD, pch=19, col="blue", xlab="beta of SNPs on LDL", ylab="beta of SNP on CAD", cex.lab=1.5,
> abline(lm(CAD~LDL), col="red", lwd=2)
> plot(HDL, CAD, pch=19,col="blue", xlab="beta of SNPs on HDL", ylab="beta of SNP on CAD", cex.lab=1.5,
> abline(lm(CAD~HDL), col="red", lwd=2)
> plot(TG, CAD, pch=19, col="blue", xlab="beta of SNPs on TG", ylab="beta of SNP on CAD",cex.lab=1.5, c
> abline(lm(CAD~TG), col="red", lwd=2)
> plot(TC,CAD, pch=19, col="blue", xlab="beta of SNPs on TC", ylab="beta of SNP on CAD", cex.lab=1.5, c
> abline(lm(CAD~TC), col="red", lwd=2)
```

5 MR and Path Analysis

After **standard beta table** was successfully created by `mktable`, user can use function `path` to perform RM analysis (regression analysis of causal variable beta values on the disease or outcome beta values), correlation among the undefined causal variables and disease and path analyses with model of

$$y \sim x_1 + x_2 + \cdots + x_m$$

where y is disease or outcome variable, x_i is undefined causal variable i. Path analysis is based on *RM* analysis(regression coefficients of the causal beta on the disease beta). `path` will produce three tables: beta coefficients,

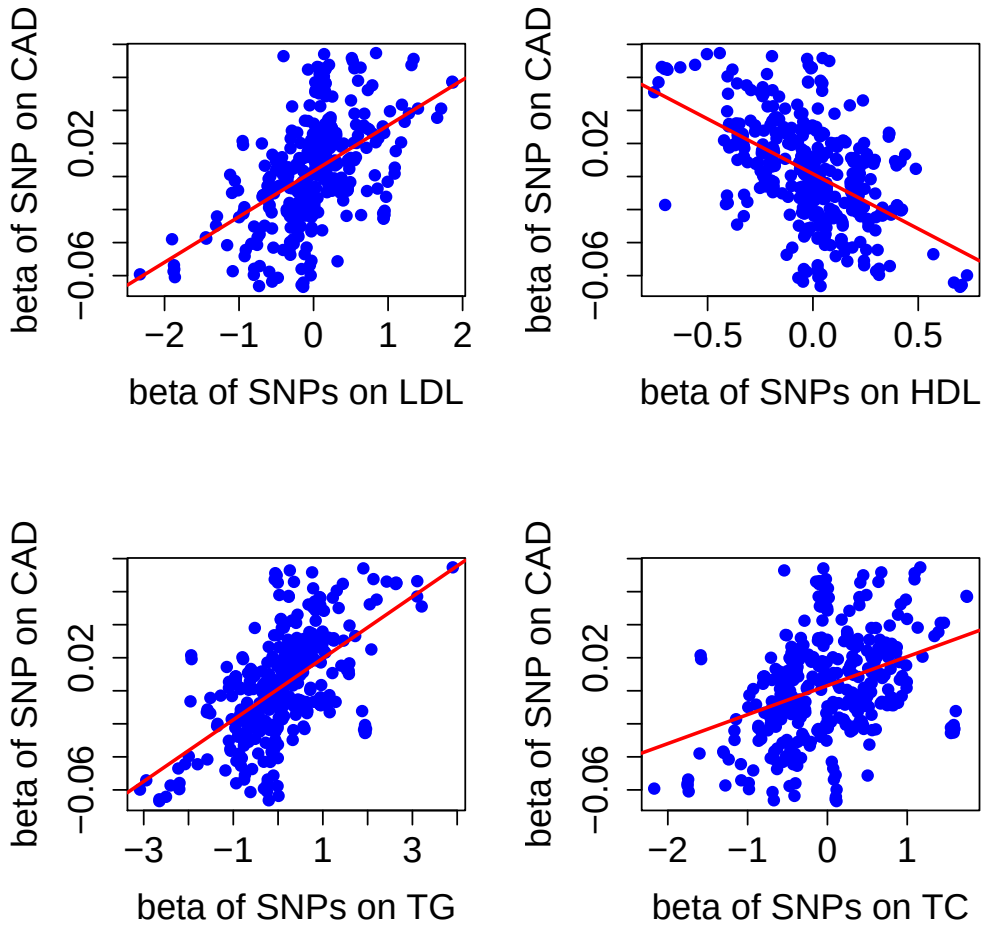

Figure 1: Scatter plots of lipid versus coronary artery disease (CAD) based on beta values of 368 SNPs regression analyses

sd values and t-test results of causal variables on disease or outcome, correlation matrix and path matrix(direct and indirect path coefficients)

