Tutorial of Uncovering causal gene-tissue pairs and variants: A multivariable TWAS method controlling for infinitesimal effects

Yihe Yang, Noah Lorincz-Comi, Xiaofeng Zhu

Contents

1 Data structure

1.1 LD reference panel

The first dataset is the LD reference panel. This dataset is derived from the 9,680 unrelated individuals we described in the paper, selected from approximately 500,000 imputed individuals in the UK Biobank (UKBB). We refined the data using the bim files from these individuals. The files we shared on Google Drive include all 9.32 million SNPs involved; however, in this tutorial, we will focus only on a subset in the PCSK9 locus. Below is a glimpse of the data structure:

```r
library(data.table)
library(dplyr)
library(tidyr)
variant=readRDS("variant.rds")
variant
```

```
##                       SNP   CHR       BP     A1     A2      Freq MarkerName
##                    <char> <int>    <int> <char> <char>     <num>     <char>
##     1: 1:54005191_CCACA_C     1 54005191      C  CCACA 0.3992490 1:54005191
##     2:    1:54012271_AG_A     1 54012271     AG      A 0.6292130 1:54012271
##     3:    1:54012621_TG_T     1 54012621     TG      T 0.6295230 1:54012621
##     4:    1:54035956_TA_T     1 54035956      T     TA 0.1912090 1:54035956
##     5:   1:54070614_TAC_T     1 54070614    TAC      T 0.9896660 1:54070614
##    ---
## 10782:          rs9919296     1 54578136      C      T 0.0209929 1:54578136
## 10783:          rs9919314     1 54579134      C      T 0.0209763 1:54579134
## 10784:           rs993075     1 56994938      G      C 0.4923560 1:56994938
## 10785:          rs9970807     1 56965664      T      C 0.0921541 1:56965664
## 10786:           rs998154     1 55596384      C      T 0.1690460 1:55596384
```

In this reference panel, `SNP` is the identifier for the variants; `CHR` represents the chromosome; `BP` indicates the base pair position in the hg19 genome build; `A1` is the effect allele as specified in the BED file; `A2` is the other allele; `Freq` denotes the frequency of the effect allele; and `MarkerName` is another unique identifier for the variants in the format CHR:BP. It is important to note that some variants in the UKBB bed file do not have an rsID. For these variants, their `SNP` are in the format CHR:BP:A2:A1.

1.2 GWAS summary data

The second dataset is the GWAS summary data, which should include at least the following columns: `SNP`, `A1`, `A2`, `Zscore`, and `N`. In this dataset, `Zscore` represents the Z-score of the marginal effect size estimates from the outcome GWAS, while `N` denotes the sample size. Other statistics can be deduced from `Zscore` and `N`, e.g.,

$$\text{BETA} = \frac{\texttt{Zscore}}{\sqrt{\texttt{N}}}, \quad \text{SE} = \frac{1}{\sqrt{\texttt{N}}}.$$

Below is an example of the dataset's structure:

```r
library(arrow)
LDL=read_parquet("LDL.parquet")
LDL[-5,]
```

```
## Key: <SNP>
##                    SNP   CHR       BP     A1     A2      Zscore       N
##                 <char> <int>    <int> <char> <char>       <num>   <int>
##    1: 1:54388067_CAA_C     1 54388067      C    CAA  -0.9536883  389563
##    2: 1:54389774_TAG_T     1 54389774    TAG      T  -0.7761577  487566
##    3:  1:54401303_CT_C     1 54401303      C     CT   1.0402465 1041795
##    4: 1:54406627_ATT_A     1 54406627      A    ATT  -3.1568909 1065377
##    5:  1:54430681_GT_G     1 54430681      G     GT  -2.6594631 1058084
##   ---
## 7958:        rs9919142     1 54579153      G      A   1.0001670 1230988
## 7959:        rs9919295     1 54578100      C      T   0.9664437 1231010
## 7960:        rs9919296     1 54578136      C      T   0.9008325 1231009
## 7961:        rs9919314     1 54579134      C      T   0.9813689 1231009
## 7962:         rs998154     1 55596384      C      T -10.8367174 1231250
```

It should be noted that we did not use the original `SNP` identifiers from the GWAS. Instead, we merged the GWAS data with the variant file using the `MarkerName` (CHR:BP) identifiers, and then assigned `SNP` from the variant file to the corresponding entries in the GWAS data. In cases where the GWAS file is based on the hg38 genome build, we use LiftOver to convert it to hg19.

1.3 eQTL and sQTL summary data

The next dataset, which has a more complex structure, is the summary data for eQTL and sQTL. We preprocessed the data provided by GTEx and other studies to retain only the following columns: `SNP`, `CHR`, `BP`, `A1`, `A2`, `P`, `Zscore`, `N`, `Gene`, `GeneSymbol`, `Tissue`, `Variable`, and `xQTL`. Here,

- `P`: the P-value of the marginal effect for the xQTL;
- `Gene`: the Ensembl ID of the gene associated with the xQTL;
- `GeneSymbol`: the symbol for the corresponding gene,
- `Tissue`: the tissue in which the gene was sequenced;
- `Variable`: the combination of GeneSymbol (or Gene for sQTL) + Tissue
- `xQTL`: an indicator of sQTL or eQTL.

An example of the dataset structure is shown below:

```
eQTLsQTL=read_parquet("eQTLsQTL.parquet")%>%mutate(P=pchisq(Zscore^2,1,lower.tail=F))%>%
         dplyr::select(SNP,CHR,BP,A1,A2,P,Zscore,N,Gene,GeneSymbol,Tissue,Variable,xQTL)
head(eQTLsQTL)
```

```
## Key: <SNP>
##                  SNP   CHR        BP     A1     A2         P     Zscore     N
##               <char> <int>     <int> <char> <char>     <num>      <num> <num>
## 1: 1:54389774_TAG_T     1 54389774    TAG      T 0.5831150  0.5488404   266
## 2: 1:54389774_TAG_T     1 54389774    TAG      T 0.9068218 -0.1170483   266
## 3: 1:54389774_TAG_T     1 54389774    TAG      T 0.7200958  0.3583308   266
## 4: 1:54389774_TAG_T     1 54389774    TAG      T 0.8886494 -0.1400135   266
## 5: 1:54389774_TAG_T     1 54389774    TAG      T 0.1775865  1.3482234   266
## 6: 1:54389774_TAG_T     1 54389774    TAG      T 0.1355475 -1.4925787   266
##                                                    Gene   GeneSymbol
##                                                  <char>       <char>
## 1: chr1:52914095:52917544:clu_46563:ENSG00000121310.16 ECHDC2(sQTL)
## 2: chr1:52914095:52916465:clu_46563:ENSG00000121310.16 ECHDC2(sQTL)
## 3: chr1:52912534:52913895:clu_46563:ENSG00000121310.16 ECHDC2(sQTL)
## 4: chr1:52912021:52913895:clu_46563:ENSG00000121310.16 ECHDC2(sQTL)
## 5: chr1:52912021:52912754:clu_46563:ENSG00000121310.16 ECHDC2(sQTL)
## 6: chr1:52911790:52921553:clu_46563:ENSG00000121310.16 ECHDC2(sQTL)
##           Tissue
##           <char>
## 1: Colon_Sigmoid
## 2: Colon_Sigmoid
## 3: Colon_Sigmoid
## 4: Colon_Sigmoid
## 5: Colon_Sigmoid
## 6: Colon_Sigmoid
##                                                              Variable   xQTL
##                                                                <char> <char>
## 1: chr1:52914095:52917544:clu_46563:ENSG00000121310.16+Colon_Sigmoid   sQTL
## 2: chr1:52914095:52916465:clu_46563:ENSG00000121310.16+Colon_Sigmoid   sQTL
## 3: chr1:52912534:52913895:clu_46563:ENSG00000121310.16+Colon_Sigmoid   sQTL
## 4: chr1:52912021:52913895:clu_46563:ENSG00000121310.16+Colon_Sigmoid   sQTL
## 5: chr1:52912021:52912754:clu_46563:ENSG00000121310.16+Colon_Sigmoid   sQTL
## 6: chr1:52911790:52921553:clu_46563:ENSG00000121310.16+Colon_Sigmoid   sQTL
```

It should be noted that entries in the format chr1:52906611:52907868:clu_52882:ENSG00000121310.16 indicate a specific splicing event for the gene ENSG00000121310, as defined by LeafCutter.

In practice, combining multiple sQTL and eQTL datasets is a challenging task. In the section where we explain the key functions, we will detail our strategy for handling and integrating these datasets.

1.4 LD refernece panel with individual

We used the UKBB BED file to estimate the LD reference, with a sample size of 9,680. Below is a glimpse of the data structure:

```
UKBBGenotype=readRDS("UKBBGenotype.rds")
UKBBGenotype[1:10,1:5]
```

```
##    rs10047036 rs12728734 rs150256195 rs74510493 rs114570917
## 1           2          2           2          2           2
## 2           2          1           2          2           2
## 3           2          2           2          2           2
## 4           1          1           2          2           2
## 5           2          2           2          2           2
## 6           2          2           2          2           2
## 7           2          2           2          2           2
## 8           2          2           2          2           1
## 9           1          2           2          2           2
## 10          2          2           2          2           2
```