```
> data(beta.data)
> mybeta <- DataFrame(beta.data)
> mod <- CAD~LDL+HDL+TG+TC
> pathvalue <- path(betav=mybeta, model=mod, outcome="CAD")
```

6 Create Path Diagram

Once user finished performance of path, user will have correlation matrix and direct path coefficients of undefined causal variable onto the disease. User is required to open a *csv* file saving results of path analysis and make table in **R Console** or copy correlation matrix without disease correlation coefficients to *excel* and copy direct path coefficients to the last column. Here is an example of making correlation and path table:

```
> 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<-as.data.frame(mypath)
> 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
```

The last column is direct path coefficients, we use "path" to name this column. With this table (for example, `mypath`), user can use function`pathdiagram`*(pathdata,disease,R2,range)* to create path diagram. Here

pathdata is path result data consisting of causal correlation matrix and direct path coefficient vectors.

disease is a string that specifies disease name. If the disease name is long or has multiple words, then we suggest an abbreviated name, for example, coronary artery disease are shorted as *"CAD"*.

$R2$, a numeric parameter, is R-square obtained from path analysis.

range is 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. For our current example, *range=c(1:3)*.

```
> library(diagram)
```

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

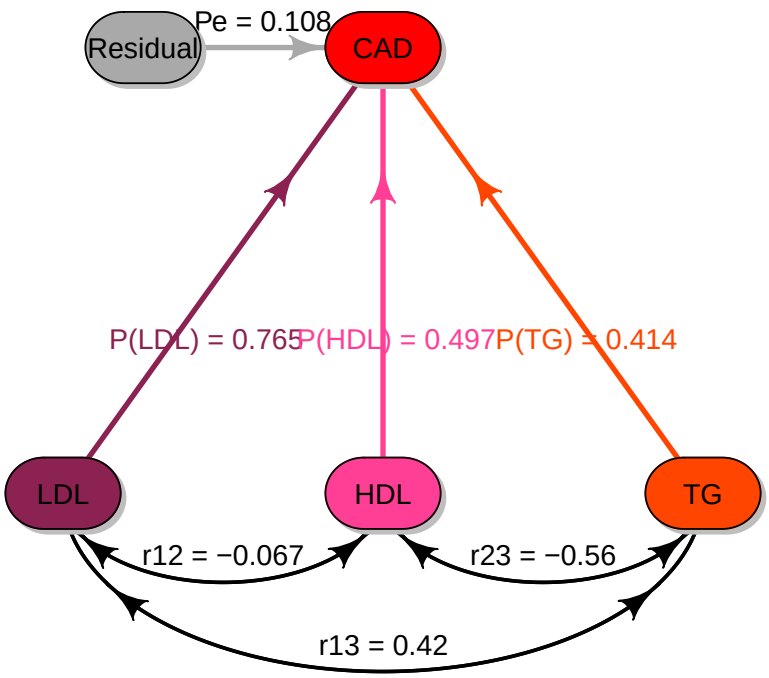

Figure 2: Path diagram demo. This path diagram shows the direct risk effects of causal variables *LDL, HDL and TG* on the disease *CAD* and their correlations.

7 Create Two-level Nested Path Diagram

Consider one of the causal variables is outcome of the other causal variables, but we also concern if all variables are risk factors for the disease of study. In this case we want to construct two-level nested path diagram using function `pathdiagram2`*(pathD,pathO,rangeD,rangeO,disease,R2D,R2O)* where

pathD is a R object that is disease path result data consisting of correlation matrix of undefined causal variables to be identified in Mendelian randomization analysis and path coefficient vector of these variables directly causing the disease of study.

pathO is a R object that is outcome path result data consisting of correlation matrix of undefined causal variables and path coefficient vector of these variables directly contributing to outcome. This outcome variable may be one of risk factors or causal variables in disease path data. These variables in *pathO* are the same with those in *pathD*.

rangeD is numeric vector, 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 is numeric vector, specifies column range for correlation coefficient matrix in *pathO*, see example in *rangeD*.

disease is a string that specifies disease name. If the disease name is long or has multiple words, then we suggest an abbreviated name, for example, *"coronary artery disease"* can be shortened as *"CAD"*.

Here is an example of *pathD* data:

```
> 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<-as.data.frame(pathD)
> 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
```

Using `pathD` and `mypath`, we can perform function `pathdiagram2` to create a two-level nested path diagram:

```
> pathdiagram2(pathD=pathD,pathO=mypath,rangeD=c(1:4),rangeO=c(1:3),disease="CAD", R2D=0.536535,R2O=0.9
```

Note that in the current version, *GMRP* can just create two-level nested path diagram, maybe in the later version, *GMRP* will create more complex path diagrams such as more than one inner path diagram andor multiple-disease path diagram using structure equations.

8 SNP Annotation Analysis

*SNP*s chosen will be annotated in function and chromosome position. Position annotation analysis will give position information of these selected *SNP*s on chromosomes including chromosome distribution and averaged intervals between *SNP*s. We use `snpposit` to perform *SNP* position annotation. This package provides 358 *SNP*s selected by `mktable`.

```
> data(SNP358.data)
> SNP358 <- DataFrame(SNP358.data)
> head(SNP358)
```

```
DataFrame with 6 rows and 3 columns
        rsid       chr     posit
    <factor> <integer> <integer>
1 rs10056022         5  74969415
```

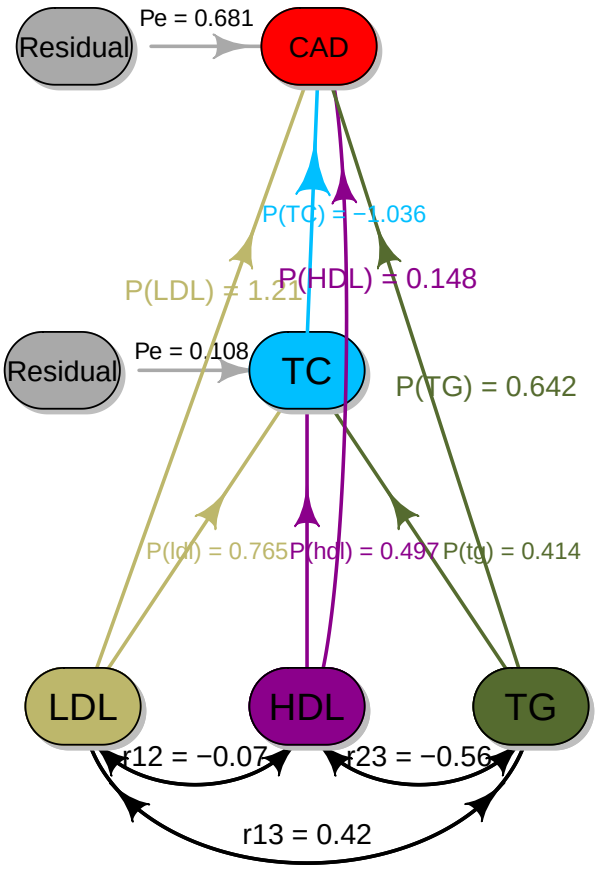

Figure 3: Demo of two-level nested *pathdiagram*. The outside *pathdiagram* shows the direct risk effects of undefined causal variables *LDL*, *HDL*, *TG* and *TC* on the disease *CAD*, the inner *pathdiagram* indicates the direct contributions of *LDL*, *HDL* and *TG* to *TC* and the correlation relationships among these variables.