2 Step-by-step analysis

2.1 Allele harmonisation

In the first step, we adjust the direction of the Z-scores in the GWAS and xQTL summary data to ensure that the effect alleles in these datasets match the effect alleles in our reference panel. This step is crucial because the LD matrix is estimated from this reference panel, and accurate LD estimation is fundamental to all statistical methods based on GWAS summary data. We wrote a function, `allele_harmonise()` in the R package `TGVIS`, to perform this step:

```
library(TGVIS)
LDL=allele_harmonise(ref_panel=variant[,c("SNP","A1","A2")],gwas_data=LDL)
eQTLsQTL=allele_harmonise(ref_panel=variant[,c("SNP","A1","A2")],gwas_data=eQTLsQTL)
eQTLsQTL=eQTLsQTL[LDL,nomatch=0]
setnames(eQTLsQTL,"i.Zscore","Zscore.y")
setnames(eQTLsQTL,"Zscore","Zscore.x")
```

In `allele_harmonise()`, we automatically set `gwas_data` as a data.table with `key="SNP"`, allowing `eQTLsQTL=eQTLsQTL[LDL, nomatch=0]` to efficiently merge the two datasets. The reason for merging these datasets, as described in here, there is a vast and heterogeneous landscape of publicly available GWAS. These studies were conducted on different genotyping platforms, using different imputation schemes, and defined on different releases of the human genome., is that there is a vast and heterogeneous landscape of publicly available GWAS. These studies were conducted on different genotyping platforms, using different imputation schemes, and defined on different releases of the human genome. Therefore, we aim to find the common variants between the GWAS and xQTL summary data to perform the analysis.

2.2 Extracting the moderately correlated variants

The next step is to remove highly correlated variants using C+T. Although SuSiE can group highly correlated or statistically duplicated variants into a single group and assign them one effect, including many redundant variants can significantly increase the dimensionality of the model. Therefore, primarily to enhance computational efficiency, we recommend retaining only moderately correlated variants.

We use the smallest p-value across all tissue pairs corresponding to each variant as the input p-value for PLINK to extract a subset of moderately correlated variants. While we will not execute the following steps in this tutorial, we will provide the code for you. You can modify the file paths as needed for your own data.

```
A=eQTLsQTL%>%dplyr::select(SNP,P)
A=A[, .SD[which.min(P)], by=SNP]
A=A[which(A$P<5e-4),]
write.table(A,"Your_path.txt",quote=F, sep="\t", row.name=F)
setwd("Your_path_to_PLINK")
system("./plink --bfile Your_bed_file --clump Your_path.txt
            --clump-field P --clump-kb 1000 --clump-p1 1e-5
            --clump-p2 1e-5 --clump-r2 0.5 --out Your_path")
setwd("Your_analysis_path")
plinkfile=fread("Your_path.clumped")
plinkfile=plinkfile$SNP
```

The most important part in this step is:

- `-clump-kb 1000`: we consider the window size to be 1M,
- `-clump-p1 1e-5`: we use the threshold of 1E-5,
- `-clump-r2 0.5`: the correlation between two variants is in the range $(-\sqrt{0.5}, \sqrt{0.5})$.

Since direct causal variants might not be linked to any gene-tissue pairs, I perform clumping on the outcome GWAS to identify outcome-associated variants:

```
A1=LDL[which(LDL$SNP%in%unique(eQTLsQTL$SNP)),]
A1$P=pchisq(A1$Zscore^2,1,lower.tail=F);
```

```
A1=A1[,c("SNP","P")]
A1=A1[which(A1$P<min(5e-8,quantile(A1$P,0.1))),]
write.table(A1,"Your_path.txt",quote=F, sep="\t", row.name=F)
setwd("Your_path_to_PLINK")
system("./plink --bfile Your_bed_file --clump Your_path.txt
            --clump-field P --clump-kb 1000 --clump-p1 1e-5
            --clump-p2 1e-5 --clump-r2 0.5 --out Your_path")
setwd("Your_analysis_path")
plinkfile1=fread("Your_path.clumped")
plinkfile1=plinkfile1$SNP
```

Finally, we merge these two lists of variants and use C+T to remove any highly correlated variants (which are typically few), resulting in the final pool of variants for analysis:

```
A=data.frame(SNP=unique(c(plinkfile,plinkfile1)))
A$P=0.05
write.table(A,"Your_path.txt",quote=F, sep="\t", row.name=F)
setwd("Your_path_to_PLINK")
system("./plink --bfile Your_bed_file --clump Your_path.txt
            --clump-field P --clump-kb 1000 --clump-p1 1e-5
            --clump-p2 1e-5 --clump-r2 0.5 --out Your_path")
setwd("Your_analysis_path")
plinkfile=fread("Your_path.clumped")
rsid=plinkfile$SNP
```

We have recorded this pool of variants in the PCSK9 locus:

```
rsid=readRDS("SNP_lowLD.rds")
gwas_eQTLsQTL=eQTLsQTL[which(eQTLsQTL$SNP%in%rsid),]
```

2.3 Regularization of LD matrix

Our next step is to estimate a "good" LD matrix. We use the POET-shrinkage method to estimate such an LD matrix. The code is as follows:

```
R0=cor(UKBBGenotype)
R0[is.na(R0)]=0;diag(R0)=1
R0=poet_shrinkage(R0)
R0=(t(R0)+R0)/2
genosnp=colnames(UKBBGenotype)
rownames(R0)=colnames(R0)=genosnp
```