```
2 rs10059560          5 156405784
3 rs10061689          5  74771387
4 rs10075465          5  74970975
5 rs10115928          9 107650843
6 rs10161126         12 110042348
```

`head` displays data format required by `snpposit`. User can create similar table for *SNP* position annotation analysis. To create chromosome position histogram, we need *graphics*:

```
> library(graphics)
```

With SNP data `SNP358`, we can perform *SNP* position annotation using function `snpposit`*(SNPdata,SNPhg19,LG,main,maxd)* where

SNPdata is R object that may be *hg19* that is a string vector(*chr##.########*) or two numeric vectors (chromosome number and *SNP* position).

SNPhg19 is a string parameter. It may be "*hg19*" or "*chr*". If *SNPhg19*="*hg19*",then *SNPdata* contains a string vector of *hg19* or if *SNPhg19*="*chr*", then *SNPdata* consists of at lest two numeric columns: *chr* and *posit*. *chr* is chromosome number and *posit* is *SNP* physical position on chromosomes. Note that "*chr*" and "*posit*" are required column names in *SNPdata* if *SNPhg19* ="*chr*".

LG is a numeric parameter that gives maximum permissible distance between positions. Its default is 10.

main is a string which is title of graph. If no title is given, then man="". Its default is "A".

maxd is a numeric parameter that is maximum distance for truncating chromosome columns. If there are not big differences among 23 chromosomes, then *maxd* can be set to be larger than 2000kbp. Its default is 2000kbp.

```
> snpposit(SNPdata=SNP358,SNP_hg19="chr",LG=10,main="A",maxd=2000)
 [1] 37 68  7  8 34 10  2 44 13  3 52 12  0  0  5 13 11  7  9
 [1] 37 68  7  8 34 10  2 44 13  3 52 12  0  0  5 13 11  7  9
```

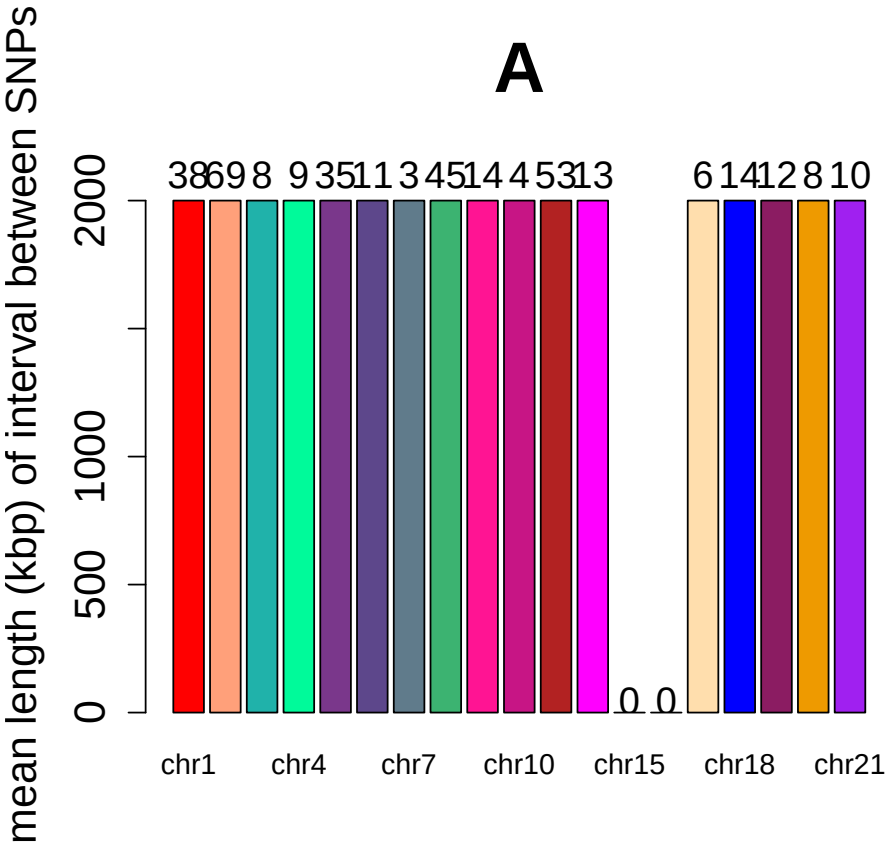

Figure 4: Chromosomal histogram of 358 selected *SNP*s. Averaged lengths of *SNP* intervals on chromosome mean that the *SNP*s on a chromosome have their averaged lengths of intervals between them. All averaged lengths over 2000kb on chromosomes were truncated, the *SNP*s on these chromosomes have at least more than 2000kbp averaged length of interval. Numbers above *chr* columns are numbers of *SNP* distributed on the chromosomes

SNP function annotation analysis has two steps:

Step 1: copy *SNP ID*s selected to **Batch Query** box in*SNP* Annotation Tool. After setting parameters and running by clicking *run button*, SNP annotation result will be obtained after running for a while. Choose consequence sheet of *UCSC* and copy the results to excel sheet,"*Predicted function*" column name is changed to "*functionunit*" name and save it as *csv* format.

Step2: input the *csv* file into *R Console* using **R** function `read.csv`. In *GMRP* package, we have provided data for *SNP* function annotation analysis.

```
> data(SNP368annot.data)
> SNP368<-DataFrame(SNP368annot.data)
> SNP368[1:10, ]
```

```
DataFrame with 10 rows and 6 columns
          SNP   Allele    Strand   Symbol       Gene function_unit
     <factor> <factor> <integer> <factor>   <factor>      <factor>
1  rs10056022      G|A         1     POC5 uc003keg.4   3downstream
2  rs10056022      G|A         1     POC5 uc003keh.4   3downstream
3  rs10056022      G|A         1  ANKDD1B uc010izt.4   3downstream
4  rs10056022      G|A         1     POC5 uc010izu.3   3downstream
5  rs10061689      G|A         1 COL4A3BP uc003kds.3      intronic
6  rs10061689      G|A         1 COL4A3BP uc003kdt.3      intronic
7  rs10061689      G|A         1 COL4A3BP uc003kdu.2      intronic
8  rs10061689      G|A         1 COL4A3BP uc011csu.2      intronic
9  rs10075465      C|G         1     POC5 uc003keg.4      intronic
10 rs10075465      C|G         1     POC5 uc003keh.4      intronic
```

We perform function `ucscannot` to summarize proportions of SNPs coming from gene various elements such as code region, introns, etc, and then create 3D pie with `pie3D` of *plotrix*.

```
> library(plotrix)
```

`ucscannot` has four parameters to be inputted: SNPn, A, B and C, a method and UCSC annotated data:

UCSCannot is annotation data obtained by performing SNP tools.

SNPn is numeric parameter for number of SNPs contained in *UCSCannot*

A is numeric parameter for title size, default=2.5.

B is numeric parameter for label size, default=1.5.

C is numeric parameter for *labelrad* distance,default=0.1.

method is 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.

```
> ucscannot(UCSCannot=SNP368,SNPn=368)
```

```
     genes      exons   introns       TUR3        UTR5  intergenes   upstream downstream
[1,]   166 0.01234568 0.8138651 0.01424501 0.001899335 0.004748338 0.05508072 0.09781576
```

```
> ucscannot(UCSCannot=SNP368,SNPn=368,A=3,B=2,C=1.3,method=2)
```

```
     genes      exons   introns       TUR3        UTR5  intergenes   upstream downstream
[1,]   166 0.01234568 0.8138651 0.01424501 0.001899335 0.004748338 0.05508072 0.09781576
```