2.4 Construction of design matrix of gene-tissue pairs

Our next step is to extract the design matrix of gene-tissue pairs from the `eQTLsQTL` data.table. We provide the function `make_design_matrix()`, which converts the Z-scores in `eQTLsQTL` into an $M \times p$ design matrix, where M is the number of variants and p is the total number of gene-tissue pairs:

```
bX0=make_design_matrix(eQTLsQTL[,c("SNP","Variable","Zscore.x")])
bX0=bX0[genosnp,]
```

In data.table `eQTLsQTL`, `Zscore.x` represents the Z-score of the xQTL effect, while `Zscore.y` is the Z-score from the outcome GWAS. We match the rows of `bX0` with the LD matrix using the code `bX0=bX0[genosnp,]`.

Our next step is to impute missing values in the Z-scores as 0. Since GTEx only provides the marginal xQTL effect sizes for variants near the gene's TSS, this results in missing values. Before imputing, we remove variants and gene-tissue pairs with excessive missing values. In this analysis, we exclude variants with more than 95% missing values and genes with more than 90% missing values:

```
bX=remove_missing_row_column(bX0,rowfirst=F,rowthres=0.95,colthres=0.9)
genosnp=rownames(bX)
R0=R0[genosnp,genosnp]
bX=as.matrix(bX[genosnp,])
bX[is.na(bX)]=0
```

2.5 eQTL selection

Our next step is to use SuSiE for gene-tissue pair eQTL selection. The first step is to extract the average sample size for each gene-tissue pair from eQTLsQTL to use as the input sample size for SuSiE:

```
VariableName=unique(eQTLsQTL$Variable)
NeQTLsQTL=eQTLsQTL[,.(NeQTLsQTL=mean(N)),by=Variable][Variable%in%VariableName,NeQTLsQTL]
names(NeQTLsQTL)=VariableName
NeQTLsQTL=NeQTLsQTL[colnames(bX)]
```

We have encapsulated a for-loop function based on `susie_rss()` to perform eQTL selection for each gene-tissue pair:

```
fiteQTL=eQTLmapping_susie(bX=bX,LD=R0,Nvec=NeQTLsQTL,L=3,pip.thres=0.5,pip.min=0.25)
```

```
##   |                                                                      |
```

```
bXest=fiteQTL$Estimate
ind=which(colSums(abs(bXest))==0)
bXest=bXest[,-ind]
bX=bX[,-ind]
NeQTLsQTL=NeQTLsQTL[colnames(bX)]
```

2.6 Performing S-Predixcan and its modifier to remove noise gene-tissue pairs

In this example, our original design matrix includes $M = 381$ variants and $p = 4878$ gene-tissue pairs:

```
dim(bX0)
```

```
## [1]  381 4878
```

After quality control and eQTL selection, we retained $p = 784$ gene-tissue pairs.

```
dim(bX)
```

```
## [1] 379 784
```

However, in many cases, the original number of gene-tissue pairs could be tens of thousands, and even after eQTL selection, there could still be thousands of gene-tissue pairs. Therefore, we perform a univariable TWAS with S-PrediXcan and its modifier to pre-reduce the dimensionality, making TGVIS and TGFM more efficient.

Let's first organize the data and set up a data frame to store the results:

```
bY=eQTLsQTL[,c("SNP","Zscore.y")]
bY=bY[!duplicated(bY$SNP),]%>%as.data.frame(.)
rownames(bY)=bY$SNP
bY=bY[genosnp,]
UVTWAS=matrix(0,ncol(bXest),6)
colnames(UVTWAS)=c("Variable","Type","Est1","P1","Est2","P2")
UVTWAS=as.data.frame(UVTWAS)
UVTWAS[,1]=colnames(bXest)
UVTWAS[,2]=eQTLsQTL$xQTL[match(UVTWAS[,1],eQTLsQTL$Variable)]
UVTWAS[,c(4,6)]=1
```

Next, we execute S-PrediXcan and its modifier:

```
for(i in 1:ncol(bXest)){
errorindicate=0
tryCatch({
bxx=bX[,i]
indx=which(bxx!=0)
bx=bXest[indx,i]
by=bY$Zscore.y[indx]
bxx=bxx[indx]
if(sum(bx!=0)==1){
pleiotropy.rm=which(bx!=0)
}else{
pleiotropy.rm=NULL
}
fitModifier=modified_predixcan(by=by,bxest=bx,LD=R0[indx,indx],
            pleiotropy.rm=pleiotropy.rm,tauvec=seq(3,21,by=3))
fitSpredixcan=modified_predixcan(by=by,bxest=bx,LD=R0[indx,indx],
            pleiotropy.rm=pleiotropy.rm,tauvec=10000000)
UVTWAS[i,3]=fitSpredixcan$theta
UVTWAS[i,4]=pchisq(fitSpredixcan$theta^2/fitSpredixcan$covtheta,1,lower.tail=F)
UVTWAS[i,5]=fitModifier$theta
UVTWAS[i,6]=pchisq(fitModifier$theta^2/fitModifier$covtheta,1,lower.tail=F)
}, error=function(e){
errorindicate=1
})
if(errorindicate == 1) next
}
```

The structure of `UVTWAS` is:

```
head(UVTWAS)
```

```
##                   Variable Type        Est1        P1       Est2        P2
## 1 ACOT11+Adipose_Visceral eQTL -0.40863663 0.7895277 -0.8790641 0.2311542
## 2    ACOT11+Adrenal_Gland eQTL -0.96258731 0.7147184 -1.1510603 0.3627408
## 3     ACOT11+Artery_Aorta eQTL  0.14092413 0.9181448  0.2025135 0.7569147
## 4    ACOT11+Artery_Tibial eQTL -0.01711601 0.9835635  0.1912081 0.6302157
## 5 ACOT11+Brain_Cerebellum eQTL -0.93448220 0.8659363  0.5692827 0.8332270
## 6     ACOT11+Brain_Cortex eQTL -0.40462949 0.8721248 -0.2022807 0.8692957
```

As described in the manuscript, we only consider data with an P-value greater than 0.5. We systematically scan each locus of the GWAS trait.

```
Genelist=UVTWAS%>%mutate(pvthres=0.05) %>%
          dplyr::filter(P1 < pvthres | P2 < pvthres) %>%
          pull(Variable)%>%unique()
```

This leaves $p = 63$ gene-tissue pairs:

```
length(Genelist)
```

```
## [1] 63
```

We then organize the data:

```
bX=bX[,Genelist]
bXest=bXest[,Genelist]
bY=eQTLsQTL[,c("SNP","Zscore.y")]%>%as.data.frame(.)
bY=bY[!duplicated(bY$SNP),]
rownames(bY)=bY$SNP
bY=bY[genosnp,]
```

2.7 Performing TGVIS

Before executing TGVIS, we do not recommend removing potential candidates for direct causal variants. Specifically, if a gene is solely an eQTL, we suggest not including these eQTLs as candidates for direct causal variants. We have defined a function, `findUniqueNonZeroRows()`, to identify the indices of these variants. The code for this function will be provided in GitHub.

Next, we execute TGVIS:

```
fittgvis=tgvis(by=bY$Zscore.y,bXest=bXest,LD=R0,Noutcome=mean(LDL$N),
                  L.causal.vec=c(1:10),varinf.upper.boundary=0.25,pip.min=0.1,
                  pleiotropy.rm=findUniqueNonZeroRows(bXest))
```

Finally, we organize the results, which is a bit more complex. The principle is to retain only those gene-tissue pairs and direct variants that are included in the 95% credible set.

```
thetagamma=c(fittgvis$theta[which(fittgvis$theta!=0)],
         fittgvis$gamma[which(fittgvis$gamma!=0)])
if(length(thetagamma)>0){
se.mrjones=c(fittgvis$theta.se[which(fittgvis$theta!=0)],
         fittgvis$gamma.se[which(fittgvis$gamma!=0)])
pip.mrjones=c(fittgvis$theta.pip[which(fittgvis$theta!=0)],
          fittgvis$gamma.pip[which(fittgvis$gamma!=0)])
pratt.mrjones=c(fittgvis$theta.pratt[which(fittgvis$theta!=0)],
            fittgvis$gamma.pratt[which(fittgvis$gamma!=0)])
cs.mrjones=c(fittgvis$theta.cs[which(fittgvis$theta!=0)],
         fittgvis$gamma.cs[which(fittgvis$gamma!=0)])
cs.pip.mrjones=c(fittgvis$theta.cs.pip[which(fittgvis$theta!=0)],
             fittgvis$gamma.cs.pip[which(fittgvis$gamma!=0)])
TGVIResult=data.frame(Variable=names(thetagamma),estimate=thetagamma,
                  se=se.mrjones,pip=pip.mrjones,pratt=pratt.mrjones,
                  cs=cs.mrjones,cs.pip=cs.pip.mrjones)
TGVIResult$Type=c(rep("TissueGene",length(which(fittgvis$theta!=0))),
              rep("SNP",length(which(fittgvis$gamma!=0))))
TGVIResult$CHR=1
TGVIResult$BP=55.5e6
TGVIResult=TGVIResult%>%group_by(Type, cs)%>%mutate(cs.pratt=sum(pratt))%>%ungroup()
}else{
TGVIResult=NULL
}
TGVIResult$xQTL=TGVIResult$Variable
if(sum(TGVIResult$Type=="TissueGene")>0){
TGVIResult$xQTL[which(TGVIResult$Type=="TissueGene")]=
get_nonzero_rows(bXest,TGVIResult$Variable[which(TGVIResult$Type=="TissueGene")])$NonzeroRows
}
row.names(TGVIResult)=NULL
TGVIResult=TGVIResult%>%dplyr::select(Variable,cs,cs.pip,cs.pratt,xQTL,
          CHR,BP,Type,estimate,se,pip,pratt)%>%arrange(.,cs,Type,Variable)
print(TGVIResult)
```

```
## # A tibble: 3 x 12
##   Variable      cs cs.pip cs.pratt xQTL    CHR     BP Type  estimate    se   pip
##   <chr>      <dbl>  <dbl>    <dbl> <chr> <dbl>  <dbl> <chr>    <dbl> <dbl> <dbl>
## 1 rs11591147     1      1   0.492  rs11~     1 5.55e7 SNP     -71.8  1.02       1
## 2 PCSK9+Who~     2      1   0.170  rs12~     1 5.55e7 Tiss~     5.61 0.154      1
## 3 rs11206517     3      1   0.0398 rs11~     1 5.55e7 SNP      19.4  0.989      1
## # i 1 more variable: pratt <dbl>
```