```
     genes      exons   introns       TUR3       UTR5  intergenes   upstream downstream
[1,]   166 0.01234568 0.8138651 0.01424501 0.001899335 0.004748338 0.05508072 0.09781576
```

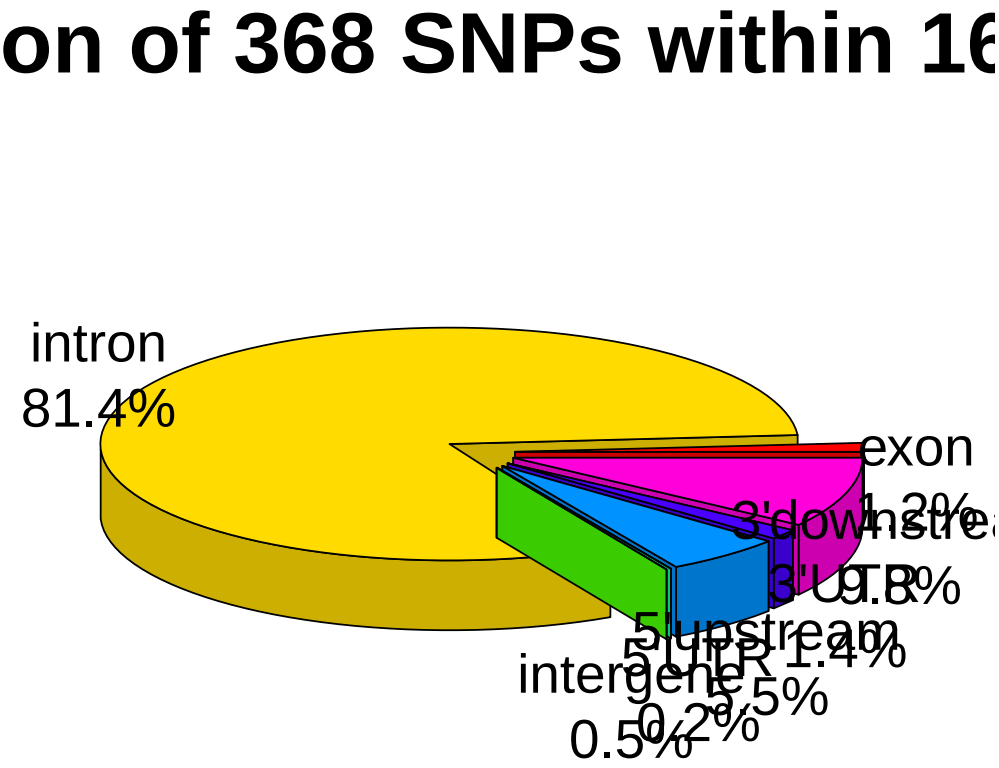

Figure 5: Distribution of the selected *SNP*s in gene function elements.

```
     genes      exons   introns       TUR3       UTR5  intergenes   upstream downstream
[1,]   166 0.01234568 0.8138651 0.01424501 0.001899335 0.004748338 0.05508072 0.09781576
```

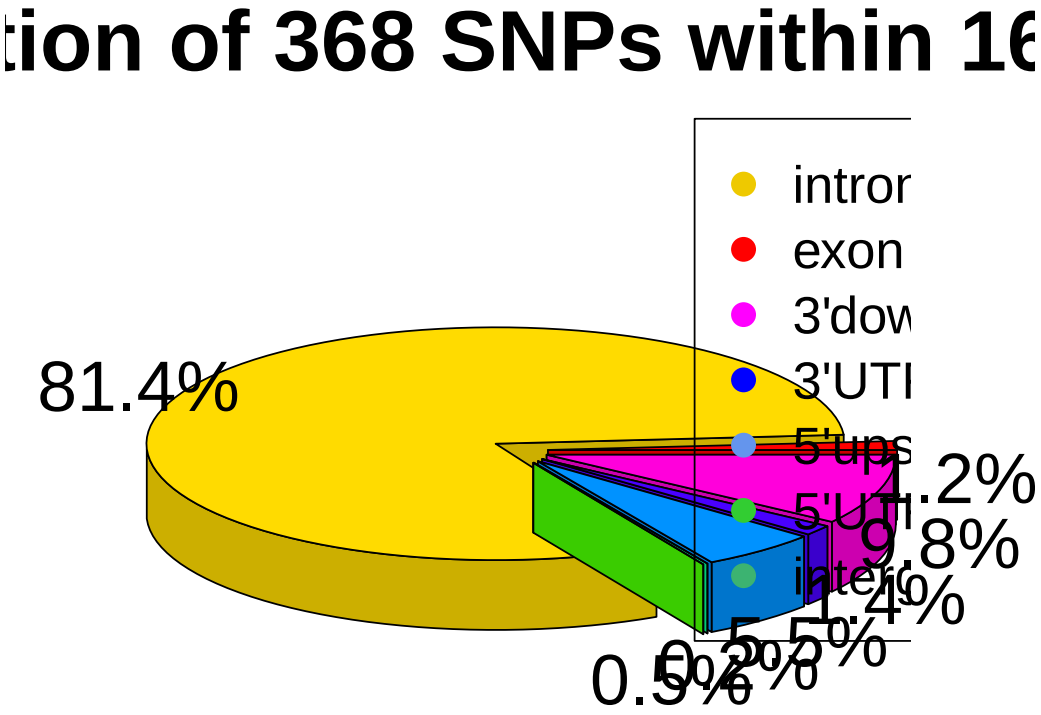

Figure 6: Distribution of the selected *SNP*s in gene function elements.

9 Session Info