2.8 The Pratt indices of gene-tissue pairs, direct causal variants, and infinitesimal effect

We have defined a function, `R2_partition`, to calculate the Pratt indices for gene-tissue pairs, direct causal variants, and infinitesimal effects.

```
TGVIPratt=R2_partition(y=bY$Zscore.y,LD=R0,
                   eta.theta=c(R0%*%bXest%*%fittgvis$theta),
                   eta.gamma=c(R0%*%fittgvis$gamma),
                   eta.upsilon=c(R0%*%fittgvis$upsilon))
TGVIPratt
```

```
##          r1        r2        r3        r4
## 1 0.9466261 0.1675219 0.5299218 0.2491823
```

Here, $r1$ is the total Pratt index, $r2$ is the Pratt index of gene-tissue pairs, $r3$ is the Pratt index of direct causal variants, and $r4$ is the Pratt index of infinitesimal effects.

3 Tuning parameter selection

3.1 Tuning parameters in xQTL selection

In xQTL selection, the function we employ is `eQTLmapping_susie`. The useage of it is

```
eQTLmapping_susie(
  bX,
  LD,
  Nvec,
  pip.thres = 0.5,
  pip.min = 0.2,
  L = 5,
  resample = F,
  sampling.time = 100
)
```

The related arguments are

- bX: A matrix of Z-scores of marginal eQTL effect estimates for tissue-gene pairs.
- LD: The LD matrix of variants.
- Nvec: A vector representing the sample sizes of tissue-gene pair eQTL studies.
- pip.thres: A threshold for individual PIP when no credible set is found. Default is `0.2`.
- pip.min : The minimum individual PIP in each 95% credible set. Used to remove variables with low PIPs within credible sets. Default is `0.05`.
- L: The number of single effects to be used in the SuSiE model. Default is `5`.
- resample: A logical value indicating whether to resample eQTL effects. Default is `FALSE`.
- sampling.time: The number of resampling iterations to perform when 'resample' is `TRUE`. Default is `100`.

First, we do not recommend performing resampling which will result similar estimate to `coef(fit)[-1]`. Besides, we do not consider `pip.thres` as a critical parameter, because it will be used only when SuSiE cannot find any 95% credible set.

The number of single effects (`L`) and the minimum individual PIP within each 95% credible set (`pip.min`) are two important tuning parameters. The choice of `L` is based on prior knowledge regarding the maximum number of causal or informative xQTLs; therefore, we set `L = 3` in our analysis. As for `pip.min`, a smaller

Table S1: Effect of tuning parameters (L and minimum PIP) on xQTL selection.

Variable	CS Pratt	Number of xQTLs	Minimum PIP	L
rs11591147	0.5103	1	0.00	3
PCSK9+Whole_Blood	0.1920	347	0.00	3
rs11206517	0.0405	1	0.00	3
rs11591147	0.5103	1	0.00	5
PCSK9+Whole_Blood	0.1933	350	0.00	5
rs11206517	0.0405	1	0.00	5
rs11591147	0.5102	1	0.05	3
PCSK9+Whole_Blood	0.1899	3	0.05	3
rs11206517	0.0405	1	0.05	3
rs11591147	0.5101	1	0.05	5
PCSK9+Whole_Blood	0.1909	4	0.05	5
rs11206517	0.0405	1	0.05	5
rs11591147	0.5103	1	0.25	3
PCSK9+Whole_Blood	0.1803	2	0.25	3
rs11206517	0.0404	1	0.25	3
rs11591147	0.5103	1	0.25	5
PCSK9+Whole_Blood	0.1803	2	0.25	5
rs11206517	0.0404	1	0.25	5

threshold may result in larger credible sets containing highly correlated xQTLs, which complicates the interpretation.