```
> sessionInfo()

R version 3.3.1 (2016-06-21)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] parallel  stats4    stats     graphics  grDevices utils     datasets  methods
[9] base

other attached packages:
[1] GMRP_1.2.0           GenomicRanges_1.26.0 GenomeInfoDb_1.10.0  IRanges_2.8.0
[5] S4Vectors_0.12.0     BiocGenerics_0.20.0  plotrix_3.6-3        diagram_1.6.3
[9] shape_1.4.2

loaded via a namespace (and not attached):
[1] zlibbioc_1.20.0  BiocStyle_2.2.0  XVector_0.14.0   tools_3.3.1      data.table_1.9.6
[6] chron_2.3-47
```

References

[1] Murray, C.J. and Lopez, A.D. (1997) Global mortality, disability, and the contribution of risk factors: Global Burden of Disease Study. *Lancet* **349**: 1436-1442.

[2] Di Angelantonio, E., Sarwar, N., Perry, P., Kaptoge, S., Ray, K.K., Thompson, A., Wood, A.M., Lewington, S., Sattar, N., Packard, C.J. et al. (2009) Major lipids, apolipoproteins, and risk of vascular disease. *JAMA* **302**: 1993-2000.

[3] Sarwar, N., Danesh, J., Eiriksdottir, G., Sigurdsson, G., Wareham, N., Bingham, S., Boekholdt, S.M., Khaw, K.T., and Gudnason, V. (2007) Triglycerides and the risk of coronary heart disease: 10,158 incident cases among 262,525 participants in 29 Western prospective studies. *Circulation* **115**: 450-458.

[4] Do, R., Willer, C.J., Schmidt, E.M., Sengupta, S., Gao, C., Peloso, G.M., Gustafsson, S., Kanoni, S., Ganna, A., Chen, J. et al. (2013) Common variants associated with plasma triglycerides and risk for coronary artery disease. *Nat Genet* **45**: 1345-1352.

[5] Sarwar, N., Danesh, J., Eiriksdottir, G., Sigurdsson, G., Wareham, N., Bingham, S., Boekholdt, S.M., Khaw, K.T., and Gudnason, V. (2007) Triglycerides and the risk of coronary heart disease: 10,158 incident cases among 262,525 participants in 29 Western prospective studies. *Circulation* **115**: 450-458

[6] Voight, B.F. Peloso, G.M. Orho-Melander, M. Frikke-Schmidt, R. Barbalic, M. Jensen, M.K. Hindy, G. Holm, H. Ding, E.L. Johnson, T. et al. (2012) Plasma HDL cholesterol and risk of myocardial infarction: a mendelian randomisation study. *Lancet* **380**: 572-580.

[7] Pichler, I., Del Greco, M.F., Gogele, M., Lill, C.M., Bertram, L., Do, C.B., Eriksson, N., Foroud, T., Myers, R.H., Nalls, M. et al. (2013) Serum iron levels and the risk of Parkinson disease: a Mendelian randomization study. *PLoS Med* **10**: e1001462.

[8] Smith, G.D. and Ebrahim, S. (2003) 'Mendelian randomization': can genetic epidemiology contribute to understanding environmental determinants of disease? *Int J Epidemiol* **32**: 1-22.

[9] Sheehan, N.A., Meng, S., and Didelez, V. (2010) *Mendelian randomisation: a tool for assessing causality in observational epidemiology. Methods Mol Biol* **713**: 153-166.

[10] 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.

[11] Wright, S. (1934) The method of path coefficients. *Annals of Mathematical Statistics* **5** (3): 161-215.

[12] Wright, S. 1921 Correlation and causation. *J.Agricultural Research* **20**: 557-585.