We examined the effect of different minimum PIP thresholds (0, 0.05, and 0.25) and different numbers of assumed single effects (L = 3 and 5) using LDL-PCSK9 as an example. The results are summarized in Table 2. We observed that varying the value of L had minimal impact, likely because (1) the number of eQTLs for PCSK9 in blood is inherently small (≤ 3), and (2) SuSiE is known to be robust when L exceeds the true number of causal variants. However, increasing the minimum PIP threshold to 0.25 reduced the number of selected eQTLs from 3 to 2, compared to when the threshold was 0.05. As a result, the CS-Pratt value for PCSK9 in blood decreased by approximately 1%.

3.2 Tuning parameters in poet shrinkage estimate

When the dimension of the LD matrix is large (e.g., $m > 100$), we suggest using the POET method to regularize such an LD matrix (https://academic.oup.com/jrsssb/article/75/4/603/7075932). Specifically, POET considers an eigenvalue decomposition of the sample LD matrix of individuals' genotypes:

$$\hat{\mathbf{R}} = \sum_{k=1}^{m} d_k \mathbf{U}_k \mathbf{U}_k^\top = \sum_{k=1}^{K} d_k \mathbf{U}_k \mathbf{U}_k^\top + \mathbf{E},$$

where $d_1 \geq d_2 \geq \cdots \geq d_m$ are the eigenvalues of $\hat{\mathbf{R}}$, $\mathbf{U}_k$ is the corresponding eigenvector of d_k, K is a cutoff, and $\mathbf{E}$ is the residual matrix. To improve the condition of $\hat{\mathbf{R}}$, the standard POET applies a covariance-thresholding method on $\mathbf{E}$, while we considered a linear shrinkage of $\mathbf{E}$:

$$\tilde{\mathbf{E}} = \alpha \mathbf{E} + (1-\alpha)\mathrm{diag}(\mathbf{E}).$$

The extended POET estimate was:

$$\tilde{\mathbf{R}} = \sum_{k=1}^{K} d_k \mathbf{U}_k \mathbf{U}_k^\top + \tilde{\mathbf{E}}.$$

We use $\tilde{\mathbf{R}}$ in the corresponding data.

We utilized the Dynamic Eigenvalue Difference Ratio (DDR) (https://ssrn.com/abstract=2827558) to select K:

$$K = \arg \min_{K_{\min} \leq k \leq K_{\max}} \frac{d_k - d_{k+1}}{d_{k+1} - d_{k+2}},$$

where $K_{\min}$ and $K_{\max}$ are two tuning parameters (default to 1 and $\min(100, m/4)$, respectively).

On the other hand, we adopted finite sample positive definiteness for the selection of α:

$$\alpha = \inf\{\alpha : \text{minimum eigenvalue of } (\alpha\mathbf{E} + (1-\alpha)\text{diag}(\mathbf{E})) > \tau\},$$

where τ (default to 0.001) was a given tolerance.

In TGVIS, the usage of POET is

```
poet_shrinkage(
  LD,
  KMax = min(15, round(nrow(LD)/2)),
  lamvec = seq(0.025, 0.25, by = 0.025),
  minvalue = 0.001
)
```

and the arguments are

- LD: A LD matrix.
- KMax: The maximum number of latent factors used in POET. Default is `min(15, round(nrow(LD)/2))`.
- lamvec: A vector of candidate shrinkage parameters. Default is `seq(0.025, 0.25, by = 0.025)`.
- minvalue: The threshold for the minimum eigenvalues used in determining the optimal shrinkage parameter. Default is `1e-3`.

We believe the default settings are robust and one may only change KMax accordingly. For example, if the dimension of the LD matrix is large, one may increase `m` to 100. Nevertheless, as the DDR method determines the effective eigenvalue cutoff, this parameter is generally not of major importance.

3.3 Tuning parameter in S-PrediXcan and its modifer

The usage of our S-Prdixcan function is

```
modified_predixcan(
  by,
  bxest,
  LD,
  tauvec = seq(3, 10, by = 1),
  rho.gamma = 1.5,
  max.iter = 15,
  max.eps = 0.005,
  ebic.factor = 2,
  normmax = 2,
  pleiotropy.rm = NULL
)
```

and the arguments are - by: A vector of Z-scores of marginal effects from the outcome GWAS (same as in ctwas).

- bxest: A vector of direct xQTL effect estimates for a tissue-gene pair.
- LD: The LD matrix of variants (same as in ctwas).
- tauvec: A vector of tuning parameters used in penalizing the direct causal effect. Default is 'seq(3,10,by=1)'.
- rho.gamma: A parameter set in the ADMM algorithm. Default is 1.5.
- max.iter: The maximum number of iterations for the ADMM algorithm. Default is 15.
- max.eps: The convergence tolerance for the ADMM algorithm. Default is 0.005.
- ebic.factor: The extended BIC factor for model selection. Default is 2.
- pleiotropy.rm: A vector of indices specifying which variants should not be considered as having direct causal effects.

When users intend to apply standard S-PrediXcan without accounting for horizontal pleiotropy, they can simply set `tauvec` to a large value, such as 10000 or any other sufficiently large number. This effectively ignores the influence of pleiotropy. Conversely, when users aim to control for horizontal pleiotropy effects, we recommend using the default settings, which automatically identify variants exhibiting substantial horizontal pleiotropic effects. Additionally, moderately increasing `rho.gamma` (e.g., setting `rho.gamma = 3`) may slightly improve performance based on our evaluations. Finally, `ebic.factor` serves as a penalty parameter that regulates the number of variants with horizontal pleiotropy; a larger `ebic.factor` will lead to sparser results by imposing stronger penalties.

3.4 Tuning parameters in TGVIS

The usage of TGVIS is

```
tgvis(
  by,
  bXest,
  LD,
  Noutcome,
  L.causal.vec = c(1:8),
  max.iter = 50,
  max.eps = 0.001,
  susie.iter = 500,
  pip.thres.cred = 0.95,
  eigen.thres = 1,
  varinf.upper.boundary = 0.25,
  varinf.lower.boundary = 0.001,
  ebic.beta = 1,
  ebic.upsilon = 1,
  pip.min = 0.05,
  pv.thres = 0.05,
  pleiotropy.rm = NULL,
  prior.weight.theta = NULL,
  prior.weight.gamma = NULL
)
```

and the arguments are

- by: A vector of Z-scores of the marginal effects from the outcome GWAS.
- bXest: A matrix of direct effect estimates based on the Z-scores of tissue-gene pairs.
- LD: The LD matrix of variants.
- Noutcome: The sample size of the outcome GWAS.
- L.causal.vec: A vector of candidate numbers of single effects used in BIC. Default is 'c(1:8)'.
- max.iter: The maximum number of iterations for the profile-likelihood algorithm. Default is 50.
- max.eps: The convergence tolerance for the profile-likelihood algorithm. Default is 1e-3.
- susie.iter: The maximum number of iterations for 'susie_rss' within the profile-likelihood algorithm. Default is 50.
- pip.thres.cred: The cumulative PIP threshold for variables in a credible set. Default is 0.95.
- eigen.thres: The threshold of eigenvalues for modelling the infinitesimal effect. Default is 1.
- varinf.upper.boundary: The upper boundary for the prior variance of infinitesimal effects, multiplied by var(y) to adapt to different locus variances. Default is 0.25.
- varinf.lower.boundary: The lower boundary for the prior variance of infinitesimal effects, not multiplied by var(y). Default is 0.001.
- ebic.beta: The extended BIC factor for causal effects of tissue-gene pairs and direct causal variants used in BIC computation. Default is 1.
- ebic.upsilon: The extended BIC factor for infinitesimal effects used in BIC computation. Default is 1.
- pip.min: The minimum PIP threshold for individual causal effects in the profile-likelihood. This is used to specify which tissue-gene pairs and direct causal variants to include in the score test of variance of infinitesimal effects. Default is 0.05.
- pv.thres: The p-value threshold for the score test. Default is 0.05.
- pleiotropy.rm: A vector of indices specifying which variants should not be considered as having direct causal effects.
- prior.weight.theta: A vector of prior weights of gene-tissue pairs, which will be used as input in SuSiE. Default is NULL.
- prior.weight.gamma: A vector of prior weights of direct causal variants, which will be used as input in SuSiE. Default is NULL.

We do not recommend modifying `eigen.thres` from its default value of 1, particularly avoiding smaller values. Our evaluations indicate that reducing `eigen.thres` may cause infinitesimal effects to contribute excessively to the outcome variance. The parameter `varinf.upper.boundary` also controls the contribution of infinitesimal effects; we set this value to 0.25, meaning that the prior variance of infinitesimal effects will not exceed 25% of the local variance of the outcome.

The parameters `L.causal.vec`, `max.iter`, `max.eps`, and `susie.iter` are generally insensitive. Users may retain their default values or adjust them as needed. In cases where SuSiE fails to converge within 50 iterations, it is advisable to increase `susie.iter` appropriately.

The parameters `ebic.beta` and `ebic.upsilon` control the penalization of degrees of freedom for gene-tissue pairs and infinitesimal effects, respectively. We recommend setting these parameters to either 0 or 1.

When `prior.weight.theta` and `prior.weight.gamma` are set to NULL, the algorithm applies uniform prior weights by default, as follows:

```
if (is.null(prior.weight.theta) == T){
prior.weight.theta = rep(1/p, p)
}
if (is.null(prior.weight.gamma) == T) {
prior.weight.gamma = rep(1/length(pleiotropy.keep), length(pleiotropy.keep))
}
else{
prior.weight.gamma = prior.weight.gamma[pleiotropy.keep]
}
prior_weights = c(prior.weight.theta, prior.weight.gamma)
```